Meta5

Transformers Guide

Version 4.3

Note

Before using this information and the product it supports, be sure to read the general information under “Notices”.

Third Edition (January 2013)

This edition applies to Version 4 Release 3 Modification Level 3 of Meta5, and to any subsequent releases until otherwise indicated in new editions or technical newsletters. Make sure you are using the correct edition for the level of the product.

Order publications through your Meta5 representative or the Meta5 branch office serving your locality.

Contents

Notices v
Trademarks vii

About This Book ix
Who Should Read This Book ix
We'd Like Your Comments ix

Chapter 1. Getting Started with Transformers 1
Looking at a Transformer Window. 1
Looking at a Transformer Controls Window . . . 3
Managing Data in a Transformer. 8
Running a Transformer 9
Using a Transformer in a Capsule Application . 9
Configuring a Blank Transformer. 13

Chapter 2. Basic Transformers. . . . 15
Append 16
Clean. 20
Clear Contents 23
Compress 31
Copy Icon 35
Join 43
Label 52
Merge 58
MultiJoin 73
Pivot 82
Select 92
Sort 94

Chapter 3. Advanced Transformers 101
Append Message 103
Concatenate 104
Conditional Clean 106
Data Format 112
Date Format 121
Error Log 1 and Error Log 2 126
Function. 132
Header. 137
Message 141
Multiple Select. 143
Multiple Split 147
Page Break 150
Period Table 155
Post Message 163
Random Number. 164
Replace 166
Row Clean. 172
Row Select 177
Split Header. 183
Substitute. 186
Subtotal 189
Switch and Text Switch 194
Text to Spreadsheet. 196
Transpose 201
Word Count 202
Write SQL 204

Chapter 4. Significance and Sample Testing Transformers. 211
ANOVA 212
ChiSquare 231
CrossTab 245
IPMean 263
Kruskal. 277
KSTest 290
NPCorrelation 309
NPIndependent 321
NPPaired 331

Chapter 5. Regression and Time Series Analysis Transformers. . . 345
AutoCorrelation 346
AutoRegression. 365
Correlation. 385
Elementary 398
Forecast. 411
Moving. 440
Regression 454
Seasonality 474
Identifying Seasonality Methods 490
Seasonality Transformer Formulas 492
Using the Seasonality Transformer with the Forecast Transformer 495

Chapter 6. Transformer Execution Language 497
Installing a TXL Program. 498
TXL Elements and Syntax 498
String Manipulation Functions 514

Chapter 7. Time Series Analysis. . 521

Glossary 523

Meta5 Publications 539

Index 541

Notices

References in this publication to Meta5 products, programs, or services do not imply that Meta5 intends to make these available in all countries in which Meta5 operates. Any reference to a Meta5 product, program, or service is not intended to state or imply that only that Meta5 product, program, or service may be used. Subject to Meta5's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the Meta5 product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by Meta5, are the responsibility of the user.

Meta5 may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Meta5, Inc.
122 West Main Street
Babylon, NY 11702
U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling (1) the exchange of information between independently created programs and other programs (including this one) and (2) the mutual use of the information that has been exchanged, should contact:

Meta5, Inc.
122 West Main Street
Babylon, NY 11702
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Various components of the Meta5 software system require the presence of one or more third party software products, including but not limited to Microsoft Excel for Meta5's Xlaunch, Xtract software product, Microsoft Word for Meta5's WordTool software product, Microsoft Outlook for Meta5's SendOut software product, Brio's BrioQuery for Meta5's AutoBrio software product, Business Objects' BusinessObjects for Meta5's BOConnect software product and/or IBM's Lotus Notes for SendNts software product. No license is granted by Meta5 to our customers for the use of these third party programs in conjunction with our software products. Furthermore, it is the obligation of our customers to ascertain whether it has sufficient licenses from these third parties, for the third party software in order to utilize the above mentioned Meta5 software programs.

Trademarks

The following terms are trademarks of Meta5, Inc. in the United States or other countries or both:

Meta5

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, Windows2000, Windows 98, Windows ME, Windows XP, Windows NT, Excel, Word, Outlook and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Brio and Brio Query are registered trademarks of Brio Software Inc.

BusinessObjects is a trademark of Business Objects SA.

Lotus Notes and Lotus 123 are Trademarks of Lotus Software, a subsidiary of IBM Corporation.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

About This Book

This document contains information about the components, features, and functions of the transformer tools found in the Meta5 Developer’s Desktop.

Who Should Read This Book

This book is for users who want to use transformers to:

- Develop applications within the Meta5 environment
- Build capsules and manipulate data
- Use advanced data analysis capabilities of Meta5

To use transformers, you should be familiar with the Meta5 desktop management tools and with basic Meta5 product terminology. You should also be familiar with the tools that provide source data to transformers.

To use regression and time series transformers, you should be familiar with analytical techniques.

To use statistical transformers, you should be familiar with statistical techniques.

We'd Like Your Comments

Your feedback is important to us in providing you with the most accurate and high-quality information. If you have any comments about this book or any other Meta5 documentation, please visit our website at:

```
http://www.meta5.com
```

There you'll find a feedback page where you can enter comments and send them to us.

Chapter 1. Getting Started with Transformers

Meta5 provides five types of transformers:

- Basic Transformers
- Advanced Transformers
- Significance and Sample Testing Transformers
- Regression and Time Series Analysis Transformers

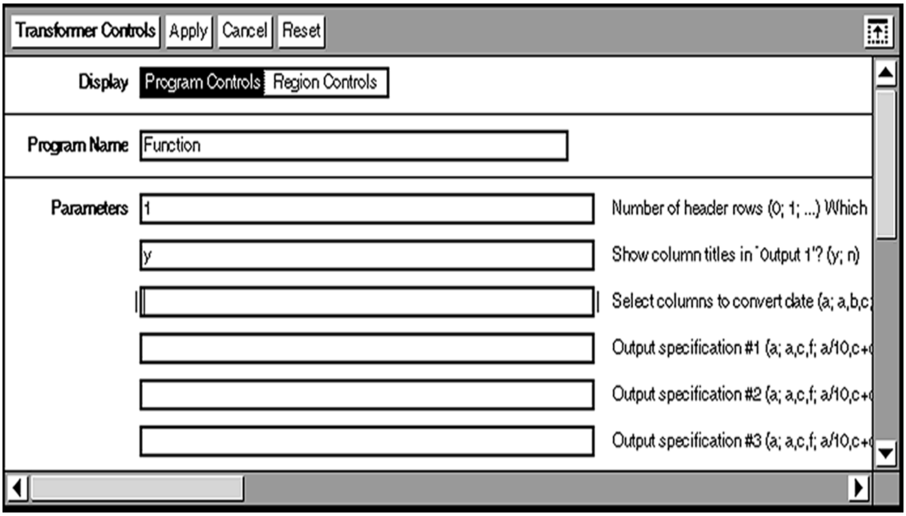
Each type of transformer is designed to perform a specific set of tasks or analyses. Although each transformer performs a different function, all transformers are similarly designed.

This chapter covers the basic information needed when working with transformers:

- “Looking at a Transformer Window”
- “Looking at a Transformer Controls Window” on page 3
- “Managing Data in a Transformer” on page 8
- “Running a Transformer” on page 9

Looking at a Transformer Window

All transformer windows include a controls area and a display area, shown in Figure 1 on page 2.



Display: Program Controls | Region Controls

Program Name: Function

Parameters:

1 — Number of header rows (0; 1; ...) Which

y — Show column titles in "Output 1"? (y; n)

Select columns to convert date (a; a,b,c;

Output specification #1 (a; a,c,f; a/10,c+

Output specification #2 (a; a,c,f; a/10,c+

Output specification #3 (a; a,c,f; a/10,c+

Figure 1. Example of transformer window

Each transformer window header contains the transformer name and the following buttons:

- The `Run` button, which starts the data-manipulation process performed by the transformer. For detailed information on this process, see “Running a Transformer” on page 9.
- The `Show Controls` button, which opens the Transformer Controls window where you can specify parameters for the transformer. For detailed information on this window, see “Looking at a Transformer Controls Window” on page 3.

When you scroll up or down in the transformer window, the controls area remains in position while the display area moves to display different data.

Controls Area

The controls area of a transformer window contains two fields that provide the following information:

- The `Program Name` field, which identifies the type of program run by the transformer, for example, Row Clean. This field is particularly useful if the icon name changed.
- The `Display Data For` field, which determines the input and output data shown in the display area.

  An *input region* is the storage area for data that is read or transferred into a transformer. The number of input region choices varies from one transformer to another.

The *output region* is a holding area for the output data generated by a transformer. The number of output choices varies from one transformer to another. In the example shown in Figure 1, the Row Clean transformer has one output choice named Results. Other transformers might have more output choices (or none), each individually named for easy identification. You can move output data to other icons by copying it directly or by transferring the information using a capsule arrow. Portions of the output can be viewed in the transformer display area.

Display Area

Use the display area to copy or move data from other icons, either manually or through a capsule application.

The following message is present in the display area the first time you open a transformer:

```
Copy input data here, if desired.
```

After you copy or move data to a transformer window, or after you run a transformer, the result is shown within a frame in the display area:

a, b, c, d

Join columns in first table (a; c, a, b, d; ¼

If only portions of data are shown in the display area, you can scroll the transformer window to view data in the display area.

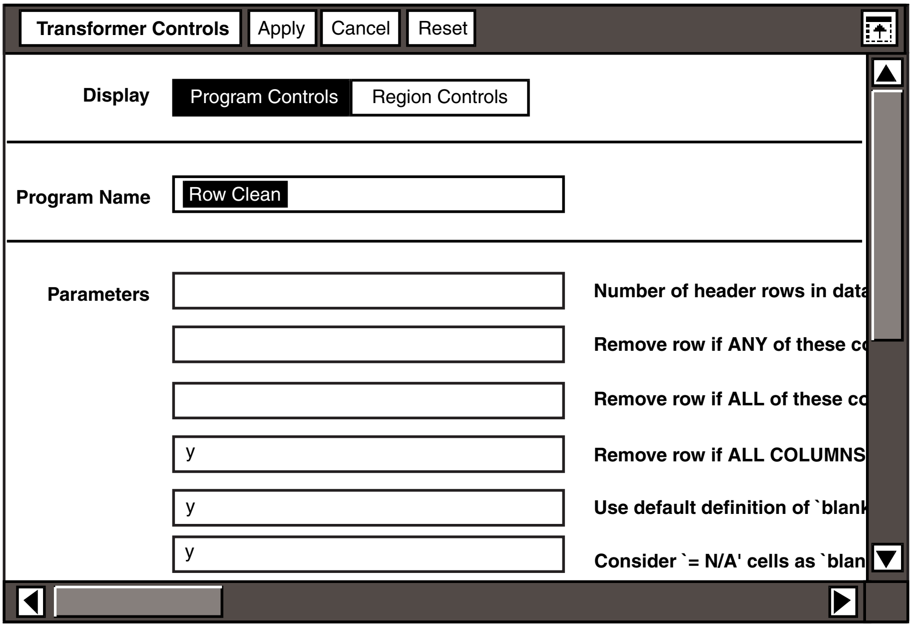
The width of each column in the display area accommodates the size of the desktop font, which is specified in the desktop Set Preferences window. It is not determined by the width of the columns in the source icon.

For example, if you set the system font to Helvetica 18 and the date format to Long, the width of each column in the transformer's display area adjusts to accommodate the larger font size and date format.

Looking at a Transformer Controls Window

From each transformer window, you can display a Transformers Controls window where you define the transformer parameters. You select the data you want to manipulate by entering the appropriate information in the Transformer Controls window.

To open the Transformer Controls window, click on the `Show Controls` button in the transformer window header. The Transformer Controls window opens, as shown in Figure 2.



Display: [✓ Program Controls] Region Controls

Program Name: Row Clean

Parameters

	Number of header rows in data
	Remove row if ANY of these co
	Remove row if ALL of these co
y	Remove row if ALL COLUMNS
y	Use default definition of `blank
y	Consider `= N/A' cells as `blan

Figure 2. Transformer Controls window

Program Controls Window

When you open the Transformer Controls window, the `Program Controls` choice is highlighted in the `Display` field. The `Program Controls` choice also displays this page if you select it from the Region Controls choice page. The `Region Controls` choice, `Program Name`, and `Parameters` fields also are displayed in this window.

Display field

Contains the `Program Controls` and the `Region Controls` choices. The `Region Controls` choice displays the `Input` and `Output Region Names` fields.

Program Name field

Identifies the type of program run by the transformer. The type of program in the `Program Name` field determines a particular function performed by the transformer and the parameters and the fields in the transformer windows.

Parameters fields

Identify the data passed by a user to a program to control processing in some way. Each parameter field is followed by a brief description and examples shown in parentheses. Throughout this guide, examples are separated by semicolons. List separators (such as a comma) in an example indicate that you can enter more than one value into that field.

In Meta5, the default decimal separator in the `Country` choice in the `Set Preferences` menu is a period. If the decimal separator in your desktop setting is set as a comma (for example, 12.34 is written as 12,34), then whenever you use a comma in an example, the comma must be followed by a space.

Except in the Text to Spreadsheet transformer, list separators in all transformers remain unchanged when you upgrade your transformers. If you previously specified a semicolon as a list separator in the Text to Spreadsheet transformer, the semicolons converted to commas when you upgrade the icon. You must manually change the commas to semicolons.

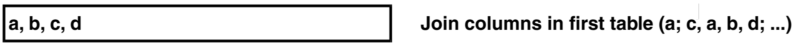
Setting Parameters in a Transformer

The `Parameters` fields specify the function performed by the transformer, the data that is retrieved from the input regions, and the arrangement of data in the output region.

Each parameter is followed by a brief description and examples, which are shown in parentheses.

The following example shows some parameter fields, followed by a brief description and several possible parameter values, which are enclosed in parentheses. The examples are separated by semicolons.

If a parameter accepts multiple values, you must use a comma to separate the values; otherwise, the list will not parse correctly.



a, b, c, d **Join columns in first table (a; c, a, b, d; ...)**

Unless specified, additional spaces and capitalization have no effect in the Transformer Controls window; you need enter only the first letter of a parameter. For example, `y` can be typed in place of `Yes`. To enter a string of digits as a text string, place a backslash before the first digit. For example, 234 is the number 234, and \234 is the text string 234.

To set or change the `Parameters` fields in a transformer:

1. Click on the `Show Controls` button in the window header of the transformer.

   The Transformer Controls window opens to display the Program Name and default parameters settings.

2. Enter the values for the parameters (if the field is blank), or delete the displayed values and enter new ones.

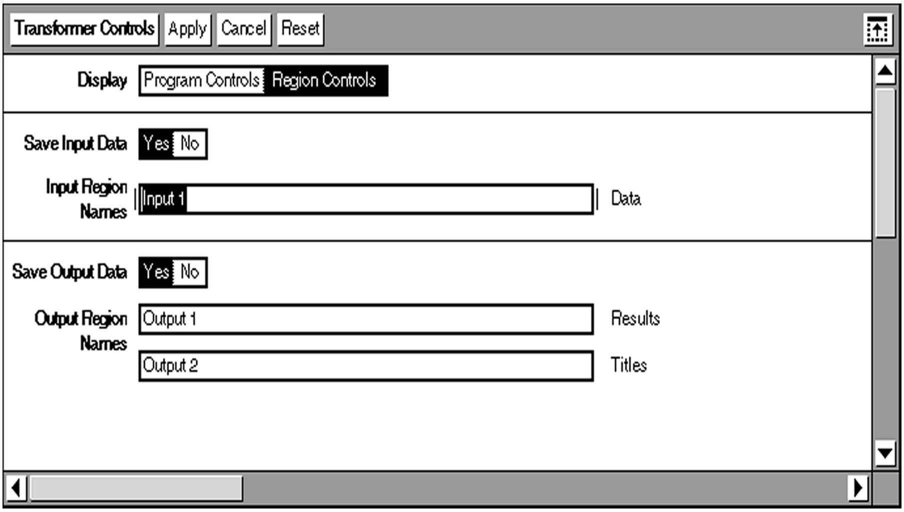
   Values in the Parameters fields are not case-sensitive.

   You can also use @-variables when the transformer is inside a capsule application. See the *Capsule User's Guide* for information about the Capsule icon's operation and @-variables.

3. Close the Transformer Controls window.

Region Controls Window

When you select the `Region Controls` choice in the Transformer Controls window, the window shown in Figure 3 opens.



Display Program Controls [✓ Region Controls]

Save Input Data [✓ Yes] No

Input Region Names: Input 1 — Data

Save Output Data [✓ Yes] No

Output Region Names: Output 1 — Results

Output 2 — Titles

Figure 3. Region Controls window

The `Input Region Names` choice differs from one transformer to another. Some transformers, such as the Message transformer, do not have an `Input Region Names` choice. In the example shown in Figure 3, the Input Region Name is called Data. The input supplied to Data consists of data read from a Spreadsheet, Query, or SQL Entry icon, or information copied directly into the transformer.

If a transformer has only one input region, you do not need to specify a destination region.

The `Output Region Names` choices vary from one transformer to another. In the example shown in Figure 3, the Row Clean transformer has one output region called Results. Output names are usually labeled to reflect the corresponding data that is shown in the display area.

Changing Region Names

After you enter a Program Name and apply the changes, you can modify the names of the input and output regions to reflect the name of the corresponding data in the display area.

To view or change the region names, select Region Controls in the Display field of the Transformer Controls window. The Region Controls window opens, as shown in Figure 3 on page 6.

Each field is specified as an input or output region, and gives a description of the data that is intended for that field. Some transformers have only one region; others have more.

The names for the input and output regions are usually preset by the system administrator. However, if you change the region names for your applications, choose names that reflect the type of data that will be shown within the display area. Each region name should be brief; you will use these names when you use the transformer in a capsule application.

Saving or Clearing Data in a Region

Data is saved in the input and output regions after it is processed by a transformer. This can be useful for troubleshooting the data flow in a transformer. If a transformer is within the framework of a capsule application, data is not retained in the last input region.

If you are interested in the results only, you can automatically clear the data from the input and output regions after the transformer is run. This minimizes the disk space required to store the data.

If you manually copy or move data into a transformer that is set to clear each region after the transformer is run, the data remains in the regions until you close the window.

To determine whether data is saved in the regions:

1. Click on the Show Controls button in the window header of a preconfigured transformer.

   The Transformer Controls window opens.

2. Select Region Controls in the Display field.

   The Region Controls window opens.

3. Select the appropriate choice in the Save Input Data field.

   This selection determines whether data is saved in all input regions. You cannot save data in one region and clear the data from another region.

4. Select the appropriate choice in the Save Output Data field.

   This selection determines whether data is saved in all output regions.

If you connect a transformer to an Out icon in a capsule application and run the capsule application, data in all of that transformer's output regions is saved automatically. Also, the Save Output Data field will be set to *Yes*, regardless of what you initially set in that field.

Managing Data in a Transformer

You can transfer data into a transformer from a Query, Data Entry, Spreadsheet, Reporter, or SQL Entry icon, or from another transformer. Data that you intend to transfer from one icon to another must have a consistent format for a transformer to run correctly. For example, within a column of numeric data, if one cell entry contains a nonnumeric character (such as a space), then either the nonnumeric character must be removed, or all data cells within the column must contain the text data type. Otherwise, the transformer will generate an error message.

If your first row contains headings, make sure that the first cell type is Text. The tables that you specify must not contain a formula in the first cell unless you want the first row to be treated as data. In a spreadsheet, all empty cells use a default formula cell type.

Copying or Moving Data into a Transformer

To copy or move data from other tools such as Spreadsheet or Text:

1. Open the transformer icon.

   You can also copy or move data to a closed transformer icon. This opens the icon, and the source data is placed in the first input region.

   If you are using a blank Transformer icon, be sure to include the program name in the `Program Name` field of the Transformer Controls window. Otherwise, you cannot transfer data into the transformer.
2. Select the appropriate input region in the `Display Data For` field of the transformer window.
3. Open the source icon that contains the data that you want to use.
4. Select the data from the source icon window.

   If the source data is from a spreadsheet, and the first row contains headings, be sure that the cell type is set to Text. The first row of the specified data must not contain a formula unless you want the first row to be treated as data.
5. Press the Copy or Move function key.
6. Click inside the input region display area of the transformer.
7. Repeat steps 2 through 6 for any additional input data.

Copying or Moving Data from a Transformer

You can copy or move transformer data to another transformer for additional format manipulation, or to another tool (such as the Spreadsheet or Text tools) for incorporating text or other data, data manipulation, printing, additional formatting, or viewing all the data.

To transfer data from a transformer:

1. Select the appropriate input or output region in an open transformer window.

2. Click on the `Select All` button in the display area of the transformer window.
3. Press the Copy or Move function key.
4. Click inside the open window of the destination icon.

Running a Transformer

Each transformer has unique parameters that determine how the transformer runs. Some parameter fields have default settings. See the chapter for the specific transformer for information on its parameter fields.

To run a transformer, click the `Run` button in the transformer window header. When the process is complete, the output region name is highlighted, the results are shown in the transformer display area, and the following message is displayed in the message area:

```
Transformer completed successfully. -- Continue.
```

You cannot stop the run process after you click the `Run` button.

The Transformer Controls window can remain open while the transformer is running.

If the transformer is inside a capsule application, click the `Run` button in the Capsule window header. This processes the entire application and, when processing is complete, the following message is displayed in the message area:

```
Complete. -- Please Continue.
```

For more information on running a Capsule icon, see the *Capsule User's Guide*.

Using a Transformer in a Capsule Application

You can also use a transformer in a capsule application to process data. Be aware of the following items when you use a transformer in a capsule application:

- The Capsule tool reads input tables from left to right, one row at a time, so be sure that you place the source icons in the order in which you want the tables processed.
- The transformer starts processing as soon as it reads the data in the last input region. Be sure that the icon containing the source data for the last input region is properly placed in the capsule application, and that the destination area is properly identified in the Arrow Options window. For information on setting arrow options, see “Setting Arrow Options” on page 11.
- When you move data between two transformers, and the source transformer has more than one output region, you must specify (in the Arrow Options window) which output region from the source transformer is to be moved to

the input region of the destination transformer. If you do not specify the destination input region, data will not transfer.

- The data for the last input region is not retained in the transformer display area. If you open a transformer after running a capsule application, the last input region is blank. A message is displayed in the transformer window stating that the data was used, but not stored.
- Columns are displayed left-justified when transferred from a source icon to a transformer.
- When running a transformer in a capsule application, you can enter @-variables into the `Parameters` fields of a transformer. The @-variable values are retrieved from the User Input Control window of the Capsule icon when you run the application. For information on using @-variables in a capsule application, see the *Capsule User's Guide*.

Naming Icons

You can use @-variables and @-keywords to name icons. The variables are expanded when a new icon is generated during capsule processing. After the icon is processed, the icon name returns to the original unexpanded form so that it can be reused. If you want to keep a copy of the icon that has the expanded name, copy it to an output container, such as a folder or an envelope.

Naming icons is useful when you are performing iterations and want each final icon to be identifiable by the data associated with it. For example, if a capsule application run generates several spreadsheet iterations of regional sales information, each spreadsheet icon it generates gives data for a different sales region. To ensure that the output folder contains several spreadsheets with different sales regions as the icon names instead of all the icons having the name Spreadsheet, enter an @-variable as the spreadsheet's icon name. When each spreadsheet is active during the capsule application run, the @-variable is expanded so that the sales region becomes the icon's name. After the icon is copied to the results folder and is therefore no longer active, the icon name returns to the unexpanded @-variable so that it is ready for the next iteration.

An icon that is renamed in this way (by retrieving its @-variable name) is never permanently renamed. Each time the icon is activated during a capsule application run, the name changes. In some cases, icons stored in output folders can be stored only with the permanently assigned name. The Copy Icon transformer is an example of this exception. For transformers such as the Copy Icon transformer, the icon with an expanded name can only be located in the output container with the permanently assigned name, not the icon with the @-variable name. Only one icon at a time is active when the capsule application is run. When the transformer is active, the icon with the @-variable name is not active and cannot be found by its expanded name.

Setting Arrow Options

When you use a transformer inside a capsule application, you must set the controls in the Arrow Options window to reflect the data's origin and its destination.

Arrows connecting a transformer with source icons must specify the input and output region names as they are shown in the `Display Data For` field of the transformer window.

To display the Arrow Options window:

1. Click on an arrow connecting the transformer to a source or destination icon, as shown in Figure 4.

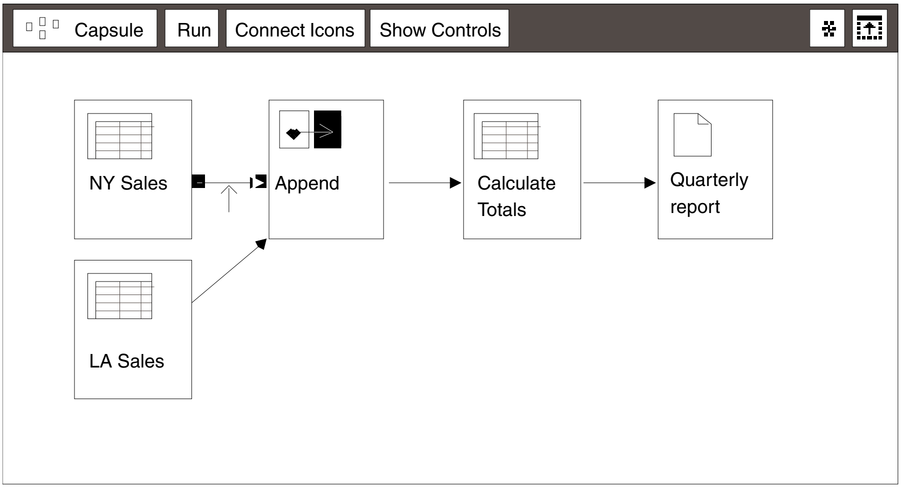
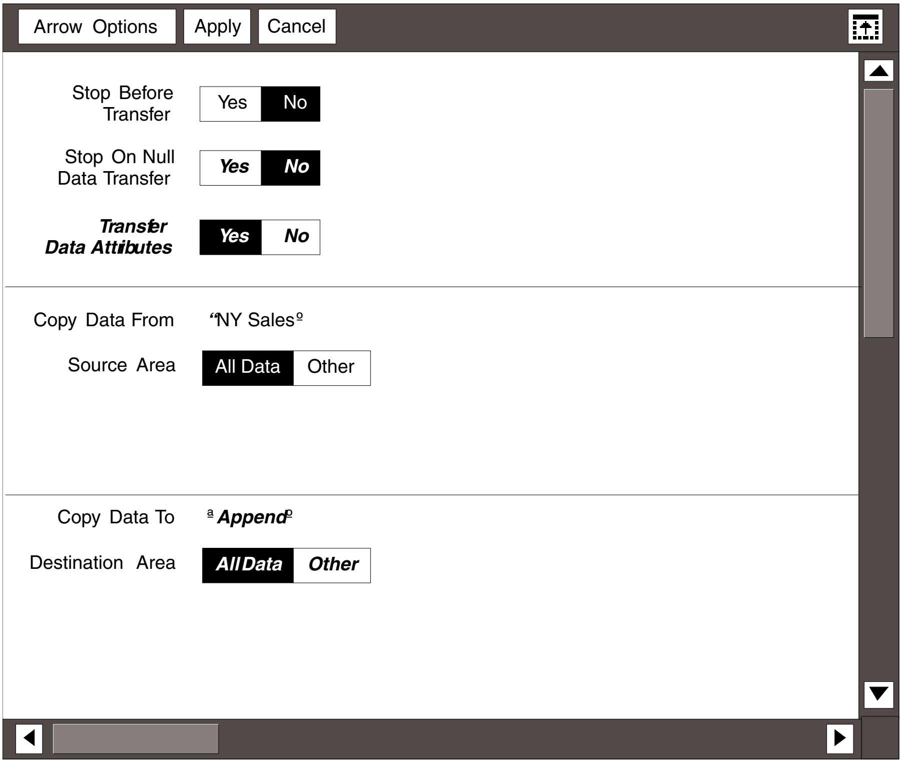


Figure 4. Arrow selection in a Capsule window

2. Open the Arrow Options window.

   The Arrow Options window opens, as shown in Figure 5.



Stop Before Transfer: Yes [✓ No]

Stop On Null Data Transfer: Yes [✓ No]

Transfer Data Attributes: [✓ Yes] No

Copy Data From “NY Salesº

Source Area: [✓ All Data] Other

Copy Data To ªAppendº

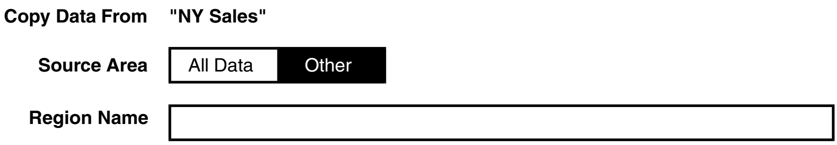
Destination Area: [✓ All Data] Other

Figure 5. The Arrow Options window

The `Copy Data From` field identifies the source icon from which the data is retrieved. The `Source Area` field allows you to choose the data you want to copy. The default, shown in Figure 5, is all data is transferred.

However, you might want to use a region of data from the source icon. To do so, you must identify the region name only in the `Source Area` field, as explained in steps 3 and 4.

3. Click on `Other` in the `Source Area` field. The `Region Name` field is displayed beneath the field of choices.



Copy Data From "NY Sales"

Source Area: All Data [✓ Other]

Region Name

4. Enter the names of the regions.

   The `Copy Data To` field identifies the transformer to which the data is transferred. The `Destination Area` field determines which input region the data will occupy.

5. Click on Other in the Destination Area field. The Region Name field is displayed beneath the field of choices.
6. In the Region Name field, type the input region names, for example, Input 1.

For additional information on the Arrow Options window, see the *Capsule User's Guide*.

Configuring a Blank Transformer

Use a blank Transformer icon when you want to create a transformer that you have written with the BASIC tool, or if you cannot locate a particular preconfigured transformer in the Transformer Icons file drawer.

If you are using a blank Transformer icon, you must enter the program name, which displays the Parameters fields.

A blank Transformer icon has no input or output choices. When you configure a blank Transformer icon to a particular transformer program, the default region names Input *n* and Output 1 are used.

To configure a blank Transformer icon:

1. Click on the Show Controls button in the Transformer window header.

   The Transformer Controls window opens, as shown in Figure 6.

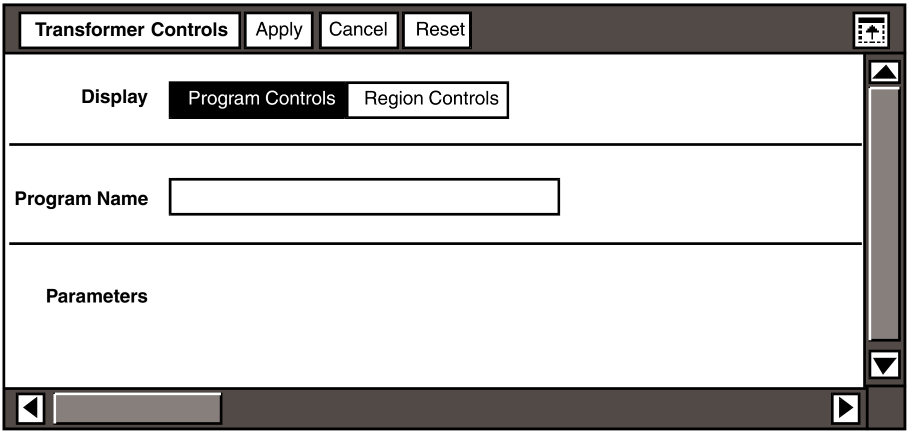


Figure 6. Transformer Controls window for a blank Transformer icon

2. Type the name of the program in the Program Name field; for example, Merge. Enter the program name exactly as shown in Table 1 on page 15, with no trailing spaces.
3. Click on the Apply button in the Transformer Controls window header.

The transformer is now programmed to perform a particular function, and the parameters generated by the program type are displayed in the `Parameters` fields.

4. Close the Transformer Controls window. The transformer window opens.

   Notice that the default input and output region names (Input *n* and Output 1) replace the region names supplied for the preconfigured transformer.

Chapter 2. Basic Transformers

You can use transformers to manipulate tabular data inserted from other Meta5 tools, such as Spreadsheet, Query, and SQL Entry. This data can either be copied or moved manually, or transferred to a capsule application.

There are several basic transformers available; each transformer is set to perform a specific function. For example, you can use the Join transformer to join columns of data from two tables, then use the Sort transformer to rearrange the data in alphabetic, numeric, or chronological order.

To locate the basic transformers:

1. Open the New Icons file drawer.
2. Open the Transformers file drawer.
3. Open the Capsule Transformers file drawer.

If you cannot find a specific transformer, see your system administrator.

Table 1 lists the basic transformers, describes their functions, and gives the page number where each transformer is described.

Table 1. Basic transformers and their functions

Transformer	Function	See
Append	Combines up to 10 tables of data into one table	“Append” on page 16
Clean	Removes columns or rows that contain blanks, N/As, zeros, or other null values.	“Clean” on page 20
Clear Contents	Clears the contents of a specified container	“Clear Contents” on page 23
Compress	Combines or deletes two or more columns or rows of data	“Compress” on page 31
Copy Icon	Copies an icon to a folder	“Copy Icon” on page 35
Join	Joins the data from two tables	“Join” on page 43
Label	Creates mailing labels for form letters	“Label” on page 52
Merge	Combines text with incoming data to produce a report or form letters	“Merge” on page 58

Table 1. Basic transformers and their functions

Transformer	Function	See
MultiJoin	Joins the data from up to five tables	“MultiJoin” on page 73
Pivot	Designs a report by rearranging the columns and rows of data	“Pivot” on page 82
Select	Deletes or reorders columns of data	“Select” on page 92
Sort	Rearranges data in alphabetic, numeric, or chronological order	“Sort” on page 94

For general information on using transformers, see “Chapter 1. Getting Started with Transformers,” on page 1.

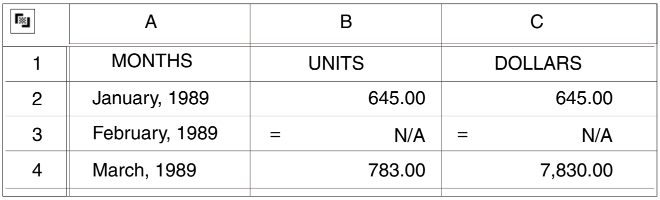
Append

The Append transformer allows you to combine up to 10 tables of data into one table. Each table is added to the output in the order it appears in the transformer's input regions.

Parameters

The 10 parameter fields in the Append transformer enable you to specify the number of heading rows for each input region. The default value is 0. If you want to exclude the heading row from the output, type `1` in the `Number of heading rows` field.

For example, suppose you copy the data in Figure 7 to the Input 1 region in the Append transformer. If you do not want the heading row (MONTHS, UNITS, and DOLLARS) transferred to the output, type `1` in the `Number of heading rows` field. If you want at least one heading row, type `0` for Input 1.



	A	B	C
1	MONTHS	UNITS	DOLLARS
2	January, 1989	645.00	645.00
3	February, 1989	= N/A	= N/A
4	March, 1989	783.00	7,830.00

Figure 7. Number of heading rows example

When you use the Append transformer in a capsule application, you must always use Input 10 as the last (or the only) input region. Otherwise, the desired results will not appear in the output region when you run the capsule.

For example, if you want to append two tables of data, assign the first table to Input 1 region and the second table to Input 10 region. Table 2 shows which input regions to use for a specified number of tables in a capsule application.

Table 2. Input locations based on the number of tables in a capsule application

Table	Input Regions to Use									
1	Input 10									
2	Input 1	Input 10								
3	Input 1	Input 2	Input 10							
4	Input 1	Input 2	Input 3	Input 10						
5	Input 1	Input 2	Input 3	Input 4	Input 10					
6	Input 1	Input 2	Input 3	Input 4	Input 5	Input 10				
7	Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 10			
8	Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7	Input 10		
9	Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7	Input 8	Input 10	
10	Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7	Input 8	Input 9	Input 10

The transformer reads all input regions in succession starting with Input 1 and appends each table without the specified heading rows to the output region.

Region Controls

The `Display Data For` field in the Append window contains the following choices:

Input 1 through Input 10
The regions where you can copy tabular data

Output 1
The area where appended data appears after running the transformer

Example

Suppose you have three tables that you want to append in a capsule application, as shown in Figure 8.

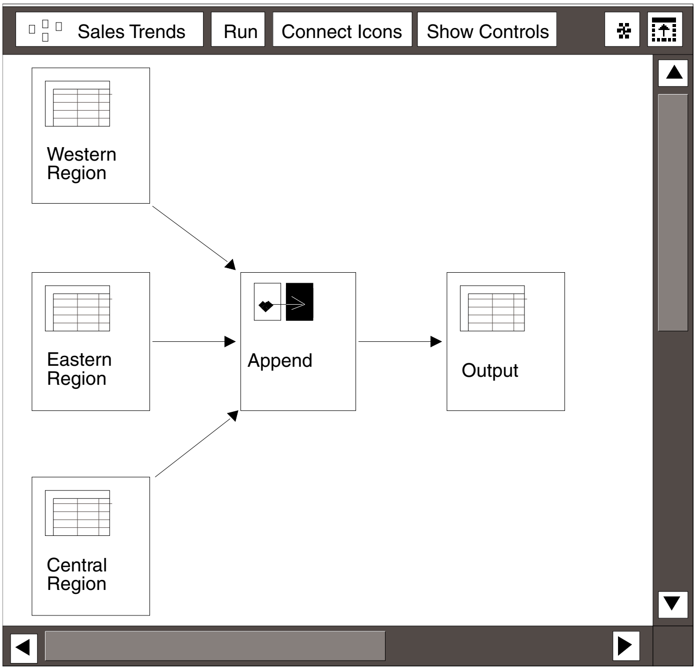
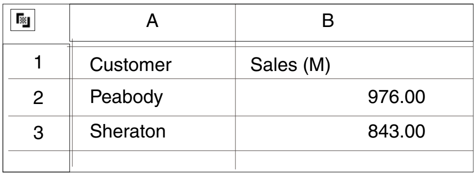


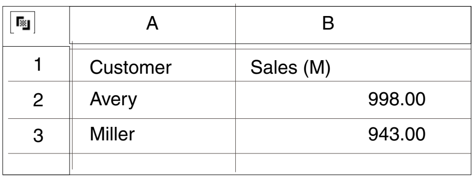
Figure 8. Append example

Use Input 1 in the Append transformer for the Western Region spreadsheet, shown in the following example:



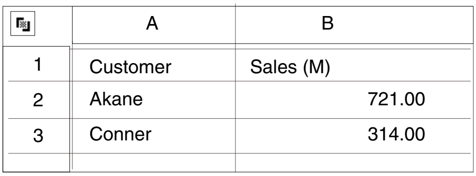
	A	B
1	Customer	Sales (M)
2	Peabody	976.00
3	Sheraton	843.00

Use Input 2 in the Append transformer for the Eastern Region spreadsheet, shown in the following example:



	A	B
1	Customer	Sales (M)
2	Avery	998.00
3	Miller	943.00

Use Input 10 in the Append transformer for the Central Region spreadsheet, shown in the following example:



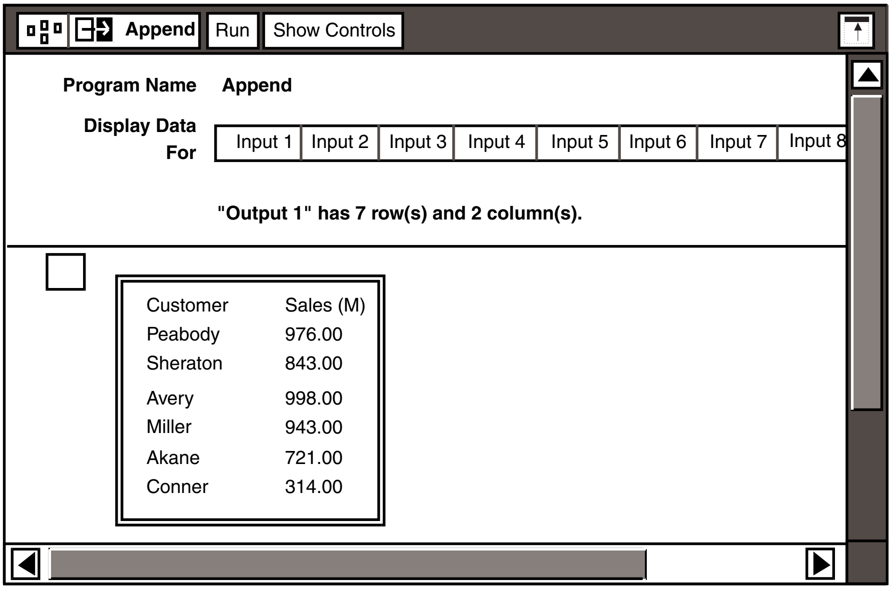
	A	B
1	Customer	Sales (M)
2	Akane	721.00
3	Conner	314.00

To create a capsule application that uses the Append transformer to append spreadsheet data:

1. Connect the three spreadsheets named Western Region, Eastern Region, and Central Region to the Append transformer. Connect the Append transformer to the Output spreadsheet.
2. Display the options for the arrow that connects the Western Region spreadsheet to the Append transformer.
3. In the Arrow Options window, click on `Other` in the `Destination Area` field and type `Input 1` as the region name. Close the Arrow Options window.
4. Display the options for the arrow that connects the Eastern Region spreadsheet to the Append transformer.
5. In the Arrow Options window, click on `Other` in the `Destination Area` field and type `Input 2` as the region name. Close the Arrow Options window.
6. Display the options for the arrow that connects the Central Region spreadsheet to the Append transformer.
7. In the Arrow Options window, click on `Other` in the `Destination Area` field and type `Input 10` as the region name. Close the Arrow Options window.
8. Open the Append transformer and click on the `Show Controls` button in the Append transformer window header.
9. Type `1` in the `Number of heading rows for Input 2` and `Number of heading rows for Input 10` fields.

   Leave 0 in the `Number of heading rows for Input 1` field to transfer the heading row in the Western Region spreadsheet to the output.
10. Close the Transformer Controls window.
11. Click on the `Run` button in the Sales Trends capsule window header.
12. When the processing is completed, open the transformer.

The result appears as shown in Figure 9, and in the spreadsheet connected to the transformer.



Program Name Append

Display Data For: Input 1 | Input 2 | Input 3 | Input 4 | Input 5 | Input 6 | Input 7 | Input 8

"Output 1" has 7 row(s) and 2 column(s).

Customer	Sales (M)
Peabody	976.00
Sheraton	843.00
Avery	998.00
Miller	943.00
Akane	721.00
Conner	314.00

Figure 9. Append output

Clean

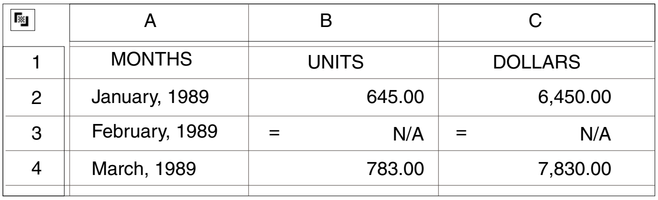
The Clean transformer removes columns or rows in a spreadsheet that contain blanks, N/As, zeros, or null values. The transformer recognizes only N/As that are produced by a function of the table, not those you enter.

Parameters

Number of Heading Rows

This parameter specifies the number of rows used as column headings within the data. These rows are excluded from the cleaning process. The default value is 0; that is, no rows are used for column headings.

For example, the following spreadsheet contains one heading row (MONTHS, UNITS, and DOLLARS). You would not want to include the heading row in the cleaning process, so you type `1` in the `Number of heading rows` field.



	A	B	C
1	MONTHS	UNITS	DOLLARS
2	January, 1989	645.00	6,450.00
3	February, 1989	= N/A	= N/A
4	March, 1989	783.00	7,830.00

Clean Rows

This parameter specifies whether to clean the data rows. Type `yes` to remove all rows that contain only blanks, N/As, zeros, or null values. `Yes` is the default value. Type `no` to retain the values in all data rows.

Columns to Check

This parameter identifies the data columns you want to check for blanks, N/As, zeros, or other null values. You can check all data columns by typing `all` in the `Columns to check` field; or type any combination of column letters, such as `b, c`. The default value is `all`.

The following scenarios are based on the example in the previous illustration.

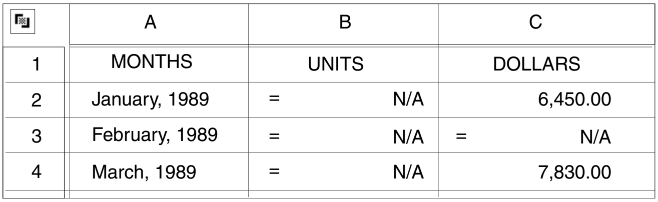
- Typing `b, c` in the `Columns to check` field removes row 3, because each column contains an N/A entry for that row. Column A is not checked, regardless of any valid entries.
- Typing `all` into the `Columns to check` field does not remove any rows because column A contains a valid entry.
- Typing `a` into the `Columns to check` field does not remove any rows because column A contains a valid entry.
- Typing `a, b` into the `Columns to check` field does not remove any rows because column A has a valid entry, even though column B contains an N/A entry.

Clean Columns

This parameter specifies whether to remove all columns that contain only blanks, N/As, zeros, or other null values.

For example, suppose you want to remove column B from the following spreadsheet because it contains only N/A entries. First, type `1` in the `Number of Heading Rows` field so that the heading row is removed from the clean process, then type `yes` in the `Clean Columns` field. The default value is `no`.

If you want to retain the values in all data columns, type `no`.

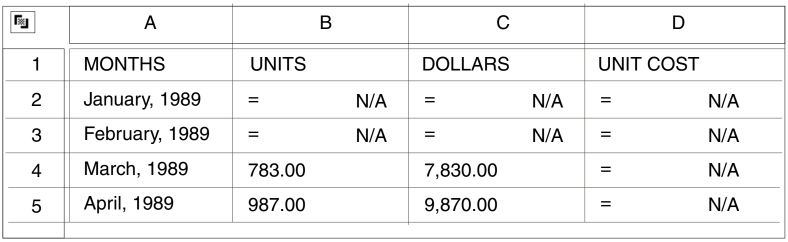


	A	B	C
1	MONTHS	UNITS	DOLLARS
2	January, 1989	= N/A	6,450.00
3	February, 1989	= N/A	= N/A
4	March, 1989	= N/A	7,830.00

The values in both the `Clean Rows` and `Clean columns` fields cannot be `no` at the same time.

Example

Suppose you have a capsule application with a spreadsheet that contains the information shown in the following example.

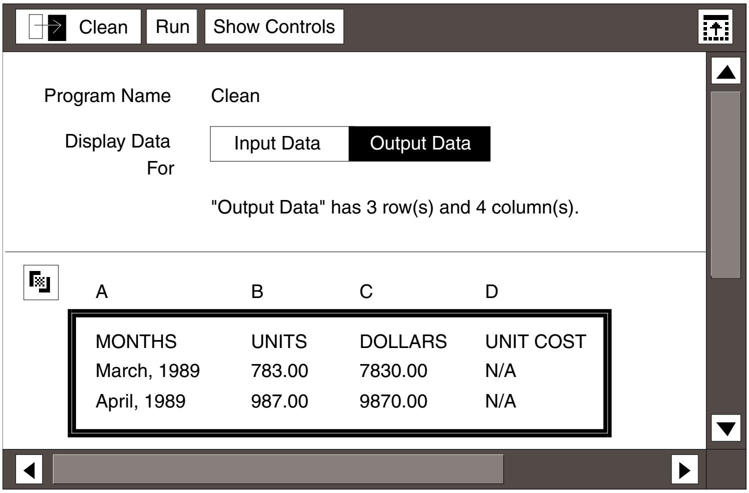


	A	B	C	D
1	MONTHS	UNITS	DOLLARS	UNIT COST
2	January, 1989	= N/A	= N/A	= N/A
3	February, 1989	= N/A	= N/A	= N/A
4	March, 1989	783.00	7,830.00	= N/A
5	April, 1989	987.00	9,870.00	= N/A

If you want to create an output that displays only valid entries (March and April), you can use the Clean transformer to remove the rows that contain N/As:

1. Click on the `Show Controls` button in the Clean transformer window header.
2. Type `1` in the `Number of heading rows` field to omit row 1 from being included in the cleaning process.
3. Type `yes` in the `Clean rows` field to remove all rows that contain only N/As.
4. Type `b, c, d` in the `Columns to check` field to keep column A from being included in the cleaning process.
5. Type `no` in the `Clean columns` field to retain column D, which otherwise would be removed from the cleaning process because it contains N/As.
6. Close the Transformer Controls window and the Clean transformer window.
7. Click outside the Capsule window to deselect the Clean transformer.
8. Click on the `Run` button in the Capsule window header.

The output appears as shown in Figure 10.



Program Name Clean

Display Data For ○ Input Data [✓ Output Data]

"Output Data" has 3 row(s) and 4 column(s).

A	B	C	D
MONTHS	UNITS	DOLLARS	UNIT COST
March, 1989	783.00	7830.00	N/A
April, 1989	987.00	9870.00	N/A

Figure 10. Example Clean transformer output

Clear Contents

The Clear Contents transformer clears the contents of specified folders and envelopes inside capsule applications. The Clear Contents transformer:

- Eliminates the need to manually search through embedded capsules to clear containers, which improves capsule performance
- Reduces the amount of disk storage space used by capsule applications on a file service

Parameters

The Clear Contents transformer has one parameter, `Name of container(s) to clear`. You can enter multiple icon names in the field, separated by a comma, or use @-variables as parameters. You can also enter a path name if the folder or the transformer is located in an embedded capsule. For information on entering path names in this field, see “Entering Path Names for Folders and Envelopes” on page 26.

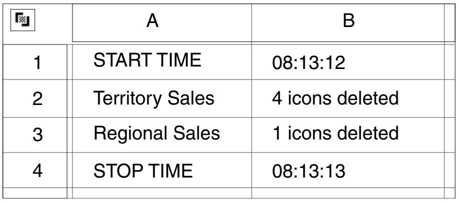
To avoid clearing the wrong container, be sure to assign unique names and path names to the containers you want to clear. The transformer cannot distinguish between two icons with identical names. If you use identical names, the transformer might inadvertently clear a container that is vital to the capsule process.

Region Controls

The `Display Data For` field has one choice, Output 1. When you run the transformer, the results appear in the Output 1 display area as well as in the Spreadsheet or Text icon connected to the transformer. The results consists of the following information:

- The time transformer started and ended its run
- The name of the folder or envelope that was cleared
- A status message, which indicates the number of icons that were cleared

Figure 11 shows results that might appear in a spreadsheet window connected to the transformer.



	A	B
1	START TIME	08:13:12
2	Territory Sales	4 icons deleted
3	Regional Sales	1 icons deleted
4	STOP TIME	08:13:13

Figure 11. Clear Contents results in a Spreadsheet window

Positioning the Clear Contents Transformer in a Capsule Application

The position of the transformer within a capsule application determines when it clears specified folders or envelopes during the run process. The transformer will not run stand-alone in a capsule and, consequently, must be connected to a Text or Spreadsheet icon.

Figure 12 shows a sample Clear Contents transformer in an embedded capsule. In this example, the transformer is positioned at the top of the capsule application to clear the Final Reports folder before the capsule is processed.

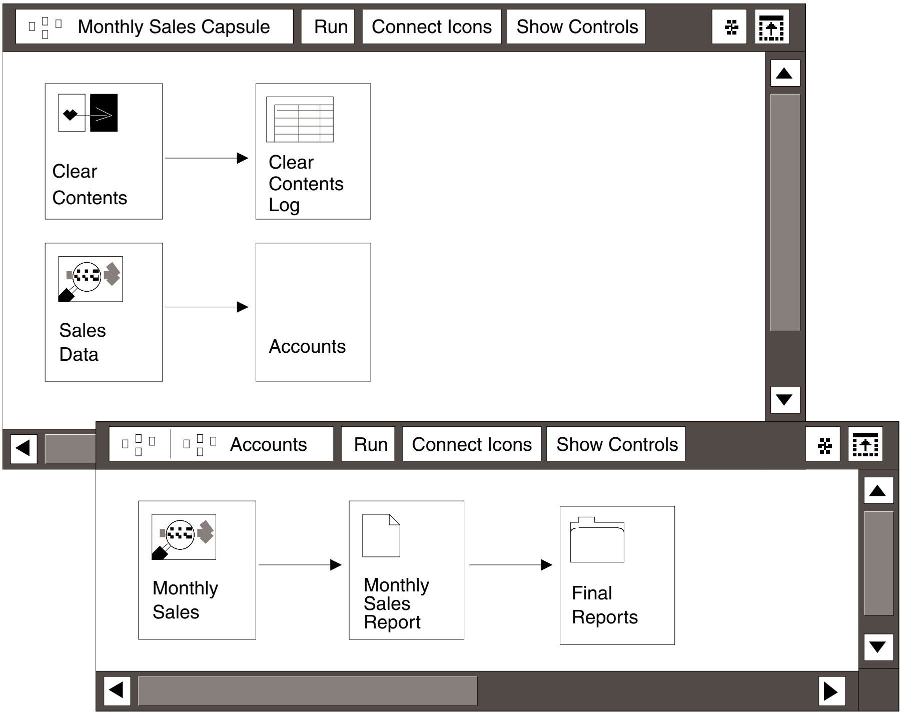


Figure 12. Sample Clear Contents setup in an embedded capsule

Below the transformer, the Sales Data Query icon is connected to an embedded capsule called Accounts, which runs an iteration process. In turn, the Accounts capsule contains a Monthly Sales query, Monthly Sales Report text document, and Final Reports folder.

When you run the capsule application, the Clear Contents transformer processes first, because it is the first icon in the outer capsule. It clears envelopes and folders named in its parameter (in this case, the Final Reports folder in the Accounts capsule), and logs the results in the Spreadsheet icon connected to the transformer. The Query icon then sends data to the Accounts capsule, and the results of the process are placed in the Final Results folder.

Folders and envelopes are the only containers that can be cleared with the Clear Contents transformer. You can clear only folders and containers for which you have change privileges. Mail trays, printers, and file drawers are not recognized by this transformer.

You can run the transformer by clicking on the `Run` button in the Capsule window header. If the transformer encounters errors during processing, the error messages are posted in the message area and the transformer's Output 1 region and, if applicable, in the capsule run log. However, an error message does not appear in the Text or Spreadsheet icon connected to the transformer.

You can also run the transformer by clicking on the `Run` button in the transformer window header. However, the results of the run will not appear in the Text or Spreadsheet icon connected to the transformer.

Entering Path Names for Folders and Envelopes

A path name identifies a logical connection between two icons, such as the Clear Contents transformer and the container to be cleared. Keywords are provided to facilitate the search for a container on a different level than the transformer; for example, in an embedded capsule, or on the desktop. Table 3 describes the different level locations, and the appropriate keywords to use when creating a path name.

Table 3. Level location of containers to be cleared

Current level	The folder or envelope is located in the same container as the Clear Contents transformer.
PARENT	The folder or envelope is in a container that is one or more levels above the container that holds the Clear Contents transformer. You can use the PARENT keyword to clear a container that is located anywhere within the capsule application. To clear a container on the desktop, use the DESKTOP keyword. The PARENT keyword must be in uppercase and must come before the path name in the `Name of container(s) to clear` field. You can make multiple references to PARENT. If a container is embedded in another container, you must specify a path name, using the vertical bar. For example, `PARENT|Sales Analysis|Monthly Sales`
DESKTOP	The folder or envelope is in a container on the desktop that contains the Clear Contents transformer. The specified parameter path name in the `Name of container(s) to clear` field must start with the keyword DESKTOP and must be in uppercase.

In addition to the keywords, you can use the vertical bar to navigate through a directory hierarchy. The vertical bar, in combination with an icon name, allows the user to travel down the directory tree and locate the icon specified in the parameters field. This concept of vertical bar navigation is illustrated in later examples.

Current Level

Current level indicates that the search for a specified folder or envelope starts at the container where the Clear Contents transformer is currently located. If you do not specify the keyword PARENT or DESKTOP in the parameter, the transformer automatically assumes the location is the default or current level.

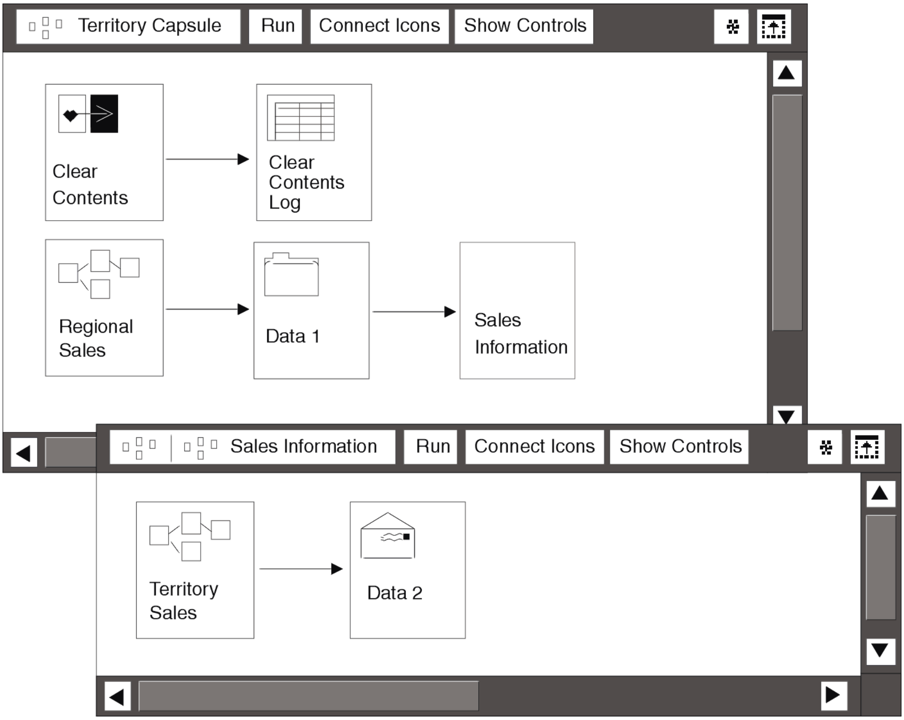


Figure 13. Sample Clear Contents Current Level setup

In the example shown in Figure 13, if you want to clear the contents of Data 1 folder and Data 2 envelope, specify the parameter path name as follows:

Parameters Data 1, Sales Information|Data 2 Name of container(s) to clear

The Data 1 folder is on the same level as the Clear Contents transformer, and the Data 2 envelope is in the Sales Information capsule. When you click on `Run`, the Clear Contents transformer begins the search in the current container and clears the contents of the folder called Data 1 and then the contents of the Data 2 envelope in the Sales Information capsule.

The output results appear in the transformer Output 1 region and the Clear Contents log connected to the transformer.

PARENT Level

The PARENT keyword specifies that a search begin in the container that is one level or more above the container that holds the Clear Contents transformer. If you want to search for a folder or envelope on the PARENT level, you must begin the path name in the `Name of container(s) to clear` field with the PARENT keyword.

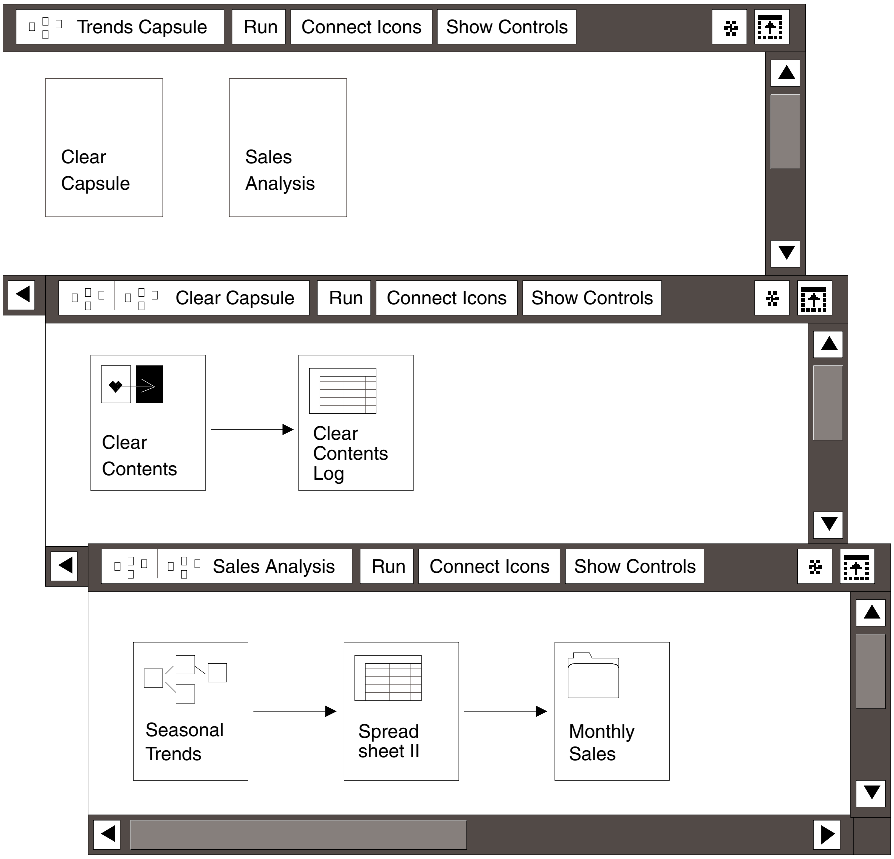


Figure 14. Sample Clear Contents PARENT setup

Figure 14 illustrates the PARENT level concept. The Clear Contents transformer is in an embedded capsule called Clear. To clear the contents of the Monthly Sales folder in the embedded Sales Analysis capsule, specify the following path name in the `Name of container(s) to clear` field:

Parameters	PARENT|Sales Analysis|Monthly Sales	Name of container(s) to clear

When you click on the `Run` button, the Clear Contents transformer performs the following process:

1. Locates the container one level above the container that holds the transformer, which is the Trends capsule.
2. Locates the Sales Analysis capsule in the Trends capsule.
3. Finds and clears the contents of the Monthly Sales folder in the Sales Analysis capsule.

4. Places the output results in the transformer Output 1 region and the Clear Contents Log connected to the transformer.

If you have several embedded capsules, it is possible to move up the directory tree from one container to another by using multiple references to PARENT, separated by a vertical bar. Because there is no PARENT beyond a user's desktop, you cannot destroy icons or inadvertently or intentionally gain access to icons that you do not own.

DESKTOP Level

The DESKTOP keyword enables users to search for and clear folders or envelopes located on their desktop. When you search for a folder or envelope at the desktop level, the parameter path name you enter in the `Name of container(s) to clear` field must start with the DESKTOP keyword.

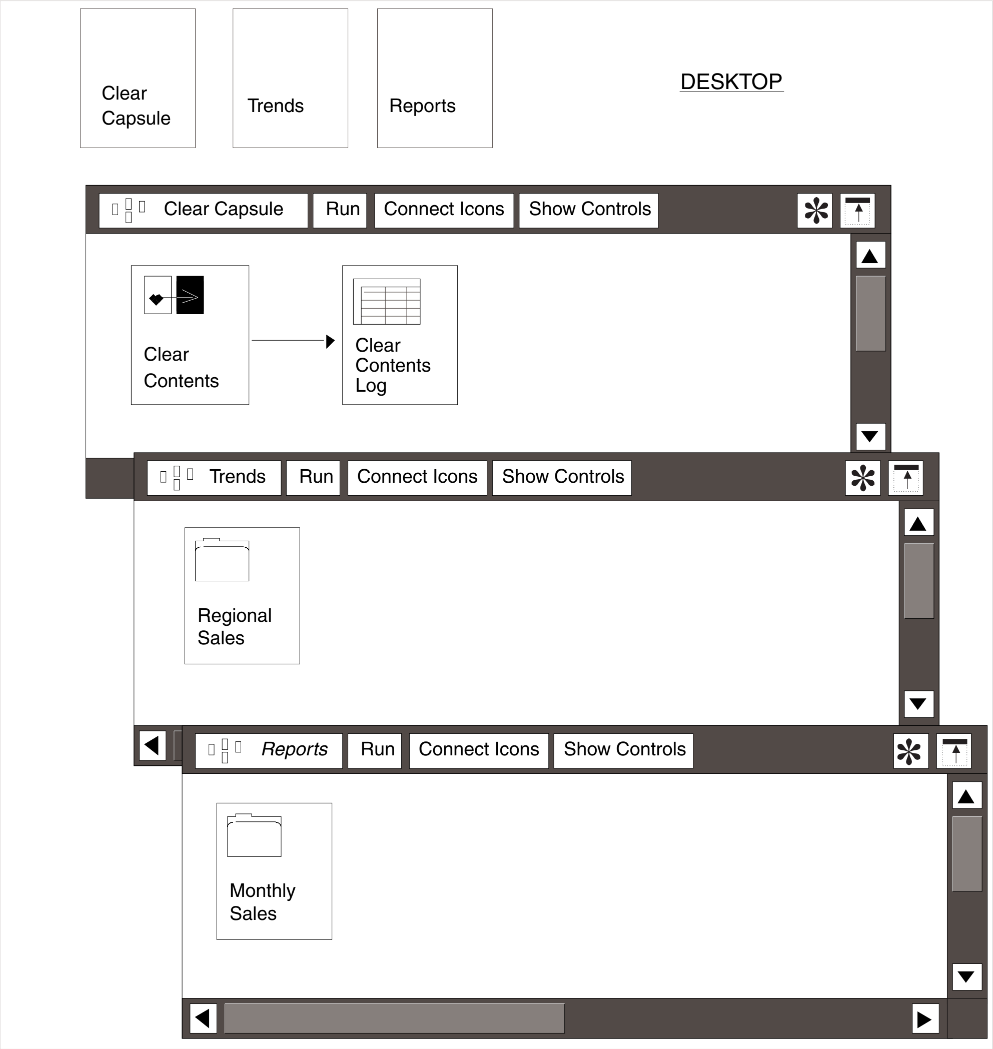


Figure 15. Sample Clear Contents DESKTOP setup

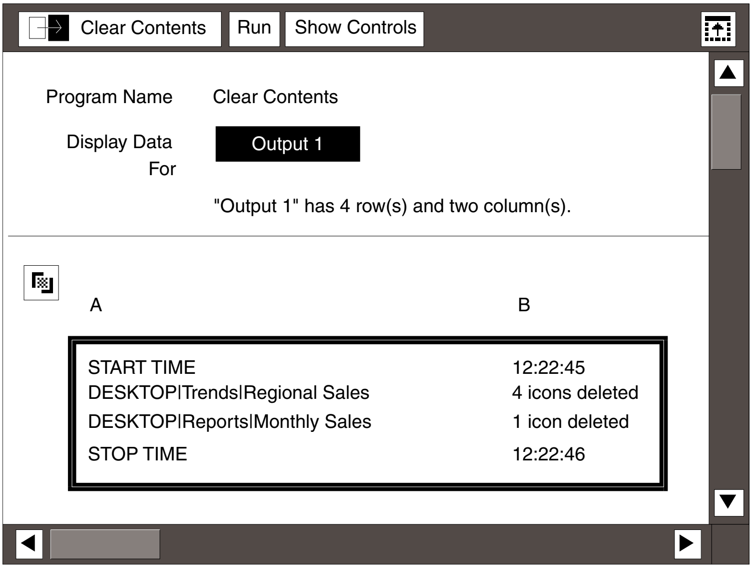
Figure 15 illustrates the DESKTOP level concept. The Clear Contents transformer is located in the Clear capsule icon on the desktop. If you want to clear two folders, Regional Sales and Monthly Sales, which are located in two capsule icons on the desktop, Trends and Reports, respectively, then specify the following path name in the `Name of container(s) to clear` field:

```
DESKTOP|Trends|Regional Sales, DESKTOP|Reports|Monthly Sales
```

When you click on the `Run` button in the Clear capsule window header, the Clear Contents transformer performs the following procedure:

1. Begins the search at the desktop level and locates the Trends capsule.
2. Locates the Regional Sales folder in the Trends capsule and clears its contents.
3. Begins the search at the desktop level again and locates the Reports capsule.
4. Locates the Monthly Sales folder in the Reports capsule and clears its contents.
5. Places the results in the transformer Output 1 region and the Clear Contents Log connected to the transformer.

Figure 16 is the example that might appear in the Output 1 region.



Program Name Clear Contents

Display Data For Output 1

"Output 1" has 4 row(s) and two column(s).

```
A                                             B
START TIME                                    12:22:45
DESKTOP|Trends|Regional Sales                 4 icons deleted
DESKTOP|Reports|Monthly Sales                 1 icon deleted
STOP TIME                                     12:22:46
```

Figure 16. Clear Contents example output

Compress

The Compress transformer allows you to combine and delete two or more columns or rows of data.

The Compress transformer is useful for removing headings that are transferred from a Query or Spreadsheet to a Text icon, and for transferring data from a Reporter icon to a Plot icon when you want to compress certain heading rows or columns. If you want to compress all heading rows or columns in a report, it is easier to do it directly in a Reporter icon.

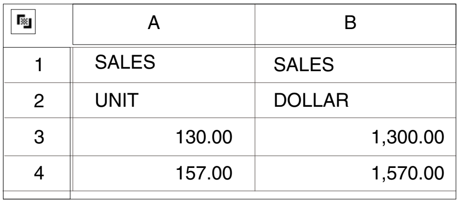
Parameters

The Compress transformer has the following parameters:

Rows to Combine

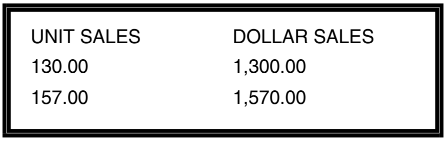
This parameter specifies the row numbers that you want to combine into one, and the order in which you want them to appear in the output. If you do not want to combine any rows, leave this field blank.

For example, suppose you have the following sales information:



	A	B
1	SALES	SALES
2	UNIT	DOLLAR
3	130.00	1,300.00
4	157.00	1,570.00

To combine rows 1 and 2 and rearrange the order in which they appear in the output, type `2, 1` in the field. You would receive the following results:



```
UNIT SALES          DOLLAR SALES
130.00              1,300.00
157.00              1,570.00
```

The transformer automatically inserts a blank space between the combined values. For example, the words UNIT SALES and DOLLAR SALES in the example appear separated by a space, although none was specified.

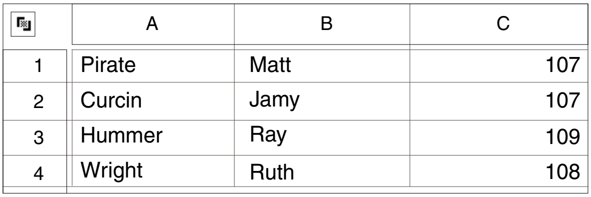
Rows to Delete

This parameter specifies the row numbers that you want to remove from the table. If you do not want to remove any rows, leave this field blank.

Columns to Combine

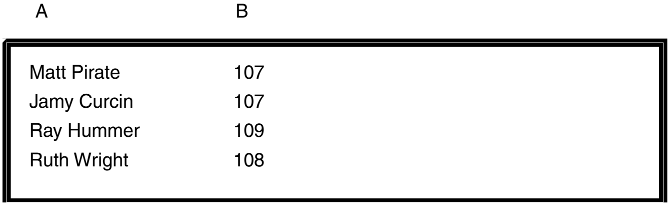
This parameter specifies the letters of the columns that you want to combine into one, and the order in which you want them to appear in the output. If you do not want to combine any columns, leave this field blank.

For example, suppose you have the following personnel listing:



	A	B	C
1	Pirate	Matt	107
2	Curcin	Jamy	107
3	Hummer	Ray	109
4	Wright	Ruth	108

To combine columns A and B into one column, type `b, a` in the `Columns to Combine` field. Because the column letters are entered in reverse order, the data in column B precedes the data in column A in the results:



A	B
Matt Pirate	107
Jamy Curcin	107
Ray Hummer	109
Ruth Wright	108

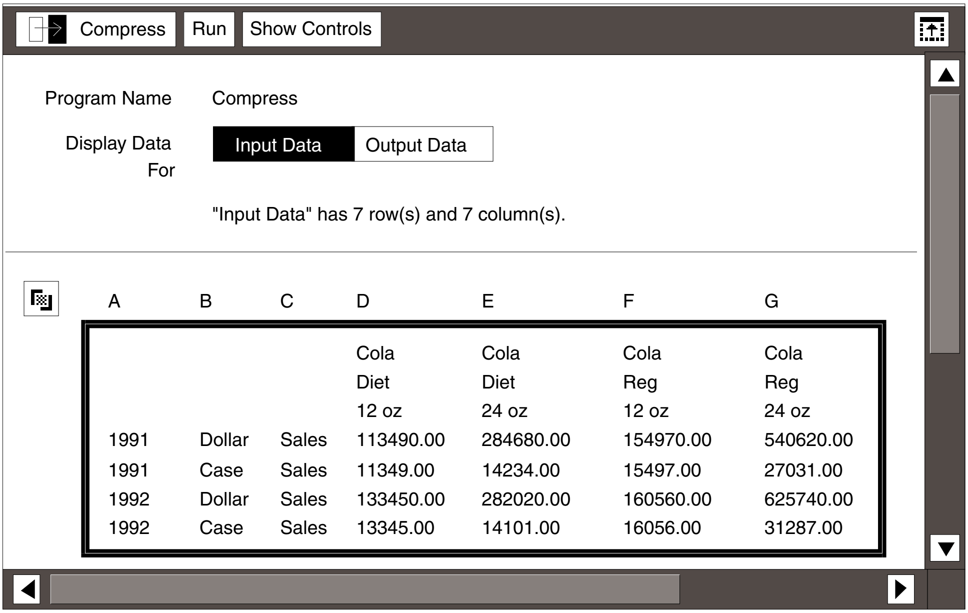
Because column width does not change in the Output Data region, the results might appear incomplete to you. However, the data has not been truncated. Transfer it to a Spreadsheet or Text icon to view all the results.

Columns to Delete

This parameter specifies the column letters of the columns that you want to remove from the table. If you do not want to remove any columns, leave this field blank.

Example

Suppose you copy data in the Input Data region of the Compress transformer window, as shown in Figure 17.



Program Name Compress

Display Data For: Input Data | Output Data

"Input Data" has 7 row(s) and 7 column(s).

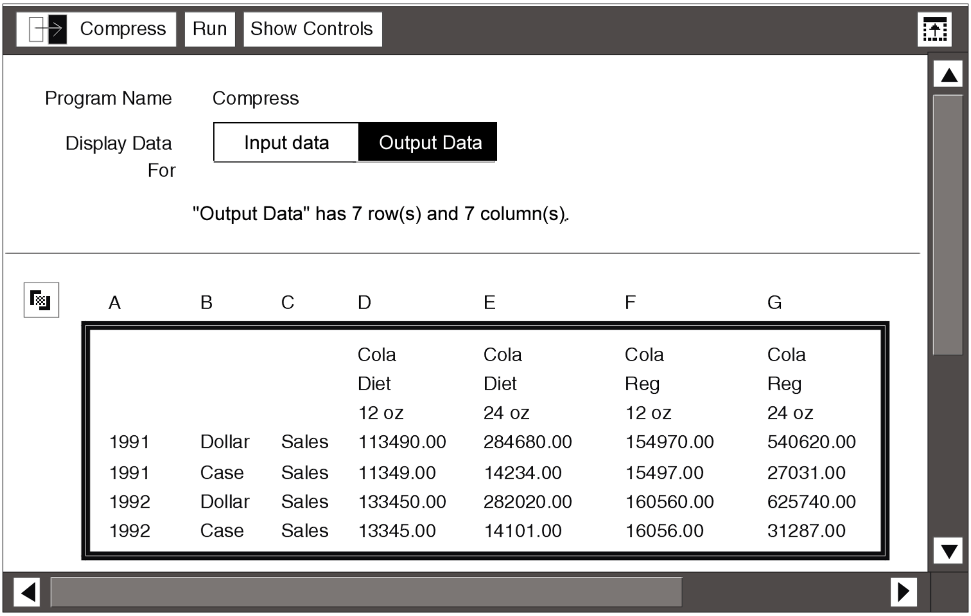
A	B	C	D	E	F	G
			Cola Diet 12 oz	Cola Diet 24 oz	Cola Reg 12 oz	Cola Reg 24 oz
1991	Dollar	Sales	113490.00	284680.00	154970.00	540620.00
1991	Case	Sales	11349.00	14234.00	15497.00	27031.00
1992	Dollar	Sales	133450.00	282020.00	160560.00	625740.00
1992	Case	Sales	13345.00	14101.00	16056.00	31287.00

Figure 17. Compress example

You can remove columns and rows that repeat the same information, and combine columns and rows to make a more concise report:

1. Click on the `Show Controls` button in the window header.
2. Type `3, 2` in the `Rows to combine` field to combine and reverse the order of rows 2 and 3.
3. Type `1` in the `Rows to delete` field to remove row 1.
4. Type `a, b` in the `Columns to combine` field to combine columns A and B.
5. Type `c` in the `Columns to delete` field to remove column C.
6. Close the Transformer Controls window.
7. Click on the `Run` button in the transformer window header.

When the run is complete, the result appears as shown in Figure 18 on page 35.



Program Name Compress

Display Data For: Input data [✓ Output Data]

"Output Data" has 7 row(s) and 7 column(s).

A	B	C	D	E	F	G
			Cola	Cola	Cola	Cola
			Diet	Diet	Reg	Reg
			12 oz	24 oz	12 oz	24 oz
1991	Dollar	Sales	113490.00	284680.00	154970.00	540620.00
1991	Case	Sales	11349.00	14234.00	15497.00	27031.00
1992	Dollar	Sales	133450.00	282020.00	160560.00	625740.00
1992	Case	Sales	13345.00	14101.00	16056.00	31287.00

Figure 18. Compress output

Copy Icon

The Copy Icon transformer copies specified icons from embedded capsules and places the results in a designated destination folder for easy access. The Copy Icon transformer eliminates the need to search through many layers of icons to find capsule results. When you copy a container that has a subtree, its entire subtree is also copied.

Parameters

The Copy Icon transformer has the following parameters:

Name of Icon(s) to Copy

This parameter specifies which icons to copy. You can enter multiple icon names separated by a comma and a space, or you can use @-variables as parameters. You can also enter a path name if the folder or the transformer is located in an embedded capsule.

To avoid copying the wrong icon, be sure to assign unique names to the icons you want to copy. The transformer cannot distinguish between two icons with identical names.

If the name of the icon you want to copy contains a comma, then you must enclose it in single quotes for the transformer to parse it correctly. Otherwise, the transformer interprets the comma as a list separator, and the icon as multiple entries.

For example, if you want to copy an icon named Nov 5, 1993 from a folder named Data, type `'Data|Nov 5, 1993'` in the `Name of icons to copy` field. This guideline is valid for @-variables that reference a value containing a comma. For example, if the icon name is @A, and the value of @A is Nov 5, 1993, type `'Data|@A'` in this field.

For information on entering path names, see “Entering Path Names for Icons” on page 38.

Name of Destination Folder

This parameter specifies the name of the folder to which you want the icons copied.

Overwrite? Y/N

This parameter specifies whether to replace (Y) an existing icon with the copied icon, or add it (N) to the icons in the destination folder.

Common Source Container

This parameter specifies a container or a path name that, when the transformer is run, is added to the beginning of each icon name listed in the `Name of icon(s) to copy` field. The container or path name must be valid for all icons.

For example, if the New York, Dallas, and Los Angeles icons are all contained in the Sales folder on the desktop, you can enter the icon names in the `Name of icons to copy` field. Then type `DESKTOP|Sales` in the `Common source container` field.

For information on specifying path names, see “Entering Path Names for Icons” on page 38.

Region Controls

The `Display Data For` field has one choice, Output 1. When you run the transformer, the results appear in the Output 1 display area, as well as in the Spreadsheet or Text icon connected to the transformer. The result consists of the following information:

- The time the transformer started and ended its run
- The name of the folder or envelope that was copied
- A status message, which indicates whether the icon was copied successfully

Positioning the Copy Icon Transformer in a Capsule Application

The position of the transformer within a capsule application determines when it copies specified icons during processing. The transformer will not run stand-alone in a capsule and must have an outgoing arrow connected to a Text or Spreadsheet icon.

You must have read access to the icons you are copying and write access to the destination folder to use this transformer. You cannot copy locked icons.

Figure 19 is a sample Copy Icon transformer set up in an embedded capsule. In this example, the transformer is positioned so that it executes and copies icons to the destination folder at the end of the capsule process.

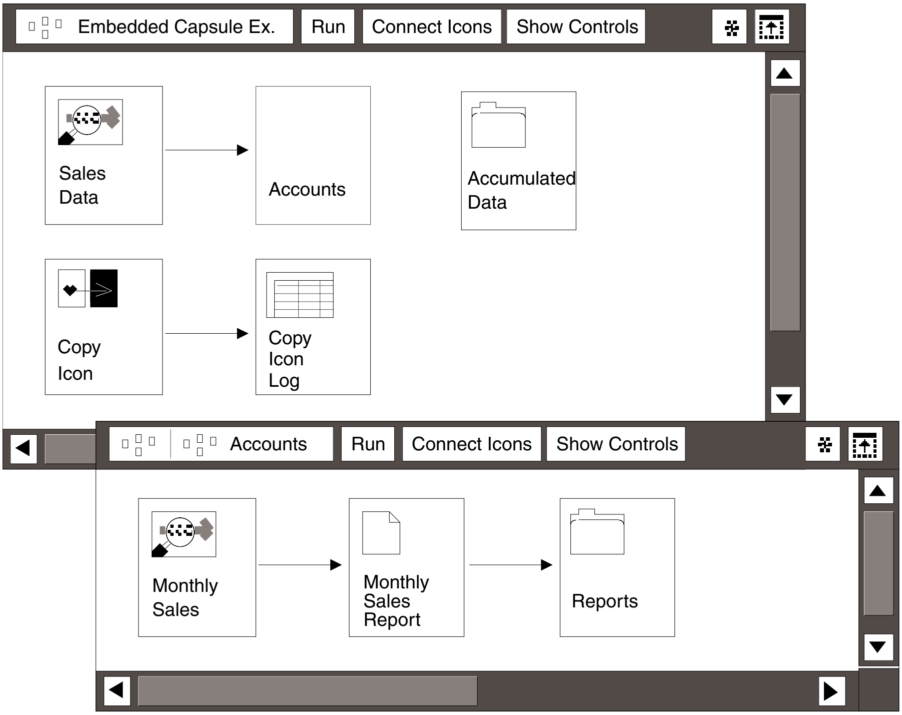


Figure 19. Sample Copy Icon setup in an embedded capsule

Above the transformer, a Sales Data Query icon is connected to an embedded capsule application called Accounts, which is running an iterative process. In turn, the Accounts capsule contains a Monthly Sales Query, a Monthly Sales Report, and a Reports folder.

When you run the capsule application, the Sales Data Query sends the data to the Accounts capsule, which in turn places the results of the iterative process in the Reports folder. After the processing sequence for the inner capsule is completed, the Copy Icon transformer, which is the last in the process of the outer capsule, copies the Reports folder into the Accumulated Data folder in the outer capsule. It also records the output results in the transformer Output 1 region and in the spreadsheet connected to the transformer.

You can run the transformer by clicking on the `Run` button in the Capsule window header. If the transformer encounters errors during processing, error messages are posted in the message area and the transformer's output region. An error message does not appear in the Text or Spreadsheet icon connected to the transformer.

You can also run the transformer by clicking on the `Run` button in the transformer window header. However, the results will not appear in the Text or Spreadsheet icon connected to the transformer.

See the *Capsule User's Guide* for more information about embedded capsules and iterative processes.

Entering Path Names for Icons

A path name identifies a logical connection between two icons, such as the Copy Icon transformer and the icon to copy. Keywords are provided to facilitate the search for an icon on a different level than the transformer; for example, in an embedded capsule or on the desktop. Table 4 on page 38 describes the different level locations, and the appropriate keywords to use when creating a path name.

Table 4. Level location of containers to be cleared

Current level	The folder or envelope is located in the same container as the Copy Icon transformer.
PARENT	The folder or envelope is in a container that is one or more levels above the container that holds the Copy Icon transformer. You can use the PARENT keyword to copy an icon up to, but not outside the top capsule level. The PARENT keyword must be entered in uppercase and must come before the path name in the `Name of icon(s) to copy` field. You can make multiple references to PARENT. If an icon is embedded in a container, you must specify a path name, using the vertical bar (|).
DESKTOP	The folder or envelope is in a container on the desktop that contains the Copy Icon transformer. The specified parameter path name in the `Name of icon(s) to copy` field must be entered in uppercase and must start with the keyword DESKTOP.

In addition to the keywords, you can use the vertical bar to navigate through a directory hierarchy. The vertical bar, in combination with an icon name, allows the user to travel down the directory tree and locate the icon specified in the parameters field. The concept of vertical bar navigation is illustrated in later examples.

Current Level

Current level indicates that the search for a specified icon starts at the container where the Copy Icon transformer is currently located. If you do not specify the keyword PARENT or DESKTOP, the transformer automatically assumes the location is on the current level.

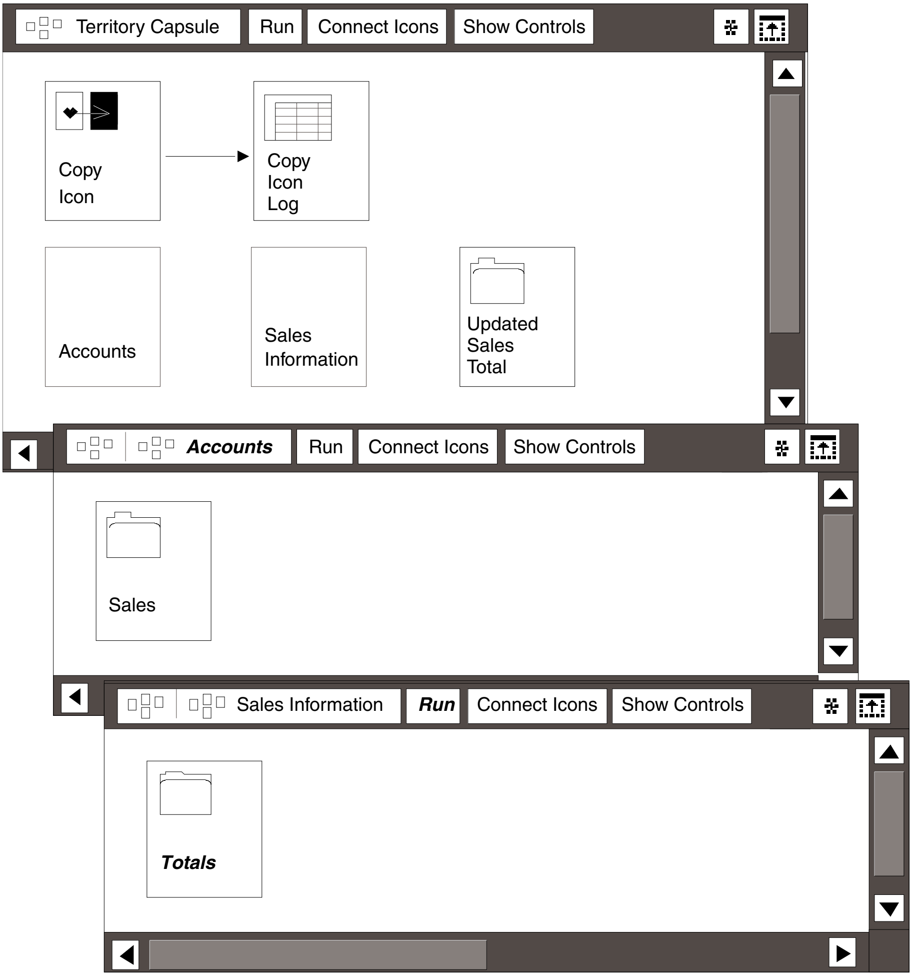


Figure 20. Sample Copy Icon Current Level setup

Suppose you have a capsule set up similar to the one shown in Figure 20. You can enter the following values in the Copy Icon Transformer Controls window:

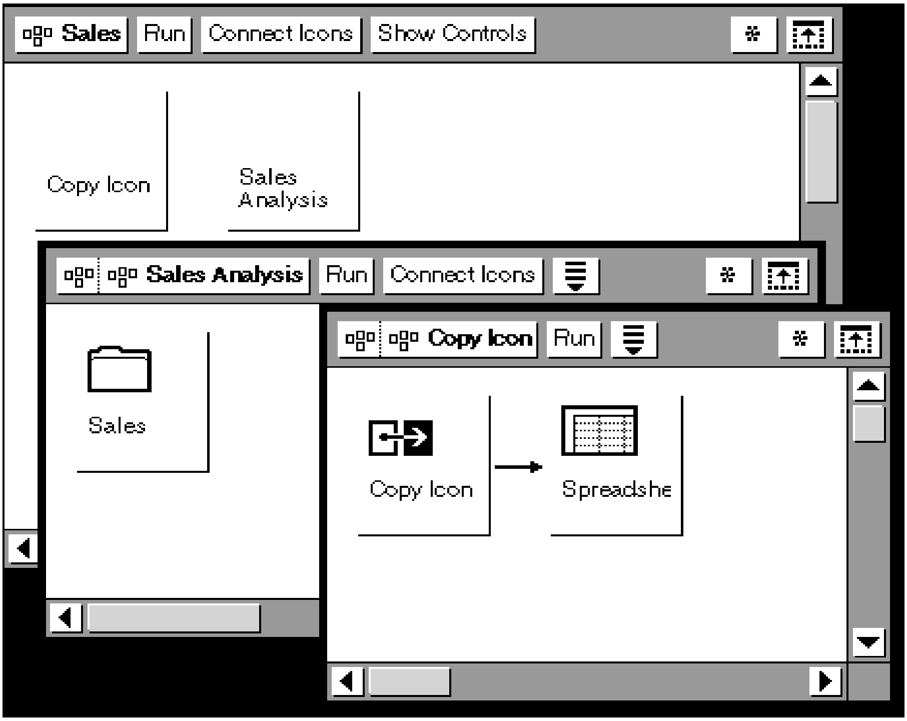
Parameters	Accounts|Sales, Sales Information|Totals	Name of icon(s) to copy
	Updated Sales Total	Name of destination folder
	Y	Overwrite Y/N
		Common source container

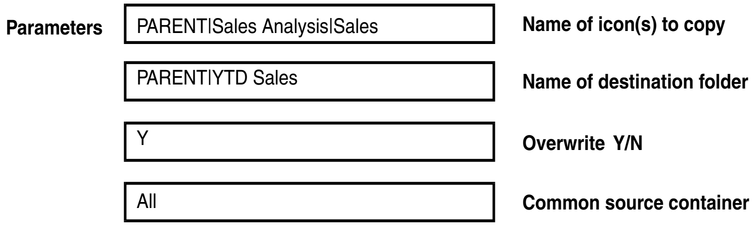
Because these containers are embedded in the same container as the transformer, they do not need a keyword specification.

When you run the Territory capsule, the transformer locates the Sales folder in the Accounts capsule, and the Totals folder in the Sales Information capsule, and copies them to the Updated Sales Total folder. When processing completes, the results appear in the Output 1 region and in the Copy Icon Log spreadsheet connected to the transformer.

PARENT Level

The PARENT keyword specifies that the search begins in the container that is one level above the container that holds the Copy Icon transformer. If you want to search for a folder or envelope on the PARENT level, you must begin the path name in the `Name of container(s) to clear` field with the PARENT keyword.





Parameters

PARENT|Sales Analysis|Sales — Name of icon(s) to copy

PARENT|YTD Sales — Name of destination folder

Y — Overwrite Y/N

All — Common source container

Figure 21. Sample Copy Icon PARENT setup

If you have several embedded capsules, it is possible to move up the directory tree from one container to another in the application by using multiple references to PARENT separated by a vertical bar.

Figure 21 illustrates the PARENT level concept. In this example, the Copy Icon transformer is in the Copy Icon capsule. If you want to copy the Sales folder in the embedded Sales Analysis capsule to a destination folder called YTD Sales, then specify the path name and the destination folder.

When you click on the `Run` button in the example shown in Figure 21, the Copy Icon transformer performs the following procedure:

1. Begins the search for the Sales folder by first checking the container one level above the container that holds the Trends capsule.
2. Locates the Sales Analysis capsule in the Trends capsule.
3. Finds the Sales folder in the Sales Analysis capsule and copies it into the YTD Sales folder.
4. Places the results in the transformer Output 1 region and in the Copy Icon Log connected to the transformer.

DESKTOP Level

The DESKTOP keyword allows users to search and copy icons on the desktop. When you search for an icon from the desktop level, the parameter path name you specify in the `Name of icon(s) to copy` field must start with the DESKTOP keyword. Similarly, if you want to copy the icon to a container on your desktop, the parameter in the `Name of destination folder` field must start with the DESKTOP keyword.

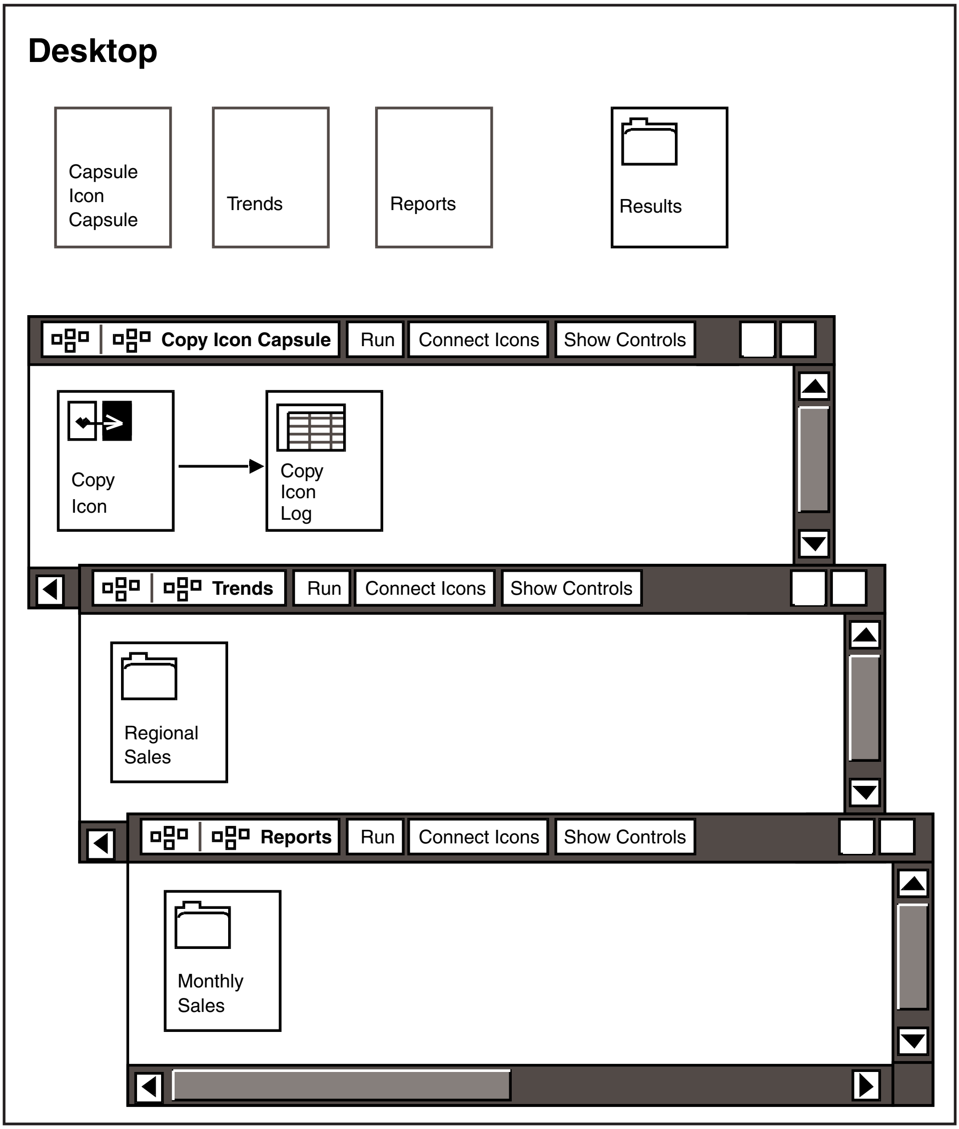
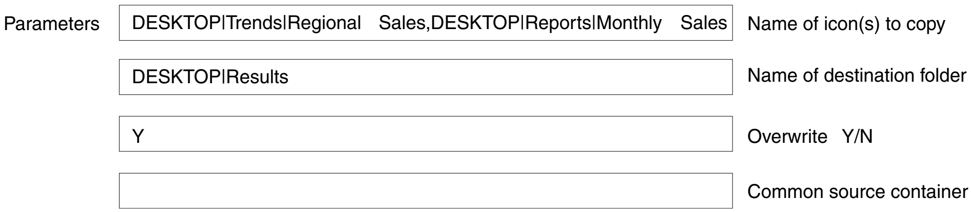


Figure 22. Sample Copy Icon DESKTOP setup

Figure 22 illustrates the DESKTOP level concept. For example, if you want to copy a folder called Regional Sales, which is in a capsule called Trends on the desktop, and a folder called Monthly Sales in the capsule called Reports on the desktop, into a folder called Results, you must specify the path name and destination folder:



Parameters

DESKTOP|Trends|Regional Sales,DESKTOP|Reports|Monthly Sales — Name of icon(s) to copy

DESKTOP|Results — Name of destination folder

Y — Overwrite Y/N

Common source container

When you click on the `Run` button in the Copy Icon capsule shown in Figure 22, the Copy Icon transformer performs the following procedure:

1. Begins the search at the desktop level. Locates and copies the Regional Sales folder in the Trends capsule to the Results folder.
2. Returns to the desktop level, locates and copies the Monthly Sales folder to the Results folder.
3. Places the results in the Output 1 region and in the Copy Icon Log connected to the transformer.

Join

The Join transformer allows you to join data from two tables. The results appear in ascending, alphanumeric, or chronological order. If your report requires a different display order, you must use an intermediate Sort transformer before copying the results to a destination icon.

The following terms are used frequently in this section:

- *Joining columns* refers to combining the columns of common data from different tables into one column of output data.
- *Common data* refers to data of a similar nature, such as dates, prices, or shipments. Any additional columns containing data that are not specified in the parameter fields are appended to the output.

Parameters

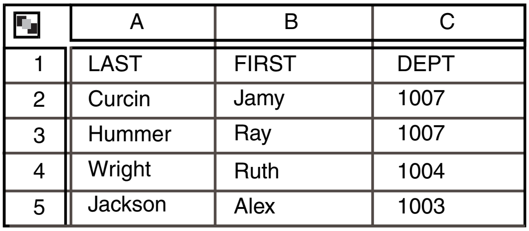
The Join transformer has the following parameters:

Join Columns in First/Second Table

These parameters identify and arrange the column letters you want to combine in the first and second tables. You can type the column letters in uppercase or lowercase.

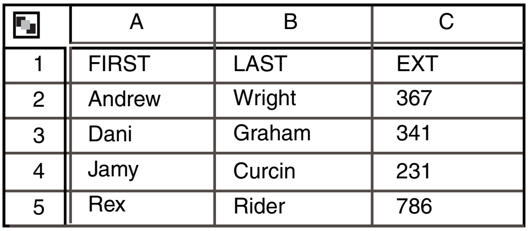
For example, the following illustration shows two tables that contain similar data (although in reverse order) in columns A and B.

Table 1



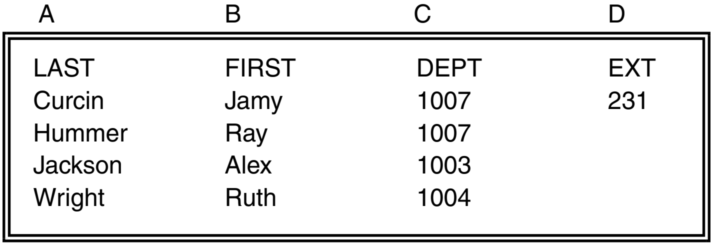
	A	B	C
1	LAST	FIRST	DEPT
2	Curcin	Jamy	1007
3	Hummer	Ray	1007
4	Wright	Ruth	1004
5	Jackson	Alex	1003

Table 2



	A	B	C
1	FIRST	LAST	EXT
2	Andrew	Wright	367
3	Dani	Graham	341
4	Jamy	Curcin	231
5	Rex	Rider	786

To join these two tables, type `a, b` in the `Join columns in first table` field and `b, a` in the `Join columns in second table` field. When the Join transformer completes processing, column A from the first table merges with column B from the second table; column B from the first table merges with column A from the second table:



A	B	C	D
LAST	FIRST	DEPT	EXT
Curcin	Jamy	1007	231
Hummer	Ray	1007	
Jackson	Alex	1003	
Wright	Ruth	1004	

Figure 23. Join transformer results

These results are based on an inner join with one specified heading row. For more information, see “Number of Heading Rows” on page 45 and “Join Type” on page 46.

The Join transformer does not remove duplicate data from tables before joining them. For example, if Table 1 in the previous example contains two rows of data for *Jackson, Alex*, then both would appear in the output regardless of which join type you specify. For information on a transformer that removes data, or allows you to specify particular data in the output, see “Clean” on page 20 or “Select” on page 92.

Specifying the Columns to Enter: The Join transformer will not run if you enter a column letter in the parameter field that is outside the range of data. For example, if you try to join two spreadsheets that contain information in columns A, B, and C, and you specify column D in the parameter fields, you will receive an error message when you run the transformer.

If a column within the range is empty, then the empty column will be incorporated in the output when the transformer is run. For example, if you join two spreadsheets that have data in columns A, B, and D, then column C will appear in the output as an empty column.

Determining the Number of Columns to Enter: The `Join columns in first table` and `Join columns in second table` fields require the same number of column letters. That is, if the `Join columns in first table` field contains two column letters, then the `Join columns in second table` field must contain two column letters. If you do not use an equal number of columns, an error message appears on your desktop, or, if applicable, in the capsule run log.

Arranging the Column Order: The order of the column letters in the `Join columns in first table` and `Join columns in second table` fields should result in joining the columns of common data. As an example, if you are joining tables that contain employee names, be sure that the column letters that represent the first and last names from each table are entered in the appropriate order:

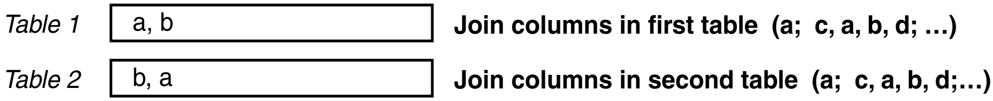


Table 1 a, b **Join columns in first table (a; c, a, b, d; …)**

Table 2 b, a **Join columns in second table (a; c, a, b, d;…)**

The Join transformer does not recognize whether you are joining the correct columns, only the similarities or differences of the data in the specified columns. You can join columns of uncommon data, although the results are likely to have minimal value.

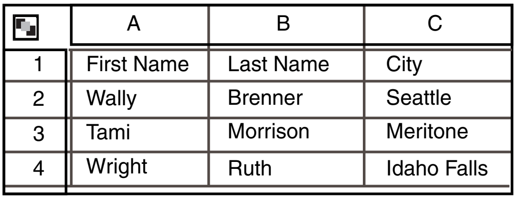
Number of Heading Rows

The `Number of heading rows` field allows you to specify how many rows are used for column headings. If you do not specify any rows, then column headings are merged and sorted with the data. The default value is 0, which means the data contains no heading rows.

All tables must contain the same number of heading rows. The headings from the first table determine the order of the headings in the output. The remaining headings from each table display in the order they appear in the input regions.

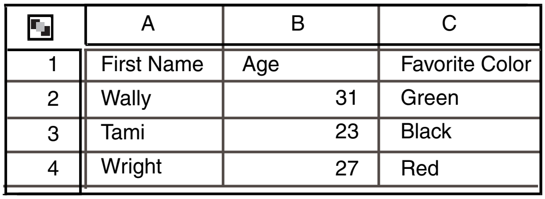
For example, suppose you want to join the following two tables based on column A, and one heading row.

Table 1



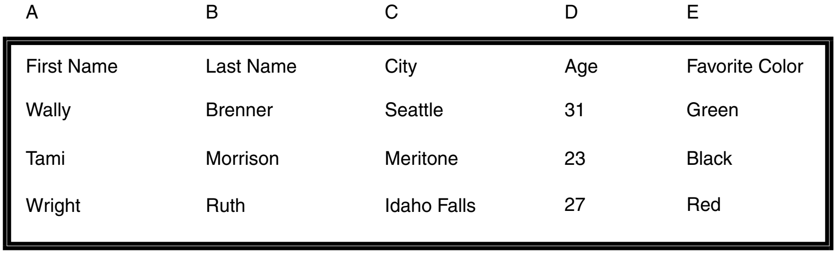
	A	B	C
1	First Name	Last Name	City
2	Wally	Brenner	Seattle
3	Tami	Morrison	Meritone
4	Wright	Ruth	Idaho Falls

Table 2



	A	B	C
1	First Name	Age	Favorite Color
2	Wally	31	Green
3	Tami	23	Black
4	Wright	27	Red

When you run the transformer, the results appear as shown here:



A	B	C	D	E
First Name	Last Name	City	Age	Favorite Color
Wally	Brenner	Seattle	31	Green
Tami	Morrison	Meritone	23	Black
Wright	Ruth	Idaho Falls	27	Red

Join Type

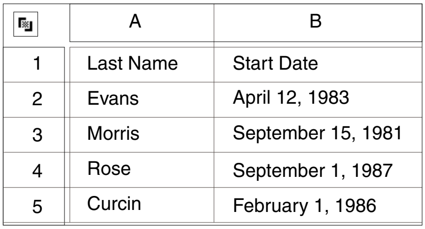
The `Join type` parameter allows you to determine the method for combining data by specifying one of four join types:

- Outer (default)
- Left
- Right
- Inner

Only the first letter is necessary to identify the join type. For example, type `L` for left. You can use uppercase or lowercase letters. The default output appears in ascending alphanumeric order.

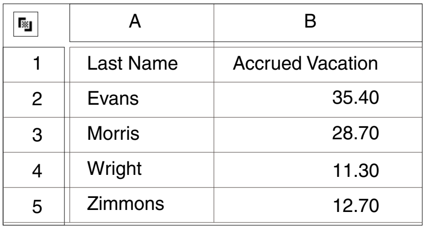
Outer: An outer join uses all of the data from both tables and inserts it into the Output Data display area. For example, you have two tables with the following information:

Table 1



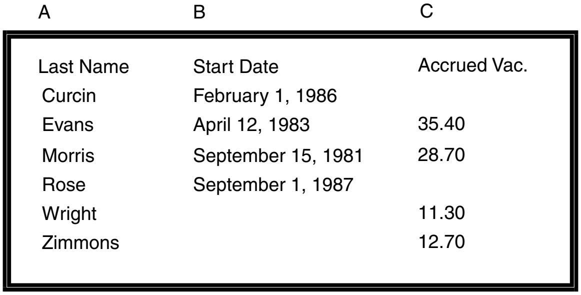
	A	B
1	Last Name	Start Date
2	Evans	April 12, 1983
3	Morris	September 15, 1981
4	Rose	September 1, 1987
5	Curcin	February 1, 1986

Table 2



	A	B
1	Last Name	Accrued Vacation
2	Evans	35.40
3	Morris	28.70
4	Wright	11.30
5	Zimmons	12.70

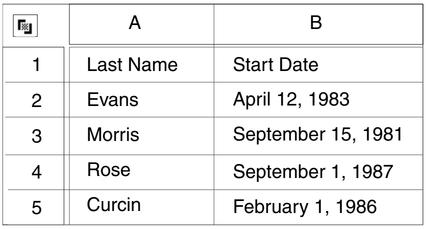
Specifying an outer join produces the results shown in the following example. In this case, the data from the tables is joined based on column A. All data from both tables appears in the output.



A	B	C
Last Name	Start Date	Accrued Vac.
Curcin	February 1, 1986	
Evans	April 12, 1983	35.40
Morris	September 15, 1981	28.70
Rose	September 1, 1987	
Wright		11.30
Zimmons		12.70

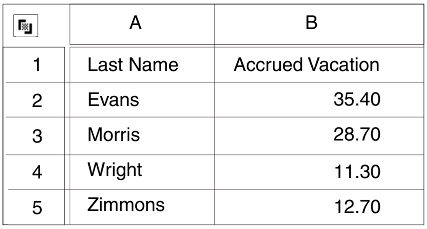
Left: Based on the specified column letters, a left join uses all of the data from the first table, then extracts from the second table only the data that matches the first table. For example, you have two tables with the following information:

Table 1



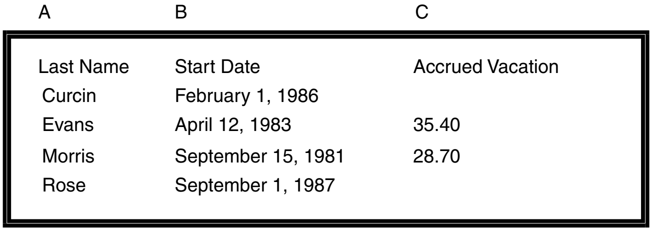
	A	B
1	Last Name	Start Date
2	Evans	April 12, 1983
3	Morris	September 15, 1981
4	Rose	September 1, 1987
5	Curcin	February 1, 1986

Table 2



	A	B
1	Last Name	Accrued Vacation
2	Evans	35.40
3	Morris	28.70
4	Wright	11.30
5	Zimmons	12.70

Specifying a left join results in the data from the second table being compared to the specified data columns in the first table. In this case, the data from the tables is merged based on column A in Table 1. *Wright* and *Zimmons* do not appear in the output because the names do not appear in the first table.

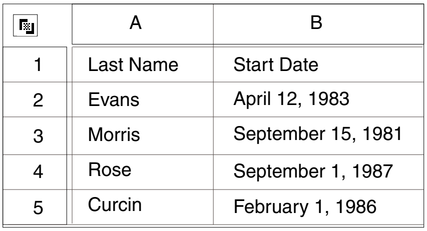


A	B	C
Last Name	Start Date	Accrued Vacation
Curcin	February 1, 1986	
Evans	April 12, 1983	35.40
Morris	September 15, 1981	28.70
Rose	September 1, 1987	

Right: Based on the specified column letter, a right join uses all of the data from the second table, then extracts the data from the first table that matches the second table. The data is then sent to the Output Data display area.

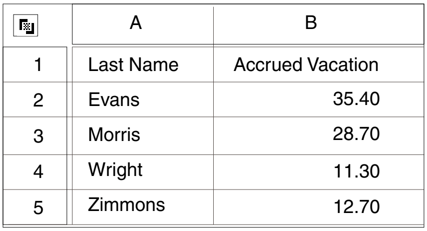
For example, you have two tables with the following information:

Table 1



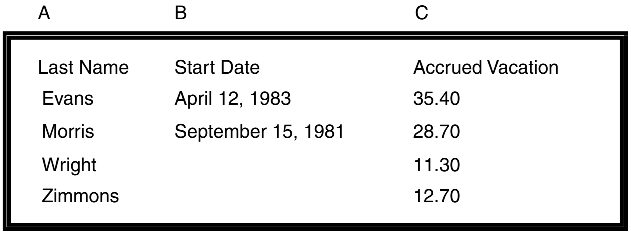
	A	B
1	Last Name	Start Date
2	Evans	April 12, 1983
3	Morris	September 15, 1981
4	Rose	September 1, 1987
5	Curcin	February 1, 1986

Table 2



	A	B
1	Last Name	Accrued Vacation
2	Evans	35.40
3	Morris	28.70
4	Wright	11.30
5	Zimmons	12.70

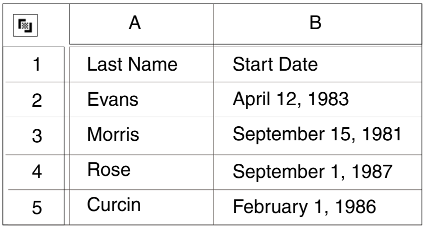
In this case, the data from the tables is merged based on column A in Table 2. *Rose* and *Curcin* do not appear in the output because the names do not appear in the first table.



A	B	C
Last Name	Start Date	Accrued Vacation
Evans	April 12, 1983	35.40
Morris	September 15, 1981	28.70
Wright		11.30
Zimmons		12.70

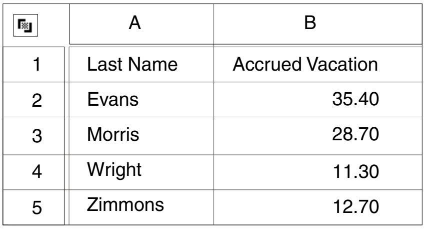
Inner: Based on the specified columns, an inner join uses only the data that matches in both tables. For example, you have two tables with the following information:

Table 1



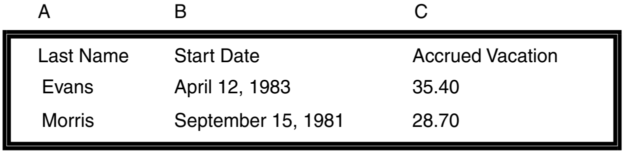
	A	B
1	Last Name	Start Date
2	Evans	April 12, 1983
3	Morris	September 15, 1981
4	Rose	September 1, 1987
5	Curcin	February 1, 1986

Table 2



	A	B
1	Last Name	Accrued Vacation
2	Evans	35.40
3	Morris	28.70
4	Wright	11.30
5	Zimmons	12.70

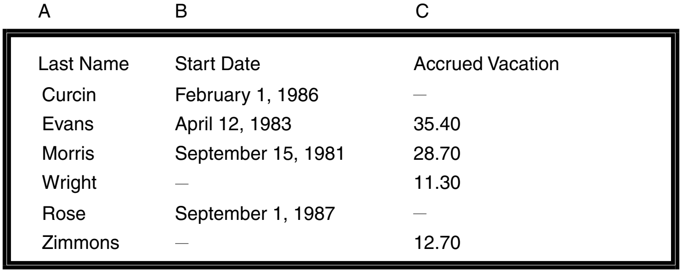
In this case, only the names *Evans* and *Morris* appear in the output because these are the only names that appear in both tables.



A	B	C
Last Name	Start Date	Accrued Vacation
Evans	April 12, 1983	35.40
Morris	September 15, 1981	28.70

Replacement Value for Nulls

The `Replacement value for nulls` parameter enables you to replace all the null values in a table with a unique character such as a dash (-):



A	B	C
Last Name	Start Date	Accrued Vacation
Curcin	February 1, 1986	—
Evans	April 12, 1983	35.40
Morris	September 15, 1981	28.70
Wright	—	11.30
Rose	September 1, 1987	—
Zimmons	—	12.70

The transformer recognizes only null values that are produced by a function of the table, not those you enter. For example, the `Replacement value for nulls` entry would replace an N/A returned by a query. However, it would not replace an N/A that you typed in a spreadsheet cell.

Region Controls

The choices in the `Display Data For` field are:

- Table 1

- Table 2
- Output Data

Table 1 and Table 2 identify the regions where you transfer source data; that is, the two tables that you want to join. Output Data identifies the region where the data appears after it is processed.

Example

This section provides an example of a capsule application that uses the Join transformer to join two tables of data. The icons must be placed so that data is transferred in the proper order, and the *Table 1* and *Table 2* regions are identified in the Arrow Options window. For information on entering information in the Arrow Options window, see “Chapter 1. Getting Started with Transformers,” on page 1, or the *Capsule User's Guide*.

Figure 24 shows an example of a capsule application that is set up to join two tables.

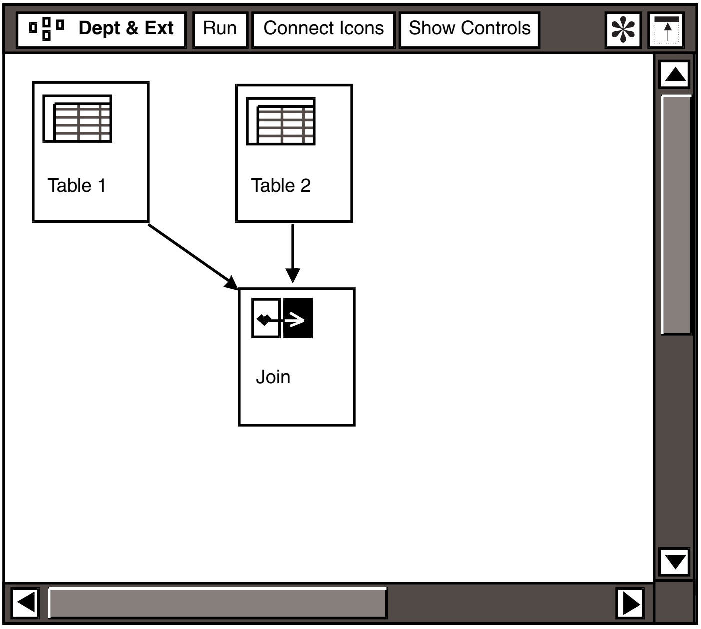


Figure 24. Join example

The following tables illustrate the data that is transferred to the Table 1 and Table 2 regions in the Join transformer. The tables should be joined based on column A or B, or based on columns A and B, because they contain common data.

Table 1

□	A	B	C
1	Last	First	Dept
2	Evans	Wally	1007
3	Morris	Tami	1009
4	Wright	Ruth	1008
5	Curcin	Jamy	1009
6	Graham	Mac	1009

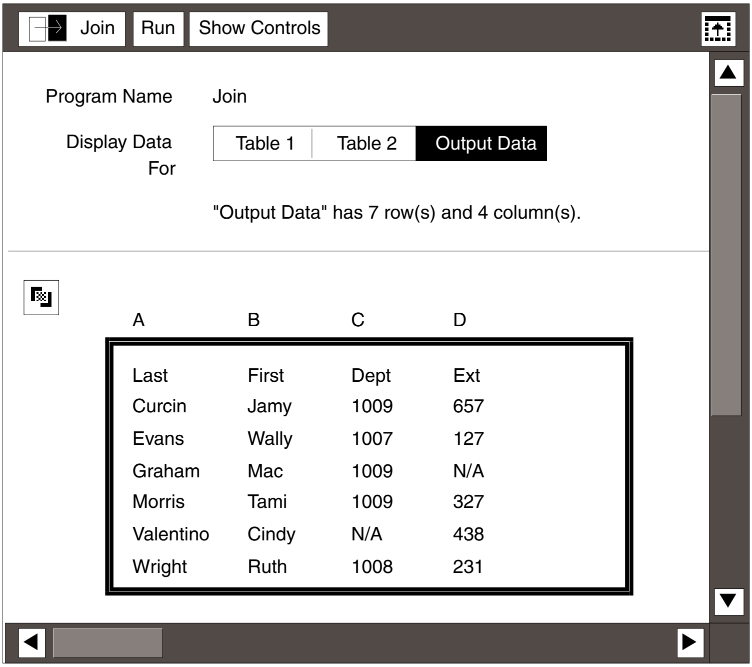
Table 2

□	A	B	C
1	Last	First	Ext
2	Evans	Wally	127
3	Morris	Tami	327
4	Wright	Ruth	231
5	Curcin	Jamy	657
6	Valentino	Cindy	438

To create the sample Join transformer:

1. Click on the `Show Controls` button in the Join transformer window header.
2. Type `a, b` in the `Join columns in first table` field to join these two columns with the data in Table 2.
3. Type `a, b` in the `Join columns in second table` field to join these two columns with the data in Table 1.
4. Type `1` in the `Number of heading rows` field to omit the first row from the join process.
5. Type `outer` in the `Join type` field to include employee names in both tables.
6. Type `N/A` in the `Replacement value for nulls` field to show that information was not available for those values.
7. Close the Transformer Controls window and the Join transformer window.
8. Click outside the Capsule window to deselect the Join transformer.
9. Click on the `Run` button in the Capsule window header.

The output appears as shown in Figure 25.



Program Name Join

Display Data For: Table 1 | Table 2 | **Output Data**

"Output Data" has 7 row(s) and 4 column(s).

A	B	C	D
Last	First	Dept	Ext
Curcin	Jamy	1009	657
Evans	Wally	1007	127
Graham	Mac	1009	N/A
Morris	Tami	1009	327
Valentino	Cindy	N/A	438
Wright	Ruth	1008	231

Figure 25. Join output

Label

The Label transformer lets you create mailing labels by retrieving and formatting names and addresses from your database or a spreadsheet. You can then send the output to a printer that contains sheets of labels in its paper tray.

Each row of the input data is used to construct a label. Each column that contains values can occupy a separate line on the label, or you can combine several consecutive columns on a single line, such as first name, last name, and title.

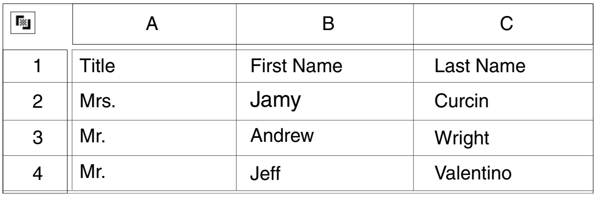
Parameters

The Label transformer has the following parameters:

Number of Heading Rows

This parameter specifies the number of heading rows in the table; in most instances these rows are omitted from the output. The default value is 0; that is, all heading rows are transferred to the output with the data rows.

For example, suppose you want to use the following data to create name tags. The first row is a heading row and, because you would not want to include this information on a name tag, type `1` into the `Number of heading rows` field.



	A	B	C
1	Title	First Name	Last Name
2	Mrs.	Jamy	Curcin
3	Mr.	Andrew	Wright
4	Mr.	Jeff	Valentino

Number of Columns Per Page

This parameter specifies the number of columns to print on each page. This number is based on the number of columns contained on a sheet of labels. The default value is 2. The maximum is 100.

Number of Lines Per Label

This parameter specifies the number of lines to print for each address. The default value is 9. Be sure that this value includes enough blank lines to ensure that each address begins on a separate label.

For example, if a sheet contains 10 labels measuring 2 x 4.25 inches each, and each address requires four lines, then you need to type 13 in the `Number of lines per label` field to ensure that the addresses begin on separate labels. Figure 26 on page 54 provides recommended settings for various label sizes.

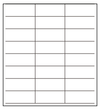
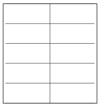
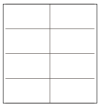
	Fi ed np t		ariable np t	
	Label size (inches):	2.75 1.5	Font size (points):	10
	Labels per page:	24	Line height:	0
	ol mns per page:	3	Lines per label:	8
	Labels per col mn:	8		
			Font size (points):	12
	Margin width (inches):	0.5, 0.5, 0.25, 0	Line height:	1.35
	Tab settings:	32, 64	Lines per label:	6
	Label size (inches):	4.25 2		
	Labels per page:	10	Font size (points):	10
	ol mns per page:	2	Line height:	1
	Labels per col mn:	5	Lines per label:	12
	Margin width (inches):		Font size (points):	12
	10 point fonts	.75, .75, .60, 0	Line height:	1
	12 point fonts	.75, .75, .75, .25	Lines per label:	10
	Tab settings:	44		
	Label size (inches):	4.25 2.75	Font size (points):	10
	Labels per page:	8	Line height:	1
	ol mns per page:	2	Lines per label:	15
	Labels per col mn:	4		
			Font size (points):	12
	Margin width (inches):	1, 1, 1, 0	Line height:	1.275
	Tab settings:	44	Lines per label:	12
T	: Margin order is listed according to left, right, top, and bottom.			

Figure 26. Recommended settings for printing on various label sizes

Number of Labels Per Column

This parameter specifies the number of labels to print in each column on the page. A new sheet of labels is used each time the printer prints the specified number of labels. The number of labels per page determines the number of sheets that are generated when you run the capsule.

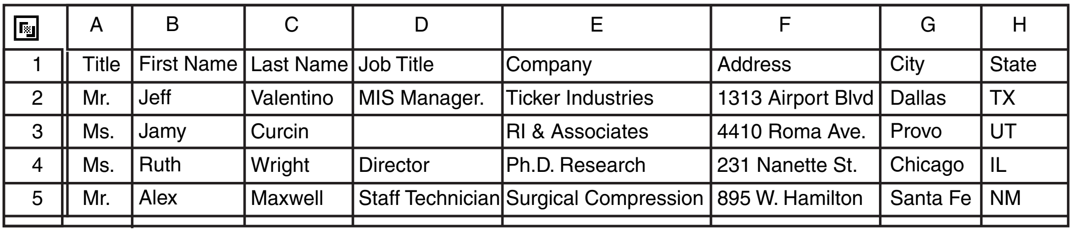
Columns to Compress: Group 1 Through Group 5

These parameters allow you to combine two or more columns of input data into a single row on the address label. Only adjacent columns can be compressed; that is, columns A and B can be compressed, but columns A and C cannot. If you do not specify any columns to compress, each column containing data is printed on a separate line. You can specify up to five groups; column letters must be entered in lowercase.

The transformer inserts one blank space between each column of data. If you want to insert additional spaces or characters in the address, you must enclose them in quotation marks. For example, to include a colon, type `ATTN:.`

If you use this method to add text to an address, you must also manually insert a space inside the quotation marks to separate the text from the data in the adjoining column. If there is no data in the adjoining column, then the characters inside the quotation marks are omitted from the output.

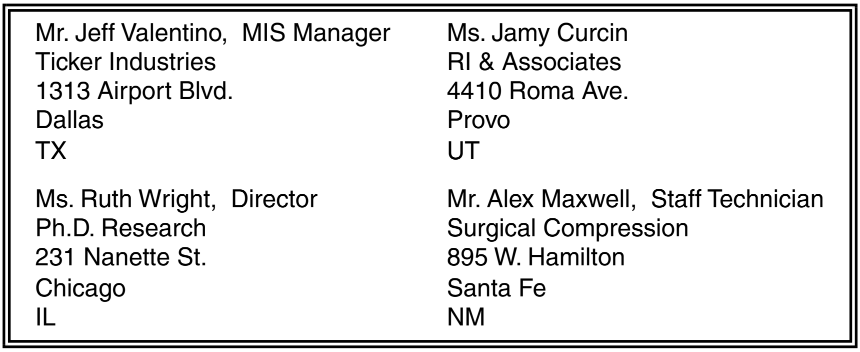
For example, suppose you want to use the following data to create address labels.



▨	A	B	C	D	E	F	G	H
1	Title	First Name	Last Name	Job Title	Company	Address	City	State
2	Mr.	Jeff	Valentino	MIS Manager.	Ticker Industries	1313 Airport Blvd	Dallas	TX
3	Ms.	Jamy	Curcin		RI & Associates	4410 Roma Ave.	Provo	UT
4	Ms.	Ruth	Wright	Director	Ph.D. Research	231 Nanette St.	Chicago	IL
5	Mr.	Alex	Maxwell	Staff Technician	Surgical Compression	895 W. Hamilton	Santa Fe	NM

To compress columns A, B, C, and D and add a comma between the last name and the job title, type `a, b, c, ', ', d` in the `Columns to compress: Group 1` field.

The comma is followed by a space, and both are enclosed in quotation marks. The results appear as shown here:



```
Mr. Jeff Valentino,  MIS Manager        Ms. Jamy Curcin
Ticker Industries                       RI & Associates
1313 Airport Blvd.                      4410 Roma Ave.
Dallas                                  Provo
TX                                      UT

Ms. Ruth Wright,  Director              Mr. Alex Maxwell,  Staff Technician
Ph.D. Research                          Surgical Compression
231 Nanette St.                         895 W. Hamilton
Chicago                                 Santa Fe
IL                                      NM
```

Note that the comma was omitted in the name of the second label because column D is blank.

To compress columns G and H, type `g, ', ', h` in the `Columns to compress: Group 2` field.

Region Controls

The `Display Data For` field in the Label window contains the following choices:

- Input Data
- Output Data

Input Data identifies the region where you transfer source data; that is, the area where you transfer the name and address information to create mailing labels. Output Data is the area where formatted labels appear after you run the transformer.

Example

This section explains how to use the Label transformer in a capsule application to generate address labels for form letters. The data in the Output Data region transfers as a single paragraph to a destination icon, such as a Text or Spreadsheet icon.

If you use a Text icon as the output icon, you must set the tabs and margins so the addresses appear in the correct format when they are transferred. For information on setting character and paragraph formats in a Text icon, see the *Text User's Guide*.

A new page is printed each time the specified limit is reached for the number of labels per page. Figure 27 shows an example of a capsule application that is set up to create labels.

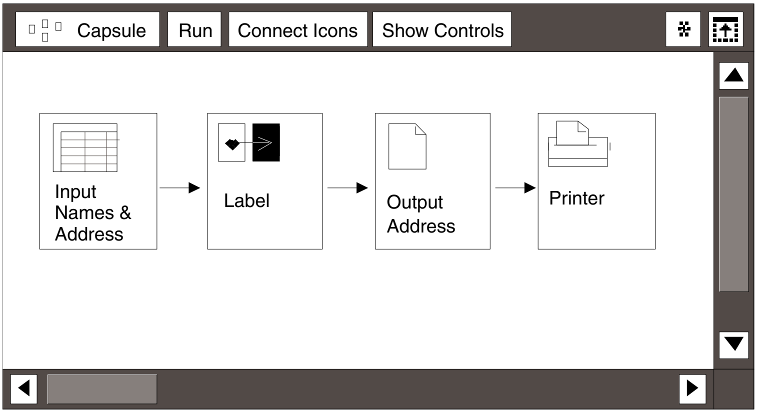
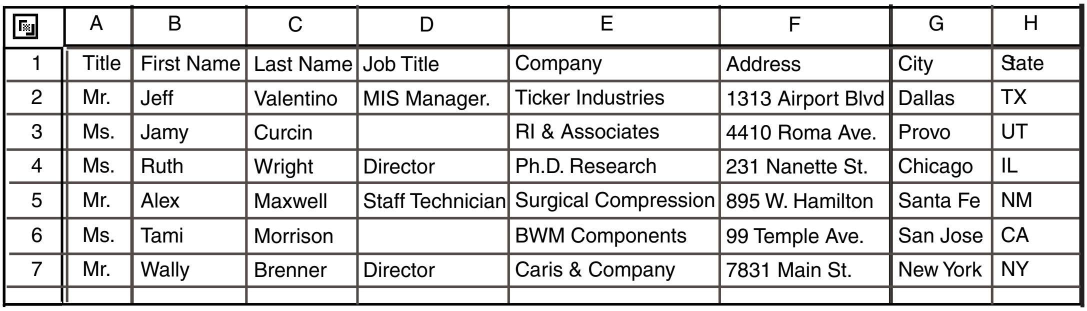


Figure 27. Label example

Suppose you have the following spreadsheet data. You can use this data to create labels in 10–point font on a sheet containing 24 labels.

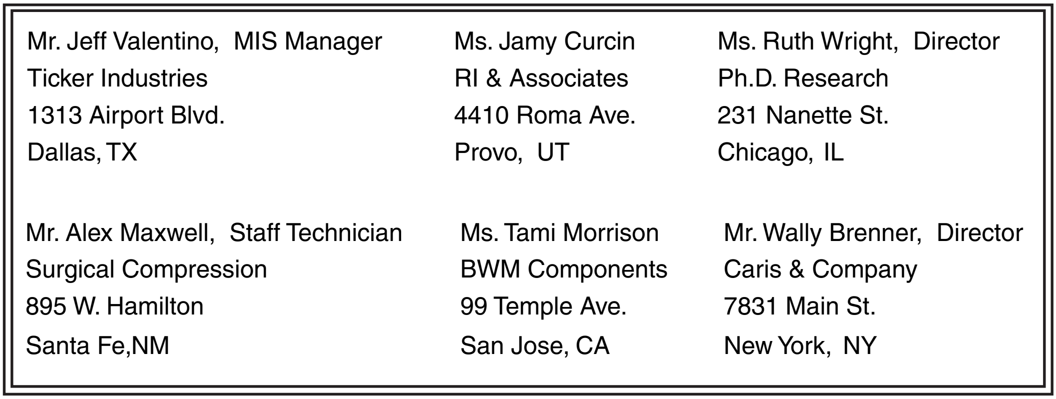


	A	B	C	D	E	F	G	H
1	Title	First Name	Last Name	Job Title	Company	Address	City	State
2	Mr.	Jeff	Valentino	MIS Manager.	Ticker Industries	1313 Airport Blvd	Dallas	TX
3	Ms.	Jamy	Curcin		RI & Associates	4410 Roma Ave.	Provo	UT
4	Ms.	Ruth	Wright	Director	Ph.D. Research	231 Nanette St.	Chicago	IL
5	Mr.	Alex	Maxwell	Staff Technician	Surgical Compression	895 W. Hamilton	Santa Fe	NM
6	Ms.	Tami	Morrison		BWM Components	99 Temple Ave.	San Jose	CA
7	Mr.	Wally	Brenner	Director	Caris & Company	7831 Main St.	New York	NY

To set up the sample Label transformer:

1. Click on the `Show Controls` button in the Label transformer window header.
2. Type `1` in the `Number of heading rows` field to exclude row 1 from the process.
3. Type `3` in the `Number of columns per page` field to print three columns of labels on each sheet.
4. Type `9` in the `Number of lines per label` field to ensure that each address is on a separate label.
5. Type `8` in the `Number of labels per column` field to account for eight labels in each column, and as a counter for starting a new page.
6. Type `a, b, c, ', ', d` in the `Columns to compress: Group 1` field to combine columns A through D into one line, and insert a comma and space between columns C and D. Columns A, B, and C will automatically be separated by one space.
7. Type `g, ', ', h` in the `Columns to compress: Group 2` field to combine columns G and H into one line, and insert a comma and space between them.
8. Close the Transformer Controls window and the Label transformer window.
9. Click outside the Capsule window to deselect the Label transformer.
10. Click on the `Run` button in the Capsule window header.

The output appears as shown in Figure 28.



Mr. Jeff Valentino, MIS Manager
Ticker Industries
1313 Airport Blvd.
Dallas, TX

Ms. Jamy Curcin
RI & Associates
4410 Roma Ave.
Provo, UT

Ms. Ruth Wright, Director
Ph.D. Research
231 Nanette St.
Chicago, IL

Mr. Alex Maxwell, Staff Technician
Surgical Compression
895 W. Hamilton
Santa Fe,NM

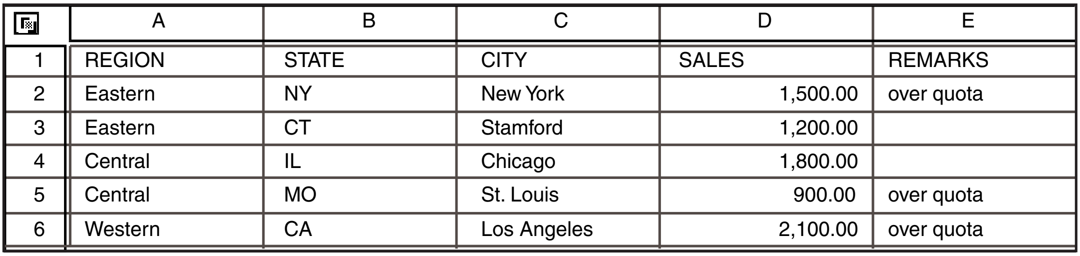
Ms. Tami Morrison
BWM Components
99 Temple Ave.
San Jose, CA

Mr. Wally Brenner, Director
Caris & Company
7831 Main St.
New York, NY

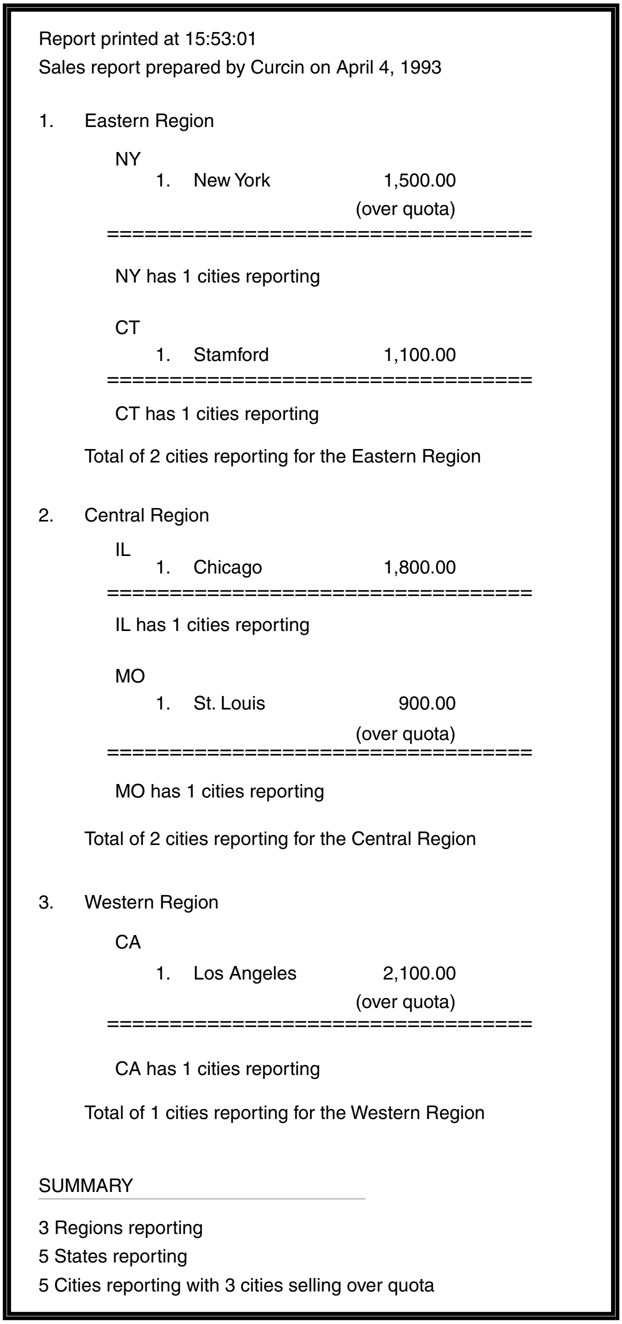
Figure 28. Label output

Merge

The Merge transformer enables you to combine source data with text to produce a report. For example, you can merge the following spreadsheet data with formats from the Text tool to create a document similar to the one shown in Figure 29 on page 59.



	A	B	C	D	E
1	REGION	STATE	CITY	SALES	REMARKS
2	Eastern	NY	New York	1,500.00	over quota
3	Eastern	CT	Stamford	1,200.00	
4	Central	IL	Chicago	1,800.00	
5	Central	MO	St. Louis	900.00	over quota
6	Western	CA	Los Angeles	2,100.00	over quota



```
Report printed at 15:53:01
Sales report prepared by Curcin on April 4, 1993

1.   Eastern Region

        NY
            1.   New York               1,500.00
                                     (over quota)
       ===================================

        NY has 1 cities reporting

        CT
            1.   Stamford               1,100.00
       ===================================

        CT has 1 cities reporting

     Total of 2 cities reporting for the Eastern Region

2.   Central Region
        IL
            1.   Chicago                1,800.00
       ===================================

        IL has 1 cities reporting

        MO
            1.   St. Louis                900.00
                                     (over quota)
       ===================================

        MO has 1 cities reporting

     Total of 2 cities reporting for the Central Region

3.   Western Region

        CA
            1.   Los Angeles            2,100.00
                                     (over quota)
       ===================================

        CA has 1 cities reporting

     Total of 1 cities reporting for the Western Region


SUMMARY
___________________________

3 Regions reporting
5 States reporting
5 Cities reporting with 3 cities selling over quota
```

Figure 29. Sample document created with the Merge transformer

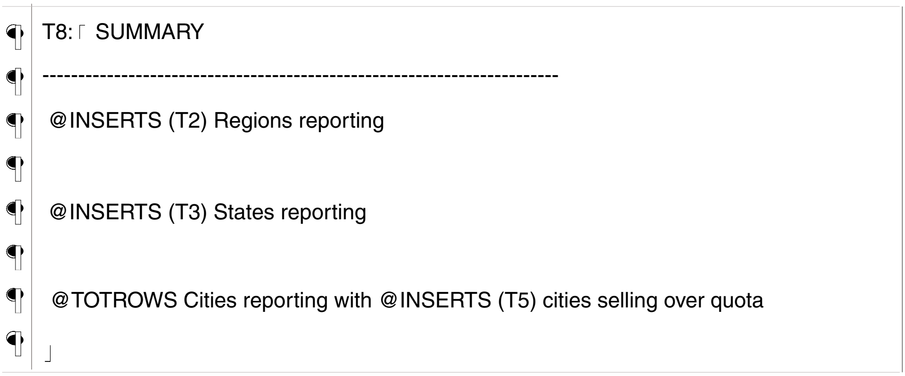
This sample document is used throughout this chapter to fully explain the concepts of the Merge transformer.

Creating Templates

After you decide how you want the source data to appear in the output Text icon, you must create templates that apply format information to specified columns and rows. You create these templates using regions in a source Text icon.

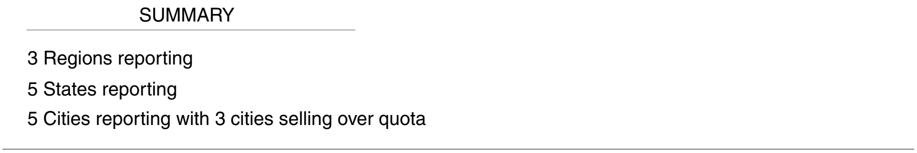
Templates can include information such as tab stops, new lines, and new paragraphs, as well as @-keywords that generate system-defined variables, such as @DATE, @TIME, and @COL. (See Table 5 on page 61 for more information on using @-keywords.) Each template is assigned to an input region in the Merge transformer window.

For example, the following illustration of a partial text window shows a template (T8) that creates a summary at the end of a generated report.



```
¶  T8: ⌜ SUMMARY
¶  --------------------------------------------------------------------
¶   @INSERTS (T2) Regions reporting
¶
¶   @INSERTS (T3) States reporting
¶
¶   @TOTROWS Cities reporting with @INSERTS (T5) cities selling over quota
¶  ⌟
```

When you run the transformer, the last lines of the report appear like the one shown in the following illustration. New paragraphs, text, and sum totals are inserted as directed by the template. You must set typeface, margin, and tab settings in the output Text icon.



```
               SUMMARY
3 Regions reporting
5 States reporting
5 Cities reporting with 3 cities selling over quota
```

The Merge transformer recognizes a special set of @-keywords in the templates, as described in Table 5 on page 61. When it reads one of these @-keywords, the program inserts the appropriate data. @-keywords must be written in uppercase.

Table 5. @-Keywords and their definitions

@-Keyword	Definition
@COL(*n*)	The value in column *n* of the current input row; for example, *@COL(a)*.
@USER	The user's desktop name.
@DATE	The current date.
@TIME	The current time.
@TOTROWS	The row number of the current incoming data row. The first data row is number 1. Heading rows are not numbered.
@CURROWS(T*n*)	The number of incoming data rows since the last insertion of template T*n* or since the beginning of the report (>=1), where T*n* is the template number.
@INSERTS	The number of times that this template has been inserted (>=1).
@INSERTS(T*n*)	The number of times that template T*n* has been inserted (>=0), where T*n* is the template number.

After you create the templates in the source Text icon, you can set up the Merge transformer to apply the template information to the source data.

Parameters

The Merge parameters are used differently than in other transformers. Except for the first parameter, each one specifies an instruction for its associated template, T1 through T15. Examples of these instructions appear for the first eight template parameters. You enter one instruction for each template you use.

For each row of incoming data, the Merge transformer checks all instructions in the order in which they appear in the parameters fields. If one or more of the instructions apply to the current data row, it locates and applies the appropriate template. For more information on instructions, see “Insert Template T1 Through T15” on page 61.

Number of Heading Rows

This parameter specifies the number of rows used as column headings for the input data. The specified number of rows are skipped and do not appear in the output report. The default value is 0 rows of column headings. If you do not specify this instruction, all incoming rows of data are included in the merge.

Insert Template T1 Through T15

These parameters specify the instructions for each template inserted from the source Text icon. The instructions tell the transformer when to apply the information supplied in the template.

For example, if you type `At last row` in the `Insert template T1` field, the information in template T1 will be inserted as the last row in the output. Table 6 briefly describes each instruction.

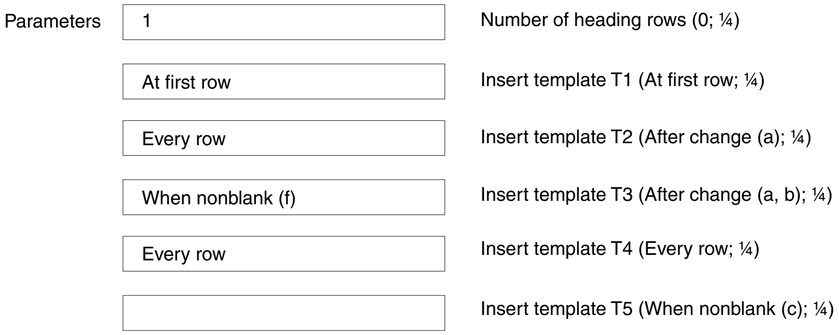
Table 6. Instructions used in the Merge transformer

Instruction	Inserts the specified template
At first row	Before any incoming data rows
After change	After a change occurs in the specified columns
Every row	For each row of incoming data
When nonblank	When it applies to the data; for example, if a comment is included in the specified column, it can be applied to the Merge output
Before change	Before a change occurs in the specified columns
At last row	After any incoming data rows

You must follow certain guidelines when inserting instructions in the Merge parameters:

- You can enter only one instruction per template field. However, you can repeat an instruction several times in different template fields. For example, you can type `At last row` for T5, T10, and T13.
- You can enter any instruction in any template field. For example, you can type `Every row` in the field associated with template T2, or T5, or T11, or in any other template field.
- Instructions are not case sensitive.
- Templates are inserted at a given row when triggered by an instruction. If more than one instruction applies for a given row of data, the templates are inserted in the order that the instructions are listed in the parameter fields.

The following example shows a partial Transformer Controls window where template T1 is inserted at the beginning of the report; templates T2 and T4 apply to each row of incoming data; and, template T3 is inserted wherever it applies.



Parameters

1 — Number of heading rows (0; ¼)

At first row — Insert template T1 (At first row; ¼)

Every row — Insert template T2 (After change (a); ¼)

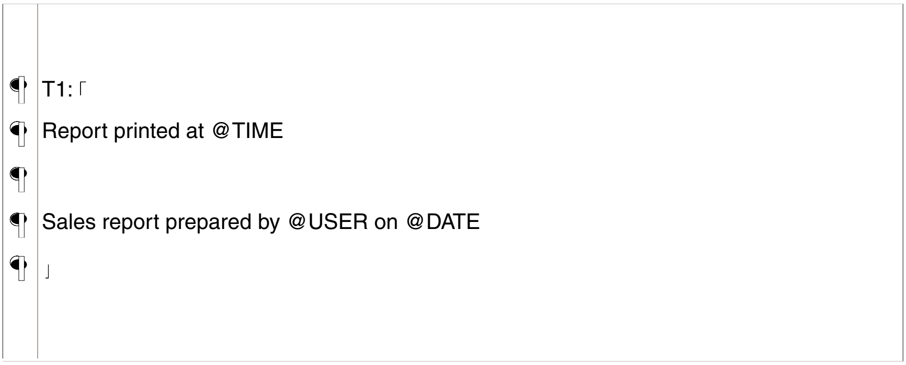
When nonblank (f) — Insert template T3 (After change (a, b); ¼)

Every row — Insert template T4 (Every row; ¼)

Insert template T5 (When nonblank (c); ¼)

The At first row instruction allows you to insert the appropriate template as the first row of data, after any heading rows and before any incoming data rows. This instruction is useful when inserting date and address information in form letters. It is also useful for generating report headers that contain report titles, your name, the date, or the time the report was created.

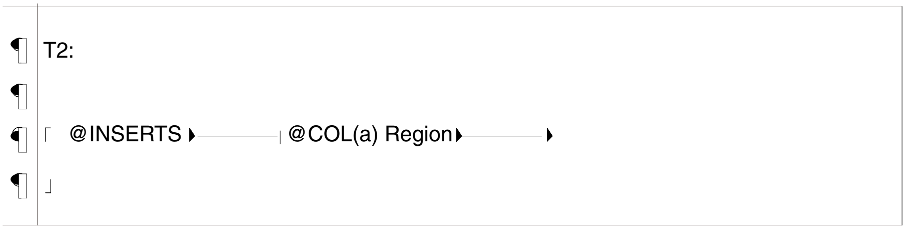
As shown in the following example, a template in the T1 parameter could result in a report providing the time it was generated (@TIME), a blank line, then a line containing the user's name (@USER) and the date (@DATE).



```
¶ T1:⌈
¶ Report printed at @TIME
¶
¶ Sales report prepared by @USER on @DATE
¶ ⌋
```

The After change instruction allows you to insert the appropriate template after a change occurs in the specified columns. The After change instruction is generally used to create group headings for the input data. You can specify a single column or several columns.

Specifying a single column locates and applies the appropriate template to a single specified column. This instruction is useful for grouping data under a common heading. For example, you can format the data so that a number appears before each group name, which allows you to keep track of the number of different groups. You can also place text after each group name, which creates a heading for the data within each group. To do this, you can create a template for the T2 parameter similar to the following example.



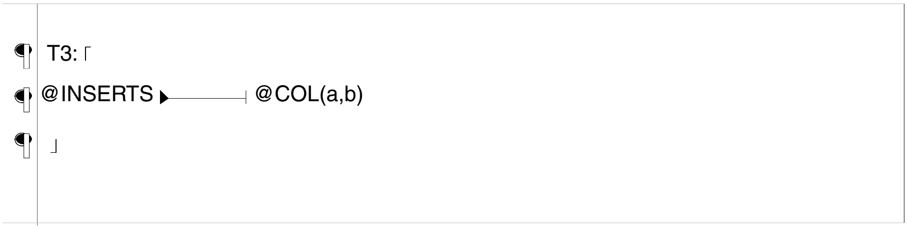
```
¶ T2:
¶
¶ ⌈  @INSERTS ▸———————| @COL(a) Region▸———————▸
¶ ⌋
```

In Figure 29 on page 59, column A contains region names. The instruction for the T2 parameter would read: *After change (a)*; that is, when the region name changes in column A, insert this template to format those data rows. The template inserts the word *Region* after each region name, and a new paragraph and tab to format the data.

Template T2 also contains two @-keywords:

> @INSERTS is replaced by a sequential number for each region name; that is, 1, 2, 3
>
> @COL(a) is replaced by the value of the data in column A; that is, the region name

Specifying several columns locates and applies the appropriate template to several specified columns. This instruction is useful for grouping common data within major groups. To do this, you create a template for the `T3` field:



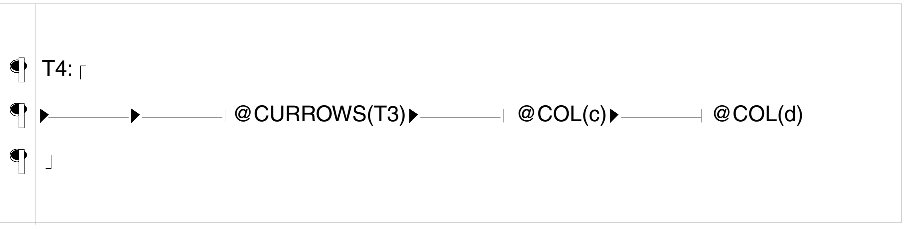
```
¶ T3: ┌
¶ @INSERTS ▸————| @COL(a,b)
¶ ┘
```

In Figure 29 on page 59, column A contains region names, and column B contains state names. The instruction for the `T3` field would read: *After change (a, b)*; that is, when the region name changes in column A or when the state name changes in column B, insert this template to format that row of data. In addition to the new-paragraph and tab stops, template T3 contains one @-keyword that replaces @COL(b) with the value of the data in column B; that is, the name of the state.

The `After change (a, b)` instruction is typically used in reports that contain groups and subgroups.

The `Every row` instruction allows you to insert the associated template for every row of incoming data.

You can format data so that subgroups of subgroups are grouped, and then present data from a specific subgroup. To do this, you create a template for the `T4` parameter similar to the one shown here:



```
¶ T4: ┌
¶ ▸————▸————| @CURROWS(T3)▸————| @COL(c)▸————| @COL(d)
¶ ┘
```

In Figure 29 on page 59, column A contains region names, column B contains state names, column C contains city names, and column D contains the sales for each city. The instruction for the `T4` parameter would read: *Every row*; that is, for each row of incoming data, insert this template to format the data.

In addition to the tab symbols, template T4 contains three @-keywords:

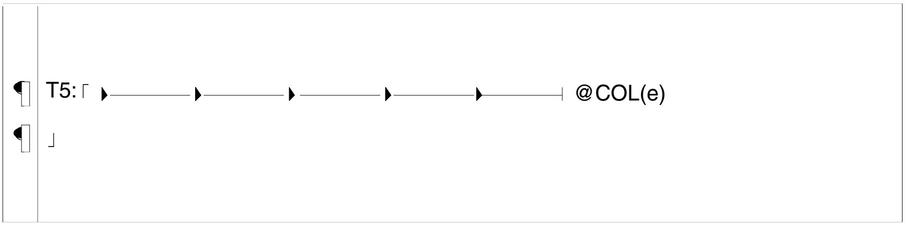
@CURROWS(T3) is replaced by the value of the number of rows of data, because template T3 was inserted

@COL(c) is replaced by the value of the data in column C; that is, the city name

@COL(d) is replaced by the value of the data in column D; that is, the sales data

The `When nonblank` instruction allows you to insert the associated template for every row where a value is detected for the specified columns. Do not insert this template if the specified column is blank. A nonblank value in any of the specified columns causes the instruction to locate and apply the template.

You can format data so that the most specific subgroup stands out in a report by including a statement whenever a criterion is met, such as additional information in another column. To do this, you can create a T5 template similar to this:

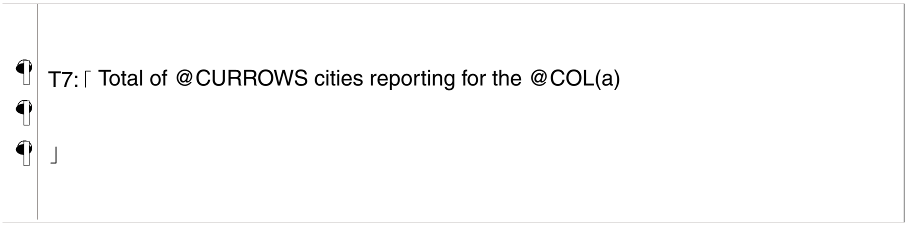


```
¶  T5:⌈ ▸────▸────▸────▸────▸────┤ @COL(e)
¶  ⌋
```

In Figure 29 on page 59, column A contains region names, column B contains state names, column C contains city names, column D contains the sales for each city, and column E contains a comment on the status of the sales. The instruction for the `T5` parameter would read: *When nonblank (e)*; that is, for each row of incoming data where a value is detected in column E, insert this template to format the data. In addition to the tab symbols, template T5 contains one @-keyword that replaces *@COL(e)* with the value of the data in column E, if the column is not blank.

The `Before change` instruction allows you to insert the associated template at the row before a change occurs in the specified columns. The `Before change` instruction is generally used to create data summaries for groups or subgroups. You can specify a single column or several columns.

Specifying a single column locates and applies the appropriate template to a single specified column. This instruction is useful for summarizing major groups of data. You can format the data by including group summaries of the data. To do this, you create a template for the T7 parameter similar to the one shown here:



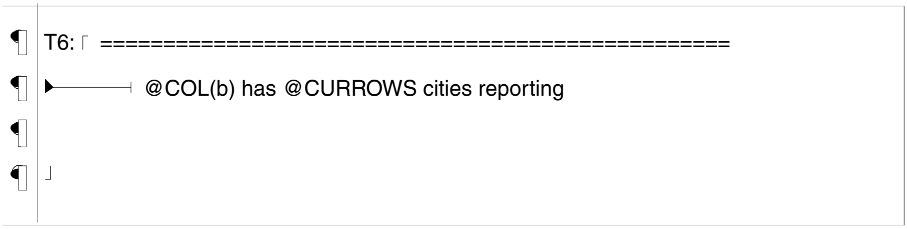
```
¶ T7:⌈ Total of @CURROWS cities reporting for the @COL(a)
¶
¶ ⌋
```

In Figure 29 on page 59, column A contains region names, column B contains state names, and column C contains city names. The instruction for the T7 parameter would read: *Before change (a)*; that is, when the region name changes in column A, insert this template to format the previous data row. In addition to the text, new line characters, and tab symbols, template T7 contains two @-keywords:

@CURROWS is replaced by the total number of cities reporting for that region

@COL(a) is replaced by the value of the data in column A; that is, the region name

Specifying several columns is useful for summarizing subgroups of data within major groups. You can format the data by including subgroup summaries of the output data. To do this, you create a template for the T6 parameter similar to the one shown here:



```
¶ T6:⌈ =================================================
¶ ▸———————⊣ @COL(b) has @CURROWS cities reporting
¶
¶ ⌋
```

In Figure 29 on page 59, column A contains region names, column B contains state names, and column C contains city names. The instruction for the T6 parameter would read: *Before change (a, b)*; that is, when the region name changes in column A or when the state name changes in column B, insert the template to format the previous data row. In addition to the text, the new-paragraph characters, and tab symbols, template T6 contains two @-keywords:

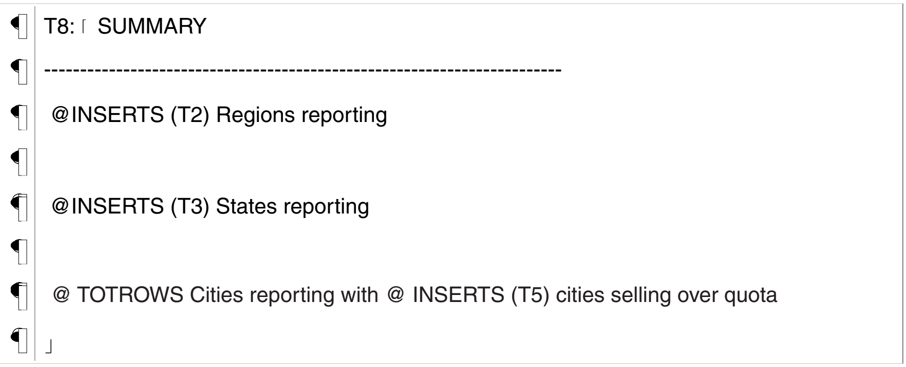
@COL(b) is replaced by the value of the data in column B; that is, the state name

@CURROWS is replaced by the number of cities reporting for that state; that is, the number of rows because template T2 was inserted into the data

The *Before change (a, b)* instruction is typically used for reports with both groups and subgroups.

The `At last row` instruction allows you to insert the associated template as the last row of data, after all of the incoming data is processed. This is useful for including summaries in a report, or closing remarks in a form letter.

To do this, you create a template for the `T8` parameter that generates a summary that appears after the last of the data in the report, similar to the following example.



```
¶ T8: ⌈ SUMMARY
¶ ---------------------------------------------------------------------
¶  @INSERTS (T2) Regions reporting
¶
¶  @INSERTS (T3) States reporting
¶
¶  @ TOTROWS Cities reporting with @ INSERTS (T5) cities selling over quota
¶ ⌋
```

In Figure 29 on page 59, column A contains region names, column B contains state names, column C contains city names, and column D contains the sales for each city. In addition to new paragraphs, template T8 used two @-keywords:

@INSERTS(T2) is replaced by the number of times template T2 was inserted

@INSERTS(T3) is replaced by the number of times template T3 was inserted

@TOTROWS is replaced by the total number of rows

@INSERTS(T5) is replaced by the number of cities reporting sales over quota

Region Controls

The `Display Data For` field in the Merge transformer contains the following choices:

- T1 through T15
- Input Data
- Output Data

T1 through T15 are the input regions you use to transfer Text templates that contain format instructions for the incoming data. In addition to text, these templates contain format characters, or symbols, such as new-line, new-page, and tab. They can also contain a special set of @-keywords.

Input Data is the area where you transfer the data that you want to merge. Output Data is the area where the results appear after you run the transformer.

Example

This section explains how to set up a capsule application using the Merge transformer.

Figure 30 on page 68 shows a capsule application that contains the format information in the source Text icon (Control Regions) and territory data in a Spreadsheet icon (Territory Data). Both icons are connected to a Merge transformer, which processes the information and creates a formatted report (Sales Report).

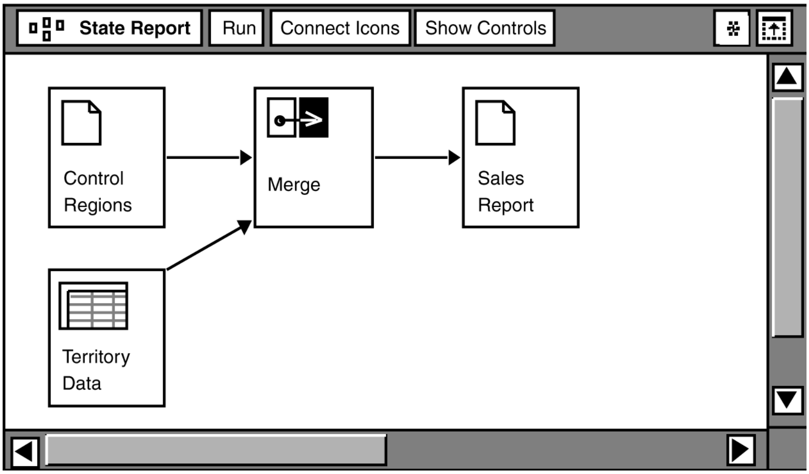
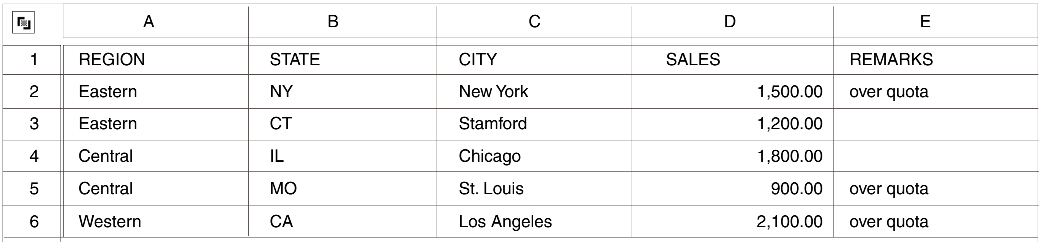


Figure 30. Merge example

The Territory Data spreadsheet contains sales data for cities in three regions, as shown in Figure 31.

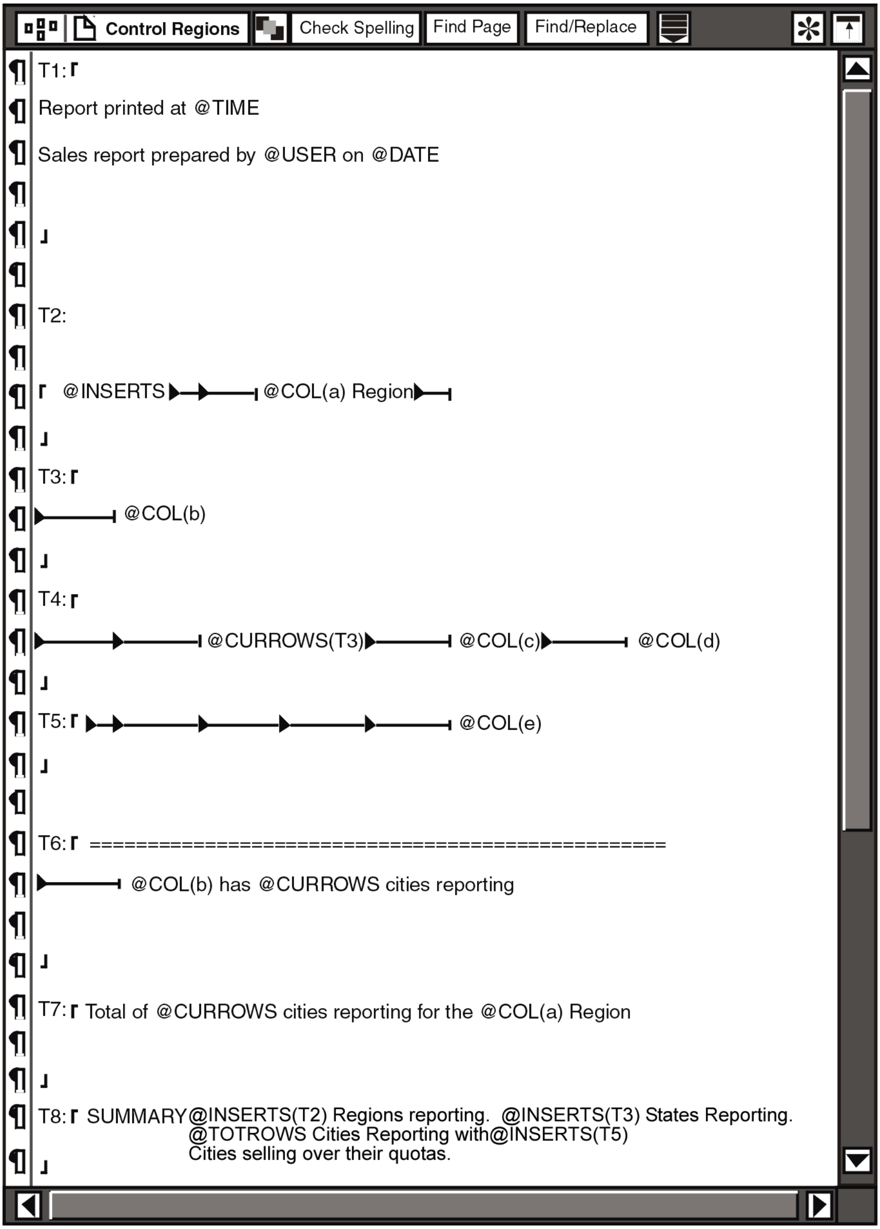


	A	B	C	D	E
1	REGION	STATE	CITY	SALES	REMARKS
2	Eastern	NY	New York	1,500.00	over quota
3	Eastern	CT	Stamford	1,200.00	
4	Central	IL	Chicago	1,800.00	
5	Central	MO	St. Louis	900.00	over quota
6	Western	CA	Los Angeles	2,100.00	over quota

Figure 31. Territory Data spreadsheet

The data in the table is used with a Control Regions document, which contains the formats for the data as it will appear in the Sales Report document. The format specified in the Control Regions document must be transferred to the Merge transformer before the table data. Figure 32 shows the contents of the Control Regions document.

In Figure 32, region markers, tab symbols, and new-paragraph characters are shown. Tab symbols and new-paragraph characters do not appear in the output. Region markers are displayed in the output, unless otherwise specified.



```
¶ T1:┌
¶ Report printed at @TIME
¶ Sales report prepared by @USER on @DATE
¶
¶ ┘
¶
¶ T2:
¶
¶ ┌  @INSERTS→→ @COL(a) Region→
¶ ┘
¶ T3:┌
¶ → @COL(b)
¶ ┘
¶ T4:┌
¶ →→ @CURROWS(T3)→ @COL(c)→ @COL(d)
¶ ┘
¶ T5:┌ →→→→→ @COL(e)
¶ ┘
¶
¶ T6:┌ ==================================================
¶ → @COL(b) has @CURROWS cities reporting
¶
¶ ┘
¶ T7:┌ Total of @CURROWS cities reporting for the @COL(a) Region
¶
¶ ┘
¶ T8:┌ SUMMARY@INSERTS(T2) Regions reporting.  @INSERTS(T3) States Reporting.
                @TOTROWS Cities Reporting with@INSERTS(T5)
                Cities selling over their quotas.
¶ ┘
```

Figure 32. Contents of the Control Regions document

Merge

First, specify the following parameters in the Merge Transformer Controls window:

1. Type `1` in the `Number of heading rows` field so that the heading row is not used in the format process.
2. Type `at first row` in the `Insert template T1` field to insert a timestamp at the beginning of the report.
3. Type `after change(a)` in the `Insert template T2` field to insert the name of each region at the appropriate point in the report.
4. Type `after change(a,b)` in the `Insert template T3` field to insert the name of each state at the appropriate point in the report.
5. Type `every row` in the `Insert template T4` field to specify that the city and sales data appear for each row.
6. Type `when nonblank(e)` in the `Insert template T5` field to print the information in the REMARKS column, when applicable.
7. Type `before change(a,b)` in the `Insert template T6` field to generate the number of cities reporting for each state.
8. Type `before change(a)` in the `Insert template T7` field to generate the number of cities reporting for each region.
9. Type `at last row` in the `Insert template T8` field to insert the report summary.
10. Close the Transformer Controls window and then close the transformer.

Next, specify the source and destination areas for the regions in the Arrow Options window.

1. Display the options for the arrow connecting the Control Regions icon to the Merge transformer.
2. In the Arrow Options window, specify the appropriate region names in the `Source Area` field.

   There must be an equal number of source areas and destination areas.
3. Specify the appropriate region names in the `Destination Area` field. The sequence in which the region names appear in this parameter must be the same as the sequence in which they are transferred to the transformer.

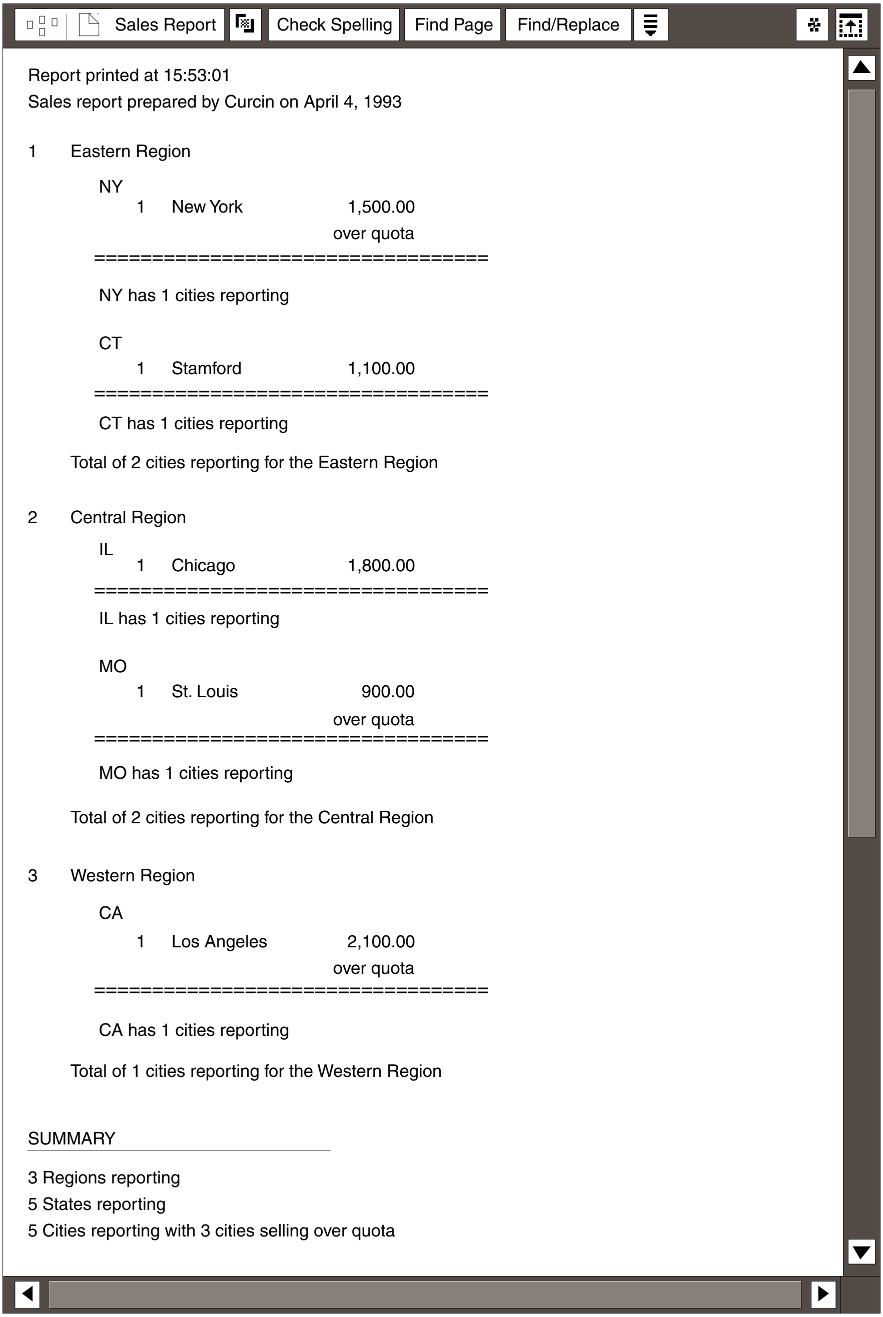
   The names must be identical to the corresponding region names in the transformer; the number of regions in the destination area must be equal to the number of regions in the source area.
4. Close the Arrow Options window connecting the Control Regions icon and the Merge transformer.
5. Display the options for the arrow connecting the Territory Data icon to the Merge transformer.
6. In the Arrow Options window, type `Input Data 16` in the `Destination Area` field. The table data must enter the Input Data area for the transformer to run.
7. Close the Arrow Options window.

8. Click outside the capsule to deselect any objects, and then click on the `Run` button in the Capsule window header.

Specify the format settings in the State Report document. Estimate the character and paragraph settings, and then modify them as necessary when the data appears in the report.

For more information on setting arrow options, see “Setting Arrow Options” on page 11, or see the *Capsule User's Guide*.

Any @-variables that you want to include in the report must be assigned in the User Input Controls window of the capsule, and in the appropriate location in the document.



```
Report printed at 15:53:01
Sales report prepared by Curcin on April 4, 1993

1      Eastern Region

          NY
               1     New York                1,500.00
                                           over quota
         ==================================

          NY has 1 cities reporting


          CT
               1     Stamford                1,100.00
         ==================================

          CT has 1 cities reporting

       Total of 2 cities reporting for the Eastern Region


2      Central Region
          IL
               1     Chicago                 1,800.00
         ==================================
          IL has 1 cities reporting


          MO
               1     St. Louis                 900.00
                                           over quota
         ==================================

          MO has 1 cities reporting

       Total of 2 cities reporting for the Central Region


3      Western Region

          CA
               1     Los Angeles             2,100.00
                                           over quota
         ==================================

          CA has 1 cities reporting

       Total of 1 cities reporting for the Western Region


SUMMARY

3 Regions reporting
5 States reporting
5 Cities reporting with 3 cities selling over quota
```

Figure 33. Completed Merge transformer example

MultiJoin

The MultiJoin transformer enables you to join data from multiple tables. It is similar to the Join transformer, except that it offers the increased flexibility of allowing you to join the data from up to five tables, rather than only two.

The MultiJoin transformer automatically sorts the data in the joined columns in ascending, alphanumeric, or chronological order. If your output requires a different display order, you must use an intermediate Sort transformer before copying the output data to a destination icon. See “Sort” on page 94 for additional information.

Parameters

The MultiJoin transformer has the following parameters:

Join Columns in Table

The `Join columns in first table` through `Join columns in fifth table` fields identify and arrange the column letters you want to combine in two or more tables. You can type the column letters in uppercase or lowercase.

The default value is `a`, which means that the results are based on the data in column A in each table. Any additional data columns are appended to the output.

Specifying the Columns to Enter: The transformer will not run if you enter a column in the input field that is outside the range of data. For example, if you try to join spreadsheets that contain information in columns A, B, and C, and you specify column D in the input fields, you will receive an error message when you run the transformer.

If a column within the range is empty, then the empty column will be incorporated in the output when the transformer is run. For example, if you join spreadsheets that have data in columns A, B, and D, then column C will appear in the output as an empty column.

Determining the Number of Columns to Enter: The `Join columns in first table` through `Join columns in fifth table` fields require the same number of column letters. For example, if the `Join columns in first table` field contains two column letters, then the `Join columns in second table` field must contain two column letters, and so on. If you do not use an equal number of columns, an error message appears on your desktop.

Arranging the Column Order: The order of the column letters in the `Join columns in first table` through `Join columns in fifth table` fields should result in joining the columns of common data. For example, if you are joining tables that contain employee names, be sure that the column letters that represent the first and last names from each table are entered in the appropriate order:

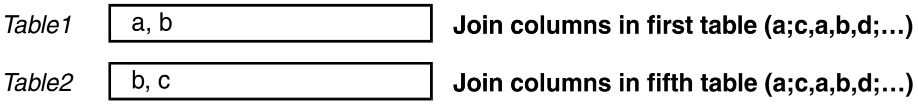


Table1	a, b	**Join columns in first table (a;c,a,b,d;...)**
Table2	b, c	**Join columns in fifth table (a;c,a,b,d;...)**

The MultiJoin transformer does not recognize whether you are joining the correct columns, only the similarities or differences of the data in the specified columns. You can join columns of unlike data, although the results are likely to have minimal value.

Removing Entries: Delete the entries from the fields you do not use. The MultiJoin transformer is unable to complete the join process, and returns an error message if a field contains a column letter for a table that does not exist.

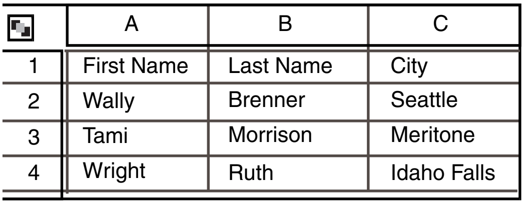
Number of Heading Rows

The `Number of heading rows` parameter allows you to specify how many rows are used for column headings. If you do not specify any rows, then column headings are merged and sorted with the data. The default value is 0, which means the data contains no heading rows.

All tables must contain the same number of heading rows. The headings from the first table determine the order of the headings in the output. The remaining headings from each table display in the order they appear in the input regions.

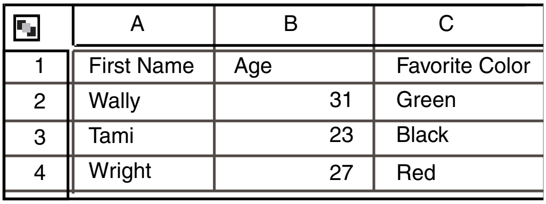
For example, suppose you want to join the following two tables based on column A, and one heading row:

Table 1



	A	B	C
1	First Name	Last Name	City
2	Wally	Brenner	Seattle
3	Tami	Morrison	Meritone
4	Wright	Ruth	Idaho Falls

Table 2



	A	B	C
1	First Name	Age	Favorite Color
2	Wally	31	Green
3	Tami	23	Black
4	Wright	27	Red

When you run the transformer, the results appear as shown here:

A	B	C	D	E
First Name	Last Name	City	Age	Favorite Color
Wally	Brenner	Seattle	31	Green
Tami	Morrison	Meritone	23	Black
Wright	Ruth	Idaho Falls	27	Red

Join Type

The `Join type` parameter allows you to specify one of four types of joins:

- Outer (default)
- Left
- Right
- Inner

Only the first letter is necessary to identify the join type. For example, type `L` for left. You can use uppercase or lowercase letters. The default output appears in ascending alphanumeric order.

Outer: An outer join uses all of the data from all of the tables and inserts it into the `Output Data` display area. For example, you have three tables with the following information:

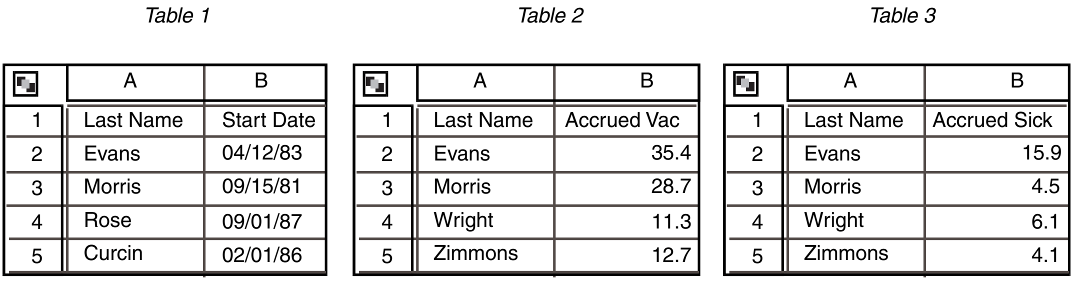


Table 1

	A	B
1	Last Name	Start Date
2	Evans	04/12/83
3	Morris	09/15/81
4	Rose	09/01/87
5	Curcin	02/01/86

Table 2

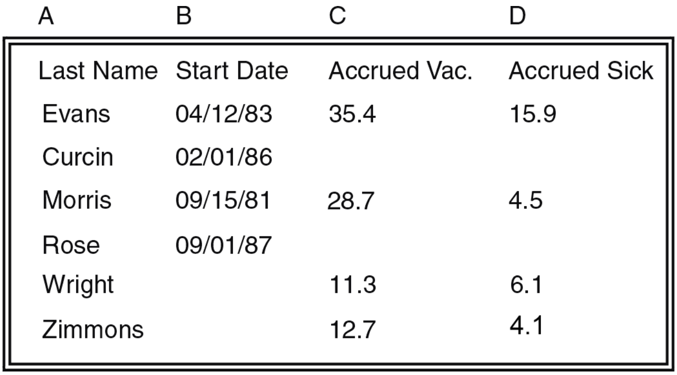
	A	B
1	Last Name	Accrued Vac
2	Evans	35.4
3	Morris	28.7
4	Wright	11.3
5	Zimmons	12.7

Table 3

	A	B
1	Last Name	Accrued Sick
2	Evans	15.9
3	Morris	4.5
4	Wright	6.1
5	Zimmons	4.1

Figure 34. Input tables for Join transformer example

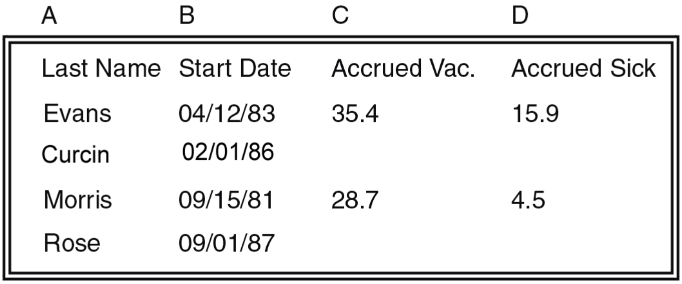
Specifying an outer join for these tables produces the results shown in the following example. In this case, the join is based on the data in column A.



A	B	C	D
Last Name	Start Date	Accrued Vac.	Accrued Sick
Evans	04/12/83	35.4	15.9
Curcin	02/01/86		
Morris	09/15/81	28.7	4.5
Rose	09/01/87		
Wright		11.3	6.1
Zimmons		12.7	4.1

Left: Based on the specified column letters, a left join uses all of the data from the first table, then extracts from the second table only the data that matches the first table.

For example, suppose you have the three tables shown in Figure 34 on page 75. Specifying a left join results in all the data from the other tables being matched to the specified column letters in the first table. In this case, the data from the tables is merged based on column A in Table 1. *Wright* and *Zimmons* do not appear in the output because the names do not appear in the first table.

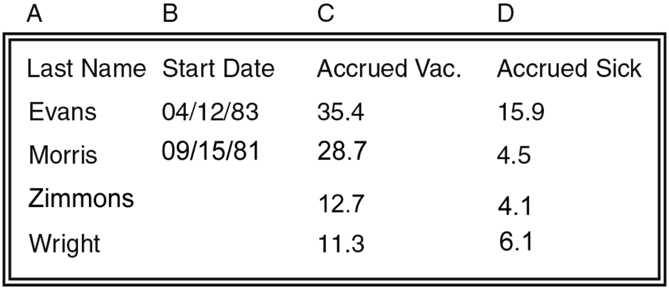


A	B	C	D
Last Name	Start Date	Accrued Vac.	Accrued Sick
Evans	04/12/83	35.4	15.9
Curcin	02/01/86		
Morris	09/15/81	28.7	4.5
Rose	09/01/87		

Right : Based on the specified column letters, a right join uses all of the data from the second table, then extracts the data from the first table that matches the second table. The resulting data is then matched to the third table. The data from the previous join that matches the third table is then joined with the third table, which is then matched to the fourth table. The data from the previous join that matches the fourth table is then joined with the fourth table, which is then matched to the fifth table. The data from the previous join that matches the fifth table is joined with the fifth table and then sent to the Output Data display area.

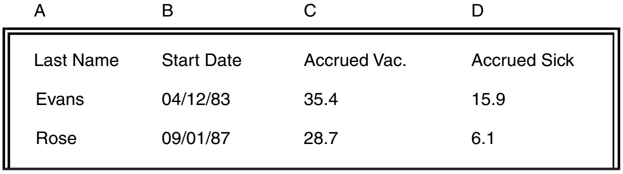
In a right join, data in a previous table must match with data in each successive table to appear in the output. Although there might be data from an earlier table that matches the last table, it will not appear in the output if the data does not match with the data in an intermediate table. All of the data in the last table appears in the output.

For example, you have the three tables shown in Figure 34 on page 75. Specifying a right join results in all the information from the other tables being matched to the specified column letters in the last table. In this case, the data from the tables is joined based on column A in Table 3. The names *Curcin and Rose* do not appear in the output because the names do not appear in the last table, and the start date for *Morris* does not appear because *Morris* does not appear in the second (intermediate) table.



A	B	C	D
Last Name	Start Date	Accrued Vac.	Accrued Sick
Evans	04/12/83	35.4	15.9
Morris	09/15/81	28.7	4.5
Zimmons		12.7	4.1
Wright		11.3	6.1

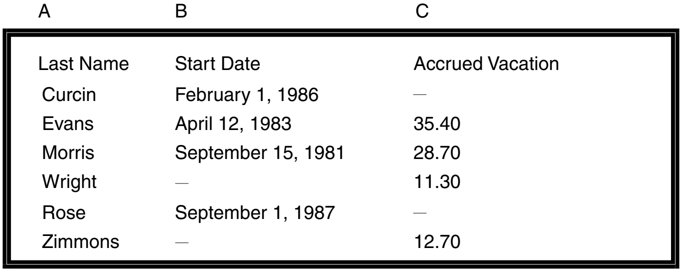
Inner: An inner join uses only the data that matches in all the tables, based on the specified columns. For example, you have the three tables shown in Figure 34 on page 75. Specifying an inner join results in all the information from each table being matched to the specified column letters in each of the other tables. In this case, the data from the tables is merged based on column A. The names *Evans* and *Rose* appear in the output because these are the only names that appear in all of the tables.



A	B	C	D
Last Name	Start Date	Accrued Vac.	Accrued Sick
Evans	04/12/83	35.4	15.9
Rose	09/01/87	28.7	6.1

Replacement Value for Nulls

This parameter enables you to replace all the null values in a table with a unique character such as a dash (-), as shown in the following examples:



A	B	C
Last Name	Start Date	Accrued Vacation
Curcin	February 1, 1986	—
Evans	April 12, 1983	35.40
Morris	September 15, 1981	28.70
Wright	—	11.30
Rose	September 1, 1987	—
Zimmons	—	12.70

The transformer recognizes only null values that are produced by a function of the table, not those you enter. For example, the value in the `Replacement value for nulls` field would replace an N/A returned by a query; however, it would not replace an `N/A` that you typed in a spreadsheet cell.

Region Controls

The `Display Data For` field in the MultiJoin transformer contains the following choices:

- Table 1 through Table 5
- Output Data

Tables 1 through 5 identify the regions where you transfer source data; that is, the tables that you want to join. Output Data identifies the region where the data appears after it is processed.

You must always use Table 5 as the last input region whenever you join fewer than five tables; otherwise, the desired results will not appear in the output region when you run the transformer.

For example, for two tables you would assign the first table to the Table 1 display area and the second table to the Table 5 display area. The MultiJoin transformer requires at least two columns. Table 7 illustrates which regions to use for a specified number of tables.

Table 7. Input locations based on the number of tables

Number of tables	**Use Display Data For options**			
2	Table 1	Table 5		
3	Table 1	Table 2	Table 5	
4	Table 1	Table 2	Table 3	Table 5

Table 7. Input locations based on the number of tables

Number of tables	Use Display Data For options				
5	Table 1	Table 2	Table 3	Table 4	Table 5

Example

Figure 35 shows an example of a capsule application that uses a MultiJoin transformer to join the data from four tables, then transfer the results to a text document.

The icons are placed so that data is transferred in the proper order; the regions (Table 1 through Table 5, and Output Data) are identified in the Arrow Options window. (For information on entering information in the Arrow Options window, see “Chapter 1. Getting Started with Transformers,” on page 1 or the *Capsule User's Guide*.)

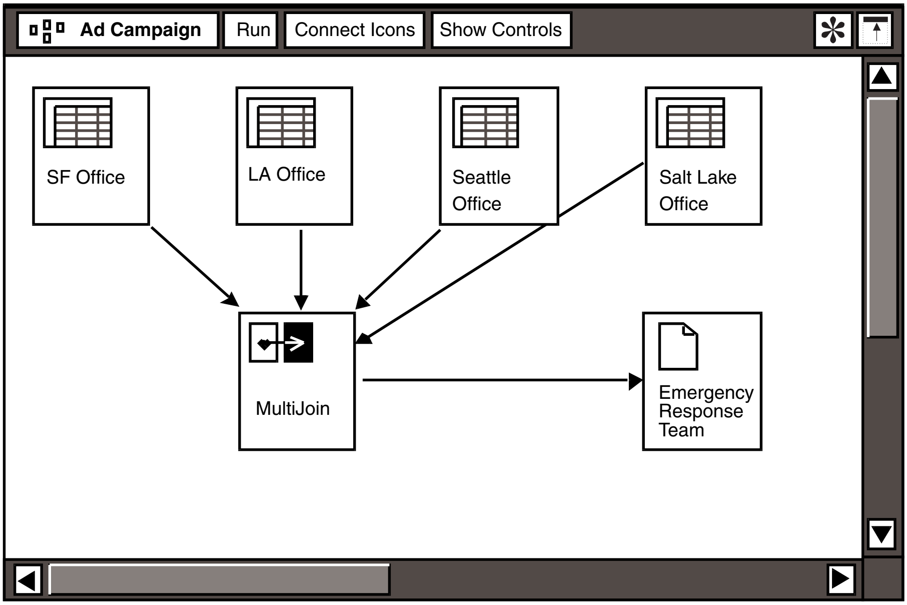
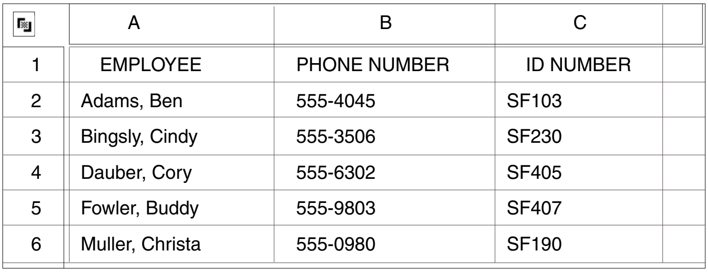


Figure 35. MultiJoin transformer example

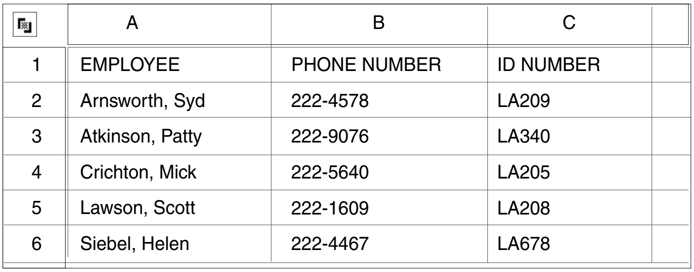
The following tables illustrate the data that is transferred to the Table 1 through Table 5 regions in the MultiJoin transformer. The tables should be joined based on column A, or based on columns A, B, and C because they contain common data.

Table 1
(SF Office)



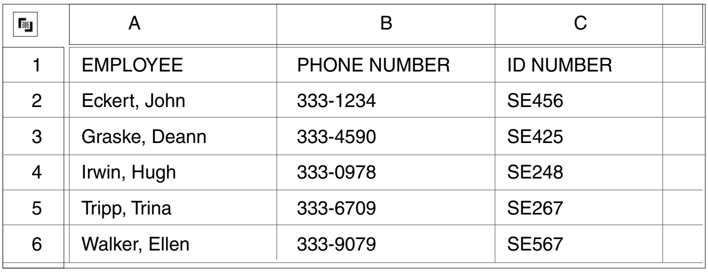
	A	B	C
1	EMPLOYEE	PHONE NUMBER	ID NUMBER
2	Adams, Ben	555-4045	SF103
3	Bingsly, Cindy	555-3506	SF230
4	Dauber, Cory	555-6302	SF405
5	Fowler, Buddy	555-9803	SF407
6	Muller, Christa	555-0980	SF190

Table 2
(LA Office)



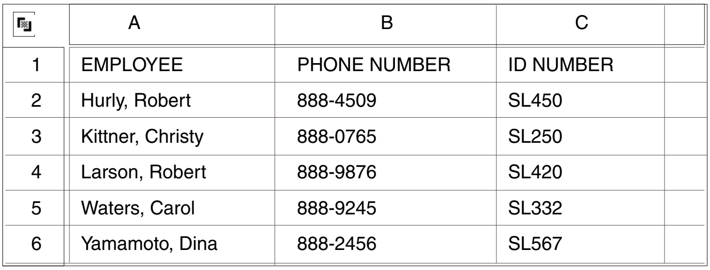
	A	B	C
1	EMPLOYEE	PHONE NUMBER	ID NUMBER
2	Arnsworth, Syd	222-4578	LA209
3	Atkinson, Patty	222-9076	LA340
4	Crichton, Mick	222-5640	LA205
5	Lawson, Scott	222-1609	LA208
6	Siebel, Helen	222-4467	LA678

Table 3
(Seattle Office)



	A	B	C
1	EMPLOYEE	PHONE NUMBER	ID NUMBER
2	Eckert, John	333-1234	SE456
3	Graske, Deann	333-4590	SE425
4	Irwin, Hugh	333-0978	SE248
5	Tripp, Trina	333-6709	SE267
6	Walker, Ellen	333-9079	SE567

Table 5
(Salt Lake Office)



	A	B	C
1	EMPLOYEE	PHONE NUMBER	ID NUMBER
2	Hurly, Robert	888-4509	SL450
3	Kittner, Christy	888-0765	SL250
4	Larson, Robert	888-9876	SL420
5	Waters, Carol	888-9245	SL332
6	Yamamoto, Dina	888-2456	SL567

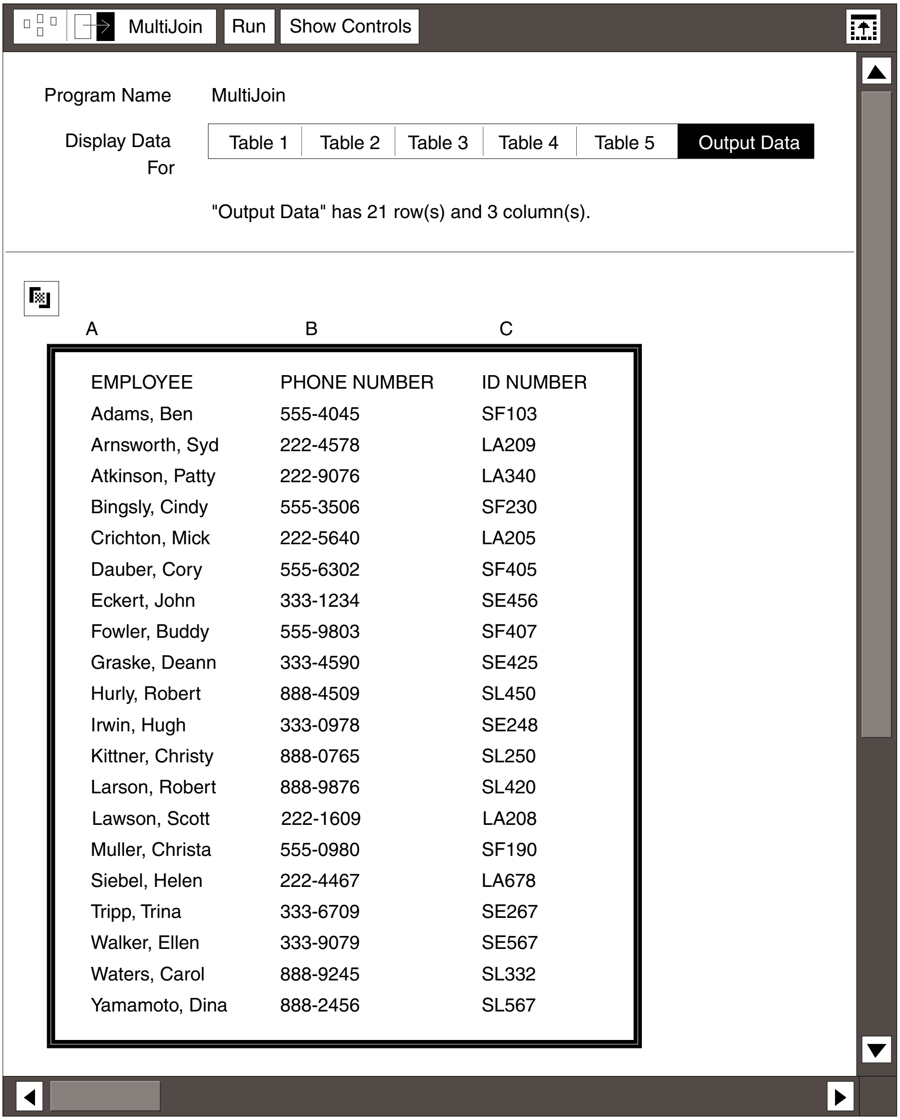
To create the MultiJoin transformer example:

1. Click on the `Show Controls` button in the MultiJoin transformer window.
2. Type `a, b, c` in the `Join columns in first table` field.

3. Type `a, b, c` in the `Join columns in second table` field.
4. Type `a, b, c` in the `Join columns in third table` field.
5. Type `a, b, c` in the `Join columns in fifth table` field.
6. Type `1` in the `Number of heading rows` field to omit the first row from the join process.

   Leave `outer` in the `Join Type` field.
7. Close the Transformer Controls window and the Join transformer window.
8. Click outside the Capsule window to deselect the Join transformer.
9. Click on the `Run` button in the Capsule window header.

The output appears as shown in Figure 36 on page 82.



Program Name MultiJoin

Display Data For: Table 1 | Table 2 | Table 3 | Table 4 | Table 5 | Output Data

"Output Data" has 21 row(s) and 3 column(s).

A	B	C
EMPLOYEE	PHONE NUMBER	ID NUMBER
Adams, Ben	555-4045	SF103
Arnsworth, Syd	222-4578	LA209
Atkinson, Patty	222-9076	LA340
Bingsly, Cindy	555-3506	SF230
Crichton, Mick	222-5640	LA205
Dauber, Cory	555-6302	SF405
Eckert, John	333-1234	SE456
Fowler, Buddy	555-9803	SF407
Graske, Deann	333-4590	SE425
Hurly, Robert	888-4509	SL450
Irwin, Hugh	333-0978	SE248
Kittner, Christy	888-0765	SL250
Larson, Robert	888-9876	SL420
Lawson, Scott	222-1609	LA208
Muller, Christa	555-0980	SF190
Siebel, Helen	222-4467	LA678
Tripp, Trina	333-6709	SE267
Walker, Ellen	333-9079	SE567
Waters, Carol	888-9245	SL332
Yamamoto, Dina	888-2456	SL567

Figure 36. MultiJoin output

Pivot

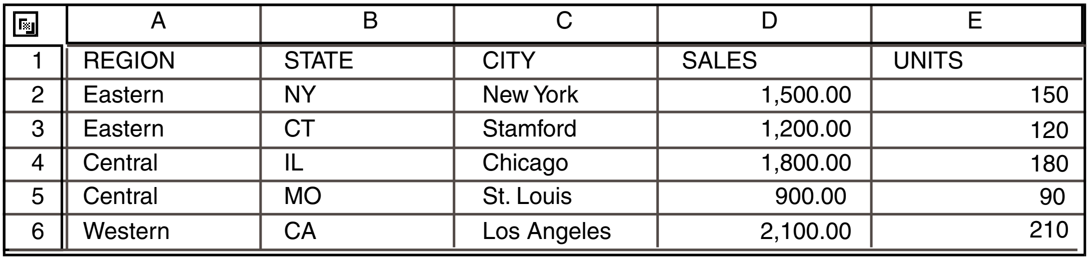
The Pivot transformer allows you to design a report by rearranging columns and data rows that you transfer from other icons.

The following terms are used in this section:

- A *dimension* is a column whose data values are used to create the column and row headings for the report.

- A *fact* is a column whose data values are used in the body of the report.

For example, the following table contains region, state, and city data as dimension columns, and sales and units data as fact columns.



	A	B	C	D	E
1	REGION	STATE	CITY	SALES	UNITS
2	Eastern	NY	New York	1,500.00	150
3	Eastern	CT	Stamford	1,200.00	120
4	Central	IL	Chicago	1,800.00	180
5	Central	MO	St. Louis	900.00	90
6	Western	CA	Los Angeles	2,100.00	210

The Pivot transformer builds the column and row headings from the dimensions data, and then inserts the facts at the appropriate column and row intersections.

Parameters

The Pivot transformer uses the parameters to essentially turn a table of data into a three-dimensional report that displays sections, columns, and rows. You can turn a table of data into a three-dimensional report by assigning the appropriate column letters in the parameter fields of the Transformer Controls window.

The first four parameters, `Section headings`, `Column headings`, `Row headings`, and `Facts`, work together to form the basic design for the report. Facts also become the data in the body of the report. Section headings are optional; however, you must enter at least one column letter for the other three headings. This is the minimum requirement to form a report from the data. Otherwise, an error message is returned, and you cannot form a report.

Section Heading

This parameter specifies the column headings you want to use as an information divider for your output. These headings appear at the beginning of each section of the output. A new section begins whenever a change occurs in the columns that you specify for this parameter.

You can enter one or several column letters to make the section headings as detailed as the data requires. However, if you use a fact column as a section heading, you can make only one entry.

You can arrange your entries in any order; they do not have to be in the same order as the input data. You can also leave this field blank.

Because a new section begins whenever a change occurs in the specified column, you must be sure that the rows of data that you want to appear in the same section are grouped together in the input data. Grouping data together might require that you use an intermediate Sort transformer to reorder the data before transferring your information into the Pivot transformer. Otherwise, the data of one group could appear within another group.

Column Headings

This parameter specifies the data to use as the column headings. You must enter at least one column letter in this field to create a report.

You can enter one or several column letters to make the output column headings as detailed as you need. You can also arrange your entries in any order; that is, they do not have to be in the same order as the input data.

Row Headings

This parameter specifies the columns whose data you want to use as the output row headings. You must enter at least one column letter in this field to create a report.

You can enter one or several column letters to make the output row headings as detailed as you require. You can also arrange your entries in any order; that is, they do not have to be in the same order as the input data.

Facts

This parameter specifies the data to include in the body of your output. You must enter at least one column letter in this field to create a report.

You can enter one or several column letters to display as many facts as your report requires. You can also arrange them in any order; that is, they do not have to be in the same order as the input data. Facts are also used for report headings. If you use a fact as a heading, you must also enter that same column letter in this field.

Because `Section headings` and `Row headings` column letters are entered in reverse order from the input data, the output will appear in that new order.

If you use a fact for a section heading, you cannot use any other columns as section headings. Also, you cannot display any other facts in the report.

Headings to Sort

This parameter specifies the sort order for the column headings in the output report. The order of row and column headings is determined in one of three ways:

- If no sort order is specified, row and column headings are arranged in the same order as they occur in the input data table.
- If you use the optional `(Row Order)` and `(Column Order)` regions in the `Display Data For` field to specify an arbitrary order, row and column headings are arranged in that order.

- If you specify an alphanumeric sort order in the `Headings to sort` field, the row and column headings occur in that order. This sort order overrides any order set in the optional input regions.

You can enter one or several column letters to sort the output column headings in ascending order, as required. If you want column headings to display in descending order, type `(d)` after the column letter in the field. You can also leave this field blank so that headings remain in the same order as the input data. Dates are sorted in chronological order.

You cannot sort section headings or facts by using the `Headings to sort` parameter. It causes an error when you run the program. To sort these parameters, enter the column letters in the field in the order you require (for example, D, B to display data in column D before the data in column B). For section headings, you can also specify an order in the input regions.

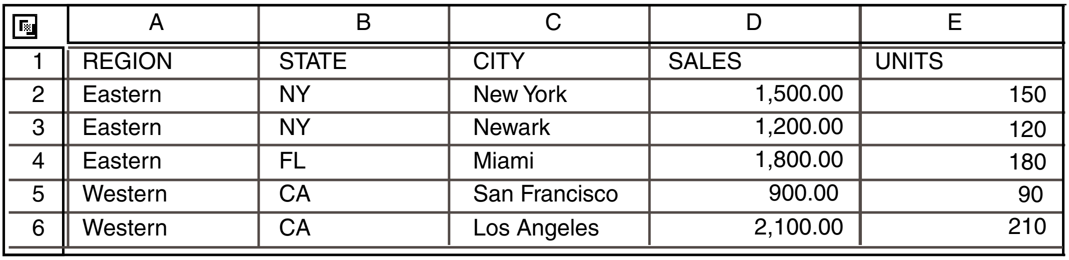
Replacement for Missing Values

This parameter allows you to enter a value to replace any missing values found in the data. Only N/As and blanks are interpreted as missing values by the Pivot transformer; for example, 0.00 is interpreted as a valid value. The transformer recognizes only N/As produced by a function of the table, not those you enter. You can leave this field blank.

To replace a missing value with a blank space, type `"  "` in this field.

Compute

This parameter specifies whether to calculate or sum values when facts appear in the same cell in the output report. For example, suppose you have the following table:



	A	B	C	D	E
1	REGION	STATE	CITY	SALES	UNITS
2	Eastern	NY	New York	1,500.00	150
3	Eastern	NY	Newark	1,200.00	120
4	Eastern	FL	Miami	1,800.00	180
5	Western	CA	San Francisco	900.00	90
6	Western	CA	Los Angeles	2,100.00	210

If you want to calculate the total sales and units for each state in each region you would enter the following values for the specified parameters:

Parameter	Value
Section heading	a
Column headings	b
Row headings	d, e

Parameter	Value
Facts	d, e
Compute	sum(d, e)
Blank rows between sections	1

The following output would result:

```
Eastern
                  NY          FL
SALES (Sum)       2700.00     1800.00
UNITS (Sum)       270.00      180.00

Western
                  CA
SALES (Sum)       3000.00
UNITS (Sum)       300.00
```

Headings

This parameter specifies which column headings to use as titles for the section, column, and row headings of the output data. The default value is `section`.

Total Rows per Section

This parameter specifies the number of rows to display for each section in the report. This is beneficial when you want to control the number of sections that appear on each page of the report.

When you print pivoted data, it is useful to know the maximum number of lines allowed on each page of the output icon (for example, Text or Spreadsheet).

The entry in the `Total rows per section` field includes the number of rows specified in the `Blank rows between sections` field. If the section contains less than the specified number of rows, the transformer inserts the remaining number of blank rows.

For example, if you specify `20` in the `Total rows per section` field and `2` in the `Blank rows between sections` field, the output will contain 18 rows of data and two blank rows. If the section contains less than 18 rows of data, the remaining rows are left blank so that there are always 20 rows in each section.

If you leave this field blank, the transformer inserts no additional rows. That is, each section in the output displays only those rows that contain data.

Blank Rows Between Sections

This parameter specifies the number of rows to insert between each section in the report. If you leave this field blank, the transformer does not insert additional rows between sections.

The number you enter in this field is incorporated in the total number of rows per section, if one is specified. If no value is specified in the `Total rows per section` field, then the number you enter in this field determines the number of rows inserted between each section.

Fill in Headings

This parameter allows you to fill in all column and row headings in the output report.

Insert Blank Row Before Change in Column

This parameter allows you to visually separate data groups in your output report by inserting blank rows. You can enter only one column letter. You can also leave this field blank, in which case no separating rows are inserted. The column specified in this field must also be used as a row heading.

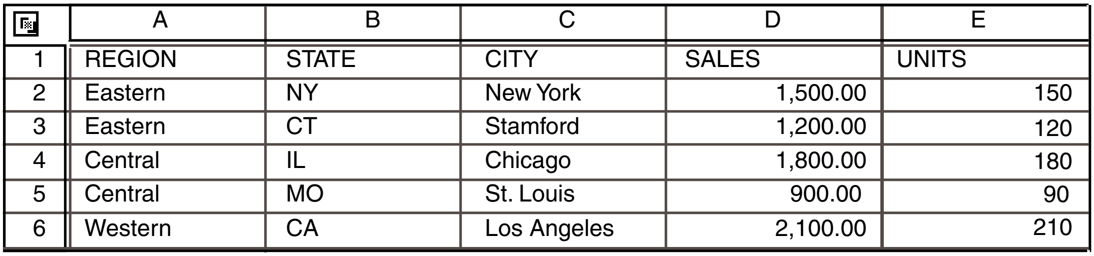
Spaces to Indent Row Headings

This parameter specifies the number of spaces for indenting row headings in the output report. There are three format styles, depending on the number that you enter in this field:

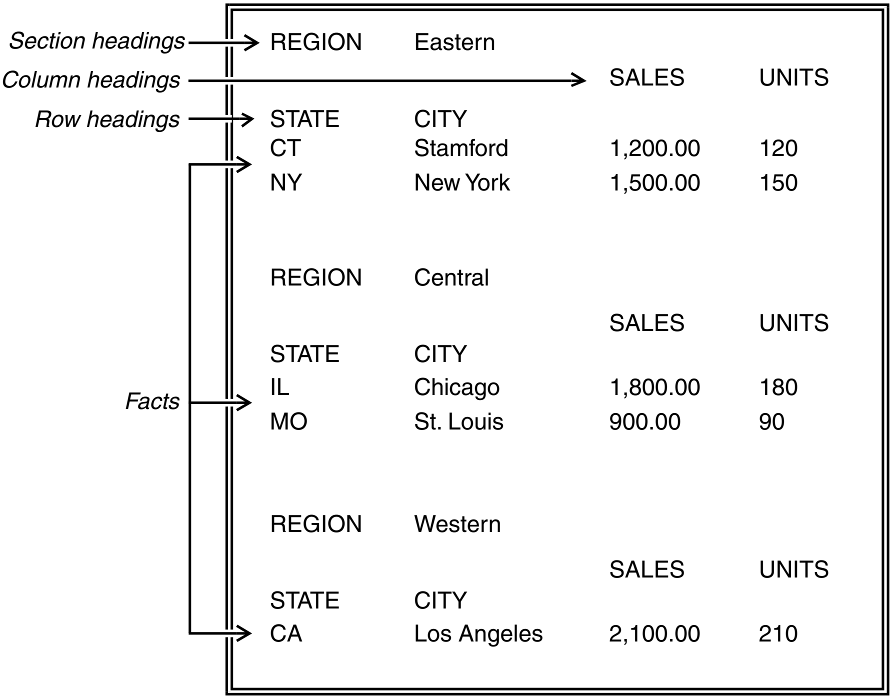
- If you leave this field blank, the row heading format causes each row level to begin in a separate column: level 1 begins in the first column, level 2 begins in the second column, and level 3 begins in the third column.
- If you type `0`, the row headings appear with each row level in the first column.
- If you enter a number other than 0, the row heading format causes each row level to appear in the first column, but level 2 and level 3 are indented two spaces each (the number entered in this field).

Pivot

The following example shows how some of the parameters determine the design for a report.



	A	B	C	D	E
1	REGION	STATE	CITY	SALES	UNITS
2	Eastern	NY	New York	1,500.00	150
3	Eastern	CT	Stamford	1,200.00	120
4	Central	IL	Chicago	1,800.00	180
5	Central	MO	St. Louis	900.00	90
6	Western	CA	Los Angeles	2,100.00	210



```
REGION     Eastern
                          SALES      UNITS
STATE      CITY
CT         Stamford       1,200.00   120
NY         New York       1,500.00   150


REGION     Central
                          SALES      UNITS
STATE      CITY
IL         Chicago        1,800.00   180
MO         St. Louis      900.00     90


REGION     Western
                          SALES      UNITS
STATE      CITY
CA         Los Angeles    2,100.00   210
```

The output data results are derived from the following parameters:

Results	**Parameters**
Section headings	a
Column headings	d, e
Row headings	b, c
Facts	d, e
Heading titles	All

Results	Parameters
Blank rows between sections	1

Region Controls

The `Display Data For` field in the Pivot transformer contains the following choices:

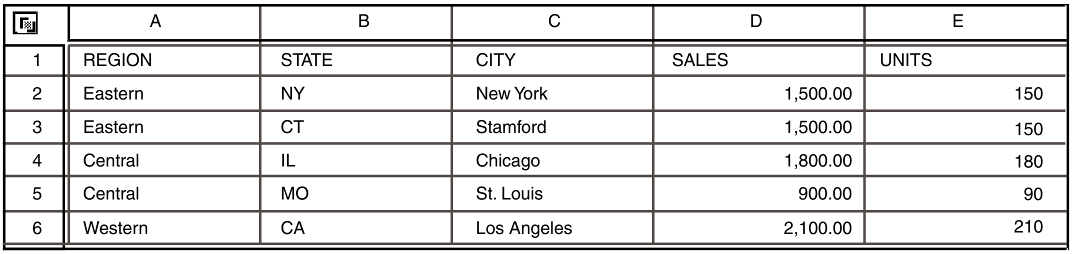
(Row Order) is an optional input region that allows you to order headings in other than alphabetic or numeric order. The (Row Order) input region overrides the order of headings appearing in the Input Data display area.

In most cases, you will not need to use this input region.

(Column Order) is an optional input region that allows you to order column headings in other than alphabetic or numeric order. When you run the Pivot transformer, the (Column Order) input region overrides the order of the headings appearing in the Input Data display area.

In most cases, you will not need to use this input region.

The Input Data display area retains the data that you want to pivot. To prevent overlapping data in the pivoted report, the rows in this region should be unique for all dimension columns. If you have duplicate data, as shown in rows 2 and 3 in the following example, an error message appears. If the duplicate row does not appear in an adjacent row, however, the transformer processes the data successfully.



	A	B	C	D	E
1	REGION	STATE	CITY	SALES	UNITS
2	Eastern	NY	New York	1,500.00	150
3	Eastern	CT	Stamford	1,500.00	150
4	Central	IL	Chicago	1,800.00	180
5	Central	MO	St. Louis	900.00	90
6	Western	CA	Los Angeles	2,100.00	210

The Output Data display area is the area where pivoted data appears after you run the transformer.

Pivot Regions

The first two input regions, (Row Order) and (Column Order), are optional and define a specific heading order for your report. Using these regions allows you to create reports that:

- Display your own company's data at the top of the report
- Create column and row headings for missing data values

- Sort section headings

Both (Row Order) and (Column Order) input regions are entered as columns of data. You do not specify the column or row heading in the input region.

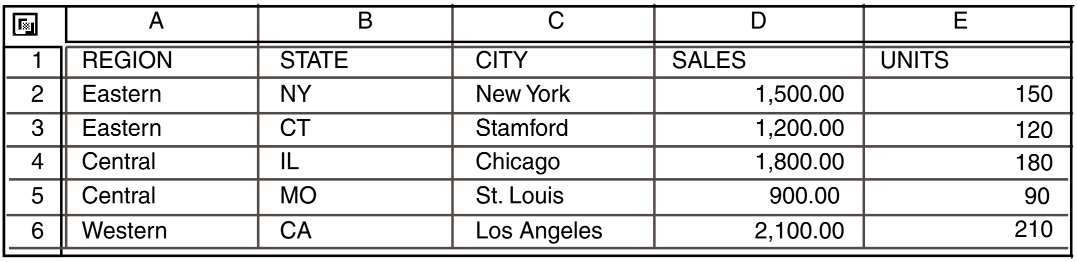
Guidelines for Using Input Regions

Use the following guidelines when defining input regions:

- You cannot use facts in the input regions, or the transformer will fail.
- The number of column letters entered in the input region must equal the number of column letters entered for the row headings or column headings fields. For example, if you have three row headings in your output report (that is, you entered three column letters in the `Row headings` field), the (Row Order) input region must contain at least one row and three columns of data.
- Any column letters that you specify in the `Headings to sort` field cannot be the same as the columns you are sorting for an input region. Otherwise, the sort order specified in that field overrides the input region order.

Example

Suppose that you have a table containing the following data:



	A	B	C	D	E
1	REGION	STATE	CITY	SALES	UNITS
2	Eastern	NY	New York	1,500.00	150
3	Eastern	CT	Stamford	1,200.00	120
4	Central	IL	Chicago	1,800.00	180
5	Central	MO	St. Louis	900.00	90
6	Western	CA	Los Angeles	2,100.00	210

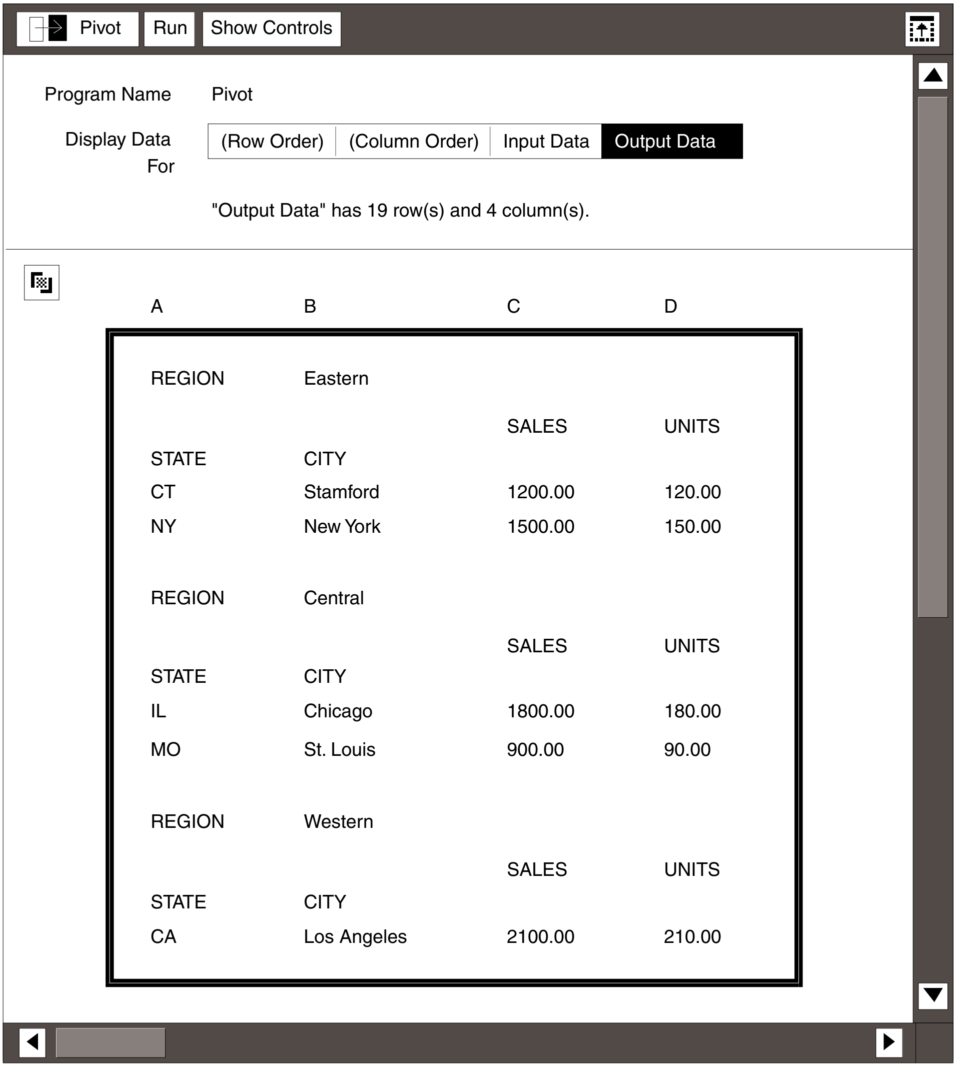
You can format the columns and rows that repeat the same information and combine columns and rows to make a more concise report.

To create the sample Pivot transformer:

1. Click on the `Show Controls` button in the transformer window header.
2. Type `a` in the `Section headings` field to use REGION as a section heading.
3. Type `d, e` in the `Column headings` field to use SALES and UNITS as column headings.
4. Type `b, c` in the `Row headings` field to use STATE and CITY as row headings.
5. Type `d, e` in the `Facts` field to use the sales and units data.

6. Type `b` in the `Headings to sort` field to sort the states in alphabetic order.
7. Type `N/A` in the `Replacement for missing values` field to replace any missing values with N/A.
8. Type `all` in the `Heading Titles` field to repeat each group of heading titles for each region.
9. Type `1` in the `Blank rows between sections` field to insert a blank row between each new region.
10. Close the Transformer Controls window and the Pivot transformer window.
11. Set the arrow options so that the data is sent to the `Input Data` display area.
12. Click outside the Capsule window to deselect the Pivot transformer.
13. Click on the `Run` button in the Capsule window header.

The output appears as shown in Figure 37 on page 92.



Program Name Pivot

Display Data For: (Row Order) | (Column Order) | Input Data | **Output Data**

"Output Data" has 19 row(s) and 4 column(s).

A	B	C	D
REGION	Eastern		
		SALES	UNITS
STATE	CITY		
CT	Stamford	1200.00	120.00
NY	New York	1500.00	150.00
REGION	Central		
		SALES	UNITS
STATE	CITY		
IL	Chicago	1800.00	180.00
MO	St. Louis	900.00	90.00
REGION	Western		
		SALES	UNITS
STATE	CITY		
CA	Los Angeles	2100.00	210.00

Figure 37. Pivot example output

Select

The Select transformer enables you to rearrange a table of data by choosing the data columns and specifying the order they appear in the output.

Parameters

The `Columns to keep` parameter specifies the data columns you want to retain, and the order in which you want them to appear in the output. Column A is the default value. You can also use numbers in this field. However, you cannot combine characters and numbers.

Region Controls

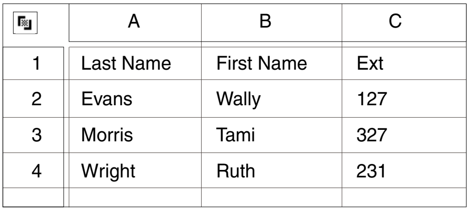
The `Display Data For` field in the Select transformer contains the following choices:

- Input Data
- Output Data

Input Data is the area where you transfer the source data. Output Data is the area where data appears after running the transformer.

Example

Suppose you have a capsule application that enables you to use the Select transformer to display and arrange certain columns from the following table:

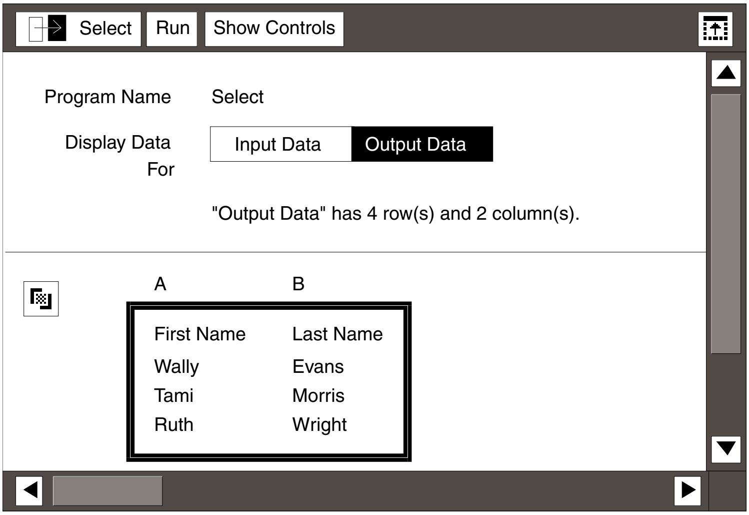


	A	B	C
1	Last Name	First Name	Ext
2	Evans	Wally	127
3	Morris	Tami	327
4	Wright	Ruth	231

To retain columns A and B and reverse their order in the output:

1. Click on the `Show Controls` button in the Select transformer window header.
2. Type `b, a` in the `Columns to keep` field to keep and reorder columns A and B.
3. Close the Transformer Controls window and the Select transformer window.
4. Click outside the Capsule window to deselect the Select transformer.
5. Click on the `Run` button in the Capsule window header.

The output appears as shown in Figure 38.



Program Name Select

Display Data For: Input Data | Output Data

"Output Data" has 4 row(s) and 2 column(s).

A	B
First Name	Last Name
Wally	Evans
Tami	Morris
Ruth	Wright

Figure 38. Select example output

Sort

The Sort transformer allows you to rearrange data columns in a table in chronological or alphanumeric order, using one of six national sort sequences. The transformer also provides formatting functions and enables you to choose the number of rows that appear in the output.

Parameters

The Sort transformer has the following parameters:

Sort Columns

This parameter specifies the columns to sort and the order in which to sort them. By default, columns sort in ascending order. To sort columns in descending order, type `(d)` after the column letter; for example, `a(d)`. You can enter column letters in uppercase or lowercase.

You can enter single or multiple sorting levels; that is, if there are duplicate values in the first column, you can sort on the values in the second column, and so on.

Enter the column letters in the order you want them sorted. For example, if you have a spreadsheet that contains employees' last and first names in columns A and B, respectively, you can type `a,b` in the `Sort columns` field. This ensures that last names that are duplicated in column A are sorted using the first names in column B.

If no columns are specified in this field, the sort order in the output appears identical to the sort order in the source icon.

If you specify a column that contains no data, the transformer completes its run successfully. The specified column is sorted; although, because there is no data, the results are not apparent.

Value of Blanks, N/As, and Errors

This parameter specifies the sort order of null values. Type `Smallest` to assign blanks, N/As, and errors the lowest value in the columns. Type `Largest` to assign them the highest value. If this field is left blank, the value defaults to `Smallest` for an ascending sort and `Largest` for a descending sort.

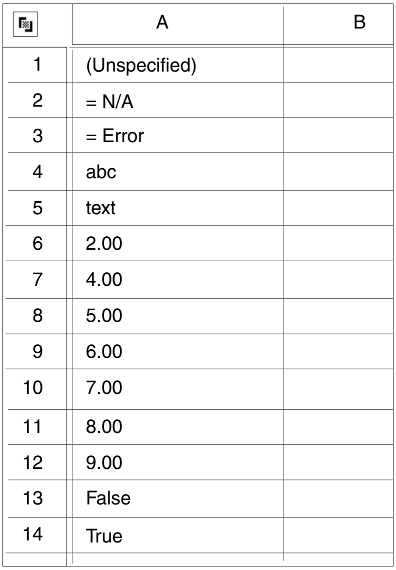
The transformer sorts only N/As produced by a function of the table, not those you enter. When you sort a column that contains special data types, the values in the `Values of blanks, N/As, errors` field and other values are sorted as described in Table 8.

Table 8. Sort order of Values of blanks, N/As, errors and other values

Values	Description
N/As	If `Largest` is entered, N/As appear before blanks (unspecified) and after errors. If `Smallest` is entered, N/As appear before errors and after blanks (unspecified).
Errors	If `Largest` is entered, errors appear after numbers, Boolean values, and N/As. If `Smallest` is entered, errors appear after N/As and before text.
False	Appears after numbers, but before True.
True	Appears after numbers, but after False.
Blanks (Unspecified)	If `Largest` is entered, blanks appear after any data. If `Smallest` is entered, blanks appear before any data.
Text	Appears before numbers.

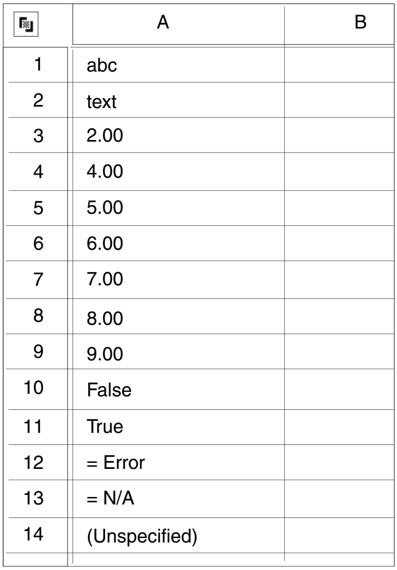
The following spreadsheets are examples of the values in the `Values of blanks, N/As, and errors` field sorted by smallest and largest values.

Smallest



	A	B
1	(Unspecified)	
2	= N/A	
3	= Error	
4	abc	
5	text	
6	2.00	
7	4.00	
8	5.00	
9	6.00	
10	7.00	
11	8.00	
12	9.00	
13	False	
14	True	

Largest



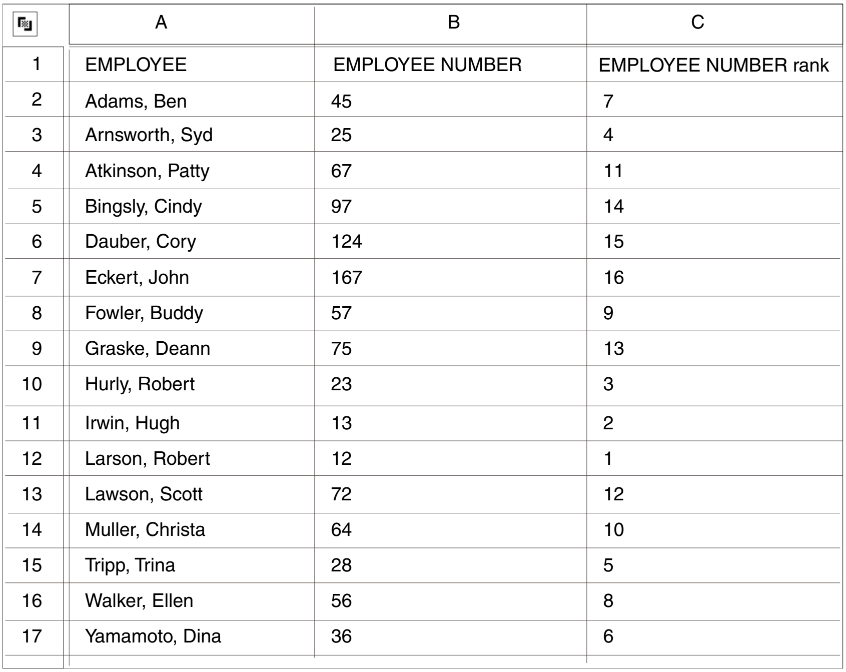
	A	B
1	abc	
2	text	
3	2.00	
4	4.00	
5	5.00	
6	6.00	
7	7.00	
8	8.00	
9	9.00	
10	False	
11	True	
12	= Error	
13	= N/A	
14	(Unspecified)	

Number of Heading Rows

This parameter allows you to specify the number of rows that are used as column headings for the data and should not be included in the sorting process. The default value is 0 rows; that is, the data contains no column heading rows.

Rank Columns

Ranking data allows you to easily determine the relative order of values in a data column. The rank value is a sequential number; that is, 1, 2, 3, and so on. When you specify a particular column letter in this field, a new column is inserted adjacent to the specified column, as shown in the following illustration.



	A	B	C
1	EMPLOYEE	EMPLOYEE NUMBER	EMPLOYEE NUMBER rank
2	Adams, Ben	45	7
3	Arnsworth, Syd	25	4
4	Atkinson, Patty	67	11
5	Bingsly, Cindy	97	14
6	Dauber, Cory	124	15
7	Eckert, John	167	16
8	Fowler, Buddy	57	9
9	Graske, Deann	75	13
10	Hurly, Robert	23	3
11	Irwin, Hugh	13	2
12	Larson, Robert	12	1
13	Lawson, Scott	72	12
14	Muller, Christa	64	10
15	Tripp, Trina	28	5
16	Walker, Ellen	56	8
17	Yamamoto, Dina	36	6

In the case where two or more data elements have the same value, the rank value will be the same; any subsequent rank values increase by the number of repetition; for example, 1, 2, 3, 3, 5, and so on.

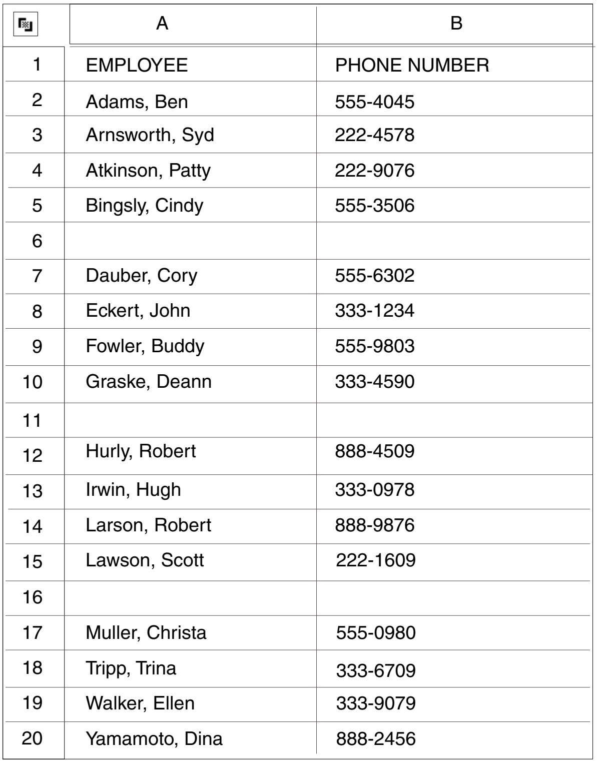
If you want to rank the columns in descending order, include `(d)` after the column letter. If no columns are specified, the Sort transformer will not rank any of the data. Specify a value in the `Number of heading rows` field to remove heading rows from the ranking process if your table contains headings.

Number of Partitions

This parameter enables you to divide your output data into groups with an equal number of rows (partitions). This is useful when you want to display the data as percentiles, quartiles, or other groupings.

The Sort transformer divides the data into the specified number of groups and then inserts a blank row after each group of rows. Heading rows are not included in the row count. If only one partition is specified, the rows will not be divided. There must be at least one group of output data, so if you type 0 or a negative number for this field, the transformer automatically converts it to one.

For example, to partition a table into four groups as shown in the following illustration, type `4` in this field. You must also specify the number of heading rows so they are not included when the data is divided.



	A	B
1	EMPLOYEE	PHONE NUMBER
2	Adams, Ben	555-4045
3	Arnsworth, Syd	222-4578
4	Atkinson, Patty	222-9076
5	Bingsly, Cindy	555-3506
6		
7	Dauber, Cory	555-6302
8	Eckert, John	333-1234
9	Fowler, Buddy	555-9803
10	Graske, Deann	333-4590
11		
12	Hurly, Robert	888-4509
13	Irwin, Hugh	333-0978
14	Larson, Robert	888-9876
15	Lawson, Scott	222-1609
16		
17	Muller, Christa	555-0980
18	Tripp, Trina	333-6709
19	Walker, Ellen	333-9079
20	Yamamoto, Dina	888-2456

Number of Rows to Output

This parameter specifies the number of rows of data to display in the output. For example, if a table contains a large quantity of entries, and you want to display only the top 10 values, type `10` in this field.

Heading rows are not included in the row count, nor are blank rows that are created by partitioning the data; however, you must also specify the number of heading rows so that any heading rows are not interpreted as data when the output rows are selected. The Sort transformer places the specified number of data rows in the output display area, beginning with the first data row. To include all data rows in the output, leave this field blank.

Language

This parameter specifies a national sort sequence. The languages supported include U.S. English, U.K. English, French, German, Italian, and Dutch. U.S. English is the default language. If you enter an unsupported language in this field, and then click on the `Run` button, an error message appears.

Region Controls

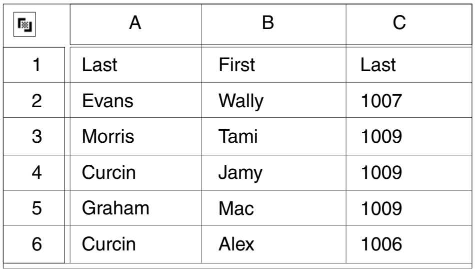
The `Display Data For` field in the Sort window contains the following choices:

- Input Data
- Output Data

Input Data identifies the region where you transfer source data. Output Data identifies the region where the data appears after it is processed.

Example

Suppose you want to sort the following information in descending alphabetic order.

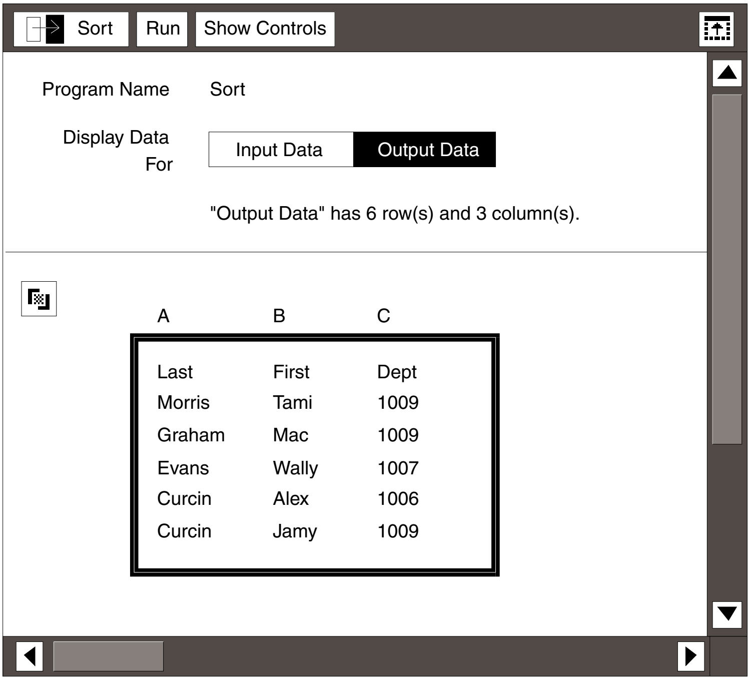


	A	B	C
1	Last	First	Last
2	Evans	Wally	1007
3	Morris	Tami	1009
4	Curcin	Jamy	1009
5	Graham	Mac	1009
6	Curcin	Alex	1006

To run the sample Sort transformer:

1. Click on the `Show Controls` button in the Sort transformer window header.
2. Type `a(d), b` into the `Sort columns` field to sort column A in descending order, but column B in ascending order.
3. Type `Largest` into the `Value of blanks, N/As, and errors` field to sort blanks to the top of the column (it does not affect this data).
4. Type `1` into the `Number of heading rows` field to eliminate row 1 from being considered in the sorting process.
5. Type a valid language in the `Language` field.
6. Close the Transformer Controls window and the Sort transformer window.
7. Click outside the Capsule window to deselect the Sort transformer.
8. Click on the `Run` button in the Capsule window header.

The output appears as shown in the following example.



Program Name Sort

Display Data For Input Data [✓ Output Data]

"Output Data" has 6 row(s) and 3 column(s).

A	B	C
Last	First	Dept
Morris	Tami	1009
Graham	Mac	1009
Evans	Wally	1007
Curcin	Alex	1006
Curcin	Jamy	1009

Figure 39. Sort output

Chapter 3. Advanced Transformers

With advanced transformers, you can create sophisticated Meta5 applications that enhance the data-manipulation capabilities of the statistical transformers, data access tools, and other application programs. These transformers provide computational and formatting enhancements that include splitting, joining, transposing, reformatting, cleaning rows, and reporting errors.

Each advanced transformer is designed to perform a specific function. For example, you can use the Row Clean transformer to selectively remove rows of data provided by a spreadsheet, query, or other row-oriented data source, or use the Write SQL transformer to translate row- and column-formatted data into SQL statements that can load the data into a database within a capsule application.

To locate the advanced transformers:

1. Open the New Icons file drawer.
2. Open the Transformer Icons file drawer.
3. Open the Advanced Transformers folder.

If you cannot find a specific transformer, see your system administrator.

Table 9 lists the advanced transformers, describes their functions, and gives the page number where each transformer is described.

Table 9. Advanced Transformers

Transformer	Function	See
Append Message	Sends text from a capsule application to the desktop message area.	“Append Message” on page 103
Concatenate	Concatenates input by row from as many as five input regions into one output region.	“Concatenate” on page 104
Conditional Clean	Selects data from input based on whether the data meets or fails user-defined criteria.	“Conditional Clean” on page 106
Data Format	Formats input data into either SQL statements for loading the data into a database as text, or into text that can be sent to environments other than Meta5.	“Data Format” on page 112
Date Format	Reformats date information into any of 10 supported date formats.	“Date Format” on page 121

Advanced Transformers

Table 9. Advanced Transformers

Transformer	Function	See
Error Log 1 and Error Log 2	Can send text messages from a capsule application to an output region, to the desktop's messages area, or to the desktop's Important Message window.	“Error Log 1 and Error Log 2” on page 126
Function	Performs calculations on data sets larger than those supported by the Spreadsheet tool by applying data-transformation expressions (mathematical, spreadsheet functions, data formatting, or date conversions) to each row of data it receives.	“Function” on page 132
Header	Reads text from a Query tool icon, SQL Entry icon, or spreadsheet and formats it into column headings suitable for a report.	“Header” on page 137
Message	Sends text messages from a capsule application to the desktop's message area or to the Important Message window.	“Message” on page 141
Multiple Select	Sends specified data columns from one input region to as many as 10 output regions.	“Multiple Select” on page 143
Multiple Split	Sends specified data rows from one input region to as many as 10 output regions.	“Multiple Split” on page 147
Page Break	Formats spreadsheet data into a paginated document that is ready for printing.	“Page Break” on page 150
Period Table	Constructs date and period tables for use in databases or as input to the Seasonality and Forecast transformers (in the Regression and Time Series group).	“Period Table” on page 155
Post Message	Sends text from a capsule application to the desktop's messages area.	“Post Message” on page 163
Random Number	Generates random number tables.	“Random Number” on page 164
Replace	Selectively substitutes values in data from a Query or Spreadsheet icon.	“Replace” on page 166
Row Clean	Selectively removes rows of data provided by a spreadsheet, query, or other row-oriented data source, based on flexible cleaning rules.	“Row Clean” on page 172
Row Select	Selects data rows based on their location in a data set provided by a Spreadsheet, Query, or SQL Entry icon.	“Row Select” on page 177
Split Header	Sends specified data rows of an input region to one of two output regions.	“Split Header” on page 183

Table 9. Advanced Transformers

Transformer	Function	See
Substitute	Finds and replaces data from a spreadsheet or other row-oriented data source.	“Substitute” on page 186
Subtotal	Creates subtotals and grand totals on columns of data	“Subtotal” on page 189
Switch and Text Switch	Reads data from a Spreadsheet, Query, or SQL Entry icon and uses if-then-else logic to select one of two input regions to receive the output.	“Switch and Text Switch” on page 194
Text To Spreadsheet	Converts information in a Text icon into Spreadsheet format.	“Text to Spreadsheet” on page 196
Transpose	Interchanges rows and columns in a table of data.	“Transpose” on page 201
Word Count	Computes checksums and counts characters, words, and paragraphs in a Text document.	“Word Count” on page 202
Write SQL	Translates row- and column-oriented data into SQL statements that can load the data into a database (within a capsule application).	“Write SQL” on page 204

For general information on using transformers, see “Chapter 1. Getting Started with Transformers,” on page 1.

Append Message

Use the Append Message transformer to automatically construct messages within a capsule application and send them to the Meta5 desktop.

The Append Message parameters are compatible with @-variables, allowing the content of messages to vary accordingly with the outcome of a capsule application run.

An Append Message transformer combines several text string messages and sends them to the message bar area. If all message parameters are empty, the transformer runs without taking any action.

Parameters

The Append Message transformer has 10 parameters, `Parameter #1` through `Parameter #10`. These parameters are input areas for the message. The input parameter values can be text strings or numbers; other data types are not supported. Numbers are shown with two decimal places of precision. Each value can contain up to 99 characters. Only one value is allowed per parameter.

Because commas and semicolons are interpreted as value separators, values containing commas or semicolons should be placed in double quotation marks.

Region Controls

The Append Message transformer has no input or output regions.

Concatenate

The Concatenate transformer joins information from as many as five Query or Spreadsheet icons into one output region. You can use the transformer to join data that does not share a unique set of keys. In addition, the Concatenate transformer can add line numbers to and remove blank lines from a data set.

Most advanced transformers can act as a dynamic buffer. Generally, when data from an SQL Entry icon or Query tool is fed into two or more arrows, a spreadsheet can act as a buffer. However, very large sets of data can exceed the amount of workstation memory available to a spreadsheet. In these cases, you can use an advanced transformer as a buffer. For example, more data can be stored in the Concatenate transformer because the number of rows that it can store is limited only by the amount of disk space available on the file server.

The Concatenate transformer is also useful in assembling and debugging applications. For example, repeatedly testing a capsule and running a complex query is time consuming, especially if the same data is returned on each test. In such cases, you can use dynamic buffering to store the results of a lengthy query. Then, when the capsule application is being tested, you need only to wait for the data transfer. The time saved is especially significant when you use dynamic buffering instead of a query to a database machine located on a remote network.

The Concatenate transformer is also useful in joining data gathered using SQL SELECT statements from two or more database tables.

Parameters

Number of header rows in data for each region (0, 0, 0, 0, 0; 1, 2, 3, 4, 5;)
This parameter specifies the number of rows of data that will be skipped and not copied to the output at the start of each input region. When this parameter is used, it requires five numbers, separated by commas. The default value for each input region is 0.

Copy first row of first header to output? (y; n)
This parameter specifies whether the output data has a header. If the value no is entered, the output will not have a header. If the value is yes, the first header row of the first valid input region is copied to the output as a header. The default value is no.

Add row numbers? (y; n) starting row number? (1; 5;)

This parameter specifies whether to add a sequence column containing row numbers. If yes is specified, column A contains the row numbers. The default value is no. The second parameter, *starting row number?*, specifies the first row number to be used, if row numbers are being added. The default value is 1. The largest row number that can display is 2147483647; the smallest is –2147483647.

Remove empty rows? (y; n)

This parameter specifies whether empty rows should be copied to the output. An empty row is defined as a row that has no columns, or all columns are of the data type *Unspecified*. The default value is no.

Region Controls

The Concatenate transformer has input and output regions.

Input Region Names

The Concatenate transformer has five input regions called Data 1 through Data 5. In a capsule application, you must use the Data 5 region; all other input regions are optional.

Output Region Names

The Concatenate transformer has one output region, Results, that contains the data from all input regions joined sequentially; it has no size limit other than the amount of disk space available on the file server. If your output is connected to a spreadsheet, make sure there is adequate space in the spreadsheet for your data.

Example

Assume the following data is included in Data 1:

Market	Year	Volume
NY	1998	234
NY	1999	876
NY	2000	934
NY	2001	745

Assume that the following data is included in Data 5:

Market	Year	Volume

LA	1998	894
LA	1999	275
LA	2000	587
LA	2001	274

When the Concatenate transformer joins the data in Data 1 and Data 5, the following data file is displayed in the Results output region:

Market	Year	Volume
NY	1998	234
NY	1999	876
NY	2000	934
NY	2001	745
LA	1998	894
LA	1999	275
LA	2000	587
LA	2001	274

The value for `Number of header rows in data` for each output region was set to 1,1,1,1,1 to remove the single line header from both input regions, and the value for `Copy first row of first header to output` was set to `yes` to add the header on the output region. The remaining parameters are set at their default values.

Conditional Clean

The Conditional Clean transformer selectively chooses rows of data from a spreadsheet, based on conditions specified by the user. Conditional Clean can automate the row selection process.

The conditional data-cleaning process is determined by a set of selection rules, presented to the transformer through Input 1 (Rules). Each selection rule is specified for one column. You can choose whether the search should ignore white space and the case of text in string comparisons, as well as the degree of precision used in numeric comparisons. When the Conditional Clean transformer runs, it checks each row of data in Input 2 (Data) and copies it to Output 1 (Results) if the selection rules are met.

There are three reasons for eliminating data by conditional rules.

Remove rows based on some criterion not supported by the Clean transformer

The Clean transformer removes all rows with zero values or N/A values. In contrast, the Conditional Clean transformer can remove rows with N/A values but allow rows with zero values. For example, the Conditional Clean transformer can discard data for survey respondents who refused to answer a question, as indicated with N/A, and leave respondents who answered with a zero in the data set.

Exception reporting

The criteria defining possible exceptions in a data set can be written as conditional selection rules. These rules can be applied to locate values lying outside the acceptable bounds.

For example, if a regional manager wants to know which stores are doing poorly, the Conditional Clean transformer can be used to search for locations with low sales volumes or unacceptable profit levels. A similar set of rules could be developed to determine which stores are doing exceptionally well.

Data filtering

If a particular data set has a few cases that deviate from the norm, the unusual cases can be removed using the Conditional Clean transformer. You can specify a pair of rules that set the minimum and maximum acceptable levels for a particular value. All data lying outside these limits can be discarded.

For example, if a marketer wants to forecast sales for the coming year, she might want to clean the historical sales data to remove periods that have very unusual sales levels that might skew the resulting forecasts. To do this, the marketer can instruct the Conditional Clean transformer to remove rows of data that have sales levels that are greater than a cutoff value or below a second cutoff value. The upper and lower cutoffs could correspond to points that are two standard deviations above and below the average sales level, respectively.

Parameters

Number of header rows in data (0; 1;)

This parameter specifies the number of rows of data at the top of Input 2 to copy to Output 1 regardless of whether they satisfy the rules. For example, if *2* is specified, the first two rows of data in Input 2 are copied to Output 1. Any whole number can be specified; the default value is 0.

Accept a row if the conditions of (any, every) rule specified in 'Input 1' is met (; a; e)

This parameter specifies whether every rule or only one rule must be satisfied for a row to be selected. *a* or *any* indicates that only one rule needs to be satisfied, and *e* or *every* indicates that all of the rules must be satisfied. The default value is *every*.

Ignore case and white space during text comparisons? (; y; n; yes; no)

This parameter specifies whether the value being compared matches a particular data value if the only differences are in spacing or the case of characters. Specifying *y* or *yes* indicates that case and white space is ignored, *n* or *no* indicates that case and white space are significant. The default value is yes.

Numeric tolerance to allow during numerical comparisons (; 0.1; 0.01; 0.001;)

This parameter specifies the amount by which two number values can differ before the transformer decides they are different. This parameter is useful when the precision of a number is different from its display value. Any real number value including a decimal point can be entered; the default value is 0.0.

The tolerance value is used only when two real numbers or a real number and an integer are compared. No tolerance is allowed when two integers are compared. Do not assume that the presence or absence of decimal places indicates whether a number has an integer or real format. Generally, a spreadsheet number is a real number.

Show data that is (accepted, rejected) by the rules specified in 'Input 1' (; a; r)

This parameter specifies whether the data in Output 1 meets the specified rules. *A* or *accepted* indicates that Output 1 contains data that meets the rules, and *r* or *rejected* indicates that data meeting the rules is discarded. The default value is *accepted*.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Rules (Input 1)
- Data (Input 2)
- Results (Output 1)

When you select one of these choices, the data associated with that choice is shown in the display area of the transformer.

Input Region Names

The Conditional Clean transformer has two input regions: Input 1 (Rules) and Input 2 (Data).

Input 1 contains the selection rules. It is formatted as a series of rows and columns that can contain any type of data, limited in size by the amount of workstation memory available.

Input 2 contains data read from a spreadsheet, Query tool, SQL Entry icon, or any other row- and column-formatted data. Input 2 can have any number of header rows and is not limited in size.

Output Region Names

The Conditional Clean transformer has one output region (Output 1) called Results. Output 1 contains data selected from Input 2 based on the conditional selection rules in Input 1; it has no size limit.

Example

A marketing agency is planning to test an outdoor advertising campaign in one state. The agency sets certain criteria for choosing the test state. First, the state must have a population of 10 million or more. Second, the state must have a land area that is less than the national average of 50,000 square miles.

To configure the Conditional Clean transformer to search for rows meeting the constraints, two selection rules are needed. Table 10 contains the rules and is connected to Input 1 of the Conditional Clean transformer.

Table 10. Spreadsheet connected to Input 1 of the Conditional Clean transformer

Column:	Operator:	Compare To:
#		
# Choose Population Of 10 Million Or More		
#		
B	> =	10,000,000
#		
# Choose Area To Be Less Than 50,000		
#		
C	<	50,000

Because a row must satisfy both of the rules, the Conditional Clean transformer parameters are set as follows:

Number of header rows in data
1

Accept a row if the conditions of (any, every) rule specified in 'Input 1' is met
e

Ignore case and white space during text comparisons?
yes

Numeric tolerance to allow during numerical comparisons
0.01

Show data that is (accepted, rejected) by the rules specified in 'Input 1'
a

Table 11 shows some of the information stored in a spreadsheet connected to Input 2 of the Conditional Clean transformer.

Table 11. Spreadsheet connected to Input 2 of the Conditional Clean transformer

State	Population	Area
Alabama	3,894,000	51,705
Alaska	402,000	591,004
Arizona	2,718,000	114,000
Arkansas	2,286,000	53,187
California	23,668,000	158,706
Colorado	2,890,000	104,091
New York	17,558,000	49,108
Ohio	10,798,000	41,330
Pennsylvania	11,864,000	45,308
Wisconsin	4,706,000	56,153
Wyoming	470,000	97,809

When the Conditional Clean transformer is run, the information shown in Table 12 on page 110 is sent to the spreadsheet connected to Output 1. Three states met the requirements and are candidates for being the test market.

This example required every rule in Input 1 to be satisfied before a row was accepted. If *any* was specified instead of *every*, the transformer creates a list of states satisfying one of the requirements, but not necessarily both. States such as Delaware would pass the size requirement, but would not meet the population requirement. Also, California would be included based on population, but its land area exceeds the size requirement.

Table 12. Spreadsheet connected to Output 1 of the Conditional Clean transformer

State	Population	Area
New York	17,558,000	49,108
Ohio	10,798,000	41,330
Pennsylvania	11,864,000	45,308

Specifying Selection Rules

You can specify a selection rule for each column. You can choose the case of text in string comparisons, the degree of precision used in numeric comparisons, and whether the search should ignore white space.

Conditional selection rules are typically written in a spreadsheet, but can originate in any row- and column-formatted source. Each row contains one rule. Column A contains the name of the column to check, column B contains the operator to use during the comparison, and column C contains the comparison value. The comparison value defines both the data type and value. For example, consider the following rule:

Column A	Column B	Column C
a	<	3.14

This rule specifies that every cell in column A is searched for numeric values that are less than the number 3.14. Whenever this condition exists, the row is copied to Output 1.

To make the selection rules less complex, the transformer simplifies the evaluation of spreadsheet data in two ways. First, the distinction between integer numbers and real numbers is ignored. Thus, if the comparison value is the real number -1.00 and the spreadsheet cell being examined contains the integer -1, the two numbers are considered equal.

The second simplification is that only the day values of dates are considered, not the display resolution. As a result, the date 1Q99 and the date January 1999 are considered equal, because both use January 1, 1999 as their day value.

The Conditional Clean transformer assumes that the first row of selection rules is a header and always ignores it. The recommended header is:

Column:	*Operator:*	*Compare To:*

Six operators are available that can be expressed in a variety of formats, as shown in Table 13.

Table 13. Conditional Clean operators

Operator	FORTRAN	Text	C
equal	.EQ.	EQ, eq	=, = =
not equal	.NE.	NE, ne	!=, <>, ><
less than	.LT.	LT, lt	<
less than or equal	.LE.	LE, le	<=

Table 13. Conditional Clean operators

Operator	FORTRAN	Text	C
greater than	.GT.	GT, gt	>
greater than or equal	.GE.	GE, ge	> =

Because all columns to the right of column C are ignored, you can enter comments anywhere within that area. In addition, a row that has a # as the first character in column A is ignored. As a result, entire rows can be used as comments. The # can also be used to temporarily suspend a rule. Any number of comments can be present, but the number of conditional selection rules is limited to 1000 rules.

Data Format

The Data Format transformer has two primary functions:

ASCII format conversion
Formatting spreadsheet data into a text file that is suitable for printing or transferring to an environment other than Meta5.

SQL code conversion
Incorporating the spreadsheet data into an SQL program that can load the data into a database table

Parameters

Add SQL upload statements to the ASCII data? (; y; n)
This parameter specifies whether the input data should be formatted for a non-Meta5 host system or for a database server. If `y` or `yes` is specified, the output data is formatted as an SQL program that uploads the input data into a database table. If `n` or `no` is specified, the output data is formatted as an ASCII text file. The default value is no.

Table name [SQL format only]
This parameter specifies the name of the database table that will be used if the data is being formatted for a database server. The table name syntax depends on the specific database server model, but the general format includes letters, digits, dollar signs, and underscores. The first character must be a letter; the maximum is 100 characters. However, many database servers limit table names to fewer characters.

Report key [SQL format only]
This parameter specifies a number value that will be used as the reportKey value for each row of output data if it is formatted as an SQL program. The reportKey identifies the set of data stored in the table. It is required if more than one discrete set of data is stored in the same database table. Upon later query, one particular set of data can be

extracted by constraining the query on the reportKey value. Any whole number value can be entered. The default value is 0.

Report name

This parameter specifies an optional title that can be used to identify the data set in output data. Up to 100 characters of text can be specified as a report name. Because commas separate parameters in the transformer, any report name containing commas must be surrounded by double quotation marks. If no report name is specified, the output data will have no title. In the output, the report name is sent to a record with a control code of 6 (see Table 16 on page 121 for definitions of control codes).

Number of rows to designate as header rows (; 0; 1; 2;) [SQL format only]

This parameter specifies the number of rows of output data to be given the special header row control code of five. This parameter only affects the SQL control codes. See "Defining SQL Control Codes" on page 120 for more information. Any whole number can be specified; the default is 0.

Output control code columns? (; y; n) [SQL format only]

This parameter specifies whether the SQL reportKey, rowNumber, and controlCode codes are to be included in the output data if the data is formatted as an SQL program. If `y` or `yes` is specified, the control codes are included in the output data. The default value is no.

Add SQL To Drop And Create Database Table? (; y; n) [SQL format only]

This parameter specifies whether the SQL program includes the SQL statements required to drop any existing tables with the name specified in the `Table Name` field, and then create a new database table. To prevent database error messages when a table that does not exist is specified, the drop statement is protected by an error ignore command. `y` or `yes` indicates that the add and drop code should be included. The default value is no.

Numeric tab stops (; 5; 4, 8; 10, 20, 30;) [Back tab]

This parameter is a list of column numbers that are to be interpreted as right-justified tab settings. This parameter expects a series of whole numbers, each separated by a comma. For example, `15,30,45` indicates that numeric tabs should be placed in columns 15, 30, and 45. This parameter is optional. The default value is no tabs.

Character tab stops (; 5; 4, 8; 10, 20, 30;) [Normal tab]

This parameter is a list of column numbers that are to be interpreted as left-justified tab settings. This parameter expects a series of whole numbers, each separated by a comma. For example, `15,30,45` indicates that character tabs should be placed in columns 15, 30, and 45. The default value is no tabs.

Centering tab stops (; 5; 4, 8; 10, 20, 30;) [Center tab]

This parameter is a list of column numbers that are to be considered as centered tabs. This parameter expects a series of whole numbers, each separated by a comma. For example, `15,30,45` indicates that centered tabs should be placed in columns 15, 30, and 45. The default value is no tabs.

Any combination of tab stops is permitted. For example, you can specify a character tab stop in column 10 and numeric tab stops in columns 45 and 60.

Copy tab stop settings to output? (; y; n)

This parameter specifies whether the contents of the preceding three parameters should be copied to Output 1. `y` or `yes` indicates that the tab settings should be included in the output data. The default value is no.

Replace tabs with spaces in output text? (; y; n)

This parameter specifies whether any tabs in the input data should be converted into space characters. If `y` or `yes` is specified, all tabs are replaced with the appropriate number of spaces to make the output line up on the tabs as specified above. The default value is no.

Character string to append to the end of each row of data

This parameter specifies an optional string of text to be placed at the end of each row of data. This string could represent a line feed or escape sequence, which are not printed. The default value is no characters append to the data.

Characters added by this parameter must be letters, numbers, or symbols. If non-printing characters or escape sequences are required, they should be applied on the target system. For example, if an end-of-line marker is required, use a backslash. After the data file is moved to the target system, a binary editor, such as the sed stream editor on a UNIX system, can be used to change all backslashes to the required marker characters.

Maximum number of characters per line of output text (; 80; 256; 1024)

This parameter specifies the maximum width of each line of output data. Any characters exceeding the specified number are discarded. For example, 80 indicates that each row of output data should have at most 80 characters. The default value is 1024.

Add blank characters to make each output line the same length? (; y; n)

This parameter specifies whether blank spaces should be appended to the end of each line of output to make each line the same length. If `y` or `yes` is entered, any line that has fewer characters than the specified maximum number of characters per line of output text will have spaces added to the end of the line to make it the maximum length. The default value is no.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)

- Results (Output 1)

When you select either of these choices, the data associated with that choice is shown in the display area of the transformer.

Input Region Names

The Data Format transformer has one input region called Data (Input 1), which contains the data to be formatted as an ASCII file or SQL code. Input 1 must be in Text icon format. Spreadsheets can be connected to the Data Format transformer if a Text icon is used as a buffer. Input 1 must have a carriage return at the end of each line of data and have tab characters between each column of data. These characteristics are the default when you use an arrow to connect a Spreadsheet icon to a Text icon. The size of Input 1 is limited to the maximum allowed by the Text icon.

Output Region Names

The Data Format transformer has one output region called Results (Output 1), which contains data read from Input 1 after it is reformatted as an ASCII file or as SQL code. Even though Output 1 is in Text icon format, it does not have a size limit. Data intended for transmission to a non-Meta5 host can use the PC Text and PC Directory icons to bypass the Text icon size limit by sending the output to your workstation's hard disk drive. For details about the PC Directory tool, see *Getting Started with the Meta5 Developer's Desktop*.

Example

Assume that the following Spreadsheet data is to be transferred to a workstation-based application.

Table 14. Spreadsheet data

Baked Beans	6 OZ	12 OZ	18 OZ	Total
Shipments	$569.00	$68.00	$34.00	$2,089.00
Back Orders	$368.00	$728.00	$316.00	$2,620.00
Orders	$1,231.00	$34.00	$406.00	$3,536.00
Total	$2,168.00	$830.00	$756.00	$8,245.00

Many programs will accept tab-delimited files. For example, if the data in Table 14 was passed through the Format icon, the following tab-delimited file would be produced:

```
Baked      6 OZ      12 OZ      18 OZ      Total
Beans
```

Shipments	569.00	68.00	34.00	2089.00
Back Orders	368.00	728.00	316.00	2620.00
Orders	1231.00	34.00	406.00	3536.00
Total	2168.00	830.00	756.00	8245.00

This data can be passed through a communications link or a PC Directory icon and transferred to a different host. Although this data lacks dollar signs and correct tabbing, it would be acceptable to any program that reads tab-delimited text.

Many applications cannot read tab-delimited text. Instead, they rely on column location to know where input data resides. The Data Format transformer can easily reformat the original data into an ASCII file in which each data field occupies the same columns. To reformat the data to ASCII:

1. Transfer the original Spreadsheet data to a Text icon through a Capsule icon arrow.
2. Connect the Text icon to Input 1 of the Data Format transformer.
3. Set the Data Format transformer parameters as follows:

   Add SQL upload statements to the ASCII data?
   no

   Number of rows to designate as header rows [SQL format only]
   1

   Output control code columns?
   no

   Add SQL to drop and create database table?
   no

   Numeric tab stops
   26, 38,50,64

   Replace tabs with spaces in output text?
   yes

4. Run the transformer.

The data shown in Table 15 is the same spreadsheet data after it has been converted into an ASCII file by the Data Format transformer. Notice that the tabs have been removed and that the columns line up when displayed in a fixed-width typeface.

Table 15. Data converted to ASCII by the Data Format transformer

Baked Beans	6 OZ	12 OZ	18 OZ	Total

Table 15. Data converted to ASCII by the Data Format transformer

Shipments	$569.00	$68.00	$34.00	$2,089.00
Back Orders	$368.00	$728.00	$316.00	$2,620.00
Orders	$1,231.00	$34.00	$406.00	$3,536.00
Total	$2,168.00	$830.00	$756.00	$8,245.00

Alternatively, the Data Format transformer could prepare the entire data set in Table 15 for storage in a database table. The Data Format transformer generates the SQL code required to store the spreadsheet in a database in its entirety by formatting the ASCII data as a single character format database column.

To produce the results shown in Table 15, set the Data Format transformer parameters as follows:

Add SQL upload statements to the ASCII data?
yes

Table name [SQL format only]
summary

Report key [SQL format only]
123

Report name
Baked Beans

Number of rows to designate as header rows [SQL format only]
1

Output control code columns?
yes

Add SQL to drop and create database table?
yes

Numeric tab stops
26, 38,50,64

Copy tab stop settings to output?
no

Replace tabs with spaces in output text?
yes

Maximum number of characters per line of output text
255

Add blank characters to make each output line the same length?
no

Data Format

When it receives the Spreadsheet data, the Data Format transformer generates the following SQL code, which could be run in an SQL Entry icon:

```
&error ignore;

drop table summary

&error message;

create table summary(reportKey number(9),rowNumber number(9),controlCode number(9),reportText
char(255))

insert into summary values (123,0,6,'Baked Beans')

insert into summary values (123,1,5,'Baked Beans          6 OZ        12 OZ      18 OZ      Total')

insert into summary values (123,2,4,'Shipments            $569.00     $68.00     $34.00     $2,089.00'
                                                                                           )

insert into summary values (123,3,4,'Back Orders          $368.00     $728.00    $316.00    $2,620.00'
                                                                                           )

insert into summary values (123,4,4,'Orders               $1,231.00   $34.00     $406.00    $3,536.00'
                                                                                           )

insert into summary values (123,5,4,'Total                $2,168.00   $830.00    $756.00    $8,245.00'
                                                                                           )
```

Converting ASCII Format Data

ASCII format conversion is required to move a data file from the Meta5 environment to a different host computer. This transformer converts the characters into the standard ASCII character set that is easily understood by most non-Meta5 environments and can be transmitted by most data communications devices. Smaller workstations support one variety of ASCII. Most large IBM computers do not understand ASCII, but usually have a converter that reads ASCII data.

Most of the ASCII conversion process is performed by the Capsule icon arrow when data is moved from a Spreadsheet into a Text icon. However, four additional processing tasks are usually required before data from a Text tool is ready for transmission to other systems:

- Tab removal and column formatting
- Record length truncation
- Record padding
- End-of-line marker and escape sequence appending

The Data Format transformer performs these tasks. You can tailor how these tasks are completed to match the specifications of your particular application.

Formatting Records

Some computer systems use record-oriented input/output (I/O). These systems read and write data in chunks of a fixed size and format. The Data Format transformer supports these systems with two parameters that control the width of each row of output data.

For systems that have a fixed record size, the `Maximum Number Of Characters Per Line Of Output Text` parameter ensures that the Data Format transformer never creates a text line that exceeds the system record size. Some systems require that all records in a file have the same length. The `Add Blank Characters To Make Each Output Line The Same Length` parameter makes certain that all output records have the same number of characters.

Systems that use record-oriented I/O typically rely on the column position in the data to locate particular elements, rather than on column separation characters, such as commas or tabs. Examples of such a system are the formatted read and write statements in the FORTRAN and COBOL programming languages. You can match the output to the requirements of almost any computing platform and application by adjusting the parameters that control the Data Format transformer's conversion of tab stops to spaces.

Converting SQL Code

The Data Format transformer facilitates the conversion of Spreadsheet data into database elements. This conversion is done by including the data in an SQL INSERT statement. The resulting SQL statements are then run to insert the data into a database table.

This method is also used by the Write SQL transformer. However, that transformer loads each column of a table into one database field. This method works if all of the values in a column are of the same data type. The method does not work well if different cells in the same column have different types of data. Because titles and column headings are character data and other column contents are numbers, this method is not the best way to store reports or spreadsheets containing headings in a database.

The Data Format transformer can overcome this application problem by formatting the data as a database table with one large text column containing each row of the ASCII input data. Thus, all the columns in any row are combined into one large text string that can be loaded into one database text field. To move the text into a database, all you need to do is set up an SQL Entry tool that refers to a Text icon containing the Data Format output. Remember that the number of characters allowed in one text field varies by type of database.

The SQL code generated by the Data Format transformer requires a database name and a table name. The database name is specified in the SQL Entry options window. The table name can be entered into the `Table name` parameter in the

Data Format Transformer Controls window; the specified or default table name is then included in the SQL code generated by the Data Format transformer.

Before data is inserted into the database, a table must be available. You can specify that the Data Format generate an SQL statement that creates a table. Alternatively, the transformer can generate SQL code that appends new data to an existing table or drops an existing table, creates a new one, and loads the data into the latest generation of the table. When the SQL code drops the existing table, any data contained in that table is erased.

When data is formatted into SQL statements, a single quotation mark embedded in the text field can cause an error when the SQL code is processed. If a single quotation mark is to be included in the text fields, precede it with another single quotation mark. This is a requirement of the Meta5 environment.

Also, when text is formatted into SQL statements, the SQL Entry tool treats any character following an at-sign character (@) embedded in a text field as an @-variable. An example is `@AA` entered in the `Report name` parameter. The contents of @AA replaces the @-variable in the output SQL code.

Defining SQL Control Codes

Although it is useful to store one set of data in a table, it is often more convenient to store multiple data sets in one table. The Data Format transformer provides this capability by allowing you to selectively retrieve entire data sets, certain parts of each data set, or the names of all data sets in a table. This capability is enabled by information stored in the database with the ASCII data. This additional information is stored in the form of special codes, called control codes, that identify the purpose of each data row. You can use control codes to build help windows in Data Entry icons and to build online report retrieval applications.

The codes are included in the generated SQL statements that precede each row of the actual ASCII text. There are three types of codes:

reportKey
Uniquely identifies the data set

rowNumber
Uniquely identifies each row of the data set

controlCode
Describes the purpose of each row

The final SQL insert statements include a value for each of these codes and the formatted ASCII text. The SQL control code names are capitalized as shown because it is a common method of naming database fields.

The Data Format transformer automatically creates values for rowNumber and controlCode. You specify the value for reportKey in a Transformer Controls window. If the reportKeys are assigned in increasing or decreasing order, you can use an SQL ORDER BY clause to sort the information when it is retrieved.

The rowNumber code is used to order the rows of a data file. If the ASCII data was inserted into a table without this key, the data would be retrieved in alphabetic or numeric order when an ORDER BY clause was used; rowNumber allows the data rows to be displayed in the proper order. You can also use rowNumber to selectively retrieve rows of data. For example, if a particular report is stored in a table, and the third row always refers to the Eastern region, a report of that region's performance over time could be generated by selecting all rows where the rowNumber is 3 with the data ordered by reportKey (to sort numerically).

The first row of data is assigned a rowNumber value of 1. Information stored in the table that was not in the input data (such as the report name) is assigned a rowNumber value of 0.

The controlCode value describes the purpose of each row of data. These codes range in value from 1 to 6 as shown in Table 16.

Table 16. ControlCode definitions

controlCode	Definition
1	Text string containing a list of user-specified right-justified tab settings
2	Text string containing a list of user-specified left-justified tab settings
3	Text string containing a list of user-specified centered tab settings
4	Data row read from the input data
5	Heading row read from the input data
6	Text string containing the user-specified report name

The controlCode allows you to retrieve rows of data based on their contents; you do not have to know where the information is located or how many rows might be referred to. For example, to view all of the report titles stored in a database table, select all rows where the controlCode is equal to 6.

Date Format

The Date Format transformer reformats columns of dates from a Spreadsheet, Query, or SQL Entry icon. The user can select any number of columns to be converted from one date format to any other supported date format.

Parameters are provided for ten different date formats. You can change a column of dates from one format to another by specifying the one- or two-letter column names in the parameter that corresponds to the desired date format. You can select as many conversions as desired; the transformer performs all selected conversions in the same run.

The Date Format transformer modifies date formats in their original location and leaves all other data unchanged. Thus, if the input spreadsheet has one date column to be reformatted and several other columns of data, the output will contain the reformatted date information in the same column as the original and the other data will be displayed as it was in the input.

There are three reasons to use the Date Format transformer:

- To convert dates into a particular format when building a report within a capsule application.
- To convert date-like spreadsheet cells into supported Meta5 date formats. The Date Format transformer recognizes a variety of unsupported date formats. This conversion is especially useful when you move data into the Meta5 environment from another system.
- To quickly convert dates stored in a database from text to numeric format. Dates are typically stored in a database in the numeric format, which takes much less storage space than text dates and speeds up data insertion, retrieval, and manipulation.

The Date Format transformer does not convert text cells with date strings in them.

Attributed Transfer Considerations

How a spreadsheet cell date and time format is set is very important when converting dates using the Date Format transformer. For example, if you displays the options for a spreadsheet cell and select `Date Type`, you have several choices from which you can select a date format. If you set the format to `Numeric` (01/12/89) or `Standard` (January 12, 1989), the Date Format transformer will preserve the source cell attribute so that the ultimate destination cell format will also be numeric or standard.

In all cases, the underlying date value (an integer representing the number of days after or prior to January 1, 1970) remains unchanged. How this value displays depends on the interplay of the resolution set by the Date Format transformer and the attributes of the source cell.

For a detailed explanation of attributed transfer, see the *Capsule User's Guide*.

Parameters

The Date Format transformer has 11 user-adjustable parameters. Any parameter can be blank; the default value assumes no action is to be taken.

Number of header rows (0; 1;)

Specifies the number of rows at the beginning of the input data that are to be considered as a heading. The value can be 0 or any positive number; the default is 0. Heading rows are copied to the output with no date format conversions.

Columns in 'Day' format (a; a, b;) [January 4, 1987]

Specifies the columns to be displayed in the Day format.

Columns in 'Week' format (a; a, b;) [Week of January 4, 1987]
Specifies the columns to be displayed in the Week format.

Columns in 'Quad Week' format (a; a, b;) [Quad week of January 4, 1987]
Specifies the columns to be displayed in the Quad Week format.

Columns in 'Month' format (a; a, b;) [January, 1987]
Specifies the columns to be displayed in the Month format.

Columns in 'Odd BiMonth' format (a; a, b;) [DJ, 1987]
Specifies the columns to be displayed in the Odd Bimonth format.

Columns in 'Quarter' format (a; a, b;) [1Q87, 1987]
Specifies the columns to be displayed in the Quarter format.

Columns in 'Year' format (a; a, b;) [1987]
Specifies the columns to be displayed in the Year format.

Columns in 'Even BiMonth' format (a; a, b;) [JF, 1987]
Specifies the columns to be displayed in the Even Bimonth format.

Columns in 'Month/Day/Year' format (a; a, b;) [1-4-87]
Specifies the columns to be displayed in the Month/Day/Year format. This parameter also specifies the character used as a separator. The first character that does not correspond to a column name will be used as a separator. For example, entering `a,-` causes column A to be written in M-D-Y formats, with a hyphen as the separator.

The Date Format transformer can format only one column at a time in M-D-Y format. If more than one column is specified in this parameter, the transformer stops running and issues an error message.

Columns in 'Numeric' format (a; a, b;) [6212]
Specifies the columns to be displayed in the Numeric format. For example, 7335 corresponds to January 31, 1990.

The Date Format transformer modifies dates in their original column locations. If a column is specified to be converted more than once, the transformer issues an error message.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- Results (Output 1)

When you select either of these choices, the data associated with that choice is shown in the display area of the transformer.

Input Region Names

The Date Format transformer has one input region called Data (Input 1). Input 1 can contain any data in any row/column format. Input 1 has no size limit. Columns that are not specified for date conversion are moved unchanged to the output.

Output Region Names

The Date Format transformer has one output region called Results (Output 1). Output 1 contains the information read from Input 1, with the specified date conversions performed. Output 1 has no size limit.

Example

In this example, the data shown in Table 17 is sent to Input 1:

Table 17. Data sent to Input 1 of the Date Format transformer

Date_Key	Quarter	Volume	Price
6574	January 1, 1988	563,567	1.56
6665	April 1, 1988	456,764	1.67
6756	July 1, 1988	672,243	1.53

The data has two date columns, column A and column B. Column A is in the Numeric Date format. Column B is in the Day format. For the final report, the dates in column A should be reformatted from Numeric to Day format while the dates in column B are reformatted from Day to Quarter format.

Set the Data Format transformer parameters as follows:

Number of header rows
1

Columns in 'Day' format
a

Columns in 'Quarter' format
b

Table 18 shows the result of running the Date Format transformer with the data shown in Table 17 and the parameter values shown.

Table 18. Results from the Date Format transformer

Date_Key	Quarter	Volume	Price
January 1, 1988	1Q88	563,567	1.56
April 1, 1988	2Q88	456,764	1.67

Table 18. Results from the Date Format transformer

July 1, 1988	3Q88	672,243	1.53

Note that column A is set to Day format, column B is set to Quarter format, and the other two columns are unchanged from their input formats.

Using Date Formats

Meta5 supports date formats based upon the number of days since, or prior to, January 1, 1970. For more information on date formats, see "Period Table" on page 155. Any date can be displayed in one of nine formats, as shown in Table 19.

Table 19. Supported date formats

Date Format	Example
Numeric	7335
Day	January 31, 1990
Week	week of January 28, 1990
QuadWeek	quad week of January 28, 1990
Month	January, 1990
EvenBiMonth	JF, 1990
OddBiMonth	DJ, 1990
Quarter	1Q90
Year	1990

Any date in one of these nine formats can be converted to any other date format.

Using M-D-Y Date Formats

The Date Format transformer also recognizes the M-D-Y date format. This format is defined as an integer in the range 1-12 (month), followed by any separator, such as a hyphen, period, space, or slash, followed by an integer from 1-31 (day), followed by any separator, followed by an integer in the range 1970-2100 or an integer in the range of 0-99 (year). If the year is in the range 70-99, 1900 is added to its value, resulting in a year value of 1970-1999. If the year value is in the range 0-69, 2000 is added resulting in a year value of 2000-2069. This feature allows years to be expressed as two-digit numbers.

Examples of M-D-Y dates include:

```
1/31/90
01/31/1990
1-31-1990
1.31.90
1 31 1990
```

The Date Format transformer can read a few variations of the M-D-Y format that include at least the first three letters of a month name, rather than the corresponding integer. In these variations, the comma between the day and the year is optional, for example:

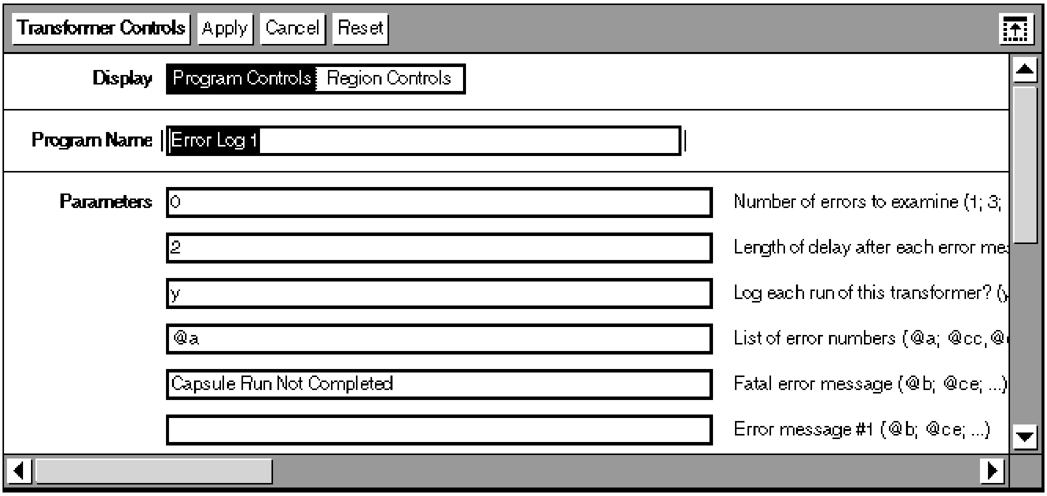
```
Jan 31, 90
Jan 31 90
Jan 31 1990
January 31, 1990
```

The Date Format transformer first converts M-D-Y dates into the numeric format, so that the M-D-Y date can be converted into any other date format. When you convert a date to the M-D-Y format, you can specify the character used to separate the parts of the date.

Error Log 1 and Error Log 2

The error log transformers, Error Log 1 and Error Log 2, allow you to flexibly trap errors that are detected while a capsule application is running and respond to them by sending the user a message. Figure 40 and Figure 41 show the Error Log 1 and Error Log 2 transformer windows. In addition, the error log transformers enable you to record the progress of a capsule application; they also allow you to stop the execution of a capsule application if conditions that you specify occur.

All error log transformer parameters are compatible with @-variables, so that these transformers can be controlled within a capsule application.



Display [✓ Program Controls] Region Controls

Program Name: Error Log 1

Parameters

0 — Number of errors to examine (1; 3;

2 — Length of delay after each error me

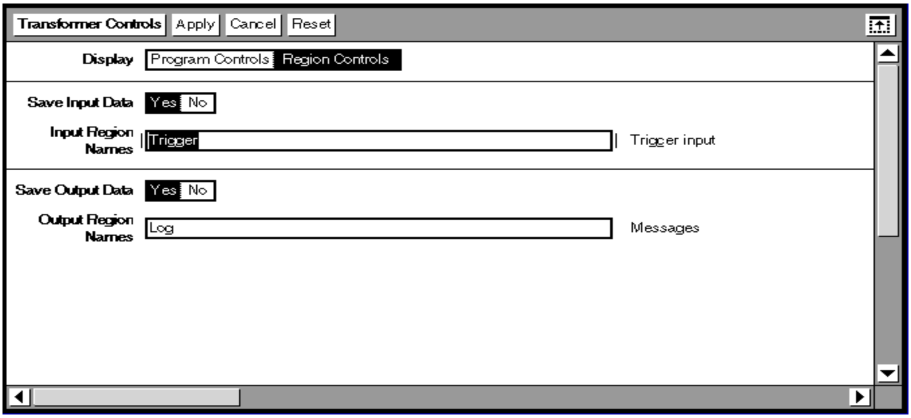
y — Log each run of this transformer? (y

@a — List of error numbers (@a; @cc, @

Capsule Run Not Completed — Fatal error message (@b; @ce; ...)

Error message #1 (@b; @ce; ...)

Figure 40. Error Log 1 transformer window



Display Program Controls [✓ Region Controls]

Save Input Data [✓ Yes] No

Input Region Names: Trigger — Trigger input

Save Output Data [✓ Yes] No

Output Region Names: Log — Messages

Figure 41. Error Log 2 transformer window

Parameters

Number of errors to examine (1; 3;)

This parameter is the number of specified errors to be checked. This option allows you to preload a list of errors into the Transformer Controls window for activation at a later time. For example, under certain conditions, you might want the transformer to check for only the first five error codes, whereas, under other conditions, you might want the transformer to check for all 10 error codes. The default is 0.

Length of delay after each error message (in seconds)? (0; 3;)

This parameter is the number of seconds the transformer should pause after placing messages in the message bar area. The default is 0.

Log each run of this transformer? (y; n)

This parameter is a yes/no question. If you enter `yes`, the program name, copyright information, and a time stamp are written to Output 1 each time the transformer runs, even if no error conditions exist. The default is `n`.

List of error numbers (@a; @cc; @cd;)

This parameter is a comma-separated (or a list separator that you have designated) list of up to 10 integers or @-variables. Each number identifies an error condition, and the value of the number determines the action taken by the transformer. The error numbers correspond to each of the 10 error messages on a one-to-one basis. For example, the first error number will be used with the first error message. The default is 0.

Fatal error message (@b; @ce;)

This parameter is the text of the important message generated when the error log transformer stops due to a fatal error.

Error Message #n

This parameter is the text of each of the 10 possible error conditions. The first message corresponds to the first integer in the `List Of Error Numbers` parameter, and so on. Each message can consist of exactly one parameter; no commas or semicolons are allowed in the message unless the entire message is enclosed in double quotation marks.

Region Controls

This section describes the input and output regions for the Error Log 1 and Error Log 2 transformers.

Input Region Names

Error Log 1 and Error Log 2 transformers have different input requirements:

- Error Log 1 requires no input. After its output region is connected to a Spreadsheet or Text icon, it will run every time the capsule application is run. The order in which it is run is determined by where it is placed in a capsule application in relation to other icon sequences. For example, if a Capsule icon contains one main stream of icons, Error Log 1 could be placed either above the main stream, which would force it to run before the stream is run, or below the main stream, which would force it to run after the stream runs.

  When Error Log 1 is placed in the upper left corner of the Capsule window, it can stop the running of a capsule application before a stream of icons is run. This placement allows you to trap cases of the user clicking on the `Run` button when running is not desired, or cases when a particular arrow must be selected before clicking on the `Run` button.

- Error Log 2 has one input region called Trigger, which is used only to run the transformer, so that it can be placed in a stream of transformers. Thus, you can control the exact point in capsule applications when Error Log 2 runs.

Output Region Names

Error Log 1 and Error Log 2 have only one output region called Log (Output 1). This output region can be connected to a Spreadsheet or Text icon, or sent to a database using the Write SQL transformer (described in “Write SQL” on page 204). Each message is on a separate line in Output 1. This output can be useful for documenting the time and outcome of a capsule application run.

In addition, you can stop the processing of a capsule application and send error messages from a capsule application to the Important Message window, or an advisory or error message to the message bar area.

Example

This example uses a capsule application that retrieves forecast expense items for department managers and places that information in a spreadsheet where each manager can review the forecasts and modify them. The modified expense items can then be stored in a database by selecting an arrow connecting the spreadsheet and an SQL Entry icon, which is called Load Data.

When managers load the Spreadsheet from an SQL Entry icon called Retrieve Data, they must click on the `Run` button. However, if a manager changes the spreadsheet and then clicks on the `Run` button, the changes are overwritten when the raw forecasts are retrieved and sent to the spreadsheet.

Managers can use the Error Log 1 transformer to resolve this problem:

- This transformer is placed in the upper left corner of the Capsule window and attached to a Text icon so that it will be the first icon that is run when the manager clicks on the `Run` button.
- The SQL code used to retrieve the initial forecasts is modified to set the @-variable @CA to 99.
- The SQL code that stores the modified expenses is modified to set that @-variable to 0.
- The parameters of the Error Log 1 Transformer Controls window are set as follows:

  Number of errors to examine

  1

  Length of delay after each error message (in seconds)?

  5

  Log each run of this transformer?

  yes

  List of error numbers

  @CA

Fatal error message

You cannot run the Capsule application now.

Error message #1

Select the arrow and click on the `Run` button.

When a manager initially opens the Capsule icon, the value of @CA is set to 0. Consequently, the `Run` button can be clicked, the Error Log 1 transformer will run and detect no errors, and the Retrieve Data SQL Entry icon will access the forecast expense information and load it into the spreadsheet. At the same time, that SQL Entry icon sets @CA to 99. Thus, when the manager tries to click the `Run` button without selecting the output arrow from the Spreadsheet, the Error Log 1 transformer will detect an error and stop the capsule application from running before the Retrieve Data SQL Entry icon runs. At the same time, the Error Log 1 transformer will send the following message to the desktop message bar area:

```
Select the arrow and click on the Run button.
```

Also, the following text is sent to a second Important Message Display window.

```
You cannot run the Capsule application now.
```

Finally, the following text is sent to the Output 1 region. Because this error stops the capsule application, the contents of this output region are not sent to the attached Text icon.

```
Date: June 4, 2001 Time: 11:47:25
Error #99 (Fatal): Select the arrow and click on the Run button.
```

After the manager takes the recommended action, @CA is reset to 0. The next time the entire capsule application is run, the Error Log 1 transformer will not detect an error and the capsule application will run correctly.

Error Codes

The error log transformers use a set of error code numbers to determine whether message text should be sent, the content of the message text, its destination, and whether any other action is necessary. Each transformer can have up to 10 error codes. Thus, each transformer can check for up to 10 error conditions while a capsule application runs and can respond to each with a specific message and, if so specified, can stop the capsule application run.

An error code can be any whole number between 1 and 99. If the error code value is greater than 1 and less than 50, the transformer sends the specified message text to the specified output. If the error code value is greater than 50, the transformer sends the message to the specified output, sends a specified fatal error message, and stops execution of the capsule application. Table 20 shows how the transformers react to specific error numbers.

Table 20. Error log transformer error code numbers

Error number	Output	Post message	Stop capsule application
1-9	x		
10-19	x		
20-49	x	x	
50-59	x	x	
60-69	x	x	
70-99	x	x	x

When an error log transformer stops a capsule application, the output regions are not transferred out of the transformer. As a result, the fatal error message is not copied from Output 1 to the icon to which it is connected. If that output information is required, you must use two copies of the error log transformer: the first to send the error text to the output icon, and the second to stop the capsule application.

Composing Error Messages

An error message is composed of three basic parts:

- The term Message, Error, or Fatal Error.

  One of these terms is automatically supplied by the transformer and depends on the error code number:

  1 through 9
  Message #nn

  10 through 49
  Error #nn

  50 or greater
  Error #nn (Fatal)

- The error number

  You can set the error number using a Spreadsheet or SQL Entry icon and transmit it to the error log transformer using an @-variable.

- The text

  You can specify the text of the error message with the appropriate `Error Message #` parameter.

The following examples show how these components fit together:

```
Message #9: Capsule now formatting a report based on 43 rows of
data.

Error #35: Fewer than 30 rows of data were returned from the
database.
The results are probably not statistically significant.

Error #99 (Fatal): No valid data rows were returned from the
database. Processing
cannot continue.
```

The content of the error message depends upon the location to which it is written. Specifically, the complete error messages above are sent only to Output 1. Only the error message text is sent to the Post Message transformer and the Append Message transformer.

Function

The Function transformer evaluates expressions and applies data transformation expressions to each data row it receives. Essentially, the Function transformer is a spreadsheet in a transformer. The expressions can be mathematical expressions, certain spreadsheet functions (including string functions), data formatting, or date conversions. The Function transformer supports the Transformer Execution Language (TXL), as described in “Appendix 6. Transformer Execution Language,” on page 497.

The Function transformer offers the following benefits:

Programming convenience
The Function transformer requires the formula to be entered only once, unlike a spreadsheet where formulas used for calculating a column of data are replicated for the entire length of the column. Because the Function transformer formulas can be written in a Text document, they can include comments to make complex projects easier to develop, maintain, and document.

Increased speed
Because only one copy of a formula is required for each column, and the transformer does not have the overhead of providing a screen display, run time is shortened.

Improved memory usage
When you use a spreadsheet, the spreadsheet program, the data file, and copies of every formula must be in the memory at the same time, which limits the size of a spreadsheet data set. The Function transformer streams data by storing only one input row at a time while calculations are running, making it possible to handle an unlimited amount of data. Because only one copy of a formula is required for each column, the transformer is compact.

You can use a Function transformer anywhere spreadsheet-type calculations are required, or anywhere a Spreadsheet icon is needed but cannot be used because the data set has more than 10,000 rows. The calculations performed by the Function transformer are especially well suited to computations where one or more equations are applied repeatedly.

One of the most powerful applications of the Function transformer is to perform calculations after accessing data. Often, data from a database service must have calculations performed on it, which typically impacts performance. In many cases, these queries can be reformulated to retrieve the data using the fastest method; then the Function transformer can be used to perform the calculations. This two-step procedure is generally much faster than using the Query tool alone, and it provides the richer set of calculations that TXL supports.

Another useful benefit of the Function Transformer is its ability to produce a subset of rows based on exception rules, using the No-Row Test function, described in “No-Row Test” on page 512.

More information about String manipulation functions can be found in Appendix A.

Parameters

Number of header rows (0; 1;) Which header row contains titles? (,0; 1;)

This parameter specifies two values. The first is the number of rows that should be read and ignored before the data is read. The value can be blank, or any number zero or larger; 1 is the default. The second value, `Which header row contains titles`, specifies the heading row to be used as the column title. The value can be blank, or any number zero or larger; no value is shown for the default. The `Which header row contains titles` value must be equal to or less than the `Number of header rows` value.

Show column titles in 'Output 1'? (y; n)

This parameter specifies whether to print the column titles in the output. If `y` or `yes` is entered and at least one column has been specified, the titles display. If not, the output will not contain a heading row. If no titles are specified, column letters will display in the output.

Select columns to convert date (a; a,b,c;)

This parameter allows you to specify that a column is to be converted to a Meta5 date. The valid values include a blank or a list of column names separated by list separators. Converted columns display in the first column of Output 1, before any other columns. The default value is blank.

If the specified input column is all alphabetic characters, the converted date is always January 1, 1970.

Output specification #n (a; a, c, f; a/10, c+d;)

These five parameters contain the TXL program that calculates the output from the input data. These parameters are free-form, as long as they represent a valid TXL program. Because each `Output specification` parameter is limited to 512 characters, five parameters are provided for the `Output specification`, numbered 1 through 5. TXL programs that extend beyond the 512 character limit can be split among the five `Output specification` parameters.

An output specification must be terminated by a comma and a space if another output specification follows. If a trailing space is not included, the final comma will be removed automatically.

Substitution string

The five `Substitution string` parameters use the #STRA, #STRB, #STRC, #STRD, and #STRE constant values in the TXL language, so that user-specified strings can be included in the output and passed in directly or by @-variables. If data is directly entered in this parameter, a maximum of 512 characters is the limit.

Store 'Output 1' on fileserver or in workstation? (f; w)

This parameter determines where the output data generated by the Function transformer is stored. If you type `w` or `workstation`, the output data remains in the workstation memory. Otherwise, it is stored on the file server. The default value is `f` for file server. This feature allows for output data sets whose size might exceed the available memory of the workstation.

The default choice is to create Output 1 on the network file server, which allows storage of an unlimited number of rows. Alternatively, the data can be stored in the workstation memory. This method is limited by the memory of the workstation, but it is more efficient when the output is to be put into another transformer. This choice is controlled by the `Store 'Output 1' on file server or in workstation` parameter.

Treat '=Error' as zero? (y; n)

This parameter specifies whether cells containing =Error are treated as a zero. `no` or n specifies that the result of any calculation containing =Error be treated as =Error. To treat =Error as a zero, enter `Yes` or `y`. The default is `n`.

The calculation 100/NA will always return =Error.

Treat text as zero? (y; n)

This parameter specifies whether cells containing text are treated as zero. `no` or `n` specifies that the result of any calculation containing text be treated as =Error. `Yes` or `y` indicates that text is treated as a zero. The default is `n`.

Treat '=N/A' as zero (z); ignore (i); pass through (p)?

This parameter specifies whether cells containing =N/A are treated as a zero, ignored, or passed through. For example:

z	Specifies that all =N/As will be replaced with zeros
i	Specifies that all =N/As will be ignored and excluded from calculations

p Specifies that the result of any calculation containing =N/A will be treated as =N/A

The calculation 100/NA will always return =Error.

The default is `z`.

Region Controls

The `Display Data For` field in the Function transformer window contains the following choices:

- Input 1 is the raw data
- Output 1 is the processed data set
- Output 2 is the log of the `Output specification n` parameters

When you select any one of these choices, the data associated with that choice is shown in the display area of the transformer.

Input Region Names

Input data can be in any format, with any number of rows or columns. Because only one row is processed at a time, there is no practical limit to the size of the input data.

Output Region Names

Output 1 contains the data from Input 1, transformed according to the specification parameters. You determine the Output 1 format.

Output 2 contains a log of the `Output specification n` parameters. Each column contains the TXL element used to calculate that column's output.

Example

In this example, the following data shows what percentage of sales dollars each causal is driving. The data is contained in a spreadsheet that is connected to Input 1 of the Function transformer.

A	B	C	D	E	F
	Sales Dollars	$Sales w/Causal	$ Sales w/Ad only	$ Sales w/Display Only	$ Sales w/display + Ad
CINCINNATI	758,387.00	64,584.00	8,522.00	40,162.00	15,897.00
CLEVELAND	1,004,822.00	50,741.00	25,150.00	21,247.00	4,342.00

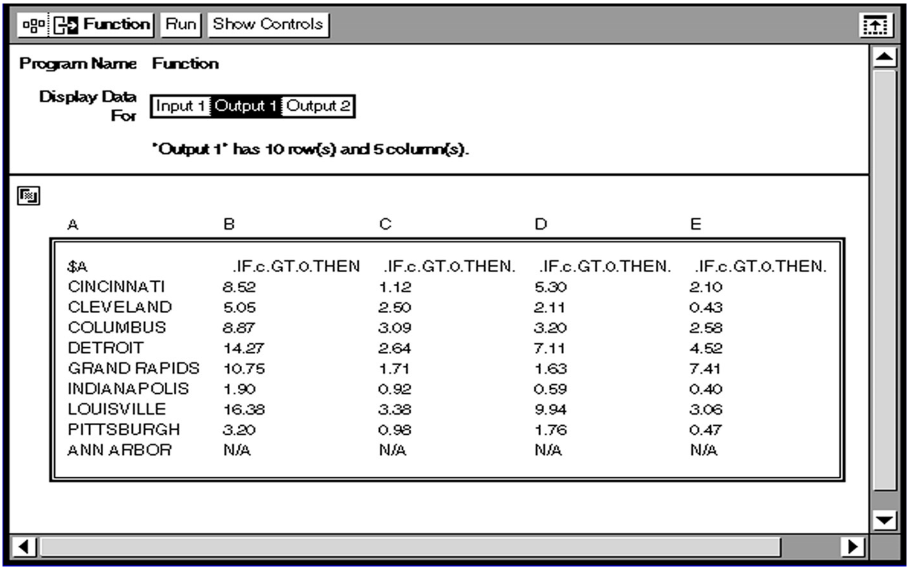
COLUMBUS	533,719.00	47,331.00	16,470.00	17,103.00	13,753.00
DETROIT	1,706,762.00	243,592.00	45,047.00	121,388.00	77,155.00
GRAND RAPIDS	464,473.00	49,943.00	7,944.00	7,583.00	34,413.00
INDIANAPOLIS	863,291.00	16,425.00	7,918.00	5,058.00	3,448.00
LOUISVILLE	454,550.00	74,470.00	15,371.00	45,193.00	13,902.00
PITTSBURGH	956,031.00	30,618.00	9,337.00	16,800.00	4,480.00

Type or copy the following TXL syntax (comments in square brackets are optional) in the `Output specification #1` field.

```
$A, [Calculate Share Sales with Causals % of Total Sales]
.IF.c.GT.0.THEN.  d/c*100. ELSE.#NA, [Calculate Share Sales w/ Ad %
of Total Sales].IF.c.GT.0.THEN. f/c*100.ELSE.#NA, [Calculate Share
Sales w/ Display % of Total Sales] .IF.c.GT.0.THEN.
f/c*100.ELSE.#NA, [Calculate Shar Sales w/ Display + Ad % of Total
Sales] .IF.c.GT.0.THEN. g/c*100.ELSE.#NA
```

If a syntax string is longer than the space in the `Output specification #1` parameter field, type the string in a Text tool and then copy the string into the parameter field.

After you run the transformer, the data shown in Figure 42 is displayed in Output 1.



Program Name Function

Display Data For Input 1 Output 1 Output 2

"Output 1" has 10 row(s) and 5 column(s).

A	B	C	D	E
$A	.IF.c.GT.o.THEN	.IF.c.GT.o.THEN.	.IF.c.GT.o.THEN.	.IF.c.GT.o.THEN.
CINCINNATI	8.52	1.12	5.30	2.10
CLEVELAND	5.05	2.50	2.11	0.43
COLUMBUS	8.87	3.09	3.20	2.58
DETROIT	14.27	2.64	7.11	4.52
GRAND RAPIDS	10.75	1.71	1.63	7.41
INDIANAPOLIS	1.90	0.92	0.59	0.40
LOUISVILLE	16.38	3.38	9.94	3.06
PITTSBURGH	3.20	0.98	1.76	0.47
ANN ARBOR	N/A	N/A	N/A	N/A

Figure 42. Function transformer output

Translating Dates

The Function transformer can perform date translations. Columns specified in the `Select columns to convert date` parameter are converted to the Meta5 day resolution format such as December 26, 1993. Date formats recognized are:

- Numeric (for example: the numeric date 8760 is December 26, 1993)
- Meta5 (for example: quad week of December 26, 1993)
- MMDDYY (for example: 12261993, 12/26/1993, 12/26/1993, 12-26-1993, or 12.26.93)

See “Period Table” on page 155 for more information about the valid date formats.

The MMDDYY format uses a number from 1 to 12 fo r the month, 1 to 31 for the day, and 1 to 99, or 1900 to 2100 for the year (year can be greater than 2100). These numbers are separated by any character. Date columns always display before the calculated columns.

Header

The Header transformer formats text read from a Query tool, SQL Entry icon, or Spreadsheet tool into column headings suitable for a report to enhance the appearance of a printed report.

The Header transformer can read text strings from a spreadsheet cell or from the data returned by the Query tool or SQL Entry tool. These text strings, up to 127 characters each, are displayed in column A of the data passed into the transformer. The Header transformer then arranges the text strings into column titles for a spreadsheet. If any column title is too wide to fit into a particular spreadsheet cell, the Header transformer creates a multiple row heading. For example, the Header transformer formats:

```
Percent Change Versus One Year Ago
```

as:

```
Percent Change
Versus One
Year Ago
```

The Header transformer lets you define exactly how the heading will be displayed. The parameters specify the number of rows that the heading can occupy, the width of the header columns, and the number of columns in the heading.

Two additional options allow further control over the appearance of a heading. The Header transformer includes a feature that lets you specify exactly where to place line breaks to create additional rows. You can also specify a special character to be used as the line break character.

Finally, you can format the text as all uppercase, lowercase, mixed case, or with initial capitals.

When you specify line breaks, the Header transformer does not take the width of individual characters into account. If you use micro-spaced fonts, strings are not always broken at the best possible place.

Parameters

The Header transformer has five Transformer Controls window options. Each parameter expects a single non-negative whole number, with the exception of `Line break symbol`, which expects a single character.

Number of rows allowed in header (1; 5;)

This parameter specifies the maximum height of the header. If the specified number is more than that allowed by the header, the extra rows are filled with blank cells. If the header requires more rows than allowed, the title is truncated. The default is 5.

Width of each header column (10; 15;) [# characters]

This parameter specifies the number of characters in each spreadsheet cell. The default value is 12 characters. Any whole number between 1 and 127 (inclusive) is allowed.

Conversion method (0; 1; 2; 3; 4)

This parameter specifies the text formatting method to use. See “Using Text Formatting Options” on page 140 for more information. The values are 0, 1, 2, 3, or 4; 0 is the default.

Line break symbol (@; #; !; $;)

This parameter specifies the symbol to be interpreted as a line break command. Any single character can be specified. The default is *.

Maximum number of columns in header (1; 5;)

This parameter specifies the number of columns in the heading. Any whole number between 1 and 99 (inclusive) is acceptable. The default is 20.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- Results (Output 1)

When you select either of these choices, the data associated with that choice is shown in the display area of the transformer.

Input Region Names

The Header transformer has one input region called Data (Input 1). Input 1 is assumed to be a Spreadsheet, Query, or SQL Entry icon data set with each cell of column A containing a text string. If the data is not a text string, it is converted into text. All Meta5 data types are supported, including dates, N/A, Error, Boolean, and numeric. Only column A is examined; extra items on each line are ignored.

Output Region Names

The Header transformer has one output region called Results (Output 1). Output 1 has one column for each row in Input 1. The number of rows in Output 1 is controlled by the parameter `Number Of Rows Allowed In Header`. Any blank cells that might be displayed in the output actually contain a single space so that empty cells will not be displayed as cells containing a single period, which is the default format of a Spreadsheet tool.

Example

In this example, if the following lines of text are in Input 1:

```
This is a longer line with many short words
Percent Change Versus Prev
Prev* Share Point Change
Percent*change*versus*prev
```

The Header transformer parameters are set as follows:

Number of rows allowed in header

5

Width of each header column
13

Conversion method
4

Line break symbol
*

Maximum number of columns in header
4

When you click on the `Run` button, the Header transformer produces the following column headers:

```
This Is A                                                      Percent
Longer Line       Percent                 Prev                 Change
With Many         Change Versus           Share Point          Versus
Short Words       Prev                    Change               Prev
```

The first two input lines were broken into multiple line headers according to the 13-character limit imposed by the `Width of Each header column` parameter. The last input line was broken at points determined by the placement of the break character (an asterisk) in the input. The third line was broken according to both methods. In addition, even though the input lines had different capitalization, all of the words in the headers begin with capital letters, followed by lowercase letters. The Header transformer automatically capitalized the first character in each word because the conversion method of 4 specifies that style of capitalization.

Using Text Formatting Options

The Header transformer supports several text processing options. The conversion methods are shown in Table 21.

Table 21. Header transformer text formatting options

Conversion Method	Definition
0	No change (default)
1	Convert all characters to uppercase
2	Convert all characters to lowercase
3	Convert the first character of each word to uppercase, leave all other characters unchanged
4	Convert the first character of each word to uppercase, convert all other characters to lowercase

Message

The Message transformer allows messages to be sent from a capsule application. It allows the capsule builder to access either of the desktop message passing methods: the message area or the Important Message window. Each method has different advantages:

- A message sent to the Important Message window is retained until the user clicks the `OK` button in the window.
- Messages sent to the Important Message window cannot be viewed until the capsule application has finished, whereas message area text can be viewed while the capsule application is running.

Messages sent to the message area include:

- Copyright messages.
- Messages for tracking the progress of a capsule application. Message transformers can be placed at various steps in the capsule application to allow the user to watch the capsule application run.
- Debugging messages. A capsule builder can populate a problem capsule application with Message transformers to trace processing and to view the run time values of @-variables.

Important messages are generally of a more permanent or serious nature. Because the message is retained until dismissed by the user, the user has time to understand or make a record of the information. Error messages are often displayed as important messages.

The Append Message and Post Message transformers enable you to send a message to the desktop message area. The Append Message and Post Message transformers are explained in “Append Message” on page 103 and “Post Message” on page 163.

Parameters

Create an important message? (y; yes; n; no)

This parameter specifies whether a message is to be sent to the Important Message Display window. `Yes` or `y` indicates that the `Message To Display` is to be sent to the Important Message Display window; `no` or `n` indicates that it should not be sent. The default value is `no`. In many cases, the value used in this parameter will actually be an @-variable which can be set to either `yes` or `no` elsewhere in the capsule application.

Write to message bar? (y; yes; n; no)

This parameter specifies whether a message is to be displayed in the desktop message area. `Yes` or `y` indicates that the `Message To Display` is to be displayed in the message area. `No` or `n` indicates that no message is to be displayed. The default value is `no`. In many cases, the value used in this parameter will be an @-variable, which can be set to either `yes` or `no` elsewhere in the capsule application.

Number of seconds to wait after writing to message bar (1; 5; 10;)

This parameter specifies the number of seconds the Message transformer is to wait before the message written to the message area is erased. Any whole number can be specified (@-variables can also be used). The default value is 5 seconds.

Terminate the capsule after displaying the message? (y; yes; n; no)

This parameter specifies whether the processing of a capsule application is to stop after the Message transformer has finished displaying its message. `Yes` or `y` causes the capsule application to stop after the Message transformer has finished. `No` or `n` indicates that the capsule application is to continue normally after the Message transformer has finished. The default value is `no`. In many cases, the value used in this parameter will be an @-variable, which can be set to either `yes` or `no` elsewhere in the capsule application.

Message to display

This parameter is the text of the message written to the message area or an Important Message window. The message can be up to 512 characters in length and can include references to @-variables. The 512-character limit includes the contents of any @-variables, not just the length of the @-variable names.

Terminating a Capsule Application

You can set a Message transformer to stop a running capsule application after the messages have been displayed. An effective error message system would write a message to the message area and to an Important Message window, and then stop the capsule application.

The Message transformer has no input or output regions.

Multiple Select

The Multiple Select transformer creates up to 10 output regions that contain data columns selected from a single source.

The Multiple Select transformer is similar to the Select transformer but provides more than one selection per transformer. Multiple Select is more efficient and faster than using several Select transformers to perform multiple selections.

Multiple Select is useful when a single SQL SELECT statement is used to return a large group of loosely related data columns. Multiple Select allows the data to be broken into pieces, regardless of how large the data set is. Thus, the user can avoid repeating a query when certain parts are used more than once. Multiple Select also allows the user to work with a query composed of several smaller queries joined together. This can save a great deal of time if the database server is located on a remote network. In this case, the number of rows often has a greater impact on transmission time than the number of columns.

Breaking data columns into separate output regions can also facilitate creating reports that summarize the different pieces of information for the same database rows.

Multiple Select is also helpful when it is used with a Plot tool. The Plot tool attempts to plot all data in a particular row or column. The Select transformer is often useful in controlling the amount of data plotted. Multiple Select allows up to 10 plot data sets to be created at a time.

Parameters

Columns to include in 'Output #n' (; a; a, b, c; all;)

This parameter specifies the input columns that should be sent to the nth output region. Columns are specified as a comma-separated list of column names. For example, `b,h,d` specifies that the nth output region should contain columns B, H, and D, in that order. The columns specified for each output region do not have to be unique; the same column can be included in more than one output region. If no columns are specified, the corresponding output region will be empty. Items specified that are not valid column names cause the transformer to stop and issue an important message.

Value to substitute in place of missing cells beyond the end of row of input data (; n/a; error; blank;)

This parameter specifies the value to be displayed in output cells when these cells are beyond the end of a row. For example, if the transformer is requested to move columns x and y to an output region but those columns are beyond the end of the longest row of data, the transformer inserts N/A, Error, or a blank in the appropriate output cells. You can specify only one of the values in Table 22 on page 144 for placement in cells beyond the end of the row.

Table 22. Multiple Select substitution values

Value	Explanation
blank	Set the cell type to blank
na	Set the cell to =N/A
n/a	Same as na
error	Set the cell to =Error

Capitalization and spacing is not significant in the special substitution values. The default value is blank.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- Output 1
- Output 2
- Output 3
- Output 4
- Output 5
- Output 6
- Output 7
- Output 8
- Output 9
- Output 10

When you select any one of these choices, the data associated with that choice is shown in the display area of the transformer.

Input Region Names

The Multiple Select transformer has one input region called Data (Input 1). Input 1 can contain any row- and column-formatted data. This input region does not have a size limit.

Output Region Names

The Multiple Select transformer has 10 output regions. Each of the output regions contains columns of data selected from Input 1. The output regions are limited in size only by the amount of available file server disk space.

Example

The following data, which is the result of a query, is contained in a spreadsheet that is connected to Input 1 of the Multiple Select transformer.

	East	West	East	West

	East	West	East	West	Sales	Sales	Share	Share
Product	Sales	Sales	Share	Share	Change	Change	Change	Change
Grn Beans 12oz	681,313	1,352,029	(20.66)	43.19	16.85	28.64	0.15	10.05
Grn Beans 18oz	225,061	471,726	(24.11)	50.51	5.57	9.99	(0.20)	3.82
Yell. Beans 12oz	143,244	210,390	NA	NA	3.54	4.46	NA	NA
Yell. Beans 18oz	6,352	9,905	4.68	33.62	0.16	0.21	0.04	0.06
Kid. Beans 24oz	17,951	46,676	(30.90)	41.88	0.44	0.99	(0.06)	0.34

The Multiple Select transformer parameters are set as follows:

Columns to include in 'Output #1'
a,b,c

Columns to include in 'Output #2'
a,d,e

Columns to include in 'Output #3'
a,f,g

Columns to include in 'Output #4'
a,h,i

The transformer output regions Output 1, Output 2, Output 3, and Output 4 are connected to four spreadsheets named Sales Data, Share Data, Sales Change Data, and Share Change Data, respectively.

After the capsule application runs, the Sales Data spreadsheet contains this information:

	East	West
Product	Sales	Sales
Grn Beans 12oz	681,313.00	1,352,029.00
Grn Beans 18oz	225,061.00	471,726.00
Yell. Beans 12oz	143,244.00	210,390.00
Yell. Beans 18oz	6,352.00	9,905.00
Kid. Beans 24oz	17,951.00	46,676.00

The Share Data spreadsheet contains this information:

	East	West
Product	Share	Share

Grn Beans 12oz	(20.66)	43.19
Grn Beans 18oz	(24.11)	50.51
Yell. Beans 12oz	NA	NA
Yell. Beans 18oz	4.68	33.62
Kid. Beans 24oz	(30.90)	41.88

The Sales Change Data spreadsheet contains this information:

	East	West
	Sales	Sales
Product	Change	Change
Grn Beans 12oz	16.85	28.64
Grn Beans 18oz	5.57	9.99
Yell. Beans 12oz	3.54	4.46
Yell. Beans 18oz	0.16	0.21
Kid. Beans 24oz	0.44	0.99

The Share Change Data spreadsheet contains this information:

	East	West
	Share	Share
Product	Change	Change
Grn Beans 12oz	0.15	10.05
Grn Beans 18oz	(0.20)	3.82
Yell. Beans 12oz	NA	NA
Yell. Beans 18oz	0.04	0.06
Kid. Beans 24oz	(0.06)	0.34

Multiple Split

The Multiple Split transformer creates up to 10 output regions containing rows of data selected from a single source. Multiple Split works like the Split Header transformer, but its functionality is extended so that it can select more than one section at a time from the input. Multiple Split is more efficient and faster than using several Split Header transformers to perform multiple selections.

The Multiple Split transformer is useful for manipulating data sets that are too large for the Spreadsheet icon. Data can be broken into multiple parts and processed separately. A second application of Multiple Split is to break spreadsheets into page-sized pieces. Thus the user can preformat report pages as desired without having to rely on the limited header and footer options provided by a spreadsheet.

Parameters

Number of header rows (; 0; 1; 5;)

This parameter specifies the number of header rows in Input 1. Any whole number can be specified; the default value is 0. The header can be selectively included in or excluded from the various output regions.

Lower, upper row numbers to include in 'Output #n' (1, 999; 5; 10;) Include header? (y; n)

This parameter is duplicated 10 times, once for each of the output regions.

This parameter specifies the range of input rows to be sent to the specified output region and whether a header is displayed in that region. The lower row number specifies the first row copied from Input 1 to Output n. The upper row number specifies the last row copied from Input 1 to Output n. All rows between the lower and upper rows are also copied. Include header specifies whether the header read from Input 1 is copied into Output n. `Yes` or `y` indicates that a header is sent to the output and `no` or `n` indicates that a header is not sent.

Each parameter value is optional; if either the first or second value is omitted, a comma must be entered as a place holder.

For example, `7,18,n` specifies that the given output region should consist of rows 7 through 18 inclusive of Input 1, with no header.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- Output 1
- Output 2
- Output 3
- Output 4
- Output 5
- Output 6
- Output 7

- Output 8
- Output 9
- Output 10

When you select any one of these choices, the data associated with that choice is shown in the display area of the transformer.

Input Region Names

The Multiple Split transformer has one input region. Input 1 contains row- and column-formatted data and is not limited in size.

Output Region Names

The Multiple Split transformer has 10 output regions. Each output region contains rows of data selected from Input 1 and is not limited in size. The number of output regions used by the transformer depends upon user specifications. For example, if only two output regions are necessary, the remaining eight regions are empty.

Example

In this example, the following data file is copied into Input 1 of the Multiple Split transformer:

Market	Product	Period	Volume
NY	A	1	234
NY	A	2	34
NY	A	3	352
NY	A	4	342
NY	B	1	23
NY	B	2	34
NY	B	3	54
NY	B	4	12
LA	A	1	345
LA	A	2	654
LA	A	3	234
LA	A	4	567
LA	B	1	23
LA	B	2	74
LA	B	3	82

LA	B	4	37

When processed by the Multiple Split transformer, the following data is sent to Output 1:

Market	Product	Period	Volume
NY	A	1	234
NY	A	2	34
NY	A	3	352
NY	A	4	342
NY	B	1	23
NY	B	2	34
NY	B	3	54
NY	B	4	12

The following data file is sent to Output 2:

Each output region has one header row copied directly from the input data.

Market	Product	Period	Volume
LA	A	1	345
LA	A	2	654
LA	A	3	234
LA	A	4	567
LA	B	1	23
LA	B	2	74
LA	B	3	82
LA	B	4	37

Output 1 contains rows 2 through 9 of the input data, and Output 2 contains rows 10 through the end of the input data. The upper limit did not need to be specified for Output 2 because the transformer stopped when it encountered the end of the data.

Page Break

The Page Break transformer formats row- and column-formatted data, such as that from a Spreadsheet, Query tool, or SQL Entry icon, into a paginated document that is suitable for printing. The Page Break transformer divides the document it creates into a series of pages. Each page consists of:

- A header, with optional white space inserted as specified
- A series of data rows, with optional white space inserted as specified

The Transformer Controls window parameters specify values for the dimensions shown in Figure 43.

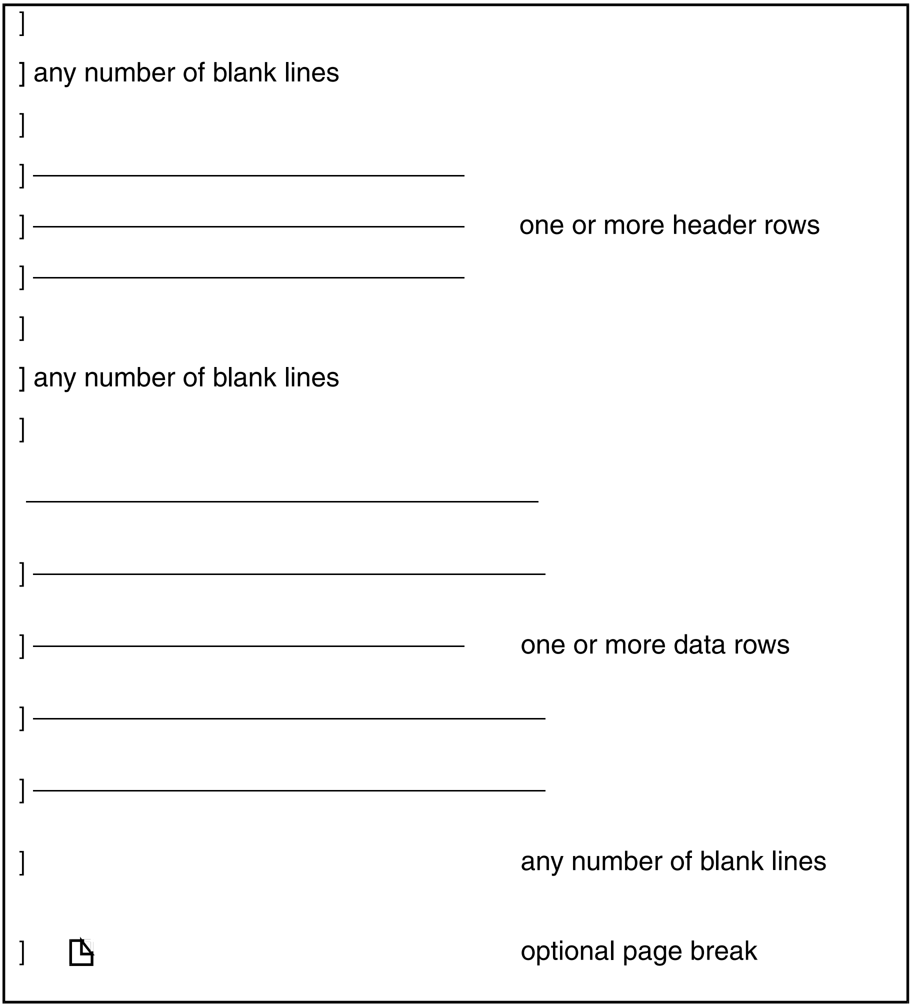

any number of blank lines
one or more header rows
any number of blank lines
one or more data rows
any number of blank lines
optional page break

Figure 43. Page Break transformer dimensions

The Page Break transformer first writes the header on a page, and then fills the page with data. If there is still more data, the Page Break transformer generates a page break if the output is to a Text Icon, and then writes the header and data for the next rows.

The page header can be part of the report data or can come from another source. The Page Break transformer can be instructed to read the header from either input region.

The Page Break transformer can insert page breaks at the appropriate points. If the output is sent to a spreadsheet, page breaks are displayed on the desktop only as small black boxes. In portrait orientation, 72 rows and five columns of a typical spreadsheet can fit on a printed page. In landscape orientation, 52 rows and eight columns will fit on a printed page. Setting up a spreadsheet for automatic pagination requires that you make a test print of one data set and then set up the parameters in the Transformer Controls window to break the pages at the appropriate points.

Parameters

The Page Break Transformer Controls window contains 10 user parameters. All parameters expect a single value, either a whole number, or, for the last parameter, a `yes` or `no` response.

Read header from 'Input 1' or from data in 'Input 2' ? (1; 2)

This parameter specifies whether the page header is to be read from Input 1 or extracted from the data in Input 2. Allowable choices are 1 and 2, referring to Input 1 and Input 2. The default is 2. All other values are interpreted as the default.

If reading header from 'Input 2', number of rows to skip before reading header? (0; 1;)

This parameter specifies the number of rows of the Input 2 data to be discarded before the header is read. In addition, if the header is being read from Input 1, this parameter specifies how many rows in Input 2 should be skipped before reading the data in Input 2. This parameter expects a positive whole number; all other values are interpreted as the default, which is 0.

If reading header from 'Input 2', number of rows in header? (0; 1;)

This parameter specifies the number of rows of the Input 2 data that are considered as header rows to be placed at the top of each page. This parameter expects a positive whole number; all other values are interpreted as the default, which is 0. The value for this parameter is ignored if the header is read from Input 1.

If reading header from 'Input 2', number of rows to skip after reading header? (0; 1;)

This parameter specifies the number of rows of the Input 2 data to be discarded after the header is read. This parameter expects a positive whole number; all other values are interpreted as the default, which is 0. The value is ignored if the header is read from Input 1.

Number of blank rows to write on each page before header (0; 1;)

This parameter specifies the number of blank spreadsheet rows or blank text lines to be placed before the page header at the start of each page. This parameter expects a positive whole number; all other values are interpreted as the default, which is 0. This number does not include the header option for the Text icon.

Number of blank rows to write on each page between the header and the data (0; 1;)

This parameter specifies the number of blank spreadsheet rows or blank text lines to be placed after the page header but before the start of the data on each page. This parameter expects a positive whole number; all other values are interpreted as the default, which is 0.

Number of rows of data to write on each page? (0; 1;)

This parameter specifies the number of data rows read from Input 2 to be written on each page. This parameter expects a positive whole number; all other values are interpreted as an error, causing the transformer to stop and issue an error message. For example, if Input 2 contains 240 rows of data, and 50 is specified for this parameter, the transformer generates a document containing four pages of 50 rows each, and a final page containing the remaining 40 rows. The default is 0.

How many fewer rows of data to write on the first page (0; 1;)

This parameter specifies how many fewer rows should be written on the first page than on the remaining pages. This parameter sets space aside on the top of the first page for a document header. This parameter expects a positive whole number; all other values are interpreted as the default, which is 0. For example, if 50 is specified as the value for the `Number of rows of data to write on each page` parameter, and a document header of 10 lines is to be placed on the first page, the parameter would be 10.

Number of blank rows to write on each page after the data (0; 1;)

This parameter specifies the number of blank spreadsheet rows or blank text lines to be placed after the data but before the start of the next page. This parameter expects a positive whole number; all other values are interpreted as the default, which is 0.

Place a page break at the end of each page (y; n)

This parameter specifies whether to place a page break symbol at the end of each page. Page breaks only work in Text icons; in Spreadsheet icons, they are displayed as little black boxes. `y` or `yes` specifies that page breaks are to be included in the output. All other responses are treated as `no`, the default value.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Heading (Input 1)
- Data (Input 2)
- Results (Output 1)

When you select any one of these choices, the data associated with that choice is shown in the display area of the transformer.

Input Region Names

The Page Break transformer has two input regions.

Input 1 (Heading) contains an optional page header. Input 1 is treated as a series of rows and columns; it can contain any type of data, and is limited only by the amount of workstation memory available.

Input 2 (Data) contains data read from a Spreadsheet tool, Query tool, SQL Entry icon, or any other row-formatted and column-formatted data. Input 2 can have any number of optional header rows. This input region does not have a size limit.

Output Region Names

The Page Break transformer has one output region called Results (Output 1), which contains the data read from Input 2 with optional page breaks and page header inserted. Output 1 is limited in size by the amount of disk space available on the file server.

Example

For this example, this is the data in Input 2 of the Page Break transformer:

Market	Product	Period	Volume	Revenue
Min	A	1	947	78,975,856.00
Min	A	2	764	55,467,458.00
Min	A	3	644	53,674,568.00
Min	A	4	575	45,634,562.00
Min	B	1	2700	234,756,458.00
Min	B	2	59	54,778.00
Min	B	3	12	4,356.00
Min	B	4	36	24,678.00

The following report is generated in Output 1 after the transformer has finished running.

The third section has only two rows of data. The number of input data rows was not evenly divisible by the number of rows of data to write on each page (three in this case). Consequently, the transformer created two pages with three rows and filled the last page with the remaining two rows.

Market	Product	Period	Volume	Revenue
Min	A	1	947	78,975,856.00
Min	A	2	764	55,467,458.00
Min	A	3	644	53,674,568.00
Market	Product	Period	Volume	Revenue
Min	A	4	575	45,634,562.00
Min	B	1	2700	234,756,458.00
Min	B	2	59	54,778.00
Market	Product	Period	Volume	Revenue
Min	B	3	12	4,356.00
Min	B	4	36	24,678.00

Period Table

The Period Table transformer generates period and date tables for a variety of applications. This transformer is especially suited for:

- Providing period input for the time-series analysis transformers, including Forecast and Seasonality transformers
- Constructing database period tables

The Period Table transformer generates the following information and writes it to the transformer's Results output:

- Meta5 Date (see “How Meta5 Formats Dates” on page 161 for details)
- Numeric date
- Alternate format date
- Period sequence with a year number
- Sequence within data set number
- Trading or week (or normal business) days
- Day of week

Parameters

Starting date, ending date (-28; 6242; “May 1, 1987;) [Note: January 1, 1970 = 0]

This parameter specifies the first and last date that should be included in the Output 1. The starting date is required, and the ending date is optional. The `Number of dates to generate` parameter can also be used to specify when the table should end. The starting and ending dates must be two numeric or Meta5 dates separated by a comma. Dates must be enclosed in double quotation marks in a Transformer Controls window parameter. For example, `8401,9100`, `"January 1, 1993"`, and `"December 1, 1994"` are valid entries.

Resolution of dates (Day; Week; QuadWeek; Month; Quarter; Year; EvenBiMonth; OddBiMonth)

This parameter specifies the format of the Meta5 date column in Output 1. This parameter also controls the number of days between successive rows in Output 1. The user can choose one of the following date resolutions: Day, Week, QuadWeek, Month, Quarter, Year, EvenBiMonth, and OddBiMonth; spacing and capitalization are not significant. Day is the default value for this parameter.

Dates before January 1, 1970 are not guaranteed to work properly with the Week and QuadWeek format in all applications. It is important to check your results. Also, the Week and QuadWeek formats cannot be edited as dates in Spreadsheet icons.

Number of dates to generate (1; 52; 144;)

This parameter specifies the number of rows in the date table. This parameter can be used if the Ending Date is not specified. Any non-negative whole number can be specified. This parameter is ignored if the `Ending Date` parameter is specified.

Output titles on period table? (y; n)

This parameter specifies whether the column titles should be included in Output 1. `Yes` or `y` specifies that column titles are to be placed in Output 1. `No` or `n` indicates that column titles should not be added to Output 1. The default is `n`.

Output Meta5 date format? (y; n)

This parameter specifies whether a column containing dates in the Meta5 date format should be included in Output 1. `Yes` or `y` specifies that the Meta5 date column is to be placed in Output 1. `No` or `n` indicates that it should not be. The default is `n`.

Output numerical date format? (y; n)

This parameter specifies whether a column containing dates in the Numeric date format should be included in the date table. `Yes` or `y` specifies that the numeric date column is to be placed in the date table. `No` or `n` indicates that it should not be. The default is `n`.

January 1, 1970 is 0.

Output date in an alternate format? (y; n) resolution (Day; Week; QuadWeek;)

This parameter specifies whether a column containing dates in an alternate date format should be included in Output 1. `Yes` or `y` specifies that the alternate date column is to be placed in Output 1. `No` or `n` indicates that it should not be. The default is `n`.

In addition, the user must specify the date format in which the dates are to be displayed. Any of the `Resolution of dates` parameter choices are valid. Additional alternate choices include `mmddyy`, `mmddyyyy`, `ddmmyy`, and `ddmmyyyy`. For example, if a quarterly data set is created, the user can view the dates in Day resolution by entering `yes, Day` in this parameter. The default resolution is Day.

Output period number? (y; n) starting period number (1; 5;)

This parameter specifies whether a column containing period numbers within a year should be included in Output 1. If the periods are more than a year, the numbers start at 1. `Yes` or `y` specifies that a period within a year number column is to be placed in Output 1. `No` or `n` indicates that it should not be. The default is `n`.

In addition, the user can specify the beginning period number if the default value of 1 is not desired (the two values must be separated by a comma). For example, if you specify `y,2` for this parameter, the transformer will create a column in the Output 1 that contains period numbers. The first date will have a period number of 2, the second date a period number of 3, and so on.

Output sequence number? (y; n) starting sequence number (1; 5;)

This parameter specifies whether a column containing sequence in the data set numbers should be included in Output 1. `Yes` or `y` specifies that a sequence number column is to be placed in Output 1. `No` or `n` indicates that it should not be. The default is `n`. In addition, the user can specify the beginning sequence number if the default value of 1 is not desired (the two values must be separated by a comma). For example, if you specify `y,13`, the transformer creates a column in the resulting table that contains sequence numbers. The first date has a sequence number of 13, the second date has a sequence number of 14, and so on.

Output trading days? (y; n) default number of trading days (1; 5;)

This parameter specifies whether a column containing the number of trading or normal business days in each period should be included in Output 1. `Yes` or `y` specifies that a trading day column is to be placed in the date table. `No` or `n` indicates that it should not be. The default is `n`. In addition, the user can specify the number of trading days if the default value is not desired (the two values must be separated by a comma). The default value is based upon the `Resolution Of Dates` value, with the value being about five-sevenths the number of days between dates implied by the `Resolution Of Dates` value. For example, if weekly dates are desired, the default number of trading days will be 5, which is five-sevenths of 7 days per week. Holidays are not considered in this calculation.

Output day of week? (y; n)

This parameter specifies whether a column containing the day of week indicated by the Meta5 date should be included in Output 1. `Yes` or `y` specifies that a day of week column is to be placed in Output 1. `No` or `n` indicates that it should not be. The default is `n`.

Date offset (0; 10; 25; End; -1; Sunday; Friday;)

This parameter specifies the offset of the alternate date from the Meta5 date. The default value of 0 displays alternate date of the beginning day of the Meta5 date period. Notice that the `Resolution of dates` parameter defines how the offset will function. An offset of -1 for a month resolution returns the last day of the period minus 1 day. An offset of -1 for a day resolution returns the Meta5 date minus one day.

Add leading zeros to the non-Meta5 alternate date formats? (y; n)

This parameter specifies whether leading zeros should be included in the dates displayed by a non-Meta5 alternate date format. `Yes` or `y` specifies that leading zeros are to be placed in Input 1. `No` or `n` indicates that they should not be. For example, May 7, 1990 might be displayed as 05-07-1990 with leading zeros, or as 5-7-1990 without leading zeros. The default is `n`.

Date format character (–; /; .;)

This parameter allows the user to specify the character separating the numbers in the `mmddyy`, `ddmmyy`, `mmddyyyy`, and `ddmmyyyy` formats. Any printable character can be specified; a hyphen (-) is the default value. For example, May 7, 1990 is displayed as `5-7-1990` when using the default value, as `5/7/1990` if the slash (/) is specified, and as `5.7.1990` if the period (.) is specified.

Table 23. Date offset examples

Offset value	Resolution of dates	Meta5 date	Alternate date	Day of week
-1	Month	January, 1993	January 30, 1993	Saturday
0	Month	January, 1993	January 1, 1993	Friday
5	Month	January, 1993	January 5, 1993	Tuesday
End	Month	January, 1993	January 31, 1993	Sunday
Monday	Month	January, 1993	January 2, 1993	Saturday
-1	Day	January, 1993	December 31, 1992	Thursday
0	Day	January, 1993	January 1, 1993	Friday

Table 23. Date offset examples

Offset value	Resolution of dates	Meta5 date	Alternate date	Day of week
5	Day	January, 1993	January 5, 1993	Tuesday
End	Day	January, 1993	January 1, 1993	Friday
Monday	Day	January, 1993	January 2, 1993	Saturday

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Results (Output 1)
- Messages (Output 2)

When you select either of these choices, the data associated with that choice is shown in the display area of the transformer.

Input Region Names

The Period Table transformer has no input regions.

Output Region Names

The Period Table transformer has two output regions called Results (Output 1) and Messages (Output 2). Output 1 can contain the data columns shown in Table 24.

Table 24. Output 1 data columns

Data column	Contents
Meta5 Date	The date printed in a format determined by the `Resolution of dates` parameter.
Numeric Date	The number value corresponding to the Meta5 date.
Alternate Date	The date printed in a format by its own resolution.
Period Number	Identifies a particular period within a year, for example, January will always be period 1 in 1993, 1994, 1995; period numbers are always reset to 1 after the maximum number of periods in a year is reached.
Sequence Number	A running count of the number of rows in Output 1.
Trading Days	The number of business days in a given period; a default value is specified in `Output trading days` parameter.
Day Of Week	The day of the week, such as Sunday.

If multiple parameters are specified, they are provided in the order in which they are displayed in Table 24 on page 159.

Output 2 is the message output region. It provides run time error messages and the date and time at which the transformer ran.

Example

To generate sample data, the Period Table transformer parameters are set as follows:

Starting date, ending date
"June 1, 1993"

Resolution of dates
Month

Number of dates to generate
4

Output titles on period table?
Yes

Output Meta5 data format?
Yes

Output numeric data format?
Yes

Output date in an alternate format?
Yes, Day

Output period number, starting period number
Yes

Output sequence number, starting sequence number
Yes

After the transformer runs, the data in Table 25 is displayed in Output 1:

Table 25. Output from the Period Table transformer

Meta5 Date	Numeric Date	Alternate Date	Period number	Sequence Number
June, 1993	8552	June 1, 1993	1	1
July, 1993	8582	July 1, 1993	2	2
August, 1993	8613	August 1, 1993	3	3
September, 1993	8644	September 1, 1993	4	4

The transformer created five output columns containing the five specified date fields for four separate dates.

How Meta5 Formats Dates

Meta5 expresses dates in a variety of formats. Each date has an associated resolution value that depends upon the time that elapses between successive dates. Supported resolutions are shown in Table 26.

Table 26. Meta5 Date formats

Date format	Example
Day	Day of the month.
Week	Seven days, starting on a Sunday.
QuadWeek	Exactly 4 weeks, starting on a Sunday.
Month	Calendar month
EvenBiMonth	Two months, such as Jan/Feb
OddBiMonth	Two months, such as Dec/Jan
Quarter	Three months, such as Jan/Feb/Mar
Year	Calendar year

When printed, a date is formatted to provide the information implied by its resolution value. For example, July 1, 1993, is displayed as follows:

Day	July 1, 1993
Week	week of June 27, 1993
QuadWeek	quad week of June 20, 1993
Month	July, 1993
EvenBiMonth	JA, 1993
OddBiMonth	JJ, 1993
Quarter	3Q93
Year	1993

Each day is assigned a number called its numeric date format. This number is the number of days since, or prior to, January 1, 1970. Thus, January 1, 1970 is date 0, and July 1, 1993 is date 8611. For backward compatibility, a date before 0 is a negative number whose absolute value expresses the number of days until January 1, 1970. The numeric date allows for accurate date calculations, especially in a database or in an SQL program.

Date Resolution

The resolution of a date implicitly defines the number of days between adjacent dates. Table 27 documents the number of days between dates, the number of periods per year, and the default number of trading days per period. The default number of trading days can change in the `Output trading days` parameter.

Table 27. Period Table defaults

Resolution	Number of days per period	Number of periods per year	Default number of trading days
Day	1	365-366	1
Week	7	52-53	5
QuadWeek	28	13	20
Month	28-31	12	22
EvenBiMonth	˜61	6	44
OddBiMonth	˜61	6	44
Quarter	˜92	4	66
Year	365-366	1	255

Alternate Date Formats

Alternate date formats provide a method for displaying dates in a format other than that implied by their resolution value. For example, an alternate date format allows the dates of a quarterly date table to be displayed in Day resolution. Any Meta5 date can be displayed in an alternate date format.

In addition to Meta5 date formats, alternate date formats provide several other date formats that are not directly supported by the desktop tools, but are often desirable for period tables and printed reports. These date formats are named after the sequence of digits that comprise the date representation. The characters m, d, and y refer to month, day, and year, respectively. Two characters indicate that two digits can be included in the number value, four characters indicate that four digits can be included in the number value. The separation character (the hyphen) and leading zeros are optional and can be specified in the last two parameters of the Programs Control window.

The following list shows alternate date formats using June 21, 1993, as an example.

mmddyy
: 062193

mmddyyyy
: 06211993

ddmmyy
210693

ddmmyyyy
21061993

yymm
9306

yymmdd
930621

Dates that seem identical cannot be equal if the resolution change is to a format that has fewer periods per year. For example, if an alternate resolution of Quarter is selected for two adjacent dates in a monthly date table, such as November 1990 and December 1990, these two dates will both be displayed as 4Q90. Even though the dates might look identical, the number of days since January 1, 1970, will be different; thus, the dates are not equal.

Post Message

The Post Message transformer automatically constructs messages within a capsule application and sends them to the Meta5 desktop.

The Post Message parameters are compatible with @-variables and thus allow the content of messages to vary accordingly with the outcome of a capsule application run.

The Post Message transformer sends text string messages to the desktop message area. The text string displays for three seconds while the transformer is running. Program names and copyright information are generally displayed in this fashion. If all message parameters are empty, the transformer runs without taking any action.

Parameters

The Post Message transformer has 10 parameters, `Parameter #1` through `Parameter #10`. These parameters are the input areas for the message. The input parameter values can be text strings or numbers; other data types are not supported. Numbers are shown with two decimal places of precision. Each value can contain up to 99 characters. Only one value is allowed per parameter. Because commas and semicolons are interpreted as value separators, values containing commas or semicolons should be placed in double quotation marks.

Region Controls

The Post Message transformer has no input or output regions.

Example

In this example, the Post Message transformer sends a copyright message to the desktop message area whenever a specific capsule application is run:

1. Copy the Post Message transformer into the upper left corner of the Capsule window.
2. Click on the `Show Controls` button in the Capsule window header and set @A to 2001.
3. Set the Post Message transformer parameters as follows:

   Parameter #1
   Copyright

   Parameter #2
   "@A, "

   Parameter #3
   ABC

   Parameter #4
   Computer

   Parameter #5
   Systems

Whenever the example capsule application is run, the string `Copyright 2001, ABC Computer Systems` is displayed briefly in the message area. Because the Post Message transformer was copied into the upper left corner of the Capsule window, the transformer runs first, and the message is displayed immediately after the `Run` button is clicked.

Random Number

The Random Number transformer generates sets of random numbers. Random numbers are useful in testing numeric and statistical applications, and as control groups in data analysis. When used with the Period Table transformer, the Random Number transformer is also useful as a quick means of generating sample data.

The results generated by the Random Number transformer are computed by a pseudo-random algorithm. Each random number is computed by applying the pseudo-random formula to the previous random number. The first random number is computed from a seed value, with each possible seed value generating a unique sequence of random numbers. The computed values are real numbers with two digits to the right of the decimal place.

The advantage of using a seed-driven pseudo-random algorithm is repeatability. Given the same seed value, the Random Number transformer will produce the same table of output. In contrast, the Spreadsheet icon changes the value of its random numbers every time the spreadsheet recalculates. Repeatability allows examples and simulations to be rerun with the same set of input values, allowing external variables to be adjusted to determine the effect of a variety of alternate actions.

If the seed value is not specified, the Random Number transformer fabricates a random seed value based upon the current time on the system clock. Thus, a different set of random numbers is generated each time the program is run. The seed value parameter can also be set using an @-variable. Thus, the process of changing the seed value on selected transformer runs can be automated.

Parameters

Each Random Number transformer parameter expects an integer value.

Number of rows to generate (1; 2; 5;)
This parameter specifies the number of rows of data to generate for Output 1.

Number of columns to generate (1; 2; 5;)
This parameter specifies the width of the data set excluding the optional sequence column.

Minimum value (0; 1; -55;)
This parameter is the lower limit of the random numbers.

Maximum value (1; 100; 1000;)
This parameter is the upper limit of the random numbers.

Sequence column? (; y; n)
This parameter enables or disables the inclusion of a sequence column in the table. A column of sequence numbers is placed in column A if *y* or *yes* is entered. If *n* or *no* is entered (the default), no sequence column is included in the table.

Seed value for random number generator (; 1; 1988;) [Blank = random seed]
This parameter is the number used to calculate the first random number in the table. Each seed value generates a unique series of random numbers. If no seed value is specified, a random seed will be generated using the system clock.

Region Controls

The `Display Data For` field in the Transformer Controls window contains one choice, Results (Output 1).

The data is shown in this display area of the transformer.

Input Region Names

The Random Number transformer has no input region.

Output Region Names

The Random Number transformer has one output region. Output 1 contains the set of random numbers computed by the transformer. The size of Output 1 is controlled by the `Number of rows to generate` and `number of columns to generate` parameters.

Example

The Random Number transformer parameters are set as follows:

Number of rows to generate
4

Number of columns to generate
3

Minimum value
0

Maximum value
100

Sequence column?
y

Seed value for random number generator
1

The following table of data is available in Output 1 after the Random Number transformer finishes running.

The Random Number transformer created three columns of random data and one column that contains sequence numbers. All of the values are greater than the specified minimum value of 0 and less than the specified maximum of 100. If the seed value was not specified, a different set of numbers would be generated each time the transformer was run.

A	B	C	D
1	51.39	17.57	30.86
2	53.45	94.76	17.17
3	70.22	22.64	49.48
4	12.47	8.39	38.96

Replace

The Replace transformer substitutes for non-data values, numeric values that are over or under a specified amount, or extra blank spaces in a data set.

Replace

The Replace transformer selectively recodes data before it is presented as a report or loaded into a database.

The most common use of the Replace transformer is to replace missing or bad data cells with an appropriate substitute value. For example, if a data set is to be loaded into a database, all N/A and Error cells must be removed from a numeric column. One strategy is to replace all such cells with a 0 to indicate a missing value. Another strategy is to define a flag value to stand in for an N/A cell. A value that never occurs in the input data, such as -99,999,999,999.0, could be used as an N/A flag. This flag provides SQL applications a convenient test for missing data; it also enables Meta5 data types to be manipulated directly from SQL.

Another common application for the Replace transformer is the conversion of dates read from a database into one of the Meta5 date formats. Dates are commonly stored in a database as four-digit integer values. (An integer date requires much less storage space than a text string date.) The Replace transformer can convert dates to numbers and numbers to dates. You can also convert from one date format to another, changing, for example, Quarterly resolution dates to their corresponding Day format.

Exception reports can also be formatted using Replace. The Replace transformer can replace values above and below preset limits with a text string or any other value, allowing messages to be substituted for values outside a preset limit, or values within a preset limit to be blanked out. Exceptions are much easier to locate and analyze when they are highlighted.

Additional uses include clipping data (resetting values that are too high or too low), removing excess white space from text, and converting data to Boolean format (logical true or false). The Replace transformer includes two options for expanding or truncating data sets. These options are useful when irregularly shaped sets are to be saved in a database table, such as output from the Regression or Forecast transformers.

Parameters

The Replace transformer has 19 parameters. Each parameter is optional, and each option is disabled if its parameter is empty. The parameters are grouped here by type:

- Input and Output Region parameters
- Replacement Control parameters
- White Space Removal parameters
- Date Conversion parameters

Input and Output Region Parameters

This section contains descriptions of the input and output region parameters. Input and output region parameters control how the input data is treated, and the size and shape of the output data. If any of these parameters are blank, that option is disabled.

Number of header rows (0; 1;)

This parameter specifies the number of header rows to be read and copied to Output 1 without modification. The default value is 0.

Columns containing values to replace (; all; none; a; a,b,c; b:f;) [leave blank for all]

This parameter specifies the columns to be checked to see if any cell data requires replacement. This parameter expects a series of one- or two-letter column names, each separated by a comma. The word `all` can be entered if all columns should be checked. For example, `b,d,h` specifies that columns B, D, and H should be scanned for possible cells in need of data replacement. The default value is `all`.

Columns to leave unchanged (; a; a,b,c; b:f;) [leave blank for none]

This parameter is used when `all` is specified in `Columns containing values to replace` field. For example, if all columns except column C are to be checked for possible replacements, specify `c` for this parameter. The default value is none.

Maximum number of columns in output (1; 5;) [leave blank for no maximum]

This parameter truncates rows with more than the specified number of columns. Any whole number value can be specified. The default is 0.

Extend rows with fewer than (; 1,; 5,;) columns, fill end of row with (; NA; 0; 1; Error; Empty; –; *;)**

This parameter extends rows with fewer than the specified number of columns to the maximum number of columns specified. Any whole number value can be specified. If this parameter is blank, no rows will be extended. In addition, this parameter allows the user to specify the contents of these new cells. Any data following the first comma is placed in each added cell. If the replacement value is not specified, the added cells will be blank.

Replacement Control Parameters

Each replacement control parameter defines a value to be inserted in place of any data that meets a particular criterion. Any valid Meta5 data type can be specified as a replacement value. The text strings `N/A`, `Error`, and `Empty` can be used to set a value to N/A, Error, and blank, respectively. Text values do not require quotation marks around them. If any of these parameters are blank, the corresponding replacement will not be performed.

Any number with more than eight digits must contain a decimal point. For example, 999,999,999 would be entered as 999999999.0.

Replace '=N/A' with (; NA; 0; 1; Error; Empty; –; *;)**

This parameter specifies the data that will be displayed in all cells containing N/A values.

Replace '=0.0' with (; NA; 0; 1; Error; Empty; –; *;)**

This parameter specifies the data that will be displayed in all cells containing either 0 or a formula that equals 0.

Replace '=Error' with (; NA; 0; 1; Error; Empty; –; *;)**

This parameter specifies the data that will be displayed in all cells containing the value of Error.

Replace Empty Cells with (; NA; 0; 1; Error; Empty; –; *;)**

This parameter specifies the data that will be displayed in all empty cells. An empty cell is any cell that is blank and lies before an occupied cell in the row.

Replace '.' with (; NA; 0; 1; Error; Empty; –; *;)**

This parameter specifies the data that will be displayed in all empty text-formatted cells.

Replace '=False' with (; NA; 0; 1; Error; Empty; –; *;)**

This parameter specifies the data that will be displayed in all cells with Boolean false values.

Replace '=True' with (; NA; 0; 1; Error; Empty; –; *;)**

This parameter specifies the data that will be displayed in all cells with Boolean true values.

Replace values below (; -999999999,; 0;) with (; NA; 0; 1; Error; Empty; –; *;)**

This parameter requires two values. The first value is the limit value. All numeric or formula cells with a value below this limit will be replaced. The second value specifies the replacement value.

Replace values above (; 999999999,; 0;) with (; NA; 0; 1; Error; Empty; –; *;)**

This parameter requires two values. The first value is the limit value. All numeric or formula cells with a value above this limit will be replaced. The second value specifies the replacement value.

White Space Removal Parameters

Columns to remove excess space from (; a; a,b,c; b:f;) [leave blank for none]

This parameter specifies the columns that should have any excess white space removed. To remove excess white space, all sequences of two or more white space characters (spaces or tabs) are compressed into a single space. Only cells containing text data will be altered. This parameter expects a list of one- or two-letter column names, each separated by a comma. For example, `a,b,c` specifies that columns A, B, and C are to be checked for excess white space. If this parameter is blank, no white space removal will be performed.

Remove leading space? (y; n)

This parameter specifies whether white space that occurs at the beginning of text data should be removed. If `y` or `yes` is specified, leading white space will be removed from cells checked as a result of being specified as `Columns to remove excess white space from`. Excess leading white space will remain unaltered if `n` or `no` is specified (the default).

Date Conversion Parameters

Columns containing dates (; a; a,b,c; b:f;) [leave blank for none]

This parameter specifies the columns containing dates that are to be replaced. This parameter expects a list of one- or two-letter column names, each separated by a comma. For example, `a,b,c` specifies that columns A, B, and C are to be checked for possible date replacement or conversion. If this parameter is blank, no date conversions or replacements will be performed.

Change date resolution to (; Numeric; Day; Week; QuadWeek; Month; Quarter; Year; EvenBiMonth; OddBiMonth)

This parameter specifies the date format that is to be for all dates located in the `Columns containing dates`. Valid choices are numeric, day, week, quadweek, month, quarter, year, evenbimonth, and oddbimonth. The default is day resolution. Because capitalization and spacing are not significant, Day and day are equally valid choices.

Replace Illegal Dates with (; NA; 0; 1; Error; Empty; –; *;)**

This parameter specifies the data that will be displayed in all cells that do not contain a valid Meta5 or numeric date.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- Results (Output 1)

When you select either of these choices, the data associated with that choice is shown in the display area of the transformer.

Replace Transformer Input Region Names

The Replace transformer has one input region called Data (Input 1). Input 1 can be in any format, with any number of rows or columns. Because only one row is processed at a time, there is no limit to the size of the input data.

Replace Transformer Output Region Names

The Replace transformer has one output region Results (Output 1). Output 1 holds the data found in Input 1, with selected cells changed as requested.

Example

The following data shows shipments volume for the ABC Company, including market share.

	Total Volume	Total Volume	Market	Market Share	Exception
Date	ABC Company	USA	Share	Performance	East
8401	2,100.66	11,002.93	19.09	99,999,999,999.00	0.00
8432	1,208.62	7,038.33	17.17	(99,999,999,999.00)	0.00
8460	1,272.26	7,321.18	17.38	(99,999,999,999.00)	0.00
8491	619.33	4,419.23	14.01	(99,999,999,999.00)	12.80
8521	1,572.89	8,657.28	18.17	0.00	0.00
8552	1,325.08	7,555.93	17.54	0.00	0.00
8582	2,254.47	11,686.54	19.29	99,999,999,999.00	0.00
8613	1,326.87	7,563.86	17.54	0.00	10.97
8644	2,370.82	12,203.65	19.43	99,999,999,999.00	0.00
8674	1,867.80	9,967.99	18.74	0.00	0.00
8705	1,948.41	10,326.27	18.87	0.00	14.89
8735	1,234.72	7,154.30	17.26	(99,999,999,999.00)	0.00

The Replace transformer parameters are set as follows:

Number of header rows
3

Columns containing values to replace
e,f

Replace '=0.0' with
Empty

Replace values below N with
–99999999998.0,POOR

Replace values above N with
99999999998.0,GOOD

Columns containing dates
A

Change date resolution to
Month

After the transformer runs, the following information is in Output 1.

	Total Volume	Total Volume	Market	Market Share	Exception:
Date	ABC Company	USA	Share	Performance	East
January, 1993	2,100.66	11,002.93	19.09	GOOD	
February, 1993	1,208.62	7,038.33	17.17	POOR	
March, 1993	1,272.26	7,321.18	17.38	POOR	
April, 1993	619.33	4,419.23	14.01	POOR	12.80
May,1993	1,572.89	8,657.28	18.17		
June, 1993	1,325.08	7,555.93	17.54		
July, 1993	2,254.47	11,686.54	19.29	GOOD	
August, 1993	1,326.87	7,563.86	17.54		10.97
September, 1993	2,370.82	12,203.65	19.43	GOOD	
October, 1993	1,867.80	9,967.99	18.74		
November, 1993	1,948.41	10,326.27	18.87		14.89
December, 1993	1,234.72	7,154.30	17.26	POOR	

The contents of columns B, C, and D were moved without modification to the output, because those three columns were not specified on the `Columns containing values to replace` parameter.

The numeric date fields in column A were reformatted into the defined date resolution of month. Also, all of the cells in columns E and F that had values of 0 were changed to empty cells. Finally, any of the values in column E that were less than -99,999,999,998.0 were changed to the value POOR while all of the values greater than 99,999,999,998.0 were changed to GOOD.

Row Clean

The Row Clean transformer selectively removes rows of data from a column- and row-formatted data source. The Row Clean transformer works much like the Clean transformer, but it provides more flexibility in the definition of nonvalid values and how the cleaning rules are evaluated. The Row Clean transformer also removes rows that contain a string.

Row Clean

The clean operation removes rows of data that contain non-data values. These non-data values can be missing data values (N/A), nonvalid computations (Error), or white-space cells inserted by an application to improve the appearance of a report. You can remove these data rows before analyzing or presenting data sets.

When more than one column is cleaned, the decision to remove a row is based on one of two types of logic. A row is removed if a cell in any of the columns under consideration has a non-data value, or if the cells in all columns under consideration contain a non-data value. These two methods correspond to OR logic and AND logic, respectively. The Row Clean transformer can perform the clean operation based on any combination of these two logic types.

The default clean operation considers a cell to be blank (contain a non-data value) if any of the following criteria are met:

- The cell is blank
- Contains N/A
- Contains Error
- Contains a zero value
- Blank spreadsheet text cell (contains a period)

These defaults are the same as the Clean transformer defaults.

In addition to the default values, you can configure the Row Clean transformer for other criteria, such as negative numbers or false.

In some applications, the data that is removed is as interesting as the data that is retained. The Row Clean transformer allows you to display only the data that would normally be removed. This option is useful for analyzing missing values in a data set.

Parameters

The Row Clean Transformer Controls window contains 15 optional parameters.

Number of header rows in data (0; 1;)

This parameter specifies the number of header rows in the Input 1 data that will be copied to Output 1 unchanged. Any whole number can be specified; the default value is 0.

Remove row if any of these columns are 'blank' (a; a,b,c;)

This parameter specifies the columns to be checked for blank values. A row is removed if any of the columns specified contain a blank value. A comma-separated list of one- or two-letter column names can be entered. For example, if `d,e,h` is entered, all rows that contain a blank value in column D, column E, or column H are removed. The default is that no columns are checked. This parameter uses OR logic to clean rows.

Remove row if all of these columns are 'blank' (a; a,b,c;)

This parameter specifies the columns to be checked for blank values. A row will be removed only if all of the columns specified contain a blank value. A comma-separated list of one or two letter column names can be entered. For example, if `d,e,h` is entered, all rows that contain a blank value in column D, column E, and column H are removed. The default is that no columns are checked. This parameter uses AND logic to clean rows.

Remove row if all columns are 'blank'? (y; n)

This parameter specifies whether to remove rows in which all cells are blank. `yes` or `y` removes blank rows. `no` or `n` removes blank rows only if the row meets the removal criteria of one of the two parameters above. The default value is `no`.

Use default definition of 'blank'? (y; n)

This parameter specifies whether the Row Clean transformer should use the default definition of blank, or if the Row Clean transformer should use the definition of blank as specified in the next eight parameters. `yes` or `y` (the default parameter setting) specifies that the default definition of blank should be used. `no` or `n` allows you to specify the definition of blank.

Consider '= N/A' cells as 'blank'? (y; n)

This parameter specifies whether cells containing N/A should be interpreted as blank. `yes` or `y` adds N/A to the definition of blank. `no` or `n` does not.

Consider '= Error' cells as 'blank'? (y; n)

This parameter specifies whether cells containing Error should be interpreted as being blank. `yes` or `y` adds Error to the definition of blank; `no` or `n` does not. The cell value must be the result of a Spreadsheet formula. Text or numeric values will not be recognized. See the parameter `Text value to consider as 'blank'` for the method to specify text values to be considered blank.

Consider '= 0.0' cells as 'blank'? (y; n)

This parameter specifies whether cells containing 0 as a formula or number value should be interpreted as being blank. `yes` or `y` adds 0 to the definition of blank; `no` or `n` does not. The cell value must be the result of a Spreadsheet formula or numeric values. Text will not be recognized. See the parameter `Text value to consider as 'blank'` for the method to specify text values to be considered blank.

Consider '.' cells as 'blank'? (y; n)

This parameter specifies whether cells containing a period (a Spreadsheet text cell with no characters) should be interpreted as being blank. `yes` or `y` adds a period (.) to the definition of blank; `no` or `n` does not.

Consider '= False' cells as 'blank'? (y; n)

This parameter specifies whether cells containing False should be interpreted as being blank. `yes` or `y` adds data value of False to the definition of blank; `no` or `n` does not. The cell value must be the result of a Spreadsheet formula. Text or numeric values will not be recognized. See the parameter `Text value to consider as 'blank'` for the method to specify text values to be considered blank.

Consider cells with negative numbers as 'blank'? (y; n)

This parameter specifies whether cells containing numeric values less than 0 should be interpreted as being blank. `yes` or `y` adds negative numbers to the definition of blank; `no` or `n` does not.

Consider empty cells as 'blank'? (y; n)

This parameter specifies whether empty cells (blank but before the last cell in a row) should be interpreted as being blank. `yes` or `y` adds empty cell (meaning that an empty cell will not be treated as a blank) to the definition of blank; `no` or `n` does not.

Consider cells beyond the end of the row as 'blank'? (y; n)

This parameter specifies whether blank cells beyond the last valid cell in a row should be interpreted as being blank. `yes` or `y` specifies that blank cells beyond the last valid cell in a row should be interpreted as being blank; `no` or `n` does not.

It is difficult to tell the difference between a cell that is blank and a cell that is beyond the end of a row, particularly for blank rows. The parameter dealing with empty cells has two parameters (empty versus beyond end of the row) for cases when this distinction proves important. Generally, these two parameters should be answered the same way.

For example, if the response to `Consider empty cells as 'blank'` is `y`, the response to the `Consider cells beyond the end of the row as 'blank'` parameter should also be `y`.

Show cleaned data ('GOOD') or removed data ('BAD') in 'Output 1' (g; b; good; bad)

This parameter specifies whether the data without rows containing blank cells (the good or cleaned data) should be placed in Output 1, or if the rows containing blank values (the bad data that was removed) instead. `good` or `g` specifies that the good data should be displayed. `bad` or `b` specifies that the bad data should be displayed. The default value is `good`.

Text value to consider as 'blank'? (; -; na; N/A; err;)

This parameter allows the user to specify a text value that will be considered blank. This parameter allows for non-spreadsheet, non-data values such as `na` from a database query. The default is no value.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)

- Results (Output 1)

When you select either of these choices, the data associated with that choice is shown in the display area of the transformer.

Input Region Names

The Row Clean transformer has one input region called Data (Input 1), which contains any row- and column-formatted data. This input does not have a size limit, nor is it required to be a fully populated rectangular region.

Output Region Names

The Row Clean transformer has one output region called Results (Output 1), which contains the data read from Input 1 minus the rows that meet a particular set of specifications. The Output 1 region does not have a size limit.

Example

In this example, the following data is sent to Input 1 of the Row Clean transformer:

Market	Product	Transaction Key	Volume	Revenue
NY	A	83973987	=N/A	32,453,452.00
NY	A	93849179	34.00	23,462,364.00
NY	A	92837978	352.00	345,234.00
NY	A	98294801	342.00	2,436,326.00
NY	B	92749278	0.00	0.00
NY	B	98394278	34.00	233,546,235.00
NY	B	92389874	54.00	=N/A
NY	B	92349387	12.00	46.00

The Row Clean transformer parameters are set as follows:

Number of header rows in data
1

Remove row if any of these columns are 'blank'
d,e

Remove row if all columns are 'blank'?
y

Use default definition of 'blank'?
n

Consider '= N/A' cells as 'blank'?

y

Consider '= Error' cells as 'blank'?

y

Consider '= 0.0' cells as 'blank'?

n

Consider '.' cells as 'blank'?

n

Consider '= False' cells as 'blank'?

n

Consider cells with negative numbers as 'blank'?

n

Consider empty cells as 'blank'?

y

Consider cells beyond the end of the row as 'blank'?

y

Show cleaned data ('GOOD') or removed data ('BAD') in 'Output 1'

g

When processed by Row Clean, the following data is displayed in Output 1:

The output data created by the Row Clean transformer has two fewer rows of data than the input data. Two rows that contained N/A values in columns D or E were removed. The remaining data is unaltered. The row containing zero values was not removed, because the transformer applied the user definition of a blank cell.

Market	Product	Transaction Key	Volume	Revenue
NY	A	93849179	34.00	23,462,364.00
NY	A	92837978	352.00	345,234.00
NY	A	98294801	342.00	2,436,326.00
NY	B	92749278	0.00	0.00
NY	B	98394278	34.00	233,546,235.00
NY	B	92349387	12.00	46.00

Row Select

The Row Select transformer allows you to select data rows based on their location in the input data. A selection rule language is provided to help build complex selection rules. You can insert blank lines, page breaks (Text icon output only), and page numbering into the output.

Parameters

The Row Select transformer has four parameters, all of which are optional.

Number of header rows (0; 1;)

This parameter specifies the number of header rows at the beginning of the input data that can be copied without change to the output. The value can be 0 or any positive number; the default is 0.

Copy header rows to output (y; n)

This parameter specifies whether to copy the header rows to Output 1 before the selected rows are copied to Output 1. This parameter's functionality can be duplicated with a selection pattern. `Yes` or `y` causes the header to be copied. `no` or `n` causes the header to be ignored. The default is `no`.

Columns to include in output (all; a; a,b;)

This parameter specifies the columns in Input 1 that should be copied to Output 1. If `all` is entered, every column in the input is copied to the output. Otherwise, the specified columns are copied to Output 1 in the order in which the they are entered. To specify a column name, enter the one- or two-letter name of the column as it is displayed in Input 1. Separate column names with commas. For example, `a,c,b` specifies that only columns A, C, and B are to be displayed in Output 1, in that order.

Setting `Columns To Include In Output` to `all` slightly improves the speed of the transformer on large data sets. The default value is `all`.

Selection pattern specification

This parameter is a list of pattern rules. (See Table 28 on page 182 for details about pattern definitions.) Entries in this parameter must be separated by commas and can be as wide as the maximum parameter stream.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data 1 (Input 1)
- Results (Output 2)

When you select either of these choices, the data associated with that choice is shown in the display area of the transformer.

Input Region Names

The Row Select transformer has one input region called Data (Input 1). Input 1 can be any type of row-formatted data. Because the Row Select transformer must be able to access rows randomly, the entire input region must be in workstation memory. Having a small amount of workstation memory will limit the size of the input.

Output Region Names

The Row Select transformer has one output region called Results (Output 1). Output 1 depends upon the selection rules, described in Table 28 on page 182. The output region is not limited in size.

Example

The following data is contained in a spreadsheet that is connected to Input 1 of the Row Select transformer.

Market	Product	Period	Volume	Revenue
NY	Diet Cola 6 pk	1	234.00	32,453,452.00
NY	Diet Cola 6 pk	2	34.00	23,462,364.00
NY	Diet Cola 6 pk	3	352.00	345,234.00
NY	Diet Cola 6 pk	4	342.00	2,436,326.00
NY	Cola 6 pk	1	23.00	2,345.00
NY	Cola 6 pk	2	34.00	233,546,235.00
NY	Cola 6 pk	3	54.00	346,523.00
NY	Cola 6 pk	4	12.00	46.00
LA	Diet Cola 6 pk	1	345.00	346.00
LA	Diet Cola 6 pk	2	654.00	456,257,626.00
LA	Diet Cola 6 pk	3	234.00	2,566,342.00
LA	Diet Cola 6 pk	4	567.00	245,642,564.00
LA	Cola 6 pk	1	23.00	24,562,632.00
LA	Cola 6 pk	2	74.00	768,456.00
LA	Cola 6 pk	3	82.00	34,576,768.00
LA	Cola 6 pk	4	37.00	45,768.00
Min	Diet Cola 6 pk	1	947.00	78,975,856.00
Min	Diet Cola 6 pk	2	764.00	55,467,458.00
Min	Diet Cola 6 pk	3	344.00	53,674,568.00

Min	Diet Cola 6 pk	4	375.00	45,634,562.00
Min	Cola 6 pk	1	27.00	234,756,458.00
Min	Cola 6 pk	2	59.00	54,778.00
Min	Cola 6 pk	3	65.00	4,356.00
Min	Cola 6 pk	4	36.00	24,678.00

The Row Select transformer parameters are set as follows:

Number of header rows
0

Copy header rows to output
n

Columns to include in output
all

Selection pattern specification
1,b,a,2,9,s,2,pn,4,pp,1,b,a,10,17,s,2,pn,4

The specified selection pattern describes a spreadsheet output broken into two pages as follows:

1,	Copy row 1 to the output
b,	Place a blank row in the output
a, 2, 9,	Copy rows 2 through 9 to the output
s, 2,	Place two blank rows in the output
pn, 4,	Write Page 1 in column d
pp,	Place a page break in the output
1,	Copy row 1 to the output
b,	Place a blank row in the output
a, 10, 17,	Copy rows 10 thru 17 to the output
s, 2,	Place two blank rows in the output
pn, 4	Write Page 2 in column d

The transformer output region Output 1 is connected to a spreadsheet. After the transformer runs, the spreadsheet contains the following report.

In this output, the solid black box symbol (á) indicates a page break. As specified, all five columns in the input region were sent to the output region. Also note that although the response to the `Copy Heading Rows to Output` was `no`, both output pages contain the header row that was included using the `Selection Pattern Specification` parameter. In addition, the location of the 16 lines of data coincide with the selection pattern specified in that parameter.

```
Market     Product            Period     Volume        Revenue


NY         Diet Cola 6 pk     1          234.00        32,453,452.00

NY         Diet Cola 6 pk     2          34.00         23,462,364.00

NY         Diet Cola 6 pk     3          352.00        345,234.00

NY         Diet Cola 6 pk     4          342.00        2,436,326.00

NY         Cola 6 pk          1          23.00         2,345.00

NY         Cola 6 pk          2          34.00         233,546,235.00

NY         Cola 6 pk          3          54.00         346,523.00

NY         Cola 6 pk          4          12.00         46.00


                                         Page #1

á


Market     Product            Period     Volume        Revenue


LA         Diet Cola 6 pk     1          345.00        346.00

LA         Diet Cola 6 pk     2          654.00        456,257,626.00

LA         Diet Cola 6 pk     3          234.00        2,566,342.00

LA         Diet Cola 6 pk     4          567.00        245,642,564.00

LA         Cola 6 pk          1          23.00         24,562,632.00

LA         Cola 6 pk          2          74.00         768,456.00

LA         Cola 6 pk          3          82.00         34,576,768.00

LA         Cola 6 pk          4          37.00         45,768.00


                                         Page #2
```

Using the Selection Rule Language

The Row Select transformer includes a language to allow you to build complex row selection rules. Each rule is called a pattern, and patterns generally consist of a pattern name followed by up to three numbers. A number is a positive integer. Real numbers have any decimal portion truncated, whereas values that are negative will be reset to 1.

If a row is specified and it does not exist, the row is ignored. Similarly, a repeating pattern terminates when it encounters a row that does not exist.

Patterns are executed sequentially. Thus, one pattern will terminate before the next pattern begins. Patterns cannot be specified recursively, that is, a pattern cannot contain a pattern.

The patterns are shown in Table 28 on page 182.

Table 28. Pattern definitions

Pattern Name	Definition
number	Copies the specified row number to the output.
a, number, number	Copies the rows starting with the first number and ending with the second number to the output.
b	Copies a single blank line to the output.
g, number, number	Copies the rows starting with the first number until the number of rows copied to the output equals the second number.
pn, number	Copies a row to the output containing the current page number. The number following *pn* specifies the column in which the page number will appear where 1 indicates column A, 2 indicates column B, and so on. The page number starts at 1, unless reset by the *ps* pattern, and is incremented on each *pp* pattern.
pp	Inserts a page break into the output. A page break works only when the output is copied into a Text icon. In a Spreadsheet, the page break is represented by solid lines. The *pp* pattern also increments the current page number printed.
ps, number	Sets the current page number to the number specified.
r, number, number	Copies the row specified by the first number to the output, and every nth row thereafter with the second number specifying *n*.
rg, number, number, number	A combination of the repetition and group patterns. The first two numbers work the same as in the repetition pattern; however, while a single row is copied by repetition, the repeat group pattern copies a group with the length of the third number.
s, number	Copies multiple blank lines to the output. The number specifies the number of blank lines to copy.

Encountering Errors

Errors in the `Selection Pattern Specification` parameter are generally caused by incorrect characters, missing commas, or the wrong amount of numbers following a pattern name. When the Row Select transformer encounters an error, it issues an important message and terminates. You can check the Capsule Run Log icon for an explanation of why the operation was terminated.

Split Header

The Split Header transformer splits data into two sections. The two pieces of the data set can then be processed or formatted independently.

There are two major uses for the Split Header transformer. The first application is to format a report heading independently of the report data. All data that passes through an arrow and into a Text icon is formatted in the typeface, style, and point size of the first character in the Text icon. Thus, formatting cannot be changed during a capsule application run. However, with the Split Header transformer, the heading and data can be broken into two pieces and sent through an arrow into named text fields. Each field can be formatted separately and then combined to create a report with different formats for the heading and data.

The second application of the Split Header transformer is to break a data set into two regions. For example, you can break data sets into pages, break the results of a query into manageable pieces, or break up a data set that is too large into smaller pieces that will fit in spreadsheets.

As used by the Split Header transformer, a header is significantly different from the header used with other transformers. For other transformers, a header is the first several lines in an input that will not be modified by the transformer. For the Split Header transformer, a header is simply the first set of data rows that is split from the rest of the input data.

Parameters

The Split Header transformer has five Transformer Controls window parameters. Each parameter is optional, and each option is disabled if its corresponding parameter is empty. The parameters expect only one value, and that value must be blank, 0, or a positive whole number.

Number of rows before header
This parameter specifies the number of rows that should be read and ignored before the first data section is read. The default is 0.

Number of rows in header
This parameter specifies the number of rows in the first section that should be read and copied to Output 1. The default is 0.

Number of rows between header and data

This parameter specifies the number of rows that should be read and ignored after the first section has been read, but before the second section is read. The default is 0.

Minimum number of data rows to copy

This parameter allows the user to define a lower limit on the size of Output 2. If there are fewer rows of data than specified, blank rows will be inserted at the end of the data.

Maximum number of data rows to copy

This parameter allows the user to place an upper limit on the size of Output 2. If more data is present than required, the remaining rows are ignored. By setting the `Minimum Number Of Data Rows To Copy` and `Maximum Number Of Data Rows To Copy` parameters to the same value, Output 2 will always have the specified number of rows.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- Heading (Output 1)
- Results (Output 2)

When you select either of these choices, the data associated with that choice is shown in the display area of the transformer.

Input Region Names

The Split Header transformer has one input region called Data (Input 1). Input 1 data can be in any row and column format, with any number of rows or columns. There is no limit to the size of Input 1.

Output Region Names

The Split Header transformer has two output regions called Heading (Output 1) and Results (Output 2). Output 1 holds the first section of data that is moved from Input 1. Output 2 holds the data remaining in Input 1 after the first section is removed.

Example

The following data is contained in a spreadsheet that is connected to Input 1 of the Split Header transformer:

Market Description	Product Description	Period	Volume Measure	Revenue Measure

Chi	Cola 6 pk	5.00	34.00	46.00
Chi	Diet Cola 6 pk	1.00	23.00	4,356.00
Chi	Diet Cola 6 pk	3.00	34.00	345,234.00
Chi	Diet Cola 6 pk	4.00	34.00	46.00
LA	Cola 6 pk	1.00	345.00	2,345.00
LA	Cola 6 pk	3.00	82.00	34,576,768.00
LA	Cola 6 pk	5.00	37.00	45,768.00
LA	Diet Cola 6 pk	4.00	37.00	45,768.00
Min	Cola 6 pk	4.00	36.00	2,436,326.00
Min	Cola 6 pk	5.00	36.00	2,436,326.00
Min	Diet Cola 6 pk	2.00	59.00	54,778.00
Min	Diet Cola 6 pk	3.00	65.00	768,456.00

The `Number of rows in heading` parameter is set to 1.

After the transformer runs, Output 1 contains the header information that could be formatted into multiple line column headings with the Header transformer.

Market Description	Product Description	Period	Volume Measure	Revenue Measure

Output 2 contains the following data:

Chi	Cola 6 pk	5.00	34.00	46.00
Chi	Diet Cola 6 pk	1.00	23.00	4,356.00
Chi	Diet Cola 6 pk	3.00	34.00	345,234.00
Chi	Diet Cola 6 pk	4.00	34.00	46.00
LA	Cola 6 pk	1.00	345.00	2,345.00
LA	Cola 6 pk	3.00	82.00	34,576,768.00
LA	Cola 6 pk	5.00	37.00	45,768.00
LA	Diet Cola 6 pk	4.00	37.00	45,768.00
Min	Cola 6 pk	4.00	36.00	2,436,326.00
Min	Cola 6 pk	5.00	36.00	2,436,326.00
Min	Diet Cola 6 pk	2.00	59.00	54,778.00
Min	Diet Cola 6 pk	3.00	65.00	768,456.00

Substitute

The Substitute transformer provides a powerful automated find-and-replace facility. The substitution process is determined by a set of rules, sent to the transformer through Input 1. When the Substitute transformer runs, it copies data that is in Input 2 to Output 1, performing the specified substitutions. You can search and replace any data in a data set with any spreadsheet-compatible data. You can specify whether white space and case should be ignored in text comparisons and the amount of precision in numeric comparisons.

Parameters

The Substitute Transformer Controls window has three parameters. Each parameter expects a single value. If a parameter is blank, its default value is used.

Number of header rows in data (0; 1;)

This parameter specifies the number of data rows at the top of Input 2 to copy to Output 1 without substitutions. For example, if 2 is specified, the first two rows are copied without any change. Any whole number can be specified. The default value is 0.

Ignore case and white space during text comparisons? (; y; n; yes; no)

This parameter specifies whether the search value matches a particular data value if the only difference is case or spacing. `y` or `yes` specifies that case and white space are ignored. `n` or `no` specifies that case and white space are significant. The default value is `yes`.

Numeric tolerance to allow during numerical comparisons (; 0.1; 0.01; 0.001;)

This parameter specifies how much two number values differ before the transformer decides they are different. This parameter is very useful when the precision of a number is different from its display value. Any real number value including a decimal point can be entered; the default value is 0.0.

The tolerance value is used only when two real numbers or a real number and an integer are compared. No tolerance is allowed when two integers are compared. Do not assume that the presence or absence of decimal places indicates whether a number has an integer or real number format.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Rules (Input 1)
- Data (Input 2)
- Results (Output 1)

When you select any one of these choices, the data associated with that choice is shown in the display area of the transformer.

Substitute Transformer Input Region Names

The Substitute transformer has two input regions called Rules (Input 1) and Data (Input 2).

Input 1 contains substitution rules. It is formatted as a series of rows and columns which can contain any type of data. The size of Input 1 is limited only by the availability of workstation memory.

Input 2 contains any row- and column-formatted data. Input 2 can have any number of header rows. This input region is not limited in size.

Substitute Transformer Output Region Names

The Substitute transformer has one output region called Results (Output 1), which contains the data from Input 2 after the requested substitutions are applied. The size of Output 1 is limited only by the amount of available file server disk space.

Example

In this example, the following output from a database query is copied into Input 2 of the Substitute transformer.

Market	Product Key	Period	Volume	Revenue
Min	A	1	947	78,975,856.00
Min	A	2	764	55,467,458.00
Min	B	1	-1	-1.00
Min	B	2	59	54,778.00

The `Number of header rows in data` parameters is set to `1`.

To transform this query data into a report, copy the following substitution rules to Input 1 of the Substitute transformer.

Column:	Locate:	Substitute:
# Volume Control Column		
D	-1.00	N/A
# Revenue Control Column		
E	-1.00	N/A
# Product Keys		

B	A	Spaghetti
B	B	Meatballs

After the Substitute transformer finishes running, the following report is in Output 1. The missing data, indicated by -1 in Input 1, has been replaced by the text N/A, and the product names have been substituted for the product keys.

Market	Product	Period	Volume	Revenue
Min	Spaghetti	1	947	78,975,856.00
Min	Spaghetti	2	764	55,467,458.00
Min	Meatballs	1	N/A	N/A
Min	Meatballs	2	59	54,778.00

Using Substitution Rules

Substitution rules are typically written in a spreadsheet, with each rule pertaining to one data column. Each row contains one rule: column A specifies the column to check, column B specifies the value to search for, and column C specifies the value to substitute.

The search and substitute values consider both the data type and contents of a cell. For example, the following rule:

Column A	Column B	Column C
a	-1	= N/A

specifies that every cell in column A should be searched for the numeric value -1. If -1 is found, it should be replaced with the value N/A.

To make the substitution rules less complex, two simplifications have been made in evaluating Spreadsheet data:

- The distinction between integer numbers and real numbers is ignored. For example, if the search value is the real number -1.00 and a spreadsheet cell contains the integer -1, the two numbers are considered equal.
- Only the day value of dates are considered, not the display resolution. As a result, the date 1Q93 and the date January 1993 are considered equal, because they both use January 1, 1993 as their day value.

The rule template always ignores the first row, so that you can place a header in the rules spreadsheet.

A suggested header is:

Substitution rules can be commented in two ways:

Column:	Locate:	Substitute:

- All columns to the right of column C are ignored and comments can be placed in these cells.
- Any row containing a # as the first character in column A is ignored. As a result, entire rows can be used as comments. The # can also be used to temporarily suspend a rule.

Any number of comments can be present, but the number of substitutions is limited to 1000 rules. A column can have multiple substitutions.

Subtotal

The Subtotal transformer calculates subtotals and grand totals on columnar data. This columnar data can be input from a Query, SQL Entry, or Spreadsheet tool, or from another transformer such as the Sort transformer.

Table 29 shows the types of aggregation calculations that the Subtotal transformer can perform.

Table 29. Aggregation calculations supported by the Subtotal transformer

Calculation	**Abbreviation**	**Purpose**
sum	sum	Adds all the individual values and reports a total
average	avg	Adds all the values and divides by the number of values, then reports the mean or average
maximum	max	Reports the highest value
minimum	min	Reports the lowest value
count	cnt	Reports the number of individual values
count distinct	cntd	Reports the number of unique values

The binary calculations addition, subtraction, multiplication, and division can be performed on the subtotal and grand total values, if desired.

After all calculations are completed by the Subtotal transformer, the results can be copied to the Text or Spreadsheet tools or to another transformer. You might find it useful to place the Subtotal transformer in a capsule application.

Parameters

The parameter fields allow you to control the type of calculation used, the columns used to calculate subtotals and grand totals, how columns are labeled, and where page breaks occur.

The Transformer Controls window of the Subtotal transformer contains 12 parameters. Each parameter is optional.

When entering values in the Transformer Controls window, the use of commas and parentheses is important. Use them as shown in the examples that follow the parameter field names. Capitalization, however, is only important when entering the text for the labels. All column references in any of the parameter fields refer to the input column.

Number of header rows in input (0; 1;...)

Specifies the number of rows at the beginning of the data that are used as column headings. These rows are then shown in the output but excluded from the totaling process. To exclude this row from the calculations, type `1` in this field.

Output columns (a, b, c; a, d, e;...)

Controls which columns of data are displayed and transferred after the calculations have been completed. The order in which you enter the column letters determines the order of those columns in the output. You do not need to use all the input columns in the output. Also, you cannot use an input column twice in the output.

For example, if you type `c, a, b, d, e, f` in this field, the Subtotal transformer takes columns A, B, C, D, E, and F from the data input window and places them in the listed order in the data output window. When these columns are displayed in the data output window, they have new column letters.

Show value on first occurrence only (a, b; a, d, e;...)

Simplifies the visual presentation of your report by removing repeated information that might obscure relevant facts. Enter the letters of the columns that you want duplicate values removed from. Regardless of the settings in this parameter, all columns contain a value in the first row of a new page and after each break group.

Data columns (d(sum,No), e(max);...)

Sets the type of calculation used on each column and whether you want N/As to be included in the calculations. The format for entering this information is the column letter, followed by the calculation abbreviation, and Yes or No in parentheses. For example, `d(avg,Yes)` indicates that the calculation will average all the values in column D, including N/A values.

Empty cells are treated the same as N/As. Specifying `Yes` or `No` is optional. If you do not specify a value, N/As are excluded.

N/As and empty cells are included in a calculation when the N/A or empty cell in the input is treated as a zero in a calculation, or in average, minimum, maximum, count, and count distinct calculations.

After you have created a subtotal for a column, you can use that subtotal in a binary calculation. The results of this binary calculation are displayed in the column you specify.

The binary calculations you can use are addition, subtraction, multiplication, and division. Use the **+**, **-**, *****, and **/** symbols to represent these calculations in this parameter. The values in the binary calculations can be subtotals from other columns, or they can be constants. For example, both `f(c*d)` and `f(d*100)` are valid entries. After all calculations are complted, the results can be copied to the Text or Spreadsheet tools, or to another transformer.

Binary calculations can refer to other binary calculation columns as long as the calculation order is correct. Calculations are performed from left to right. A result of one binary calculation that is used in another calculation must be displayed to the left of that calculation in this field.

For example, `c(sum), d(avg), f(c+d), e(f*100)` is an acceptable entry. But if the order were `c(sum), d(avg), e(f*100), f(c+d))`, the calculation would not work because the calculation in column F must be completed before the calculation in column E.

You cannot specify how to treat N/As in binary calculations. If one of the subtotal values for a binary calculation is an N/A, the calculation results will be an N/A.

If you specify a column as a data column, you cannot specify that same column as a break group column.

Show totals only (Yes; No)

Specifies whether you want the output to consist of the detail data (the original values) and the totals, or only the totals. The totals include both subtotals and grand totals, if selected. The valid entries are `Yes` and `No`. If you leave this parameter blank, both detail data and totals are displayed in the output.

Create Grand Total (Yes; No)

Creates grand totals in those columns that contain calculations. When grand totals are calculated, they use all the original values, not the subtotal calculation results. Valid entries for this parameter are `Yes` and `No`.

For example, if you have a column with 100 individual values divided into 8 subtotals, a grand total calculation would use the original 100 values to calculate a single average, rather than creating an average of the 8 subtotal values.

You can also choose to display grand totals even if subtotals are not calculated, which happens when break groups are not entered.

Grand Total label (Grand Total)

Specifies the text you want to be displayed in the row with the grand total calculation results. You can enter only one grand total label.

Break group columns (c, d, e:g; ...)

Specifies which columns to use to separate individual values into groups for subtotals. You can specify individual columns or a range of columns.

If you specify a range of columns, the values from all the columns in the range are considered as a single value. When that value changes, a subtotal is calculated. To enter a range, type the beginning and ending column letters, separated by a colon. To use a range, the columns in the range must be contiguous, and none of the columns in the range can be used as a data column or in another break group. When entering a range, be sure to enter the left-most column first. For example, `C:A` is not a valid entry for specifying columns A through C.

You can enter up to 25 break group columns or ranges in this parameter. Separate each column letter with a comma. You cannot specify a column as a break group if it is used as a data column.

Label (Subtotal, Market Subtotal, Period Subtotal, ...)

Specifies descriptive text to use in the subtotal labels for the break group columns.

You enter the label text in the same order that you entered the break columns. For example, if you entered the break group columns as `C,A`, enter the corresponding labels in the same order. If you do not enter a label for a column, the default label *Subtotal* is used. If you want a column to have no label, enter a space surrounded by double quotation marks.

Include break group value in subtotal label (Before, After, No, No, ...)

Provide more descriptive labels for your subtotals by including the break group value in the label.

Valid entries for this parameter field are:

Before

Adds the break group value to the beginning of the label; for example, `CHIPS 8 OZ Subtotal`

After

Adds the break group value to the end of the label; for example, `Subtotal CHIPS 8 OZ`

No

Excludes the break group value from the label

Specify the columns in the same order the columns are entered in the `Break group columns` parameter field. If you enter a single value, that value is used for all columns. If you leave this parameter blank, the break group values are added after the subtotal labels.

Page break on new value (Yes, No, No, ...)

Starts each section of your report on a new page, based on selected break group columns.

You can enter `Yes` or `No` for each break group column you specify. Enter the values in the same order that you specified the columns in the `Break group columns` parameter field. If you enter a single `Yes` or `No` value, a new page is started for each break group column.

The first row at the top of a new page contains values in each column, and the column headings are repeated on each page, regardless of the value in the `Show first break group value only` parameter.

When you copy the results of the Subtotal transformer to a Text document, the page breaks are copied with the data. When you copy the results to a Spreadsheet tool, the page break character is replaced by a small square, and the page breaks do not occur in the correct places.

If you enter an incorrect value in one of the parameter fields, an error message is displayed when you run the capsule application. Follow the instructions in the error message and then run the capsule application again.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Input 1
- Output 1

When you select either of these choices, the data associated with that choice is shown in the display area of the transformer.

Input Region Names

The Subtotal transformer has one input region called Input 1, which expects tabular data, such as data from a Spreadsheet, Query, or SQL Entry tool.

To transfer data into the Subtotal transformer:

1. Select `Input 1` in the `Display Data For` field if it is not already selected.
2. Open the icon that contains the data you want to use.

   Data can be input from a Query, SQL Entry, or Spreadsheet tool, or from another transformer.
3. Select the data from the source icon window, and press the Copy or Move function key.
4. Click in the display area of the Subtotal transformer.

If there is an error in an input data cell, all subtotals and grand totals that use that data in their calculations will contain an error, which displays as `"Error"` in the report. In addition, any other calculation that uses that data, such as a binary calculation reported in other columns, will also contain an error.

Output Region Names

The Subtotal transformer has one output region, Output 1. This region contains the values calculated by the transformer.

Switch and Text Switch

The Switch and Text Switch transformers select one of two input regions to be sent to the output based upon if-then-else logic that is evaluated when a capsule application runs. These transformers allow you to construct an application that functions much like a multiple-path capsule application.

With one exception, the two transformers operate identically. The only difference is the type of input regions manipulated by the transformers.

- The Switch transformer has two input regions that can be connected to a Spreadsheet, SQL Entry icon, or any other row- and column-oriented source.
- The Text Switch transformer has two input regions that can be connected to a Text icon or any other text formatted source.

The Switch and Text Switch transformers can join two divergent paths so that the data of the selected path is passed to the output region. The transformer selects the path using true/false logic. Alternatively, 0/1 and yes/no logic are also supported.

Switch and Text Switch allow run-time decision making within a capsule application. However, this function has a disadvantage: both possible paths are executed before the selected path is chosen. This function usually does not present a problem when transformers or spreadsheets are placed in a capsule application branch. However, because SQL code is always executed, side effects are possible when an SQL Entry icon is placed in a branch. Consequently, you should be careful when placing an SQL Entry icon in a branch.

For a large and complex application, it might be desirable to run one of several different SQL programs, based upon some user-selectable or run-time condition. The Text Switch transformer can be used to select between one of two different sets of code.

Parameters

The Switch and Text Switch Transformer Controls windows have one parameter, `Control value (0 or FALSE —> Data 1; 1 or TRUE —> Data 2)`. This parameter determines whether Input 1 or Input 2 is passed to Output 1. The particular input region is chosen using the logic for each data type. There is no default.

Text strings are not case sensitive. Parameter values are set by @-variables, the values of which can be determined during the processing of the capsule application.

Table 30. Logic for Switch and Text Switch input region

Data type	Control value content	Input region
Numeric	Any value £0.0001	Data 1 (Input 1)
	Any value > 0.0001	Data 2 (Input 2)
Text	true, yes, two, ok, or y	Data 2 (Input 2)
	other text	Data 1 (Input 1)
Other data types	blank	Data 1 (Input 1)

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data 1 (Input 1)
- Data 2 (Input 2)
- Results (Output 1)

When you select any one of these choices, the data associated with that choice is shown in the display area of the transformer.

Input Region Names

The Switch and Text Switch transformers have two input regions called Data 1 (Input 1) and Data 2 (Input 2). Input 1 and Input 2 represent the end of the *then* and *else* paths through the capsule application, respectively. If both inputs regions are empty, the transformer runs until the region connected to Input 2 is transferred along its arrow during the capsule application run.

The Switch transformer is designed to work only with a row- or column-formatted input, such as data read from a Spreadsheet, Query, or SQL Entry icon. The Text Switch transformer is designed to work with text data, such as that in a Text icon.

Output Region Names

The Switch and Text Switch transformers have one output region called Results (Output 1), which contains the data from either Input 1 or Input 2, depending upon the setting of the `Control Value` parameter.

Example

Two different reports are generated by two different sets of SQL statements in a capsule. To make it easy to select and run the reports, the developer creates a Data Entry icon where the user specifies the report desired. The data entry icon sets the value of @B to 0 for a report of shipments data and 1 for a report of returned shipments.

To use the Text Switch transformer in this application, the developer connects a Text icon to Input 1 and another Text icon to Input 2 of the Text Switch transformer. The contents of the first Text icon for shipments are:

```
Select prodID, prodDesc,volume, revenue from ShipLedger where
volume >
1000000 order by volume
```

The Text icon connected to Input 2 contains the SQL statements for the returns report:

```
Select prodID, prodDesc, returnvol, returncost from ReturnLedger
where
returncost > 1000 order by returnvol
```

The value of @B is set in a Data Entry icon in the capsule application containing the Text Switch transformer. If the user wants a shipments report, the value of @B is 0, so the contents of Input 1 are sent to the output and the attached SQL Entry icon receives SQL statements needed to produce a shipments report.

The contents of Output 1 are:

```
Select prodID, prodDesc, volume, revenue from ShipLedger where
volume >
1000000 order by volume
```

Alternatively, if the user requests the returns report, the value of @B is set to 1. This forces the Text Switch transformer to send the contents of Input 2 to Output 1. In that case, the output contains the SQL statements necessary to produce a returns report. The contents of Output 1 are:

```
Select prodID, prodDesc, returnvol, returncost from ReturnLedger
where
returncost > 1000 order by returnvol
```

Text to Spreadsheet

The Text To Spreadsheet transformer converts information contained in a Text icon into spreadsheet format. This transformer is helpful for dealing with reports formatted as Text icons and information transferred from other computing environments.

Text to Spreadsheet

The Text To Spreadsheet transformer is similar to the Data Format transformer. The Text To Spreadsheet transformer transfers data from a Text icon to a spreadsheet.

The Text To Spreadsheet transformer supports several formatting features beyond those offered by the Format tool. You can use any character to specify column boundaries, including spaces, tabs, and commas. This flexibility is useful for transferring files into an Meta5 system from programs that use the tilde (˜) or vertical bar (|) to delimit columns. Extra white space can be removed, eliminating much of the manual cleanup required when data is imported into Meta5.

The Text To Spreadsheet transformer automatically converts data into an appropriate Spreadsheet data format. Text, such as 123.45, is formatted as a number. All data types are supported, including Boolean data, special data values such as N/A and Error, and all date formats.

Parameters

The Text To Spreadsheet Transformer Controls window has five parameters.

Column separation character (tab; space; comma; :; –;)

This parameter specifies the symbol to indicate the start of a new column. This symbol is typically a tab character, but any character is allowed. The words `tab` and `space` let you enter these special characters. All other characters, including the comma and semicolon, can be entered as they are shown. If this parameter is blank, `tab` is the default value.

Retain column separation character in output? (y; n)

This parameter specifies whether the column separation character will be removed from the output data. `Yes` retains the character. `No` removes the character. The default is `no`.

Remove leading and trailing white space? (y; n) remove excess internal white space? (y; n)

This parameter specifies whether extra spaces or tabs will be removed before columns are separated. This parameter expects two answers, separated by a comma. The first part of this parameter concerns white space at the beginning or the end of a data value. It is generally desirable to remove leading and trailing white space, especially if the data is to be uploaded into a database. `Yes` removes leading and trailing white space; `no` does not. The default is `no`.

The second part of this parameter deals with removing excess internal white space, which is more than one consecutive space or tab. `Yes` specifies that excess internal white space is to be removed; `no` specifies that it should not be removed. The default is `no`.

All leading, internal, and trailing white space is removed if the `Convert text to native data type` parameter has a value of `yes` regardless of the value for `Remove leading and trailing white space` parameter.

Convert text to native data types? (y; n)

This parameter specifies whether the text information in Input 1 is to be formatted as text or converted to other data types. `Yes` converts data to its native data type; `no` causes all data to be formatted as text. The default is `no`.

In some circumstances, these parameters might interact in subtle ways. For example, if the data is to be converted into its native format, but the separation character is retained, some data types might not be recognized. For example, if the separation character is a semicolon (;), numerical data would not be recognized as numbers because the semicolon symbol is not part of the definition of a number.

Remove excess white space before separating data into columns? (y; n)

This parameter specifies whether extra spaces or tabs will be removed before data is converted into columns. `No` retains the extra space; `yes` removes the character. The default is `no`.

If you previously specified a semicolon as a list separator in the Text to Spreadsheet transformer, the semicolons will convert to commas when you upgrade the icon. You must manually change the commas back to semicolons.

When the desktop preference for either the thousands separator or the decimal separator is set to comma, incorrect results might be produced if a comma is specified as the Text To Spreadsheet column separation character and the incoming data contains a reference to a capsule @-variable that is a number. For example, if @A has the value of 1234 and is referenced in a Text icon on a desktop with the thousands separator set to comma, it will display as 1,234. If @A was type Text rather than Number, no comma will be inserted.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Text (Input 1)
- Results (Output 1)

When you select either of these choices, the data associated with that choice is shown in the display area of the transformer.

Input Region Names

The Text To Spreadsheet transformer has one input region called Text (Input 1) that expects data from a Text icon or a PC Text icon.

Output Region Names

The Text To Spreadsheet transformer has one output region called Results (Output 1) that contains the information after it is converted to Spreadsheet format.

Example

Assume the following data is contained in a text document transferred from a database residing on an IBM RISC System/6000 computer. For this data to be used in a capsule application, it must be converted into Spreadsheet format.

Full Name;Address;City;State;Zip;Customer

Doe, Jim;3304 W 4TH St.;Tampa;FL;12345;Y

Mason, Kathy;P.O. Box 345;Fitchburg;WI;53703;N

Wilson, John;18 West Place;Chicago;IL;60609;Y

The Text To Spreadsheet transformer parameters are set as follows:

Column separation character
;

Retain column separation character in output?
n

Convert text to native data types?
y

After the Text To Spreadsheet transformer runs, the following Spreadsheet data is displayed in Output 1. The data in the Full Name, Address, City, State, and Customer columns is formatted as text fields, and the data in the Zip column is formatted as integer numbers.

Full Name	Address	City	State	Zip	Customer
Doe, Jim	304 W 4TH St.	Tampa	FL	12,345	Y
Mason, Kathy	P.O. Box 345	Fitchburg	WI	53,703	N
Wilson, John	18 West Place	Chicago	IL	60,609	Y

Determining Data Type Conversion

The Text To Spreadsheet transformer completes data type conversions in the following order:

1. Unspecified (blank)
2. Integer number
3. Real number
4. Boolean
5. Date
6. N/A

7. Error
8. Text

The Text To Spreadsheet transformer attempts data type conversions only after it separates the data into columns and removes all white space from the start and end of each cell. After these steps are complete, the following definitions determine the data type of a particular cell:

Unspecified
A string with no characters is converted to a blank cell. This conversion is performed when two adjacent column separation characters are located in the input.

Integer number
An optional **+** or **-** character, followed by 1 to 10 digits; for example, -234. All internal commas are discarded, and one matched set of parentheses is interpreted as a negative.

Real number
An optional **+** or **-** character, followed by zero or more digits, an optional decimal point, and zero or more digits; for example, -234.23. All internal commas are discarded, and one matched set of parentheses is interpreted as a negative.

Boolean
The character strings **true** and **false** are recognized as Boolean data. Case is not significant. Thus, **FALSE**, **False**, and even **FaLsE** are recognized as Boolean false values.

Date
Any character string that is a valid Meta5 date format, except a year date, is recognized as a date. Year dates are formatted as integers. The date resolution is determined by the date string value and format.

N/A
A text string of NA, na, or n/a is recognized as N/A data. Case is not significant.

Error
A text string of Error is recognized as Error data. Case is not significant.

Text
If the contents of a cell do not fit any of the previous definitions, it is formatted as text.

Although it is often difficult to move data between hardware platforms, each system generally has a facility to save information as a text file (ASCII file). From an IBM workstation, users can usually print a file as a DOS text file, which is simply a text file with some printer formatting information. A DOS text file is generally easy to convert into a spreadsheet.

Many workstations support the DIF, RFT, and WKS file formats. Although Meta5 has several tools to convert these files directly into spreadsheets, converting these files to Text icons has several advantages, such as faster processing and flexibility for fixing any problems that might have occurred during the move to the Meta5 environment.

IBM mainframe computers generally do not use the ASCII character set commonly found on workstations, but most systems have a file conversion utility that creates ASCII files that can be read by the Text To Spreadsheet transformer. Also, some IBM mainframe dial-in ports automatically perform the conversion to ASCII. When you convert large workstation-format data files into spreadsheets, it might be more convenient to connect the PC Text icon to a Text To Spreadsheet transformer.

Transpose

The Transpose transformer interchanges rows and columns in a data set. The transpose operation is useful when data read from a database in row format is to be presented in a column-oriented report.

Parameters

The Transpose transformer has no parameters.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- Results (Output 1)

When you select either of these choices, the data associated with that choice is shown in the display area of the transformer.

Input Region Names

The Transpose transformer has one input region called Data (Input 1). Input 1 can be in any format, with any number of rows or columns, provided the data set will fit entirely within workstation memory. Data can be read from any tool that can be connected to a spreadsheet. Also, the input data need not be a rectangular region and the length of each row or column can vary.

Output Region Names

The Transpose transformer has one output region called Results (Output 1) that contains the data read from Input 1, but with its rows and columns interchanged.

Example

In this example, the following data is read from a database table:

Market	Volume	Share	% Change
Chicago	56,900	59%	9.42%
New York	36,835	24%	-7.91%
Minneapolis	12,316	36%	53.95%
Total	106,051	37%	6.05%

The data is in row format; each line of data has all of the information needed for a particular market.

The final report is to be presented in column format, with the Market names across the top. The Transpose transformer generates the following report:

Market	Chicago	New York	Minneapolis	Total
Volume	56,900	36,835	12,316	106,051
Share	59%	24%	36%	37%
% Change	9.42%	-7.91%	53.95%	6.05%

Word Count

The Word Count transformer counts characters, words, and paragraphs in a Text icon. In addition, the Word Count transformer computes a file checksum value.

This transformer is useful for text processing and for application development. For application developers, the Word Count transformer is also useful to ensure that the code will meet tool or database size limitations. Use the Word Count transformer to keep your code to correct length. Also, because SQL can only be a certain length for different database platforms, use this transformer to debug problems.

The Word Count transformer computes a checksum value, which represents entire contents of a text file. The checksum value is used to determine if two similar files are the same; identical files have the same checksum value. Two files with minor differences will almost always have different checksums. This feature is useful if two or more people are editing a file and you are not sure if changes have been made to a copy of a file.

Parameters

The Word Count transformer has no parameters.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Text (Input 1)
- Results (Output 1)

When you select either of these choices, the data associated with that choice is shown in the display area of the transformer.

Input Region Names

The Word Count transformer has one input region called Text (Input 1), which is a text region that can be connected directly to a Text icon or copied into Input 1.

Output Region Names

The Word Count transformer has one output region called Results (Output 1), which reports the number of characters, words, and paragraphs in Input 1.

Example

The following information is contained in a Text icon that is connected to Input 1 of the Word Count transformer.

```
SECTION HEADING
Document: 197 89-0003-16-B

I. First Level Heading

   First level text

   A. Second Level Heading

      Second Level descriptive text
```

After the transformer runs, the following information is contained in Output 1.

# Characters	134
# Words	20
# Paragraphs	6
Check Sum	60,566

Applying Definitions

The definitions of a character, word, and paragraph as they apply to the Word Count transformer are:

- A character is any printable alphabetic, numeric, or punctuation symbol, a space, or a tab. Items that are not characters include Enter, Shift+Enter, page breaks, field markers, and page numbers.
- A word is any continuous sequence of characters terminated by a space, tab, Enter, or Shift+Enter.
- A paragraph is any continuous sequence of characters terminated by an Enter or Shift+Enter. Blank lines are not counted as paragraphs.
- A checksum is a number that uniquely represents an entire Text icon. While it is possible, but unlikely, that two different icons will have the same checksum, two similar but different files will have different checksum values. For example, even transposing two characters in a Text icon will result in different checksums.

Write SQL

The Write SQL transformer translates row- and column-based data into SQL statements. This transformer is useful in a capsule application to create the SQL to load data into a database through an SQL Entry tool, allowing you to quickly load large amounts of data.

The Write SQL transformer formats data into SQL statements. Each row of data is combined with an SQL fragment to become a complete SQL statement. For example, consider the following data:

Date	Volume
Jan	10
Feb	35
Mar	18

Given the SQL fragment:

```
insert into TEMP_002 values (
```

the transformer generates the following SQL statements:

```
insert into TEMP_002 values ('Jan', 10)
insert into TEMP_002 values ('Feb', 35)
insert into TEMP_002 values ('Mar', 18)
```

Parameters

The Write SQL transformer has nine parameters; only the second parameter, `Select columns to output`, is required. The other parameters have default values that often do not require adjustment.

Number of header rows in data (0; 1;)

This parameter is the number of rows in Input 4 that are skipped before reading the data. This parameter prevents column headers from being formatted into SQL statements. Any positive integer can be specified; the default value is 0.

Select columns to output (d; d,e,f;)

This parameter is a list of Input 4 columns that should be formatted into SQL statements. Each column is treated as one item of an INSERT statement and each row is treated as a different insert statement. Enter a list of column letters, or their numeric equivalent, separated by commas. For example, the specification `a,c,d` will have the transformer format the data in columns a, c, and d into SQL insert statements. The specification a:d would format data in columns a through d, inclusively.

Preamble SQL statement (create table)

This parameter is a text string that precedes the body of the generated SQL statements. The preamble can consist of any string that forms a valid SQL statement. It also can be specified using @-variables. The preamble string can be up to 512 characters. This parameter will be truncated if it is greater than 512 characters after expansion of @-variables. Because the transformer uses commas to separate parameter values, the preamble must be placed in double quotation marks if the SQL statement contains commas. Using this parameter is more efficient than using Input 1. If Input 1 is used with this parameter, the string in Input 1 will be displayed in the output before the contents of this parameter.

Each line SQL statement (Insert table)

This parameter is the SQL fragment that prefaces each row of data in Input 4. Generally, this is an INSERT statement in the form: `insert into table_xx values`. In the output, the open parenthesis is matched by a closing parenthesis that follows a list of values found in Input 4.

Postamble SQL statement (drop table)

This parameter is like the Preamble SQL statement, except it is placed after the INSERT statements formed from the input data. If used with Input 3, the value of this parameter will be displayed in the results before the value found in Input 3.

Number of decimal places in real numbers (1; 5;)

This parameter determines the precision of the numbers that are converted into SQL statements. The default value is 2, and any value from 0 to 40 is acceptable. Meta5 generally carries 14 decimal places of precision. If 0 is specified, each number is converted into an integer, and no decimal point is included in the SQL statements. Leading and trailing zeros are removed from all numbers to reduce the size of the SQL code.

Format dates as numbers or text (n; t)

This parameter controls the appearance of dates in the output. In number format, each date is formatted as the number of days since January 1,

1970. Specifying `t` formats each data as a text string, based on the date format. The default is `n`. The transformer does not change a date format that is present in Input 4.

Format error, n/a, unspecified as text, blank, or 0 (t; b; z;)

This parameter controls the appearance of each of these data types. If text is specified, the words Error, N/A, and Unspecified are placed in the SQL statements. If 0 is specified, the number 0 is placed in the SQL statements; if blank is specified, a blank string is placed in the SQL statements. The default is 0.

Format boolean as numbers or text (n; t)

This parameter controls the appearance of true and false data. In the text mode, the words True and False are placed in the SQL statements. Using the number format results in the number 1 replacing True, and 0 replacing False. The number format conserves storage space and can speed up certain data manipulations. The default is `t`.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Preamble (Input 1)
- Insert (Input 2)
- Postamble (Input 3)
- Data (Input 4)
- SQL (Output 1)

When you select any one of these choices, the data associated with that choice is shown in the display area of the transformer.

Input Region Names

The Write SQL transformer has four input regions. Only Input 4 is required. Write SQL starts running when the first row of data arrives in Input 4, even if it receives nothing from Inputs 1, 2, and 3 (Preamble, Insert, and Postamble).

Input 1 contains SQL code that is run before the body of the code is processed. This input can have any number of rows and columns. If the code is under 512 characters, it can be entered directly into the `Preamble SQL statement` parameter.

Input 2 is the SQL statement that precedes each row of data in the body of the SQL statements. Typically, it is an INSERT statement in the form: `insert into tablename values(`. The closing parenthesis is automatically added to each SQL statement. If the code is under 512 characters, it can be entered into the `Each Line SQL statement` parameter.

Input 3 is SQL code that is run after the body code. If the amount of code is under 512 characters, it can be specified in the `Postamble SQL statement` parameter.

Input 4 is the data that forms the body of the generated SQL statements. Each data row is formatted into SQL-compatible data types and combined with the SQL fragment read from Input 2, or from the `Each line SQL statement` parameter if Input 2 is empty.

Output Region Names

The Write SQL transformer has one output region. Output 1 is the SQL code generated from the input regions and parameters. For the best results, connect this output to a Text Icon. The SQL code is displayed in five subsections in the order shown:

1. Preamble read from Input 1 (Preamble) input region
2. Preamble read from the `Preamble SQL statement` parameter
3. Body code formed from Input 2 (Insert) and Input 4 (Data) input region
4. Postamble read from the parameter `Postamble SQL statement`
5. Postamble read from Input 3 (Postamble) input region

If the output is sent to a Text icon, it is delimited with tab characters.

Example

The following data is contained in a spreadsheet that is connected to the transformer's Input 1 (Preamble) input region:

```
create table DatePeriod (period integer, seq integer, perseq
integer, days integer)
```

The following data is contained in a spreadsheet that is connected to the transformer's Input 4 (Data) input region:

Date	Seq #	Period #	Trading Days
January 1, 1983	5	1	22
February 1, 1983	6	2	23
March 1, 1983	7	3	24
April 1, 1983	8	4	25
May 1, 1983	9	5	25
June 1, 1983	10	6	24
July 1, 1983	11	7	23
August 1, 1983	12	8	22

September 1, 1983	13	9	22
October 1, 1983	14	10	23
November 1, 1983	15	11	24
December 1, 1983	16	12	25

The Write SQL transformer parameters are set as follows:

Number of header rows in data
1

Select columns to output
a:d

Each line SQL statement
Insert into Date Period values (

Number of decimal places in real numbers
9

Format dates as numbers or text
n

When the transformer runs, the following information is sent to a Text icon that is connected to Output 1.

```
create table DatePeriod (period integer, seq integer, perseq
integer, days integer)
insert into DatePeriod values (   8401,    5.,     1.,     22.    )
insert into DatePeriod values (   8432,    6.,     2.,     23.    )
insert into DatePeriod values (   8460,    7.,     3.,     24.    )
insert into DatePeriod values (   8491,    8.,     4.,     25.    )
insert into DatePeriod values (   8521,    9.,     5.,     25.    )
insert into DatePeriod values (   8552,    10.,    6.,     24.    )
insert into DatePeriod values (   8582,    11.,    7.,     23.    )
insert into DatePeriod values (   8613,    12.,    8.,     22.    )
insert into DatePeriod values (   8644,    13.,    9.,     22.    )
insert into DatePeriod values (   8674,    14.,    10.,    23.    )
insert into DatePeriod values (   8705,    15.,    11.,    24.    )
insert into DatePeriod values (   8735,    16.,    12.,    25.    )
```

An SQL Entry icon can then be configured to obtain the SQL statements from this Text icon. When those SQL statements are run, a table named DatePeriod containing four fields is created. Then, the values found in the Write SQL transformer are inserted into the DatePeriod database table.

SQL Syntax Support

The Write SQL transformer supports all Meta5 data formats. The Write SQL transformer encloses character data in single quotation marks and handles all embedded quotation marks.

You can specify how the transformer handles several data types. For example, the N/A data type can be stored in a database as the character string N/A, a blank character string, or 0. Blank values and Error are handled in a similar manner. Boolean values, such as True and False, can be formatted as the text strings `true` and `false`, or as numeric values 1 and 0.

The Write SQL transformer supports all Meta5 date formats and the 4-digit numeric date format. You can specify whether dates are formatted as numbers or as character strings. See “Period Table” on page 155 for more information on date formats.

Numeric values are read in the precision supported by the Meta5 desktop, which is typically 14 decimal digits. You can specify the number of decimal places for the numbers in the generated SQL statements. In general, the fewer decimal places, the smaller the resulting code and the faster the code executes.

The Write SQL transformer does no SQL validity checking.

@-Variables Support

In the transformer input regions, @-variables are treated differently than they are in the Transformer Controls window. In any input region, an @-variable name is replaced with its @-variable value when you place an equal sign (=) (@-variable name) in an input region. Alternatively, in a Transformer Controls window, an @-variable is replaced by its value.

For example, suppose the @-variable @A has the value *revenue*. If @A is displayed in an input region, *@A* is displayed in the SQL code. If @A is in the Transformer Controls window, *revenue* is displayed in the SQL code. If \@\A is displayed in the Transformer Controls window, @A is displayed in the SQL code.

How Data Types Are Handled

Table 31 details the handling of data types of a spreadsheet cell.

Table 31. Write SQL transformer data type handling

Data type	Handling
Text string	A sequence of up to 512 characters. Text strings are enclosed in single quotation marks and all internal single quotation marks are represented by two adjacent single quotation marks.
Integer	Integers are formatted as they appear.

Table 31. Write SQL transformer data type handling

Data type	Handling
Decimal number	Real numbers are sent to the output with the number of decimal places specified in the `Number of decimal places in real numbers` parameter. Leading and trailing zeros are removed; if you specify 0 decimal places, the decimal point is removed.
N/A	This value is formatted in one of three ways depending upon the value of the `Format error, n/a, unspecified as text, blank, or zero` parameter. If that parameter value is *text*, N/A values are formatted as the text string *N/A*. If the value is *blank*, N/A values are formatted as a blank text string. All other values are formatted as 0 (two single quotes with no space). If the value is 0, N/A values are formatted as 0.
Error	These values are formatted in one of three ways depending upon the value of the `Format error, n/a, unspecified as text, blank, or zero` parameter. If the value is *text*, Error is formatted as the text string *Error*. If the value is *blank*, Error is formatted as a blank text string. If the value is 0, Error is formatted as 0.
Unspecified	This value is formatted in one of three ways depending upon the value of the `Format error, n/a, unspecified as text, blank, or zero` parameter. If the value is *text*, blank cells are formatted as the text string Unspecified. If the value is *blank*, blank cells are formatted as a blank text string. If the value is 0, blank cells are formatted as 0.
Date	Dates are formatted in one of two ways depending upon the value of the `Format dates as numbers or text` parameter. If the value is *text*, a date is formatted using date format rules and Day resolution is assumed if resolution information is not available. If the parameter value is *numbers*, the date is formatted as an integer equal to the number of days since January 1, 1970.
Boolean	True/false data is treated in one of two ways depending upon the value of the `Format boolean as numbers or text` parameter. If the value is *text*, true is formatted as the string True, and false is False. If the value is *numbers*, true is formatted as the integer 1, and false as the integer 0.

Chapter 4. Significance and Sample Testing Transformers

Significance and Sample Testing transformers allow you to determine the statistical significance of differences in data samples and to perform distribution comparisons of sets of data. Each Significance and Sample Testing transformer performs a specific function. For example, you can use the ANOVA (analysis of variance) transformer to examine differences in distributions of a variable across several groups of data. You can use the CrossTab transformer to study the distribution of cross-classified data.

To locate the Significance and Sample Testing transformers:

1. Open the New Icons file drawer.
2. Open the Transformer Icons file drawer.
3. Open the Capsule Transformers folder.

If you cannot find a specific transformer, see your system administrator.

Table 32 lists the Significance and Sample Testing transformers, describes their functions, and gives the page number where each transformer is described.

Table 32. Significance and Sample Testing transformers

Transformer	Function	See
ANOVA	Displays differences in distributions of a variable across several groups of data.	“ANOVA” on page 212
ChiSquare	Displays associations between pairs of variables and differences in distributions between an observed sample and a theoretical sample.	“ChiSquare” on page 231
CrossTab	Constructs contingency tables that you can use to study distributions of cross-classified data.	“CrossTab” on page 245
IPMean	Performs the t-test to measure differences in the distribution of a variable between two paired or independent samples.	“IPMean” on page 263
Kruskal	Performs the Kruskal-Wallis nonparametric test to determine whether different samples have different distributions of a variable.	“Kruskal” on page 277
KSTest	Performs the Kolmogorov-Smirnov nonparametric test to determine whether an observed distribution differs from an expected distribution or whether the distribution of a variable differs between two groups.	“KSTest” on page 290

Table 32. Significance and Sample Testing transformers

Transformer	Function	See
NPCorrelation	Computes Spearman Rank and Kendall's Tau nonparametric correlation coefficients to summarize the relationships among two or more variables.	“NPCorrelation” on page 309
NPIndependent	Performs the Mann-Whitney and Wilcoxon Rank-Sum nonparametric tests to determine whether two independent samples have different distributions of a variable.	“NPIndependent ” on page 321
NPPaired	Performs the Wilcoxon Signed Rank and Sign nonparametric tests to determine whether two independent samples have different distributions of a variable.	“NPPaired” on page 331

For general information on using transformers, see “Chapter 1. Getting Started with Transformers,” on page 1.

ANOVA

The ANOVA transformer computes the following analysis of variance (ANOVA) statistics:

- One-way ANOVA
- Two-way ANOVA
- Three-way ANOVA
- Linear contrast analysis

The ANOVA models are widely used statistical methods for determining whether the means of several groups are different. Linear contrast analysis extends the usefulness of ANOVA models by combining two or more smaller groups to create analysis categories.

ANOVA models are often used to compare experimental data in which a set of observations can be broken into two or more groups. The mean of each group is used as a basis for the analysis of variance test. The typical question is: Are the means of each group the same, or are they different? The answer to this question is based on the behavior of the data. If each data observation lies near its corresponding group mean and the group means are well separated, it is safe to assume that the means are different. However, if the data observations do not lie near their group mean, or if the data values from each group overlap, then differences in the group mean values might not be significant.

For example, suppose two groups have a mean of 22.5 and 23.4, respectively. The difference between these two values is 0.9, or about 4%. If all values in the first group lie between 22.2 and 22.6 and all values in the second group are between 23.3 and 23.5, then the mean values seem to be different, and the conclusion is that the data seems to represent two different populations. The data in this case has some variability between the two groups and has very little variability within each group.

Alternatively, suppose two groups have the same means, 22.5 and 23.4, respectively, but the distribution of observations is very large and there is a great deal of overlap between groups. In this case, the variability within groups is so large that it overwhelms the differences between the groups. Thus, the means are not significantly different, and the groups seem to be two samples taken from the same population or two identical populations.

The ANOVA transformer provides statistics that measure whether the variation between two groups is so large that it is more important than the variation within groups. If the variation between groups is more important than the variation within groups, the groups can be considered as different.

Parameters

All of the parameters described here are displayed in the ANOVA Transformer Controls window. Most of these parameters are also displayed in the Data Entry icon contained in the transformer's demonstration capsule application. However, to simplify use of the transformer for new users, some of the more advanced parameters might be omitted from the Data Entry icon; they can be accessed only in the Transformer Controls window.

Number of header rows (; 1; 5;) row containing title (,1; ,5;)

This parameter specifies the number of rows in the input data that are not used in calculating ANOVA statistics. This parameter also specifies the row number containing column titles that are used as labels in the output. The two values should be separated by a comma or by the character defined as the list separator for your system. You can specify any number of header rows; the default value is 0. The title row number should be less than or equal to the number of header rows. If the header rows value is specified, the default title row is the last row of the header rows. If no header is specified, the default is no title row.

Report name (ANOVA Statistics)

This parameter is a title for the output report, for example, *ANOVA Statistics*. If the `Report Name` parameter is blank, no title is displayed in either Output 1 (ANOVA Table) or Output 2 (Summary).

Columns for category (a; a,b; b,c,a;)

This parameter specifies the columns containing category information. A category column contains information used to break a data set into one or more subclassifications. At least one and as many as three columns can be specified. No default values are provided. The `Columns For Category` parameter indirectly determines the ANOVA model that is used. If one column is entered, the transformer computes one-way ANOVA. Entering two columns selects a two-way ANOVA; entering three columns selects a three-way model. For example, `d,f` specifies that ANOVA should compute a two-way ANOVA using columns D and F for category information.

Data column (a; b; c;)

This parameter specifies the column used to compute the analysis of variance statistics. Only one data column value can be entered. For example, the entry `c` computes the analysis of variance model using column C as the source of the sample data. No default values are provided.

Perform linear contrast analysis? (no; yes, 1, 0, -1;)

This parameter specifies that a linear contrast expression be performed in a one-way ANOVA. For higher level ANOVAs, this parameter is ignored. If this parameter is blank, linear contrast analysis is not performed.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- ANOVA (Output 1)
- Summary (Output 2)
- Messages (Output 3)

When you select any one of these choices, the data associated with that choice displays in the display area of the transformer.

Input Region Names

The ANOVA transformer has only one input region, Input 1 (Data). You can modify the input name to reflect the name of the corresponding data in the display area. Input 1 consists of data read from a Spreadsheet, Query, or SQL Entry icon, or information copied directly into the transformer. Input 1 contains columns containing data (data columns) and columns containing category information (`Columns for Category`). If extra columns are present. they are ignored. The input data can have any number of header rows. Column titles are read from the specified row. If titles are not found, default titles are provided.

Output Region Names

ANOVA generates three output regions that are not limited in size. You can modify the output name to reflect the name of the corresponding data in the display area.

Output 1 (ANOVA Table) consists of the following results formatted into an ANOVA table:

- Sample source description
- Number of degrees of freedom
- Sums of squares

- Mean sums of squares
- F-value
- p-value

Output 2 (Summary) contains summary statistics for each group, including:

- Sample variable description
- Sample count
- Sample mean
- Sample variance
- Sample standard error

Output 3 (Messages) contains:

- Transformer run-time messages
- Warnings
- Error messages
- A timestamp for documentation purposes

Examples

This example demonstrates a one-way ANOVA (Case I) and a two-way ANOVA (Case II). It shows the sales impact of a promotion run in Chicago, New York, and Los Angeles. Assume that the following data is contained in a Spreadsheet window that is connected to Input 1 of the ANOVA transformer:

Promotion	Market	Period	Sales
Before	Chicago	1	37
Before	Chicago	2	16
Before	Chicago	3	33
Before	Chicago	4	40
Before	LosAngeles	1	56
Before	LosAngeles	2	62
Before	LosAngeles	3	57
Before	LosAngeles	4	72
Before	NewYork	1	34
Before	NewYork	2	41
Before	NewYork	3	64
Before	NewYork	4	64

After	Chicago	5	54
After	Chicago	6	17
After	Chicago	7	21
After	Chicago	8	49
After	LosAngeles	5	62
After	LosAngeles	6	72
After	LosAngeles	7	61
After	LosAngeles	8	91
After	NewYork	5	48
After	NewYork	6	64
After	NewYork	7	63
After	NewYork	8	34

Case I: The Transformer Controls window parameters are set as follows to perform one-way ANOVA:

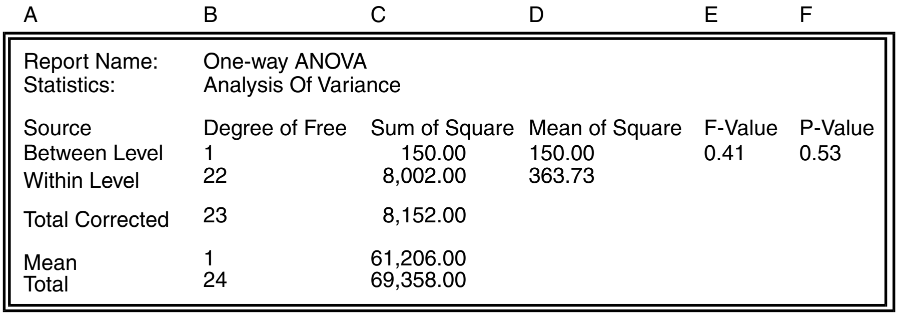
Number of header rows
1, 1

Report name
One-way ANOVA

Columns for category
a

Data column
d

When the transformer run finishes, the following information is displayed in Output 1:



```
A                B                C                D                E         F
Report Name:     One-way ANOVA
Statistics:      Analysis Of Variance

Source           Degree of Free   Sum of Square    Mean of Square   F-Value   P-Value
Between Level    1                   150.00        150.00           0.41      0.53
Within Level     22                8,002.00        363.73

Total Corrected  23                8,152.00

Mean             1                61,206.00
Total            24               69,358.00
```

In this example, there are only two groups (before and after) in the category column A. It seems that the advertising campaign did not have a significant impact on sales. The resulting small F-value (0.41) and high p-value (0.53) indicate that the means are not different. In other words, the p-value in this example indicates that there is a 53% chance that the sample means are equal.

In Output 2, there is a difference in the average sales before and after the advertising campaign, but the standard deviations indicate that the values in each group overlap so much that all of the data values lie in essentially the same range:

```
A              B              C              D              E

Report Name:   One-way ANOVA
Statistics:    ANOVA Summary

Variable       Count          Mean           Variance       Std Deviation
After Ads      12             53.00          444.91         21.09
Before Ads     12             48.00          282.55         16.81
All Data       24             50.50
```

Case II: The Transformer Controls window parameters are set as follows to compute two-way ANOVA:

Number of header rows
1, 1

Report name
Two-way ANOVA

Columns for category
a, b

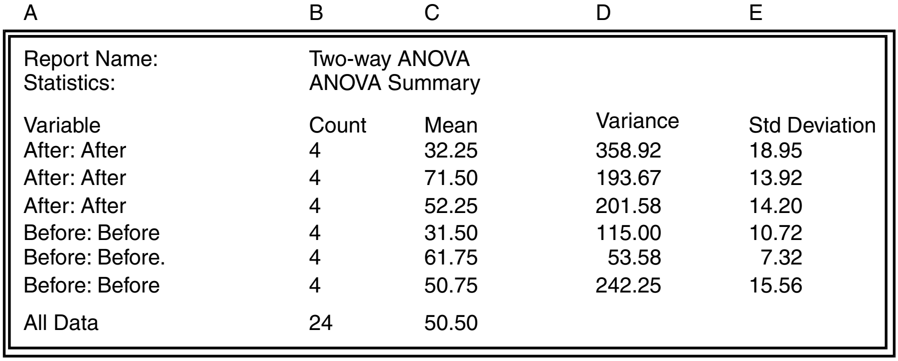
Data column
d

When the transformer has finished running, the following information is displayed in Output 1:

```
A                B               C              D               E          F

Report Name:     Two-way ANOVA
Statistics:      Analysis Of Variance

Source           Degree of Free  Sum of Square  Mean of Square  F-Value    P-Value
Promotion        1                  150.00          150.00       0.77       0.39
Market           2                4,434.25        2,217.13      11.42       0.00
Promotion: Promo 2                   72.75           36.38       0.19       0.83
Residual         18               3,495.00          194.17
Total Corrected  23               8,152.00

Mean             1               61,206.00
Total            24              69,358.00
```

The transformer calculated a two-way ANOVA on each category column, and then computed the interactions between the category columns. The p-values are high for Promotion and the interaction of Promotion and Market, indicating that the group means are probably not different. Alternatively, the p-value for Market is almost 0, indicating that there is almost no chance that the means are equal for the different markets. Although this example does not attempt to analyze markets, there are significant differences among the three markets in average sales.

The following information is displayed in Output 2:



```
A                          B           C               D              E

Report Name:               Two-way ANOVA
Statistics:                ANOVA Summary

Variable                   Count       Mean            Variance       Std Deviation
After: After               4           32.25           358.92         18.95
After: After               4           71.50           193.67         13.92
After: After               4           52.25           201.58         14.20
Before: Before             4           31.50           115.00         10.72
Before: Before.            4           61.75            53.58          7.32
Before: Before             4           50.75           242.25         15.56

All Data                   24          50.50
```

In this report, the means for each region are nearly equal before and after the advertising campaign, except in the Los Angeles market. The Los Angeles market shows a definite increase in sales after the advertising campaign, but the p-values have not absolutely confirmed it.

Using ANOVA Statistics

Analysis of variance is an extension of the two-sample t-test. The t-test determines whether two-paired or independent samples have different distribution. The ANOVA method derives its name from the principle that a total sum of squares can be resolved into several component parts. There are different model types within analysis of variance: one-way ANOVA, two-way ANOVA, and three-way ANOVA.

One-way ANOVA deals with two columns of data. The first column, called the data column, contains the actual data observation values. The second column contains category (classification) data. The information in the category column is used to segregate the data observations into two or more data groups.

For example, in analyzing the results of an advertising campaign, the category values "before" and "after" can be used to differentiate data samples collected before the campaign began from the data samples collected after the campaign ended. The ANOVA transformer is designed to perform one-way analysis of variance for either balanced data that have an equal number of observations in each classification or unbalanced data that have an unequal number of observations in each classification.

The two-way and three-way ANOVA models use two or three levels of classification. The groups are defined by providing the ANOVA transformer with two or three category columns in addition to a data observations column. These additional category data columns are often referred to as treatment and replication columns.

The two- and three-way ANOVA statistics are computed in a manner similar to one-way ANOVA, but the calculations involve a more complex test scheme to compute differences between the various grouping levels, as well as among the groups. The ANOVA transformer performs two-way and three-way analysis of variance for balanced data that has an equal number of observations in each replication and between treatments. If the data being analyzed is not balanced, the ANOVA transformer will stop processing and issue an error message.

Linear contrast analysis is another way to examine the mean of each classification, with the magnitude of the differences among each classification. Contrast analysis is performed by defining a set of linear combinations of classifications. Specifically, the definition can be expressed as:

$$L = a_1 \times \bar{X}_1 + a_2 \times \bar{X}_2 + \ldots + a_k \times \bar{X}_k$$

where the a_k terms are user-defined constants and the X_k terms are the sample means. See “Specifying Linear Contrast Expressions” on page 230 for more information about linear contrast analysis.

Defining One-Way ANOVA Formulas

The symbol definitions shown in Table 33 on page 219 apply to the equations in this section.

Table 33. One-way ANOVA symbol definitions

Symbols	Definition
df_B	Degrees of freedom for between the group means
df_M	Degrees of freedom for the overall mean
df_T	Degrees of freedom for the entire data set
df_{TC}	Corrected degrees of freedom (excludes df for the means)
df_W	Degrees of freedom for within a group
F	One-way ANOVA F-statistic
I	Number of groups
J_i	Number of observations in the *i*th group
N	Total number of observations
S_B	Sum of squares calculated between groups

Table 33. One-way ANOVA symbol definitions

Symbols	Definition
$\bar{S}_B$	Mean of the sum of squares calculated between groups
S_M	Sum of squares calculated on the overall mean
S_T	Sum of squares calculated on the entire data set
S_{TC}	Corrected sum of squares (excludes sum of squares calculated on the means)
S_W	Sum of squares calculated within a group
$\bar{S}_W$	Mean of the sum of squares calculated within groups
Y	Overall mean of all observations
Y_i	Mean of each group
Y_{ij}	*j*the observation value in group i

In one-way ANOVA models, the means are calculated first using the following two formulas:

$$\bar{Y}_i = \frac{\sum_{j=1}^{J_i} Y_{ij}}{J_i}$$

$$\bar{Y} = \frac{\sum_{i=1}^{I} \sum_{j=1}^{J_i} Y_{ij}}{N}$$

The number of degrees of freedom calculated between groups is:

$$df_B = I - 1$$

The between-groups sum of squares is:

$$S_B = \sum_{i=1}^{I} J_i \quad (\bar{Y}_i - \bar{Y})^2$$

The between-groups mean sum of squares is:

$$\bar{S}_B = \frac{S_B}{df_B}$$

The within-groups degrees of freedom are:

$$df_W = N - I$$

The within-groups sum of squares is:

$$S_W = \sum_{i=1}^{I} \sum_{j=1}^{J_i} (Y_{ij} - \bar{Y})^2$$

The within-groups mean sum of squares is:

$$\bar{S}_W = \frac{S_W}{df_W}$$

The degrees of freedom for the overall mean are:

$$df_M = 1$$

The overall mean sum of squares is:

$$S_M = N \times \bar{Y}^2$$

The degrees of freedom for the entire data set are:

$$df_T = N$$

The corrected sums of squares for the entire data set are:

$$S_{TC} = \sum_{i=1}^{I} \sum_{j=1}^{J_i} Y_{ij}^2 - N \times \bar{Y}^2$$

The corrected degrees of freedom for the entire data set are:

$$df_{TC} = N - 1$$

The sum of squares for the entire data set is:

$$S_T = \sum_{i=1}^{I} \sum_{j=1}^{J_i} Y_{ij}^2$$

The one-way ANOVA F-value is:

$$F = \frac{\bar{S}_B}{\bar{S}_W}$$

The one-way ANOVA p-value can be obtained through a lookup in the F-statistic table using the F-value, df_B, and df_W. ANOVA performs this conversion automatically.

The one-way ANOVA null hypothesis is expressed as:

$$\mathrm{H}_0\colon \bar{\mathrm{Y}}_1 = \bar{\mathrm{Y}}_2 = \ldots = \bar{\mathrm{Y}}_\mathrm{I}$$

Examining the p-value helps determine whether the null hypothesis should be accepted or rejected. If the p-value is close to 0, it is unlikely that the F-value was found due to chance and thus, the null hypothesis should be rejected.

The format shown in Table 34 on page 222 is used to present one-way ANOVA results in the transformer's first output region. The mathematical symbols are defined in Table 33 on page 219.

Table 34. One-way ANOVA table format

Source	**Degrees of Freedom**	**Sum of Squares**	**Mean Sum of Squares**	**F-Value**	**p-Value**
Between	df_B	S_B	$\bar{S}_B$	F	p
Within	df_W	S_W	$\bar{S}_W$		
Total Correct	dfT_C	S_{TC}		df_M	S_M
Total	df_T	S_T	$\bar{S}_{TC}$Mean		

Defining Two-Way ANOVA Formulas

The symbol definitions in Table 35 apply to the equations in this section.

Table 35. Two-way ANOVA symbol definitions

Symbols	Definition
df_{C1}	Degrees of freedom for category column 1
df_{C2}	Degrees of freedom for category column 2
df_I	Degrees of freedom for the category interaction
df_{RES}	Degrees of freedom for the residual category
df_M	Degrees of freedom for the group means
df_T	Degrees of freedom for the entire data set
df_{TC}	Corrected degrees of freedom for the entire data set
df_W	Degrees of freedom for within a group
F_{C1}	Two-way ANOVA F-statistic for category column 1
F_{C2}	Two-way ANOVA F-statistic for category column 2
F_I	Two-way ANOVA F-statistic for category column interaction
I	Number of unique classes in category column 1
J	Number of unique classes in category column 2
K	Total number of observations in each unique class
N	Total number of observations
S_{C1}	Sum of squares for category column 1
S_{C2}	Sum of squares for category column 2
S_I	Sum of squares calculated for the category interaction
S_{RES}	Sum of squares calculated for the residual category
S_{TC}	Sum of squares calculated or the group means
$\bar{S}_{C1}$	Mean of the sum of squares for category column 1
$\bar{S}_{C2}$	Mean of the sum of squares for category column 2
$\bar{S}_I$	Mean of the sum of squares for the category interaction
$\bar{S}_{RES}$	Mean of the sum of squares for the residual category
$\bar{Y}$	Overall mean of all observations
$\bar{Y}_i$	Mean of each of the first category groups

Table 35. Two-way ANOVA symbol definitions

Symbols	Definition
$\overline{Y}_j$	Mean of each of the second category groups
$\overline{Y}_{ij}$	Mean of each combination of first and second category
Yijk	kth observation in the ith class of the first category and the jth class of the second category

In two-way ANOVA models, the means are calculated first using the following formulas:

$$\overline{Y}_{ij} = \frac{\sum_{k=1}^{K} Y_{ijk}}{K}$$

$$\overline{Y}_{i} = \frac{\sum_{j=1}^{J} \sum_{k=1}^{K} Y_{ijk}}{J \times K}$$

$$\overline{Y}_{j} = \frac{\sum_{i=1}^{I} \sum_{k=1}^{K} Y_{ijk}}{I \times K}$$

$$\overline{Y} = \frac{\sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} Y_{ijk}}{I \times J \times K}$$

The degrees of freedom are calculated by the following formulas:

$$df_{C1} = I - 1$$

$$df_{C2} = J - 1$$

$$df_I = df_{C1} \times df_{C2}$$

$$df_{RES} = I \times J \times (K - 1)$$

$$df_M = 1$$

$$df_T = I \times J \times K$$

$$df_{TC} = df_T - df_M$$

The sum of squares statistics are calculated by the following formulas:

$$S_{C1} = \sum_{i=1}^{I} (\bar{Y}_i - \bar{Y})^2 \times J \times K$$

$$S_{C2} = \sum_{j=1}^{J} (\bar{Y}_j - \bar{Y})^2 \times I \times K$$

$$S_I = \sum_{i=1}^{I} \sum_{j=1}^{J} (\bar{Y}_{ij} - \bar{Y}_i - \bar{Y}_j + \bar{Y})^2 \times K$$

$$S_{RES} = S_{TC} - S_I - S_{C2} - S_{C1}$$

$$S_M = I \times J \times K \times \bar{Y}^2$$

$$S_T = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} Y_{ijk}^2$$

$$S_{TC} = S_T - S_M$$

The mean of the sum of squares statistics are calculated by the following formulas:

$$\bar{S}_{C1} = \frac{S_{C1}}{df_{C1}}$$

$$\bar{S}_{C2} = \frac{S_{C2}}{df_{C2}}$$

$$\bar{S}_{I} = \frac{S_{I}}{df_{I}}$$

$$\bar{S}_{RES} = \frac{S_{RES}}{df_{RES}}$$

The F-values are calculated by the following formulas:

$$F_{C1} = \frac{\bar{S}_{C1}}{\bar{S}_{RES}}$$

$$F_{C2} = \frac{\bar{S}_{C2}}{\bar{S}_{RES}}$$

$$F_{I} = \frac{\bar{S}_{I}}{\bar{S}_{RES}}$$

The two-way ANOVA p-values are obtained through a lookup in the F-statistic table using each of the F-values along with df_{RES}, and one of the values df_{C1}, df_{C2}, and df_{I}. ANOVA automatically performs this conversion.

The two-way ANOVA null hypotheses are expressed as:

$$H_{0-C1}: \bar{Y}_1 = \bar{Y}_2 = ... = \bar{Y}_I$$

$$H_{0-C2}: \bar{Y}_1 = \bar{Y}_2 = ... = \bar{Y}_J$$

$$H_{0-I}: \bar{Y}_{11} = \bar{Y}_{12} = \bar{Y}_{21} = ... = \bar{Y}_{IJ}$$

Each null hypothesis can be tested by examining its corresponding p-value. If a p-value is close to 0, it is unlikely that its respective F-value was determined due to chance and thus, the associated null hypothesis should be rejected.

The two-way ANOVA format in Table 36 on page 228 is used to present two-way ANOVA results in the transformer's first output region. The mathematical symbols are defined in Table 35 on page 223.

Table 36. Two-way ANOVA table format

Source	Degrees of Freedom	Sum of Squares	Mean Sum of Squares	F-Value	p-Value
Category #1	df_{C1}	S_{C1}	$\bar{s}_{C1}$	F_{C1}	p_{C1}
Category #2	df_{C2}	S_{C2}	$\bar{s}_{C2}$	F_{C2}	p_{C2}
Interaction	df_{I}	S_{I}	$\bar{s}_{I}$	F_{I}	p_{I}
Residual	df_{RES}	S_{RES}	$\bar{s}_{RES}$		
Mean	df_{M}	S_{M}			
Total	df_{T}	S_{T}			

Defining Three-Way ANOVA Formulas

The formulas for three-way ANOVA are much the same as two-way, but have an additional dimension. Because the formulas are lengthy and the exact implementations are very complicated, the three-way ANOVA formulas are not explained here.

Table 37 and Table 38 on page 229 contain three-way ANOVA symbol definitions and table format.

Table 37. Three-way ANOVA symbol definitions

Symbols	Definition
df_{I-1}	Degrees of freedom for first category by second category interaction
df_{I-2}	Degrees of freedom for first category by third category interaction
df_{I-3}	Degrees of freedom for second category by third category interaction
df_{I-4}	Degrees of freedom for first by second by third category interaction
FI_{I-1}	F-statistic for the first by second category interaction
F_{I-2}	F-statistic for the first by third category interaction
F_{I-3}	F-statistic for the second by third category interaction
F_{I-4}	F-statistic for the first by second by third category interaction
p_{I-1}	Probability of the F-statistic for the first by second category interaction
p_{I-2}	Probability of the F-statistic for the first by third category interaction
p_{I-3}	Probability of the F-statistic for the second by third category interaction
p_{I-4}	Probability of the F-statistic for the first by second by third category interaction
S_{I-1}	Sum of squares for first category by second category interaction
S_{I-2}	Sum of squares for first category by third category interaction
S_{I-3}	Sum of squares for second category by third category interaction
S_{I-4}	Sum of squares for first by second by third category interaction
$\bar{S}_{I-1}$	Mean sum of squares for first by second category interaction
$\bar{S}_{I-2}$	Mean sum of squares for first by third category interaction
$\bar{S}_{I-3}$	Mean sum of squares for second by third category interaction
$\bar{S}_{I-4}$	Mean sum of squares for first by second by third category interaction

Table 38. Three-way ANOVA table format

Source	Degrees of Freedom	Sum of Squares	Mean Sum of Squares	F-Value	p-Value
Category #1	df_{C1}	S_{C1}	$\bar{S}_{C1}$	F_{C1}	p_{C1}
Category #2	df_{C2}	S_{C2}	$\bar{S}_{C2}$	F_{C2}	p_{C2}
Category #3	df_{C3}	S_{C3}	$\bar{S}_{C3}$	F_{C3}	p_{C3}
Interaction: Cat #1 x Cat #2	df_{I-1}	S_{I-1}	$\bar{S}_{I-1}$	F_{I-1}	p_{I-1}

Table 38. Three-way ANOVA table format

Source	Degrees of Freedom	Sum of Squares	Mean Sum of Squares	F-Value	p-Value
Interaction: Cat #1 x Cat #3	df_{I-2}	S_{I-2}	$\bar{S}_{I-2}$	F_{I-2}	p_{I-2}
Interaction: Cat #2 x Cat #3	df_{I-3}	S_{I-3}	$\bar{S}_{I-3}$	F_{I-3}	p_{I-3}
Interaction: Cat #1 x Cat #2 x Cat #3	df_{I-4}	S_{I-4}	$\bar{S}_{I-4}$	F_{I-4}	p_{I-4}
Residual	df_{RES}	S_{RES}	$\bar{S}_{RES}$		
Total Corrected	df_{TC}	S_{TC}			
Mean	df_M	S_M			
Total	df_T	S_T			

Specifying Linear Contrast Expressions

In one-way models, the ANOVA transformer computes linear contrasts if a linear contrast expression is supplied in the `Perform Linear Contrast Analysis` parameter. A linear contrast expression consists of two parts: a control value and an optional list of sample weight values.

The control value is either `yes`, `no`, or blank. If the value is `no` or blank, linear contrast analysis is not computed. A control value of `yes` indicates that linear contrasts should be computed.

Sample weight values indicate the amount of emphasis to be placed on a particular sample. In the following linear contrast expression:

$$L = a_1 \divideontimes \bar{X}_1 + a_2 \divideontimes \bar{X}_2 + ... + a_k \divideontimes \bar{X}_k$$

the a_1 values are the sample weights, and X_k represents each of the sample means. Linear contrast analysis requires that the following expression be satisfied:

$0 = a_1 + a_2 + ¼ + a_k$

Thus, the sum of all of the weights must be 0. The first sample weight value read from the `Linear Contrast Analysis` parameter is assigned to the first subclassification sample (X_1); successive weights are assigned to successive samples. For example, if the linear contrast of $X_1 + X_2 = 2X_3$ is desired, the expression must first be rewritten by moving all of the terms to one side:

$0 = X_1 + X_2 - 2X_3$

The linear contrast expression is:

```
yes,1,1,-2
```

Examples of each of the five possible variations of a linear contrast expression:

<no expression>
No linear contrasts are performed.

no
No linear contrasts are performed.

no,1,-1,2
No linear contrasts are performed.

yes
This is an error because sample weights are not provided.

yes,1,-2,0,-1,2
Linear contrast will be computed using the expression:

$$X_1 + 2X_5 = 2X_2 + X_4$$

This expression was obtained by rearranging the terms of the expression:

$$0 = X_1 - 2X_2 + 0X_3 - X_4 + 2X_5$$

ChiSquare

The ChiSquare transformer performs the chi-square test and the chi-square goodness-of-fit test. These statistics can help answer questions such as:

- Are the values of one variable associated with the values of another?
- Are the values of one variable independent of the values of a second variable?
- Is the distribution of variable values what we would expect?

These statistics are nonparametric tests and are well suited for situations when sample sizes are small or the variables under study might not be normally distributed. Both tests make the best use of data that cannot be precisely measured.

The chi-square and chi-square goodness-of-fit tests are designed to evaluate data organized in contingency or cross-tabulation tables. These tables are devices for organizing values of one or more variables. In organizing the values of only one variable, the cross-tabulation table consists of two columns with one or more rows of header or label information. The subsequent rows in the left column contain the different values of the variable; subsequent rows in the right column contain the number of observations that have each value. The last row contains a total count of the observations in all of the categories. The following example shows a single variable cross-tabulation table:

Variable:	Job	Number
	Sales	15
	Manager	7
	Clerk	38
	Other	5
	Total	65

When responses of two variables are organized, the top one or more rows of a cross-tabulation table contain table header information and labels for one variable and its values. The left columns contain label information for the second variable and its values. The bottom row and the right column contain total counts of observations in the given row or column. These row and column totals are also referred to as marginals or marginal counts.

The rest of the table consists of the intersection of values in the first variable with the values of the second variable. These intersections are referred to as cells. Cells contain the number of observations that correspond to the combination of a specific value of the first variable and a specific value of the second.

The following example shows a two-variable or two-dimensional cross-tabulation table. This table summarizes a sample containing 37 men and 28 women.

		Sex		
Job	Statistic	Male	Female	Total
Sales	Count	5	10	15
Manager	Count	5	2	7
Clerk	Count	23	15	38
Other	Count	4	1	5
Total	Count	37	28	65

The previous contingency table was created with the CrossTab transformer, which is described in “CrossTab” on page 245. The ChiSquare transformer can read tables in this format, as well as tables in other formats.

Parameters

All of the parameters described in this section are displayed in the ChiSquare Transformer Controls window. Most of these parameters are also displayed in the Data Entry icon contained in the transformer's demonstration capsule application. However, to simplify use of the transformer for new users, some of the more advanced parameters might be omitted from the Data Entry icon; they can be accessed only in the Transformer Controls window.

Number of header rows (; 1; 5;) row containing title (,1; ,5;)

This parameter specifies the number of input rows to be skipped and not used in calculating the chi-square statistics. This parameter also specifies the row number containing column titles to be used as labels in the output. The two values should be separated by a comma.

You can specify any number of header rows. The default value is 0. The number of title rows should be less than or equal to the number of header rows. If you specify the number of header rows, the default title row is the last row of the header rows. If no header row is specified, the default is no title row.

Report name (; ChiSquare Test;)

This parameter is a title for the output report, for example, *chiSquare Example*. If `Report Name` is blank, there will be no title in Output 1.

Data columns (a,b; a,b,c; a:e;)

This parameter specifies the column or columns containing the cell counts used to compute the goodness-of-fit or chi-square statistics. One or more data columns must be specified. Each column should contain cell counts.

If your table includes a column of row totals, that column should not be included in the list of columns. If the `chiSquare Statistics` parameter is specified as `Multiple` and you want the standard chi-square test, all of the columns specified are considered as part of the same table.

In contrast, if the `ChiSquare Statistics` parameter is specified as `One Sample`, which requests the goodness-of-fit test, each column is considered as a separate table and a different analysis is completed for each column.

Data rows (1,2; 1,2,3; 1:20;)

This parameter specifies two or more rows of cell counts used to compute the goodness-of-fit or chi-square statistics. If your table includes a row of column totals, that row should not be included in the list of columns. For example, if your input table has four rows and three columns of cell counts, excluding column and row totals, you would specify `1-4` in this parameter.

Treat data as 'One' sample or 'Multiple' samples (; One; Multiple;)

This parameter specifies whether the chi-square goodness-of-fit test or the standard chi-square test will be run. `One Sample` causes the goodness-of-fit test to run, because that test uses only one sample or column of data. `Multiple` causes the standard chi-square to run, because that test uses multiple columns or samples. The default value is `Multiple`.

Expected distribution for 'One Sample' test (; 5; 5,6,4;)

This parameter specifies the cell counts used as the expected distribution in the chi-square goodness-of-fit test. Cell counts should be integers that are separated by commas. You can specify the expected count of every cell.

If you specify fewer counts than the number of cells in the table, the last specified count is used for all remaining cells. For example, if there are three cells and the entry is `12, 10`, the transformer uses an expected count of 12 for the first cell and an expected count of 10 for the last two cells.

The total of all specified cell counts should be equal to the total cell counts in the observed table. If the totals differ, a warning is displayed. If no counts are specified, the median of the observed cell counts is used. This parameter is ignored if you request the standard chi-square.

Group column as (; a; b,male,female,only;)
This parameter allows segregation of input data into a set of user-specified groups or ranges that are treated as separate data sets. If the `Group Column As` parameter is blank, one group is created that contains all of the input data.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- Results (Output 1)
- Messages (Output 2)

When you select any one of these choices, the data associated with that choice displays in the display area of the transformer.

Input Region Names

The ChiSquare transformer has only one input region, Input 1 (Data). You can modify the input name to reflect the name of the corresponding data in the display area. Input 1 consists of data read from a Spreadsheet, Query, or SQL Entry tool, or data copied directly into the transformer. Input 1 contains a series of columns containing cell counts (Data Columns). If extra columns are present, they are ignored. The input data can have any number of header rows. Column titles are read from the specified row. If titles are not found, default titles are provided.

Output Region Names

The ChiSquare transformer generates two output regions. Both regions have no size limit.

You can modify the output name to reflect the name of the corresponding data in the display area.

Output 1 (Results), which is the test output region, consists of the following information:

- Number of rows in the table
- Number of columns in the table
- Degrees of freedom associated with the table
- Raw chi-square test statistic
- Significance level associated with the raw chi-square statistic
- Yates' chi-square statistic
- Significance level associated with the Yates' chi-square statistic

Output 2 (Messages) contains:

- Transformer run-time messages
- Warnings
- Error messages
- A timestamp for documentation purposes

One-Sample Chi-Square Example

One year ago, a marketing firm conducted a survey of its employees. One of the findings of the survey was that many of the employees felt that the company did not recognize their contributions. The question that measured this opinion asked employees how well the company recognized their contributions. The valid responses to that question were five ordinal values ranging from 1 to 5. A response of 5 indicated that the employee felt that the company did very well at recognizing employee contributions. In contrast, a response of 1 indicated that the employee felt that the company did very poorly at recognizing their contributions. The responses to this question and the number of employees who chose them are shown here:

Response	No. of Employees
Very Poorly	17
Poorly	40
Adequately	38
Well	21
Very Well	4
Total	120

Because employees' perception of their company's recognition of their contribution is an important component of morale, the company initiated a program aimed at improving morale that included awards to employees that saved the company money, recognition awards to employees that had been with the company over five years, and an employee newsletter. Now, the company's human resources department wants to know whether the company's efforts to improve employee morale have been effective. To help gauge this, the department surveys employees again and asks the same questions. The responses to the recognition question in the second survey are in a one-dimensional cross-tabulation table:

Variable:	Recognition
Very Poorly	13
Poorly	28
Adequately	43
Well	33
Very Well	3
Total	120

By comparing these two tables, the company's human resources department can tell that in the current survey more respondents said that the company did an adequate or good job of recognizing their contributions. Also, in the recent survey, fewer people said that the company did a poor or very poor job of recognizing their work.

To know how likely it is that these differences happened by chance, the table containing responses to the most recent survey is copied into Input 1 of the ChiSquare transformer.

The cell counts from last year's survey are included in the `One Sample Expected Distribution` parameter. The one-sample ChiSquare transformer parameter settings are set as follows:

Number of header rows, row containing title
1, 1

Report name
One-sample ChiSquare Example

Data Columns
b

Data rows
2-7

Treat data as 'One' sample or 'Multiple' samples
One sample

Expected distribution for 'One Sample' test
17, 40, 38, 21, 4

When the transformer runs, the following information is displayed in Output 1:

```
A                    B               C              D
Project:             One-sample ChiSquare Example
One-sample Variable  No. of Employees

Statistics           df              Value          Sig-p
No. Rows                             6
No. Columns                          1
ChiSquare            4               12.31          0.02
Yates ChiSquare      4               10.92          0.03
```

It is apparent from this information that the differences in responses between last year's and this year's recognition question are unlikely to be caused by chance. Thus, the human resources department can conclude that the program to improve morale is having its intended effect. This year, more employees feel that the company adequately recognizes their contributions.

Two-Sample Chi-Square Example

Although the human resources staff is happy that more employees believe the company recognizes their contributions, they wonder if the department in which an employee works has any bearing on the employee's feelings of recognition. To determine whether such differences exist, a new cross-tabulation table is constructed from the most recent survey responses. That table is two-dimensional: degree of recognition is one dimension and the department in which an employee works is the other dimension:

Variable:			
Recognition	Marketing	EDP/Acctng	Total
Very Poorly	4.00	9.00	13.00
Poorly	9.00	19.00	28.00
Adequately	14.00	29.00	43.00
Well	21.00	12.00	33.00
Very Well	2.00	1.00	3.00
Total	50.00	70.00	120.00

From the input data, it is apparent that there are some differences between the two groups. Individuals in the marketing department seem more likely to believe that the company does a good job at recognizing their efforts. At the same time, individuals in the EDP and Accounting groups are less likely to feel that the company does a good job at recognizing their contributions.

The two-sample ChiSquare transformer parameter settings are set as follows:

Number of header rows, row containing title
2, 2

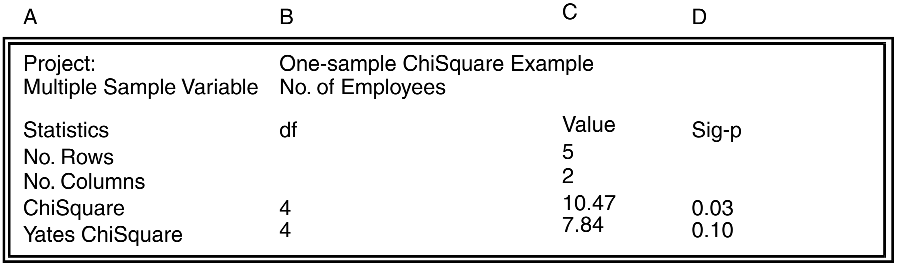
Report name
One-sample ChiSquare Example

Data Columns
b:c

Data rows
3-7

Treat data as 'One' sample or 'Multiple' samples
Multiple

When the transformer runs, the following information is displayed in Output 1:



```
A                         B                              C              D

Project:                  One-sample ChiSquare Example
Multiple Sample Variable  No. of Employees

Statistics                df                             Value          Sig-p
No. Rows                                                 5
No. Columns                                              2
ChiSquare                 4                              10.47          0.03
Yates ChiSquare           4                              7.84           0.10
```

This output indicates that there is reason to believe that there is an association between department membership and feelings of recognition. The uncorrected significance level of 0.03 indicates that there is only a 3% likelihood that the observed difference could have happened by chance. However, the 0.10 significance level associated with the Yates chi-square indicates that there is a 10% chance that there is no difference between the two groups. When the two significance levels are quite different, it is important to know which one is more accurate. Information in Output 2 can help make this decision. The contents of that output region are shown in the following example:

```
Date: July 18, 1990

Time: 15:52:11

Warnings and Error messages:

Warning: (2x2) Contingency Cell[6][1] = 1.2500 with expected value > 5.0

Table: Cell[6][2] = 1.7500 with expected value > 5.0
```

Because this output indicates that two of the ten table cells had expected values of less than 5, the Yates chi-square statistic and its significance level are more appropriate measures of association.

The human resources department believes that it is riskier to accept the null hypothesis than to reject it. They believe that the costs associated with not finding a difference even when one exists (such as higher turnover in the EDP and Accounting departments due to morale problems) are higher than the costs associated with finding a difference when one does not exist. As a result, they conclude that even though there is a 10% likelihood that the observed difference occurred due to chance, there is enough evidence to reject the null hypothesis and accept the alternate hypothesis that there is an association between department membership and feelings of recognition. As a result, the human resources department will begin investigating reasons for this association and try to improve the feelings of recognition among the EDP and Accounting staff.

The Chi-Square Goodness-of-Fit Test

Data used in the chi-square goodness-of-fit test must be organized in cross-tabulation tables and must be measured on the nominal or ordinal level. Nominal data has values that can be differentiated but cannot be arranged in any particular order. While the categories can have numeric values, those values have no inherent meaning. Gender is an example of a nominal form of data. Ordinal data has values that can be arranged from lowest to highest. An example of an ordinal form of data is a movie rating scale that runs from 1, which indicates a very bad movie, to 5, which indicates a very good movie. Although this test can accommodate higher level data, other statistical methods are usually more appropriate.

It is sometimes important to learn how many observations fall into various categories. For example, a market researcher might be interested in the number of respondents that are familiar or unfamiliar with a certain product. The chi-square goodness-of-fit test measures the degree of correspondence between the actual number of observations and the expected number of observations in one or more categories.

The null hypothesis of the chi-square goodness-of-fit test is that there is no difference between the actual and expected number of observations in each category. The expected counts in each category are usually specified by the user and are often based on historical information or on the results of other statistical techniques. The total of the expected cell counts must equal the total of the observed cell counts. If they are different, the test might incorrectly find significant differences among distributions that are actually similar. If you specify expected cell counts that do not add up to the observed total, a warning is sent to Output 2.

To test the null hypothesis, the chi-square test statistic is calculated for the cross-tabulation table as follows:

- For each category or table cell, the expected value is subtracted from the actual value.
- For each cell, the result of that subtraction is squared and divided by the expected count.
- The results of this division are summed to arrive at the chi-square value.

- To find out whether the chi-square value is significant, the degrees of freedom associated with the cross-tabulation table must be known. The degrees of freedom are always one less than the number of categories.
- After the chi-square value and the degrees of freedom have been calculated, the chi-square transformer calculates the probability of obtaining the observed chi-square given the degrees of freedom.

If the probability, which is issued as a significance level, is close to one, it indicates that the chi-square statistic probably occurred by chance and that the actual and expected distributions are very similar. However, if the significance level is close to 0, the observed chi-square value was unlikely to occur by chance, and there is a real difference between the expected and actual distributions. If the actual and expected distributions are the same, the conclusion is that they fit.

Chi-Square Goodness-of-Fit Test Formulas

The definitions shown in Table 39 apply to the equations in this section.

Table 39. Goodness-of-fit test symbol definitions

Symbol	Definition
A_i	Actual count for any given cell
c^2	Chi-square test statistic for the entire table
$c^2{}_Y$	Yates' corrected chi-square statistic for the entire table
D_i	Residual or difference between the actual and expected count for any given cell
df	Degrees of freedom for the entire table
E_i	Specified expected count for any given cell
N	Number of cells in the table

The residual or difference between the actual and expected count for any given cell is calculated as follows:

$$D_i = A_i - E_i$$

The raw chi-square test statistic for the table is calculated as follows:

$$\chi^2 = \sum_{i=1}^{N} \frac{D_i^2}{E_i}$$

If D_i is greater than 0, the corrected chi-square test statistic is calculated as follows:

$$\chi_Y^2 = \sum_{i=1}^{N} \frac{|(D_i - 0.5)|^2}{E_i}$$

If D_i is less than 0, the corrected chi-square test statistic is calculated as follows:

$$\chi_Y^2 = \sum_{i=1}^{N} \frac{|(D_i + 0.5)|^2}{E_i}$$

The degrees of freedom associated with the table are calculated as follows:

$$df = N - 1$$

The ChiSquare transformer then estimates the probability of obtaining a chi-square statistic as large as the observed statistic, given the degrees of freedom. If that probability, which is returned as a significance level, is close to 0, the observed chi-square statistic did not occur due to chance and the actual cell counts are different from the expected cell counts. Conversely, if the significance level is close to one, the observed chi-square statistic is likely to occur by chance and there is no real difference between the expected and actual cell counts.

The Standard Chi-Square Test

Like the goodness-of-fit test, the standard chi-square test requires ordinal or nominal level data organized in a cross-tabulation table. The chi-square test also analyzes differences between actual and expected cell counts. While the goodness-of-fit test determines if the observed cell counts vary from the expected cell counts, the chi-square test looks for associations between variables. An association occurs when observations with a given value for one variable are more likely to have the specific value of the second variable. An association between two variables indicates that the variables are related; when two variables are not associated, they are independent. Thus the chi-square test indicates whether variables are associated or independent. Although the chi-square test can tell whether an association exists, it cannot determine the strength of that association.

To determine whether an association exists, the transformer calculates a chi-square test statistic. Whereas the goodness-of-fit test uses specified expected counts, the chi-square test calculates expected cell counts based on the row and column totals associated with any cell. This expected cell count represents the number of observations expected in a cell if there was no association between the row and column variables. After the expected cell counts are calculated, they are subtracted from the actual counts, and the differences are then squared and divided by the expected count. The results of these calculations for each cell are then summed across the entire table to arrive at the chi-square test statistic.

The ChiSquare transformer then estimates the probability of obtaining the observed chi-square value given the degrees of freedom. The degrees of freedom are equal to one less than the number of rows multiplied by one less than the number of columns. If the probability is very large, the observed chi-square probably occurred due to chance and the two variables are independent. If the probability or significance level is very small, the observed chi-square statistic probably did not occur due to chance and there is an association between the two variables.

Chi-Square Test Formulas

The definitions shown in Table 40 apply to the equations in this section.

Table 40. Chi-square test symbol definitions

Symbol	Definition
A_{rc}	Actual count for any given cell
C	Number of columns in the table
c^2	Chi-square test statistic for the entire table
c^2_Y	Yates' corrected chi-square statistic for the entire table
D_{rc}	Residual or difference between the expected and actual count for any given cell
df	Degrees of freedom for the entire table
E_{rc}	Specified expected count for any given cell
N	Number of cells in the table
R	Number of rows in the table
S_r	Sum of counts for any given row
S_c	Sum of counts for any given column
S_t	Sum of counts for the entire table

The expected count for any given cell is calculated as follows:

$$E_{rc} = \frac{S_r \times S_c}{S_t}$$

The residual or difference between the actual and expected count for any given cell is calculated as follows:

$$D_{rc} = A_{rc} - E_{rc}$$

The raw chi-square test statistic for the table is calculated as follows:

$$\chi^2 = \sum_{r=1}^{R} \sum_{c=1}^{C} \frac{D_{rc}^2}{E_{rc}}$$

If D_{rc} is greater than 0, the corrected chi-square test statistic is calculated as follows:

$$\chi_Y^2 = \sum_{r=1}^{R} \sum_{c=1}^{C} \frac{|D_{rc} - 0.5|^2}{E_{rc}}$$

If D_{rc} is less than 0, the corrected chi-square test statistic is calculated as follows:

$$\chi_Y^2 = \sum_{r=1}^{R} \sum_{c=1}^{C} \frac{|D_{rc} + 0.5|^2}{E_{rc}}$$

The degrees of freedom associated with the table are calculated as follows:

$$df = (R-1) \quad (C-1)$$

The ChiSquare transformer then estimates the probability of obtaining a chi-square statistic as large as the observed one, given the degrees of freedom. If that probability, which is returned as a significance level, is close to 0, the observed chi-square statistic did not occur due to chance and the two variables are associated. Conversely, if the significance level is close to one, the observed chi-square statistic is likely to occur by chance, and the two variables are independent.

Yates' Correction

When any of the expected cell counts are small (less than 5), the estimation of the chi-square distribution might not be completely accurate. To improve the accuracy of the test, the ChiSquare transformer uses a technique for correcting the chi-square test statistic called Yates' correction. This correction subtracts 0.5 from the absolute value of the difference between the actual and expected cell counts.

After that point, the calculation of the chi-square statistic, degrees of freedom, and significance are the same as with the raw chi-square test. The transformer always provides a corrected chi-square statistic and its significance and labels them as the Yates' chi-square. Whenever an expected cell count drops below 5, the transformer sends a warning to Output 2. Consequently, if the table has relatively low actual cell counts, it is a good idea to check the Output 2 region. If there are warnings, use the Yates' chi-square.

Specifying Data Columns and Rows

To specify the input columns containing data, enter a list, a range of columns, or both, using either the letters or the numbers associated with the columns. For example, if the input data is in the first three columns of a spreadsheet, a `Data Columns` list specification would be `a,b,c` or `1,2,3`. A list of columns is simply a series of column letters or numbers separated by commas. The columns specified need not be contiguous. For example, if the `Data Columns` specification is `a,c`, the transformer gathers the data from the first and third columns of the input region.

The `Data Columns` parameter also accepts ranges of columns. A range of columns consists of the number or letter associated with first data column, a colon, and the letter or number associated with the last data column. For example, if the transformer should use the first five columns of data, the `Data Columns` specification would be `1:5` or `a:e`.

The `Data Columns` parameter also accepts a combination of lists and ranges. For example, if the input data occurs in the first, second, and fourth through sixth columns, the parameter specification would be `a,b,d:f` or `1,2,4:6`.

Like the `Data Columns` parameter, the `Data Rows` parameter accepts either lists or ranges of data rows. The data row lists or ranges must consist of the numbers associated with the rows of the input region. A list form of the specification is simply the row numbers separated by columns. The row numbers need not be contiguous. For example, if the input data occurred in the first, third, and fifth rows of the input region, the `Data Rows` specification would be `1,3,5`.

A range of data rows consists of the first row of data, a colon, and the last row of data. For example, if the input data is located in the first 10 rows of the input data, the `Data Rows` range specification would be `1:10`. Like the `Data Columns` parameter, the `Data Rows` parameter also accepts combinations of lists and ranges. For example, if the data occurs in the second, fourth, and sixth through tenth rows of the input region, the `Data Rows` specification would be `2,4,6:10`.

Specifying Grouping Features

A grouping expression consists of three parts: a group column, an optional list of grouping criteria, and an optional group type specifier. Each group is treated as a distinct set of data; a set of chi-square tables is generated for every specified group.

The group column is a single data column containing information that determines the group to which a particular data element belongs. For example, if the first column of the input data contains the grouping information, enter `a`, for column A.

A list of grouping criteria can follow the column name to specify the groups that are created. The criteria can be a list of text values, numeric values, or dates, each separated by a comma. If grouping criteria are not present, the ChiSquare transformer creates groups for every unique value of the group column.

The group-type specifier controls whether the grouping criteria are treated as members of a group or limits of a range. If the type specifier *only* is present, a group is created for each item in the grouping criteria list. Only values that exactly match a particular grouping criterion are added to the corresponding group. If the type specifier is not present, the grouping criteria are treated as the end points of a series of ranges. Any value greater than the first end point and less than or equal to the second end point is treated as part of that particular range.

Examples of each of the four possible variations of a grouping expression are:

<no expression>
One group is created containing all of the input data

a A group is created for each unique value in column A

a, 10, 20
Three ranges are created: all values up to and including 10, all values above 10 but less than or equal to 20, and all values above 20

a, 10, 20, only
Two groups are created: all values where column A is 10, and all values where column A is 20

CrossTab

The CrossTab transformer performs two-dimensional cross-tabulation. The statistics computed include:

- Count
- Sum
- Mean
- Row totals
- Column totals
- Percentage of all data (based on count or sum)
- Percentage of selected data (based on count or sum)
- Percentage of row (based on count or sum)
- Percentage of column (based on count or sum)
- Expected value (based on count or sum)

- Residual value (based on count or sum)
- Chi-square (based on count or sum)
- Chi-square level of significance

When a Query tool is connected to the input of this transformer, cross-tabulation statistics can be generated for virtually an unlimited number of rows of data, because the Query tool can retrieve data one row at a time and feed it to the transformer. Cross-tabulation statistics never require more than one input row to be stored in the workstation; therefore, results can be computed for very large amounts of data. The transformer can perform these calculations very quickly, thus it has a speed advantage over the Spreadsheet tool.

Because the CrossTab transformer does not require sorted input data, a significant amount of time is saved compared to other data grouping methods that require sorted data. CrossTab can group date, text, numerical, or even Boolean data (true/false answers).

Parameters

All of the parameters described in this section are displayed in the Transformer Controls window. Most of these parameters are also displayed in the Data Entry icon contained in the transformer's demonstration capsule application. However, to simplify use of the transformer for new users, some of the more advanced parameters might be omitted from the Data Entry icon; they can be accessed only in the Transformer Controls window.

The CrossTab transformer has four groups of parameters:

- User
- Primary Grouping
- Secondary Grouping
- Expert

Although the choices might seem complex, many parameters are of the "set and forget" type. After they are set, only a few parameter values must be adjusted.

Certain parameters accept dates. All Meta5 date formats are supported. To enter a date, enclose it in double quotation marks. For example, the second day of March, 1990, would be entered as `"March 2, 1990"`. Using quotation marks tells the transformer to ignore the comma in the date.

User Parameters

Number of header rows (1; 2;)

This parameter allows a header to precede the data in Input 1. The first row of the header, if it exists, labels the results. If no labels are found, default labels are used.

Report name (CrossTab Test;)

This parameter, consisting of up to 100 characters, identifies the results in Output 1.

Data columns (a; a,b; b,c,a;)

This parameter specifies the columns with which to calculate statistics. The column names to enter are the spreadsheet column names, such as `a`, `b`, or `ac`. You can specify any number of columns, in any order, as long as each name is separated by a comma. For example, `d, h, aa, b` asks for statistics on the fourth, eighth, twenty-seventh, and second columns.

Cross-tabulation

This parameter allows you to ask for a specific type of cross-tabulation descriptive statistic; enter its name into the `Cross-Tabulation` parameter field. Type `Count, Sum, Mean` to compute the count, sum and mean of each Data Column, respectively. Enter `all` to compute each type of cross-tabulation for each data column. The Cross-Tabulation names cannot be abbreviated because they are not unique. Using a Data Entry tool as a control panel allows the user to select any of these statistics without having to enter the names into the transformer.

Table 41 shows all available cross-tabulation statistics.

Table 41. Available cross-tabulation statistics

Symbol	Definition
all	All cross-tabulation statistics
chisqc	Chi-square based on count (with degrees of freedom and significance)
chisqs	Chi-square based on sum (with degrees of freedom and significance)
count	Count
expectedc	Expected count
expecteds	Expected sum
mean	Mean
residualc	Residual count
residuals	Residual sum
sum	Sum
%ac	% of all data based on count
%as	% of all data based on sum
%pc	% of primary (row) based on count
%ps	% of primary (row) based on sum
%sc	% of secondary (column) based on count

Table 41. Available cross-tabulation statistics

Symbol	Definition
%ss	% of secondary (column) based on sum
%tc	% of selected data based on count
%ts	% of selected data based on sum

Primary and Secondary Grouping Parameters

The grouping features allow the user to specify how the input data is categorized. Primary grouping parameters break a data file into groups that are presented as a series of rows in the CrossTab output. Secondary grouping parameters break each group into subgroups that are presented as a series of columns in the cross-tabulation output. Provisions also exist to examine data that does not fall into a group or subgroup. Data that does not fit into a particular category is labeled *Other*. All grouping features function as described, even if the input data is not sorted.

Primary Grouping Parameters: Primary selection column (a; b;)

This is the data column containing Cross-Tab row (stub) heading. This parameter is the column used to group the data into row categories. This column can contain any type of data; data types can be mixed within this column. Only one column can be specified.

Primary selection criteria (5,10,15; male,female;)

This forms the Cross-Tab row heading. This parameter contains the values of the categories used to group the input data. One group is created for each item in this parameter; all data matching a given criterion is included in the corresponding group. The criteria specified can be of any data type, including dates; different types can be mixed. If no `Primary Selection Criteria` values are specified, all groups will be automatically located.

Break primary selection into groups or ranges? (g; r) sort selection criteria (y; n)

This parameter modifies the treatment of the `Primary Selection Criteria` values. If `group` is the entry, each group consists of data that exactly matches its corresponding criterion. If `range` is the entry, each group consists of data less than or equal to the criterion value.

For example, if the values `21345, 90125` are entered as the criteria for a grouping column containing zip codes, and `group` is the entry for this parameter, two groups will be created: one for data rows that have a zip code of 21345, and the other for data rows that have a zip code of 90125. If `range` is the entry instead, three groups will be created: the first for data rows with zip codes up through 21345, the second for rows with zip codes above 21345 and less than or equal to 90125, and the last group for data rows with zip codes above 90125.

All ranges exclude the starting value and include the ending value. The default is `g,n`.

To compare data of differing types, a hierarchy establishes when items are considered less than or equal. Data from a Spreadsheet icon is arranged with numbers first, followed by letters, and then dates. The Query tool provides access to a wider variety of data formats. These enhanced data types, listed first to last, are:

- Integer numbers
- Real numbers
- Text strings
- Date
- Boolean
- Unspecified
- Error
- N/A
- Full-date

Two additional rules augment this hierarchy. First, only the numerical values of integer and real numbers are compared so that values such as 2 and 2.00 are in the same group. Second, only the day value of dates and full-dates are compared to eliminate any differences between the internal formats.

Show 'Items Not Selected' in primary selection? (y; n)

This parameter allows the user to enable or disable displaying the items not selected, that is, items that did not fit into one of the primary groups. This option is disabled when ranges are selected or all groups are found, because there are no items unselected in either of these cases. The default value is `n`.

Secondary Grouping Parameters: The secondary grouping feature breaks the primary groups into subgroups. The secondary grouping parameters are similar to the corresponding primary grouping parameters.

Secondary selection column (a; b;)

This is the data column containing Cross-Tab column (banner) heading. This parameter is the column that groups each primary group/range into subcategories that are represented as columns in a cross-tabulation table. This column can contain any type of data; data types can be mixed. Only one column can be specified.

Secondary selection criteria (5,10,15; male,female;)

This forms the Cross-Tab column heading. This parameter contains the names of the categories for each subgroup in the input data. One subgroup is created for each item in this parameter; all data that matches a given criterion is included in the corresponding subgroup. The criteria can be of any data type, including dates; different types can be mixed. If no `Secondary Selection Criteria` values are specified, all groups are automatically located.

Break secondary selection into groups or ranges? (g; r) sort selection criteria (y; n)

This parameter works like the primary option, but uses the secondary criteria.

Show 'Items Not Selected' in secondary selection? (y; n)

This parameter allows the user to enable or disable displaying the items not selected, that is, items in a particular primary group that did not fit into one of the secondary subgroups. This option is disabled when secondary ranges are selected or all secondary groups are found, because there are no items not selected in either of these two cases. The default value is `n`.

Expert Parameters

The expert parameters let you adjust features of each of the output regions. Answering `yes` to a question enables the feature; answering `no` disables the feature.

Create 'Output 1', 'Output 2', 'Output 3'? (y,y,y; n,n,n)

This parameter lets you turn off a particular output region if it is not required, to increase the speed of the transformer and save file storage space. For example, if only Output 2 is needed, type `no, yes, no` to turn off Output 1 and Output 3.

If plots are not being created, turn off Output 3 to save time; the amount of information provided to the user is not reduced.

Show groups/ranges with zero values in Outputs 2, 3? (y,y; n,n)

This parameter controls the printing of data groups that have no members (all statistics are 0). Answering `no` to this question means that all such groups will not be printed. This feature is controllable for each output. Therefore, `no, yes` means that zero values are shown for Output 2, but not Output 3.

To produce graphs with a consistent look between different program runs, do not turn off zero values in Output 3. When a data group is 0 and the zero values are turned off, the plot's X-axis shows fewer items than it does when the data group is not 0.

In 'Output 2', insert blank lines after Sub-Totals? (y; n) suppress repeated headings? (y; n)

These parameters control the appearance of the report output. Answering `yes` to the first question adds blank lines after each subtotal, thus increasing readability. Answering `yes` to `Suppress Repeated Heading` causes the name of each group to be printed only once. For example, if a particular result is displayed as follows:

Market	Date	Statistic	Volume
Atlanta	1Q88	Count	10
Atlanta	1Q88	Sum	15
Atlanta	2Q88	Count	20
Atlanta	2Q88	Sum	25

The same result will be displayed as follows when repeated headers are suppressed:

Market	Date	Statistic	Volume
Atlanta	1Q88	Count	10
		Sum	15
	2Q88	Count	20
		Sum	25

For maximum readability, turn both parameters on if the report is printed. If the report output is used for further analysis in another transformer or capsule application, turn both parameters off; the missing data items and blank rows can cause problems. The default value is `n,n`.

In 'Output 3', show totals? (y; n) show sub-totals? (y; n)

This parameter determines which data is sent to the plot output. Answering `yes` to `Show Totals` adds the final total of all data read to the graph output. Answering `yes` to `show sub-totals` adds the subtotal of each primary group to the graph. The default value is `n,n`.

Answering `no` to both questions and answering `yes` to `Show Groups/Ranges with Zero Values in Output 3` results in output with one row for each combination of primary and second grouping, which is necessary for further analysis.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1
- Table (Output 1)
- Report (Output 2)
- Plot (Output 3)
- Messages (Output 4)

When you select any one of these choices, the data associated with that choice displays in the display area of the transformer.

Input Region Names

The CrossTab transformer has only one input region, Input 1 (Data). You can modify the input name to reflect the name of the corresponding data in the display area. Input 1 contains columns of data that are cross-tabulated. It must contain at least three columns: two grouping columns and one or more data columns. If extra columns are present, they are ignored. The data can have any number of header rows. The contents of the first header row are used for labeling the results. If the first header row is empty, default labels are provided. Missing or non-numeric values, including N/A and Error in input data columns, are detected and reported in Output 4.

Output Region Names

The CrossTab transformer has four output regions:

- Output 1 (Table) contains the report title, the type of statistical report, a cross-tabulation table for each input data column, and chi-square table, if requested.
- Output 2 (Report) contains the same data as Output 1 in a report-oriented format.
- Output 3 (Plot) contains most of the information found in the first two outputs; however, the information is formatted for compatibility with the Plot tool. Each data group is displayed in one spreadsheet row and has a unique identifier. The unique identifier can be a sequence number from 1 to the number of groups, or a name formed from the data group names. Output 3 is also suitable for input into other statistical transformers for further analysis.
- Output 4 (Messages) contains notes, warnings, and error messages issued by the transformer. When applications are running in a capsule environment, you can save all messages by connecting this output regtion to a Spreadsheet or Text icon.

Examples

The following data file contains sales and revenue information for three market segments of a corporation. The data is spread over five periods, encompassing 41 rows. Because the data was abstracted from a sales database, each sale returns one row from the query.

Period	Market	Revenue
1	Chi	3,356
1	Chi	32,453,452
1	Chi	34,453,452
1	Chi	33,453,452

1	Chi	31,453,452
1	LA	24,562,632
1	LA	1,345
1	Min	234,756,458
1	Min	35,453,452
1	Min	4,356
1	Min	78,975,856
2	Chi	22,462,364
2	Chi	134,567,987
2	LA	47,768
2	LA	456,257,626
2	Min	23,462,364
3	Chi	345,234
3	LA	38,576,768
3	LA	55,467,458
3	Min	3,436,326
3	Min	24,678
4	Chi	45,634,562
4	Chi	44,634,562
4	Chi	46
4	LA	245,642,564
4	LA	45,768
4	Min	2,436,326
4	Min	37,453,452
4	Min	346
4	Min	36,453,452
4	Min	47,634,562
5	Chi	42,634,562
5	Chi	43,634,562
5	Chi	41,634,562
5	Chi	42
5	LA	145,642,564
5	LA	46,768

5	Min	2,345
5	Min	1,436,326
5	Min	46,634,562
5	Min	53,674,568

In this example, the CrossTab transformer parameters were set as follows:

Number of header rows
1

Report name
Multi-Market Example

Data columns
c

Cross-tabulation
Sum

Primary selection column
b

Break primary selection into groups or ranges?
g, y

Show 'Items Not Selected' in primary select?
y

Secondary selection column
a

Break secondary selection into groups or ranges? sort selection criteria
g, y

Show 'Items Not Selected' in secondary selection?
y

Create 'Output 1', 'Output 2', 'Output 3'?
y, y, y

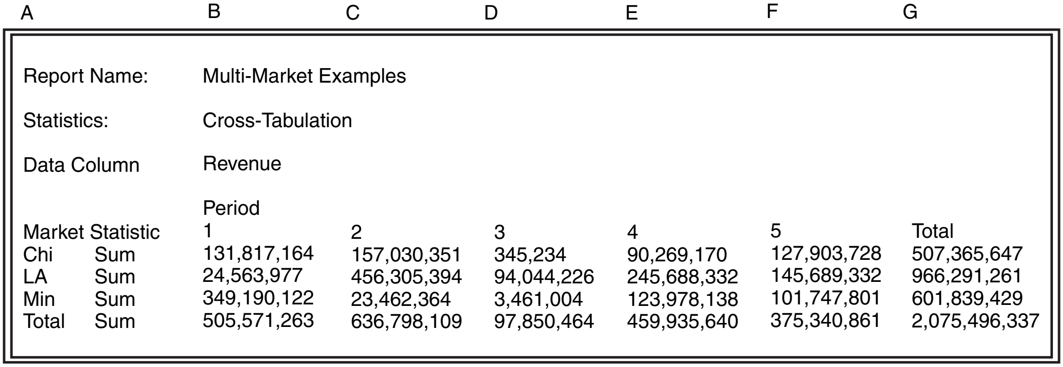
Show groups/ranges with zero values in Outputs 2, 3?
y, y

In 'Output 2', insert blank lines after Sub-Totals? suppress repeated headings?
y, y

In 'Output 3', show totals? show sub-totals?
y

Cross-tabulating data can yield a large amount of information, especially if different views of the data are presented. The following cross-tabulation presents revenue by period and region, with totals for each period and market. The cross-tabulation was performed by requesting the sum statistic using column B as the primary grouping column, column A as the secondary grouping column, and column C as the data column. No group or range criteria were specified, and the group/range feature was set to `group`.

After the transformer runs, the following information is displayed in Output 1:



```
A              B              C             D             E             F             G

Report Name:   Multi-Market Examples

Statistics:    Cross-Tabulation

Data Column    Revenue

               Period
Market Statistic 1            2             3             4             5             Total
Chi    Sum     131,817,164    157,030,351   345,234       90,269,170    127,903,728   507,365,647
LA     Sum     24,563,977     456,305,394   94,044,226    245,688,332   145,689,332   966,291,261
Min    Sum     349,190,122    23,462,364    3,461,004     123,978,138   101,747,801   601,839,429
Total  Sum     505,571,263    636,798,109   97,850,464    459,935,640   375,340,861   2,075,496,337
```

This result shows that Period 3 had much lower revenue than other periods and that the Los Angeles market revenue was almost as great as the other two markets combined. Additionally, a significant drop in revenue in the Chicago market during Period 3 might be worth investigating.

When cross-tabulation tables like the previous one are considered, new questions often arise. For example, what is the distribution of sales within different dollar value ranges? The data summarized in the previous table can be used to answer this question by modifying the CrossTab transformer as follows:

Number of header rows
1

Report name
Multi-Market Example

Data columns
c

Cross-tabulation
Count

Primary selection column
c

Primary selection criteria
10000000, 50000000

Break primary selection into groups or ranges?

r, y

Show 'Items Not Selected' in primary select?

y

Secondary selection column

b

Break secondary selection into groups or ranges? sort selection criteria

g, y

Show 'Items Not Selected' in secondary selection?

y

Create 'Output 1', 'Output 2', 'Output 3'?

y, y, y

Show groups/ranges with zero values in Outputs 2, 3?

y, y

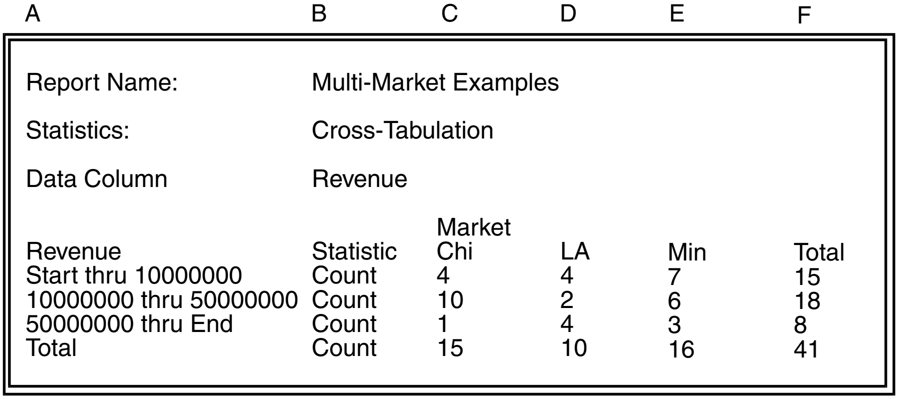
In 'Output 2', insert blank lines after Sub-Totals? suppress repeated headings?

y, y

In 'Output 3', show totals? show sub-totals?

y

After the transformer runs, the following information is in Output 1:



A	B	C	D	E	F
Report Name:	Multi-Market Examples				
Statistics:	Cross-Tabulation				
Data Column	Revenue				
		Market			
Revenue	Statistic	Chi	LA	Min	Total
Start thru 10000000	Count	4	4	7	15
10000000 thru 50000000	Count	10	2	6	18
50000000 thru End	Count	1	4	3	8
Total	Count	15	10	16	41

The primary column was set to C, with ranges set at $10,000,000 and $50,000,000. The secondary column was set to B, and C was again used as the data column. The count statistic indicates the number of sales contributed by each region. The cross-tabulation table makes it apparent that the Los Angeles region has fewer sales overall, but many of its sales were very large. This fact is important because the previous table shows that the Los Angeles region had the most total revenue on the least number of sales.

CrossTab

The CrossTab transformer was used to obtain another pertinent view of the same data. To obtain the other view, the parameters were set as follows:

Number of header rows
1

Report name
Multi-Market Example

Data columns
c

Cross-tabulation
Count, Chisqe

Primary selection column
c

Primary selection criteria
10000000, 50000000

Break primary selection into groups or ranges?
r, y

Show 'Items Not Selected' in primary select?
y

Secondary selection column
a

Break secondary selection into groups or ranges? sort selection criteria
g, y

Show 'Items Not Selected' in secondary selection?
y

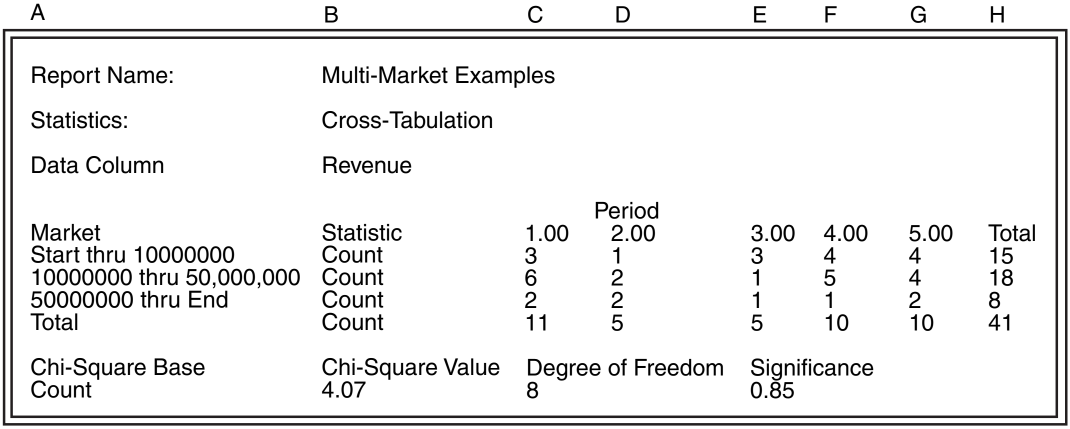
Create 'Output 1', 'Output 2', 'Output 3'?
y, y, y

Show groups/ranges with zero values in Outputs 2, 3?
y, y

In 'Output 2', insert blank lines after Sub-Totals? suppress repeated headings?
y, y

In 'Output 3', show totals? show sub-totals?
y

After the transformer runs, the following information is in Output 1:



```
A                           B                  C        D               E        F        G        H

Report Name:                Multi-Market Examples

Statistics:                 Cross-Tabulation

Data Column                 Revenue

                                                          Period
Market                      Statistic          1.00     2.00            3.00     4.00     5.00     Total
Start thru 10000000         Count              3        1               3        4        4        15
10000000 thru 50,000,000    Count              6        2               1        5        4        18
50000000 thru End           Count              2        2               1        1        2        8
Total                       Count              11       5               5        10       10       41

Chi-Square Base             Chi-Square Value   Degree of Freedom        Significance
Count                       4.07               8                        0.85
```

This final view compares the size of each sale to the period of the sale. This table was computed using the same primary column and ranges as the previous view, and Column A was used as the secondary column. This table shows that sales of various sizes are evenly distributed, except that small and medium sales were off during period 2 and period 3. The cause of this variation might be an unusual outside incident (such as a strike), the result of an internal problem (a quality control problem in a factory causing a product shortage), or a seasonal variation.

To investigate this difference in sales during different periods, note the chi-square table shown in the output. The chi-square value is a relatively low 4.07. The probability value indicates an 85% chance that the variation in the sales numbers is due to randomness. Although the changes in sales composition during periods 2 and 3 might be interesting, there is no real reason to be concerned with these variations, because they are probably random. Based on the data available, the conclusion is that the size of the sale might be sensitive to the market, but there is no evidence that it is associated with sale period.

Cross-Tabulation Tables

The CrossTab transformer condenses data into user-defined groups for analysis or input to other Meta5 statistical transformers. Two grouping levels should be specified. For example, sales information can be grouped by product and market to determine the relative contribution of each product on sales within each market. In the cross-tab format, all statistics are computed for each cell of a table. Each table cell represents the combination of a primary grouping variable value and a secondary grouping variable value. In a cross-tabulation table, the primary grouping variable defines table rows, and the secondary variable defines table columns.

For example, a table representing sales by product and market has one row for every product and one column for every market. Each cell contains sales levels for each product in each market. One cell might represent sales of Product A in the Minneapolis market, another cell might represent sales of Product B in the Minneapolis market, and yet another cell might represent the sales of Product A in New York.

CrossTab Statistics

The CrossTab transformer computes a variety of statistics. The basic statistics are:

count
The total number of times a particular combination of primary and secondary column values occur.

sum
The summation of the data elements associated with a particular combination of primary and secondary column values.

mean
The ratio of sum to count (arithmetic mean).

Many of the statistics calculated by the CrossTab transformer are also calculated by the Elementary transformer. However, the CrossTab transformer presents them in cross-tabulation table formats not supported by the Elementary transformer. The CrossTab transformer provides counts, sums, and means in the formats listed here:

Primary or row statistic
Calculated for each primary grouping independent of the secondary grouping variable

Secondary or column statistic
Calculated for each secondary grouping variable independent of the primary grouping variable

Total statistics
Calculated for the entire table independent of the primary and secondary grouping variables

Using row, column, and table totals, the CrossTab transformer computes several types of percentage statistics:

row percent
Calculated by dividing the cell count or sums by its respective row count or sum.

column percent
Calculated by dividing the cell count or sum by its respective column count or sum.

total percents
Calculated in two different ways:

- The percent of total can be calculated by dividing the cell value by all data, including the data not selected. This measure is known as percent of all data.
- The percent of total can be calculated by dividing the cell value by only the selected data in the table. This measure is known as percent of selected data.

Expected values and residual values are higher level statistical analyses. The row, column, and table totals are used to calculate the cell values that are expected if the row and column variables are unrelated. The residual value is the difference between the expected cell value and the actual value.

The highest level statistic computed by CrossTab is the chi-square goodness-of-fit value. Chi-square, computed from the expected and residual values, provides a single value that summarizes the entire data set. A small chi-square value indicates that the residual values are relatively small; a larger chi-square value indicates that the residuals are relatively large.

The CrossTab transformer automatically converts the chi-square value into a probability of the likelihood that the observed chi-square occurred by chance. If the probability is near one, the observed chi-square is likely to occur by chance, and the row and column variables are independent of each another. Conversely, if the probability is near 0, the chi-square is unlikely to occur by chance, and the row and column variables are associated with each other.

Whenever the transformer encounters a non-numeric input value in a column that is specified as a data column, the entire row is excluded from the analysis. Thus, that row is not included in cell, row, column, or table summary statistics. In contrast, if the transformer encounters an invalid value in a grouping column, it creates a new row or column for that value.

Formulas for CrossTab Statistics

The definitions in Table 42 apply to the formulas in this section.

Table 42. CrossTab formula symbol definitions

Statistic	Definition
c	Total table count
c_i	Observed total count for any given row
c_j	Observed total count for any given column
c_{ij}	Observed count for any given cell
$c^2{}_c$	Chi-square value for the table count
$c^2{}_s$	Chi-square value for the table sum
ec_{ij}	Expected count for any given table cell

Table 42. CrossTab formula symbol definitions

Statistic	Definition
es_{ij}	Expected sum for any given table cell
I	Number of rows in the table (primary grouping values)
J	Number of columns in the table (secondary grouping values)
m	Total table mean
n	Number of data elements (the number of rows of data)
rc_{ij}	Residual count for any given table cell
rs_{ij}	Residual sum for any given table cell
s	Total table sum
s_i	Observed total sum for any given row
s_j	Observed total sum for any given column
s_{ij}	Observed sum for any given cell
x_n	Any given observed data element

The value for the table count is:

$$c = n$$

The value for the table total sum is calculated as follows:

$$s = x_1 + x_2 + \ldots + x_n$$

The value for the table total mean is calculated as follows:

$$\bar{m} = \frac{s}{n}$$

The value for any given cell (c_{ij}) is simply the number of observations with grouping values that match the column and row location of the cell.

The value for any given row total count is calculated as follows:

$$c_i = \sum_{j=1}^{J} c_{ij}$$

for `i = 1, 2, ¼, I`

The value for any given column total count is calculated as follows:

$$c_j = \sum_{i = 1, j}^{I} c_{ij}$$

for `j = 1, 2, ¼, J`

For any given cell, the value of the expected count is calculated as follows:

$$ec_{ij} = \frac{(c_i \times c_j)}{c}$$

for `i = 1, 2, ¼, I; j = 1, 2, ¼, J`

For any given cell, the value of the residual count is calculated as follows:

$$rc_{ij} = c_{ij} - ec_{ij}$$

for `i = 1, 2, ¼, I; j = 1, 2, ¼, J`

For the entire table, the chi-square value based on cell counts is calculated as follows:

$$\chi_c^2 = \sum_{i = 1}^{I} \sum_{j = 1}^{J} \frac{rc_{ij}^2}{ec_{ij}}$$

For any given cell, the sum (s_{ij}) is calculated by adding together the data values of observations having grouping values that match the column and row location of the cell.

The value for any given row total sum is calculated as follows:

$$s_i = \sum_{i, j = 1}^{J} s_{ij}$$

for `i = 1, 2, ¼, I`

The value for any given column total sum is calculated as follows:

$$s_j = \sum_{i = 1, j}^{I} s_{ij}$$

for `j = 1, 2, ¼, J`

For any given cell, the value of the expected sum is calculated as follows:

$$es_{ij} = \frac{(s_i \times s_j)}{s}$$

for `i = 1, 2, ¼, I; j = 1, 2, ¼, J`

For any given cell, the value of the residual sum is calculated as follows:

$$rs_{ij} = s_{ij} - es_{ij}$$

for *i = 1, 2, , I; j = 1, 2, , J*

For the entire table, the chi-square value based on cell sums is calculated as follows:

$$\chi_s^2 = \sum_{i = 1}^{I} \sum_{j = 1}^{J} \frac{rs_{ij}^2}{es_{ij}}$$

The degrees of freedom value for the table is calculated as follows:

$$df = (I - 1) \times (J - 1)$$

The CrossTab transformer uses the chi-square value and the degrees of freedom to compute a probability value.

IPMean

The IPMean (independent paired mean) transformer compares distributions of two samples. The IPMean transformer handles data in the following formats:

- Two or more samples with equal numbers of observations
- Two or more samples with unequal numbers of observations
- Matched pair samples (for example, before-after types of data)

IPMean

The IPMean transformer is based on the t-statistic. The t-statistic measures how two populations are related by checking whether the data values in each sample overlap in a particular fashion. The t-statistic is well suited for small samples. Many statistical tests require that the data samples be normally distributed. In practice, samples of less than 30 observations generally cannot be expected to be normally distributed.

IPMean computes two types of t-statistics: the independent t-test and the matched pair t-test. The matched pair t-test examines differences between individual pairs of observations in two samples. The independent t-test examines differences between two samples taken as a whole. The independent t-test is widely used when there is no suitable basis for pairing.

Parameters

All of the parameters described in this section are displayed in the Transformer Controls window. Most of these parameters are also displayed in the Data Entry icon contained in the transformer's demonstration capsule application. However, to simplify use of the transformer for new users, some of the more advanced parameters might be omitted from the Data Entry icon; they can be accessed only in the Transformer Controls window.

Number of header rows (; 1; 5;) row containing titles (; ,1; ,5;)

This parameter specifies the number of rows in the input data that are skipped and not used in calculating statistics. This parameter also specifies the row number containing column titles that are used as labels in the output. The two values should be separated by a comma.

You can specify any number of header rows; the default value is 0. The title row number should be less than or equal to the number of header rows. If the header row's value is specified, the default title row is the last row of the header rows. If no header is specified, the default is no title row.

Report name (; IPMean Test;)

This parameter is a title for the output report, for example, *T-statistic tests*. If `Report name` is blank, there will be no title in Output 1.

Data columns (a,b; a,b,c;)

This parameter specifies the columns used to compute the t-test statistics. At least two data columns are required, but any number of data columns can be specified. For example, `a,b` causes the transformer to gather the data from columns A and B.

Treat data as 'Independent' samples or 'Paired' samples (; Independent; Paired;)

This parameter specifies whether matched pair t-test statistics or independent t-test statistics are to be computed. Valid values are `Independent` and `Paired`, which can be abbreviated to `i` or `p`. The default value is `Independent`.

Group column as (; a; b,male,female,only;)

This parameter segregates the input data into a set of user-specified groups or ranges. If `Group Column As` is blank, one group is created, containing all of the input data.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- Results (Output 1)
- Messages (Output 2)

When you select any one of these choices, the data associated with that choice displays in the display area of the transformer.

Input Region Names

The IPMean transformer has only one input region, Input 1 (Data). You can modify the input name to reflect the name of the corresponding data in the display area. Input 1 consists of data read from a Spreadsheet, Query, or SQL Entry tool, or data copied directly into the transformer. Input 1 contains a series of columns containing data (data columns) or grouping information (Group Column as). If extra columns are present, they are ignored. The input data can have any number of header rows. Column titles are read from the specified row. If titles are not found, default titles are provided.

Output Region Names

The IPMean transformer generates two output regions. Each output has no size limit.

Output 1 (Results) is a report of statistics computed for each variable (input data column). The statistics include:

- Variable names
- Count
- Mean
- Standard deviation
- Number of degrees of freedom
- t-value
- p-value (significance level)
- F-statistic (independent method only)
- F-significance value (independent method only)

Output 2 (Messages) contains:

- Transformer run time messages
- Warnings
- Error messages

Independent T-Test Example

Two groups of rats are each fed a high-calorie diet. The diet of one group contains a higher amount of protein, whereas the second group receives a low-protein diet. Does the protein content of the food affect the growth rate of the rats?

At the start of the test, there were 15 rats in each group. During the test, several of the rats died, and several more were disqualified from the test for other reasons. The weight gain (in grams) of the remaining rats is summarized. Notice that the two samples no longer have an equal number of observations.

Low Protein	High Protein
70.00	134.00
118.00	146.00
101.00	104.00
85.00	119.00
107.00	124.00
132.00	161.00
94.00	107.00
	83.00
	113.00
	129.00
	97.00
	123.00

This data is presented to Input 1 of the IPMean transformer. The IPMean transformer parameters are set as follows:

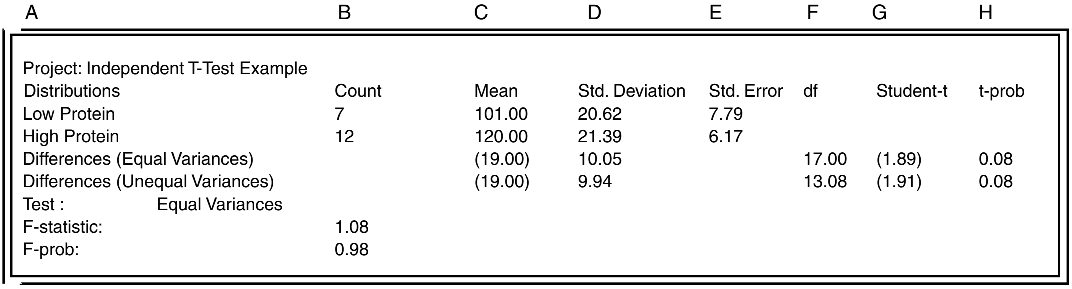
Number of header rows
1, 1

Report name
Independent T-Test Example

Data column
a, b

Treat data as 'Independent' samples or 'Paired' samples
Independent

When the transformer run finishes, the following report is displayed in Output 1:



```
A                                  B               C             D               E           F       G            H

Project: Independent T-Test Example
Distributions                      Count           Mean          Std. Deviation  Std. Error  df      Student-t    t-prob
Low Protein                        7               101.00        20.62           7.79
High Protein                       12              120.00        21.39           6.17
Differences (Equal Variances)                      (19.00)       10.05                       17.00   (1.89)       0.08
Differences (Unequal Variances)                    (19.00)       9.94                        13.08   (1.91)       0.08
Test :              Equal Variances
F-statistic:                       1.08
F-prob:                            0.98
```

In this result, the standard deviation values are not exactly the same. The F-significance value of 0.98 implies that there is an 98% chance that the standard deviation values are statistically the same. Thus, the t-values and p-values computed under the equal variances assumption should be used.

The high-protein diet showed a slightly greater average weight gain. For the assumption of unequal variances, the t-value is -1.91 and the p-value is 0.08. There is only an 8% chance that the null hypothesis is true. At the 90% confidence level, the conclusion is that the dietary protein level affects the growth rate of a rat.

Matched Pair T-Test Example

An employee of the XYZ Corporation can choose one of two delivery routes. Route A consists mainly of city streets; route B involves driving on a highway. Route A is shorter, but the travel speeds are generally much lower than on route B. Which route has the shortest travel time?

To answer this question, the employee conducts an experiment. Every day for a week, he tosses a coin to choose route A or route B. Each day of the following week, he takes the other route home. The experiment is then repeated to add extra reliability to the results. The following data is collected; Route A and Route B numbers are time in minutes.

Date of Week	Rout A	Rout B
Monday	28.7	25.4
Tuesday	24.8	24.9
Wednesday	25.1	23.9
Thursday	26.1	26.6
Friday	30.3	28.8

Monday	26.2	25.8
Tuesday	25.3	25.0
Wednesday	23.9	23.3
Thursday	25.8	24.8
Friday	31.4	30.3

The route data is sent to Input 1 of the IPMean transformer. The IPMean transformer parameters are set as follows:

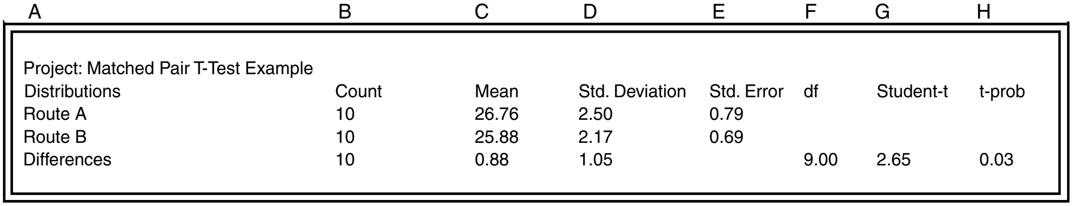
Number of header rows
1, 1

Report name
Matched Pair T-Test Example

Data column
b, c

Treat data as 'Independent' samples or 'Paired' samples
Paired

When the transformer run finishes, the following report is displayed in Output 1:



```
 A                              B            C          D            E         F       G          H

Project: Matched Pair T-Test Example
Distributions                  Count        Mean       Std. Deviation  Std. Error  df      Student-t  t-prob
Route A                        10           26.76      2.50           0.79
Route B                        10           25.88      2.17           0.69
Differences                    10           0.88       1.05                       9.00    2.65       0.03
```

In this result, the t-value is rather large and the p-value is small. Because the null hypothesis is that there is no difference between route A and route B, and there is only a 2.6% chance that the null hypothesis is true, there must be a significant difference in the travel time required for each route. Examining the means shows that route B requires less travel time.

Independent T-Test Statistics

The independent t-test uses data that is randomly chosen (sampled) from two or more populations. All samples can have the same number of observations or different numbers of observations; IPMean accurately handles either type of data.

The data samples are compared by computing a t-value based on several descriptive statistics, including the sample means and the population variances. Because the population variances are generally not known, they are estimated using the sample variance (variance computed for each sample) and the pooled variance (variance computed for all observations). The pooled variance can be calculated two different ways, depending upon whether the variances of each population are the same or different. IPMean displays both results, along with the resulting t-values.

To determine which t-value to use, IPMean computes the F-statistic for the observed variances. A large F-significance value indicates that the t-value computed for different variances should be used.

The null hypothesis is that the two samples were chosen from two populations that are statistically identical. This hypothesis can be tested by converting the t-statistic into a probability factor (p-value). The p-value gives the probability used to decide whether to accept the null hypothesis. A p-value near one means that the samples represent identical populations, whereas a p-value near 0 means that the populations are different.

Matched Pair T-Test Statistics

The matched pair t-test uses data that is randomly chosen (sampled) from two or more populations. In addition, each observation in one sample is related in some manner to an observation in each of the other samples. Data samples taken before and after an event fit this category. For example, assume an assembly line machine is adjusted during a weekend. To test if the adjustment has an impact on production, the number of units of output for the previous and following week can be compared. The data for the Monday before the adjustment and the Monday after the adjustment can be paired, and so on for the other six observations.

Two assumptions are built into the matched pair t-test:

- The differences between individual pairs of observations are distributed about some mean value. This mean is assumed to represent the average differences in the populations represented by the samples.
- The differences between any pair of observations and the mean difference are assumed to be normally and independently distributed with a population mean equal to 0.

In most situations, the value of population variance will not be known. Given the two assumptions, the population variance can be estimated.

The null hypothesis is that the differences between the two samples are statistically insignificant. This hypothesis can be tested by converting the t-statistic into a probability factor (p-value). The p-value gives the probability used to decide whether to accept the null hypothesis.

Formulas for Independent T-Test Statistics

The definitions in Table 43 apply to the formulas in this section.

Table 43. Independent T-test statistics symbol definitions

Symbol	Definition
A	A column of data representing a series of observations
A_i	A particular observation in sample A
$\bar{A}$	Mean of sample A
B	A second column of data representing a series of observations
B_i	A particular observation in sample B
$\bar{B}$	Mean of sample B
D	Difference between the two sample means
df_{I-EQ}	Degrees of freedom assuming equal variances
df_{I-NE}	Degrees of freedom assuming unequal variances
df_A	the degrees of freedom for sample A
df_B	the degrees of freedom for sample B
E_A	Standard error of sample A
E_B	Standard error of sample B
F_I	the F-statistic value for the independent method
N	Total number of observations ($n_A + n_B$)
n_A	Number of observations in sample A
n_B	Number of observations in sample B
S_A	Standard deviation of sample A
S_A^2	Variance of sample A
S_B	Standard deviation of sample B
S_B^2	Variance of sample B
S_{EQ}	The pooled standard deviation assuming equal population variances
S_{EQ}^2	The pooled variance assuming equal population variances
S_{NE}	The pooled standard deviation assuming unequal population variances
S_{NE}^2	The pooled variance assuming unequal population variances
t_{I-EQ}	The t-statistic value assuming equal population variances
t_{I-NE}	The t-statistic value assuming unequal population variances

The sample means are calculated as follows:

$$\bar{A} = \frac{\sum_{i=1}^{n_A} A_i}{n_A}$$

$$\bar{B} = \frac{\sum_{i=1}^{n_B} B_i}{n_B}$$

The difference between the sample means, for equal or unequal variances, is calculated as follows:

$$D = \bar{A} - \bar{B}$$

The sample variances are calculated as follows:

$$S_A^2 = \frac{\sum_{i=1}^{n_A} (A_i - \bar{A})^2}{n_A - 1}$$

$$S_B^2 = \frac{\sum_{i=1}^{n_B} (B_i - \bar{B})^2}{n_B - 1}$$

The sample standard deviations are calculated as follows:

$$S_A = \sqrt{S_A^2}$$

$$S_B = \sqrt{S_B^2}$$

The sample standard errors are calculated as follows:

$$E_A = \frac{S_A}{\sqrt{n_A}}$$

$$E_B = \frac{S_B}{\sqrt{n_B}}$$

The pooled variance of the independent samples method is calculated as follows, assuming that the populations have the same variances:

$$S_{EQ}^2 = \frac{[(n_A - 1) \times S_A^2] + [(n_B - 1) \times S_B^2]}{(n_A + n_B - 2)} \times \frac{n_A + n_B}{n_A \times n_B}$$

The pooled variance of the independent samples method is calculated as follows, assuming that the populations have different variances:

$$S_{NE}^2 = \frac{S_A^2}{n_A} + \frac{S_B^2}{n_B}$$

The pooled standard deviation, assuming equal variances, is calculated as follows:

$$S_{EQ} = \sqrt{S_{EQ}^2}$$

The pooled standard deviation, assuming unequal variances, is calculated as follows:

$$S_{NE} = \sqrt{S_{NE}^2}$$

Assuming that population variances are equal, the t-statistic is calculated as follows:

$$t_{I-EQ} = \frac{\bar{A} - \bar{B}}{S_{EQ}}$$

Assuming that population variances are not equal, the t-statistic is calculated as follows:

$$t_{I-NE} = \frac{\bar{A} - \bar{B}}{S_{NE}}$$

Assuming that the population variances are equal, the degrees of freedom are calculated as follows:

$$df_{I-EQ} = n_A + n_B - 2$$

Assuming that the population variances are not equal, the degrees of freedom are calculated as follows:

$$df_{I-NE} = \frac{[S_{NE}^2]^2}{\frac{S_A^4}{n_A^2 \times (n_A - 1)} + \frac{S_B^4}{n_B^2 \times (n_B - 1)}}$$

The paired t-test null hypothesis is given by the expression:

$$H_0: \bar{A} - \bar{B} = 0$$

The p-value gives the probability used to decide whether to accept the null hypothesis. The p-value is obtained by locating the t-statistic value in a t-table using the appropriate degrees of freedom. The IPMean transformer automatically performs this conversion.

The F-statistic, used to determine whether variances are equal or unequal, is calculated using a method that maximizes the value of F. If the variance for sample A is larger than the variance for sample B, F_I is calculated as:

$$F_I = \frac{S_A^2}{S_B^2}$$

If the variance for sample B is larger than the sample A variance, F_I is calculated as:

$$F_I = \frac{S_B^2}{S_A^2}$$

The degrees of freedom for the two samples are calculated as:

$$df_A = n_A - 1$$

$$df_B = n_B - 1$$

The F-significance value is obtained by locating the F_I value in an F-statistic table given df_1 and df_2, where df_1 represents the degrees of freedom for the sample with the largest variance and df_2 represents the degrees of freedom for the other sample. The IPMean transformer automatically performs this conversion.

Formulas for Matched Pair T-Test Statistics

The definitions in Table 44 apply to the equations in this section.

Table 44. Matched pair t-test statistics symbol definitions

Symbol	Definition
A	A column of data representing a series of observations
A_i	A particular observation in sample A
$\bar{A}$	Mean of sample A
B	A second column of data representing a series of observations
B_i	A particular observation in sample B
$\bar{B}$	Mean of sample B
d_i	Difference between observations in a particular pair
$\bar{D}$	Mean of the differences between the two samples
df_P	Degrees of freedom for the paired method
E	Standard error of the paired samples
N	Number of observation pairs
S	Standard deviation of the paired samples
S^2	Variance of the paired samples
S_A	Standard deviation of sample *A*
S_A^2	Variance of sample A
S_B	Standard deviation of sample B
S_B^2	Variance of sample B
t_P	The t-statistic value for the paired method

The sample means are calculated as follows:

$$\bar{A} = \frac{\sum_{i=1}^{N} A_i}{N}$$

$$\bar{B} = \frac{\sum_{i=1}^{N} B_i}{N}$$

The sample variances are calculated as follows:

$$S_A^2 = \frac{\sum_{i=1}^{N} (A_i - \bar{A})^2}{N - 1}$$

$$S_B^2 = \frac{\sum_{i=1}^{N} (B_i - \bar{B})^2}{N - 1}$$

These values are displayed as standard deviations by the IPMean transformer. The standard deviations are calculated as follows:

$$S_A = \sqrt{S_A^2}$$

$$S_B = \sqrt{S_B^2}$$

The difference between each pair value is calculated as follows:

$$d_i = A_i - B_i$$

The mean of the differences is calculated as follows:

$$\bar{D} = \frac{\sum_{i=1}^{N} d_i}{N}$$

The variance of the differences is calculated as follows:

$$S^2 = \frac{\sum_{i=1}^{N} (d_i - \bar{D})^2}{N - 1}$$

The standard deviation of the differences is calculated as follows:

$$S = \sqrt{S^2}$$

The standard error of the differences is calculated as follows:

$$E = \frac{S}{\sqrt{N}}$$

The t-statistic is calculated as follows:

$$t_P = \frac{\bar{D}}{\bar{E}}$$

The number of degrees of freedom is calculated as follows:

$$df_P = N - 1$$

The paired t-test null hypothesis is given by the expression:

$$\mathrm{H}_0 : \bar{\mathrm{D}} = 0$$

The p-value gives the probability used to decide whether to accept the null hypothesis. The p-value is obtained by locating the t_P value in a t-statistic table using df_P degrees of freedom. The IPMean transformer automatically performs this conversion.

Specifying Grouping Features

A grouping expression consists of three parts: a group column, an optional list of grouping criteria, and an optional group type specifier. Each group is treated as a distinct set of data, and a set of t-test tables is generated for every requested group.

The group column is a single column containing the information that determines the group to which a particular data element belongs. For example, if the first column of the input data contains the grouping information, the entry would be `a`, for column A.

A list of grouping criteria can follow the column name. Grouping criteria allow you to specify the groups that are created. The criteria can be a list of text values, numerical values, or dates, each separated by a comma. If grouping criteria are not present, the IPMean transformer creates groups for every unique value of the group column.

The group type specifier controls whether the grouping criteria are treated as members of a group or limits of a range. If the type specifier `only` is present, a group is created for each item in the grouping criteria list. Only values that exactly match a particular grouping criterion are added to the corresponding group. If the type specifier is not present, the grouping criteria are treated as the end points of a series of ranges. Any value that is greater than the first end point and less than or equal to the second end point is treated as part of that particular range.

Examples of each of the four possible variations of a grouping expression follow:

<no expression>
One group is created, containing all of the input data.

a
A group is created for each different value in column A.

a, 10, 20
Three ranges are created: all values up to and including 10, all values above 10 but less than or equal to 20, and all values above 20.

a, 10, 20, only
Two groups are created: all values where column A is 10, and all values where column A is 20.

Kruskal

The Kruskal transformer uses the Kruskal-Wallis test of medians to analyze sample distributions. The test determines if samples have different medians. If the medians are different, the inference is that the samples were drawn from different populations. The transformer helps answer questions such as:

- Are there differences among the distributions of several samples?
- Are the samples drawn from different populations?

- Which samples are different from others?

Because the Kruskal-Wallis statistic is a nonparametric test, it is ideal for situations when the assumption of normally distributed data does not hold. The Kruskal-Wallis statistic uses more information and is thus more likely to find actual differences than other nonparametric tests.

Parameters

All of the parameters described in this section are displayed in the Transformer Controls window. Most of these parameters are also displayed in the Data Entry icon contained in the transformer's demonstration capsule application. However, to simplify use of the transformer for new users, some of the more advanced parameters might be omitted from the Data Entry icon; they can be accessed only in the Transformer Controls window.

Number of header rows (; 1; 5;) row containing title (; ,1; ,5;)

This parameter specifies the number of rows in the input data that are skipped and not used in calculating statistics. This parameter also specifies the row number containing column titles that are used as labels in the output. The two values should be separated by a comma.

You can specify any number of header rows; the default value is 0. The title row number should be less than or equal to the number of header rows. If the header rows value is specified, the default title row is the last row of the header rows. If no header is specified, the default is no title row.

Report name (; Kruskal Test;)

This parameter is a title for the output report; for example, *Kruskal-Wallis Example*. If `Report Name` is blank, there is no title in Output 3.

Column containing category information (a; b; c;)

This parameter specifies the column that contains values used to identify mutually exclusive samples or groups of observations. Only one column can be specified. If more than one is specified, any column after the first is ignored. For example, the entry would be `a` if column A identifies the samples.

Data column (a; b; c;)

This parameter specifies the column containing the observation values used to compute the Kruskal-Wallis statistics. One or more columns should be specified. An analysis for each data column is provided in each of the output regions. For example, if column B in Input 1 contains data values, the entry would be `b`.

Compute multiple comparisons analysis (;y; n)

This parameter specifies whether this type of analysis is completed. The allowable values are `yes` and `no`. If `yes` is specified, Kruskal compares each pair of groups in the input data. The default value for this parameter is `no`.

Contrast analysis (; "1,2+3,4";)

This parameter specifies pairs of groups to be compared. It also allows the user to specify whether groups should be combined before the comparison. The user can specify up to 10 different sets of contrasts. If this parameter is blank, no contrast analysis is completed.

Double quotation marks are required in this parameter.

Significance value to use in computations (; 0.70; 0.75; 0.80; 0.85; 0.90; 0.95; 0.99)

This parameter specifies a value used by the transformer in contrast analysis and multiple comparisons to decide whether differences between two samples are significant. Legal values for this parameter are 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, and 0.99. If you specify any value other than these, the transformer uses the next highest value. For example, if you specify `0.82`, the transformer uses 0.85. The default value is 0.90.

Compute analysis of variance (; y; n)

This parameter specifies whether standard one-way ANOVA statistics and sample summary statistics, such as mean, standard deviation, and variance are computed and sent to Output 2. If you specify `yes`, the transformer provides those statistics. If you specify `no`, the transformer does not provide those statistics. The default response is `no`.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- Summary (Output 1)
- Results (Output 2)
- Analysis (Output 3)
- Messages (Output 4)

When you select any one of these choices, the data associated with that choice displays in the display area of the transformer.

Input Region Names

The Kruskal transformer has only one input region, Input 1 (Data). You can modify the input name to reflect the name of the corresponding data in the display area. Input 1 consists of data read from a Spreadsheet, Query, or SQL Entry tool, or data copied directly into the transformer. Input 1 is treated as a series of columns containing statistical data (Data Column) or category information (Columns for Category). If extra columns are present, they are ignored. The input data can have any number of header rows. Column titles are read from the specified row. If titles are not found, default titles are provided.

Output Region Names

Kruskal generates four output regions. Each output region has no size limit.

Output 1 (Summary), the group summary output, consists of the following information:

- Sample names
- Number of observations in each sample
- Average rank of each sample
- Actual sum of ranks for each sample
- Expected sum of ranks for each sample
- Difference between the actual and expected rank sums for each sample
- H-statistic for all samples
- Corrected (for ties) H-statistic for all samples
- Significance associated with the H-values

Output 2 (Results), includes the following information, if requested:

- Normal ANOVA table
- Number of observations in each sample
- Average value for each sample
- Variance for each sample
- Standard deviation for each sample

Output 3 (Analysis), the multiple comparisons and contrasts output region, includes the following information, if requested:

- Names of the samples (groups) in the pair (for example, Group1 vs. Group2)
- Average rank of each sample
- Difference of rank sums between samples (the Observed Value)
- Critical rank sum difference value at the specified significance level
- Significance decision (for example, significant vs. not significant)

Output 4 (Messages) contains:

- Transformer run time messages
- Warnings
- Error messages
- Timestamp for documentation purposes

Examples

This example shows an application of the Kruskal-Wallis test in an experimental study. A university professor conducts an experiment to compare the effectiveness of three methods of instruction. The first method called Large Lecture consists of lectures to large groups. The second, called Small Lecture, consists of lectures to small groups. The third, called Seminar consists of seminars or discussion sessions with small groups. Out of a population of college seniors, 10 are randomly assigned to the Small Lecture sample, 10 are randomly assigned to the Seminar sample, and 30 are selected for a Large Lecture sample. The null hypothesis is that the three methods of instructions are equally effective. The effectiveness of the training is measured by achievement on a test.

The following data is presented to Input 1:

Group	Test Score
Large Lecture	60
Large Lecture	55
Large Lecture	60
Large Lecture	84
Large Lecture	91
Large Lecture	46
Large Lecture	49
Large Lecture	63
Large Lecture	69
Large Lecture	71
Large Lecture	73
Large Lecture	79
Large Lecture	89
Large Lecture	98
Large Lecture	92
Large Lecture	51
Large Lecture	54
Large Lecture	65
Large Lecture	86
Large Lecture	82
Large Lecture	58
Large Lecture	38

Large Lecture	63
Large Lecture	72

Large Lecture	78
Large Lecture	83
Large Lecture	86
Large Lecture	92
Large Lecture	98
Large Lecture	60
Small Lecture	75
Small Lecture	78
Small Lecture	91
Small Lecture	86
Small Lecture	84
Small Lecture	65
Small Lecture	73
Small Lecture	89
Small Lecture	98
Small Lecture	81
Seminar	86
Seminar	98
Seminar	89
Seminar	85
Seminar	81
Seminar	77
Seminar	72
Seminar	70
Seminar	88
Seminar	83

The Kruskal-Wallis test example parameters are set as follows:

Number of header rows
1, 1

Report name
Kruskal-Wallis Example

Column containing category information
a

Data column
b

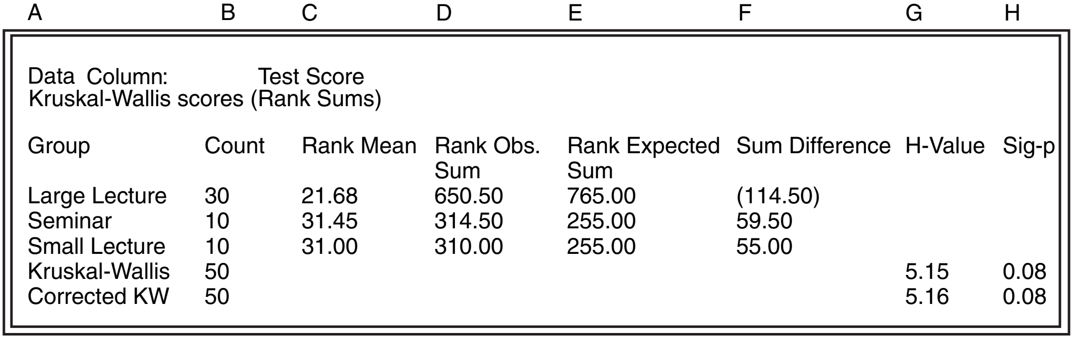
Compute multiple comparisons analysis
y

Contrast Analysis
y, "1,2+3"

Significance value to use in computations
0.95

Compute analysis of variance
y

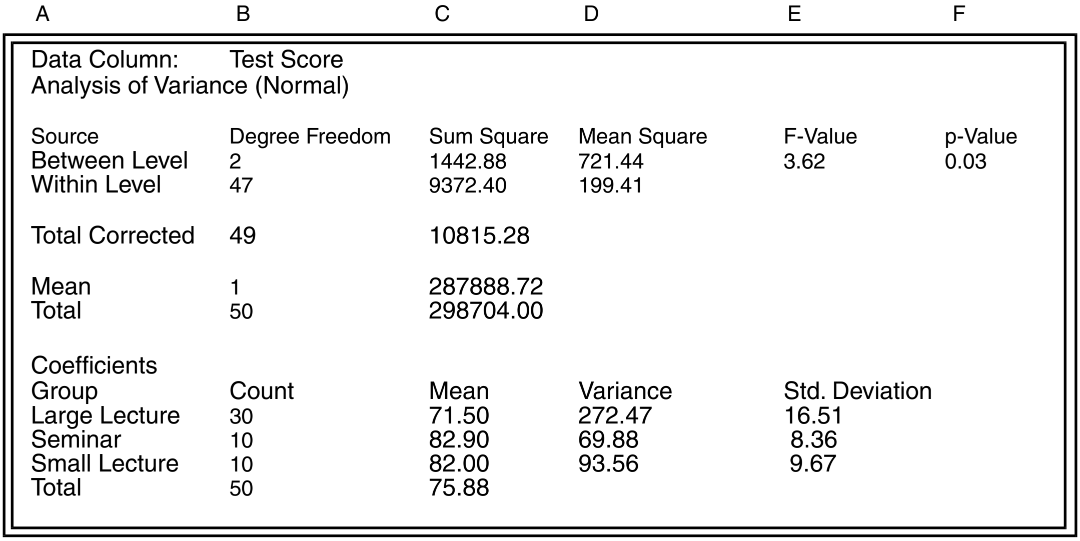
After the transformer runs, the following information is displayed in Output 1:



A	B	C	D	E	F	G	H
Data Column:	Test Score						
Kruskal-Wallis scores (Rank Sums)							
Group	Count	Rank Mean	Rank Obs. Sum	Rank Expected Sum	Sum Difference	H-Value	Sig-p
Large Lecture	30	21.68	650.50	765.00	(114.50)		
Seminar	10	31.45	314.50	255.00	59.50		
Small Lecture	10	31.00	310.00	255.00	55.00		
Kruskal-Wallis	50					5.15	0.08
Corrected KW	50					5.16	0.08

In Output 1, it is apparent that the Large Lecture sample has a very different distribution from the other two samples. It has the smallest rank mean and the largest difference between observed and expected rank sums. The sample also makes it apparent that the other two groups are quite similar; rank means of the Small Lecture and Seminar samples are almost equal. In addition, the difference between the observed and expected mean sums for each sample is similar. The probability of 0.08 indicates that there is only an 8% likelihood that the differences detected by the Kruskal transformer occurred by chance. Thus, there is a 92% chance that the transformer discovered a stable set of differences.

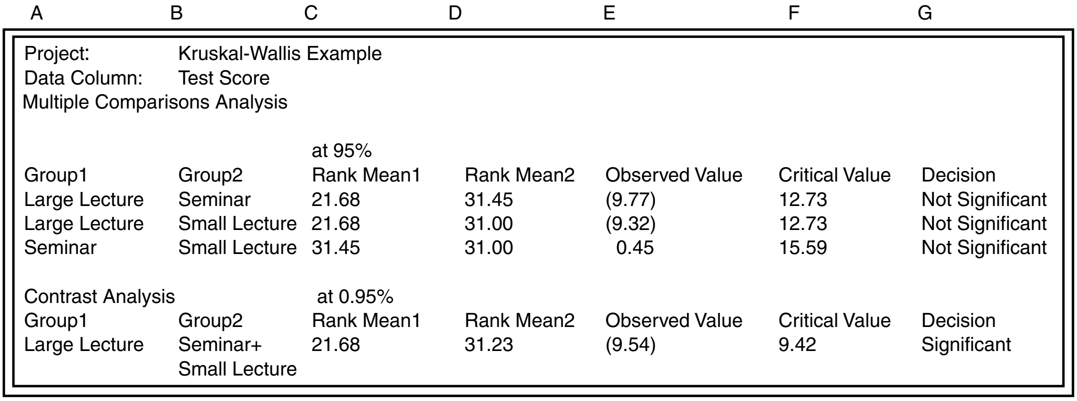
Because the data being analyzed is interval level, it is appropriate to apply the ANOVA method. That information is sent to Output 2:



```
A                 B                  C              D                  E               F

Data Column:      Test Score
Analysis of Variance (Normal)

Source            Degree Freedom     Sum Square     Mean Square        F-Value         p-Value
Between Level     2                  1442.88        721.44             3.62            0.03
Within Level      47                 9372.40        199.41

Total Corrected   49                 10815.28

Mean              1                  287888.72
Total             50                 298704.00

Coefficients
Group             Count              Mean           Variance           Std. Deviation
Large Lecture     30                 71.50          272.47             16.51
Seminar           10                 82.90          69.88               8.36
Small Lecture     10                 82.00          93.56               9.67
Total             50                 75.88
```

The one-way analysis of variance technique assumes that the samples are independent and have a normal distribution. If it seems that the data fits those assumptions, you can apply the ANOVA results within the Kruskal-Wallis test. The F-statistic in the Output 2 table is quite large (3.62); consequently, the probability associated with the F is 0.03. Because there is only a 3% chance that the null hypothesis (there is no difference among the sample distributions) is true, the ANOVA table supports the decision to accept the alternate hypothesis that the samples are different. The ANOVA output supports the Kruskal-Wallis output indicating that there is a real difference in the way the three groups scored on the tests.

Although the previous information proves that there are differences among the samples, it is still not certain which samples differ from others. The Multiple Comparisons Analysis and Contrast Analysis tables found in Output 3 are shown in Figure 44. These techniques should demonstrate which samples differ from others.



```
A               B               C               D               E                   F               G
Project:        Kruskal-Wallis Example
Data Column:    Test Score
Multiple Comparisons Analysis

                                at 95%
Group1          Group2          Rank Mean1      Rank Mean2      Observed Value      Critical Value  Decision
Large Lecture   Seminar         21.68           31.45           (9.77)              12.73           Not Significant
Large Lecture   Small Lecture   21.68           31.00           (9.32)              12.73           Not Significant
Seminar         Small Lecture   31.45           31.00             0.45              15.59           Not Significant

Contrast Analysis                at 0.95%
Group1          Group2          Rank Mean1      Rank Mean2      Observed Value      Critical Value  Decision
Large Lecture   Seminar+        21.68           31.23           (9.54)              9.42            Significant
                Small Lecture
```

Figure 44. Multiple Comparisons Analysis and Contrast Analysis tables

The Multiple Comparisons Analysis table contains results of all possible comparisons between sample pairs. At the 95% confidence level, none of the pairs of samples have different distributions. The Contrast Analysis table shows the result of the contrast. When the distribution of the Large Lecture sample is compared with the combined distribution of the Small Lecture and Seminar samples, there is a significant difference.

The information supplied by the Kruskal transformer indicates that there is a real difference between the Large Lecture and the combined Small Lecture and Seminar distributions. Because the rank mean for the combined Small Lecture and Seminar group is larger than the Large Lecture rank mean, it seems that the small group teaching settings are more effective than the large group instruction setting.

Using Kruskal-Wallis Statistics

The Kruskal-Wallis test analyzes ranks of ordinal-level data in a framework similar to the analysis of variance (ANOVA) test used with interval- or ratio-level data. Unlike the ANOVA technique, the Kruskal-Wallis test does not assume that the data is normally distributed. As a result, this test is also useful in analyzing interval- or ratio-level data that does not have a normal distribution.

Ordinal data contains values that can be arranged from lowest to highest. In addition, distances among ordinal values cannot be measured. Interval data has most of the characteristics of ordinal data. However, the distance between values of interval data can be measured. Ratio data has the characteristics of interval data, but it also has a well-defined zero point.

In the Kruskal-Wallis test, the values in all groups are combined and ranked sequentially. The ranks in each group are then summed, and differences in the rank sums are analyzed to determine whether they are due to chance or to actual differences in the sample distributions. If the samples have different distributions, they were probably drawn from different populations.

Conversely, the null hypothesis is that the samples have the same distribution. If the sample medians and distributions are the same, the samples were either drawn from one population or from different populations that have the same distribution. One way to state the null hypothesis is:

$$H_0 \colon \overline{M}_1 = \overline{M}_2 = \overline{M}_3 = \ldots = \overline{M}_k$$

where M_1 is the median for the first sample, M_2 is the median for the second sample, and so on, and k is the number of independent samples.

An independent sample is a group of observations that are not related to another group of observations in the analysis. For example, if test subjects are randomly assigned to one of two groups based on the flip of a coin, the samples are independent. However, consider matched two-test subjects. If one subject is assigned to one group and the other subject to another group, the samples are not independent.

Comparison and Contrast Analyses

Although the Kruskal-Wallis test tells us whether there is a significant difference between one sample and any other, it does not tell which samples differ from others or how many samples are different. Extensions to the Kruskal-Wallis test provide this information. The extensions are multiple comparison analysis and contrast analysis.

Multiple comparison analysis compares each sample to every other sample. In other words, it checks for significant difference between every pair of samples. This process is helpful if there is a significant difference among samples, but you do not know which samples are different.

If you have a good idea of which sample pairs are different, there is an alternate way to verify differences. Using the contrast analysis extension of the Kruskal-Wallis test, you can specify the pairs of samples to be checked for differences. You can also combine samples before doing comparisons. For example, three groups, A, B, and C, are used to test the effectiveness of a new treatment for a disease. Sample A is the control group and receives no therapy; sample B receives drug therapy; sample C receives physical therapy. Although it is useful to know whether there are differences among the samples, it is more useful to know whether any kind of therapy provides a different outcome than no therapy. Contrast analysis allows comparison of the outcome for sample A with the outcome of the combined B and C samples.

Because contrast analysis performs fewer comparisons than the multiple comparison technique, it requires less time and resource to compute the results. It also produces more concise output than the alternate technique. The disadvantage of contrast analysis is that you must know a great deal about the data to specify which sample pairs to compare or which samples to combine. In general, if you have no idea which samples might differ from others, have no reason to combine groups, and are not concerned with the amount of time it takes to calculate the necessary measures, multiple comparison is a useful technique. If you want to look for differences among specific pairs of samples, or would like to combine groups before doing comparisons, the contrast analysis technique is more useful.

Making Decisions

For multiple comparisons and contrasts, the Kruskal transformer automatically determines whether differences between samples are significant. It does this by first calculating the difference between the rank mean of one sample and the rank mean of the second sample. It then compares the absolute value of that difference with a critical value. If the difference meets or exceeds the critical value, there is a significant difference between the two samples, and the transformer provides that information in Output 3.

The critical value is the smallest difference that is significant based on the confidence level specified by the user. The confidence level can be thought of as the likelihood that the null hypothesis is not true. For example, a confidence level of 0.99 indicates that there is a 99% chance that the null hypothesis is false.

Defining Formulas

The definitions in Table 45 apply to the equations in this section.

Table 45. Kruskal transformer symbol definitions

Symbol	Definition
D_j	Difference between the observed and expected rank sums for any given sample
E_j	Expected rank sums for any given sample
g	Number of groups of tied values
H	Kruskal-Wallis test statistic
H_C	Value of H corrected for ties
k	Number of samples
L	Correction factor for H when there are ties
n_j	Number of observations in any given sample
N	Total number of observations
R_i	Rank value of any given observation in any given sample with respect to the combined samples
$\bar{R}_j$	Observed mean rank for any given sample

Table 45. Kruskal transformer symbol definitions

Symbol	Definition
S_j	Observed rank sums for any given sample
U_i	Number of observations in all samples that tied on a given value

The calculation of the observed sum of ranks for any given sample is as follows:

$$S_j = \sum_{i=1}^{n_j} R_i$$

The calculation of the mean observed rank for any given sample is as follows:

$$\bar{R}_j = \frac{S_j}{n_j}$$

The calculation of the expected sum of ranks for any given sample is as follows:

$$E_j = n_j \quad \frac{(N+1)}{2}$$

The calculation of the difference between the observed and expected rank sums for any given sample is as follows:

$$D_j = S_j - E_j$$

H is the main Kruskal-Wallis test statistic; it measures the difference among sample rank sum differences. If there are at least five observations in each sample, the distribution of H is very close to the chi-square distribution. Consequently, the transformer uses the chi-square distribution when it automatically converts H into a probability. Using that probability, you can decide whether to accept or reject the null hypothesis. The transformer also uses H and the chi-square distribution to calculate critical values for use in multiple comparisons analysis and contrast analysis.

The calculation of H for the Kruskal-Wallis test statistic is as follows:

$$H = \left[\frac{12}{N(N+1)} \times \sum_{j=1}^{k} n_j \bar{R}_j^2 \right] - 3(N+1)$$

The Kruskal-Wallis test assumes that none of the values will be tied. When ties occur, there is a correction for the calculation of H that maintains the integrity of the statistic. First, equal values are assigned the mean of the ranks they would have received. For example, if two observations share the lowest value in a data set and should occupy ranks 1 and 2, both are assigned an average rank of 1.5. These averaged ranks are included in the calculation of rank sums. Then the value of H is corrected with the following calculations:

$$L = 1 \quad \frac{\sum_{i=1}^{g} [U_i^3 - U_i]}{N^3 - N}$$

The corrected value of H is calculated as:

$$H_C = \frac{H}{L}$$

Contrast Analysis

Each contrast consists of integers separated by commas or plus signs. A contrast specification must be enclosed in double quotation marks. The integers represent samples: the first sample value is identified with a 1, the second sample is identified with a 2, and so on. Commas separate samples that should be treated individually, and plus signs indicate that groups should be combined before completing the analysis. For example, if there are three groups and you want to know if group 1 differs from the combined distribution of group 2 and group 3, specify `"1, 2+3"`. If you specify more samples than exist in the data set, the Kruskal transformer issues an error message.

KSTest

The KSTest (Kolmogorov-Smirnov Test) transformer analyzes sample distributions. The technique used is called the Kolmogorov-Smirnov test, K-S test, or goodness-of-fit test. You can use it to answer any of the following questions:

- Is the observed distribution different from a theoretical or estimated distribution?
- Are two distributions from the same population?
- Are the two samples drawn from different populations?

Although you can use any of several other statistical methods to answer the same questions answered by the K-S test, this technique offers the following advantages over other techniques:

- The K-S test is more powerful, or more likely to find real differences between two distributions than chi-square tests.

- With small samples, the K-S test is more powerful than the t-test technique.
- The K-S test can deal appropriately with categories containing few observations. When each category contains few observations, other statistics require merging of categories, which causes the loss of valuable information.
- Like other nonparametric tests, the Kolmogorov-Smirnov test does not make assumptions about the distribution underlying the data.
- The test accommodates many types of data, including ordinal-, interval-, and ratio-levels of information.
- The KSTest transformer can conduct two distinct, but related, types of analysis:
  - The first type of analysis compares an observed distribution with a known theoretical distribution, which could be derived from some other statistical technique or from historical information.
  - The second type of analysis compares two observed samples to determine if they have different distributions.

In both types of analysis, a distribution is simply a number of cases or rows that have given values. If the distributions of the two samples are different, you can infer that the samples are drawn from different populations. If the distributions are the same, the inference is that they come from the same population or from two populations with identical distributions.

Parameters

All of the parameters described in this section are displayed in the Transformer Controls window. Most of these parameters are also displayed in the Data Entry icon contained in the transformer's demonstration capsule application. However, to simplify use of the transformer for new users, some of the more advanced parameters might be omitted from the Data Entry icon; they can be accessed only in the Transformer Controls window.

Number of header rows (; 1; 5;) row containing titles (; ,1; ,5;)

This parameter specifies the number of input rows that are skipped and not used in calculating the chi-square statistics. It also specifies the row number containing column titles that are used as labels in the output. The two values should be separated by a comma.

You can specify any number of header rows; the default value is 0. The title row number should be less than or equal to the header rows. If the number of header rows is specified, the default title row is the last row of the header rows. If no header row is specified, the default is no title row.

Report name (; KS Test Statistics;)

This parameter is a title for the output report for example, *K-S test Statistics*. If `Report Name` is blank, there is no title in Output 1.

Kolmogorov-Smirnov test method (One; Two)

This parameter specifies the sample method that the user wants computed. Valid methods are `One` (one-sample) and `Two` (two-sample). The default value is `One`.

Data columns (a; a,b; a,b,c;)

This parameter specifies the columns used to compute the Kolmogorov-Smirnov statistics. At least one data column is required, but any number can be specified. The transformer completes an analysis for each combination of data columns. For example, if a one-sample test is requested and the value of data columns is `a,b,c`, the transformer compares each column of data with the theoretical distribution. If a two-sample test is requested and the value of data columns is `a,b,c`, the transformer compares the following pairs of distributions: a with b, a with c, and b with c.

'One sample' test mean and standard deviation (; 14.0,2.0;)

This parameter specifies the mean and standard deviation values used by the one-sample method to compute the expected frequency distribution. For example, to compare the observed distribution with a theoretical normal distribution that has a mean of 15 and a standard deviation of 3.0, specify `15, 3.0`. If the parameter is blank, the transformer uses the mean and standard deviation of the observed sample to calculate the expected frequency distribution. This type of analysis is also called the Lilliefors statistic. If a two-sample test is specified, this parameter is ignored.

'Two sample' test method (; Grouped; Ungrouped)

This parameter specifies the type of data under analysis. If the data is ordinal, specify `Grouped`. If the data is interval, the value of this parameter should be set to `Ungrouped`. If this parameter is blank, the KSTest transformer assumes that the data is ordinal and a grouped analysis is performed.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- Results (Output 1)
- Summary (Output 2)
- Messages (Output 3)

When you select any one of these choices, the data associated with that choice displays in the display area of the transformer.

Input Region Names

The KSTest transformer has one input region, Input 1 (Data). You can modify the input name to reflect the name of the corresponding data in the display area. Input 1 consists of data read from a Spreadsheet, Query, or SQL Entry tool, or copied directly into the transformer. Input 1 is formatted as a series of columns that contain statistical data (Data Columns). If extra columns are present, they are ignored. The input data can have any number of header rows. Column titles are read from the specified title row; however, if titles are not found, default titles are provided.

Output Region Names

The KSTest transformer generates three output regions that have no size limits.

Output 1 (Results), which is the Kolmogorov-Smirnov test region, contains summary information about the samples and test output, including:

- Sample variable names
- Sample counts
- Sample means
- Sample standard deviations
- Largest positive difference between the two samples' cumulative distributions (D)
- Probability of D

Output 2 (Summary), which is the distribution region, consists of intermediate results of the test, including sample distribution summaries for both samples. Specifically, the region contains the following information:

- Sample distributions
- Cumulative frequency distributions
- Deviation of cumulative distributions
- Normal-Z or z-score distributions (if a one-sample test is specified)

Output 3 (Messages) contains:

- Transformer run-time messages
- A timestamp for documentation purposes
- Warnings
- Error messages

One-Sample KS Test Transformer Example

As a first step in simulating the effects of a new scheduling policy, the Metropolitan Bus Company wants to learn more about rush hour delay times on a route that follows a busy downtown street. Specifically, the company needs to know if the delays are distributed like delays on other bus routes. Over the 30-day sample period, 21 delays occur. From studies of bus routes on other streets in the downtown area, the company knows that the mean delay time of 4.0 minutes has a standard deviation of 1.5 minutes. The null hypothesis is therefore formulated as:

H_o: The delays on this bus route are normally distributed with a mean of 4.0
and standard deviation of 1.5 minutes

The following data is presented to Input 1 of the KSTest transformer:

	Time Delay
Count	(Minutes)
1	4.1
2	3.9
3	5.2
4	4.8
5	3.0
6	7.1
7	2.9
8	6.2
9	5.9
10	5.6
11	4.7
12	4.2
13	4.0
14	5.3
15	4.9
16	3.1
17	7.2
18	3.0
19	6.3
20	6.0
21	4.7

The KSTest transformer parameters are set as follows:

Number of header rows
1

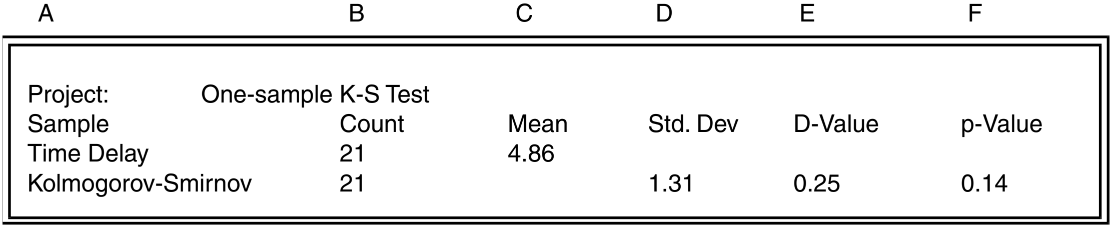
Report name
One-sample K-S test

Kolmogorov-Smirnov test method
One

Data columns
b

'One sample' test mean and standard deviation
4.0, 1.5

When the transformer run finishes, the following report is created in Output 1:



A	B	C	D	E	F
Project: One-sample K-S Test					
Sample	Count	Mean	Std. Dev	D-Value	p-Value
Time Delay	21	4.86			
Kolmogorov-Smirnov	21		1.31	0.25	0.14

The sample mean (4.8619) and the sample standard deviation (1.30747) are close to the hypothesis mean (4.0) and standard deviation (1.5). The maximum estimated D-value, which is the largest absolute difference between the observed and expected cumulative distribution functions, has a value of 0.25106. Although this value is relatively large, indicating that the actual and expected distributions fit together poorly, the p-value, which is the probability that the null hypothesis is true, has a moderately low value of 0.14163. If the company can risk a 14% chance of being wrong, it can reject the null hypothesis and say that the delays on the bus route under study do not share the same distribution as delays on other bus routes.

Each row of Output 2 displays the calculation of all the observed and expected cumulative values:

Time Delay	Obs. Cumulative	Normal-Z	Exp. Cumulative	Deviation
2.90	0.05	(0.73)	0.23	(0.18)
3.00	0.10	(0.67)	0.25	(0.16)
3.00	0.14	(0.67)	0.25	(0.11)
3.10	0.19	(0.60)	0.27	(0.08)
3.90	0.24	(0.67)	0.47	(0.24)
4.00	0.29	0.00	0.50	(0.21)

4.10	0.33	0.07	0.53	(0.19)
4.20	0.38	0.13	0.55	(0.17)
4.70	0.43	0.47	0.68	(0.25)
4.70	0.48	0.47	0.68	(0.20)
4.80	0.52	0.53	0.70	(0.18)
4.90	0.57	0.60	0.73	(0.15)
5.20	0.62	0.80	0.79	(0.17)
5.30	0.67	0.87	0.81	(0.14)
5.60	0.71	1.07	0.86	(0.14)
5.90	0.76	1.27	0.90	(0.14)
6.00	0.81	1.33	0.91	(0.10)
6.20	0.86	1.47	0.93	(0.07)
6.30	0.90	1.53	0.94	(0.03)
7.10	0.95	2.07	0.98	(0.03)
7.20	1.00	2.13	0.98	0.02)

As shown in the last column (Deviation), the expected cumulative distribution of delays is lower than the observed cumulative distribution in all but the last row. This measure indicates that most of the observed delays fell below the normal distribution of delays on other streets. Because the distribution of delays for this route can differ from the distribution of delays on other routes, the bus company will have to be careful when simulating the effects of the schedule change.

One-Sample Ungrouped Data Example

An identical aptitude test was given to all seniors in two high schools, one in a rural area, the other in an urban area. The State Education Department wants to know whether the two groups have the same distributions of scores. The null hypothesis is therefore formulated as:

H_0: The urban and rural samples have the same distribution of test scores.

To answer this question, the scores of eight students are randomly selected from each high school. The scores are displayed as an ungrouped data table and are presented to Input 1 of the KS Test transformer as follows:

Student #	Urban Group	Rural Group
1	98	81
2	84	95

3	92	74
4	40	48
5	62	74
6	74	60
7	70	62
8	79	49

The KS Test transformer parameters are set as follows:

Number of header rows
1

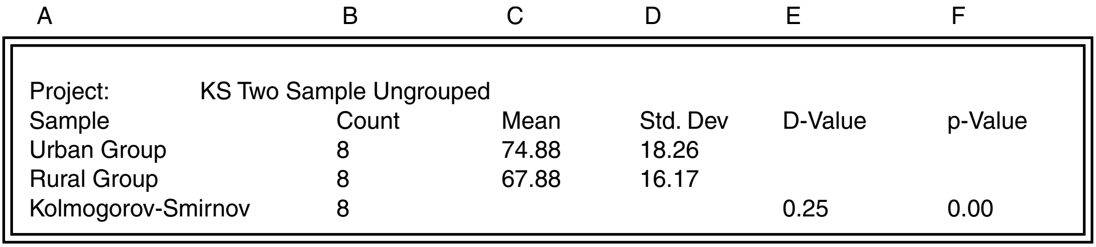
Report name
KS One-sample Ungrouped

Kolmogorov-Smirnov test method
Two

Data columns
b, c

'Two sample' test method
Ungrouped

When the transformer run finishes, the following report is created in Output 1:



A	B	C	D	E	F
Project: KS Two Sample Ungrouped					
Sample	Count	Mean	Std. Dev	D-Value	p-Value
Urban Group	8	74.88	18.26		
Rural Group	8	67.88	16.17		
Kolmogorov-Smirnov	8			0.25	0.00

In this report, the means and the standard deviations of the two samples are not very similar. The maximum difference-D has a value of 0.25000. The p-value of 0 shows that the cumulative distribution of the first sample is significantly different from the distribution of the second sample.

Output 2 shows the combined data, the observed cumulative values of the two input data, and their differences:

Urban Group:Rural Group	Input Data	Cumulative 1	Cumulative 2	Deviation
	40.00	0.13	0.00	0.13
	48.00	0.13	0.13	0.00

49.00	0.13	0.25	(0.13)
60.00	0.13	0.38	(0.25)
62.00	0.25	0.50	(0.25)
62.00	0.25	0.50	(0.25)
70.00	0.38	0.50	(0.13)
74.00	0.50	0.75	(0.25)
74.00	0.50	0.75	(0.25)
74.00	0.50	0.75	(0.25)
79.00	0.63	0.75	(0.13)
81.00	0.63	0.88	(0.25)
84.00	0.75	0.88	(0.13)
92.00	0.88	0.88	0.00
95.00	0.88	1.00	(0.13)
98.00	1.00	1.00	0.00

The information in this table supports the conclusions drawn in the first table. It indicates that in every case, the aptitude scores of the urban students were higher than the scores of the rural students. Thus, the State Board of Education should reject the null hypothesis, and conclude that the two distributions of the aptitude test scores are different.

One-Sample Grouped Data Example

The Scholastic Aptitude Test (SAT) has been designed to represent the knowledge level of high school seniors. In this example, there is a sample of SAT scores for students from two different high schools; one is in the northeastern region of the U.S., and the other is in the southeast. The U.S. Department of Education wants to know whether these two schools have the same SAT score distributions. The null hypothesis formulated to help answer this question is:

H_o: The students from the Northeast will have the same distribution of SAT scores as the students from the Southeast

Six SAT scoring levels and the number of students from each school who fall into those categories are displayed in the grouped table, which is sent to Input 1 of the KSTest transformer:

SAT Score	Southeast School	Northeast School
<= 400	46	36
401-450	42	39

451-500	33	24
501-550	25	36
551-600	21	29
> 600	33	40

The KS Test transformer parameters are set as follows:

Number of header rows
1

Report name
KS Two-sample Grouped

Kolmogorov-Smirnov test method
Two

Data columns
b, c

'Two sample' test method
Grouped

The transformer creates the following report in Output 1:

A	B	C	D	E	F
Project:	KS Two Sample Grouped				
Sample	Count	Mean	Std. Dev	D-Value	p-Value
Southeast School	6	33.33	9.56		
Northeast School	6	34.00	6.22		
Kolmogorov-Smirnov	6			0.12	0.11

This report indicates that the means and standard deviations of the two samples are somewhat alike. The maximum difference D has a value of 0.11971. The p-value of 0.11066 indicates that there is about an 11% chance that the null hypothesis is true and that the distributions are the same.

Output 2 shows the distributions of the two groups:

Southeast School	Northeast School	Cumulative 1	Cumulative 2	Deviation
46.00	36.00	0.23	0.18	0.05
88.00	75.00	0.44	0.37	0.07

121.00	99.00	0.61	0.49	0.12
146.00	135.00	0.73	0.66	0.07
167.00	164.00	0.84	0.80	0.03
200.00	204.00	1.00	1.00	0.00

Based on this table, it seems that the students in the northeastern high school are somewhat more likely to be in the lower score group than are students from the southeastern high school. Because the implications of finding such different SAT score distributions are serious, the education department wants to be certain that the difference they are finding did not happen due to chance. Because of this and the fact that there is an 11% chance that the distributions really are not different, they will accept the null hypothesis and conclude that there is no significant difference between the two distributions of SAT scores.

Using One-Sample K-S Test Statistics

The one-sample Kolmogorov-Smirnov test is useful for studying the distribution of interval or ratio values for a number of different observations. Interval-level variables have values that can logically be arranged from high to low; the distances between interval values are equal and can be precisely measured. Ratio-level variables are like interval variables, but also have an absolute zero value. For example, income is a ratio variable, whereas a score that indicates socio-economic position is an interval variable.

In the one-sample test, the known distribution is composed of values for individual observations. The theoretical distribution is assumed to be normally distributed and is thus defined by its mean and standard deviation. Using that information, the transformer estimates the expected values of each observation. It then compares the cumulative frequency of the observed distribution with that of the theoretical distribution. The cumulative frequency is a proportion; it is the product of dividing the sum of values up to the current observation by the sum of all values in the sample. The transformer then identifies the largest absolute difference, D, between the observed cumulative frequency distribution *S(a)* and the expected cumulative frequency distribution $S_e(a)$. Another way of stating this operation is:

$$D = \mathrm{Max}\left|S(a) - S_e(a)\right|$$

The null hypothesis is that there is no difference between the observed distribution and the theoretical normal distribution. Another way to state the null hypothesis is:

$$\mathrm{H}_0\colon \mathrm{S(a)} = \mathrm{S_e(a)}$$

To accept or reject the null hypothesis, it is necessary to know the probability of the *D* value occurring due to chance. The KSTest transformer automatically estimates the probability of *D*, given the size of the sample. A high probability indicates that the value of *D* could easily occur due to chance and that the distributions are probably the same. Conversely, a small probability indicates that *D* is unlikely to occur due to chance, that the observed distribution is different from the theoretical distribution, and that they are likely to come from different populations.

The one-sample K-S test can be used to determine if a model predicting demand for a product matches the actual demand distribution. The observed sample would be sales volume for the markets under study, and the expected distribution would be predicted sales volume in those same markets.

Defining One-Sample Formulas

The symbols in Table 46 are used in the equations in this section.

All symbols containing an A summarize the observed distribution, and all symbols containing an E summarize the expected distribution. Also, values of the expected sample mean and standard deviation ($\bar{E}$ and S_E, respectively) can be specified in the Transformer Controls window. If they are not specified, the transformer uses the mean and standard deviation of the observed sample ($\bar{A}$ and S_A respectively). This form of analysis is called the Lilliefors statistic.

Table 46. One-sample formula symbol definitions

Symbol	Definition
A_i	Value for any given row of the observed sample
$\bar{A}$	Mean of the observed sample
A_k	Cumulative relative frequency of the *k*th smallest data value in the observed sample.
d_k	Deviation between expected and observed cumulative relative frequency of the *k*th smallest data value
D	Maximum absolute value of d_k
$\bar{E}$	Specified mean of the expected sample
E_k	Cumulative relative frequency of the *k*th smallest data value in the expected sample
k	Row number of the *k*th smallest observed value
N	Total number of rows in the observed sample
S_A	Standard deviation of the observed sample
S_E	Specified standard deviation of the expected sample
z_k	The z-score (Normal-Z) associated with the expected row value
C^2	Chi-square value associated with D

The observed sample mean is calculated as follows:

$$\bar{A} = \frac{\sum_{i=1}^{N} A_i}{N}$$

The observed sample standard deviation is calculated as follows:

$$S_A = \sqrt{\frac{\sum_{i=1}^{N} (A_i - \bar{A})^2}{(N-1)}}$$

The expected z-score for any given row of the expected sample is calculated as follows:

$$z_k = \frac{(A_k - \bar{E})}{S_E}$$

The cumulative relative frequency for any given row of the observed sample is calculated as follows:

$$A_k = \frac{\sum_{i=1}^{k} A_i}{N}$$

The cumulative relative frequency for any given row of the expected sample is calculated as follows:

$$E_k = \text{prob}(z_k)$$

The deviation between the expected and observed cumulative frequency for any given row is calculated as follows:

$$d_k = E_k - A_k$$

The value D is the maximum absolute value of d_k:

$$D = \text{Max}|d_k|$$

The chi-square value for *D* is calculated with the following formula:

$$\chi^2 = 4 \times D^2 \times \sqrt{N}$$

Two degrees of freedom are always associated with this chi-square. To decide whether the null hypothesis should be accepted or rejected, you need to convert the chi-square to a probability using the chi-square distribution. The KSTest transformer automatically performs this conversion and returns the probability as the p-value.

Two-Sample Statistics

The two-sample test is designed to test whether the observed distributions of two independent samples are the same. This test can be used with ordinal, interval, or ratio data. Ordinal-level data contains values that can be arranged from lowest to highest, but the distances between those values cannot be measured.

The two-sample test is sensitive to differences in any type of distribution characteristic including mean, variance, and skewness. It is especially useful when there are many ties or many observations in the first sample that have matching observations in the second sample. This condition and small sample size prohibit the use of a similar nonparametric statistical technique called the Mann-Whitney test.

The two-sample test can be used with two types of distributions: ungrouped and grouped. In an ungrouped data table, each row describes a specific pair of observations. For example, one row could contain the income of an individual in the first sample and the income of an individual in the second sample. In a grouped distribution, each row contains counts of the number of individuals or items that fit into a given category. For example, a row in a grouped two-sample data set could contain the number of people in the first sample who earn more than $50,000 and the number of individuals in the second sample with the same level of earnings.

Like the one-sample test, both types of the two-sample Kolmogorov-Smirnov test are based on differences between the observed cumulative distribution functions of the two samples. The only difference between the two forms of the two-sample test lies in the method for calculating the cumulative frequency distribution. For the grouped two-sample test, the cumulative frequency distributions for both observed samples are calculated much as in the one-sample test. The ungrouped two-sample test is based on an alternate technique that ranks the values of individual observations.

In both forms of the two-sample test, the transformer calculates the test statistic after defining the cumulative frequency distributions. To compute the test statistic, the transformer finds the largest absolute difference, D, between the first sample's cumulative frequency distribution *S1(a)* and the second sample's cumulative frequency distribution *S2(a)*. Another way of describing this operation is shown in the following formula:

$$D = \mathrm{Max}\left|S(a) - S_e(a)\right|$$

The null hypothesis is that the two samples have the same distributions and have been drawn from one population or two populations with identical distributions. Another way to state the null hypothesis is:

$$H_0\colon S(a) = S_e(a)$$

To help the user decide whether the null hypothesis should be accepted or rejected, the transformer provides a probability associated with the D value. The transformer calculates the probability that a difference at least as large as D could happen by chance, given the size of the two samples. If the probability is high, D could occur due to chance, even if there is no real difference between the two distributions; the null hypothesis should be accepted. If the probability of getting D is very small, it indicates that the difference between the samples is unlikely to occur by chance. Therefore, the null hypothesis that the two distributions are the same should be rejected in favor of the alternate hypothesis that there is a real difference between the two distributions and that the two samples were drawn from different populations.

Two-Sample Ungrouped Formulas

The first step in the two-sample ungrouped test is to combine the values of both samples, sort them into ascending order, and assign them sequential rank values. Rank distributions are then constructed for each sample. If the combined rank value is based on a value from sample A, the corresponding sample A rank value is set to the combined rank value. If the combined rank value originated in sample B, the corresponding sample A rank value is set to the previous value of sample A rank value. Conversely, if the combined rank value is based on a value from sample B, the sample B rank value is set to the combined rank. Otherwise, the sample B rank value is set to the previous sample B value.

The symbols in Table 47 are used in the equations in this section.

Table 47. Two-sample ungrouped test symbol definitions

Symbol	Definition
A_i	Value of sample A for any given row of original distribution
$\bar{A}$	Mean of sample A values in original distribution
B_i	Value of sample B for any given row of original distribution
$\bar{B}$	Mean of sample B values in original distribution

Table 47. Two-sample ungrouped test symbol definitions

Symbol	Definition
d_i	Deviation of the cumulative frequencies for samples A and B
D	Maximum absolute value of d_i
F_{Ai}	Cumulative frequency for any given row in sample A
F_{Bi}	Cumulative frequency for any given row in sample B
N	Number of rows in the new distribution
N_A	Number of rows in the original sample A
N_B	Number of rows in the original sample B
R	Rank value of any row in the combined distribution
R_{Ai}	Rank value for any given row of sample A in new distribution
R_{Bi}	Rank value for any given row of sample B in new distribution
S_A	Standard deviation of sample A values in original distribution
S_B	Standard deviation of sample B values in original distribution
C^2	Chi-square value associated with D

All symbols containing an A summarize the first distribution, and symbols containing a B summarize the second distribution.

The observed sample means for the original distribution are calculated as follows:

$$\bar{A} = \frac{\sum_{i=1}^{N_A} A_i}{N_A}$$

$$\bar{B} = \frac{\sum_{i=1}^{N_B} B_i}{N_B}$$

The observed sample standard deviations for the original distribution are calculated as follows:

$$S_A = \sqrt{\frac{\sum_{i=1}^{N_A} (A_i - \bar{A})^2}{(N_A - 1)}}$$

$$S_B = \sqrt{\frac{\sum_{i=1}^{N_B} (B_i - \bar{B})^2}{(N_B - 1)}}$$

The cumulative frequency for any given row of the samples is calculated as follows:

$$F_{Ak} = \frac{R_{Ai}}{N_A}$$

$$F_{Bk} = \frac{R_{Bi}}{N_B}$$

The deviation between the expected and observed cumulative frequency for any given row is calculated as follows:

$$d_k = F_{Ak} - F_{Bk}$$

The value D is simply the maximum absolute value of d_i:

$$D = \mathrm{Max}|d_i|$$

The chi-square value for D is calculated with the following formula:

$$\chi^2 = 4 \times D^2 \times \sqrt{\frac{(N_A \times N_B)}{(N_A + N_B)}}$$

Two degrees of freedom are always associated with this chi-square. To decide whether the null hypothesis should be accepted or rejected, you need to convert the chi-square to a probability using the chi-square distribution. The KSTest transformer automatically does this conversion and returns the probability.

Defining Two-Sample Grouped Formulas

The symbols in the Table 48 are used in the formulas for the two-sample grouped test.

Table 48. Two-sample grouped test symbol definitions

Symbol	Definition
A_i	Value for any given row of sample A
$\bar{A}$	Mean of the values in sample A
B_i	Value for any given row of sample B
$\bar{B}$	Mean of the values in sampleB
D	Maximum absolute value of *dk*
d_k	Deviation of the cumulative frequencies for samples A and B
F_{Ak}	Cumulative frequency for any given row in sample A
F_{Bk}	Cumulative frequency for any given row in sample B
k	Any given row number
NA	Number of rows in the sample A
NB	Number of rows in the sample B
O_A	Sum of the observations in sample A
O_B	Sum of the observations in sample B
S_A	Standard deviation of values in sample A
S_B	Standard deviation of the values in sample B
C^2	Chi-square value associated with D

All symbols containing an A summarize the first distribution, and symbols containing a B summarize the second distribution.

The observed sample means are calculated as follows:

$$\bar{A} = \frac{\sum_{i=1}^{N_A} A_i}{N_A}$$

$$\bar{B} = \frac{\sum_{i=1}^{N_B} B_i}{N_B}$$

The observed sample standard deviations are calculated as follows:

$$S_A = \sqrt{\frac{\sum_{i=1}^{N_A} (A_i - \bar{A})^2}{(N_A - 1)}}$$

$$S_B = \sqrt{\frac{\sum_{i=1}^{N_B} (B_i - \bar{B})^2}{(N_B - 1)}}$$

The observed sample sums are calculated as follows:

$$O_A = \sum_{i=1}^{N_A} A_i$$

$$O_B = \sum_{i=1}^{N_B} B_i$$

The cumulative frequency for any given row of the samples is calculated as follows:

$$F_{Ak} = \frac{\sum_{i=1}^{k} A_i}{O_A}$$

$$F_{Bk} = \frac{\sum_{i=1}^{k} B_i}{O_B}$$

The deviation between the expected and observed cumulative frequency for any given row is calculated as follows:

$$d_k = F_{Ak} - F_{Bk}$$

The value D is simply the maximum absolute value of d_k :

$$D = \mathrm{Max}|d_k|$$

The chi-square value for D is calculated with the following formula:

$$\chi^2 = 4 * D^2 * \sqrt{\frac{(N_A * N_B)}{(N_A + N_B)}}$$

Two degrees of freedom are always associated with this chi-square. To decide whether the null hypothesis should be accepted or rejected, you need to convert the chi-square to a probability using the chi-square distribution. The KSTest transformer automatically does this conversion and returns the probability.

NPCorrelation

The NPCorrelation (nonparametric correlation) transformer provides measures of correlation between two variables that can be interpreted like conventional correlation coefficients. The NPCorrelation transformer is designed to calculate the following statistics:

- Spearman rank correlation coefficients
- Significance levels of the Spearman correlation coefficients

- Kendall's tau correlation coefficients
- Significance levels of the Kendall's tau coefficients

Unlike the standard correlation coefficient, which summarizes actual data values, nonparametric correlation statistics summarize ranks of the actual data values. Because these statistics use ranks, they are ideal for summarizing relationships between variables that are not normally distributed.

Parameters

All of the parameters described in this section are displayed in the Transformer Controls window. Most of these parameters are also displayed in the Data Entry icon contained in the transformer's demonstration capsule application. However, to simplify use of the transformer for new users, some of the more advanced parameters might be omitted from the Data Entry icon; they can be accessed only in the Transformer Controls window.

Number of header rows (; 1; 5;) row containing titles (; ,1; ,5;)

This parameter specifies the number of rows in the input data that are skipped and not used in calculating statistics. This parameter also specifies the row number containing column titles that are used to label the output. The two values should be separated by a comma.

You can specify any number of header rows; the default value is 0. The title row number should be less than or equal to the number of header rows. If the header rows value is specified, the default title row is the last row of the header rows. If no header is specified, the default is no title row.

Report name (; NPCorrelation Test;)

This parameter is an output report title, for example, *NPCorrelation Example*. If `Report Name` is blank, there is no title in Output 2.

Data columns (a,b; a,b,c;)

This parameter specifies the columns used to compute the correlation statistics. At least two data columns are required, but more can be specified. For example, `a,b,c` computes correlation statistics for the A vs. B, A vs. C, and B vs. C pairs of columns.

Nonparametric correlation method (All; Spearman; Kendall)

This parameter specifies whether Spearman, Kendall, or both sets of statistics are calculated. Allowable values are `All`, `Spearman`, and `Kendall`, each of which can be abbreviated to its first letter. For example, if `All` is specified, the transformer calculates Kendall and Spearman statistics. The default value is `All`.

Correlation statistics (All; Correlation; Significance)

This parameter specifies whether the transformer provides correlation coefficients, significance levels, or both. Allowable values are `All`, `Cor`, and `Sig`, each of which can be abbreviated to its first letter. For example, if the significance levels are unnecessary, specify `Cor`. The default value is `All`.

Group column as (; a; b,male,female;)

This parameter segregates the input data into a set of user-specified groups or ranges that are treated as separate sets. If `Group Column As` is blank, one group is created containing all of the input data.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- Spearman (Output 1)
- Kendall (Output 2)
- Messages (Output 3)

When you select any one of these choices, the data associated with that choice displays in the display area of the transformer.

Input Region Names

The NPCorrelation transformer has only one input region, Input 1 (Data). You can modify the input name to reflect the name of the corresponding data in the display area. Input 1 consists of data read from a tool, such as a Spreadsheet, Query, or SQL Entry tool, or data copied directly into the transformer. Input 1 contains columns of statistical data (Data Columns) or grouping information (Group Column As). If extra columns are present, they are ignored. The input data can have any number of header rows. Column titles are read from the specified row. If titles are not found, default titles are provided.

Output Region Names

The NPCorrelation transformer generates three output regions that are not limited in size.

If requested, Output 1 (Spearman) contains tables of the Spearman correlation statistics, including:

- Spearman's rank correlation coefficients
- Significance levels associated with those coefficients

If requested, Output 2 (Kendall) contains tables of the Kendall's tau correlation statistics such as:

- Kendall's tau correlation coefficients
- Significance levels associated with those coefficients

Output 3 (Messages) contains:

- Transformer run-time messages
- Warnings
- Errors messages

Example

A manufacturer and marketer of dog food is developing a new dog food product. The company plans to hire a television star to endorse the product during its introduction. Before finalizing the product introduction campaign, the company's marketing staff wants to be certain that they have picked the right person to endorse their dog food. To help answer this question, the company's market researchers gather a group of 25 dog owners, tell them about the new product, and show them a preliminary television commercial in which the star promotes the dog food. Each of the 25 are then interviewed; three of the questions and the possible responses are shown here.

```
How much do you like or dislike the television star?

1. Dislike a great deal.
2. Dislike a little.
3. Don't like or dislike.
4. Like a little.
5. Like a great deal.

How much does the star's endorsement of the product help you decide
whether to try the dog food?

1. It doesn't help at all.
2. It helps a little.
3. It helped a lot.
4. It made me decide to try the product.

Do you think you'll try the product?

1. I definitely won't.
2. I probably won't.
3. I may or may not.
4. I probably will.
5. I definitely will.
```

The dog food company is hoping that there will be a positive association between liking the star, believing her, and intending to buy the product. The null hypothesis is that there will be no association among those variables.

The following data is placed in the Input 1 region:

ID Number	Like the Star?	Endorse Help?	Try the product?
1	4	3	3

2	1	2	1
3	3	3	3
4	5	3	4
5	3	4	3
6	1	1	2
7	4	4	2
8	4	4	4
9	5	1	1
10	4	2	1
11	3	4	3
12	3	3	4
13	4	4	4
14	4	4	5
15	3	3	4
16	4	3	3
17	3	4	5
18	5	4	4
19	4	4	3
20	4	4	5
21	3	2	2
22	2	1	2
23	3	2	1
24	4	4	3
25	3	4	3

The NPCorrelation transformer parameters are then set as follows:

Number of header rows
1, 1

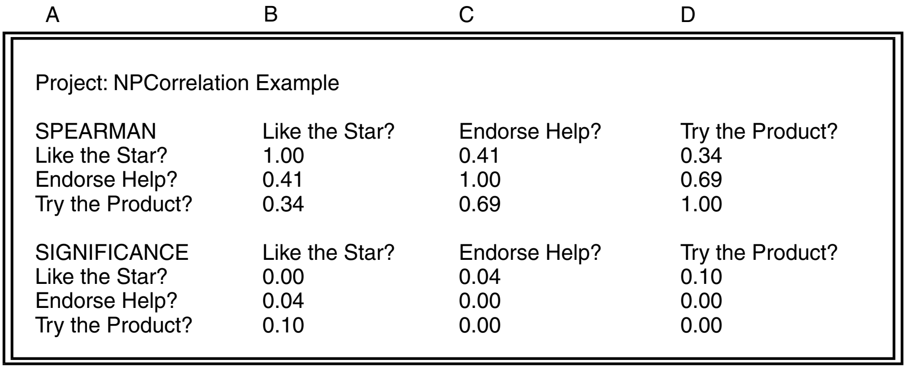
Report name
NPCorrelation Example

Data columns
g:d

Nonparametric correlation method
All

Correlation statistics
All

When the transformer has finished, the following report is displayed in Output 1:

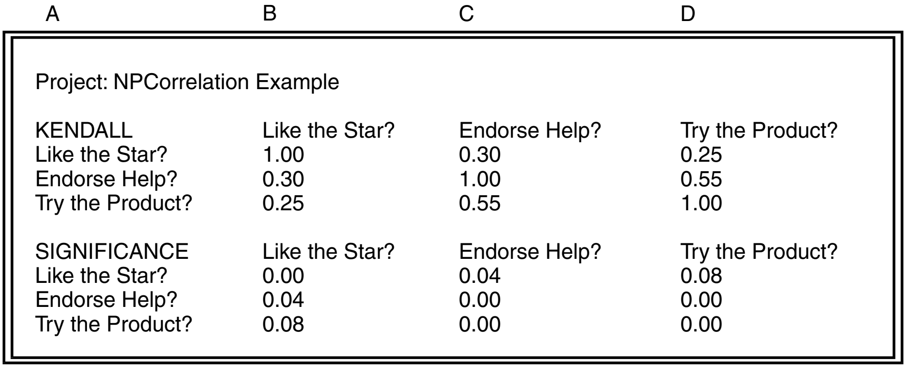


```
A                   B                 C                   D

Project: NPCorrelation Example

SPEARMAN            Like the Star?    Endorse Help?       Try the Product?
Like the Star?      1.00              0.41                0.34
Endorse Help?       0.41              1.00                0.69
Try the Product?    0.34              0.69                1.00

SIGNIFICANCE        Like the Star?    Endorse Help?       Try the Product?
Like the Star?      0.00              0.04                0.10
Endorse Help?       0.04              0.00                0.00
Try the Product?    0.10              0.00                0.00
```

In the Output 1 table, it is apparent that, in general, the three variables are positively associated. However, the level of association varies quite a bit. The smallest association is a 0.34 association between how much the pet owners like the star and whether the pet owners plan to purchase the dog food. The strongest association is between whether the pet owners felt the star's endorsement helps them decide whether to buy the dog food and how likely they are to buy the product. The coefficient for this association indicates that the ranks of the Endorse Help variable have a 0.69 correlation with the ranks of the Try the Product variable.

All of these associations are quite significant. The significance level of 0.095 for the Like the Star, Try the Product coefficient has the highest level and is thus least significant. However, that significance level indicates that there is less than a 10% chance that the observed coefficient happened due to chance. Alternatively, the Endorse Help, Try the Product association is the most significant. With a significance of less than 0.01, this association indicates that there is a better than 1% chance that the observed Spearman coefficient happened by chance. The fact that the significance levels are so low indicates that the relationships summarized by the Spearman coefficients are quite stable and would be likely to reoccur if the survey was taken again.

Because both types of correlation statistics are selected, the Kendall's tau procedure statistics shown here are sent to Output 2:



```
A                B                  C                    D

Project: NPCorrelation Example

KENDALL          Like the Star?     Endorse Help?        Try the Product?
Like the Star?   1.00               0.30                 0.25
Endorse Help?    0.30               1.00                 0.55
Try the Product? 0.25               0.55                 1.00

SIGNIFICANCE     Like the Star?     Endorse Help?        Try the Product?
Like the Star?   0.00               0.04                 0.08
Endorse Help?    0.04               0.00                 0.00
Try the Product? 0.08               0.00                 0.00
```

The Kendall's tau results reinforce the Spearman results, showing that all variables are positively associated. As expected, the tau coefficients are significantly lower than the Spearman coefficients. In spite of the lower coefficients, the same pattern of rank coefficients is apparent in the Spearman table. The largest association is between the Endorse Help and the Try the Product variables. Also, the significance levels are similar to those found in the Spearman output. In fact, the tau coefficients are slightly more significant than the Spearman coefficients.

The information in these two output regions gives the research staff enough evidence to reject the null hypothesis that there is no association among these variables. They can also accept the alternate hypothesis that these variables are positively associated. These tables also tell the company that the spokesperson is a good choice. She is well liked, as demonstrated by all the high scores in the raw data column for the Like the Star variable. More importantly, the strong positive correlation between the Endorse Help and Try the Product variable ranks shows that individuals who felt that the spokesperson's endorsement helped them decide whether to buy the dog food were likely to try it. This correlation indicates that the spokesperson's endorsement is likely to have a positive impact on the introduction of the new dog food.

Spearman Rank Correlation Statistics

The Spearman correlation statistic differs from the conventional (also called Pearson or product-moment) correlation coefficient in that the values that are summarized by the coefficient. Whereas the Pearson correlation summarizes raw values of interval- or ratio-level data, the Spearman correlation statistic summarizes the ranked values of ordinal-, interval-, or ratio-level data. Ordinal data contains values that can be arranged from lowest to highest; the distance between ordinal values cannot be measured. Interval data also contains values that can be ranked; however, the distance between interval values can be measured. Ratio data has all of the characteristics of interval data, but it also has a well-defined zero point.

Although Spearman coefficients can be interpreted like Pearson correlation coefficients, there are subtle differences between them. The Pearson correlation coefficient measures the amount of change in one variable that can be attributed to the change in another variable. In comparison, the Spearman correlation indicates whether changes in one variable correspond to changes in another variable. In other words, as one variable increases or decreases, so does the other. The Spearman correlation coefficient measures the consistency of the relationship. If an increase in one variable always results in an increase in another, the Spearman correlation is 1.0. Conversely, if an increase in one variable always results in a decrease of another variable, the Spearman correlation coefficient is -1.0. If there is no pattern between changes in the variables, the Spearman correlation is 0.

The first step in calculating a Spearman correlation coefficient is to create rank values for two variables; each rank value is based on the distribution of that variable. The observation with the lowest raw value receives a rank of 1, and the observation with the second lowest raw value receives a rank value of 2, and so on. If more than one observation has the same value (a condition also known as a tie), they are assigned the mean of the ranks that would have been assigned. For example, if three observations have the lowest values for a given variable, all three are assigned rank values of 2, because that is the mean of ranks 1, 2, and 3.

After both variables have been ranked, the Spearman correlation coefficient is calculated identically to the Pearson correlation coefficient and can be thought of as a Pearson correlation of rank values. Although the Spearman correlation coefficient provides a measure of the consistency of a relationship between two variables, some degree of consistency could occur due to chance. To determine whether the observed relationship is real and did not occur due to chance, a significance level, which measures the likelihood that the null hypothesis is true, is calculated for each correlation coefficient. The null hypothesis is that there is no real relationship between the two variables. A significance level that is near 0 indicates that the observed correlation is unlikely to occur due to chance and there is a real association between the variables.

Kendall's Tau Correlation Statistics

Kendall's tau is another measure of association between the two variables. It is interpreted like the Spearman coefficient but is calculated differently.

Like the Spearman coefficient, Kendall's tau uses ranked values of variables; it also ranges from -1.0 to +1.0. A Kendall's tau of 1.0 indicates that as the value of one variable increases, the value of the other variable always increases; that is, the two variables agree (a positive association). In contrast, a tau of -1.0 indicates that as the value of one variable increases, the value of the other variable always decreases; that is, they disagree (a negative association). A tau of 0 indicates that there is no association between two variables.

The first step in calculating tau is to sort the observations into ascending order based on the ranks of the first variable. Next, the number of concordant and discordant pairs are counted. In a concordant pair, both values (one of the first variable and one of the second) of an observation are either higher or lower than the values of a second observation. In a discordant pair, one value of the first observation is higher than a value of a second observation, whereas the other value of the first observation is lower than the value in a second observation.

For example, in the following table, the first observation's values are concordant with the values of the second observation, because the values of both variables are lower in the first observation than they are in the second. The values of the first case are also concordant with the values in the third observation, because the values in the first observation are both lower than the values in the third. Finally, the values of the second observation are discordant with the values of the third, because the value of variable A in the second case is lower while the value of variable B is higher than the corresponding values in the third observation. Thus, this table contains two concordant pairs and one discordant pair:

Observations	Variables	
	A	B
1	2	4
2	4	6
3	13	5

After the concordant and discordant pairs have been counted, the sum of the discordant pairs is subtracted from the sum of the concordant pairs. Tied values are excluded from this sum. To get the measure tau, the resulting number is divided by a measure of the maximum number of possible concordant pairs.

For sample sizes of 10 or more observations, tau is normally distributed, so the transformer uses the normal distribution to calculate a significance level. The significance level is the probability that the null hypothesis is true. Thus, if the significance level is quite small, there is a small likelihood that the coefficient occurred by chance, and the null hypothesis that there is no association between the two variables can be rejected. A small significance level also means that the alternate hypothesis, that there is an association between the two variables, can be accepted.

Comparison of Spearman Rank and Kendall's Tau Statistics

If the transformer calculates both Kendall's tau and Spearman rank correlation coefficients for the same set of data, it is apparent that correlations measured by Spearman rank coefficients have significantly higher values than correlations measured by Kendall's tau coefficients. This difference is due to the fact that the statistics are based on different scales. The Spearman rank coefficient uses the sums of squared differences as its underlying scale, whereas Kendall's tau coefficient uses the number of concordant pairs as its underlying scale. Because the two coefficients use different scales, they are numerically different and should not be used to compare two different associations. For example, it is inappropriate to say that the association between variable A and variable B is stronger than the association between variable B and variable C, because the Spearman's coefficient for A and B is higher than the Kendall's tau for variables B and C.

There is also a small difference in the way the statistics should be interpreted. As mentioned earlier, the Spearman rank coefficient measures the correlation between ranked values. In contrast, Kendall's tau measures the difference between the probability of concordant and discordant pairs of ranked values.

Tau has one advantage over the Spearman coefficient. When the sample sizes are very small and have less than 10 pairs of observations, tau is more effective in finding real associations between samples.

Although there are some differences between the correlation statistics, both are based on ranked values and in most applications, they capture the same amount of information about the association between two variables. One is just as likely to find a real association as the other. As a result, the significance level associated with a Kendall's tau coefficient for one pair of variables should be similar to the significance level associated with a Spearman's rank coefficient for the same variable pair.

Spearman Rank Correlation Formulas

The definitions in Table 49 apply to the equations in this section.

Table 49. Spearman rank correlation formula symbol definitions

Symbol	Definition
D_i	Difference between any given pair of ranks
MR	Maximum value of the sum of squared differences
n	Number of paired observations for variable A and variable B
P_i	First rank in a given rank pair
Q_i	Second rank in a given rank pair
R	Spearman rank correlation coefficient
S	Sum of the squared differences

The sample D_i and its sum of squares are calculated as follows:

$$D_i = P_i - Q_i$$

$$S = \sum_{i=1}^{n} D_i^2$$

The maximum value MR is calculated as follows:

$$MR = n \times (n^2 - 1)$$

The Spearman coefficient of rank correlation R is defined as:

$$R = 1 - \left[6 \times \frac{S}{MR}\right]$$

When a sample contains many ties, a correction is applied to the calculation of the Spearman coefficient. Because of its complexity, this correction is not explained here.

If requested, the transformer automatically converts the Spearman coefficient into a significance level. If the significance level is very small, the null hypothesis that there is no association between the two variables is rejected in favor of the alternate hypothesis that there is an association.

Kendall's Tau Correlation Formulas

The definitions in Table 50 apply to the equations in this section.

Table 50. Kendall's tau symbol definitions

Symbol	Definition
N	Total number of pairs
S	Difference between U and V
T	Ratio of the measure of degree of disagreement
U	Number of concordant pairs
V	Number of discordant pairs

The difference between the sums of concordant and discordant pairs is calculated as follows:

$$S = U - V$$

The Kendall's tau coefficient T is defined as follows:

$$T = \frac{2 * S}{N * (N - 1)}$$

If there are many tied values, a correction is applied to the calculation of the tau correlation coefficient. Because of its complexity, this correction is not explained here.

If requested, the NPCorrelation transformer automatically converts the tau coefficient into a significance level. If the significance level is small, the null hypothesis can be rejected in favor of the alternate hypothesis that the variables are associated.

Specifying Data Columns

To specify the input columns containing data, enter a list, a range of columns, or both, using either the letters or the numbers associated with the columns. For example, if the input data is in the first three columns of a Spreadsheet II, a `Data Columns` list specification would be `a,b,c` or `1,2,3`. A list of columns is simply a series of column letters or numbers separated by commas. The specified columns do not have to be contiguous. For example, if the `Data Columns` specification is `a,c`, the transformer gathers the data from the first and third columns of the input region.

The `Data Columns` parameter also accepts ranges of columns. A range of columns consists of the number or letter associated with first data column, a colon, and the letter or number associated with the last data column. For example, if the transformer should use the first five columns of data, the `Data Columns` specification would be `1:5` or `a:e`.

The `Data Columns` parameter also accepts a combination of lists and ranges. For example, if the input data occurs in the first, second, and fourth through sixth columns, the parameter specification would be `a,b,d:f` or `1,2,4:6`.

Specifying Grouping Features

A grouping expression consists of three parts: a group column, an optional list of grouping criteria, and an optional group type specifier. Each group is treated as a distinct set of data, and a set of correlation tables is generated for every requested group.

The group column is a single data column containing information that determines the group to which a particular data element belongs. For example, if the first column of the input data contains the grouping information, the entry would be `a`, referring to column A.

A list of grouping criteria can follow the column name. Grouping criteria allow you to specify exactly which groups are created. The criteria can be a list of text values, numeric values, or dates, each separated by a comma. If grouping criteria are not present, the NPCorrelation transformer creates groups for every unique value of the group column.

The group type specifier controls whether the grouping criteria are treated as members of a group or limits of a range. If the type specifier `only` is present, a group is created for each item in the grouping criteria list. Only values that exactly match a particular grouping criterion are added to its corresponding group. If the type specifier is not present, the grouping criteria are treated as the end points of a series of ranges. Any value that is greater than the first end point and less than or equal to the second end point is treated as part of that particular range.

Examples of each of the four possible variations of a grouping expression are:

<no expression>
One group is created containing all of the input data.

a A group is created for each different value in column A.

a, 10, 20
Three ranges are created: all values up to and including 10, all values above 10 but less than or equal to 20, and all values above 20.

a, 10, 20, only
Two groups are created: all values where column A is 10 and all values where column A is 20.

NPIndependent

The NPIndependent (nonparametric independent) transformer uses the Mann-Whitney test and the Wilcoxon rank-sum test to determine whether two independent samples are different. The two tests, which use slightly different computations but are functionally identical, can help answer questions such as:

- Do the two samples have significantly different distributions?
- Do the two samples come from different populations?
- Do the two samples come from the same population?

Because these statistics are nonparametric tests, they are ideal for situations when sample sizes are quite small or the variables under study can not be normally distributed. Both tests make the best use of data that cannot be precisely measured. They are also useful for comparing two samples that have a different number of observations. Finally, both tests are the most powerful of the nonparametric tests and are most likely to find real differences between two samples.

Parameters

All of the parameters described in this section are displayed in the Transformer Controls window. Most of these parameters are also displayed in the Data Entry icon contained in the transformer's demonstration capsule application. However, to simplify use of the transformer for new users, some of the more advanced parameters might be omitted from the Data Entry icon; they can be accessed only in the Transformer Controls window.

Number of header rows (; 1; 5;) row containing titles (; ,1; ,5;)

This parameter specifies the number of rows in the input data that are skipped and not used in calculating statistics. This parameter also specifies the row number containing column titles that are used as labels in the output. The two values should be separated by a comma.

You can specify any number of header rows; the default value is 0. The title row number should be less than or equal to the number of header rows. If the header rows value is specified, the default title row is the last row of the header rows. If no header is specified, the default is no title row.

Report name (; NP-Independent Test;)

This parameter is an output report title, for example, *NPIndependent Test*. If it is blank, there is no title in Output 1.

Data columns (; a,b; a,b,c;)

This parameter specifies the columns containing the observation values used to compute the paired sample statistics. Two or more data columns must be specified, each column containing observations for one sample. If more columns are specified than are needed to complete one analysis, several analyses are performed. For example, if `Data Columns` is specified as `a,b,c`, an analysis is completed for A vs. B, A vs. C, and B vs. C.

Statistical Method (All; Mann-Whitney; Wilcoxon)

This parameter specifies whether the Mann-Whitney test, Wilcoxon rank-sum test or both are computed. The valid responses are `All`, `Mann-Whitney`, and `Wilcoxon`, which can be abbreviated as `a`, `m` and `w`, respectively. For example, if you want the output for both tests, specify `a` or `All`. The default response is `All`.

Group column as (a; b,male,female,only;)

This parameter segregates the input data into a set of user-specified groups or ranges that are treated as separate data sets. If this field is blank, one group is created that contains all of the input data.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- Results (Output 1)
- Messages (Output 2)

When you select any one of these choices, the data associated with that choice displays in the display area of the transformer.

Input Region Names

NPIndependent has only one input region, Input 1 (Data). You can modify the input name to reflect the name of the corresponding data in the display area. Input 1 consists of data read from a tool, such as a Spreadsheet, Query, or SQL Entry tool, or data copied directly into the transformer. Input 1 contains columns of statistical data (Data Columns). If extra columns are present, they are ignored. The input data can have any number of header rows. Column titles are read from the specified row. If titles are not found, default titles are provided.

Output Region Names

NPIndependent generates two output regions that are not limited in size.

Output 1 (Results), the test output region, consists of the following information:

- Sample names
- Number of observations in each sample
- Mean of the raw values for each observed sample
- Standard deviation of the raw values for each observed sample

If the Mann-Whitney test is requested, the following information is included in Output 1:

- The expected or mean value of U when the samples are the same
- The expected standard deviation of U when the samples are the same
- The Mann-Whitney U-value
- The z-score associated with the U-value
- The two-tailed probability associated with the Mann-Whitney test z-score

If the Wilcoxon rank-sum test is requested, the following information is included in Output 1:

- The expected or mean value of W when the samples are the same
- The expected standard deviation of W when the samples are the same
- The sum of the ranks of the first sample, which is called the W-value

- The z-score associated with the W-value
- The two-tailed probability associated with the Wilcoxon rank-sum test z-score

Output 2 (Messages) contains:

- Transformer run-time messages
- Warnings
- Error messages
- A timestamp (for documentation purposes)

Examples

A movie production corporation wants to determine the effectiveness of its promotion of a new film. It conducts a survey that first determines whether the respondents have heard of the movie. If the respondents have heard of the movie, they are asked how likely it is that they will see the movie. If they have not heard of the movie, they are given the names of the movie's stars and then asked whether they will watch the film. The possible responses to this question range from “I will definitely see the movie,” which is coded as 5, to “I will definitely not see the movie,” which is recorded as 1.

The research or alternative hypothesis in this survey is that people who have heard of the movie from advertisements, reviews, or articles are more likely to want to see the movie than people who know almost nothing about the movie. Thus, the null hypothesis is that people who have heard about the movie are as likely to want to see it as people who know almost nothing about it.

The following data is included into Input 1 of the NPIndependent transformer:

Heard of Movie	Haven't Heard
3.00	3.00
1.00	4.00
4.00	3.00
2.00	3.00
2.00	4.00
3.00	3.00
4.00	4.00
1.00	1.00
3.00	2.00
2.00	3.00
1.00	3.00

3.00	4.00
2.00	2.00
3.00	4.00
3.00	2.00
2.00	3.00
2.00	4.00
1.00	3.00
2.00	2.00
3.00	3.00
5.00	3.00
2.00	4.00
3.00	4.00
4.00	
1.00	

The NPIndependent transformer parameters are set as follows:

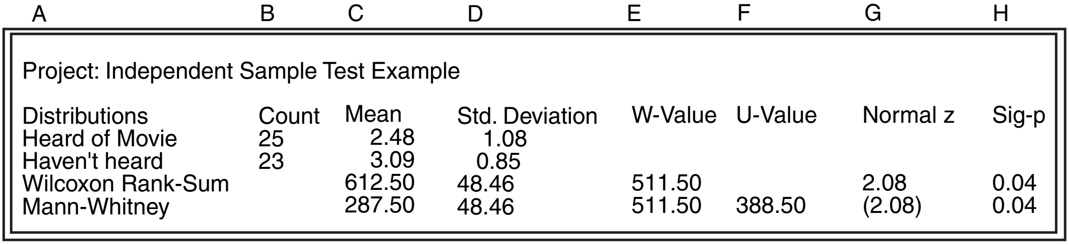
Number of header rows
1, 1

Report name
Independent Sample Test Example

Data columns
a, b

Statistical method
All

After the transformer runs, the following information is displayed in Output 1:



```
 A                     B       C          D             E          F            G           H

 Project: Independent Sample Test Example

 Distributions         Count   Mean       Std. Deviation  W-Value  U-Value      Normal z    Sig-p
 Heard of Movie        25        2.48       1.08
 Haven't heard         23        3.09      0.85
 Wilcoxon Rank-Sum             612.50     48.46           511.50                2.08        0.04
 Mann-Whitney                  287.50     48.46           511.50   388.50       (2.08)      0.04
```

As expected, several of the statistics are identical for the Mann-Whitney and the Wilcoxon rank-sum tests. The shared values include the W-value, the standard deviation of the W-values and U-values, the Z-score, and the significance levels.

The first apparent result is that the significance levels associated with the Mann-Whitney and the Wilcoxon rank-sum tests are very small. This result indicates that the null hypothesis (that there is no difference between the two samples) should be rejected in favor of the alternative hypothesis (that there is a difference between the samples). However, the research hypothesis not only stated that the distributions were different, it also stated that those who know about the movie would be more likely to want to see the movie. In other words, the research hypothesis also specified the direction of the difference.

Unfortunately for the movie company, the direction of the difference between the two observed samples does not coincide with the direction specified in the research hypothesis. The direction based on the Mann-Whitney U-value is obvious. A high U-value (relative to the mean expected U) indicates that many of the observations in the first sample (those that knew of the movie) were lower than each observation in the second sample (those that did not know of the movie). In addition, the direction of the difference is indicated by the fact that the actual W-value is lower than the mean expected W-value provided by the Wilcoxon rank-sum test. The mean expected W is the value expected if there is no difference between the sample distributions. Thus, if the observed W-value is lower, it indicates that the first sample has lower values than the second.

It seems that there is a difference in the likelihood of attending the movie between people who have heard about the movie and people who have not. Because the difference was not in the anticipated direction, the effectiveness of the movie's promotion cannot be determined. The results of this test point out the complexity of the decision-making process. It is possible that promoting the movie with advertising, media events, and press releases did raise the intended audience's interest. However, mitigating factors, such as reviews of the movie or negative word of mouth, could have undermined the effects of the studio's promotions. To determine the reason for the failure, the movie company could launch a more detailed study of the audience's movie-watching decisions.

Comparison of Mann-Whitney and Wilcoxon Rank-Sum Statistics

Data used in the independent sample tests must be measured on the ordinal, interval, or ratio level. Ordinal data has values that can be arranged from lowest to highest; the distances among ordinal values cannot be measured. Interval data is like ordinal data, but distances between interval values can be measured. Ratio data has all of the characteristics of interval data, but also has a well-defined zero point. Independent samples contain subjects that are chosen at random from different populations or from the same population; subjects in one sample are not related to subjects in another sample.

The Mann-Whitney and Wilcoxon rank-sum tests are computed in the same way with one small exception. They share the same underlying scale, result in related test statistics, and are equally likely to find real differences between samples. The scale used by both statistics is simply the rank values of the combined samples. The rank values are obtained by combining the two sample distributions, assigning a value of 1 to the observation with the lowest value, a value of 2 to the next highest value, and so on. If any values are tied, they are assigned the mean of the ranks that would have been assigned if there had been no ties. For example, if two observations share the lowest value, they are assigned a rank of 1.5, because that is the mean of the ranks 1 and 2.

After the combined ranks have been assigned, the ranks of observations in each sample are summed. If the null hypothesis that the two groups are from the same population is true, the rank sums of both groups would be very similar. If the sums of the ranks are very different, the null hypothesis is probably not true, and the groups are from different populations (the alternative hypothesis). To reject or accept these hypotheses, you must know how likely it is for the observed difference to occur due to chance. The independent sample tests differ in the calculation of the statistic used to determine the likelihood of the observed difference.

In the Wilcoxon rank-sum test, the ranks of the first group are summed. Because the sum of the ranks of the first group determines the sum of the ranks of the second group, the sum operation with the second group is not necessary. The next step is to find out the odds of obtaining the first sample's sum of ranks, given the number of observations in both samples.

The Mann-Whitney test also uses the summed ranks of the first group. However, that sum is then subtracted from a number; the remainder of that subtraction is equal to the number of observations in the first sample that are less than each observation in the second sample. If the remainder is very small, very few observations in the first sample are smaller than the observations in the second sample. If this number is very large, many of the observations in the first sample are smaller than the observations in the second sample. If either of these situations occurs, the null hypothesis that the two samples are the same should be rejected.

The test statistics created for the Wilcoxon rank-sum test and the Mann-Whitney test are converted into z-scores, which are used to estimate the two-tailed probability of obtaining the observed test scores due to chance. This probability is issued as a significance level. A two-tailed probability measures the likelihood of one observed sample not being equal to the other sample in either direction. In other words, it checks for one distribution being higher and lower than the other distribution.

If the important fact is that either one sample is higher than another or one sample is lower than the other, a one-tailed probability is appropriate. A one-tailed significance level is exactly onefold of a two-tailed significance. Thus, even though the NPIndependent transformer only provides the two-tailed significance level, it is easy to calculate the one-tailed significance level. If the significance level is very small, the observed scores are unlikely to occur due to chance, and the null hypothesis that the samples are from the same population should be rejected in favor of the alternate hypothesis that the samples are from different populations.

Although these two tests are calculated somewhat differently, the resulting significance levels are identical. As a result, they are both just as likely to find real differences between samples and can be used interchangeably. Both tests are provided in the NPIndependent transformer, because some users might be familiar with one of the tests and not the other.

Wilcoxon Rank-Sum Test Formulas

The definitions in Table 51 apply to the equations in this section.

Table 51. Wilcoxon rank-sum formula symbol definitions

Symbol	Definition
A_i	Rank value of any given observation in the first sample with respect to combined samples
N_A	Number of observations in the first sample
N_B	Number of observations in the second sample
S	Expected standard deviation of *W*
z	Normal z-score for the observed *W*
W	Sum of the ranks in the first sample
$\bar{X}$	Expected value of W assuming the samples are identically distributed

The sum of the ranks of the first sample is calculated as follows:

$$W = \sum_{i=1}^{N_A} A_i$$

The expected value of W assuming identically distributed samples is calculated as follows:

$$\bar{X} = \frac{N_A \times (N_A + N_B + 1)}{2}$$

The expected standard deviation is calculated as follows:

$$S = \sqrt{\frac{(N_A \times N_B \times (N_A + N_B + 1))}{12}}$$

The normal z-score for the observed W is calculated as follows:

$$z = \frac{(W - \bar{X})}{S}$$

The NPIndependent transformer automatically converts the z-score into a two-tailed probability or significance level. The significance level indicates the likelihood that the observed W occurred due to chance. Thus, if the significance level is very close to 0, W is unlikely to occur due to chance, and the null hypothesis that the samples are from the same populations should be rejected. Conversely, if the significance level is very close to 1, W is likely to occur due to chance, and the null hypothesis that the samples are from the same population should be accepted.

Mann-Whitney Test Formulas

The definitions in Table 52 apply to the equations in this section.

Table 52. Mann-Whitney test formula symbol definitions

Symbol	Definition
A_i	Rank value of any given observation in the first sample, with respect to the combined samples
N_A	Number of observations in the first sample
N_B	Number of observations in the second sample
S	Expected standard deviation of X
z	Normal z-score for the observed U
U	Number of observations in the first sample that are smaller than each observation in the second sample
$\bar{X}$	Expected value of U assuming the samples are identically distributed
W	Sum of the ranks in the first sample

The sum of the ranks of the first sample is calculated as follows:

$$W = \sum_{i=1}^{N_A} A_i$$

The number of observations in the first sample that are smaller than the number of observations in the second sample is calculated as follows:

$$U = N_A \text{※} N_B + \frac{N_A \text{※} (N_A + 1)}{2} - W$$

The expected value of U for identically distributed samples is calculated as follows:

$$\bar{X} = \frac{N_A \ast N_B}{2}$$

The expected standard deviation of U is calculated as follows:

$$S = \sqrt{\frac{N_A \ast N_B \ast (N_A + N_B + 1)}{12}}$$

The normal z-score for the observed W is calculated as follows:

$$Z = \frac{(U - \bar{X})}{S}$$

The NPIndependent transformer automatically converts the z-score into a two-tailed probability or significance level indicating the likelihood that the observed U-value occurred due to chance. Thus, if the significance level is very close to 0, U is unlikely to occur due to chance, and the null hypothesis that the samples are from the same populations should be rejected. Conversely, if the significance level is very close to 1, U is likely to occur due to chance, and the null hypothesis that the samples are from the same population should be accepted.

Specifying Grouping Features

A grouping expression consists of three parts: a group column, an optional list of grouping criteria, and an optional group type specifier. Each group is treated as a distinct set of data, and a set of test results is generated for every requested group.

The group column is a single data column containing information that determines the group to which a particular data element belongs. For example, if the first column of the input data contains the grouping information, the entry would be a, for column A.

A list of grouping criteria can follow the column name to specify the groups to be created. The criteria can be text values, numeric values, or dates, each separated by a comma. If grouping criteria are not present, the transformer creates groups for every unique value of the group column.

The group type specifier controls whether the grouping criteria are treated as members of a group or limits of a range. If the type specifier `only` is present, a group is created for each item in the grouping criteria list; only values that exactly match a particular grouping criterion are added to the corresponding group. If the type specifier is not present, the grouping criteria are treated as the end points of a series of ranges. Any value that is greater than the first end point and less than or equal to the second end point is treated as part of that particular range.

Examples of each of the four possible grouping expressions are:

<no expression>
One group is created containing all of the input data.

a
A group is created for each different value in column A.

a, 10, 20
Three ranges are created: all values up to and including 10, all values above 10 but less than or equal to 20, and all values above 20.

a, 10, 20, only
Two groups are created: all values where column A is 10 and all values where column A is 20.

NPPaired

The NPPaired (nonparametric paired) transformer uses the sign test and the Wilcoxon signed-rank test to compare paired samples or to compare an observed sample with a theoretical sample. The transformer helps answer questions such as:

- Are values in one sample different than values in another sample?
- Does a treatment or event affect the responses of a sample?
- Does an observed sample distribution differ from the predicted distribution?

Because these statistics are nonparametric tests, they are ideal for situations when sample sizes are small or the variables under study can not be normally distributed. These tests are also useful when the median, rather than the mean, is the best indicator of a distribution's midpoint. Finally, both tests make good use of data that cannot be precisely measured.

Parameters

All of the parameters described in this section are displayed in the Transformer Controls window. Most of these parameters are also displayed in the Data Entry icon contained in the transformer's demonstration capsule application. However, to simplify use of the transformer for new users, some of the more advanced parameters might be omitted from the Data Entry icon; they can be accessed only in the Transformer Controls window.

Number of header rows (; 1; 5;) row containing titles (; ,1; ,5;)
This parameter specifies the number of rows in the input data that are skipped and not used in calculating statistics. This parameter also specifies the row number containing column titles that are used as labels in the output. The two values should be separated by a comma.

You can specify any number of heading rows; the default value is 0. The title row number should be less than or equal to the number of heading rows. If the heading rows value is specified, the default title row is the last row of the heading rows. If no heading is specified, the default is no title row.

Report name (; NP-Paired Test;)

This parameter is a title for the output report, for example, *NPPaired Example*. If this parameter is blank, there is no title in Output 1.

Data columns (a; a,b; a,b,c;)

This parameter specifies the column or columns containing the observation values used to compute the paired sample statistics. One or more data columns must be specified; each column should contain observations for a sample. If more columns are specified than are needed to complete one analysis. several analyses are performed. For example, if `Data Columns` is specified as `a, b, c` and two-sample tests are requested, an analysis is completed for A vs. B, A vs. C, and B vs. C. If this parameter is specified as `a, b, c` and one-sample tests are requested, an analysis will be completed for A vs. M, B vs. M, and C vs. M, where M is the specified median.

Treat data as 'One' sample or 'Paired' samples (; One; Paired)

This parameter tells the transformer whether it should perform one-sample or two-sample tests. Valid responses are `One` or `Paired` (or `o` or `p`). The default is `Paired`. If a paired-sample test is requested, more than one column should be specified in the `Data Columns` parameter. If a one-sample test is requested, the expected median should be specified on the `Median value for 'One' sample test` parameter.

Statistical method (All; SignTest; Wilcoxon)

This parameter specifies whether the sign test, Wilcoxon signed-rank test, or both are computed. The valid responses are `All`, `SignTest`, and `Wilcoxon`, which can be abbreviated as `a`, `s`, and `w`, respectively. For example, if you want the output for both tests, specify `All` or `a`. The default response is `All`.

Median value for 'One sample' test (; 10.0; 21.5;)

This parameter specifies the value used as a theoretical median when a one-sample sign test or Wilcoxon signed-rank test is performed. The value for this parameter can be any real number. For example, to compare an observed sample with a theoretical distribution that has a median of 15, specify `15` in this parameter. This parameter must be specified if you request a one-sample test in the `Treat data as` parameter. This parameter is ignored if the `Treat data as` parameter specifies a two-sample test.

Group column as (a; b,male,female,only;)

This parameter segregates the input data into a set of user-specified groups or ranges treated as separate sets. If this field is blank, one group is created containing all of the input data.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- Results (Output 1)
- Messages (Output 2)

When you select any one of these choices, the data associated with that choice displays in the display area of the transformer.

Input Region Names

NPPaired has only one input region, Input 1 (Data). You can modify the input name to reflect the name of the corresponding data in the display area. Input 1 consists of data read from a Spreadsheet, Query, or SQL Entry tool, or copied directly into the transformer. Input 1 is a series of columns containing statistical data (Data Columns). If extra columns are present, they are ignored. The input data can have any number of heading rows. Column titles are read from the specified row. If titles are not found, default titles are provided.

Output Region Names

NPPaired generates two output regions; neither has a size limit.

Output 1 (Results), which is the test output region, consists of the following information:

- Sample names
- Number of observations in each sample
- Mean of the raw values for each observed sample
- Signed rank mean, if the Wilcoxon test is requested
- Mean number of plus signs, if the sign test is requested
- Standard deviation of the raw values for each observed sample
- Standard deviation of the signed ranks, if the Wilcoxon test is requested
- Standard deviation of the number of plus signs, if the sign test is requested
- The number of observed plus signs, if the sign test is requested
- Sum of the positive signed ranks, if the Wilcoxon test is requested
- Z-score associated with the observed number of pluses, if the sign test is requested
- Z-score associated with the observed positive rank sum, if the Wilcoxon test is requested
- Probability associated with the sign test Z-score, if requested
- Probability associated with the Wilcoxon signed rank Z-score, if requested

Output 2 (Messages) contains:

- Transformer run time messages
- Warnings
- Error messages
- A timestamp (for documentation purposes)

One-Sample Example

This example illustrates an application of the sign test and Wilcoxon signed-rank test with one sample. A chain of upscale women's apparel stores is considering whether to locate a new store in a regional mall in a growing suburb. One of the concerns about the location is that, historically, the suburb has attracted families with incomes lower than the apparel chain's target market. However, the developer of the mall has provided information showing that in the past two years, the value of homes being built in the suburb has increased dramatically. This information indicates that the composition of the suburb might be changing.

The fashion chain wants to verify this trend. It has obtained an estimate of income for families in the suburb. Because a small number of high incomes can easily inflate the mean, the median is the most commonly used measure of central tendency for income. One year ago, the estimated median family income in the suburbs, adjusted for inflation, was $28,500.

The chain would like to compare incomes of families who have recently moved into the suburb with the inflation-adjusted estimated median. They obtain the incomes of a random sample of families who have moved to the suburb within the past year. The working hypothesis is that the incomes of those who have recently moved into the suburb are significantly higher than the estimated median family income. Thus, the null hypothesis is that there is no difference between the estimated median family income and the incomes of families that are now moving into the suburb. The sign test and Wilcoxon signed-rank test are well suited for answering this question because they allow the comparison of observed values with a known median.

The following data is copied into Input 1 of the NPPaired transformer:

Family ID	Income (in 1000s)
1	35.80
2	30.90
3	32.50
4	27.50
5	25.90
6	28.90
7	37.60

8	56.70
9	63.00
10	40.90
11	35.40
12	29.20
13	31.20
14	51.00
15	33.50
16	32.90

The parameters for the one-sample example are set as follows:

Number of header rows
1, 1

Report name
One-sample NPPaired Example

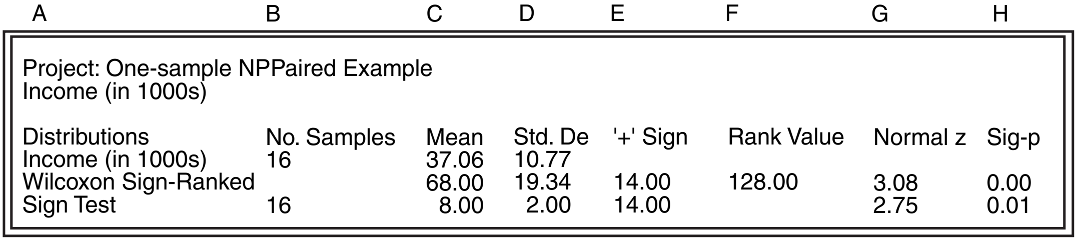
Data columns
b

Treat data as 'One' sample or 'Paired' samples
One

Statistical method
All

Median value for 'One Sample' test
28.5

After the transformer run finishes, the following information is displayed in Output 1:



```
A                    B              C        D        E          F            G            H

Project: One-sample NPPaired Example
Income (in 1000s)

Distributions        No. Samples    Mean     Std. De  '+' Sign   Rank Value   Normal z     Sig-p
Income (in 1000s)    16             37.06    10.77
Wilcoxon Sign-Ranked                68.00    19.34    14.00      128.00       3.08         0.00
Sign Test            16              8.00     2.00    14.00                   2.75         0.01
```

The information in this output region confirms the alternate hypothesis. Based on the significance levels, the conclusion is that the incomes of the observed sample differ significantly from the median family income estimated a year ago. Because there are 14 plus signs, all but two of the families in the sample had incomes above the estimated median family income of $28,500. Thus, the mall developer's claim that the suburb is becoming more upscale is supported by these tests. This result indicates that the mall might provide a suitable location for the apparel store.

Paired Sample Example

This example provides an illustration of the use of a sign test and the Wilcoxon signed-rank test with two samples. A cosmetics company is introducing a new perfume targeted to teenage girls. It wants to make certain that its promotions are effective in increasing awareness of the perfume in the target market. To gauge the effectiveness of its promotions, the company surveys a random sample of teenage girls to determine their awareness of the product before the advertising campaign starts. The same sample is surveyed again after the advertising campaign has been running for two weeks to measure their awareness of the product.

The primary question in each survey is how much the individual has heard about the new perfume (not including what she has heard in the course of the surveys). The possible responses are ordinal values ranging from 1, which indicates the person hasn't heard of the product, to 5, indicating that the person has heard a great deal about the product. The research or alternate hypothesis in this study is that the advertising campaign will increase awareness of the product in the target market. Thus, the null hypothesis is that there will be no change in product awareness.

The following information is copied into Input 1 of the NPPaired transformer:

Respondent ID	Awareness Before	Awareness After
1	3	5
2	3	4
3	4	5
4	5	5
5	3	3
6	3	2
7	2	5
8	3	3
9	2	1
10	3	4
11	2	5
12	2	4
13	5	4

14	2	5
15	5	5
16	3	5
17	1	5
18	2	3
19	3	4
20	1	4

The parameters for the paired sample are set as follows:

Number of header rows
1, 1

Report name
Paired-Sample NPPaired Example

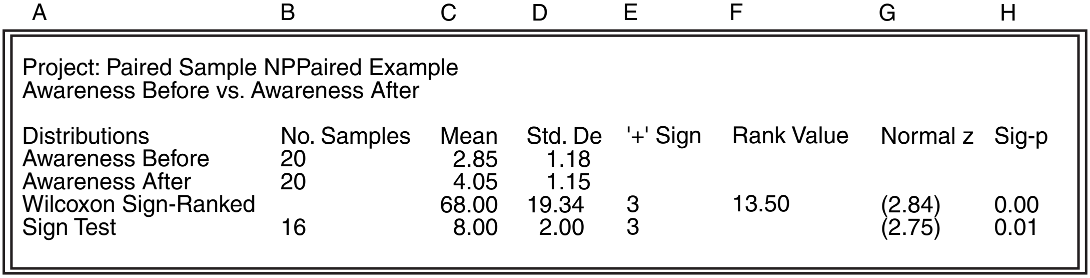
Data columns
b,c

Treat data as 'One' sample or 'Paired' samples
Paired

Statistical method
All

Median value for 'One Sample' test
10.0

After the transformer run finishes, the following information is found in Output 1:



```
A                    B             C        D        E          F             G            H

Project: Paired Sample NPPaired Example
Awareness Before vs. Awareness After

Distributions        No. Samples   Mean     Std. De  '+' Sign   Rank Value    Normal z     Sig-p
Awareness Before     20             2.85     1.18
Awareness After      20             4.05     1.15
Wilcoxon Sign-Ranked               68.00    19.34    3          13.50         (2.84)       0.00
Sign Test            16             8.00     2.00    3                        (2.75)       0.01
```

This table shows that product awareness before and after the advertising campaign is significantly different. According to the sign test, only three of the 16 respondents (when ties are excluded) were less aware of the product after the campaign. Conversely, 13 must have been more aware of the product after the promotion. The reported probabilities prove that the difference is significant and unlikely to be due to chance. In fact, the increase in awareness is even more effective than indicated by these probabilities. Because the concern is only with finding a significant increase, the actual significance level is one-half of the reported significance. Thus, the significance associated with the Wilcoxon signed-rank test is actually 0.00, and there is almost a 100% certainty that the difference between the samples is real and did not occur due to chance.

From the results shown in the table, the cosmetic company can conclude that its advertising campaign seems to be having its intended effect, raising awareness of the new perfume among the teenage girls.

Sign Test Statistics

The only requirement for data used in a paired sample sign test is that it must be possible to determine that one of a pair of values is greater than the other. As a result, the sign test is best suited for analyzing ordinal data. Ordinal data has values that can be arranged from lowest to highest; the distances among ordinal values cannot be measured.

Like the Wilcoxon test, the sign test is designed to look for differences between paired samples. In paired samples, observations in one sample are related to observations in another sample. An example of paired samples is a group of people who are asked if they recognize name brands before and after an advertising campaign; their responses before the campaign form one sample, and their responses after the campaign form the second sample. The responses for each person comprise a pair. Another example of paired samples contains the responses of men in one sample and the responses of women in another sample.

The first step in the sign test is to determine the observation in each pair that has a higher value. If the first value is higher than the second, a **+** is assigned to the pair. If the reverse is true, a **-** is assigned to the pair.

The next step is to count the number of pairs that have plus signs. If any pairs are tied, or have equal values, they are excluded from the analysis.

The sign test null hypothesis is that the sample distributions are the same and that the likelihood of the first value of a pair exceeding the second value is the same as the likelihood of the second value being greater than the first. In other words, the odds of either condition are 50/50. If the majority of signs are either pluses or minuses and the odds are not 50/50, it might be because the samples are different. To decide whether samples are different, the likelihood that the observed distribution occurred due to chance must be known. The NPPaired transformer automatically estimates that probability and issues it as a significance level. If the significance level is small, it is unlikely that the distribution occurred due to chance, and the null hypothesis can be rejected in favor of the alternate hypothesis that the samples are different.

The significance level provided by the NPPaired transformer is based on a two-tailed probability. A two-tailed probability measures the likelihood of one observed distribution not being equal to the other sample. If the important fact is that one distribution is either higher than or lower than another distribution, a one-tailed probability is appropriate. A one-tailed significance level is exactly half the value of a two-tailed significance level. Thus, although the transformer provides a two-tailed significance level, it is easy to calculate a one-tailed significance level.

In addition to comparing two samples, the NPPaired transformer can also compare an observed sample with a theoretical sample or distribution. In this case, the observed sample values are compared with the median of the theoretical distribution. If the observed value is higher than the specified median, a plus sign is assigned to the observation. If the observed value is lower than the specified median, a minus sign is assigned to the observation. If any observations are equal to the median, they are excluded from the analysis. From this point, the calculation of a one-sample sign test is identical to the calculation of a two-sample sign test. The number of plus signs are counted, and the probability of that count for the given number of observations is used to accept or reject the null hypothesis.

In the one-sample test, the null hypothesis is that the observed distribution is the same as the theoretical distribution. If the two-tailed probability of getting the observed number of pluses is high, the null hypothesis that there is no real difference between the observed distribution and the theoretical distribution is accepted. However, if the probability of the observed number of pluses is quite low, the null hypothesis is rejected in favor of the alternate hypothesis that the distributions are different.

Wilcoxon Signed-Rank Test Statistics

Like the two-sample sign test, the two-sample Wilcoxon signed-rank test is designed to compare paired samples. Although the sign test simply determines the direction of differences between two values, the Wilcoxon test measures the relative magnitude of differences between pairs. In other words, pairs that show large differences have more weight than pairs that have small differences. As a result, the Wilcoxon test is best suited for analyzing ordinal-level or interval-level data. (Interval level data has values that can be ordered from lowest to highest; the distance among interval values can be precisely measured.)

The first step in the Wilcoxon signed-rank test is to subtract the values in the second sample from values in the first sample. If the values in a pair are equal, the pair is excluded from the analysis. Next, the absolute (without plus or minus signs) differences are ranked from lowest to highest. When differences are tied, the average of the ranks that would have been assigned if there were no ties is given to the tied values. For example, if two pairs have the lowest value, they are both assigned the rank of 1.5 (the average of 1 and 2). The ranks are then given the sign (plus or minus) of the difference upon which they are based. The positive-signed ranks are then summed.

If the null hypothesis is true and there is no difference between the two samples, the sum of the positive-signed ranks should be relatively small. If the sum of the positive-signed ranks is large relative to the number of observations, the null hypothesis is probably not true. To judge whether the null hypothesis should be accepted or rejected, the chances of getting the observed sum of positive ranks must be known. The NPPaired transformer automatically calculates the two-tailed probability of finding the observed sum of ranks due to chance; this value is called the significance level. If it is large, there is a good chance of finding the observed sum by chance and the null hypothesis should be accepted. If the significance level is small, it is unlikely that the sum occurred by chance, and it is likely that a real difference exists between the samples.

Like the sign test, the Wilcoxon signed-rank test offers a one-sample test in addition to the paired-sample test. In the one-sample test, the specified median of the theoretical distribution is subtracted from each of the observations. If the median is equal to the observed value, the observation is excluded from the analysis. The remaining steps are the same as in a two-sample test. The absolute values of differences are then ranked, and the sign of the original differences is applied to the ranks. The positive-signed ranks are then summed. If the observed sample has the same distribution as the one summarized by the specified median (the null hypothesis), the result should be a relatively small sum of ranks. Alternatively, if the sum of the ranks is very large, the observed distribution is probably different from the specified distribution (the alternate hypothesis).

To accept or reject the null hypothesis, the probability of obtaining the given sum of signed ranks should be known. If the two-tailed significance supplied by the transformer is high, the difference is likely to result from chance, and the null hypothesis that there is no real difference should be accepted. If the probability is low, the observed difference is unlikely to occur due to chance, and the null hypothesis should be rejected because there is a real difference between the observed and theoretical differences.

Comparison of Signed and Wilcoxon Signed-Rank Tests

Although the sign test and the Wilcoxon signed-rank test use similar methods, they have several significant differences that make one or the other better for specific applications. The sign test uses only the direction of a difference in samples, whereas the Wilcoxon test uses the direction and the relative size of the difference. Because the Wilcoxon test uses more information about differences, it is more likely to find real differences between samples. In fact, in large ordinal-level or interval-level distributions, the sign test will find only one-third to two-thirds of the differences found by the Wilcoxon signed-rank test.

In spite of this disadvantage, the sign test is a more appropriate measure of distribution differences than the Wilcoxon test when the variables are dichotomous (have two possible values such as yes or no) that are arbitrarily given numeric values. In addition, the sign test is usually more effective in finding differences when a sample has many tied values.

Sign Test Formulas

The definitions in Table 53 apply to the formulas in this section.

Table 53. Sign test formula symbol definitions

Symbol	Definition
A_i	Value of any given observation in the first observed sample
B_i	Value of any given observation in the second observed sample
D_i	Difference between the first sample and either the second sample or the specified median for any given observation
M	Theoretical median in a one-sample test
N	Number of observations or pairs of observations
z	Normal z-score for the observed *S*
S	Standard deviation of the mean number of plus signs for the sample
S_i	Sign (either plus or minus) of the difference for any given observation
S_p	Number of the plus signs for any data set
$\bar{X}$	Mean number of plus signs for the sample

For any observation, the difference between the first sample and the second sample is calculated as:

$$D_i = A_i - B_i$$

For any observation, the difference between the first sample and the theoretical median is calculated as:

$$D_i = A_i - M$$

For any observation, the sign (S_i) is the plus or minus sign preceding the difference value. A space preceding a difference value is interpreted as a plus.

The number of plus signs for a sample (S_p) is the count of observations with a sign (S_i) equal to a plus.

The mean number of plus signs is calculated as:

$$\bar{X} = \frac{N}{2}$$

The standard deviation of the mean number of plus signs is calculated as:

$$S = 0.5 * \sqrt{N}$$

If the number of observed plus signs is less than the mean number of plus signs, the Z-score for the number of observed plus signs is calculated as follows:

$$z = \frac{(S_p + 0.5 - \bar{X})}{S}$$

If the number of observed plus signs is greater than the theoretical mean number of plus signs, the Z-score for the number of observed plus signs is calculated as follows:

$$z = \frac{(S_p - \bar{X} - 0.5)}{S}$$

The transformer uses the normal distribution to automatically convert the Z-score into a probability, which is the likelihood of the observed number of plus signs occurring due to chance. If the probability is large, the null hypothesis that there is no difference between the two distributions should be accepted. If it is small, the null hypothesis should be rejected in favor of the alternate hypothesis that the samples are different.

Wilcoxon Signed-Rank Test Formulas

The definitions in Table 54 apply to the formulas in this section.

Table 54. Wilcoxon signed-rank test formula symbol definitions

Symbol	Definition
A_i	Value of any given observation in the first observed sample
B_i	Value of any given observation in the second observed sample
D_i	Difference between the first sample and either the second sample or the specified median for any given observation
M	Theoretical median in a one-sample test
N	Number of observations or pairs of observations
R_i	Rank of the absolute difference for any given observation
z	Normal Z-score for the observed S
S	Standard deviation of the mean rank for the sample
S_i	Signed rank of the difference for any given observation
S_p	Sum of the positive signed ranks for any set of samples
$\bar{X}$	Expected rank mean for the sample

For any observation in a two-sample test, the difference is calculated as:

$$D_i = A_i - B_i$$

For any observation in a one-sample test, the difference is calculated as:

$$D_i = A_i - M$$

For any observation, the rank (R_i) of the difference is the integer rank associated with the absolute difference value. For example, if a sample had three differences (2, -3, -4), the ranks of the absolute differences would be 1, 2, and 3.

For any observation, the signed rank (S_i) of the difference is the integer rank as calculated with the sign of the original difference. For example, if a sample had three differences (2, -3, -4), the signed ranks of the differences would be 1, -2, and -3.

The sum of the positive signed ranks for any set of samples is calculated as:

$$S_p = \sum_{i=1}^{N} +S_i$$

The expected mean sum of ranks is calculated as:

$$\bar{X} = \frac{N \divideontimes (N+1)}{4}$$

The standard deviation of the expected sums of ranks is calculated as:

$$S = \sqrt{\frac{N \divideontimes (N+1) \divideontimes (2 \divideontimes N + 1)}{24}}$$

The corrected Z-score for the observed sum of signed ranks is calculated as:

$$z = \frac{S_p - \bar{X}}{S}$$

To determine whether the null hypothesis should be accepted, the NPPaired transformer uses the normal distribution to automatically convert the Z-score into a probability, which is issued as a significance level. The probability represents the likelihood that the observed sum of signed ranks occurred due to chance and that there is no real difference between the two samples. If the probability is large, the null hypothesis is accepted. If it is small, the null hypothesis is rejected in favor of the alternate hypothesis that there is a difference between the two samples.

Specifying Grouping Features

A grouping expression consists of three parts: a group column, an optional list of grouping criteria, and an optional group type specifier. Each group is treated as a distinct set of data, and a set of sign tests is generated for every requested group.

The group column must be a single data column that determines the group to which a particular data element belongs. For example, if the first column of the input data contains the grouping information, the entry would be *a*, for column A.

A list of grouping criteria can follow the column name to specify the groups that are created. The criteria can be text values, numeric values, or dates, each separated by a comma. If grouping criteria are not present, the transformer creates a group for every unique value of the group column.

The group type specifier controls whether the grouping criteria specify the members of a group or the limits of a range. If the type specifier *only* is present, a group is created for each item in the grouping criteria list; only values that exactly match a particular grouping criterion are added to the corresponding group. If the type specifier is not present, the grouping criteria are treated as the end points of a series of ranges. Any value that is greater than the first end point and less than or equal to the second end point is treated as part of that range.

Examples of each of the four possible variations of a grouping expression are:

<no expression>
One group is created containing all of the input data.

a A group is created for each unique value in column A.

a, 10, 20
Three ranges are created: all values up to and including 10, all values above 10 but less than or equal to 20, and all values above 20.

a, 10, 20, only
Two groups are created: all values where column A is 10, and all values where column A is 20.

Chapter 5. Regression and Time Series Analysis Transformers

Regression and Time Series Analysis transformers perform the following tasks:

- Smooth time series
- Create summary statistics on groups of data
- Calculate seasonality factors
- Forecast trends
- Identify relationships
- Regression analysis

Using these transformers in the capsule, you can create multiple-regression applications that include full-modeling, forward, backward, and stepwise procedures, and identification and removal of outlying data. Statistics generated through these transformers include significance level, tolerance, autocorrelation, Durbin-Watson, F-statistics, t-statistics, and multiple indexes of determination.

Several transformers comprise the Regression and Time Series Analysis transformers group; each transformer is designed to perform a specific function. For example, you can use the Correlation transformer to measure linear relationships among columns of data or variables, or use the Regression transformer to create a mathematical model or equation for a set of data.

You can use the Transformer Execution Language (TXL) with some Regression and Time Series Analysis transformers. For more information, see "Appendix 6. Transformer Execution Language," on page 497.

To locate the Regression and Time Series Analysis transformers:

1. Open the New Icons file drawer.
2. Open the Transformer Icons file drawer.

If you cannot find a specific transformer, see your system administrator

Table 55 lists the Regression and Time Series Analysis transformers, describes their functions, and gives the page number where each transformer is described.

Table 55. Regression and Time Series Analysis transformers

Transformer	Function	See
AutoCorrelation	Helps you analyze the relationship between values in a time series and previous values (lags) of the series.	"AutoCorrelation" on page 346

Table 55. Regression and Time Series Analysis transformers

Transformer	Function	See
AutoRegression	Helps you construct models that estimate future values of a time series based on previous values.	“AutoRegression” on page 365
Correlation	Measures linear relationships among columns of data or variables.	“Correlation” on page 385
Elementary	Generates summary statistics about variable distributions.	“Elementary” on page 398
Forecast	Helps you predict values of a series based on historical information.	“Forecast” on page 411
Moving	Computes moving averages and rolling sums to smooth and summarize time series information.	“Moving” on page 440
Regression	Helps you create a mathematical model or equation for a set of data, so you can analyze the relationships within that model or equation and estimate values of a variable based on one or more predictor variables.	“Regression” on page 454
Seasonality	Measures the seasonal component of a time series and optionally removes its effect.	“Seasonality” on page 474

For general information on using transformers, see “Chapter 1. Getting Started with Transformers,” on page 1.

AutoCorrelation

The AutoCorrelation transformer calculates the following autocorrelation (serial correlation) statistics:

- Autocorrelation coefficient *r*
- Partial autocorrelation coefficient *pr*

The autocorrelation coefficient *r* is similar to the standard correlation coefficient in that it describes the association or mutual dependence among values of two time series data sets. However, whereas the Correlation transformer uses two distinct variables, the AutoCorrelation transformer uses two pieces of the same variable taken from different time periods. The partial autocorrelation coefficient *pr* is similar to the standard partial correlation coefficient. It describes the correlation relationship between the residual error components of various data points separated by time.

An autocorrelation coefficient varies from -1, which indicates a perfectly linear negative correlation, to +1, which indicates a perfectly linear positive correlation. The autocorrelation and partial autocorrelation coefficients are calculated for lags, which represent the number of periods between current and previous periods. For example, an autocorrelation for the sixth lag represents the relationship between current period values and values six previous periods.

Analysis of autocorrelation coefficients provides the following information:

- Structure of a data set
- Patterns within a data set
- Trend
- Length of seasonality
- Degree of randomness

A stationary time series data set has values that fall near a constant mean. Such a series shows no growth or decline over time. A nonstationary series does not have a constant mean; the nonstationary component is often referred to as the trend. When a trend is observed, a differencing method should be applied before analyzing the autocorrelation results.

The partial autocorrelation coefficient is a measure of correlation. It is used to identify the extent of the relationship between current values of a variable and earlier values, while holding the effects of all other lags constant.

The estimated partial autocorrelation coefficient *pr* can also be used to determine if a data series is stationary. The following test should be performed if the results of autocorrelation are to be used with an autoregression or autoregressive integrated moving average (ARIMA) forecast model. First, the order of the model is determined. Order is described as *AR(p)* where *p* is the time lag value of the largest positive autocorrelation coefficient. In an AR($_1$) model, the stationary requirement is:

$$|pr_1| < 1$$

With p>1, the test becomes the sum of all absolute partial autocorrelations:

$$|pr_1| + |pr_2| + ... + |pr_i| < 1$$

The notation pr_i refers to the *i*th partial autocorrelation value, and the vertical bars indicate that the absolute value should be taken (the **+** or **-** sign is disregarded).

Parameters

All of the parameters described in this section are displayed in the Transformer Controls window. Most of these parameters are also displayed in the Data Entry icon contained in the transformer's demonstration capsule application. However, to simplify use of the transformer for new users, some of the more advanced parameters might be omitted from the Data Entry icon; they can be accessed only in the Transformer Controls window.

Number of header rows (; 1; 5;) row containing titles (,1; ,5;)

This parameter specifies the number of rows in the input data that are skipped and not used in calculating statistics. This parameter also specifies the row number containing column titles that are used as labels in the output. The two values should be separated by a comma and space. You can specify any number of header rows; the default value is 0. The title row number should be less than or equal to the number of header rows. If the header rows value is specified, the default title row is the last row of the header rows. If no header is specified, the default is no title row.

Report name (; AutoCorrelation Test;)

This parameter is a title for the output report, for example, the *AutoCorrelation Test*. If the `Report name` parameter is blank, there is no title output.

Data column (a; b; c;)

This parameter specifies the columns used to compute the autocorrelation statistics. At least one data column is required, but any number of data columns can be specified. For example, `a, b` computes autocorrelations for both column A and column B.

Length of long term 'Seasonal' differencing (; 0; 12; 24;)

This parameter specifies the lag used to compute a long-term difference. Any whole number is allowed. For example, if there are 12 periods in a season, you would type `12`, which would result in the creation of a new series in which the value for January 1990 is the result of subtracting the raw value for January 1989 from the raw value of January 1990, and the value for January 1989 is the result of subtracting the raw value for January 1988 from the raw value of January 1989. The default value is 0 (no differencing). For more information on seasonal differencing, see “Differencing Method” on page 362.

Length of short term 'Trend' differencing (; 0; 1; 2;)

This parameter specifies the number of times that short-term or trend differencing is to be applied to the input data. Any whole number is allowed. For example, if the time series has a curved upward trend, you would type `2`, which specifies that the input data should be trend-differenced twice. The default value is 0 (no differencing). For more information on short-term differencing, see “Differencing Method” on page 362.

Number of time lags for auto/partial correlations (12; 20; 30;)

This parameter specifies the upper limit of the lags for which autocorrelation statistics will be computed. Any whole number can be specified. If the value exceeds the number of data rows in Input 1, it is reset to the default value which is one less than the number of rows remaining in the differenced series.

Output differenced series? (;y; n)

This parameter specifies whether the results of each differencing step are sent to Output 4. Valid responses are `Yes` or `No`, which can be abbreviated to `Y` or `N`, respectively. The default is `No`.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- Coefficients (Output 1)
- Results (Output 2)
- Partials (Output 3)
- Series (Output 4)
- Messages (Output 5)

When you click on any one of these choices, the data associated with that choice displays in the display area of the transformer.

Input Region Names

The AutoCorrelation transformer has only one input region, Input 1 (Data). You can modify the input name to reflect the name of the corresponding data in the display area. Input 1 consists of data read from a tool, such as the Spreadsheet, Query, or SQL Entry tools, or copied directly into the transformer. Input 1 is treated as a series of columns. These columns can contain statistical data (data columns). The input data can have any number of header rows. Column titles are read from the specified row; however, if titles are not found, there will be no titles.

Whenever a variable in the data set has missing values, the entire observation or row is excluded from all calculations. This holds true even if the variables with the missing values are not used in the actual calculation or if the row or observation contains some partial data.

Consider the following example. A data set contains variables A and B. Variable A contains 10 observations, and variable B contains 12. In this data set, the first ten observations would be used in all calculations and the last two would be excluded, even in calculations where only variable B is specified.

Whenever the transformer encounters an observation with missing values, a message is displayed in the Important Message window or in the Capsule icon run log. The message informs the user that the particular observation will be excluded from the calculation.

Output Region Names

The AutoCorrelation transformer generates five output regions; none has a size limit.

Output 1 (Coefficients) is a report of statistics computed for each variable (input data column), its autocorrelation, and partial autocorrelation. The statistics include:

- Count
- Mean
- Standard error
- Maximum value
- Minimum value
- First-half mean (for the autocorrelation only)
- Second-half mean (for the autocorrelation only)

Output 2 (Results) contains the autocorrelation results. The information provided for each variable includes:

- Time lag number
- Autocorrelation coefficient, r_k
- t-value for each autocorrelation
- Upper confidence limit value
- Lower confidence limit value

Output 3 (Partials) contains the partial autocorrelation results. The information provided for each variable includes:

- Time lag number
- Partial autocorrelation coefficient
- t-value for each partial autocorrelation
- Upper confidence limit value
- Lower confidence limit value

Output 4 (Series), if requested, contains the original series and the results of each level of differencing.

Output 5 (Messages) contains:

- Transformer run-time messages
- Warnings
- Error messages

Examples

This example uses two views (AutoCorrelation Example Case #1 and Case #2) of the same data. Assume the following data, which represents the monthly closing price of Company XYZ common stock, is sent to Input 1 of the AutoCorrelation transformer:

Table 56. Input data for the AutoCorrelation transformer

Date	Closing Price
January, 1980	61.500
February, 1980	62.125
March, 1980	61.500
April, 1980	64.500
May, 1980	64.250
June, 1980	63.875
July, 1980	64.375
August, 1980	62.375
September, 1980	62.000
October, 1980	62.125
November, 1980	62.625
December, 1980	62.125
January, 1981	60.750
February, 1981	61.625
March, 1981	61.375
April, 1981	59.375
May, 1981	58.500
June, 1981	58.500
July, 1981	58.000
August, 1981	58.250
September, 1981	57.500
October, 1981	56.875
November, 1981	57.500
December, 1981	58.625
January, 1982	58.000
February, 1982	58.125
March, 1982	57.250
April, 1982	57.375
May, 1982	57.450
June, 1982	57.250
July, 1982	57.625

Table 56. Input data for the AutoCorrelation transformer

August, 1982	58.500
September, 1982	58.250
October, 1982	56.750
November, 1982	56.375
December, 1982	56.375
January, 1983	56.125
February, 1983	55.000
March, 1983	54.875
April, 1983	55.000
May, 1983	52.875
June, 1983	52.250
July, 1983	52.750
August, 1983	53.375
September, 1983	53.250
October, 1983	53.250
November, 1983	53.375
December, 1983	53.625
January, 1984	53.875
February, 1984	53.000
March, 1984	51.750
April, 1984	52.125

The AutoCorrelation transformer parameters are set as follows:

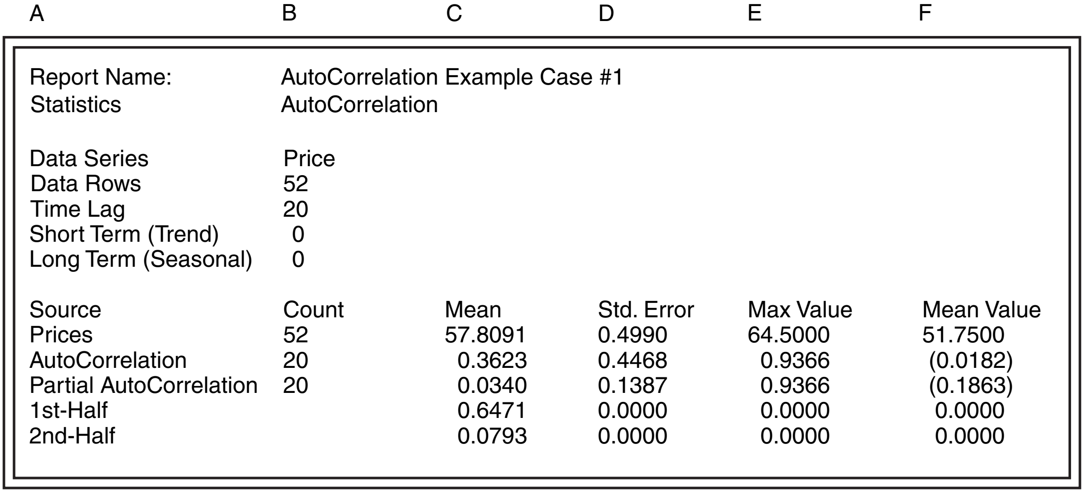
Number of header rows
1

Report name
AutoCorrelation Example Case #1

Data column
b

Number of times lags for auto/partial correlations
20

When parameters are not specified, the transformer uses the default value. When the transformer run finishes, the following report is displayed in Output 1:



```
A                          B                   C                  D                  E                    F

Report Name:               AutoCorrelation Example Case #1
Statistics                 AutoCorrelation

Data Series                Price
Data Rows                  52
Time Lag                   20
Short Term (Trend)          0
Long Term (Seasonal)        0

Source                     Count               Mean               Std. Error         Max Value            Mean Value
Prices                     52                  57.8091            0.4990             64.5000              51.7500
AutoCorrelation            20                    0.3623           0.4468               0.9366              (0.0182)
Partial AutoCorrelation    20                    0.0340           0.1387               0.9366              (0.1863)
1st-Half                                         0.6471           0.0000               0.0000               0.0000
2nd-Half                                         0.0793           0.0000               0.0000               0.0000
```

AutoCorrelation

The first-half mean is very different than the mean of the second half. This indicates that a trend is present in the data. The next report is displayed in Output 2 (autocorrelation result):

A	B	C	D	E	F
Variable	Time Lag	Autocorrelation	T-Value	Upper 2*Std. Error	Lower 2*Std. Error
Prices	1	0.9366	6.7540	0.2774	(0.2774)
	2	0.8660	3.7639	0.4603	(0.4603)
	3	0.8179	2.8593	0.5721	(0.5721)
	4	0.7533	2.2969	0.6559	(0.6559)
	5	0.6865	1.9084	0.7194	(0.7194)
	6	0.6157	1.6030	0.7682	(0.7682)
	7	0.5457	1.3555	0.8052	(0.8052)
	8	0.4836	1.1609	0.8332	(0.8332)
	9	0.4187	0.9800	0.8545	(0.8545)
	10	0.3466	0.7966	0.8702	(0.8702)
	11	0.2651	0.6020	0.8807	(0.8807)
	12	0.1959	0.4418	0.8868	(0.8868)
	13	0.1553	0.3490	0.8901	(0.8901)
	14	0.1049	0.2351	0.8922	(0.8922)
	15	0.0557	0.1246	0.8932	(0.8932)
	16	0.0339	0.0760	0.8934	(0.8934)
	17	0.0170	0.0382	0.8935	(0.8935)
	18	(0.0025)	(0.0055)	0.8936	(0.8936)
	19	(0.0138)	(0.0309)	0.8936	(0.8936)
	20	(0.0182)	(0.0407)	0.8936	(0.8936)
Chi-Square = 282.68 For Degree Of Freedom = 19 With Probability = 0.00000					
	22	(0.03)	(0.08)	0.89	(0.89)

Because it is much easier to spot patterns when data is presented in a chart, the example plot, shown in Figure 45, is constructed to show the results in Output 2.

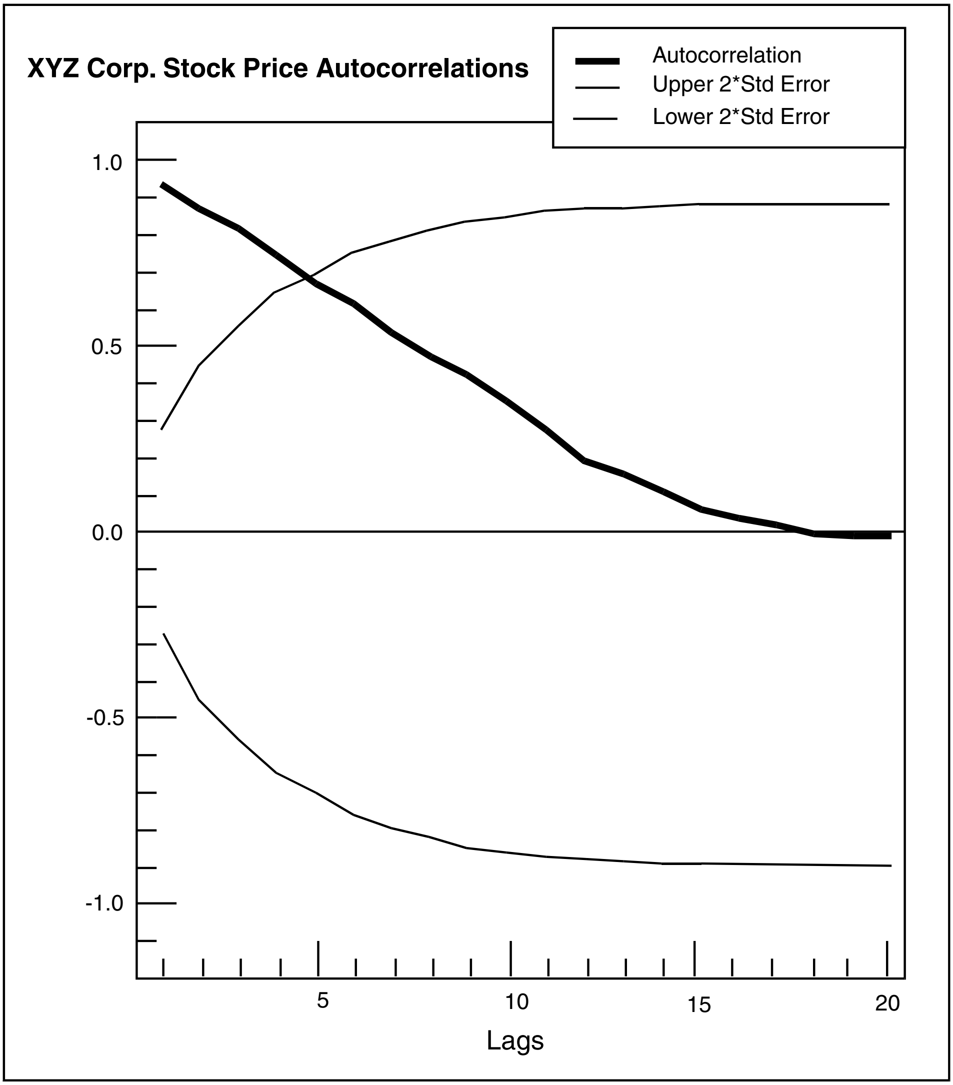
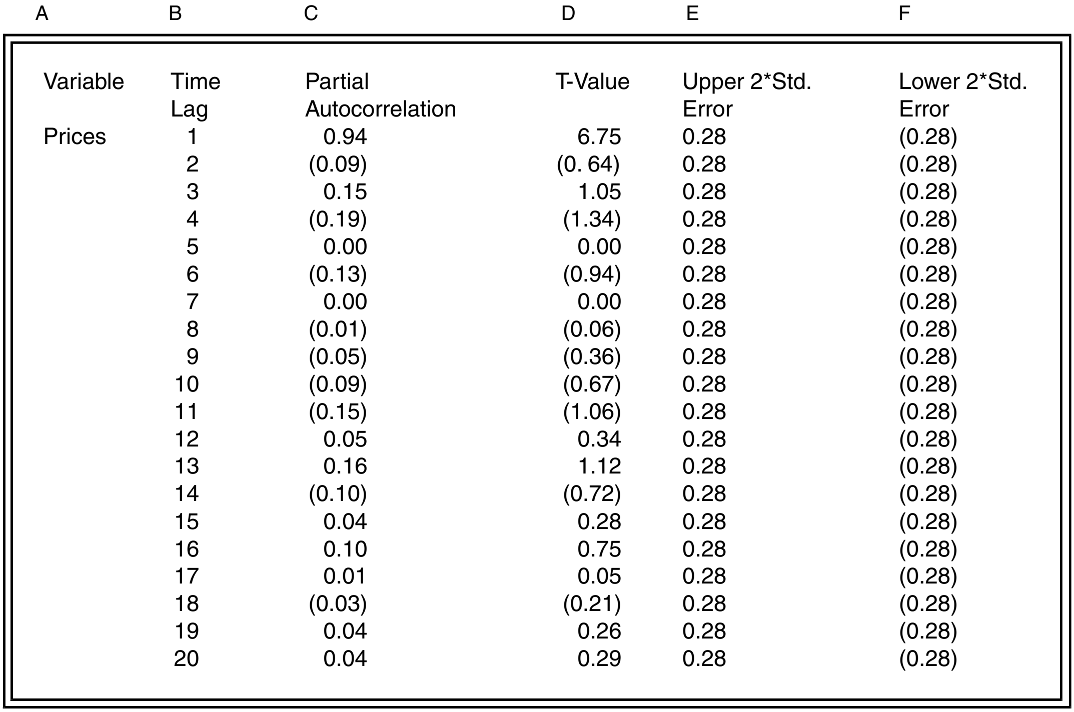


Figure 45. Autocorrelation for Case #1 chart

The estimated autocorrelation coefficient drops gradually toward 0. In fact, the first four autocorrelation coefficients fall outside of the two times standard error boundary. This boundary forms the 95% confidence limit, which indicates that there is a 95% likelihood that the autocorrelations are real and did not happen by chance. The fact that so many of the initial autocorrelations are outside the confidence limit is characteristic of a nonstationary data series that could benefit from differencing.

AutoCorrelation

The next report, which is the table of partial autocorrelation coefficients, is displayed in Output 3:



A	B	C	D	E	F
Variable	Time Lag	Partial Autocorrelation	T-Value	Upper 2*Std. Error	Lower 2*Std. Error
Prices	1	0.94	6.75	0.28	(0.28)
	2	(0.09)	(0. 64)	0.28	(0.28)
	3	0.15	1.05	0.28	(0.28)
	4	(0.19)	(1.34)	0.28	(0.28)
	5	0.00	0.00	0.28	(0.28)
	6	(0.13)	(0.94)	0.28	(0.28)
	7	0.00	0.00	0.28	(0.28)
	8	(0.01)	(0.06)	0.28	(0.28)
	9	(0.05)	(0.36)	0.28	(0.28)
	10	(0.09)	(0.67)	0.28	(0.28)
	11	(0.15)	(1.06)	0.28	(0.28)
	12	0.05	0.34	0.28	(0.28)
	13	0.16	1.12	0.28	(0.28)
	14	(0.10)	(0.72)	0.28	(0.28)
	15	0.04	0.28	0.28	(0.28)
	16	0.10	0.75	0.28	(0.28)
	17	0.01	0.05	0.28	(0.28)
	18	(0.03)	(0.21)	0.28	(0.28)
	19	0.04	0.26	0.28	(0.28)
	20	0.04	0.29	0.28	(0.28)

The example plot in Figure 46 is generated with the partial autocorrelation data.

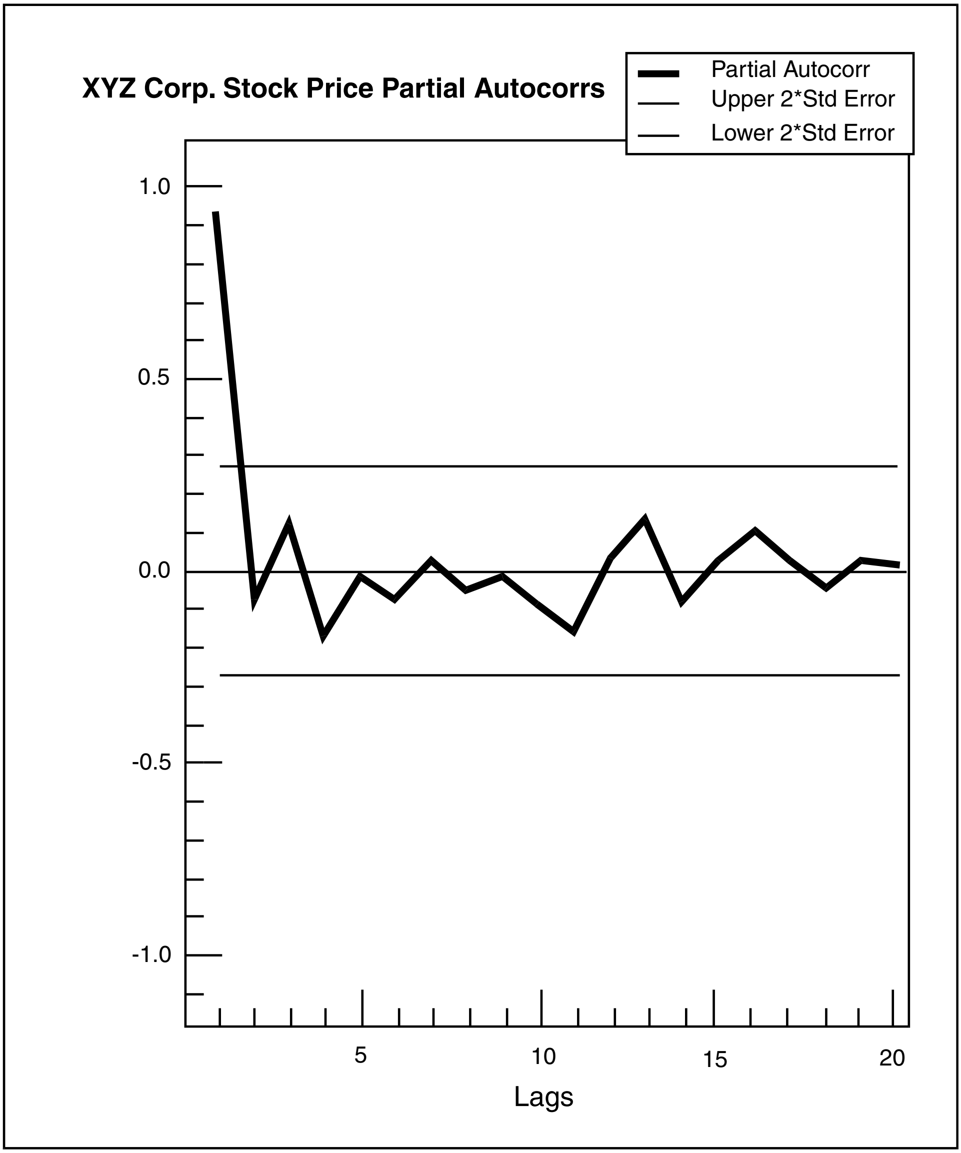


Figure 46. Partial autocorrelation for Case #1 chart

The estimated partial autocorrelation coefficients are centered around 0. Because there is only one value that lies outside of the confidence limit boundaries, the order of this model is 1. This first order autocorrelation model has the stationary condition satisfied because `|pr1| = 0.934 < 1`. However, even though `|pr1|` is less than 1, it is still fairly close to 1, which can diminish its statistical significance.

The results shown in Figure 46 indicate that the data should be differenced. To enable the differencing algorithm, the AutoCorrelation transformer parameters are set as follows:

Number of header rows
1

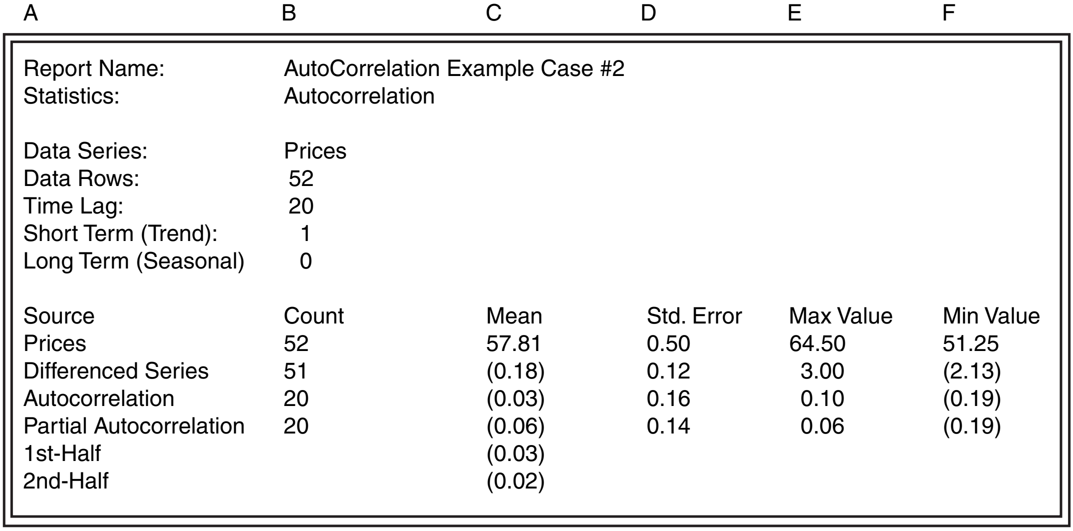
Report name
AutoCorrelation Example Case #2

Data column
b

Length of short term 'Seasonal' differencing
1

Number of times lags for auto/partial correlations
20

The `Length of Short Term 'Trend' Differencing` parameter is set to 1 to specify that the data should be differenced once. When the transformer has finished, the following report is displayed in Output 1:



```
A                         B                    C               D              E              F

Report Name:              AutoCorrelation Example Case #2
Statistics:               Autocorrelation

Data Series:              Prices
Data Rows:                 52
Time Lag:                  20
Short Term (Trend):         1
Long Term (Seasonal)        0

Source                    Count                Mean            Std. Error     Max Value      Min Value
Prices                    52                   57.81           0.50           64.50          51.25
Differenced Series        51                   (0.18)          0.12             3.00         (2.13)
Autocorrelation           20                   (0.03)          0.16             0.10         (0.19)
Partial Autocorrelation   20                   (0.06)          0.14             0.06         (0.19)
1st-Half                                       (0.03)
2nd-Half                                       (0.02)
```

The result of Output 1 shows an improvement over the earlier results. The mean of the first half is much closer to the mean of the last half indicating that the effect of the nonstationary mean is much less pronounced after the first difference.

The following table is the Output 2 autocorrelation result:

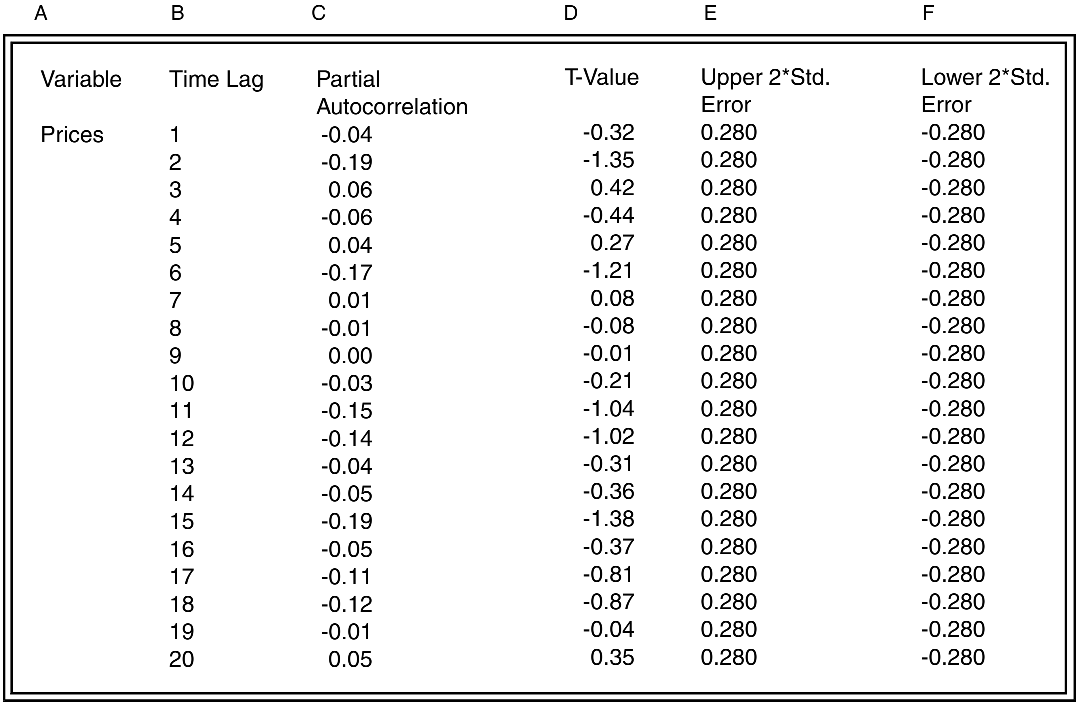
A	B	C	D	E	F
Variable	Time Lag	Autocorrelation	T-Value	Upper 2*Std. Error	Lower 2*Std. Error
Prices	1	-0.04	-0.32	0.28	-0.28
	2	-0.19	-1.33	0.28	-0.28
	3	0.07	0.52	0.29	-0.29
	4	-0.03	-0.20	0.29	-0.29
	5	0.01	0.10	0.29	-0.29
	6	-0.14	-0.98	0.29	-0.29
	7	0.00	0.03	0.29	-0.29
	8	0.06	0.38	0.30	-0.30
	9	-0.03	-0.22	0.30	-0.30
	10	-0.02	-0.11	0.30	-0.30
	11	-0.13	-0.88	0.30	-0.30
	12	-0.10	-0.63	0.30	-0.30
	13	0.03	0.18	0.30	-0.30
	14	-0.02	-0.14	0.31	-0.31
	15	-0.16	-0.12	0.31	-0.31
	16	0.01	0.08	0.31	-0.31
	17	0.02	0.14	0.31	-0.31
	18	-0.06	-0.40	0.31	-0.31
	19	0.06	0.39	0.31	-0.31
	20	0.10	0.63	0.31	-0.31

Chi-Square = 9.38 For Degree Of Freedom = 19 With Probability = 0.96671

The estimated autocorrelation coefficients drop off quickly towards 0 after the third time lag. Another important feature of this result is that all autocorrelation coefficient values fall within the boundary of two standard error units, which is typical of a data series with a stationary mean. Further differencing should not be required. Because there now is a stationary mean, you can determine the degree of the model. The largest positive autocorrelation coefficient is at lag 20, with lag 3 being the next best choice.

This result indicates that an appropriate autoregression model would be composed of two terms, the 20th and the 3rd lags.

In Output 3 of the partial autocorrelations shown in the following example, the estimated partial autocorrelation coefficient values are scattered around 0. This data is suited for a three-degree autoregression model; the stationary condition for an *AR(2)* model is satisfied because `|pr3 + pr7|= 0.07 < 1`.



A	B	C	D	E	F
Variable	Time Lag	Partial Autocorrelation	T-Value	Upper 2*Std. Error	Lower 2*Std. Error
Prices	1	-0.04	-0.32	0.280	-0.280
	2	-0.19	-1.35	0.280	-0.280
	3	0.06	0.42	0.280	-0.280
	4	-0.06	-0.44	0.280	-0.280
	5	0.04	0.27	0.280	-0.280
	6	-0.17	-1.21	0.280	-0.280
	7	0.01	0.08	0.280	-0.280
	8	-0.01	-0.08	0.280	-0.280
	9	0.00	-0.01	0.280	-0.280
	10	-0.03	-0.21	0.280	-0.280
	11	-0.15	-1.04	0.280	-0.280
	12	-0.14	-1.02	0.280	-0.280
	13	-0.04	-0.31	0.280	-0.280
	14	-0.05	-0.36	0.280	-0.280
	15	-0.19	-1.38	0.280	-0.280
	16	-0.05	-0.37	0.280	-0.280
	17	-0.11	-0.81	0.280	-0.280
	18	-0.12	-0.87	0.280	-0.280
	19	-0.01	-0.04	0.280	-0.280
	20	0.05	0.35	0.280	-0.280

This example was based on monthly time series data. The usual question when dealing with time-series data is whether a statistically significant amount of seasonal variations are present in the data. In this case, the answer is no. The largest autocorrelation values are found at 3 and 20 time lags in Case #2. There is no indication of a 12-month cycle in this data.

AutoCorrelation Transformer Formulas

The definitions described in Table 57 apply to the equations used in this section.

Table 57. AutoCorrelation symbol definitions

Symbol	Definition
k	A particular time lag
K	Maximum time lag value (specified as a parameter)
N	Number of observations in the input data set
pr_k	Partial autocorrelation coefficient at time lag *k*

Table 57. AutoCorrelation symbol definitions

Symbol	Definition
r_k	AutoCorrelation coefficient at time lag *k*
$s(pr_k)$	Standard error of the partial autocorrelation
$s(r_k)$	Standard error of the autocorrelation
t_k	The t-statistic for autocorrelation at time lag *k*
t'_k	The t-statistic for partial autocorrelation at time lag *k*
Z	A data set (column in the input data)
z_i	A particular value in the data set *z*
z	Mean value of z
z_s	Difference between any value and its mean

The estimated autocorrelation coefficient r_k is:

$$- \quad \sum \quad \times \quad - \quad +$$

$$r_k = \frac{\sum_{i=1}^{N-k} (z_i - \bar{z}) \times (z_{i+k} - \bar{z})}{\sum_{i=1}^{N} (z_i - \bar{z})^2}$$

The estimated partial autocorrelation coefficients are computed in a *k x k* matrix. The matrix is set up using the following three formulas:

$$pr_{1,1} = r_1$$

$$pr_{a,b} = pr_{a-1,b} - pr_{a,a} \times pr_{a-1,a-b}$$

for `a=2, 3,¼, k; b=1, 2, ¼, k-1`

$$pr_{m,m} = \frac{r_m - \sum_{n=1}^{k} pr_{m-1,n} \times r_{m-n}}{1 - \sum_{n=1}^{k} (pr_{m-1,n} \times r_n)}$$

for `m = 2, 3, ¼, k`

The estimated partial autocorrelation coefficients, pr_k, are taken from the main diagonal of the *k x k* matrix. Thus,

$$pr_k = pr_{a,b}$$

for `a = k; b = k`

The standard error of the sampling distribution of r_k is calculated as follows:

$$s(r_k) = \sqrt{\frac{1 + 2 \times \sum_{j=1}^{k-1} r_j^2}{N}}$$

Similarly, the standard error of the sampling distribution of pr_k is calculated as follows:

$$s(pr_k) = \frac{1}{\sqrt{N}}$$

The autocorrelation null hypothesis test H_o*:* $r_k = 0$ is the basis for the t-statistic:

$$t_k = \frac{r_k}{s(r_k)}$$

The partial autocorrelation null hypothesis test H_o*:* $pr_k = 0$ also is the basis for the t-statistic:

$$t'_k = \frac{pr_k}{s(pr_k)}$$

Differencing Method

Differencing is a method for removing seasonality and trend effects from a series before autocorrelation or autoregression statistics are computed. A differenced series is the result of subtracting lag values from the values of the original series.

For an autoregression or ARIMA model to produce valid results, the input series must be stationary (no trend) and relatively free of seasonal effects. (ARIMA models provide a method of describing both stationary and nonstationary time series.) A stationary series has values that fall near a constant mean; it shows no growth or decline over time. A nonstationary series does not have a constant mean. The nonstationary component is often referred to as the trend.

A seasonal effect is a cyclical increase or decrease in the series. For example, the sales of suntan oil is much higher in the early summer than during the rest of the year; thus, suntan oil sales have an annual seasonal component. Each year, they go through a period of growth and decline that is quite stable.

A plot of the autocorrelation coefficients versus the time lag number is used to test the presence of trend or seasonal effects. If the plot of autocorrelation coefficients includes a few high values for the first one to three lags, with the rest falling to 0, the data set has little or no trend (stationary mean). Alternatively, if the autocorrelations for four or more of the first lags are significantly different than 0 and the plot forms a diagonal line, a trend exists (nonstationary mean). Another indication of a nonstationary mean is visible in the first-half/last-half mean values. If the series contains a significant trend, the mean of the first-half autocorrelation coefficient will be quite different from the mean of the last-half and neither value will be close to 0.

In a stationary series, plots of autocorrelation coefficients provide a good indication of seasonal effects. If the autocorrelations are arranged in wave patterns from low to high values with some values being significantly different from 0, it indicates that seasonality exists in the series. The length of seasonality is equal to the number of lags between crests of these waves.

Strong trend and seasonal effects can obscure autocorrelation relationships in a series. When this occurs, it is best to reduce the trend and seasonal effects by differencing a series at the beginning of the autocorrelation process. The AutoCorrelation transformer allows you to remove these components with two types of differencing: short-term (trend) differencing, and long-term (seasonal) differencing.

Short-term (trend) differencing consists of subtracting adjacent values of the data, using their difference as a new time series. Trend differencing is an iterative process; a series is differenced and the results of the differencing, in turn, can be differenced. Each successive difference is computed on the previous difference. Thus, the third difference would be computed by differencing the result of the second difference. The AutoCorrelation transformer allows you to specify any degree of short-term differencing. The following formulas illustrate how a two-degree short-term difference is computed.

The definitions described in Table 58 apply to the equations used in this section.

Table 58. Differencing formula symbol definitions

Symbol	Definition
i	A specific period in a series
N	Number of periods in the input series

Table 58. Differencing formula symbol definitions

Symbol	Definition
U	Result of first differencing
W	Result of second differencing
X	Original input data

The first short-term difference is computed from the original data:

$$U_i = X_i - X_{i-1}$$

for `i = 2, 3, ¼, N`

The second short-term difference is computed from the result of the first difference:

$$W_i = U_i - U_{i-1}$$

for `i = 3, 4, ¼, N`

The short-term differencing algorithm produces a series that has one less element for each degree of differencing.

Long-term or seasonal differencing works very much like short-term differencing. However, in long-term differencing, the value for a period one season ago is subtracted from the current period's value. For example, to compute a difference when the length of seasonality is one year and the current period is January 2000, the transformer would subtract the January 1999 value from the January 2000 value. With long-term differencing, the data set loses as many data points as are in a season. For example, if the original series contains 48 monthly data points and the length of a season is 12, long-term differencing produces a data set with 36 data points.

If both long-term and short-term differencing is specified, the transformer completes the long-term differencing first and then applies the short-term differencing. The following example illustrates how the transformer would apply a long-term differencing of six and a short-term differencing of two. This type of differencing would be useful for stabilizing a bimonthly series that has strong trend and seasonal components.

Original Series	Long-Term Differencing	1st Short-Term Differencing	2nd Short-Term Differencing

30			
26			
22			
18			
14			
10			
17	-13		
16	-10	3	
11	-11	-1	-4
7	-11	0	1
1	-13	-2	-2
0	-10	3	5

AutoRegression

The AutoRegression transformer performs the following autoregression-based analyses:

- Analysis of the past behavior of a time-series data set
- Prediction of the future behavior of a time-series data set

Because of their flexibility, autoregression models are widely used in time-series analysis. These models are also known as AR_p models, where *AR* represents autoregression and *p* represents the model order.

The autoregression technique is useful in identifying dependencies among data collected sequentially. These dependencies can then be used to project a time series into the future. The AutoRegression transformer is based upon the regression model. The major difference between regression and autoregression is in the selection of the independent variables. Regression uses one or more parallel time series as independent variables; autoregression creates independent variables from previous data points of the dependent variable. This time shift helps to identify cycles within a time series.

For example, regression can be used to analyze sales volume based upon levels of advertising, price, and prevailing interest rates. The AutoRegression transformer might perform the same sales volume analysis based upon sales levels 3, 6, 12, and 24 months prior to the current month. The Regression transformer is useful for finding relationships, whereas the AutoRegression transformer is best at locating patterns.

AutoRegression

A simple autoregression equation can be expressed as:

$$y_i = a + b \divideontimes y_{i-k}$$

This formula states that the value (y_i) for any point in time (i) is a multiple (b) of some previous value (y_i-k) at some point (k time periods) in the past, plus a constant (a). Autoregression attempts to find the typical parameter values (a, b) that best relate past and current data values.

A previous data value is known as a lag; the number of lags incorporated into an autoregression model is referred to as the degree of autoregression. A higher-degree autoregression algorithm incorporates more than one data pattern in the autoregression equation. When the degree (p) is greater than 1, the multiple autoregression equation is expressed as:

$$y_i = a + b_1 \divideontimes y_{i-1} + b_2 \divideontimes y_{i-2} + \ldots + b_p \divideontimes y_{i-p}$$

The success of autoregression depends upon the dependencies and randomness that exist among the time-series data values. A high degree of dependency among the data values improves the model. This condition can be verified by examining the pattern created by the autocorrelation coefficients. Randomness tends to degrade autoregression models. Small autocorrelation values indicate that randomness might exist in the data. Finally, autoregression models assume that the input time-series data is stationary (there is no trend). Nonstationary data can be converted to a stationary series by applying the short-term differencing algorithm in autoregression. The existence of a linear trend in the data is indicated by strong autocorrelation coefficients at the first several lags.

The autoregressive technique is most appropriate for short-term forecasting. The reliability of forecasts derived with autoregression deteriorates after four to five forecast periods.

Parameters

All of the parameters described in this section are displayed in the Transformer Controls window. Most of these parameters are also displayed in the Data Entry icon contained in the transformer's demonstration capsule application. However, to simplify use of the transformer for new users, some of the more advanced parameters might be omitted from the Data Entry icon; they can be accessed only in the Transformer Controls window.

Number of header rows (; 1; 5;) row containing titles (,1; ,5;)

This parameter specifies the number of rows in the input data that are skipped and not used in calculating statistics. This parameter also specifies the row number containing column titles that are used as labels in the output. The two values should be separated by a comma. You can specify any number of header rows; the default value is 0. The title row number should be less than or equal to the number of header rows. If the header rows value is specified, the default title row is the last row of the header rows. If no header is specified, the default is no title row.

Report name (; AutoRegression Forecast;)

This parameter is a title for the Output 2 report, for example, *Autoregression Forecasting Tests*. If the `Report name` parameter is blank, there is no title in Output 2.

Date/Period column (a; b; c;)

This parameter specifies a column of dates to associate with the autoregression data. Any column can be specified; if this parameter is blank, a sequence number column is provided.

Data column (a; b; c;)

This parameter specifies the columns used to perform the autoregression analysis. At least one data column is required. For example, `b` performs autoregression analysis using column B.

Long term 'Seasonal' differencing (0; 12; 24;)

This parameter specifies the number of times that long-term or seasonal differencing is to be applied to the input data. Any positive whole number is valid. For example, 12 specifies that the input data should be seasonally differenced by subtracting the value for the observation 12 periods earlier from the value for the current period. The default value is 0 (no differencing).

Short term 'Trend' differencing (0; 1; 2;)

This parameter specifies the number of times that short-term or trend differencing is to be applied to the input data. Any positive whole number is valid. For example, 2 specifies that the input data should be trend differenced twice. The default value is 0 (no differencing).

Orders of autoregression to include in model (; 1,2; 1,3,5,12;)

This parameter specifies the lags that should be incorporated into the autoregression model. Lags are specified with whole numbers greater than 0 and should be separated by commas. For example, if you want the first and second lags in a model, you would type `1, 2`. There is no default value. Choosing appropriate values for this parameter requires examining the autocorrelation and partial autocorrelation coefficients.

Number of time lags for auto/partial correlations (; 12; 20; 30;)

This parameter specifies the largest time lag used in estimating the autocorrelation and partial autocorrelation coefficients. Any whole number is valid. The default value is the number of values (rows) in the input time series. For example, if autocorrelations for the first 24 lags are needed, you would type `24`.

Number of periods to forecast (; 12; 24;)

This parameter specifies the number of periods that are forecast. Any whole number is valid. The default is 0. For example, if forecasts for the next 12 months are needed, you would type `12`.

Any significance that a forecast carries diminishes when the number of future periods approaches one-half the number of values in the input data series.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- Results (Output 1)
- Statistics (Output 2)
- Coefficients (Output 3)
- Residuals (Output 4)
- Messages (Output 5)

When you click on any one of these choices, the data associated with that choice displays in the display area of the transformer.

Input Region Names

The AutoRegression transformer has only one input region, Input 1 (Data). You can modify the input name to reflect the name of the corresponding data in the display area. Input 1 consists of data read from a tool, such as a Spreadsheet or SQL Entry tool, or data copied directly into the transformer. Input 1 is treated as a series of columns. The AutoRegression transformer input data (data column) is a column containing numbers, in increasing chronological order. Only one data column is expected, but if additional data columns are specified, they are processed sequentially.

A column containing date information (date columns) can also be included. Dates must be valid date formats, or their numeric equivalent. If extra columns are present, they are ignored. The input data can have any number of header rows. Column titles are read from the specified row. If titles are not found, default titles will be provided. The transformer does not recognize dates that are changed so that they are no longer consistent with other dates.

For example, if you change the date Dec 2000 in the following example to Dec 1999, the transformer does not recognize the inconsistency:

```
Nov 2000
Dec 2000
Jan 2000
```

Whenever you have a variable in the data set with missing values, the entire observation or row is excluded from all calculations. This holds true even if the variables with the missing values are not used in the actual calculation or if the row or observation contains some partial data.

Consider the following example. A data set contains variables A and B. Variable A contains 10 observations and variable B contains 12. In this data set, the first ten observations would be used in all calculations and the last two would be excluded, even in calculations where only variable B is specified.

Whenever the transformer program encounters an observation with missing values, a message is displayed in the Important Message window. The message informs the user that the particular observation will be excluded from the calculation.

Output Region Names

The AutoRegression transformer generates five output regions that are not limited in size.

Output 1 (Results) is the autoregression result, which includes the following information:

- Variable (data column) name
- Date column, if the Date Column is specified
- Actual data values
- Predicted values
- Residual values
- Percent residual (represented as a proportion)
- Final result of differencing

Output 2 (Statistics) contains the following statistics computed during the autoregression process:

- Estimated autoregression coefficients for each lag
- Standard error values for each lag
- t-statistic values for each lag
- F-statistic values for each lag
- The probability of F for each lag
- Mean square error for the entire model
- Root mean square error for the entire model

Output 3 (Coefficients) contains autocorrelation and partial autocorrelation tables, which include the following information for each lag:

- Time lag
- Autocorrelation coefficient
- The t-value
- Upper confidence limit value
- Lower confidence limit value

- Partial autocorrelation coefficient
- Chi-square of the autocorrelations
- Probability of the observed chi-square

Output 4 (Residuals) contains an analysis of the autoregression residuals, including the following information for the first specified number of lags:

- Time lag
- Autocorrelation coefficient
- The t-value
- Upper confidence limit value
- Lower confidence limit value
- Partial autocorrelation coefficient
- Chi-square of the autocorrelations
- Probability of the observed chi-square

Output 5 (Messages) contains:

- Transformer run-time messages
- Warnings
- Error messages
- A timestamp for documentation purposes

Examples

The data analyzed in this example measures monthly housing starts from April 1997 through March 2000 in the Minneapolis and St. Paul metropolitan area. The goal is to create a model that predicts how many housing units will be started each month of 2001. The data contained in a Spreadsheet icon connected to Input 1 of the transformer is shown in the following example:

Date	Housing Permits
April, 1997	2,120
May, 1997	2,295
June, 1997	1,990
July, 1997	1,450
August, 1997	2,010
September, 1997	1,450
October, 1997	1,500
November, 1997	1,450

December, 1997	1,110
January, 1998	750
February, 1998	920
March, 1998	1,610
April, 1998	2,550
May, 1998	1,770
June, 1998	2,300
July, 1998	2,250
August, 1998	1,790
September, 1998	2,580
October, 1998	2,310
November, 1998	2,000
December, 1998	1,400
January, 1999	700
February, 1999	910
March, 1999	1,500
April, 1999	2,220
May, 1999	2,225
June, 1999	1,910
July, 1999	1,500
August, 1999	2,000
September, 1999	1,400
October, 1999	1,550
November, 1999	1,350
December, 1999	1,100
January, 2000	700
February, 2000	900
March, 2000	1,660

The first step in an autoregression analysis is to determine whether the series is stationary and relatively free from seasonality by inspecting the autocorrelations generated by the AutoRegression transformer.

To obtain that information, the AutoRegression transformer parameters are set as follows:

Number of header rows
1, 1

Report name
AutoRegression Example Case #1

Date/period column
A

Data column
B

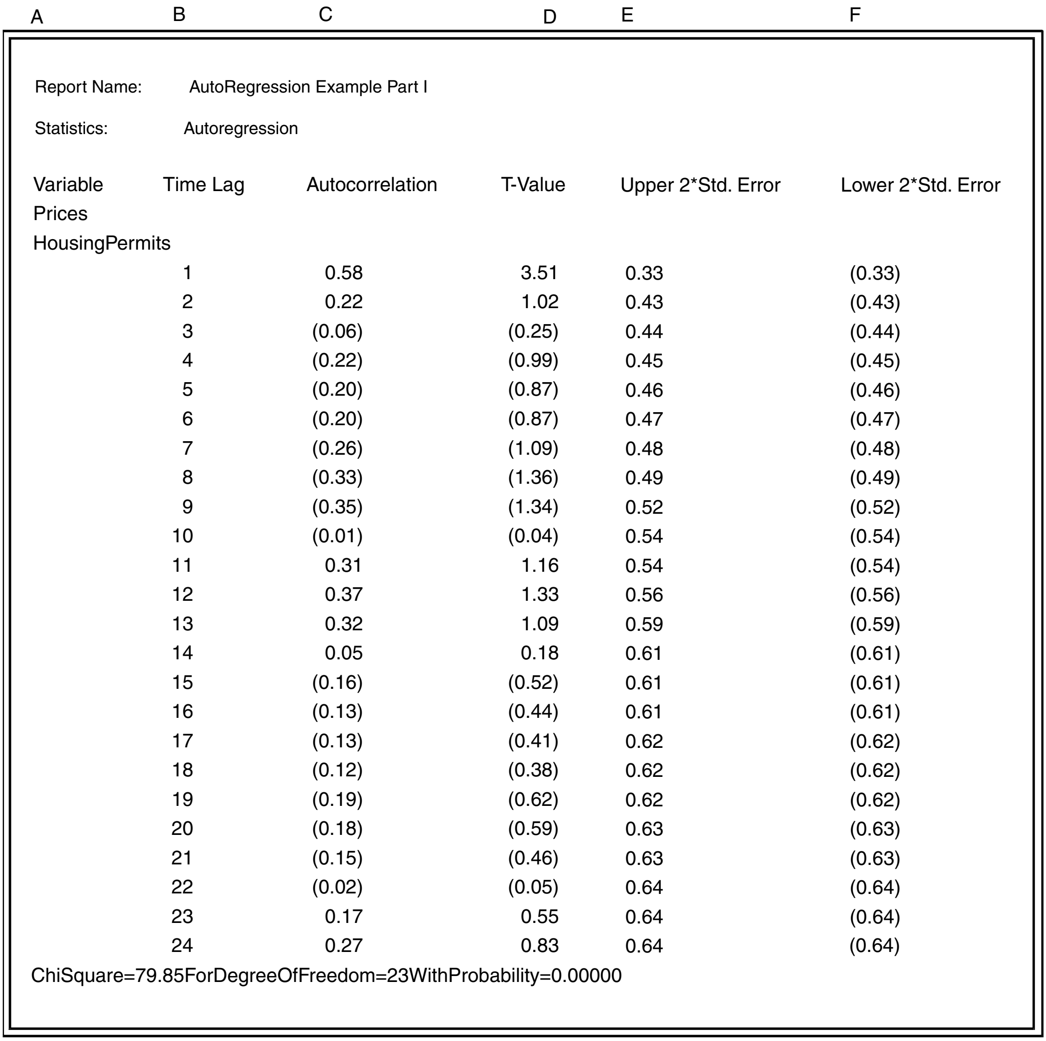
Long term 'Seasonal' differencing
0

Short term 'Trend' differencing
0

Orders of Autoregression to include in model
1,2

Number of time lags for auto/partial correlations
24

The parameters request no differencing and no terms or orders in the autoregression model. This is because the only output that is important at this stage is the autocorrelation. The `Number of time lags for auto/partial` `correlations` parameter is set to 24, so the transformer places the following autocorrelation information for the first 24 lags in Output 3:



Report Name: AutoRegression Example Part I

Statistics: Autoregression

Variable	Time Lag	Autocorrelation	T-Value	Upper 2*Std. Error	Lower 2*Std. Error
Prices					
HousingPermits					
	1	0.58	3.51	0.33	(0.33)
	2	0.22	1.02	0.43	(0.43)
	3	(0.06)	(0.25)	0.44	(0.44)
	4	(0.22)	(0.99)	0.45	(0.45)
	5	(0.20)	(0.87)	0.46	(0.46)
	6	(0.20)	(0.87)	0.47	(0.47)
	7	(0.26)	(1.09)	0.48	(0.48)
	8	(0.33)	(1.36)	0.49	(0.49)
	9	(0.35)	(1.34)	0.52	(0.52)
	10	(0.01)	(0.04)	0.54	(0.54)
	11	0.31	1.16	0.54	(0.54)
	12	0.37	1.33	0.56	(0.56)
	13	0.32	1.09	0.59	(0.59)
	14	0.05	0.18	0.61	(0.61)
	15	(0.16)	(0.52)	0.61	(0.61)
	16	(0.13)	(0.44)	0.61	(0.61)
	17	(0.13)	(0.41)	0.62	(0.62)
	18	(0.12)	(0.38)	0.62	(0.62)
	19	(0.19)	(0.62)	0.62	(0.62)
	20	(0.18)	(0.59)	0.63	(0.63)
	21	(0.15)	(0.46)	0.63	(0.63)
	22	(0.02)	(0.05)	0.64	(0.64)
	23	0.17	0.55	0.64	(0.64)
	24	0.27	0.83	0.64	(0.64)

ChiSquare=79.85ForDegreeOfFreedom=23WithProbability=0.00000

Figure 47 on page 374 shows a plot of the autocorrelations and partial autocorrelations.

The plot of autocorrelations shows no sign of a strong trend in this data. If the series contained a strong trend component, more than the first autocorrelation would be significant. In fact, if there was a strong trend component, the first three or more autocorrelations would be significant. A significant autocorrelation is one that is beyond the upper or lower confidence limits, which are labeled as *2*Std Error* in the plot. If an autocorrelation is beyond these limits, it indicates that there is only a 5% likelihood that the observed coefficient was found by chance.

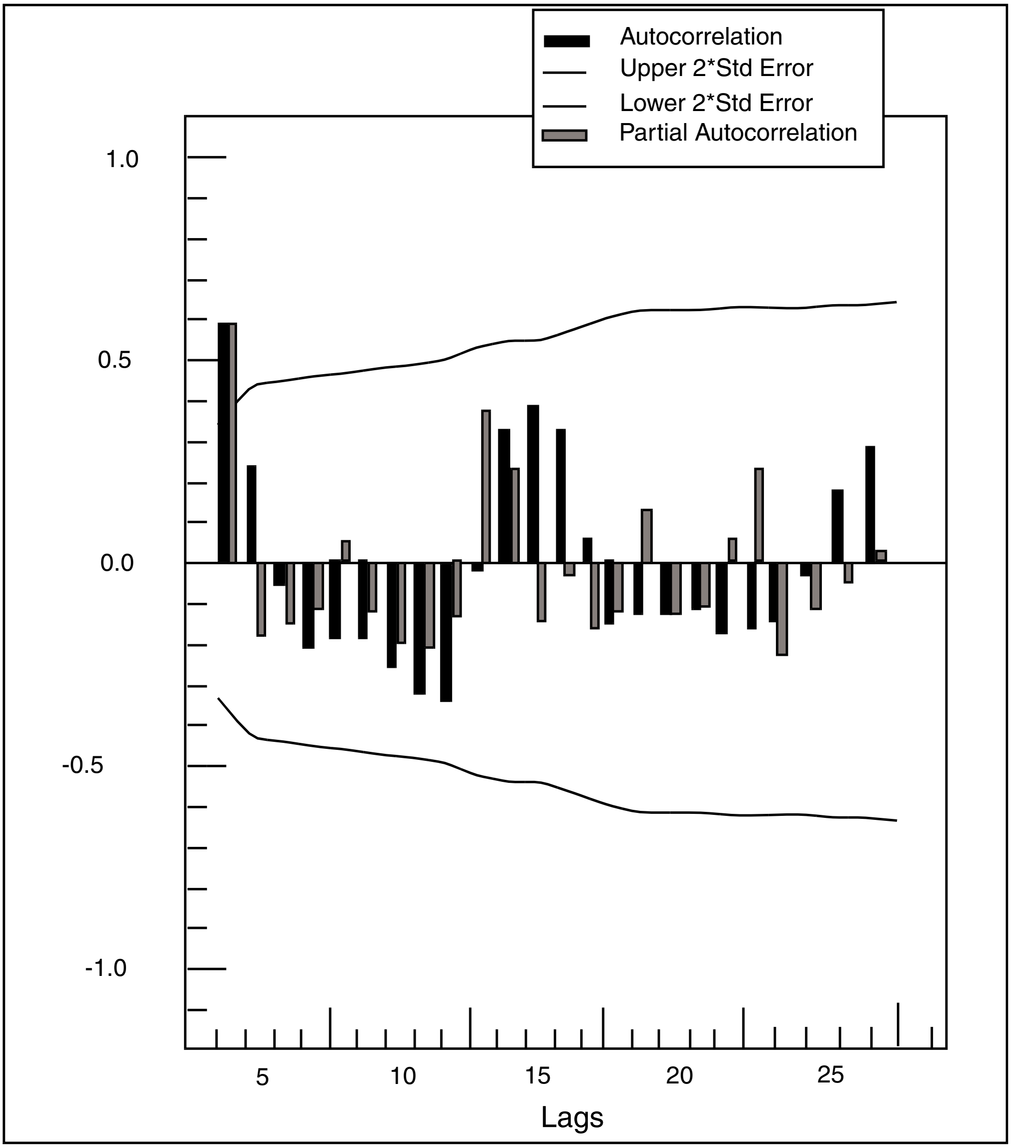


Figure 47. Autocorrelations and partial autocorrelations

The plot also shows no sign of a strong seasonal component in this series. If there was a seasonal component to the series, the autocorrelations for the lag that is equal to the length of the season would be significant. Because the data is collected monthly and there are 12 months in a season or year, the autocorrelation for the 12th lag would be significant. Because the autocorrelation for the 12th lag is 0.37 and is not greater than the upper confidence level or less than the lower confidence level, it is not significant.

When it is evident that there are no strong seasonal or trend components in the series, it is time to determine which lags should be included in the model. As mentioned earlier, all lags with significant positive autocorrelations should be included in the autoregression model. In this case (Autoregression transformer example Case #2), only one lag has a significant positive autocorrelation. Thus, only the first lag will be included in the autoregression model. The AutoRegression transformer parameters are set as follows:

Number of header rows
1,1

Report name
AutoRegression Example Case #2

Date/period column
A

Data column
B

Long term 'Seasonal' differencing
0

Short term 'Trend' differencing
0

Orders of Autoregression to include in model
1, 2

Number of time lags for auto/partial correlations
24

Number of periods to forecast
21

After the transformer runs, the following information is in Output 2, the statistics output:

A	B	C	D	E	F
Report Name:	AutoRegression Example Part II				
Statistics:	Autoregression				
Data Series:	Housing Permits				
Data Rows:	36				
Short Term (Trend):	0				
Long Term (Seasonal):	0				
Orders Of AR:	1				
MSE:	193,307.16				
RMSE:	439.67				
Source	Coefficient (Est.)	Std. Error	T-Value	F-Value	Probability
Constant	670.17	240.00	2.79	7.80	0.01
Phi 1	0.58	0.14	4.21	17.75	0.00
Mean	1,645.28	88.03	3.12	9.70	

The previous example provides evidence that this model has some strengths. First, the probability of the constant and the coefficient for the first lag (Phi 1) are near 0. This fact indicates that they are unlikely to occur by chance and are thus significant. However, the root mean square error is quite large. The value of 439.67 indicates that on average, this model either overpredicted or underpredicted housing starts by about 25% of the average level of housing starts (1,645).

For more evidence of the adequacy of the model, look at the predicted values, the actual values, and errors found in Output 1. That information and a plot of the predicted, actual, and residual values is shown in the following example:

A	B	C	D	E
PERIODS	Housing Permits	PREDICTED	PREDICTED ERROR	% ERROR
April, 1996	2,120.00			
May, 1996	2,295.00	1,909.47	385.53	16.80
June, 1996	1,990.00	2,011.77	-21.77	-1.09
July, 1996	1,450.00	1,833.48	-383.48	-26.45
August, 1996	2,010.00	1,517.81	492.19	24.49
September, 1996	1,450.00	1,845.17	-395.17	-27.25
October, 1996	1,500.00	1,517.81	-17.81	-1.19

November, 1996	1,450.00	1,547.03	-97.03	-6.69
December, 1996	1,110.00	1,517.81	-407.81	-36.74
January, 1997	750.00	1,319.05	-569.05	-75.87
February, 1997	920.00	1,108.60	-188.60	-20.50
March, 1997	1,610.00	1,207.98	402.02	24.97
April, 1997	2,550.00	1,611.34	938.66	36.81
May, 1997	1,770.00	2,160.84	-390.84	-22.08
June, 1997	2,300.00	1,704.87	595.13	25.88
July, 1997	2,250.00	2,014.69	235.31	10.46
August, 1997	1,790.00	1,985.47	-195.47	-10.92
September, 1997	2,580.00	1,716.56	863.44	33.47
October, 1997	2,310.00	2,178.38	131.62	5.70
November, 1997	2,000.00	2,020.54	-20.54	-1.03
December, 1997	1,400.00	1,839.32	-439.32	-31.38
January, 1998	700.00	1,488.58	-788.58	-112.65
February, 1998	910.00	1,079.37	-169.37	-18.61
March, 1998	1,500.00	1,202.14	297.86	19.86
April, 1998	2,220.00	1,547.03	672.97	30.31
May, 1998	2,225.00	1,967.93	257.07	11.55
June, 1998	1,910.00	1,970.85	-60.85	-3.19
July, 1998	1,500.00	1,786.71	-286.71	-19.11
August, 1998	2,000.00	1,547.03	452.97	22.65
September, 1998	1,400.00	1,839.32	-439.32	-31.38
October, 1998	1,550.00	1,488.58	61.42	3.96
November, 1998	1,350.00	1,576.26	-226.26	-16.76
December, 1998	1,100.00	1,459.35	-359.35	-32.67
January, 1999	700.00	1,313.20	-613.20	-87.60
February, 1999	900.00	1,079.37	-179.37	-19.93
March, 1999	1,660.00	1,196.29	463.71	27.93
April, 1999		1,640.57		
May, 1999		1,196.29		
June, 1999		1,640.57		

July, 1999	1,629.21
August, 1999	1,369.49
September, 1999	1,629.21
October, 1999	1,622.57
November, 1999	1,470.74
December, 1999	1,622.57
January, 2000	1,618.68
February, 2000	1,529.93
March, 2000	1,618.68
April, 2000	1,616.41
May, 2000	1,564.53
June, 2000	1,616.41
July, 2000	1,615.09
August, 2000	1,584.76
September, 2000	1,615.09
October, 2000	1,614.31
November, 2000	1,596.58
December, 2000	1,614.31

Figure 48 on page 379 shows that the model does not fit the data very well. In fact, in one case, the model's predictions are off by over 100% of the actual housing starts.

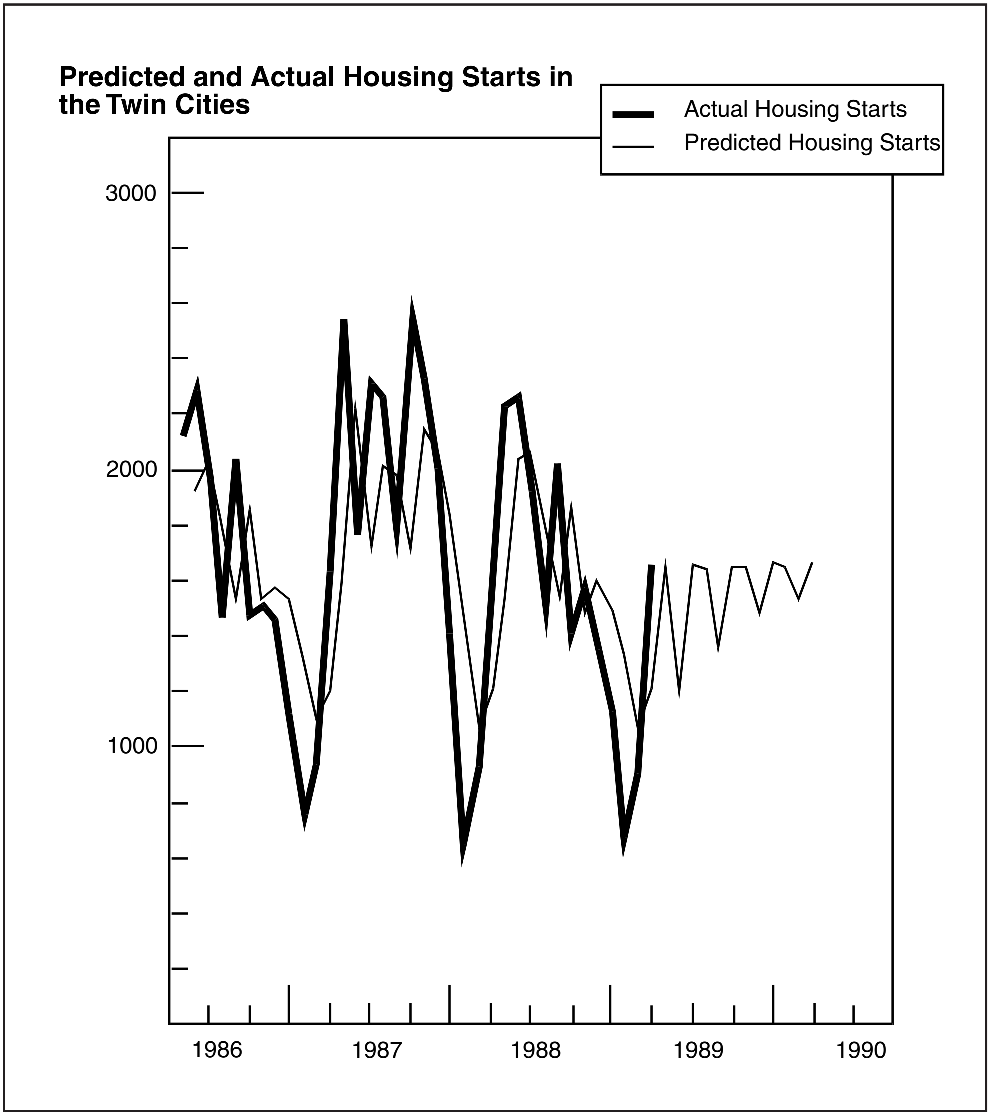


Figure 48. Predicted, actual, and residual values

One way to improve the model would be to try adding another lag to it. Although none of the other lags was significant, the lag with the largest positive autocorrelation could be added. If that model had a lower root mean square error, it could be used to predict housing starts.

Differencing Method

Differencing is a method for removing seasonality and trend effects from a series before autocorrelation or autoregression statistics are computed. A differenced series is the result of subtracting lag values from the values of the original series.

For an autoregression or autoregressive integrated moving average (ARIMA) model to produce valid results, the input series must be stationary and relatively free of seasonal effects. (ARIMA models provide a method of describing both stationary and nonstationary time series.) A stationary series has values that fall near a constant mean; it shows no growth or decline over time. A nonstationary series does not have constant mean. The nonstationary component is often referred to as the trend.

A seasonal effect is a cyclical increase or decrease in the series. For example, the sales of suntan oil are much higher in the early summer than during the rest of the year; thus, suntan oil sales have an annual seasonal component. Each year, they go through a period of growth and decline that is quite stable.

A plot of the autocorrelation coefficients versus the time lag number is used to test the presence of trend or seasonal effects. If the plot of autocorrelation coefficients includes a few high values for the first one to three lags with the rest falling to 0, the data set has little or no trend (stationary mean). Alternatively, if the autocorrelations for four or more of the first lags are significantly different than 0 and the plot forms a diagonal line, a trend exists (nonstationary mean).

In a stationary series, plots of autocorrelation coefficients provide a good indication of seasonal effects. If the autocorrelations are arranged in wave patterns from low to high values with some values being significantly different from 0, it indicates that seasonality exists in the series. The length of seasonality is equal to the number of lags between crests of these waves.

Strong trend and seasonal effects can obscure autoregressive relationships in a series. When this occurs, it is best to reduce the trend and seasonal effects by differencing a series at the beginning of the autoregression process. The AutoRegression transformer allows you to remove these components with two types of differencing: short-term (trend) differencing and long-term (seasonal) differencing.

Short-term differencing consists of subtracting adjacent values of the data, using their difference as a new time-series. Trend differencing is an iterative process; a series is differenced and the results of the differencing, in turn, can be differenced. Each successive difference is computed on the previous difference. Thus, the third difference would be computed by differencing the result of the second difference. The AutoRegression transformer allows you to specify any degree of short-term differencing.

The definitions described in Table 59 apply to the equations used in this section:

Table 59. Differencing formula symbol definitions

Symbol	Definition
i	A specific period in a series
N	Number of periods in the input series
W	Result of second differencing
U	Result of first differencing
X	Original input data

The following formulas illustrate how a two-degree short-term difference is computed. The first short-term difference is computed from the original data:

$$U_i = X_i - X_{i-1}$$

for $i = 1$, $u_i = 0$

The second short-term difference is computed from the result of the first difference:

$$W_i = U_i - U_{i-1}$$

for $i < 3$, $w_i = 0$

The short-term differencing algorithm produces a series that has one fewer element for each degree of differencing.

Long-term differencing works very much like short-term differencing. However, in long-term differencing, the value for a period one season ago is subtracted from the current period's value. For example, to compute a difference when the length of seasonality is one year and the current period is January 2000, the transformer would subtract the January 1999 value from the January 2000 value. With long-term differencing, the data set loses as many data points as are in a season. For example, if the original series contains 48 monthly data points and the length of a season is 12, long-term differencing produces a data set with 36 data points.

The value of long-term differencing is set to the number of periods in a season. Thus, if the data being analyzed is measured on a monthly basis and the seasonal pattern occurs once a year, long-term differencing would be set to 12 because there are 12 months in a season.

If both long-term and short-term differencing is specified, the transformer completes the long-term differencing first and then applies the short-term differencing. The following example illustrates how the transformer would apply a long-term differencing of six and a short-term differencing of two. This type of differencing would be useful for stabilizing a bimonthly series that has strong curved trend and annual seasonal components.

Original Series	Long-Term Differencing	1st Short-Term Differencing	2nd Short-Term Differencing
30			
26			
22			
18			
14			
10			
17	-13		
16	-10	3	
11	-11	-1	-4
7	-11	0	1
1	-13	-2	-2
0	-10	3	5

AutoRegression Transformer Formulas

The definitions described in Table 60 on page 382 apply to the equations used in this section.

Table 60. AutoRegression formula symbol definitions

Symbol	**Definitions**
a	Intercept value of the autoregression equation
b_k	Optimized autoregression coefficient at time lag k
df	Degrees of freedom for the Ljung and Box chi-square
E_i	Residual difference between actual and predicted values
k	A particular time lag
K	Maximum time lag value (specified as a parameter)
MSE	Mean squared error of the predicted series
N	Number of observations in the input data set
pr_k	Partial autocorrelation coefficient at time lag k
p	Degree of autoregression

Table 60. AutoRegression formula symbol definitions

Symbol	Definitions
r_k	Autocorrelation coefficient at time lag k
RMSE	Root mean squared error of the predicted series
x_i	Difference between any value and its mean
$\bar{x}$	Mean value of X
X	Data set predicted by autoregression
c_2	Ljung and Box chi-square for the autocorrelation coefficients
z_i	A particular value in the data set Z
$\bar{z}$	Mean value of Z
Z	A data set (column in the input data)

The estimated autocorrelation coefficient r_k is:

$$r_k = \frac{\sum_{i=1}^{N-k} (z_i - \bar{z}) \times (z_{i+k} - \bar{z})}{\sum_{i=1}^{N} (z_i - \bar{z})^2}$$

The estimated partial autocorrelation coefficients are computed in a *k x k* matrix. The matrix is set up using the following three formulas:

$$pr_{1,1} = r_1$$

$$pr_{a,b} = pr_{a-1,b} - pr_{a,a} \times pr_{a-1,a-b}$$

for *a = k; b= k where a,b > 1*

$$pr_{m,m} = \frac{r_m - \sum_{n=1}^{m-1} (pr_{m-1,n} \times r_{m-n})}{1 - \sum_{n=1}^{m-1} (pr_{m-1,n} \times r_n)}$$

for `m = 2, k`

The estimated partial autocorrelation coefficients pr_k are taken from the main diagonal of the *K x K* matrix. Thus,

$$pr_k = pr_{a,b}$$

for `a = k; b = k`

The standard error of the sampling distribution of r_k is the square root of its estimated variance. This standard error, designated s(r_k), is calculated as follows:

$$s(r_k) = \sqrt{\frac{1 + 2 \times \sum_{j=1}^{k-1} r_j^2}{N}}$$

The autocorrelation null hypothesis test ($H_o{:}r^o{}_k = 0$) is the basis for the t-statistic:

$$t_k = \frac{r_k}{s(r_k)}$$

The F-statistic is approximated for each time lag by the equation:

$$F_k = t_k^2$$

The Ljung and Box chi-square statistic for autocorrelations is:

$$\chi^2 = N * (N+2) \times \sum_{j=1}^{K} \frac{r_j^2}{N-j}$$

The degrees of freedom for the Ljung and Box chi-square statistic is:

$$df = K - 1$$

The predicted time series is computed from the autoregression equation:

$$X_t = a + b_1 * X_{t-1} + b_2 * X_{t-2} + \ldots + b_p * X_{t-p}$$

The mean square error of the predicted series is:

$$MSE = \frac{\sum_{i=p}^{N} E_i^2}{N-p}$$

The root mean square error of the predicted series is:

$$RMSE = \sqrt{MSE}$$

Orders of Autoregression

The order of an autoregression model specifies the lags that are incorporated into a model. In general, increasing the number of lags in a model increases the explanatory power of the model. However, adding lags to an autoregression model also increases the amount of error in a model and might result in a model that is difficult to interpret. Also, each additional lag in a model increases the run time of the transformer.

In general, the `Orders of autoregression to include in the model` parameter should list all the lags that have significant autocorrelations. A significant autocorrelation has a value that is lower than the lower 2x standard error or greater than the upper 2x standard error. These limits are reported with the autocorrelations in Output 3.

Correlation

The Correlation transformer is designed to calculate the following correlation-related statistics:

- Correlation coefficient *r*
- Covariance
- Significance level

The correlation coefficient *r* represents the magnitude of the relationship between two variables or columns of data. The covariance value represents a measure of association between variables using Pearson's method. The significance level is the two-tailed critical value for the acceptance or rejection of the null hypothesis (r = 0). The Correlation transformer can also compute correlation on more than two data variables by applying the correlation algorithm to all combinations of variables.

The correlation coefficient *r*, which ranges between plus and minus one, is a measure of the mutual relationship between two variables. A value near 0 indicates little correlation between the variables, a value near +1 or -1 indicates a great deal of correlation. When two variables have a positive correlation coefficient, an increase in one variable indicates an increase in the second variable. A correlation coefficient of less than 0 indicates a negative correlation; that is, when one variable shows an increase in value, the other variable shows a decrease.

From a formal mathematical viewpoint, consider two variables U and W. If r = 1, then U and W are perfectly positively correlated. The possible values of U and W all lie on a straight line with a positive slope in the (U, W) plane.

If r=0, then the variables are not correlated; that is, they are linearly unassociated with each other. This does not mean, however, that U and W are statistically independent. If r = -1, then U and W are perfectly negatively correlated, and the possible values of U and W lie on a straight line in the (U, W) plane with negative slope.

Statistical testing generally consists of formulating a hypothesis about the data and then testing that hypothesis. In correlation, the null hypothesis is defined as the case that the data is not correlated. The alternative hypothesis would then be that the data is correlated. The significance level, based on the t-statistic, specifies the probability that the null hypothesis is true. For example, if a given correlation has a significance level of .20, there is a 20% chance that null hypothesis is true. In other words, there is an 80% chance that the alternative hypothesis is true and the data is correlated.

Parameters

All of the parameters described in this section are displayed in the Transformer Controls window. Most of these parameters are also displayed in the Data Entry icon contained in the transformer's demonstration capsule application. However, to simplify use of the transformer for new users, some of the more advanced parameters might be omitted from the Data Entry icon; they can be accessed only in the Transformer Controls window.

Number of header rows (1; 5;) row containing titles (; ,1; ,5;)

This parameter specifies the number of rows in the input data that are skipped and not used in calculating statistics. This parameter also specifies the row number containing column titles that are used to label the output. The two values should be separated by a comma. You can specify any number of header rows; the default value is 0. The title row number should be less than or equal to the number of header rows. If the header rows value is specified, the default title row is the last row of the header rows. If no header is specified, then the default is no title row.

Report name (Correlation Test;)

This parameter is a title for the output report, for example, *Correlation statistic tests*. If this parameter is blank, then there is no title in Output 1.

Data columns (a,b; a,b,c;)

This parameter specifies the columns used to compute the correlation statistics. At least two Data columns are required, but any number can be specified. For example, a,b,c computes correlation statistics for the A vs. B, A vs. C, and B vs. C pairs of columns.

Correlation statistics (All; Cor; Cov; Sig)
This parameter specifies the correlation statistics to be computed. Allowable values are all, cor (correlation), cov (covariance), and sig (significance level). The default value is all.

Group column as (; a, 5,10,15; b,male,female;)
This parameter segregates the input data into a set of user-specified groups or ranges. If this parameter is blank, then one group is created, containing all of the input data.

Create 'Plot' output? (; y; n)
This parameter determines whether results are sent to Output 2. Valid responses are Yes, which indicates that the results should be sent to Output 2 and No, which indicates that no results should be sent to Output 2. These responses can be abbreviated to y and n. No is the default value.

Region Controls

The Display Data For field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- Results (Output 1)
- Plot (Output 2)
- Messages (Output 3)

When you click on any one of these choices, the data associated with that choice displays in the display area of the transformer.

Input Region Names

The Correlation transformer has only one input region, Input 1 (Data). You can modify the input name to reflect the name of the corresponding data in the display area. Input 1 consists of data read from a tool, such as a Spreadsheet, Query, or SQL Entry tool, or copied directly into the transformer. Input 1 is treated as a series of columns that can contain statistical data (Data Columns) or grouping information (Group Column as). If extra columns are present, they are ignored. The input data can have any number of header rows. Column titles are read from the specified row; however, if titles are not found, default titles will be provided.

Output Region Names

The Correlation transformer generates three output regions that are not limited in size.

Output 1 (Results) consists of a series of tables:

- A correlation table, if the correlation statistic option is specified
- A covariance table, if the covariance statistic option is specified
- A significance level table, if the significance level option is specified

Output 2 (Plot) contains the same information contained in Output 1. However, Output 2 is formatted for easy use with the Plot tool.

Output 3 (Messages) contains:

- Transformer run-time messages
- Warnings
- Error messages
- A timestamp for documentation purposes

Examples

This example uses four views of the same data. Assume the following data is presented to Input 1 of the Correlation transformer in each of the following cases:

Area	Characteristic B	Characteristic C
2	57.00	89.00
1	120.00	30.00
1	101.00	82.00
3	137.00	50.00
3	119.00	39.00
2	117.00	22.00
2	104.00	57.00
2	73.00	32.00
1	53.00	96.00
1	68.00	31.00
3	118.00	88.00

Case #1: The Transformer Controls window parameters are set as follows:

Number of header rows
1, 1

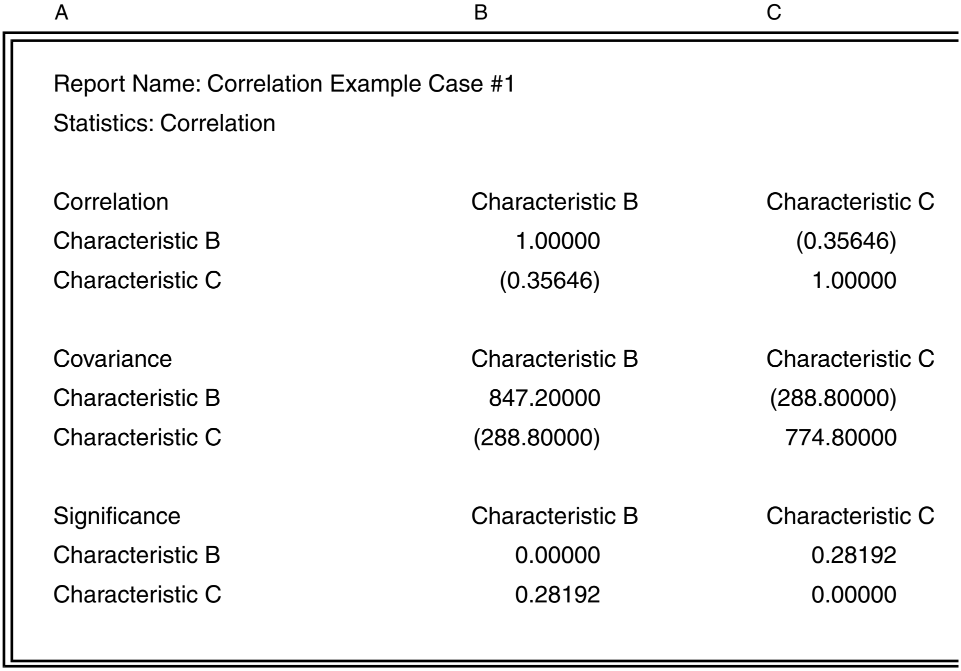
Report name
Correlation Example Case #1

Data column
b:c

Correlation statistics
All

When a parameter is blank, the default value is used by the transformer.

When the transformer has run, the following report is displayed in Output 1:



```
A                                             B                               C

Report Name: Correlation Example Case #1
Statistics: Correlation

Correlation                            Characteristic B        Characteristic C
Characteristic B                           1.00000                 (0.35646)
Characteristic C                          (0.35646)                  1.00000

Covariance                             Characteristic B        Characteristic C
Characteristic B                         847.20000               (288.80000)
Characteristic C                        (288.80000)              774.80000

Significance                           Characteristic B        Characteristic C
Characteristic B                           0.00000                 0.28192
Characteristic C                           0.28192                 0.00000
```

The correlation coefficient of Characteristic B and Characteristic C is approximately -0.36, which does not indicate a high degree of correlation. The Significance table shows that there is a 28% probability that the null hypothesis (the data samples are not correlated) is true. In this case, you would not want to propose that Characteristic B and Characteristic C have a correlation of -0.36 unless you can afford to be wrong 28% of the time. The relatively high value of the significance level indicates that the estimated value of *r* is not very reliable. This situation is probably due to the small number of data points used in this example.

Case #2: This view of the data takes advantage of the grouping features of the Correlation transformer. Column a is selected as the group column. Because no grouping criteria are specified, the Correlation transformer locates every group in column A and computes statistics.

The Transformer Controls window parameters for Case #2 are set as follows:

Number of header rows
1, 1

Report name
Correlation Example Case #2

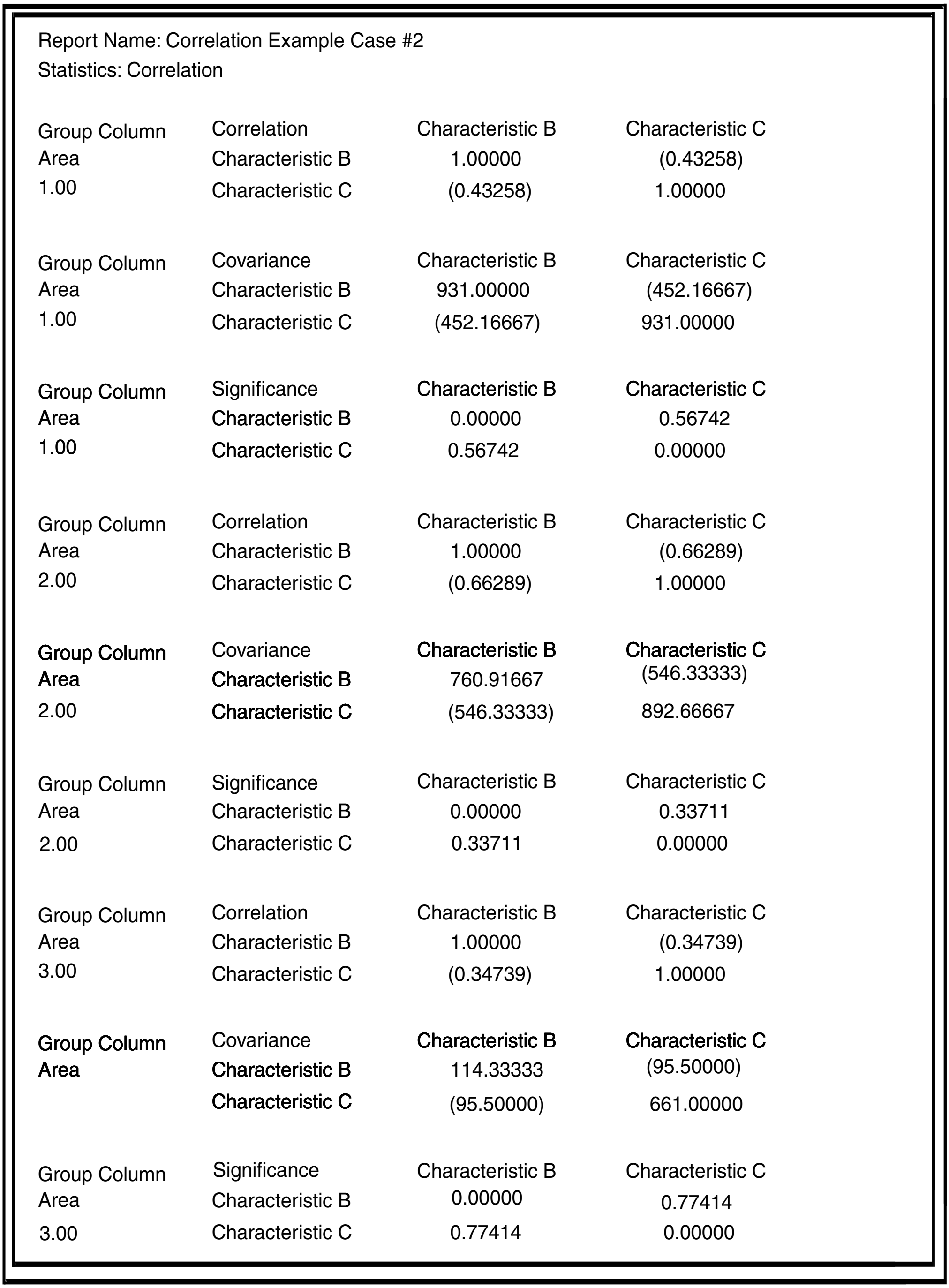
Correlation

Data column
b:c

Correlation statistics
All

Group column as
a

When the transformer run finishes, the following report is displayed in Output 1:



```
Report Name: Correlation Example Case #2
Statistics: Correlation

Group Column    Correlation         Characteristic B    Characteristic C
Area            Characteristic B       1.00000             (0.43258)
1.00            Characteristic C      (0.43258)             1.00000


Group Column    Covariance          Characteristic B    Characteristic C
Area            Characteristic B     931.00000            (452.16667)
1.00            Characteristic C    (452.16667)           931.00000


Group Column    Significance        Characteristic B    Characteristic C
Area            Characteristic B       0.00000              0.56742
1.00            Characteristic C       0.56742              0.00000


Group Column    Correlation         Characteristic B    Characteristic C
Area            Characteristic B       1.00000             (0.66289)
2.00            Characteristic C      (0.66289)             1.00000


Group Column    Covariance          Characteristic B    Characteristic C
Area            Characteristic B     760.91667            (546.33333)
2.00            Characteristic C    (546.33333)           892.66667


Group Column    Significance        Characteristic B    Characteristic C
Area            Characteristic B       0.00000              0.33711
2.00            Characteristic C       0.33711              0.00000


Group Column    Correlation         Characteristic B    Characteristic C
Area            Characteristic B       1.00000             (0.34739)
3.00            Characteristic C      (0.34739)             1.00000


Group Column    Covariance          Characteristic B    Characteristic C
Area            Characteristic B     114.33333             (95.50000)
                Characteristic C     (95.50000)           661.00000


Group Column    Significance        Characteristic B    Characteristic C
Area            Characteristic B       0.00000              0.77414
3.00            Characteristic C       0.77414              0.00000
```

Three groups were located in column A: Area 1.00, Area 2.00, and Area 3.00. This view of the data shows that Area 2.00 has a greater correlation coefficient than Areas 1.00 or 3.00. However, the significance value for that correlation is 0.3371, which indicates that there is a 33% probability that the correlation could have occurred due to chance. Thus, that correlation should not be taken too seriously. In fact, none of the significance values are low enough to indicate a significant correlation at the 0.10 level. The reason for this is probably due to the small number of observations in each of the groups.

Case #3: This case allows examination of individual groups. If only a particular part of the data is important, then you can use grouping criteria to limit the amount of information computed by the Correlation transformer. This example demonstrates how to focus on Area 3.00 by setting the grouping criteria to 3, followed by the group type specifier only.

The Transformer Controls window parameters for Case #3 are set as follows:

Number of header rows
1, 1

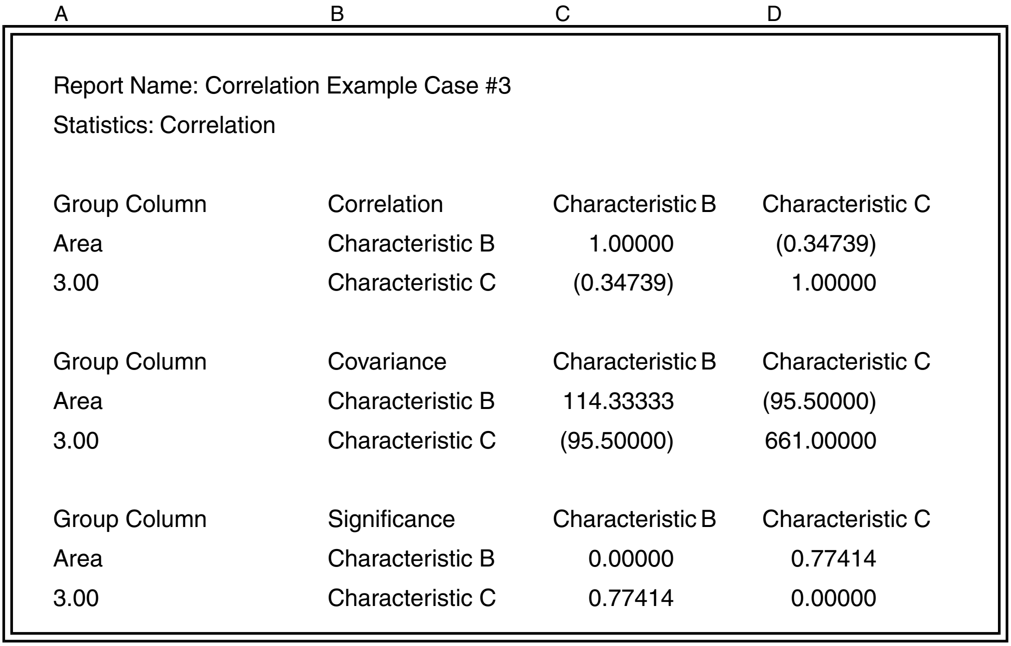
Report name
CorreCase #3

Data column
b:c

Correlation statistics
All

Group column as
a, 3, only

When the transformer has finished, the following report is displayed in Output 1. This report contains information that is also found in the Case #2, but the results are limited to the Area 3.00 data.



```
A                 B                 C                D

Report Name: Correlation Example Case #3
Statistics: Correlation

Group Column      Correlation       Characteristic B   Characteristic C
Area              Characteristic B      1.00000          (0.34739)
3.00              Characteristic C     (0.34739)          1.00000

Group Column      Covariance        Characteristic B   Characteristic C
Area              Characteristic B    114.33333         (95.50000)
3.00              Characteristic C    (95.50000)        661.00000

Group Column      Significance      Characteristic B   Characteristic C
Area              Characteristic B      0.00000           0.77414
3.00              Characteristic C      0.77414           0.00000
```

Case #4: An alternate application of the grouping features summarizes data using ranges. In this example, the grouping criterion has been set to 1.5, and the grouping type specifier has been omitted.

The Transformer Controls window parameters for Case #4 are set as follows:

Number of header rows
1, 1

Report name
Correlation Example Case #4

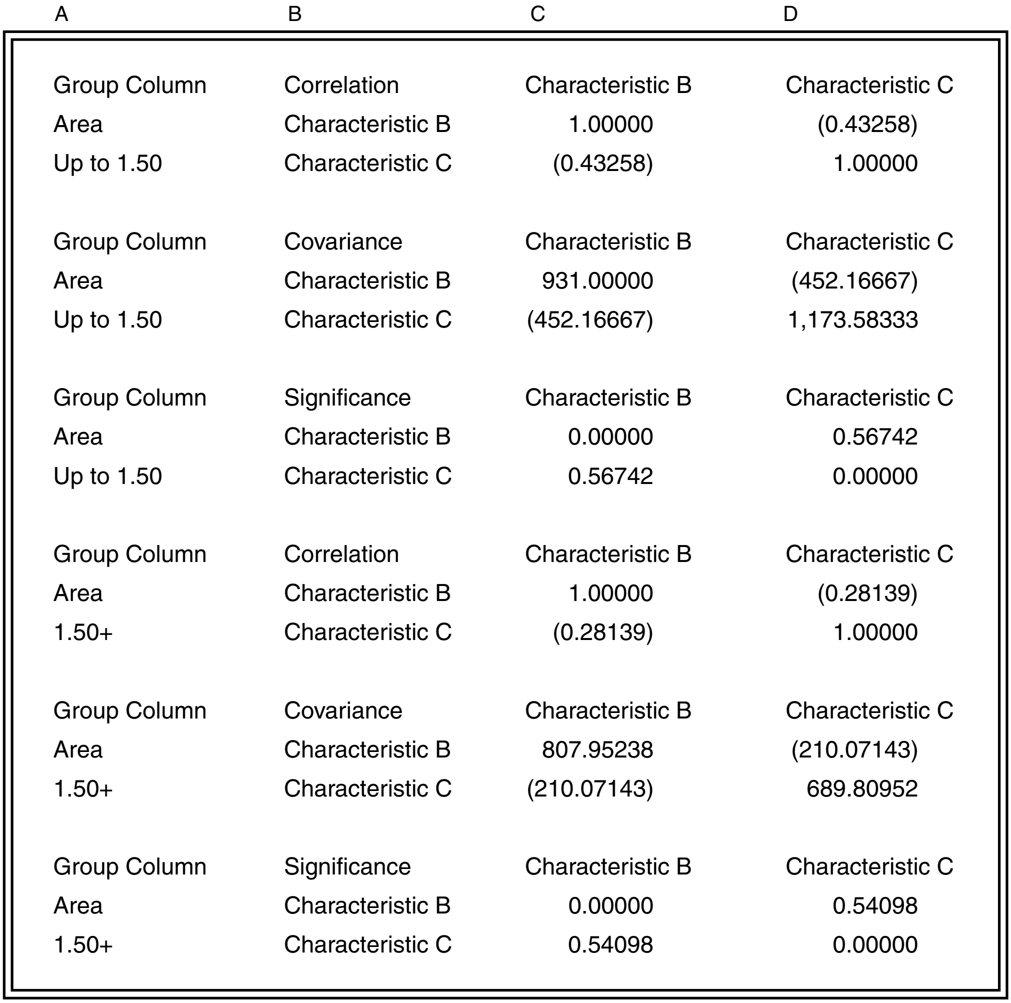
Data column
b:c

Correlation statistics
All

Group column as
a, 1.5

Correlation

When the transformer run finishes, the following report is displayed in Output 1:



```
A                   B                   C                   D

Group Column        Correlation         Characteristic B    Characteristic C
Area                Characteristic B         1.00000            (0.43258)
Up to 1.50          Characteristic C        (0.43258)            1.00000

Group Column        Covariance          Characteristic B    Characteristic C
Area                Characteristic B       931.00000          (452.16667)
Up to 1.50          Characteristic C      (452.16667)         1,173.58333

Group Column        Significance        Characteristic B    Characteristic C
Area                Characteristic B         0.00000             0.56742
Up to 1.50          Characteristic C         0.56742             0.00000

Group Column        Correlation         Characteristic B    Characteristic C
Area                Characteristic B         1.00000            (0.28139)
1.50+               Characteristic C        (0.28139)            1.00000

Group Column        Covariance          Characteristic B    Characteristic C
Area                Characteristic B       807.95238          (210.07143)
1.50+               Characteristic C      (210.07143)          689.80952

Group Column        Significance        Characteristic B    Characteristic C
Area                Characteristic B         0.00000             0.54098
1.50+               Characteristic C         0.54098             0.00000
```

This report contains two ranges: *Up to 1.50*, which includes values in Area 1.00, and *1.50+*, which includes the remaining areas, 2.00 and 3.00.

Correlation Transformer Formulas

The definitions described in Table 61 apply to the equations used in this section.

Table 61. Correlation transformer symbol definitions

Symbol	**Definition**
$C_{U,W}$	Covariance of the data columns U and W

Table 61. Correlation transformer symbol definitions

Symbol	Definition
N	Number of observations in each data column
S_U^2	Variance of the data column U
S_W^2	Variance of the data column W
U	A data column with N observations
$\overline{U}$	Average of all elements in the data column U
U_i	Any particular member of the data column U
W	A second data column with N observations
$\overline{W}$	Average of all elements in the data column W
W_i	Any particular member of the data column W

Correlation

For the data columns U and W:

$$\bar{U} = \frac{\sum_{i=1}^{N} U_i}{N}$$

$$\bar{W} = \frac{\sum_{i=1}^{N} W_i}{N}$$

$$S_U^2 = \sum_{i=1}^{N} \frac{(U_i - \bar{U})^2}{N-1}$$

$$S_W^2 = \sum_{i=1}^{N} \frac{(W_i - \bar{W})^2}{N-1}$$

$$C_{U,W} = \frac{\sum_{i=1}^{N} (U_i - \bar{U}) \times (W_i - \bar{W})}{N-1}$$

For data columns U and W, the correlation coefficient r is calculated as follows:

$$r = \frac{C_{U,W}}{\sqrt{S_W^2} \times \sqrt{S_U^2}}$$

The significance level is based upon the t-test statistic, which is calculated as follows:

$$t = r \times \sqrt{\frac{N-2}{1-r^2}}$$

The value *t* is actually the t-statistic, which must be converted to a probability factor using the *t* distribution. The Correlation transformer automatically performs this conversion and displays the significance level as a probability factor.

Specifying Data Columns

To specify the input columns containing data, enter a list, a range of columns, or both, using either the letters or the numbers associated with the columns. For example, if the input data is in the first three columns of a spreadsheet, the Data Columns list specification would be `a,b,c` or `1,2,3`. A list of columns is simply a series of column letters or numbers separated by commas. The columns specified do not have to be contiguous. For example, if the Data Columns specification is `a,c`, the transformer gathers the data from the first and third columns of the input region.

This parameter also accepts ranges of columns. A range of columns consists of the number or letter associated with first data column, a colon, and the letter or number associated with the last data column. For example, if the transformer should use the first five columns of data, the Data Columns specification would be `1:5` or `a:e`.

The `Data Columns` parameter also accepts a combination of lists and ranges. For example, if the input data occurs in the first, second, and fourth through sixth columns, the parameter specification would be `a,b,d:f` or `1,2,4:6`.

Specifying Grouping Features

A grouping expression consists of three parts: a group column, an optional list of grouping criteria, and an optional group type specifier. Each group is treated as a distinct set of data, and a set of correlation tables is generated for every requested group.

The group column is a single column that contains information determining the group to which a particular data element belongs. For example, if the first column of the input data contains grouping information, the entry for column A would be `a`.

A list of grouping criteria can follow the column name. Grouping criteria specify the groups to be created. The criteria can be text values, numeric values, or dates. Each value must be separated by a comma. If grouping criteria are not present, the Correlation transformer creates a group for each unique value in the grouping column.

The group type specifier controls whether the grouping criteria are treated as members of a group or as limits of a range. If the type specifier *only* is present, a group is created for each item in the grouping criteria list. Only values that exactly match a particular grouping criterion are added to the corresponding group. If the type specifier is not present, the grouping criteria are treated as the end points of a series of ranges. Any value greater than the first end point and less than or equal to the second end point is treated as part of that particular range.

An example of each of the four possible variations of a grouping expression follows:

<no expression>
One group is created containing all of the input data.

a A group is created for each unique value in column A.

a, 10, 20
Three ranges are created: all values up to and including 10, all values above 10 but less than or equal to 20, and all values above 20.

a, 10, 20, only
Two groups are created: all values where column A is 10 and all values where column A is 20.

Elementary

The Elementary transformer calculates the following descriptive statistics:

- Count
- Sum
- Mean
- Variance
- Standard deviation
- Standard error
- Minimum
- Maximum
- Range
- Coefficient of variation

There are several advantages to using the Elementary transformer, rather than using a Spreadsheet icon or other method, to compute the statistics listed.

The Elementary transformer condenses data into user-defined groups for analysis or for subsequent input to other transformers. Up to two levels of groupings can be specified. For example, input data that consists of product, market, period, and sales data can be grouped by product and market to yield the sum of the sales for all periods.

When you connect a Query tool to the input of this transformer, you can generate statistics for virtually an unlimited number of rows of data, because the Query tool can retrieve data one row at a time and feed it to the transformer. The statistics previously listed never require more than one input row to be stored in the workstation; therefore, statistics can be computed for large amounts of data. The transformer can perform these calculations quickly; thus, there is a speed advantage over using a spreadsheet as well.

The Elementary transformer does not require the input data to be sorted, saving a significant amount of time compared to other data grouping methods. The Elementary transformer can group data in any input format used in Meta5, whether it is date, text, numeric, or even Boolean data (true/false answers).

Parameters

All of the parameters described in this section are displayed in the Transformer Controls window. Most of these parameters are also displayed in the Data Entry icon contained in the transformer's demonstration Capsule icon. However, to simplify use of the transformer for new users, some of the more advanced parameters might be omitted from the Data Entry icon; they can be accessed only in the Transformer Controls window.

The Elementary transformer has many parameters that might seem complex; however, many of them are of the "set and forget" type: when set up, only a few parameters need to be adjusted.

Dates can be entered for certain parameters. All Meta5 date formats are directly supported. To enter a date, enclose it in double quotation marks. For example, March 2, 2000, would be entered as `"March 2, 2000"`. Quotation marks tell the transformer to ignore the comma in the date, rather than breaking the date into two pieces.

The Elementary transformer has four groups of parameters:

- User
- Primary Grouping
- Secondary Grouping
- Expert

User Parameters

Number of header rows (1; 2;)

This parameter allows a header to precede the data in Input 1. The first row of the header, if it exists, is used to label the results. If no labels are found, default labels are used. The default label is 0.

Report name (summary statistics;)

This parameter, consisting of up to 100 characters, identifies the results in Output 1.

Data columns (a; a,b; b,c,a;)

This parameter specifies the columns on which statistics are to be performed. The column names are the spreadsheet column names, such as a, b, or ac. Any number of columns can be specified, and in any order, as long as each name is separated by a comma. For example, `d, h, AA, b` can ask for statistics on the fourth, eighth, twenty-seventh, and second columns, respectively.

Descriptive statistics (All; Count, Sum, Mean Variance, StdDev, StdErr, Min, Max, Range, CoVariation)

This parameter specifies the type of statistics calculated. Type `Count`, `Sum`, `Mean` to compute the count, sum and mean of each data columns, respectively. Type `All` to compute each type of statistic for each data column. The descriptive statistics names cannot be abbreviated because they would no longer be unique.

Primary Grouping Parameters

The grouping feature categorizes the input data and calculates statistics on each resulting category. Primary grouping parameters allow a data file to be divided into groups; secondary grouping parameters allow each group to be divided into subgroups. You can also examine data that does not fall in a group or subgroup. All grouping features work as described, even when the input data is not in sorted order.

Primary selection column (a; b;)

This parameter is the column used to group the data into categories. This column can contain any type of data, and the data types can be mixed. Only one column can be specified. If no column is specified, the grouping features are disabled, and statistics are computed for the data file as a whole.

Primary selection criteria (5,10,15; male,female;)

These parameters are the category values used to group the input data. One group is created for each item entered into this parameter, and all data that matches a given criterion is included in the corresponding group. The criterion specified can be of any data type, including dates, and different data types can be mixed. If no values are specified, a group is created for each unique value in the primary grouping column.

Break primary selection into groups or ranges? (g; r) sort selection criteria? (y; n)

This parameter modifies the treatment of the `Primary selection criteria` values. If `group` is selected, each group consists of data that exactly matches its corresponding group name. If `range` is selected, each group consists of data that is less than or equal to the group name. For example, using zip codes, if the values 21345, 90125 are given as the selection criteria and group is selected, two groups are created: one for data with zip code 21345, and the other for zip code 90125. If range is selected, then three groups are created: the first for zip codes up through 21345, the second for zip codes from 21346 through 90125, and the last group for zip codes above 90125. All ranges exclude the starting value and include the ending value. To compare data of differing types, a hierarchy is used to rank items. For data derived from a spreadsheet, data is arranged with numbers first, followed by letters, and then dates. The Query tool provides access to a wider variety of data formats. These enhanced data types, ordered from first to last, are:

- Integer numbers

- Real numbers
- Text strings
- Date
- Boolean
- Unspecified
- Error
- N/A
- Full date

Two additional rules augment this hierarchy. First, only the numeric value of integer and real numbers are compared, so that 2 and 2.00 are in the same group. Second, only the day value of dates are compared to eliminate any differences caused by internal formats. The default is `n`.

Show 'Items not selected' in primary selection? (y; n)
This parameter enables or disables displaying the items not selected; that is, items that did not fit into one of the primary groups. This parameter is disabled when ranges are selected or all groups are found, since there are no items not selected in either of these cases. The default is `n`.

The following input data set serves as the basis of an example of primary grouping:

Market	Score
Atlanta	1
Boston	2
New York	3
Boston	4
New York	5
Boston	6
New York	7
Tampa	8

The Elementary transformer parameters are set as follows:

Data column
B

Primary selection column
A

Primary selection criteria
New York, Boston

Descriptive statistic
Sum

When groups are requested, the following output is generated:

Market	Sum
Boston	12
New York	15
Total Selected	27
Items Not Selected	9
Total	36

If ranges are requested, the output becomes:

Market	Sum
Start thru Boston	13
Boston thru New York	15
New York thru End	8
Total	36

In the case of ranges, notice that Atlanta was included in *Start thru Boston*, while Tampa is included in *New York thru End*.

If the `Primary selection criteria` field was left blank, all groups would be located:

Market	Sum
Atlanta	1
Boston	12
New York	15
Tampa	8
Total	36

Secondary Grouping Parameters

The secondary grouping feature divides the primary groups into subgroups. The secondary grouping parameters are very similar to the primary grouping parameters.

Secondary selection column (a; b;)

This parameter is the column used to group each primary group/range into subcategories. This column can contain any type of data, and the data types can be mixed. Only one column can be specified. If no column is specified, the secondary grouping features are disabled, and statistics are computed for the data file as a whole or any primary groupings.

Secondary selection criteria (5,10,15; male,female;)

These parameters are the names of the categories used for each subgroup in the input data. One subgroup is created for each item entered into this parameter, and all data that matches a given criterion is included in the corresponding subgroup. The criterion specified can be of any data type, including dates, and different data types can be mixed. If no values are specified for this parameter, a group is created for each unique value in the secondary grouping column.

Break secondary selection into groups or ranges? (g; r) sort selection criteria? (y; n)

This parameter works like the `Break primary selection into groups or  ranges?` parameter on the secondary criteria.

Show 'items not selected' in secondary selection? (y; n)

This parameter enables or disables displaying the items not selected, that is, items in a particular primary group that did not fit into one of the secondary subgroups. This parameter is disabled when secondary ranges are selected or all secondary groups are found, because there are no items not selected in either of these two cases.

Expert Parameters

The expert parameters allow the user to adjust features of each of the output regions. Answering `yes` to a question enables the feature; answering `no` disables the feature. This section includes an example for each parameter.

Create 'Output 1', 'Output 2', 'Output 3'? (y,y,y; n,n,n)

This parameter allows the user to turn off a particular output region, if it is not required, to increase the speed of the transformer and save file storage space. For example, if you only need Output 2, you would type `no,yes,no` to turn off Output 1 and Output 3. The default is `n,n,n`.

If you do not need to plot elementary output, turn off Output 3 to save time without reducing the amount of information provided.

Show groups/ranges with zero values in Outputs 1, 2, 3? (y,y,y; n,n,n)

This parameter controls the printing of data groups that have no members (all statistics are 0). Answering `no` to this question specifies that all such groups are not printed. This feature is controllable for each output; thus, typing `no,yes,yes` specifies that zero values are shown for Output 2 and Output 3, but not Output 1.

To produce plots with a consistent appearance from one program to the next, do not turn off zero values in Output 3. Doing so might result in fewer items on the x-axis of the plot when a data group is 0. In Output 1, turning off zero values might reduce the size of the Spreadsheet, making it much easier to read. The default is `n,n,n`.

In 'Output 2', insert blank lines after subtotals? (y; n) suppress repeated headings? (y; n)

These parameters control the appearance of the report output. Answering `yes` to the first question adds blank lines after each subtotal, increasing readability. Answering `yes` to the `Suppress repeated heading` parameter causes the name of each data group to be printed only when. The default is `n, n`.

For example, if a particular result is displayed as follows:

Market	Date	Product	Volume
Atlanta	1Q88	A	10
Atlanta	1Q88	B	15
Atlanta	2Q88	A	20
Atlanta	2Q88	B	25

The same result will be displayed as follows when repeated headings are suppressed:

Market	Date	Product	Volume
Atlanta	1Q88	A	10
		B	15
	2Q88	A	20
		B	25

Turn both parameters on if the report is being printed (maximum readability). If the report output is used for further analysis in another transformer or Capsule icon, turn both parameters off, because the missing data items and blank rows can cause problems when the results are used with other tools that expect a data stream with no missing cells.

In 'Output 3', show totals? (y; n) show subtotals? (y; n)

These parameters control which data is sent to the plot output. Answering `yes` to `Show totals` adds the final total of all data read to the graph output. Answering `yes` to `Show subtotals` adds the subtotal of each primary group to the graph. The default is `n, n`.

If no secondary groups are selected, Output 3 is empty unless one of these two questions is answered `yes`. If no groups are selected, the Output 3 is empty unless `Show totals` is answered `yes`.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- Results (Output 1)
- Report (Output 2)
- Plot (Output 3)
- Messages (Output 4)

When you click on any one of these choices, the data associated with that choice displays in the display area of the transformer.

Input Region Names

The Elementary transformer has only one input region, Input 1 (Data). Input 1 contains the columns of data to be used for calculating the statistics. The data can have any number of header rows, or no header. The first row of the header is used for labeling the results. If no labels are entered, default labels are provided. Extra columns of data might be present, but they will be ignored; however, missing data is detected and reported through Output 4.

Whenever a variable in the data set is missing values, the entire observation or row is excluded from all calculations. This is true even if the variables with the missing values are not used in the actual calculation, or if the row or observation contains some partial data.

Consider the following example. A data set contains variables A and B. Variable A contains 10 observations and variable B contains 12. In this data set, the first 10 observations would be used in all calculations and the last two would be excluded, even in calculations where only variable B is specified.

Whenever the transformer program encounters an observation with missing values, a message is displayed in the Important Message window. The message informs the user that the particular observation will be excluded from the calculation.

Output Region Names

The Elementary transformer has four output regions. Each of the first three outputs can be selectively disabled if it is not required.

Output 1 (Results) contains the report title, the type of statistical report, and a table of statistics for each requested group of data. This output is useful for comparing each data column in a group.

Output 2 (Report) contains the same data as Output 1 in a report-oriented format. This output is useful for comparing relations among groups. Output 2 can also be formatted as a data file input into other transformers. Formatting is controlled through the `Insert blank lines after subtotals` and `Suppress repeated headings` parameters.

Output 3 (Plot) contains most of the information found in the first two outputs; however, it is in a format compatible with the Plot tool for displaying the results for reports, presentations, or quick visual verification. Each data group is displayed on one spreadsheet line, with a unique identifier for each group. The unique identifier is a sequence number from 1 to the number of groups, or a name formed from the data group names.

Output 4 (Messages) is used for notes, warnings, and error messages issued by the transformer. Thus, for applications running in a capsule application environment, all messages can be saved by connecting this output to a Spreadsheet or Text icon.

Examples

In this example, the following sales information is stored in a Spreadsheet icon attached to Input 1.

Market	Quarter	Revenue
Atlanta	1.00	32,453,452.00
Boston	1.00	24,562,632.00
New York	1.00	234,756,458.00
Atlanta	2.00	22,462,364.00
Boston	2.00	456,257,626.00
New York	2.00	23,462,364.00
Atlanta	3.00	345,234.00
Boston	3.00	55,467,458.00
New York	3.00	3,436,326.00
Atlanta	4.00	45,634,562.00
Boston	4.00	245,642,564.00
New York	4.00	37,453,452.00

The eastern regional manager wants to know how his region as a whole performed last year, and he wants to know how each of the districts did relative to one another. The manager can get that information by setting the Elementary transformer parameters as follows:

Number of header rows
1

Report name
Elementary Example

Data columns
c

Description Statistics

Sum, Count, Mean, StdDev

Primary selection column

a

Show 'Items not Selected' in secondary selection?

y

Create 'Output 1', 'Output 2', 'Output 3'?

y, y, n

Show groups/ranges with zero values in Outputs 1, 2, 3?

y, y, y

In 'Output 2', insert blank lines after subtotals? suppress repeated headings?

y, y

After the transformer runs, the following information is available in Output 1.

The information in the preceding report shows that the sales in the Boston district

```
Report Name:                                   Elementary Example
Statistics:                                    Descriptive Statistics



Primary Group:                                 Atlanta
Secondary Group:                               Sub-Total

Data Column:                                   Revenue
Count                                          4
Sum                                            100,895,612.00
Mean                                           25,223,903.00
Standard Deviation                             19,108,773.28



Primary Group:                                 Boston
Secondary Group:                               Sub-Total

Data Column:                                   Revenue
Count                                          4
Sum                                            781,930,280.00
Mean                                           195,482,570.00
Standard Deviation                             199,447,162.28



Primary Group:                                 New York
Secondary Group:                               Sub-Total

Data Column:                                   Revenue
Count                                          4
Sum                                            299,108,600.00
Mean                                           74,777,150.00
Standard Deviation                             107,562,629.59
```

```
Primary Group:                    Total Selected
Secondary Group:                  Sub-Total

Data Column:                      Revenue
Count                             12
Sum                               1,181,934,492.00
Mean                              98,494,541.00
Standard Deviation                140,289,706.85


Primary Group:                    All Data Read
Secondary Group:                  Total

Data Column:                      Revenue
Count                             12
Sum                               1,181,934,492.00
Mean                              98,494,541.00
Standard Deviation                140,289,706.85
```

are much stronger than sales in other districts. The average sales per quarter and the total sales across the year are almost three times the average and total sales levels of the other two districts. Alternatively, the large standard deviation values relative to the mean sales for the Boston and New York districts indicate that sales fluctuated a great deal from quarter to quarter. In contrast, sales in the Atlanta region seem to be much more stable.

Elementary Transformer Formulas

The definitions described in Table 62 apply to the equations used in this section where *x* is the column of data.

Table 62. Formula symbol definitions

Symbol	Definition
C	Coefficient of the data elements
E	Standard error of data elements
Min	Minimum value of the data elements

Table 62. Formula symbol definitions

Symbol	Definition
Max	Maximum value of the data elements
n	Number of data elements (the number of rows of data)
R	Range of the data element
s	Standard deviation of data elements
S	Sum of the data elements
s^2	Variance of the data elements
$\bar{x}$	Mean of the data elements
x_i	Any given element

For the column of data *x*:

```
Count: n
```

Sum:

$$S = \sum_{i=1}^{n} x_i$$

Mean:

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

Variance:

$$s^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})}{n-1}$$

Standard Deviation:

$$s = \sqrt{s^2}$$

Standard Error:

$$E = \frac{s}{\sqrt{n}}$$

Coefficient of Variation:

$$C = s/\bar{x}$$

Min: the smallest value in (x_1, x_2, , x_n)
Max: the largest value in (x_1, x_2, , x_n)
Range: R = Max - Min

Forecast

The Forecast transformer predicts future activities by executing eight popular forecasting methods. Using historical data, the transformer and the models it constructs can help answer questions such as:

- What will inventory levels be next week?
- What will sales of an established product be next year?
- What will revenues be next year?

Each of the eight forecasting methods is suited to a specific application. The models include:

- Naive II method
- Linear least squares
- Log least squares
- S-curve least squares
- Single exponential smoothing
- Brown's one-parameter linear exponential smoothing
- Brown's one-parameter quadratic exponential smoothing
- Holt's two-parameter linear exponential smoothing

In addition to constructing one or more of these various models, the transformer evaluates the models and identifies the ones that best fit your data.

Parameters

All of the parameters described in this section are displayed in the Transformer Controls window. Most of these parameters are also displayed in the Data Entry icon contained in the transformer's demonstration capsule application. However, to simplify use of the transformer for new users, some of the more advanced parameters might be omitted from the Data Entry icon; they can be accessed only in the Transformer Controls window.

Number of header rows for 'Input 1', 'Input 2', 'Input 3' (; 1,1,1,; 2,1,1;)

This parameter specifies the number of rows in each of the input regions that are not used in the seasonality analysis. The transformer expects a header row specification for each of the input regions. You can specify any number of header rows; each value must be separated by a comma. The default value is 0,0,0.

Which model(s)? (All; Holts; Log LSQ; Linear LSQ; S-Curve LSQ; Single; Double; Triple; NaiveII)

This parameter specifies the methods to be used to forecast new models. Valid responses are one or more of the following: Linear LSQ, Log LSQ, S-Curve LSQ, Single, Double, Triple, Holts, and NaiveII. These values respectively correspond to the linear least squares, log least squares, S-curve least squares, single exponential smoothing, Brown's-one parameter linear exponential smoothing, Brown's one-parameter quadratic exponential smoothing, Holt's two-parameter linear exponential smoothing, and the naive II modeling methods. To use all the methods, type `All`. There is no default for this parameter. If there are 13 or fewer periods with historical data, the transformer will not have enough information to create an adequate forecasting model. As a result, it will ignore the specification `All` and use only the naive II method to forecast.

If your data set contains many observations with zeros or negative (-n) values, the forecasts produced by the S-curve least squares or log least squares methods can be unstable. This is because both forms of modeling involve log transformations. Because it is impossible to obtain a log of a negative or zero value, these values are reset to either a small positive number or the average of the preceding and following values. Thus, if you use these two modeling techniques with a series having many negative or zero values, the series used to construct the model might be very different than the observed series. This discrepancy could make your model unstable. This complication is inherent in the S-curve and log least squares forecasting methods.

If your time series starts out with one or more zero values followed by nonzero values, the optimal model computed by the transformer might produce results that, while being correct, will seem strange. Any type of increase from a zero value might be shown as an exponential change and overstate the rate of growth. As a result, the double or triple exponential growth models will seem to be optimal. The forecasted values in such a situation will normally rise sharply. Consequently, if the first periods of a series are all zeros (for example, if sales volume for a product shows up as 0 for periods that occurred before the product was introduced), eliminate those periods from the series before using the Forecast transformer.

Length of seasonality (1; 4; 6; 12; 52;)

This parameter specifies the number of periods in each season. Length of seasonality must be one or greater; there is no default.

Column for period #, seasonality, {, Avg # trading days} for 'Input 1' (a,b,c; a,b;)

This parameter specifies the columns containing the period numbers and seasonality adjustment factors in Input 1. You can also optionally specify the column containing the average number of trading days in each period. The column specifications should be separated by a comma. This parameter must be specified even if Input 1 is empty.

Column for date, volume {, other columns to keep} for 'Input 2' (a,b; d,c;)

This parameter specifies the columns in Input 2 that contain the date and data used to derive forecast models. It can also contain specifications for columns that will be copied without modification from Input 2 to Output 1. Column specifications must be separated by commas.

Column for date, period # {, sequence #, trading days} for 'Input 3' (a,b; b,a,d,c;)

This parameter specifies the columns in Input 3 that contain the dates and period numbers used to verify the dates in the other input regions. This parameter can also contain optional specifications for sequence # and trading days. If the sequence number is omitted, the transformer creates one. If the trading days column is omitted, the transformer assumes that each period has one trading day. All column specifications must be separated by commas.

Date resolution (Day; Week; QuadWeek; Month; Quarter; Year; BiMonth; EvenBiMonth; OddBiMonth)

This parameter specifies the level at which the input data is gathered. The valid responses are Day, Week, QuadWeek, Month, Quarter, Year, BiMonth, EvenBiMonth, and OddBiMonth. None of these specifications can be abbreviated. The default value for this parameter is Day. The transformer uses this information to format the date values in the output regions and to check the date values in the input regions for missing data.

De-Seasonalize input data? (y; n) Re-seasonalize output data? (y; n)

This parameter specifies whether to remove the effects of seasonality from the input data before the forecasts are prepared and whether to add the seasonality effects back into the forecasted values. The valid value for each parameter is `Yes` or `No` (`y` or `n`). Each response must be separated by a comma. The default value for this parameter is `n,n`.

Adjust input for trading days? (y; n) readjust output for trading days? (y; n)

This parameter specifies whether the transformer should use the average number of trading days values in Input 1 to remove the effects of trading days from the series before the forecasts are computed and whether that information should be used to add the trading day effect to the forecasted values. The valid value for each parameter is `Yes` or `No` (`y` or `n`). Each response must be separated by a comma. The default value for this parameter is `n,n`.

Periods to begin and end forecast comparison (2,14; 24,72;)

This parameter specifies the beginning and ending periods of the time frame used to evaluate the effectiveness of the forecasting models. You can specify the beginning and ending periods with a positive number (n), negative number (-n), or 0. If the period specification is a positive number, the procedure assumes that the number is an absolute sequence number and that the comparison period starts or ends with the period that matches that sequence number. If 0 is entered, the transformer translates it into the final period of historical data. The transformer translates a negative number into the period that is *n* periods before the final period.

If the specified end period occurs before the beginning period, the transformer resets the beginning value to 3 and the end value to the last period with historical data. The beginning period must occur at least two periods before the end period. If there are fewer periods in the comparison time frame, the transformer adjusts the beginning period so that there are three periods in the comparison time frame. Also, the beginning period must be at least three periods after the first period of historical data. If you specify a beginning period before that, the transformer resets the beginning specification to 3. If the specified end period occurs after the last period of historical data, the transformer resets it to the last period. Finally, if there are fewer than 13 periods of historical data, the transformer ignores the comparison period specification and compares the results of the naive II model across the entire time frame.

Number of periods to forecast (24; 6;)

This parameter specifies the number of periods for which the transformer should forecast new values. Valid values are any integer of one or greater. There is no default value for this parameter.

Set parameters for forecasting models {alpha, beta} (;0.2, 0.2;)

This parameter specifies the alpha and beta values used in the exponential smoothing methods. The two values must be separated by a comma. There are no default values for this parameter. If it is empty, the optimal values of alpha and beta are estimated using the specified or default values of the `Lower limit, upper limit, step increment for alpha, beta` parameter. This parameter does not affect the values of a and b that are estimated in the least squares methods.

Lower limit, upper limit, step increment for alpha, beta (; 0.025, 0.5, 0.1, 0.2, 0.1, 0.2)

This parameter specifies the upper limit, lower limit, and size of the step used in the search for the optimal exponential smoothing alpha and beta values. These values are used only if no alpha or beta values are specified in the `Set parameters for forecasting methods` parameter. The six values must be separated by commas. The first three values specify the lower limit, upper limit, and step for alpha. The second set of three values specify the lower limit, upper limit, and step for beta. The default values for these parameters are 0.0, 1.0, 0.1, 0.0, 1.0, 0.1

This parameter allows values between 0.0 and 1.0.

Output titles on forecast data? (y; n) on model statistics (y; n)
This parameter specifies whether titles are added to Output 1, the forecast data, and Output 2, the model statistics. The valid responses for each specification are `Yes` or `No` (`y` or `n`). There is no default value for this parameter.

Reset zero value data points to average of adjacent data points (y; n)
This parameter specifies whether zero data values are to be replaced with the mean of the value preceding and the value following the zero. Valid responses to this parameter are `Yes` or `No` (`y` or `n`). When computing the log least squares and S-curve least squares model, the transformer must take the log of data points. Because it is impossible to take the log of 0, the transformer must reset zeros to another value. If you type `No` for this parameter, it substitutes 0 with a very small positive number. If you type `Yes`, it substitutes 0 with the mean of the values before and after 0. This substitution is made only during the internal calculations. As a result, zero values are displayed in the output regions. There is no default value for this parameter.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Seasonality (Input 1)
- Data (Input 2)
- Periods (Input 3)
- Results (Output 1)
- Statistics (Output 2)
- Messages (Output 3)

When you click on any one of these choices, the data associated with that choice displays in the display area of the transformer.

Input Region Names

The Forecast transformer has three input regions. All three consist of data read from a Spreadsheet, Query, or SQL Entry tool, or copied directly into the transformer.

Input 1 (Seasonality) contains columns with the period numbers and seasonality factors used to deseasonalize incoming data. This region can also optionally contain a column containing the average number of trading days. If other columns are present, they are ignored. This input region is usually created by the Seasonality transformer as Output 2. The input is optional; if your series does not have a strong seasonal component, you can leave this region empty.

Input 2 (Data) contains the data from which the forecasts will be created and evaluated. It must contain one column with the period number and another column with the volume or input series from which the forecasts will be created. This region can also contain other columns of information not being used in the calculations; they can be copied to Output 1. This input region can be created by the Seasonality transformer as Output 1.

Input 3 (Periods) of the Forecast transformer must contain period table information. For example, it must contain a column with a valid date and a column with a period number. Period numbers should be sequential integers that correspond to each of the periods in a season and repeat from season to season. For example, all months of January could be assigned period numbers of 1, months of February could be assigned 2, and so on. Input 3 can also contain two other optional columns of data. The first, a sequence number column, contains a sequence of integers representing the number of periods in the series, starting with 1. The second, a trading days column, contains the number of trading days in each period.

Although Input 1 and Input 2 can have missing data rows (assuming each contains zeros), the period table, Input 3, cannot have missing data rows, because it is used to determine when data is missing in Input 1 and Input 2. Any data that exists for a period in Input 1 or Input 2 that is not recorded in Input 3 is ignored.

Whenever you have a variable in the data set with missing values, the entire observation or row is excluded from all calculations. This holds true even if the variables with the missing values are not used in the actual calculation or if the row or observation contains some partial data.

Consider the following example. A data set contains variables A and B. Variable A contains 10 observations and variable B contains 12. In this data set, the first 10 observations would be used in all calculations and the last two would be excluded, even in calculations where only variable B is specified.

Whenever the transformer program encounters an observation with missing values, a message is displayed in the Important Message window. The message informs the user that the particular observation will be excluded from the calculations.

Output Region Names

There are three output regions in the Forecast transformer; none is limited in size.

Output 1 (Results), the forecast region, contains the following information:

- The date for each period
- The period number for each period

- The sequence number for each period
- The input series labeled Volume

  If the user specifies that volume should be deseasonalized before forecasting, *Volume* is deseasonalized. Otherwise, *Volume* is the raw volume.
- The seasonality adjustment factor associated with each period
- The trading day adjustment factor associated with each period
- Any columns that were optionally moved from Input 2
- The following information for each type of forecast calculated:
  - — The forecast for the comparison period
  - — The forecast for the forecast period
  - — The forecast adjustment for season and trading day

The order in which the forecast information is displayed in this region depends upon the accuracy of the forecasts. Information for the most accurate forecast is first, the second most accurate forecast is second, and so on.

Output 2 (Statistics), the model summary output, contains information that describes the performance of each of the models during the comparison time frame. For each model, it provides the following information:

- The model's rank based on its mean squared error
- The mean squared error
- The mean percent error
- The mean absolute percent error
- The cumulative error
- The Durbin-Watson statistic
- The alpha values for the exponential smoothing models or a values for the least squares model
- The beta values for the exponential smoothing models or b values for the least squares models

Output 3 (Messages), the messages output region, contains:

- Transformer run-time messages
- Warnings
- Error messages
- A timestamp for documentation purposes

Examples

This example uses the same data set used in the first example of the Seasonality transformer (described in “Seasonality” on page 474). However, in this example, the data has been modified by the Seasonality transformer, which removed the seasonality component from the previous two and one half years of shipments data for a chocolate marketer. The management of the chocolate company wanted to know the seasonally adjusted sales volume over that time period. Now, the management is making out the budget for the coming year. One of the first pieces of information they need is projected sales. To derive that information, they would like to forecast the chocolate shipments for the coming year.

The following information, which is derived from the first four columns of the Seasonality transformer Output 2, shown in Figure 54 on page 485, is presented to Input 1 of the Forecast transformer:

		Final	Avg #
Date	Period	Seasonality	Trade Days
January, 1987	1	1.14	1
February, 1987	2	1.00	1
March, 1987	3	1.04	1
April, 1987	4	1.40	1
May, 1987	5	1.40	1
June, 1987	6	0.61	1
July, 1987	7	0.76	1
August, 1987	8	0.42	1
September, 1987	9	1.26	1
October, 1987	10	0.63	1
November, 1987	11	1.32	1
December, 1987	12	1.01	1

The following information, which comes from the Output 1 region of the Seasonality transformer, is used as Input 2 for the Forecast transformer:

							Trad	
	Seq	Per	Trad	Original	Ses	DeSeas	Day	Final
Date	#	#	Days	Volume	Fact	Volume	Fact	Volume
January, 1986	1	1	1	11,995	1.14	10,502.46	1	10,502.46
February, 1986	2	2	1	8,050	1.00	8,040.08	1	8,040.08
March, 1986	3	3	1	5,809	1.04	5,604.84	1	5,604.84
April, 1986	4	4	1	9,198	1.40	6,554.43	1	6,554.43
May, 1986	5	5	1	11,034	1.40	7,881.33	1	7,881.33

June, 1986	6	6	1	5,480	0.61	8,999.17	1	8,999.17
July, 1986	7	7	1	5,636	0.76	7,391.38	1	7,391.38
August, 1986	8	8	1	2,343	0.42	5,625.82	1	5,625.82
September, 1986	9	9	1	10,870	1.26	8,608.42	1	8,608.42
October, 1986	10	10	1	4,091	0.63	6,476.65	1	6,476.65
November, 1986	11	11	1	11,466	1.32	8,667.70	1	8,667.70
December, 1986	12	12	1	10,960	1.01	10,832.79	1	10,832.79
January, 1987	13	1	1	13,045	1.14	11,421.81	1	11,421.80
February, 1987	14	2	1	9,473	1.00	9,461.33	1	9,461.33
March, 1987	15	3	1	8,645	1.04	8,341.17	1	8,341.10
April, 1987	16	4	1	11,946	1.40	8,512.64	1	8,512.64
May, 1987	17	5	1	12,835	1.40	9,167.74	1	9,167.74
June, 1987	18	6	1	5,743	0.61	9,431.06	1	9,431.06
July, 1987	19	7	1	8,071	0.76	10,584.77	1	10,584.77
August, 1987	20	8	1	3,115	0.42	7,479.48	1	7,479.48
September,1987	21	9	1	12,499	1.26	9,898.50	1	9,898.50
October, 1987	22	10	1	5,513	0.63	8,727.88	1	8,727.88
November, 1987	23	11	1	13,984	1.32	10,571.17	1	10,571.17
December, 1987	24	12	1	11,051	1.01	10,922.73	1	10,922.70
January, 1988	25	1	1	8,935	1.14	7,823.21	1	7,823.21
February,1988	26	2	1	10,379	1.00	10,366.21	1	10,366.21
March, 1988	27	3	1	12,307	1.04	11,874.47	1	11,874.47
April, 1988	28	4	1	16,688	1.40	11,891.75	1	11,891.75
May, 1988	29	5	1	15,873	1.40	11,337.71	1	11,337.71
June, 1988	30	6	1	6,636	0.61	10,897.54	1	10,897.54
July, 1988	31	7	1	7,786	0.76	10,211.01	1	10,211.01
August, 1988	32	8	1	6,374	0.42	15,304.71	1	15,304.71
September,1988	33	9	1	11,733	1.26	9,291.87	1	9,291.87
October, 1988	34	10	1	8,047	0.63	12,739.57	1	12,739.57
November, 1988	35	11	1	11,763	1.32	8,892.21	1	8,892.21
December, 1988	36	12	1	7,192	1.01	7,108.52	1	7,108.52
January, 1989	37	1	1	6,419	1.14	5,620.28	1	5,620.28
February, 1989	38	2	1	6,080	1.00	6,072.51	1	6,072.51

March, 1989	39	3	1	9,677	1.04	9,336.90	1	9,336.90
April, 1989	40	4	1	7,456	1.40	5,313.09	1	5,313.09
May, 1989	41	5	1	17,184	1.40	12,274.13	1	12,274.13
June, 1989	42	6	1	18,850	0.61	30,955.18	1	30,955.18

The third input region for the Forecast transformer is the period table. A portion of that table is shown here:

	Period	Sequence
Date	Number	Number
January, 1986	1	1
February, 1986	2	2
March, 1986	3	3
.	.	.
.	.	.
October, 1990	10	58
November, 1990	11	59
December, 1990	12	60

Note the relatively large range of alpha and beta values and the small step increment specified on the `Lower limit, upper limit  for alpha, beta` parameter. These values force the transformer to search through many possible alpha and beta values to find the optimal values. The advantage is that the program is most likely to find the best parameter values; the disadvantage is that it requires additional processing time.

The Forecast transformer parameters used in this analysis are set as follows:

Number of header rows for 'Input 1', 'Input 2', 'Input 3'
2, 3, 2

Which Model
All

Length of seasonality
12

Column for period #, Seasonality, for 'Input 1'
b, c

Column for date, volume for 'Input 2'
a, g

Column for date, period # for 'Input 3'
a, b, c

Date resolution
Month

De-Seasonalize input data?
n, n

Adjust input for trading days? readjust output for trading days?
n, n

Periods to begin and end forecast comparison
–24, –12

Number of periods to forecast
18

Lower limit, upper limit, step increment for alpha, beta
0.1, 0.8, 0.05, 0.1, 0.8, 0.05

Output titles on forecast data?
y, y

Reset zero value data points to average of adjacent data points
n

After the transformer runs, it writes the comparison forecasts, final forecasts and adjusted forecasts for each type of model to Output 1. The comparison and final forecasts are shown in the following example for only the modeling technique with the lowest MSE during the comparison period. Also, only the output for the comparison period and later periods is shown.

	Per	Seq			Holts	Holts
Date	#	#	Volume	Season	Comp	Frcst
August, 1987	8	20	7,479	0.42	8,641.18	
September, 1987	9	21	9,899	1.26	8,688.76	
October, 1987	10	22	8,728	0.63	9,246.93	
November, 1987	11	23	10,571	1.32	9,443.77	
December, 1987	12	24	10,923	1.01	10,003.72	
January, 1988	1	25	7,823	1.14	10,549.56	
February, 1988	2	26	10,366	1.00	10,284.54	
March, 1988	3	27	11,874	1.04	10,583.57	
April, 1988	4	28	11,892	1.40	11,163.18	
May, 1988	5	29	11,338	1.40	11,652.18	

June, 1988	6	30	10,898	0.61	11,923.14	
July, 1988	7	31	10,211	0.76	12,021.10	
August, 1988	8	32	15,305	0.42		
September, 1988	9	33	9,292	1.26		
October, 1988	10	34	12,740	0.63		
November, 1988	11	35	8,892	1.32		
December, 1988	12	36	7,109	1.01		
January, 1989	1	37	5,620	1.14		
February, 1989	2	38	6,073	1.00		
March, 1989	3	39	9,337	1.04		
April, 1989	4	40	5,313	1.40		
May, 1989	5	41	12,274	1.40		
June, 1989	6	42	30,955	0.61		
July, 1989	7	43		0.76		13,263.13
August, 1989	8	44		0.42		13,708.55
September, 1989	9	45		1.26		14,153.96
October, 1989	10	46		0.63		14,599.38
November, 1989	11	47		1.32		15,044.80
December, 1989	12	48		1.01		15,490.22
January, 1990	1	49		1.14		15,935.63
February, 1990	2	50		1.00		16,381.05
March, 1990	3	51		1.04		16,826.47
April, 1990	4	52		1.40		17,271.89
May, 1990	5	53		1.40		17,717.31
June, 1990	6	54		0.61		18,162.72
July, 1990	7	55		0.76		18,608.14
August, 1990	8	56		0.42		19,053.56
September, 1990	9	57		1.26		19,498.98
October, 1990	10	58		0.63		19,944.39
November, 1990	11	59		1.32		20,389.81
December, 1990	12	60		1.01		20,835.23

Figure 49 shows the data in plot form.

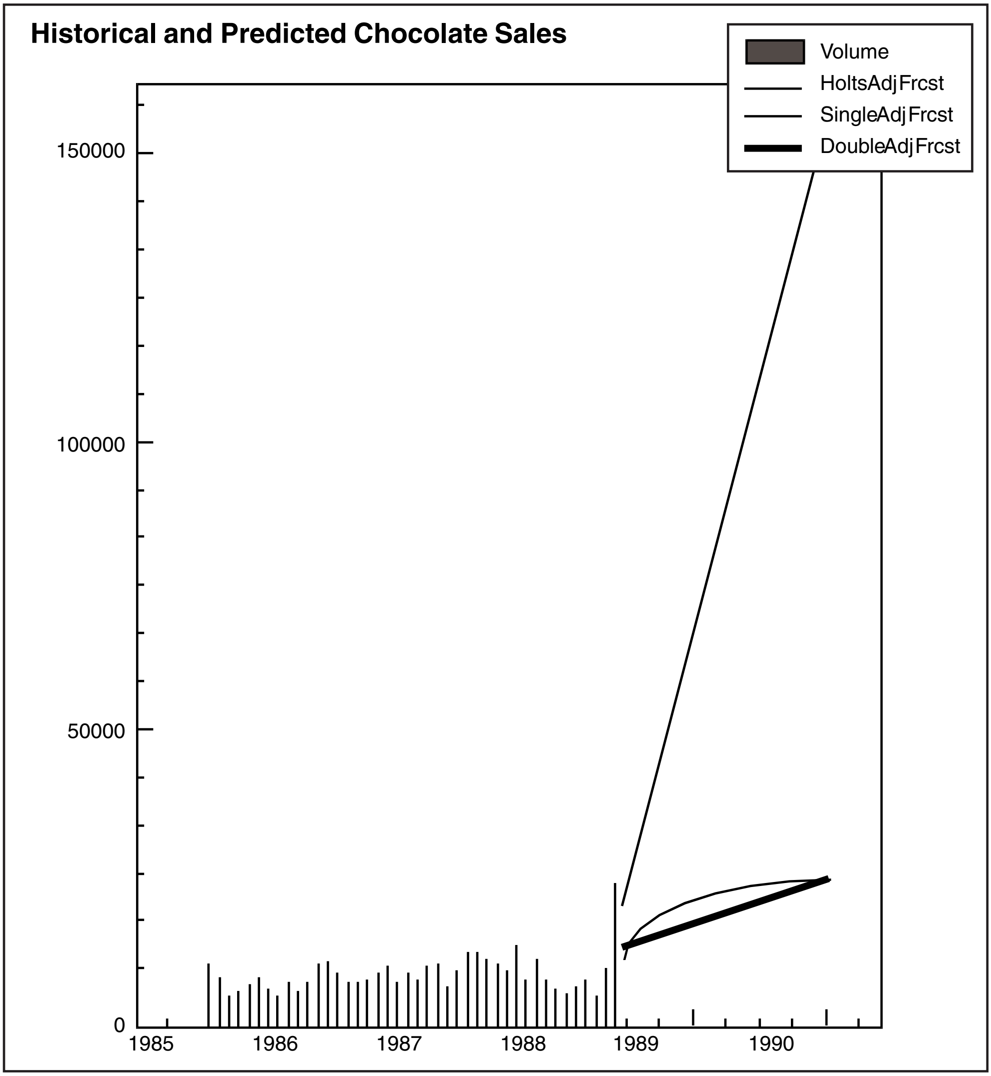


Figure 49. Forecasting techniques

Figure 49 shows that the forecasting technique that yields the smallest mean squared error is the Holt's two-parameter method. Unfortunately, because of the nature of all of the exponential smoothing techniques, forecasts of future chocolate shipments were heavily influenced by the last month of historical data. That period had an unusually high deseasonalized volume, which was largely an artifact introduced when the Seasonality transformer adjusted the unusually high raw volume. Because of the emphasis on the last period of historical data and the fact that exponential smoothing methods are best suited for short-term forecasts, the forecasts for the final projected periods seem wrong. For example, the final forecast volume is nearly 150,000 cases (the last value in the Holts Frcst column).

This value represents an increase of almost 500 percent from the deseasonalized chocolate shipments of the last period of historical data, which was only 18 months before the end of the forecast period.

The contents of Output 2 are shown in the following example to provide a means of checking the accuracy of the other modeling techniques:

A	B	C	D	E	F	G	H	I
	Rank by	Mean	Mean	Mean Abs	Cumulative	Durbin-		
Model	MSE	Sqr Err	% Err	% Err	Err	Watson	alpha	beta
LOG LSQ	1	1,391,233.80	(2.59)	10.24	(1,030.49)	1.77	8.95	0.01
Linear LSQ	2	1,450,217.945	(3.13)	10.26	(1,595.90)	1.70	7,228.98	114.51
Holts	3	1,605,173.70	(3.10)	11.28	(2,199.94)	1.90	0.20	0.15
Triple	4	1,877,091.38	(0.73)	12.66	1,234.84	1.83	0.10	
Double	5	1,921,205.12	0.28	12.73	2,511.61	1.80	0.15	
Single	6	1,973,236.03	2.09	12.66	4,757.37	1.86	0.25	
S-Curve LSQ	7	3,003,413.47	8.22	15.42	12,395.97	0.83	9.15	(0.23)
Naive II	8	3,308,936.99	(2.31)	16.06	(373.77)	2.57		

This example shows that the exponential smoothing methods produced the most accurate forecasts during the comparison period. In addition, it is obvious that the alpha values chosen for these models (see the column titled `alpha`) are relatively low between 0.10 to 0.35, indicating that they were most accurate when they did not place too much emphasis on recent historical values. In addition, it is reassuring that only the S-curve LSQ method is less accurate than the naive II method, in which forecast values are simply the deseasonalized volume from the last historical period.

Although the exponential smoothing models were the most accurate during the comparison period, the forecasts produced by the simple linear least square model were not much worse than those produced by the Holt's method. This fact is proven by the values in the Mean % Error column. The Holt's method produced forecasts that were off by 10.2 percent, whereas the linear least squares method produced forecasts that were off by 10.6 percent.

Because of the exaggerated nature of the final forecasts produced by the Holt's model and the fact that the linear least squares model produced forecasts that were nearly as accurate, the linear least squares results will be examined. In general, the least squares techniques are best suited for relatively long-term forecasts, such as this example. Because there is little evidence in the historical data of an exponential or S-curve growth rate, the linear least squares technique seems to be the most appropriate choice for providing the best forecasts of chocolate shipments in the coming year.

You can locate the linear least squares forecast information in Output 1. However, you can easily rerun the transformer and request only that modeling technique. In addition, you can add the seasonality component back into the forecast values and move the raw chocolate historical information from Input 2 to Output 1. Doing this will help you compare the forecast values to the raw chocolate shipments data.

When the transformer runs again, it uses the same input data used in the previous example. However, the parameter settings will change slightly so that the transformer provides the information previously outlined. The Forecast transformer parameters for the second run are set as follows:

Number of header rows for 'Input 1', 'Input 2', 'Input 3'
2, 3, 2

Which Model
Linear LSQ

Length of seasonality
12

Column for period #, Seasonality, for 'Input 1'
b, c

Column for date, volume for 'Input 2'
a, g, e

Column for date, period # for 'Input 3'
a, b, c

Date resolution
Month

De-Seasonalize input data?
n, y

Adjust input for trading days? readjust output for trading days?
n, n

Periods to begin and end forecast comparison
–24, –12

Number of periods to forecast
18

Output titles on forecast data?
y, y

Reset zero value data points to average of adjacent data points
n

After the transformer runs, the comparison forecast, final forecasts, seasonally adjusted forecasts, and other information are sent to Output 1, shown in the following report. To minimize the amount of space used in this example, information for only the periods during and after the comparison interval are shown.

						Linear	Linear	Linear
						LSQ	LSQ	LSQ Adj
Date	Period #	Seq #	Volume	Seasonality	Orig Column	Compare	Forecast	Frcst
August, 1987	8	20	7,479.48	0.42	3,115	9,642.66		
September, 1987	9	21	9,899.50	1.26	12,499	9,762.14		
October, 1987	10	22	8,728.88	0.63	5,513	9,881.62		
November, 1987	11	23	10,571.17	1.32	13,984	10,001.00		
December, 1987	12	24	10,922.73	1.01	11,051	10,120.58		
January, 1988	1	25	7,823.21	1.14	8,935	10,240.06		
February, 1988	2	26	10,366.21	1.00	10,379	10,359.54		
March, 1988	3	27	11,874.47	1.04	12,307	10,479.02		
April, 1988	4	28	11,891.75	1.40	16,688	10,598.49		
May, 1988	5	29	11,337.71	1.40	15,873	10,717.97		
June, 1988	6	30	10,897.54	0.61	6,636	10,837.45		
July, 1988	7	31	10,211.01	0.76	7,786	10,950.00		
August, 1988	8	32	15,304.71	0.42	6,374			
September, 1988	9	33	9,291.87	1.26	11,733			
October, 1988	10	34	12,739.57	0.63	8,047			
November, 1988	11	35	8,892.21	1.32	11,763			
December, 1988	12	36	7,108.52	1.01	7,192			
January, 1989	1	37	5,620.28	1.14	6,419			
February, 1989	2	38	6,072.51	1.00	6,080			
March, 1989	3	39	9,336.90	1.04	9,677			
April, 1989	4	40	5,313.09	1.40	7,456			
May, 1989	5	41	12,274.13	1.40	17,184			

						Forecast	
June, 1989	6	42	30,955.18	0.61	18,850		
July, 1989	7	43		0.76		12,152.75	9,267
August, 1989	8	44		0.42		12,267.26	5,152.25
September, 1989	9	45		1.26		12,381.76	15,601.02
October, 1989	10	46		0.63		12,496.27	7,872.65
November, 1989	11	47		1.32		12,610.78	16,646.22
December, 1989	12	48		1.01		12,725.28	12,852.54
January, 1990	1	49		1.14		12,839.79	14,637.36
February, 1990	2	50		1.00		12,954.30	12,954.30
March, 1990	3	51		1.04		13,068.80	13,591.55
April, 1990	4	52		1.40		13,183.31	18,456.63
May, 1990	5	53		1.40		13,297.81	18,616.94
June, 1990	6	54		0.61		13,412.31	8,181.52
July, 1990	7	55		0.76		13,526.83	10,280.39
August, 1990	8	56		0.42		13,641.33	5,729.36
September, 1990	9	57		1.26		13,755.84	17,332.36
October, 1990	10	58		0.63		13,870.35	8,738.32
November, 1990	11	59		1.32		13,984.85	18,460.01
December, 1990	12	60		1.01		14,099.36	14,240.35

Figure 50 shows the data in plot form.

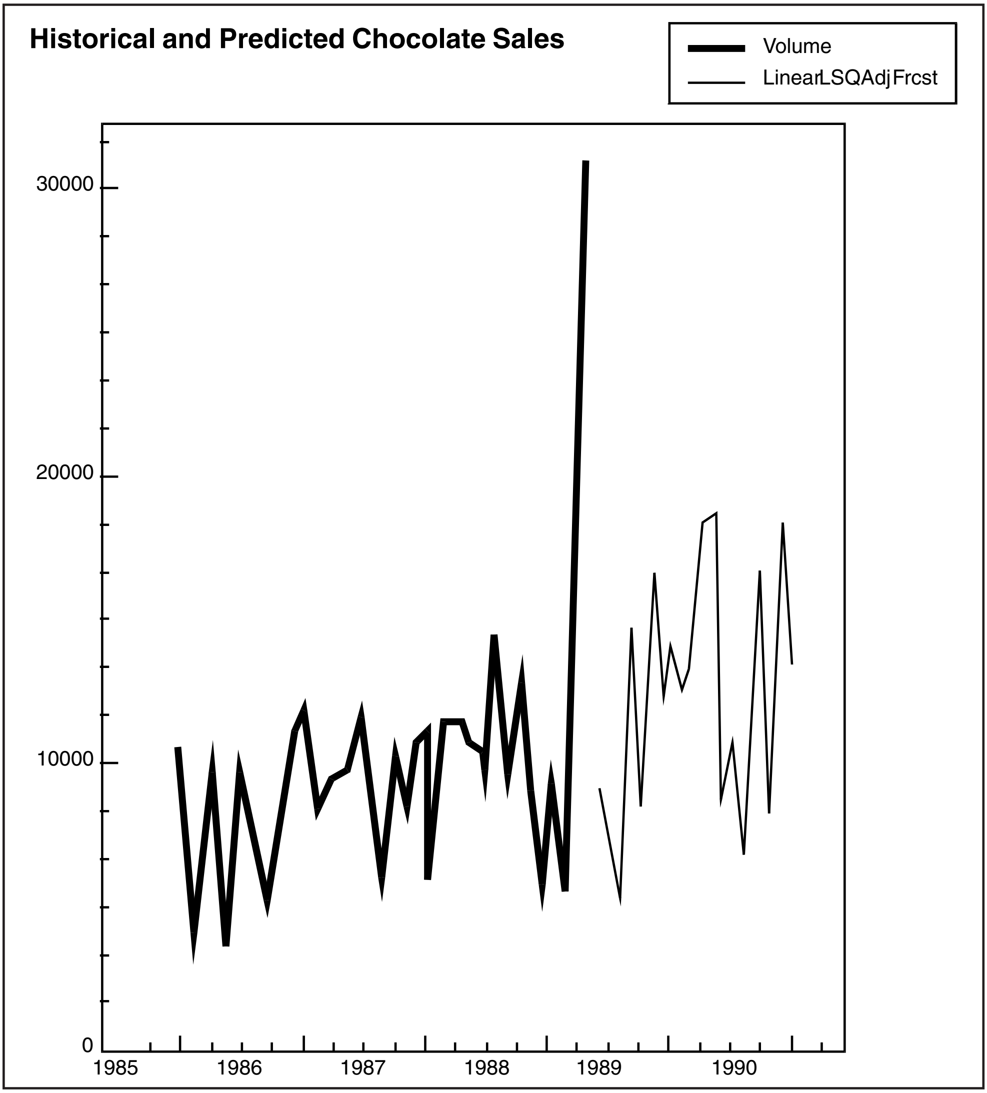


Figure 50. Linear least squares model

This information shows that the linear least squares model produces more reasonable long-term forecasts. In general, if you are trying to forecast one to six periods, the exponential smoothing techniques probably produce the best forecasts. In contrast, if you are forecasting six or more periods into the future, the least squares techniques usually provide the most reliable forecasts.

The information in Figure 50 also shows the effect of adding the seasonal component back into the forecasted values. For example, in June 1990, the original (deseasonalized) forecast value in the column called Lin LSQ Frcst is 13,412 cases. After the seasonality component is added, the forecast in the column titled Lin LSQ Adj Frcst is 8,167 cases. These measures correspond to the deseasonalized shipment level of 30,955 and raw shipment level of 18,850 of a year prior (in the Orig Vol and Vol columns, respectively). Adding the seasonal effect is especially important during the budgetary process, when a corporation plans revenues and expenses for each period. Because the linear LSQ method seems to provide stable and reasonable forecasts of chocolate shipments, management can use this information while preparing the 1990 budget.

Forecasting and Analyzing Data

The Forecast transformer is designed to analyze and summarize the trend in a series using a mathematical formula. It then uses that formula to project the trend into the future. Because of the way in which all of the forecasting methods summarize a trend, any form of seasonality will confuse the technique and result in an inferior model. If your data shows evidence of seasonality, you should use the Seasonality transformer to remove the seasonal component from the series before analyzing the series with the Forecast transformer.

Although the Forecast transformer must analyze a deseasonalized series, it can read a raw series and deseasonalize it using seasonality factors computed by the Seasonality transformer. After the forecasts are computed, it can also add the seasonality component back into forecasted values. The two transformers are designed to work together. For more information on using these two transformers together, see “Using the Seasonality Transformer with the Forecast Transformer” on page 495.

Besides the seasonality component, other time series components can confuse the Forecast transformer so that it produces less than optimal forecasts. In such situations, there are ways to remove those components from a series. For example, if there is a great deal of randomness in a series, computing a moving average using the Moving transformer or the Seasonality transformer can reduce the impact of that component. Alternatively, if a series contains a great deal of the cycle component, using a longer or shorter series might help reduce its impact. If a series contains the results of an anomaly (such as a strike, product recall, or manufacturing change), you might be able to adjust data to minimize the impact of that event.

Forecasting Methods

The Forecast transformer creates forecasts using one or more of the following forecasting methods:

- Naive method II
- Linear least squares
- Log least squares
- S-curve least squares
- Single exponential smoothing
- Brown's one-parameter linear exponential smoothing

- Brown's one-parameter quadratic exponential smoothing
- Holt's two-parameter linear exponential smoothing

The forecasting procedure consists of several steps:

1. The transformer creates initial forecasts using data for the first period through the period preceding the specified comparison period.
2. The transformer uses the initial model to forecast the values of the series during the comparison period. For the comparison period, the predicted values are then subtracted from the actual value. The result of this subtraction is called the error.
3. The error values are squared and averaged for all of the intervals in the comparison period.
4. The models are ranked in ascending order, based on the mean of the squared errors (MSE); having the smallest mean squared error value, the first model is the one that best forecasted the values in the comparison period.
5. The transformer constructs a new set of final models that incorporate all of the historical series, including the values in the comparison period.
6. The transformer uses this set of models to compute the final forecasted series. Thus, it uses only a portion of the series to compute the initial models, but then uses all of the historical data to construct the final models.

Constructing Models

The eight models constructed by the Forecast transformer are based on three methods:

- Naive method II
- Least squares methods
- Exponential smoothing methods

Naive Method II Model

The naive method II model is the simplest of the forecasting models. The forecast for all future periods for this model is the deseasonalized last historical value of a series. This method provides a baseline for other models. If none of them is better at predicting values of a series, review the series carefully to make certain that a relatively stable trend exists.

Least Squares Methods Model

The least squares group of forecasting models uses the least squares statistical technique to fit a trend line to a series and forecast future values. The technique used by these methods forms the basis of regression analysis, which constructs an equation describing a line. The goal of building the equation is to specify a line that is closest to all of the observations. The distance between a regression line and all of the observations is measured by summing the squares of the vertical distance between each observation and the line. Thus, the line with the least squares is the best predictor.

With these models, all values in a series have the same influence on the forecast. In each of these models, the series is the dependent variable, and time is the independent variable. In general, these models are more appropriate for long-term forecasting. Each of the three least square models is described in this section.

The linear least squares model estimates a series by adding a constant or intercept to the product of the multiplication of the time period by a coefficient. It is useful for long-term forecasting of relatively smooth series that have trends that increase or decrease at a constant level.

The log least squares model uses a linear least squares equation as an exponent to the value e˜2.71828 (the base of the natural logarithm) to predict the values of a series. Another way of evaluating the log least squares equation is to use the standard linear least squares equation to predict the logarithm of the series. This model is also known as the exponential growth model, and it is best suited for long-term forecasting when growth rates grow exponentially.

The S-curve least squares model is also built on the standard linear least squares equation. However, in this equation, the time period value is used as a divisor of the coefficient. This modified linear least squares formula is then used as the exponent of the natural log value. Another way to evaluate this equation is that the log of the series value is equal to the intercept plus the result of dividing the coefficient by the time period. This model is best for fitting long-term forecasting of series to trends whose growth starts slowly, accelerates quickly, peaks, and eventually levels off.

Exponential Smoothing Methods Model

The exponential smoothing technique is composed of four different models based on the premise that many series have a built-in memory; that is, the best predictor of any point in a series is the previous value of the series. These models are best suited for making short-term forecasts because they place the most predictive power on the most recent values in a series. They lose their predictive power as more and more periods are forecast and they use up the most recent information. These models are relatively easy to interpret and compute and are very popular in certain applications that require repeated forecasting of the same series.

Within the set of exponential smoothing models, each model is computed differently and, as a result, predicts different types of trends. Though they use different calculations, all models use a coefficient called alpha. Alpha is a number ranging from 0 to one that represents the weight given to different values in a series when a new series is estimated. When alpha is equal to one, only the most recent observation in a series is used to forecast a series. When it is equal to 0, earlier observations have more weight in estimating the level of a series; thus, more data is used and the resulting estimated trend is smoother. As alpha gets larger, fewer observations influence the estimated value; thus, the estimates are very responsive to changes in the observed trend. A brief description of each technique follows.

The single exponential smoothing model estimates a value in a series based on the previous forecast value and the previous observation. The previous forecast is multiplied by one minus alpha and added to the product of multiplying alpha times the previous observed value. Thus, each predicted value carries information from previous values, so that the influence of earlier values decreases as the forecast proceeds. In this model, smaller values of alpha are most appropriate for series that have a large amount of randomness; larger alphas are most appropriate for series that are relatively stable. Single exponential smoothing is most appropriate for forecasting series with little or no trend.

Brown's one-parameter or double linear exponential smoothing model starts with the same calculations used by the single exponential smoothing model but uses additional calculations to estimate the trend component, resulting in double exponential smoothing. The value of alpha should be significantly smaller than the alpha used in the single exponential model, in most cases falling between 0.1 and 0.3. This model is most appropriate for forecasting series that show a significant upward or downward linear trend.

Brown's one-parameter quadratic or triple exponential smoothing model is an extension of the one-parameter linear exponential smoothing model. Besides the double exponential smoothing technique, it also uses a triple exponential smoothing. Together, these smoothing techniques can account for quadratic trends. The values of alpha usually vary between 0.1 and 0.3. Whereas the equation used by the linear smoothing model translates into a straight line, the equation used in the quadratic model translates into a parabola. Because of this, the quadratic technique does very well at predicting series that form curves or have turning points. This characteristic also makes this technique susceptible to randomness in the data, which could cause very erratic forecasts.

Holt's two-parameter linear exponential smoothing model is similar to the one-parameter linear technique in that it identifies the trend component and uses it in the forecast. However, as the name implies, in addition to the alpha coefficient used by the other exponential smoothing techniques, it uses a second coefficient, beta. The values of beta and alpha are analogous, beta is used in the equations to estimate the trend whereas alpha is used to smooth the most recent values of the series and thus reduce randomness. Although this model has the disadvantage of requiring two parameters, it is useful for certain types of series for which different weights should be assigned to the randomness and trend components.

Setting the Values of Alpha and Beta

All of the exponential smoothing techniques use a parameter called alpha, and Holt's uses a second parameter called beta, to weight the effects of past observations on forecast values. When the values of alpha and beta are near one, more emphasis is placed on the most recent observations. Alpha or beta values near 0 place more emphasis on past values and result in smoother forecasts.

You specify the value of alpha in the Transformer Controls window of the Forecast transformer. If you provide a specific alpha value, all of the exponential smoothing models use it to construct forecasting models. However, each exponential smoothing method reacts differently to the same alpha value. Consequently, the alpha value that provides the best results for one modeling technique might not be best for another technique. Because of this difference and the difficulty in determining the optimal alpha value for even one model, the Forecast transformer provides a function for automatically estimating the best value of alpha for each exponential modeling method. If you do not enter alpha and beta values, the transformer automatically estimates optimal values.

The transformer constructs forecast models for all of the possible values of alpha and beta, which are constrained by the specified or default lower limit, upper limit and step increment values. For each type of model, the transformer then determines the value that results in a model with the smallest mean of the squared errors (MSE) value. For example, if the Holt's exponential smoothing method is chosen, the transformer finds the combination of alpha and beta that results in the most accurate model during the comparison period. These optimal alpha and beta parameters are then used in making the final set of forecasts. The alpha and beta values used by each exponential smoothing technique are then written to Output 2.

Forecasting Transformer Formulas

The definitions described in Table 63 apply to the forecasting methods used in this section.

The naive method II forecasted values are calculated as follows:

Table 63. Forecast transformer symbol definitions

Symbol	Definition
a	Intercept of the line used for linear least squares forecast method.
a¢	Intercept of the line used for exponential growth least squares forecast method.
a¢¢	Intercept of the line used for S-curve least squares forecast method.
alpha	Coefficient that smooths the average value of the data.
a_m	Estimate of current data value in Brown's Quadratic method at time *m.*
b	Slope of the line used for linear least squares forecast methods.
b¢	Slope of the line used for exponential growth least squares forecast method.
b¢¢	Slope of the line used for S-curve least squares forecast method.
beta	Coefficient that smooths the slope (or trend) of the data.
b_m	Linear trend estimate in Brown's Quadratic method at time *t*.
c	Number of periods in the comparison range.
CE	Cumulative error of the model.
c_m	Quadratic trend estimate in Brown's Quadratic method at time m.
DW	Durbin-Watson statistic for the model
e	A constant approximately equal to 2.71828 (the base of the natural logarithm).
F_t	Forecast value for any given period.
m	Number of periods of historical data.
n	Number of periods to forecast.
MSE	Mean squared error of the model.
MPE	Mean percent error of the model.
MAPE	Mean absolute percent error of the model.
R_t	Residual for any given time period.
S_m	Holt's single exponentially smoothed value of data for timem.
$S¢_m$	Single exponentially smoothed value of the data for time m.
$S¢¢_m$	Double exponentially smoothed value of the data for time *m*.
$S¢¢¢_m$	Triple exponentially smoothed value of the data for time *m*.
t	Sequence number for the observation or forecast value. This value is computed by the transformer and is not obtained from Input 3.
T_m	Smoothed value of the trend for time m.
X_m	Actual value for any given historical value at time m.

Table 63. Forecast transformer symbol definitions

Symbol	Definition
X_t	Actual (if historical) or forecast (if not historical) value for time t.
X¢$_t$	Natural log of the actual (if historical) or forecast (if not historical) value for time t.

$$F_t = X_m$$

for `t = 1, 2, 3, ¼, n`.

In the following calculations for the comparison period *m*, the number of periods of historical data, includes values up to the beginning of the comparison period. In the calculation of the final measures, *m* includes all historical values.

Two parameters (a and b) are displayed in the linear least squares methods and are calculated as follows:

$$a = \frac{\sum_{t=1}^{m} X_t}{m} - \left[b \times \frac{\sum_{t=1}^{m} t}{m} \right]$$

$$b = \frac{\left[m \times \sum_{t=1}^{m} (t \times X_t) - \sum_{t=1}^{m} t \times \sum_{t=1}^{m} X_t \right]}{\left[m \times \sum_{t=1}^{m} t^2 \right] - \left[\sum_{t=1}^{m} t \right]^2}$$

The linear least squares forecast values are calculated as follows:

$$F_t = a + (b \times t)$$

`for t = 1, 2, 3, ¼, n`

Two parameters (a' and b') are displayed in the log least squares methods and are calculated as follows:

$$a' = \frac{\sum_{t=1}^{m} X'_t}{m} - \left[b' \frac{\sum_{t=1}^{m} t}{m} \right]$$

$$b' = \frac{\left[m \times \sum_{t=1}^{m} (t \times X'_t) - \sum_{t=1}^{m} t \times \sum_{t=1}^{m} X'_t \right]}{\left[m \times \sum_{t=1}^{m} (t)^2 \right] - \left[\sum_{t=1}^{m} t \right]^2}$$

The log least squares forecast values are calculated as follows:

$$F_t = e^{a' + b't}$$

```
for t = 1, 2, 3, ¼, n
```

Two parameters (a' and b') are displayed in the S-curve least squares methods and are calculated as:

$$a'' = \frac{\sum_{t=1}^{m} X'_t}{m} - \left[b'' \frac{\sum_{t=1}^{m} \frac{1}{t}}{m} \right]$$

$$b'' = \frac{\left[m \times \sum_{t=1}^{m} (\frac{1}{t} \times X'_t) - \sum_{t=1}^{m} \frac{1}{t} \times \sum_{t=1}^{m} X'_t \right]}{\left[m \times \sum_{t=1}^{m} (\frac{1}{t})^2 \right] - \left[\sum_{t=1}^{m} \frac{1}{t} \right]^2}$$

The S-curve least squares forecast values can be calculated as follows:

$$F_t = e^{a'' + \frac{b''}{t}}$$

`for t = 1, 2, 3, ¼, n`

The single exponential smoothing forecast values are calculated as follows:

$$F_{t+1} = (\text{alpha} \times X_t) + (1 - \text{alpha}) \times F_t$$

`for t = m, m + 1, m + 2, ¼, m + n`

The Brown's one-parameter linear exponential smoothing forecasted values are calculated as:

$$S'_m = (\text{alpha} \times X_m) + (1 - \text{alpha}) \times S'_{m-1}$$

$$S''_m = (\text{alpha} \times S'_m) + (1 - \text{alpha}) \times S''_{m-1}$$

$$b_m = \frac{\text{alpha}}{1 - \text{alpha}} \times (S'_m - S''_m)$$

$$a_m = (2 \times S'_m) - S''_m$$

$$F_{m+i} = a_m + (b_m \times i)$$

```
for i = 1, 2, ¼, n when i = 1:

$S_m$ = $X_m$
$S_m$ = $S_m$
```

The Brown's one-parameter quadratic exponential smoothing forecasted values are calculated as follows:

$$S'_m = (\text{alpha} \times X_m) + (1 - \text{alpha}) \times S_{m-1}$$

$$S''_m = (\text{alpha} \times S'_m) + (1 - \text{alpha}) \times S''_{m-1}$$

$$S'''_m = (\text{alpha} \times S''_m) + (1 - \text{alpha}) \times S'''_{m-1}$$

$$a_m = (3 \times S'_m) - (3 \times S''_m) + S'''_m$$

$$b_m = \left[\begin{array}{c} \dfrac{\text{alpha}}{2 \times (1 - \text{alpha})^2} \times \\ [((6 - (5 \times \text{alpha})) \times S'_m) - ((10 - (8 \times \text{alpha})) \times S''_m) + (4 - (3 \times \text{alpha})) \times S'''_m] \end{array} \right.$$

$$c_m = \frac{\text{alpha}^2}{(1 - \text{alpha})^2} \times [S'_m - (2 \times S''_m) + S'''_m]$$

$$F_{m+i} = a_m + (b_m \times i) + [0.5 \times c_m \times i^2]$$

```
for i = 1, 2, ¼, n
when i = 1:
S¢$_m$ = X$_m$
S¢¢$_m$ = S¢$_m$
S¢¢$_m$ = S¢¢$_m$
```

The Holt's two-parameter linear exponential smoothing forecasted values are calculated as follows:

$$S_m = (\text{alpha} \times X_m) + (1 - \text{alpha}) \times (S_{m-1} + T_{m-1})$$

$$T_m = \text{beta} \times (S_m - S_{m-1}) + (1 - \text{beta}) \times T_{m-1}$$

$$F_{m+i} = S_m + (T_m \times i)$$

```
for i = 1, 2, ¼, n (n = number of periods to forecast)
```

where:

```
m = 1: So = Xo
To = X1 -Xo
```

Comparing Model Statistics

The residual for any given comparison period is calculated as:

$$R_t = X_t - F_t$$

The mean squared error is calculated as follows:

$$MSE = \frac{\sum_{t=1}^{c} R_t^2}{c}$$

The mean percent error is calculated as follows:

$$MPE = 100 \times \left[\frac{\sum_{t=1}^{c} \frac{R_t}{X_t}}{c} \right]$$

The mean absolute percent error is calculated as follows:

$$MAPE = 100 \times \left[\frac{\sum_{t=1}^{c} \frac{|R_t|}{X_t}}{c} \right]$$

The Durbin-Watson statistic is calculated as follows:

$$DW = \frac{\sum_{t=1}^{c} (R_t - R_{t-1})^2}{\sum_{t=1}^{c} R_t^2}$$

Moving

The Moving transformer allows users to compute moving averages and rolling sums. The following types of moving averages and rolling sums are directly supported:

- Single moving average
- Double moving average
- Spencer's 15-term weighted moving average
- Henderson's 5-term weighted moving average
- Henderson's 9-term weighted moving average
- Henderson's 13-term weighted moving average
- Henderson's 23-term weighted moving average
- n-term rolling sum

Moving averages and rolling sums are widely used in inventory control, statistical quality control, time-series analysis, promotion analysis, and data smoothing. Moving averages also form the basis of most analyses of seasonality algorithms, including the census X-11 decomposition method implemented in the Seasonality transformer.

Moving averages redistribute events that occur briefly over a wider period of time. This redistribution serves to remove noise, random occurrences, and large peaks or valleys from time-series data. The moving average method can be applied to a time-series data set to remove the effects of seasonal variations, extract the data trend, enhance the long-term cycles, and smooth a data set before performing higher level analysis.

The single moving average is the most basic of all the moving average methods. The algorithm creates a new time-series data set in which each element is a linear combination of past and future data values; all data values are weighted equally. For example, a five-term single moving average creates a new time series in which each new term is the average of the previous two periods, the current period, and the two immediately following periods, each equally weighted.

The single moving average is useful in a number of applications. For example, a three-term to five-term single moving average reduces the effects of period misalignment. Period misalignment occurs when an event that affects a time-series data set does not occur during the same period each year. An example would be a summer sale promotion, which would generally be scheduled about the same time each year, but not necessarily during the same period. A single moving average over one year removes the effects of seasonal variations by redistributing the variations across the year. The yearlong moving average yields a data set that represents the data trend. A moving average that is one period shorter or longer than a year tends to enhance the effects of random variations while reducing the effects of seasonality. Three-month, nine-month, and fifteen-month moving averages (encompassing enough terms to represent the specified time periods) work well for smoothing a time-series data set.

A double moving average is a moving average applied to the results of a single moving average. The net result is that the periods near the center of the double moving average are weighted much more heavily than periods further away from the center period. A double moving average filters out a great deal more of the period-to-period variations than does a single moving average. It is useful for removing residual noise, long-term cycles, and seasonal effects from a data set.

The Moving transformer supports several special-purpose moving average algorithms. Spencer's 15-term moving average is designed for analyzing the noise component of time-series data sampled at monthly intervals. Henderson's moving averages weight past and future time periods in a nonlinear fashion, and tends to remove higher-order noise components.

Parameters

All of the parameters described in this section are displayed in the Transformer Controls window. Most of these parameters are also displayed in the Data Entry icon contained in the transformer's demonstration capsule application. However, to simplify use of the transformer for new users, some of the more advanced parameters might be omitted from the Data Entry icon; they can be accessed only in the Transformer Controls window.

Number of header rows (; 1; 5;) row containing titles (,1; ,5;)

This parameter specifies the number of rows of input data that will not be used in the calculation of a moving average or rolling sum. This parameter also specifies the row containing column titles that are used as labels in the output. The two values should be separated by a comma.

You can specify any number of header rows; the default value is 0. The title row number should be less than or equal to the number of header rows. If the header rows value is specified, the default title row is the last row of the header rows. If no header is specified, the default is no title row.

Report name (Moving Average;)

This parameter is a title that will be placed in the output report, for example, *Four-Period Rolling Sum*. If this parameter is blank, there is no report title in Output 1.

Data columns (a; a,b;)

This parameter specifies the columns used to compute a moving average or rolling sum. At least one data column is required, but any number can be specified. For example, `a,b,c` computes moving averages or rolling sums on columns A, B, and C.

This field expects data columns of equal length. If two columns of different lengths are specified in this field, the transformer will calculate the moving average values for the shorter column (assuming that zeros were present in the cells in the short column adjacent to the longer column values). If you want to calculate two data columns of different lengths, connect another copy of the transformer to the input spreadsheet.

Other columns to keep (a; a,b;)

This parameter specifies the data columns to be transferred from the input region to the output region without having a moving average or rolling sum calculated on them. For example, `f,b` indicates that columns F and B are to be included in Output 1. Any number of columns can be specified. The default value is that no columns are to be copied from Input 1 to Output 1.

Calculate 'moving average' or 'rolling sum' (Mv; Rs; Moving Average; Rolling Sum)

This parameter specifies whether the moving average or rolling sum method should be computed on each of the data columns. Allowable choices are `moving average` or `rolling sum`, which can be abbreviated `mv` and `rs`. There is no default value.

Number of periods for 'Rolling sum' (; 3; 4; 12; 24;)

This parameter specifies the number of data points to be incorporated into each rolling sum. Any number of periods can be specified. The default value is 2.

Type of moving average (; Single; Double; Spencer15; Henderson5; Henderson9; Henderson13; Henderson23)

This parameter specifies the moving average method to be used. Allowable choices include: Single, Double, Spencer15, Henderson5, Henderson9, Henderson13, and Henderson23, abbreviated `s`, `d`, `s15`, `h5`, `h9`, `h13`, and `h23`, respectively. The default is Single.

Number of periods for 'single' moving average (; 3; 4; 5; 12; 24;)

This parameter specifies the number of data points to be incorporated into each moving average if the single moving average method is selected. Any number of periods can be specified. The default value is 3.

If this parameter is set to an even number, the transformer actually performs an *Nx2* double moving average where *N* is the specified number of periods. It does this because there is no middle row with which it can align the average when center alignment is requested.

Number of periods and number of terms for 'Double' moving average (; 3,3; 3,5; 3,9;)

This parameter specifies the number of data points and the number of terms to be incorporated into each moving average if the double moving average method is selected. Any number of periods and number of terms can be specified, provided the sum of the two numbers is even (a requirement of the double moving average method). If the double moving average method is selected, this parameter expects two whole numbers, separated by a comma. For example, to compute a 12 x 4 double moving average, type `12,4`. The default value is 3,3 (a 3 x 3 double moving average).

Align 'moving average' or 'rolling sum' result with (; Center; First; Last) elements of 'Data column'

This parameter specifies how the output values should be oriented. Allowable values include `first`, `center`, and `last` (`f`, `c`, and `l`). If this parameter is blank, the default value is `center`. See “Specifying Alignment of Output” on page 452 for an explanation of this parameter.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- Results (Output 1)
- Messages (Output 2)

When you click on any one of these choices, the data associated with that choice displays in the display area of the transformer.

Input Region Names

The Moving transformer has only one input region, Input 1 (Data). You can modify the input name to reflect the name of the corresponding data in the display area. Input 1 consists of data read from a Spreadsheet, Query, or SQL Entry tool, or copied directly into the transformer. Input 1 is treated as a series of columns that can each contain a time-series data set (Data columns). If extra columns are present, they are ignored when moving averages and rolling sums are calculated.

The input data can have any number of header rows. Column titles are read from the specified header row; if titles are not found, default titles are provided. All columns in Input 1 must have the same number of rows.

Whenever a variable in the data set has missing values, the transformer will pad a 0.0 for that missing value. This padding maintains the integrity of the data set in performing the moving average calculations.

Output Region Names

The Moving transformer generates two output regions; neither has a size limit.

Output 1 (Messages) contains the selected data columns and the moving average or rolling sum calculations specified, as well as the optional report title and moving average method. If moving average is selected, a ratio column is also presented to show the ratio of the moving average to the original data.

Output 2 (Messages) contains transformer run-time messages, a time stamp for documentation purposes, warnings, and error messages.

Examples

In this example, the following data is copied into the Moving transformer input region. The data is shown folded into two columns; the date is in column A, the volume is in column B.

Date	Volume	Date	Volume
1Q70	2,816.00	1Q73	38,984.00
2Q70	1,704.00	2Q73	22,223.00
3Q70	915.00	3Q73	28,908.00
4Q70	1,184.00	4Q73	16,985.00
1Q71	1,475.00	1Q74	18,063.00
2Q71	9,071.00	2Q74	17,977.00
3Q71	1,871.00	3Q74	550.00
4Q71	42,027.00	4Q74	2,723.00
1Q72	31,499.00	1Q75	10,529.00
2Q72	35,700.00	2Q75	7,224.00
3Q72	34,657.00	3Q75	48,076.00
4Q72	20,245.00	4Q75	31,729.00

In this example, the Moving transformer parameters are set as follows:

Number of header rows
1, 1

Report name
Moving Average Demo

Data columns
b

Other columns to keep
a

Calculate 'Moving Average' or 'rolling sum'
mv

Type of moving average
Single

Number of periods for 'Single' moving average
4

Align 'Moving Average' or 'Rolling Sum' result with
center

The report generated by the Moving transformer in Output 1 is as follows:

		Moving	
Date	Volume	Average	Ratio
1Q70	2,816.00		
2Q70	1,704.00		
3Q70	915.00	1,487.13	0.62
4Q70	1,184.00	2,240.38	0.53
1Q71	1,475.00	3,280.75	0.45
2Q71	9,071.00	8,505.63	1.07
3Q71	1,871.00	17,364.00	0.11
4Q71	42,027.00	24,445.63	1.72
1Q72	31,499.00	31,872.50	0.99
2Q72	35,700.00	33,248.00	1.07
3Q72	34,657.00	31,460.88	1.10
4Q72	20,245.00	30,711.88	0.66
1Q73	38,984.00	28,308.63	1.38
2Q73	22,223.00	27,182.50	0.82
3Q73	28,908.00	24,159.88	1.20

4Q73	16,985.00	21,014.00	0.81
1Q74	18,063.00	16,938.50	1.07
2Q74	17,977.00	11,611.00	1.55
3Q74	550.00	8,886.50	0.06
4Q74	2,723.00	6,600.63	0.41
1Q75	10,529.00	11,197.25	0.94
2Q75	7,224.00	20,763.75	0.35
3Q75	48,076.00		
4Q75	31,729.00		

The time-series data set Volume is data collected over a period of six years at quarterly intervals. The four-term single moving average is given in the Moving Average column, and the ratio to the moving average is given in the Ratio column. The first two and last two columns do not have moving average and ratio values; there is not sufficient information available to compute values for these four rows.

Because centered alignment was requested with a single moving average that had an even number of periods, the transformer actually used a 4 x 2 double moving average. Whenever the transformer encounters an even number of periods for a single moving average, it does an *n x 2* moving average, where *n* represents the specified number of periods. It does this because when there is an even number of terms in a single moving average, there is no center value with which to align the result.

Figure 51 shows this data in plot form.

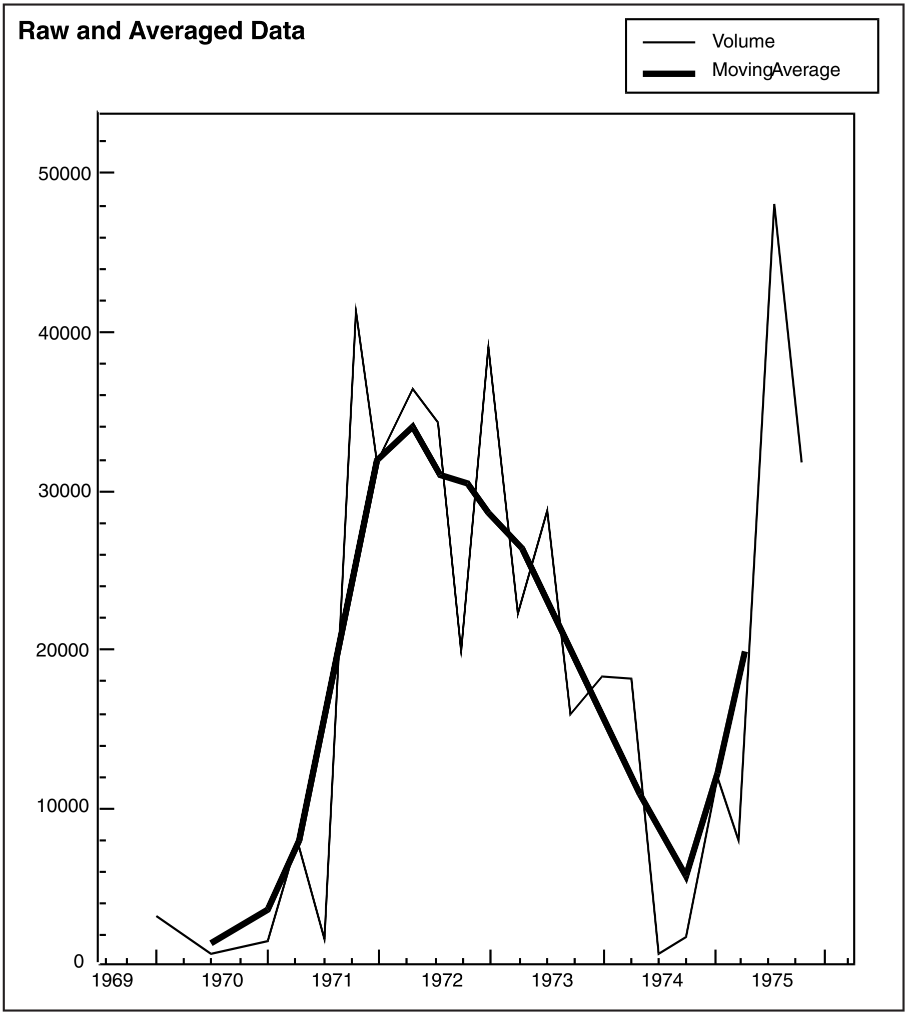


Figure 51. Smoothing example

The moving average result does not have values as large or as small as the original data, which is one effect of smoothing. Another effect of smoothing is evident in the period-to-period variations in the data. The original data has many adjacent peaks and valleys, whereas the moving average results in a continuous curve, free of period-to-period peaks or valleys.

This data set does not seem to have seasonal variations. If all of the first quarter moving average ratios are compared, no pattern is apparent nor is there any pattern for the second, third, and fourth quarters. If a particular quarter has moving average ratios that are consistently above or below 1.0, you can conclude that the data has seasonal variations.

Two different sources of variation are obvious in the data set. The first source of variability is the trend component. The trend is evident by plotting the moving average data. The other source of variability is randomness, as seen by the fact that the ratio values are quite different from 1.0, in most cases.

The trend component in this example is displayed as an S-curve. It is possible that the trend is actually a long-term cycle that would tend to repeat itself. The only way to confirm or disprove this theory would be to obtain more data. Since the length of one cycle seems to be five years, at least five more years of data would be required to reach a conclusion.

The Moving Average Ratio

The Moving transformer computes the moving average ratio, a value that compares the original data value to its corresponding moving average value. Specifically, the ratio indicates the proportion of the original value represented by the moving average value. For example, if the original value for a particular period is 915 and the moving average value is 1487, the original value is 0.62 of the moving average value.

Moving average ratios have two applications in data analysis. First, the moving average ratio value provides an indicator of the amount of noise in a data set. The farther the ratios are from 1, the more noise the data contains. For example, a data set that has many ratios that are greater than 1.5 or ratios that are less than 0.5 has much more randomness or noise than a data set that has all ratios less than 1.5 and greater than 0.5. The second application of moving average ratios is as indicators of the effects of seasonality. If a moving average is computed with a term that covers exactly one year, the moving average ratios provide rough estimates of seasonal effects. The quality of the seasonal indicators is improved if the time-series data set contains very little or no noise.

Rolling Sum Statistics

A rolling sum is the sum of the *n* most recent terms in the past. A rolling sum is useful for providing an up-to-date trend line for an indicator, for example, total yearly sales based on monthly sales values. The yearly value is computed for each month, providing a trend line that can be further analyzed. A 12-term rolling sum provides a yearly indicator based on monthly data, whereas a three-term rolling sum provides the same information based on quarterly data.

A second application of rolling sums is period aggregations. Suppose an analyst has data for monthly sales, but wants to see the data presented by quarter. The analyst computes a three-term rolling sum, and then removes all but every third data row. The remaining rows are the quarterly composites. Any aggregation can be computed in this manner. The Row Select transformer can be used to automate the row selection process that chooses the final aggregation values.

Moving Average Formulas

The definitions described in Table 64 apply to the formulas used in this section.

Table 64. Moving average formula symbol definitions

Symbol	Definition
A	Input data set.
A_i	A particular element of the data set A.
H_5	Resulting Henderson's 5-term moving average data set.
H_9	Resulting Henderson's 9-term moving average data set.
H_{13}	Resulting Henderson's 13-term moving average data set.
H_{23}	Resulting Henderson's 23-term moving average data set.
N	Number of observations in the data set A.
MD	Resulting double moving average data set.
MS	Resulting single moving average data set.
MS_i	A particular element of the single moving average data set.
p	Number of periods in the moving average.
R	Resulting moving average ratio data set.
S_{15}	Resulting Spencer's 15-term moving average data set.
t	Number of terms in the double moving average.
Z	Result of any one of the previous moving averages.

The single moving average is calculated by the formula:

$$MS_i = \frac{\displaystyle\sum_{j = i - [(p-1)/2]}^{i + [(p-1)/2]} A_j}{p}$$

The double moving average is calculated by the formula:

$$MS_i = \frac{\displaystyle\sum_{j = i - [(t-1)/2]}^{i + [(t-1)/2]} \left[\sum_{k = j - [(p-1)/2]}^{j + [(p-1)/2]} A_k \right]}{p \times t}$$

The Spencer's 15-point moving average is calculated by the formula:

$$S_{15i} = \left[\begin{array}{l} -0.009 \times A_{(i-7)} - 0.019 \times A_{(i-6)} - 0.016 \times A_{(i-5)} + \\ 0.009 \times A_{(i-4)} + 0.066 \times A_{(i-3)} + 0.144 \times A_{(i-2)} + \\ 0.209 \times A_{(i-1)} + 0.231 \times A_{i} + 0.209 \times A_{(i+1)} + \\ 0.144 \times A_{(i+2)} + 0.066 \times A_{(i+3)} + 0.009 \times A_{(i+4)} - \\ 0.016 \times A_{(i+5)} - 0.019 \times A_{(i+6)} - 0.009 \times A_{(i+7)} \end{array}\right]$$

The Henderson's 5-point moving average is calculated by the formula:

$$H_{5i} = \left[\begin{array}{l} -0.073 \times A_{(i-2)} + 0.294 \times A_{(i-1)} + 0.558 \times A_{i} + \\ 0.294 \times A_{(i+1)} - 0.073 \times A_{(i+2)} \end{array}\right]$$

$$H_{5i} = \left[\begin{array}{l} -0.073 \times A_{(i-2)} + 0.294 \times A_{(i-1)} + 0.558 \times A_{i} + \\ 0.294 \times A_{(i+1)} - 0.073 \times A_{(i+2)} \end{array}\right]$$

The Henderson's 9-point moving average is calculated by the formula:

$$H_{9i} = \left[\begin{array}{l} -0.041 \times A_{(i-4)} - 0.010 \times A_{(i-3)} + 0.119 \times A_{(i-2)} + \\ 0.267 \times A_{(i-1)} + 0.330 \times A_{i} + 0.267 \times A_{(i+1)} + \\ 0.119 \times A_{(i+2)} - 0.010 \times A_{(i+3)} - 0.041 \times A_{(i+4)} \end{array}\right]$$

The Henderson's 13-point moving average is calculated by the formula:

$$H_{13i} = \left[\begin{array}{l} -0.019 \times A_{(i-6)} - 0.028 \times A_{(i-5)} + 0.0 \times A_{(i-4)} + \\ 0.066 \times A_{(i-3)} + 0.147 \times A_{(i-2)} + 0.214 \times A_{(i-1)} + 0.240 \times A_{i} + \\ 0.214 \times A_{(i+1)} + 0.147 \times A_{(i+2)} + 0.066 \times A_{(i+3)} + \\ 0.0 \times A_{(i+4)} - 0.028 \times A_{(i+5)} - 0.019 \times A_{(i+6)} \end{array}\right]$$

The Henderson's 23-point moving average is calculated by the formula:

$$H_{23i} = \left[\begin{array}{l} -0.004 \times A_{(i-11)} - 0.011 \times A_{(i-10)} - 0.016 \times A_{(i-9)} \\ 0.015 \times A_{(i-8)} - 0.005 \times A_{(i-7)} + 0.013 \times A_{(i-6)} + \\ 0.039 \times A_{(i-5)} + 0.068 \times A_{(i-4)} + 0.097 \times A_{(i-3)} + \\ 0.122 \times A_{(i-2)} + 0.138 \times A_{(i-1)} + 0.148 \times A_{i} + \\ 0.138 \times A_{(i+1)} + 0.122 \times A_{(i+2)} + 0.097 \times A_{(i+3)} + \\ 0.068 \times A_{(i+4)} + 0.039 \times A_{(i+5)} + 0.013 \times A_{(i+6)} \\ 0.005 \times A_{(i+7)} - 0.015 \times A_{(i+8)} - 0.016 \times A_{(i+9)} \\ 0.011 \times A_{(i+10)} - 0.004 \times A_{(i+11)} \end{array}\right]$$

The moving average ratio is calculated by the following formula, assuming *Z* represents the result of the moving average for which you are computing the ratio:

$$R_i = \frac{A_i}{Z_i}$$

Rolling Sum Formulas

The definitions described in Table 65 apply to the equations used in this section.

The rolling sum is calculated by the formula:
Table 65. Rolling sum formula definitions

Symbol	Definition
A	Input data set.
A_i	A particular element of the data set A.
N	Number of observations in the data set A.
p	Number of periods in the rolling sum.
RS	Resulting rolling sum data set.
RS_i	A particular element of the rolling sum data set t.

$$RS_i = \sum_{j = i \pm p}^{i} A_j$$

Specifying Alignment of Output

Because a moving average or rolling sum value is a composite of several input data values, there will always be fewer output values than input values. The `Align moving average or rolling sum with` parameter lets you specify exactly where the output values are placed. The choices are center value, first value, and last value. `First` places the result values immediately after the header, with blank cells following the final output value. `Last` places the result values so that the final output value lies on the same row as the final input data value. `Center` evenly divides the extra space between the top and bottom of the output column, thus centering the moving average or rolling sum results.

The following examples illustrate the effects of each `Align moving average or rolling sum with` option on a single moving average (of length three).

The moving average ratio column depends on the alignment selected. In all cases, the ratio compares the raw data and the moving average value with which it is aligned. When the alignment is changed, each raw data value is compared to a different moving average value. For example, when `last` is chosen, the first ratio value of 1.5 is obtained by comparing the data value 12.0 to the moving average value 8.0 (12.0 / 8.0 = 1.5). If `center` is chosen instead, the first ratio value is obtained using the same moving average value (8.0), but now it is compared to a different data value (8.0). In the latter case, the ratio is 1.0 (8.0 / 8.0 = 1.0).

Align Moving Average Or Rolling Sum With		First
Data	Moving Average	Ratio
4.0	8.0	0.5
8.0	10.0	0.8
12.0		
10.0		
Align Moving Average Or Rolling Sum With		Center
Data	Moving Average	Ratio
4.0		
8.0	8.0	1.0
12.0	10.0	1.2
10.0		
Align Moving Average Or Rolling Sum With		Last
Data	Moving Average	Ratio
4.0		
8.0		
12.0	8.0	1.5
10.0	10.0	1.0

Regression

The Regression transformer performs multiple linear regression analysis. Multiple regression is a statistical procedure that uses values of one or more independent variables to estimate the values of a dependent variable. Some of the uses of this tool are:

- Summarizing variables
- Studying relationships among variables
- Examining deviations from those general relationships

Regression analysis is one of the most versatile data analysis techniques. It can be used with many different forms of data, including nominal, ordinal, interval, and ratio. It can be applied to answer a wide range of questions such as:

- What is the relationship between product pricing and sales?
- What kinds of advertising are most effective for increasing product share?

One strength of regression analysis is that its results are relatively easy to understand and explain to others, because they can be represented with charts and plots. For example, the relationship between two variables is much easier to grasp when it is presented in a plot. The ease of sending regression output to the Plot tool, or an Excel spreadsheet via Xlaunch, allows you to take full advantage of this strength.

Parameters

All of the parameters described in this section are displayed in the Transformer Controls window. Most of these parameters are also displayed in the Data Entry icon contained in the transformer's demonstration Capsule icon. However, to simplify use of the transformer for new users, some of the more advanced parameters might be omitted from the Data Entry icon; they can be accessed only in the Transformer Controls window.

The Regression Transformer Controls window contains 19 user parameters.

Number of header rows (; 1; 5;) row containing titles (,1; ,5;)

This parameter specifies the number of rows in the input data that are skipped and not used in calculating statistics. This parameter also specifies the row number containing column titles that are used as labels in the output. The two values should be separated by a comma. You can specify any number of header rows; the default value is 0.

Negative numbers are not valid values.

The title row number should be less than or equal to the number of header rows. If the header rows value is specified, the default title row is the last row of the header rows. If no header is specified, the default is no title row.

Report name (; Regression Analysis;)

This parameter is a title for the Output 2 and Output 3 reports, for example, *Regression tests*. If the `Report Name` parameter is blank, there is no title in Output 2 or Output 3.

Dependent variable (a; b;) independent variable columns (b; b,c,d;)

This parameter specifies the column used as the dependent variable and the columns used as the independent or predictor variables in the regression analysis. One dependent variable and one or more independent variables are required. The first column specified is the dependent variable; all other columns are independent variables. For example, if you specify `a,b,c`, the transformer uses columns B and C as predictors of the values in column A.

The Regression transformer cannot read dependent or independent variables that are beyond the 97th input column (column CS in Spreadsheet). If a variable is specified beyond that column, the transformer stops and issues an error message.

Independent variable columns to always include in the model (b; b,c;)

This parameter specifies data columns that should always be included as predictors in any regression analysis. There is no default value for this parameter, and specifying it is optional. However, if all columns are specified to be included in the Capsule icon's User Input Control window, the transformer will run the FullModel as the default and disregard other specified models.

Even if a column is specified in this parameter, that column might be excluded from the model because it has an unacceptable tolerance level. Forcing such a variable into a model would result in an inferior model.

Regression method (FullModel; Forward; Backward; Stepwise)

This parameter specifies how independent variables are included or excluded from the regression equation. Enter one of the four possible values: FullModel, Forward, Backward, and Stepwise. The default value is `Full Model`.

If you specify other than `FullModel` and the `Independent variable columns to always include in the model` parameter is set such that all columns are to be included, the transformer will run the `FullModel` and disregard all other specified models.

Remove outliers? (; y; n) normal deviate cutoff limit (; ,1.5; ,2.0;) [2.0=0.9775%]

This parameter selects whether rows with aberrant predicted dependent variable values are excluded from the analysis. It also allows you to set the Z-score value used to identify outliers. For example, if you specify `Yes,2`, any row having dependent values more than two Z-scores from the mean are excluded. The default values are `No` and 2.0.

F to enter, F to remove (3.9; 2.80;)

This parameter specifies the F-statistics required for a variable to be included or excluded from a regression equation. The default `F to enter` value is 3.9; the default `F to remove` value is 2.8. The forward regression method uses the `F to enter` value to select new variables to include in the regression equation. Alternatively, the backward method uses the `F to remove` value to remove variables from the equation. The stepwise method uses both values to add and delete variables from the equation. The full model method does not use this parameter.

Override default multicolinearity limit? (; y; n) Multicolinearity level (,0.01;) formula (1-R2)

This parameter specifies how a multicolinearity measure is calculated and used for removing variables from the analysis. The appropriate response for the first characteristic is `yes`, which indicates that multicolinearity should be calculated, or `no`, which indicates that multicolinearity should not be calculated. The appropriate value for the second characteristic depends upon the formula chosen for the multicolinearity formula. If the formula is *1-R2* (also referred to as tolerance), the level must be less than or equal to 1. If the formula is *1/(1-R2)* (also known as an inflation factor), the level must be greater than or equal to 1. If the parameter is blank, no multicolinearity measures are calculated. If the parameter contains only `yes`, the *1-R2* formula for multicolinearity and a tolerance level of 0.01 are used to eliminate variables from the analysis. If you choose to have it calculated, the multicolinearity measure is printed in Output 2.

Maximum number of steps (; 5; 20;)

This parameter specifies the maximum number of times variables will be added to or removed from the regression equation. The default value is 20 (for Stepwise method only).

Print 'ANOVA' at each step in 'Details' output? (; y; n)

This parameter specifies whether an ANOVA table is sent to Output 3 at each step of the analysis. The default response is *no*. The regression method should be set to Stepwise.

Print 'F-Value' table at each step in 'Details' output? (; y; n)

This parameter specifies whether you want a table of F-statistics printed to Output 3 at each step of the analysis. The default response is `no`. If you want a table of F-statistics printed to Output 3, the regression method should be set to Stepwise.

Print 'Regression coefficients' at each step in 'Details' output? (; y; n)

This parameter specifies whether a table of regression coefficients is printed to Output 3 at each step of the analysis. The default response is `no`. The regression method should be set to Stepwise.

Print 'Partial correlation coefficients' at each step in 'Details' output? (; y; n)

This parameter specifies whether a table of partial correlation coefficients is printed to Output 3 at each step of the analysis. The default response is no. If you want a table of F-statistics printed to Output 3, the regression method should be set to Stepwise.

Print all summary tables at each step in 'Details' output? (; y; n)

This parameter specifies whether all of the information provided by the four previous parameters is sent to Output 3. If you specify yes, all other print parameters (ANOVA, F-enter, F-remove, regression coefficients, and partial correlation coefficients) will be printed at every step of the analysis. The default response is no.

Calculate partial regression residuals? (; y; n)

This parameter specifies whether partial regression residuals are calculated and sent to Output 7. The default value is no. Calculation of regression residuals consumes additional time and workstation resources.

Warning limits: lower & upper Durbin-Watson, autocorrelation, multicolinearity, fraction of outliers

This parameter specifies whether a message is sent to Output 4 if any of these measures are violated. The first two values in this parameter represent the lowest and highest Durbin-Watson values accepted. If the analysis results in a Durbin-Watson measure outside this range, a warning is issued; the default values are 1.5 to 2.5. The third value is the highest acceptable autocorrelation value; its default value is 1.0. The fourth value is the highest acceptable multicolinearity value and its default value is 0.01. The fifth value is the highest acceptable value of the fraction of outliers measure. Regression keeps track of the proportion of rows that have predicted dependent variables that are outliers. If the fraction of rows with outliers surpasses the specified level, a warning is issued. The default value for Fraction of outliers is 0.001.

Maximum number of rows to output in 'Details', 'Residuals', and 'Partials' outputs. (; n; 999;)

This parameter indicates the maximum number of rows to output to Output 3, Output 5, and Output 7. The default value is n or no, which indicates no limit.

Print F to enter at each step in 'Details' output? (; y; n)

This parameter specifies whether the F-statistics for variables not in the equation are shown in the step-by-step output. Valid responses are yes and no; the default value is no. The F-statistics are printed if the chosen Regression method is Stepwise.

Other columns to keep (; a; a,b; a,b,c;)

This parameter specifies columns in Input 1 that should be copied to Output 5 without modification. There is no default value for this parameter.

Region Controls

The `Display Data For` field in the Transformer Controls window contains the following choices:

- Data (Input 1)
- Summary (Output 1)
- ANOVA (Output 2)
- Details (Output 3)
- Messages (Output 4)
- Residuals (Output 5)
- Correlation (Output 6)
- Partials (Output 7)

When you click on any one of these choices, the data associated with that choice displays in the display area of the transformer.

Input Region Names

The Regression transformer has only one input region, Input 1 (Data). You can modify the input name to reflect the name of the corresponding data in the display area. Input 1 consists of data read from a Spreadsheet, Query, or SQL Entry tool, or data copied directly into the transformer. Input 1 contains columns of statistical data (data columns) and other columns that can be moved without modification to an output region (Columns to Keep). If other columns are present, they are ignored. The input data can have any number of header rows. Column titles are read from the specified row. If titles are not found, default titles are provided.

Whenever a variable in the data set has missing values, the entire observation or row is excluded from all calculations. This is true even if the variables with the missing values are not used in the actual calculation or if the row or observation contains some partial data.

Consider the following example. A data set contains variables A and B. Variable A contains 10 observations and variable B contains 12. In this data set, the first 10 observations would be used in all calculations and the last two would be excluded, even in calculations where only variable B is specified.

Whenever the transformer encounters an observation with missing values, a message is displayed in the Important Message window. The message informs the user that the particular observation will be excluded from the calculation.

Output Region Names

The Regression transformer has seven output regions. Output 1 (Summary), the summary region, lists the regression parameters, summarizes the analysis results, and includes the following choices:

- Number of rows of data
- Dependent variable name and names of independent variables in the model
- Predictor variable selection and exclusion criteria
- Definition and treatment of outliers and multicolinearity
- Number of steps specified and completed
- R^2
- Adjusted R^2
- RMS (residual mean squared) error
- Estimates and standard errors of B
- Tolerance or inflation measure
- Beta weights
- t-statistic
- F-statistic and the probability of F

Output 2 (ANOVA) includes summary ANOVA statistics and regression results as follows:

- ANOVA table
- RMS error
- R^2
- Adjusted R^2
- Durbin-Watson statistic
- Autocorrelation statistic
- Variable means
- Estimates and standard errors of B
- Tolerance or inflation measure
- Beta weights
- t-statistic
- F-statistic and the probability of F

Output 3 (Details) includes statistics calculated at each step of the regression analysis as follows:

- Regression parameters
- A partial correlation matrix (optional)
- ANOVA table (optional)
- RMS error
- R^2

- Adjusted R^2
- Durbin-Watson statistic
- Autocorrelation statistic
- Variable means
- Estimates and standard errors of B
- Tolerance or inflation measure
- Beta weights
- t-statistic
- F-statistic and the probability of F

Output 4 (Messages) contains run-time and transformer messages, as follows:

- A timestamp for documentation purposes
- Warning messages
- Error messages

Output 5 (Residuals) contains a table of residual information. Each row has the following information:

- The observation number
- Actual value of the dependent variable
- Predicted value of the dependent variable
- Residual value of the dependent variable
- Normal deviation value (Z-scores) of the dependent variable
- Actual values of predictor variables

Output 6 (Correlation) contains a correlation matrix for the dependent and independent variables.

Output 7 (Partials) contains a table of dependent and independent partial residuals for every pair of dependent and independent variables.

Examples

A company that markets sparkling water would like a better understanding of the factors that affect sparkling water sales in the city that is the company's primary market. To identify the most important predictors of water sales, the company constructs a table containing sales volume and other information that might be related to sales volume. The resulting table, which is copied into Input 1, follows.

A	B	C	D	E	F	G	H
			# of		# Days		% of
Sales	# Pro-	Ave. $ per	Competitor's	# Rainy	100	% Stores	Promotion
Volume	motions	Promotion	Prom.	Days	degrees	Selling	$ on TV
25,254.00	7	2.65	36	3	2	78	0.78
25,373.00	7	2.65	38	2	2	78	0.78
28,664.00	7	2.65	38	2	1	79	0.70
21,998.00	8	2.83	38	1	0	79	0.70
20,975.00	7	2.65	36	0	0	78	0.70
20,485.00	4	2.00	35	0	0	78	0.60
13,782.00	1	1.00	32	1	0	76	0.60
13,890.00	0	0.00	28	0	1	74	0.60
23,977.00	10	3.16	37	3	1	76	0.60
19,015.00	2	1.41	38	2	2	77	0.60
19,047.00	2	1.41	38	2	1	78	0.60
22,207.00	8	2.83	38	1	0	79	0.60
22,994.00	5	2.24	39	3	1	79	0.60
26,140.00	8	2.83	42	2	1	80	0.60
29,195.00	6	2.45	43	3	2	81	0.60
23,215.00	9	3.00	42	1	1	81	0.48
21,038.00	7	2.65	43	1	0	81	0.60
21,504.00	3	1.73	44	1	1	80	0.60
15,806.00	5	2.24	44	0	1	79	0.48
14,153.00	3	1.73	44	1	0	80	0.48
22,972.00	14	3.74	44	2	1	79	0.60
17,898.00	5	2.24	45	3	2	78	0.48
19,231.00	8	2.83	46	2	1	78	0.48

The Transformer Controls window parameters are set as follows:

Number of header rows
1, 1

Report name
Regression Example

Regression method
FullModel

Remove outliers?
yes, 2.0

F to enter, F to remove
3.9, 2.0

Regression

Override default multicolinearity limit?

yes, 0.001, 1–R**2

Maximum number of steps

7

Print all summary tables at each in 'Details' output?

yes

Calculate partial regression residuals?

yes

Warning limits

1.5, 2.5, 0.5, 0.01, 0.10

After the transformer runs, the following report is in Output 3:

```
Project:                                                      Regression Example
Method:                                                       Full Model
Dep. Variable:                                                Sales Volume

ANALYSIS OF VARIANCE
Source              Degree     Sum of Squares      Mean Sum Sqr      F-Value     Prob>F
Regression          7          335,547,819.97      47,935,402.85     9.72        0.00
Residual            16         78,921,421.99       4,932,588.87
Total corrected     23         414,469,241.96

RMS Error           2,220.94
R**2                0.81
Durbin-Watson:      1.70
AutoCorrelation:    0.02

COEFFICIENT VARIABLES
Variable          Mean       Estimated-B   Std Error B   Tolrnce   Beta     t-calc.   F-calc.   Prb >abs(F)
Y-Intercept       1.00       (84,475.52)   31,303.98     1.00      0.00     (2.70)    7.28      0.02
# Promotions      6.08       168.17        608.62        0.05      0.13     0.28      0.08      0.79
Ave. $ per Prom.  2.34       2,563.72      2,860.70      0.04      0.49     0.90      0.80      0.38
# of Comp. Prom.  39.71      (497.61)      259.29        0.16      (0.53)   (1.92)    3.68      0.07
# Rainy Days      1.54       554.74        669.94        0.46      0.13     0.83      0.69      0.42
# Days>100 deg    0.92       2,184.30      989.88        0.43      0.37     2.21      4.87      0.04
% Stores Selling  78.50      1,418.65      481.69        0.33      0.56     2.95      8.67      0.01
% of Proms: TV    0.60       7,272.78      8,874.51      0.35      0.15     0.82      0.67      0.43
Dependent Var     21,345.79
```

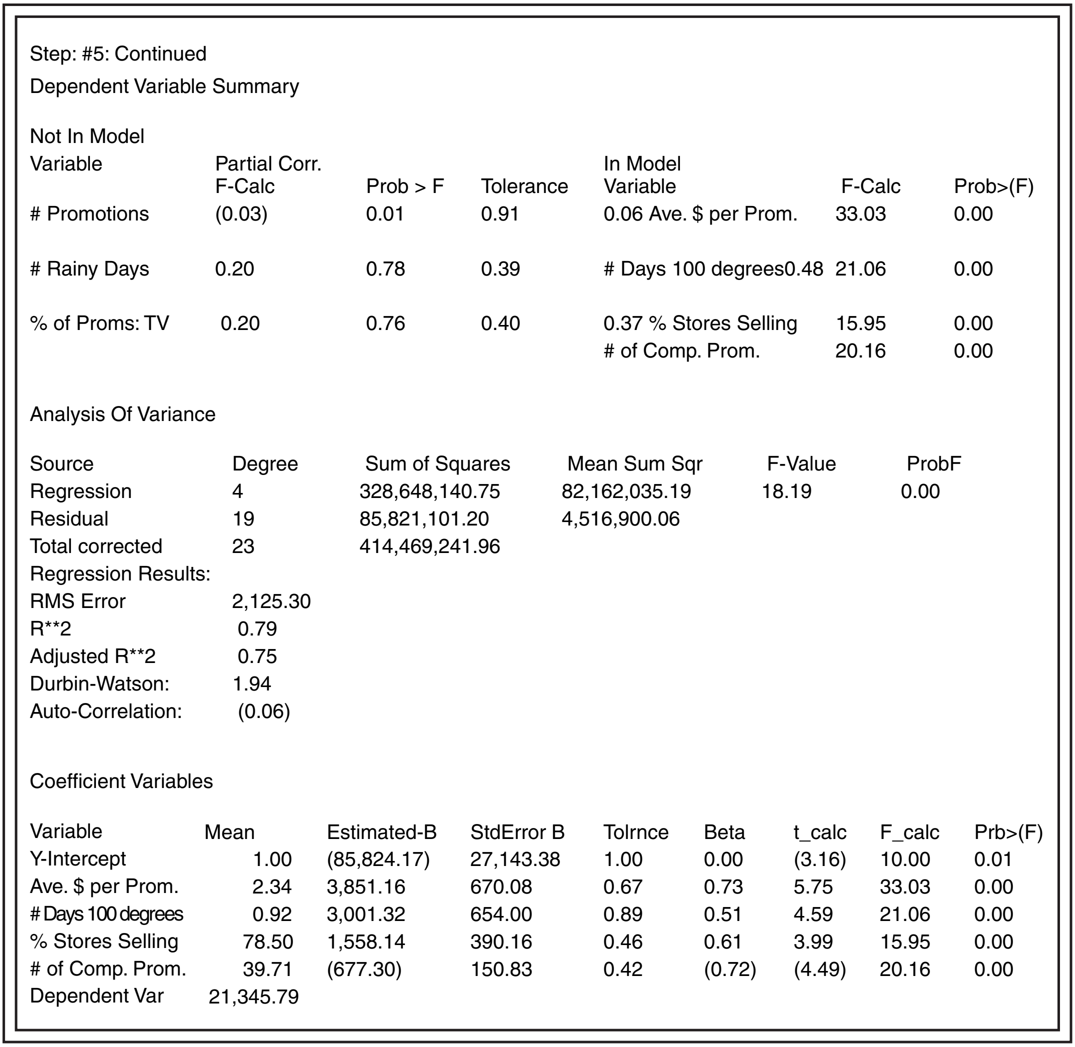
This regression model does a reasonable job of describing the variation in sales volume. The R2 of 0.81 indicates that the model accounts for 81% of the variation in the sample's sales volume. The adjusted R2 of 0.73 indicates that if this model was applied to the general population, it would determine only 73% of the variation in sales volume. The very low probability of the F-statistic in the ANOVA table confirms that the model describes a significant amount of variation in sparkling water sales.

Measures based on the residuals provide more evidence for accepting the model. The Durbin-Watson value of 1.7, though not far from the lower cutoff value of 1.5, proves that there is no strong correlation between sequence and residuals. The autocorrelation of 0.02 also proves that there is no obvious pattern among the residuals.

The Coefficient Variables table indicates the variables that are the most important predictors. The predictor, % Stores Selling, with a beta value of 0.56, is the most important determinant of sparkling water sales. Two other important predictors are # of Competitors' Promotions and Average $ Spent per Promotion. This table also presents some troubling information. First, one of the most important predictors, Average $ per Promotion, and some of the other predictors are not very significant. The high t probabilities indicate that the relationships detected by the model might be due to chance. More troublesome still, are the very low tolerances of two predictors indicating that there is a great deal of multicolinearity among them, which could reduce the effectiveness of the model.

Although the model seems to do an adequate job of predicting the values of the dependent variable in the sample, the violation of the multicolinearity assumption and the low significance of the predictors could make the model very fragile. These problems indicate that in other situations, the model might not be as effective.

To find a more robust model, the Regression transformer is run again. This time, the `Method` parameter is changed from `FullModel` to `Stepwise`. All of the other parameters remain the same as they were in the first analysis. Because a high degree of multicolinearity is an exclusion criterion for predictors, the stepwise method might avoid the multicolinearity problem in the previous model. The output from the last step of the second analysis follows.



```
Step: #5: Continued
Dependent Variable Summary

Not In Model
Variable        Partial Corr.                              In Model
                F-Calc          Prob > F    Tolerance      Variable                  F-Calc      Prob>(F)
# Promotions    (0.03)          0.01        0.91           0.06 Ave. $ per Prom.     33.03       0.00

# Rainy Days    0.20            0.78        0.39           # Days 100 degrees0.48    21.06       0.00

% of Proms: TV   0.20           0.76        0.40           0.37 % Stores Selling     15.95       0.00
                                                           # of Comp. Prom.          20.16       0.00


Analysis Of Variance

Source               Degree        Sum of Squares       Mean Sum Sqr        F-Value         ProbF
Regression           4            328,648,140.75       82,162,035.19        18.19           0.00
Residual             19           85,821,101.20        4,516,900.06
Total corrected      23           414,469,241.96
Regression Results:
RMS Error            2,125.30
R**2                  0.79
Adjusted R**2         0.75
Durbin-Watson:       1.94
Auto-Correlation:     (0.06)


Coefficient Variables

Variable             Mean        Estimated-B     StdError B    Tolrnce    Beta      t_calc    F_calc    Prb>(F)
Y-Intercept              1.00    (85,824.17)     27,143.38     1.00       0.00      (3.16)    10.00     0.01
Ave. $ per Prom.         2.34    3,851.16        670.08        0.67       0.73      5.75      33.03     0.00
# Days 100 degrees       0.92    3,001.32        654.00        0.89       0.51      4.59      21.06     0.00
% Stores Selling        78.50    1,558.14        390.16        0.46       0.61      3.99      15.95     0.00
# of Comp. Prom.        39.71    (677.30)        150.83        0.42       (0.72)    (4.49)    20.16     0.00
Dependent Var        21,345.79
```

The first thing to note is that at 0.79, the R^2 produced by this method is not as high as the 0.81 R^2 produced by the full model method. However, the adjusted R^2 of the new model is a little better than the adjusted R^2 produced by the earlier run, indicating that in other situations, the new model will be more effective at predicting sparkling water sales. The significant F-statistic in the ANOVA table shows that the model accounts for a significant amount of the variation in the dependent variable.

The Durbin-Watson and autocorrelation measures indicate that there is no strong pattern in the residuals. Figure 52 is a plot of the predicted, actual, and residual values of the dependent variable. It reinforces the fact that there is no obvious pattern in the residuals. This model thus satisfies the assumption that residuals are randomly distributed.

Another advantage of the new model is that there are fewer predictors, making the model easier to interpret. The most important predictor of sales is Average $ per Promotion. With a beta of 0.73, the amount spent per promotion is positively related to sparkling water sales. Close behind it in importance is # of Competitors Promotions. Not surprisingly, the amount spent per promotion is negatively related to the dependent variable. The other two predictors are # Days with Temperatures above 100 and %Stores Selling. Both are positively related to sparkling water sales, and both have relatively high beta values. The importance of these variables as predictors of sales is reinforced by the fact that they have the highest partial correlations with the dependent variable.

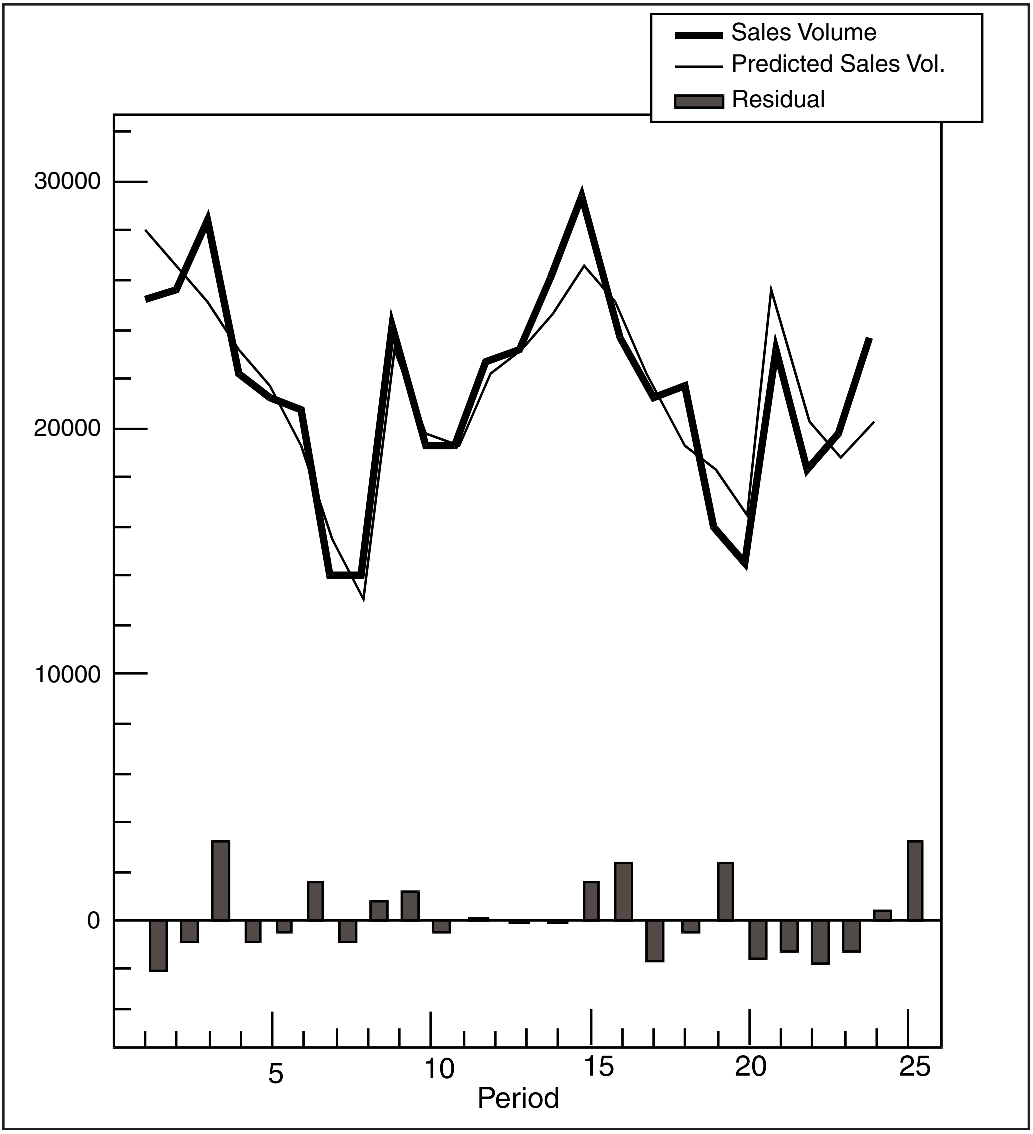


Figure 52. Predicted, actual, and residual values of the dependent variable

Two other characteristics found in the Coefficient Variables table reinforce the notion that this model is superior to the earlier model. First, all of the predictors have very significant F-statistics, indicating that the relationships detected by the model are unlikely to be due to chance. Second, none of the predictors have low tolerance levels. This situation indicates that there is little multicolinearity among the independent variables. Though the final model created by this analysis is free of multicolinearity, models created in earlier steps of the analysis had a great deal of it. In the second step of the analysis, the variable % of Promotion $ on TV entered the model. That variable had a tolerance of almost 0. After other predictors entered the model, the predictive power of that variable dropped significantly. Because of the low tolerance of this variable, it was removed from the model in the last step of the analysis.

Because it is relatively effective at predicting the dependent variable and it seems to meet most of the assumptions of the regression technique, the second regression model should be accepted. The information supplied by this analysis could help the sparkling water company formulate new tactics to increase sales. For example, it could try to determine why more expensive promotions seem to improve sales, then develop new ones that include characteristics of the more expensive promotions. In addition, the marketing people could develop a strategy aimed at limiting the effectiveness of their competition's promotions.

Multiple Regression Models

The Regression transformer performs as follows:

1. Starts with a table of input data and considers each column as a variable. One of the variables, the dependent variable, is estimated from the values of other variables. The other variables are known as independent or predictor variables. After reading the data, the transformer quantifies the relationships that all predictors have with the dependent variable. That information can be used to identify the most important or influential predictors.
2. Incorporates the independent variables into a formula or equation that can be used to predict values of the dependent variable. You can have the Regression transformer automatically select the independent variables that are the most powerful predictors of the dependent variable, thus keeping the model simple and effective.
3. Predicts values of the dependent variable. It then compares the predicted values with the actual values in the input table. The differences between actual and predicted values are called residuals. By examining residuals, the transformer can determine whether the regression equation does an adequate job of predicting values of the dependent variable.

The first two steps are the most important ones, if you are interested only in summarizing the data and looking at relationships among variables.

If you are also interested in developing a model that predicts values of the dependent variable, the third step is equally important.

Like other statistical techniques, regression analysis rests on several assumptions about the input data and the relationships of the dependent and independent variables. Throughout the analysis, the Regression transformer automatically checks the input data, intermediate results, and final results for conditions that violate the basic assumptions of the technique. If such conditions arise, the transformer informs you and sometimes takes steps to avoid the potential problem.

The Regression Equation

The mathematical model or equation produced is in the form of:

$$Y = c + b_1 \divideontimes X_1 + b_2 \divideontimes X_2 + b_3 \divideontimes X_3 + \ldots + b_n \divideontimes X_n$$

where:

Y is the dependent variable column.

c is a constant (the y-intercept).

X_1, X_2, X_3 are independent variable columns.

b_1, b_2, b_3 are the estimated equation coefficients.

For example, if the dependent variable is sales (Y) and the independent variables are price (X_1) and number of promotions (X_2), the following regression equation would give an estimate of sales:

```
Sales = 629.83+86.02 x price + 278.75 x #promotions
```

To estimate sales for a row that has a price of $1.98 and a number of promotions equal to four, insert those values into the regression equation:

```
1915.15 = 629.83 + 86.02 x 1.98 + 278.75 x 4
```

Regression Transformer Statistics and Concepts

A linear relationship between two variables indicates that as the values in one variable increase, the values of the second variable increase or decrease in a linear fashion. When the values of one variable are plotted against another, the data points should fall into a long, narrow band. If they form some other pattern, the relationship between the two variables is probably not linear. Standard regression analysis assumes that all predictors have a linear relationship with the dependent variable.

The slope in a regression model is the amount of change in the dependent variable that results from an increase in the independent variable; it measures the linear relationship between the dependent variable and predictor. For example, in the simple regression equation that follows, every time the value of X increases by 1, the value of Y increases by 5:

$$Y = 20 + 5 \divideontimes X$$

Thus, the slope is 5. In this example the slope is positive, so that as the value of the predictor increases, the value of the dependent variable increases. However, if the slope is negative, an increase in the predictor results in a decrease in the dependent variable. The slope for each of the independent variables is also known as a predictor coefficient.

The intercept or baseline is the value of the dependent variable when the independent variable is 0 or is extrapolated to 0. In the previous simple regression equation, the intercept is 20, indicating that when the value of X is 0, the value of Y is 20.

The F-statistic, calculated for each predictor, is a measure of the amount of variation in the dependent variable that is due to the individual predictor compared to the amount of variation due to other factors. Large F-statistics suggest a strong linear relationship between the predictor and the dependent variable.

The t-statistic, calculated for the intercept and each predictor in the regression equation, is a measure of whether these equation elements are different from 0. In regression analysis, the null hypothesis is that the predictor's slopes and the intercept are equal to 0. Thus, after it is converted to a probability, the t-statistic gives evidence for accepting or rejecting the null hypothesis. The Regression transformer automatically performs this conversion. If the probabilities of the t-statistics are high, the null hypothesis should be accepted. If the probabilities of the t-statistics are low, the null hypothesis should be rejected because the regression coefficients and the intercept found in the analysis probably did not occur due to chance.

A beta weight is a standardized version of the predictor coefficient in a regression equation. Because the original unit of measurement for each predictor varies a great deal, it is difficult to compare regression coefficients from one predictor to another. However, when the coefficients are standardized, predictors with the largest absolute beta weights have the largest influence on the value of the dependent variable.

The R^2 value or coefficient of determination is an estimate of the variation in the dependent variable that is accounted for by the regression equation. The R^2 value ranges from 0 to one; 0 indicates that none of the linear variation in the dependent variable is attributable to the independent variables. In contrast, an R^2 of one indicates that all of the linear variation in the dependent variable is accounted for by the predictors. Another way to view R^2 is that it is a measure of how well the regression equation fits the sample data.

The adjusted R^2 is a variation of R^2. The R^2 calculated for a sample often overstates the power of the regression equation in predicting values of the dependent variable in the population. This is especially a problem when there are many predictors in the model. The Regression transformer adjusts R^2 for the number of predictors in the model to create a more realistic measure for the entire population.

The residual mean squared (RMS) error, also known as the standard error of the estimate, is another method for measuring how well the regression equation fits the sample. It is based on the difference between the predicted and the actual values of the dependent variable. Models that closely fit the data have smaller RMS error values than models that do not closely fit the data.

A normal deviate or Z-score measures the deviation of the predicted value from the actual value. It is a test of the goodness of the prediction model. The formula for this calculation is:

```
z = Residual/RMS error
```

Conversely, to identify cases that are very different from the mean, cases that are two Z-scores above or below the group mean can be found.

Regression Methods

Out of any set of predictors, it is possible to create a large number of different regression equations. For example, from two independent variables, you could construct three regression equations. Two of the equations would have only one predictor; the third would have both predictors. The goal of multiple regression is to find the subset of variables that best predicts the values of the dependent variables.

The Regression transformer offers several methods for constructing regression equations:

- Full model
- Forward
- Backward
- Stepwise

Except for the full model method, the model-building methods provided in the Regression transformer are designed to efficiently identify the best subset of predictors; each method adds or removes variables one at a time. Adding or removing a variable to or from the model is referred to as a step. In these techniques, the partial correlation and the F-statistic are the measures used to determine the importance of each predictor. Independent variables with higher partial correlations and F-statistics are better predictors of the dependent variable.

Full Model Method

The full model method constructs a regression equation using all specified independent variables. Although it provides a good understanding of the effectiveness of an entire set of predictors, it does not try to identify the most useful independent variables.

Forward Method

The forward method adds independent variables to the equation according to their importance in predicting the dependent variable. Variables with the highest partial correlations and F-statistics are added first.

Backward Method

The backward method starts with all independent variables in the regression equation and sequentially removes them. In this case, variables with the lowest F-statistics are removed first.

Stepwise Method

The stepwise method is a combination of the forward and backward methods. The first two steps are identical to the forward method. However, after the second predictor is included, both variables are evaluated for exclusion. If one predictor has a low F-statistic, it is removed. After the third step, all independent variables not in the equation are evaluated. If the one with the highest partial correlation has a sufficiently high F-statistic, it is included. Each subsequent step that includes a variable is followed by a step in which variables are evaluated for exclusion.

Each of these methods has different strengths and weaknesses. Of the four, the stepwise method is most likely to determine the simplest and most effective regression equation. However, at each step it must calculate many different intermediate statistics, requiring more computation time and workstation resources than the other methods. In contrast, the full model method uses the least amount of computation time, but it does not attempt to reduce the number of predictors. As a result, unless you know a great deal about the predictors or have only a limited number of predictors, this method could result in an equation with an artificially high R^2. These three methods might not always yield the same equation. It is often best to try more than one method of equation construction.

Controlling the Analysis

The forward, backward, and stepwise methods of independent variable selection are all iterative processes, which complete a number of steps repeatedly until a particular criterion is satisfied. The number of iterations required by each equation construction method varies according to the number of independent variables and the predictive power of those variables. In addition to the limits imposed by the data, the Regression transformer allows you to specify criteria that ensure that an effective regression equation is efficiently created.

One method of optimizing the equation-building process is to specify the F-statistics at which variables can be considered for inclusion to, or exclusion from, regression equations. These criteria, specified in the `F to enter, F to remove` parameter (see “Parameters” on page 454), give the Regression transformer a framework with which it can limit the number of variables considered for inclusion or exclusion.

The function of these criteria depends on the method selected for building the regression equation. For example, in the forward method, only variables with F-statistics at or above the *F to enter* criterion are considered for inclusion in the regression equation. The analysis usually ends when all variables satisfying the *F to enter* criterion are in the regression equation. In the backward method, only variables with F-statistics less than the *F to remove* level are eliminated from the regression equation. This process stops when all variables in the regression equation have F-statistics above the *F to remove* criterion.

The stepwise method uses both criteria. At each entry step, all variables considered for inclusion must meet the *F to enter* value. At each elimination step, all variables considered for removal must have F-statistics which are less than the *F to remove* criterion. This process usually stops when all variables in the equation have F-statistics above the *F to remove* criterion and all variables not in the equation have F-statistics below the *F to enter* criterion. If you set the *F to remove* criterion above the value of the *F to enter* criterion, the stepwise method could enter and remove the same variables repeatedly. To avoid this kind of looping, make certain that the *F to remove* value is lower than the value of *F to enter*.

In addition to the indirect limitation of iterations, the Regression transformer offers a more direct method for limiting the number of iterations. The `Maximum number of steps` parameter (see “Parameters” on page 454) forces an end to the analysis when the specified number of inclusion and exclusion steps is surpassed. This method is particularly useful with the stepwise method, which might complete many iterations and consume a considerable amount of time while building an equation.

Ensuring the Integrity of the Analysis

Like other statistical techniques, regression analysis rests on a number of assumptions about the distribution and relationships of the dependent and independent variables. Violations of these assumptions reduces the reliability of the resulting model. The Regression transformer provides several methods for detecting violations of assumptions.

The first method is a check for multicolinearity. Multicolinearity occurs when independent variables are highly correlated with one another; if present, it can result in an unreliable analysis. The Regression transformer can calculate multicolinearity measures for all predictor variables, using one of two formulas. Both formulas use a form of the regression coefficient R2 in which the pertinent predictor becomes the dependent variable, and all of the other independent variables are the predictors. The first measure is called the tolerance level and is calculated with the formula `1-R2`. As the level of multicolinearity increases, the tolerance measure approaches 0. The second measure is called the inflation factor, and its formula is `1/(1-R2)`. As the level of multicolinearity increases, this measure increases. If requested, these measures are printed in the ANOVA table in Output 2.

When the Regression transformer calculates multicolinearity, it also uses the measure to eliminate variables from the analysis. You can specify the level of multicolinearity that is unacceptable. If any of the predictors violate that level, the transformer does not consider them for inclusion in the regression equation.

Besides specifying a multicolinearity elimination criterion, you can also specify a multicolinearity level at which the Regression transformer will send a warning to

Output 4. If any of the predictors in the final regression equation violates the specified multicolinearity level, the transformer issues a warning.

The Regression transformer also informs you if unacceptable Durbin-Watson values are detected. A Durbin-Watson value measures whether there is a correlation between residuals and their sequence numbers. If this value is small, it indicates a positive correlation between sequence and residuals; a large value indicates a negative correlation. It is often useful to specify a range of values that indicate little positive or negative correlation, so that the Regression transformer informs you if the model leaves some strong pattern in the residuals that should have been incorporated into the regression equation.

If requested, the Regression transformer also issues a warning if there is a high degree of autocorrelation in dependent variable residuals. The first order autocorrelation statistic supplied by the Regression transformer helps identify patterns in the residuals; it varies between -1 and 1, with values close to 0 indicating little pattern in the residuals. As the autocorrelation measure approaches 1 or -1, it indicates that each residual is strongly related to the residual before it. A significant pattern in the residuals indicates that the regression equation does not adequately explain variation in the dependent variable.

Finally, the Regression transformer informs you of a large volume of outliers in the predicted dependent variable. If the proportion of outliers exceeds the specified limit, the transformer sends a warning to Output 4. In addition to allowing you to decide what proportion of residuals should force a warning, the Regression transformer also allows you to define outliers and specify whether they should be included in the analysis. Outliers are defined in terms of normal deviates or Z-scores. For example, if the entry in the outlier definition parameter is 2, any observation more than two Z-scores from the mean is flagged as an outlier.

Calculating Partial Regression Residuals

In addition to this series of warning and exclusion functions, the Regression transformer provides another tool for detecting assumption violations. If requested, the transformer calculates partial residuals for dependent and independent variables. Pairs of partial residuals are created for each combination of dependent and independent variables. The residuals are calculated by subtracting estimated values of dependent and independent variables from their actual values. The estimated values for both variables are created by using all of the independent variables, except the one in the pair, as predictors in a regression equation.

This operation removes all of the linear trends in both dependent and independent variables that are explained by the remaining predictors. Because these other linear terms are removed from both variables, a plot of the dependent partial residuals by the independent partial residuals should show a linear trend. If such a trend is not evident, the relationship between the predictor and dependent variable might not be linear. Modifying the predictor in some way, such as taking the log of it or squaring it, might make the relationship linear and satisfy one of the main assumptions of multiple regression.

Because partial residual charts show the relationship of each of the predictors with the dependent variable while controlling for the other predictors, examining them can help identify points that are very different from the general relationship and can have undue influence on the regression coefficient. Thus, residual charts are also helpful for locating outliers that can make your regression equation less effective.

Specifying Columns

To specify the input columns, enter a list, a range of columns, or both, using either the letters or the numbers associated with the columns. For example, if the input data is in the first three columns of a Spreadsheet window, a column list specification would be `a,b,c` or `1,2,3`. A list of columns is simply a series of column letters or numbers separated by commas. The columns specified do not have to be contiguous. For example, if the columns specification is `a,c`, the transformer gathers data from the first and third columns of the input region.

The column parameters also accept ranges of columns. A range of columns consists of the number or letter associated with first data column, a colon, and the letter or number associated with the last data column. For example, if the transformer should use the first five columns of data, the columns specification would be `1:5` or `a:e`.

The column parameters also accept a combination of lists and ranges. For example, if the input data occurs in the first, second, and fourth through sixth columns, the parameter specification would be `a,b,d:f` or `1,2,4:6`.

Seasonality

The Seasonality transformer uses a form of the census X-11 decomposition method to identify seasonality effects and adjust a series based on those effects. The Seasonality transformer can be used for the following purposes:

- Identifying seasonal effects in a series
- Removing seasonality from a series to improve the accuracy of forecasts
- Identifying past seasonal effects and adding those effects to a projected series

This method automatically adjusts for differences in the number of trading days and eliminates outlier values.

Parameters

All of the parameters described in this section are displayed in the Transformer Controls window. Most of these parameters are also displayed in the Data Entry icon contained in the transformer's demonstration capsule application. However, to simplify use of the transformer for new users, some of the more advanced parameters might be omitted from the Data Entry icon; they can be accessed only in the Transformer Controls window.

Number of header rows for 'Input 1', 'Input 2', 'Input 3' (; 1,1,1; 2,1,1;)

This parameter specifies the number of input rows in each of the input regions that are not used in the seasonality analysis. A header row specification for each of the input regions is required. Any number of header rows can be specified; each value must be separated by a comma. The default value is 0,0,0.

Length of seasonality (1; 4; 6; 12; 52;) minimum number of seasons (, 2; 3; 5;)

This parameter specifies the number of periods in each season and the number of seasons of data required for the transformer to compute the seasonality statistics. The length of seasonality must be one or greater; minimum number of seasons must be two or greater. There is no default for these parameters.

Use 'Input 2' for computing seasonality factors if 'Input 1' is empty? (y; n)

This parameter specifies whether the transformer should use the data in Input 2 to derive the seasonality factors if Input 1 is empty. Valid responses are `Yes` and `No` (`y` and `n`). The default is `No`. If the response is `No` and Input 1 is empty, the transformer will issue an error message and stop. If the response is `Yes` and Input 1 is empty, seasonality factors are computed based on the data in Input 2; and those factors will be applied to the same data.

Column for date, volume for 'Input 1' (a,b; a,b,c;)

This parameter specifies the columns in which the transformer will find the dates and data to compute the seasonality factors. The two column specifications should be separated by a comma. If Input 1 is empty, you must still enter two column specifications in this parameter.

Column for date, volume {, other columns to keep} for 'Input 2' (a,b; d,c;)

This parameter specifies the columns in Input 2 that contain the date and data used to derive seasonality factors and to which the seasonality factors will be applied. This parameter can also optionally contain specifications for columns that will be moved unchanged from Input 2 to Output 1. All column specifications should be separated by commas.

Column for date, period # {, sequence #, trading days} for 'Input 3' (a,b; b,a,d,c;)

This parameter specifies the columns in Input 3 that contain the dates and period numbers used to verify the dates in the other input regions. This parameter can also contain the optional specifications for sequence # and trading days. Column descriptions should be separated by commas. If a sequence number column is omitted, the transformer creates one. If there is no entry for Trading days, the transformer assumes that each period has one trading day.

Seasonality

Date resolution (Day; Week; QuadWeek; Month; Quarter; BiMonth; EvenBiMonth; OddBiMonth; Year)

This parameter specifies the level at which the input data is gathered. The valid responses are Day, Week, QuadWeek, Month, Quarter, BiMonth, EvenBiMonth, OddBiMonth, and Year. The transformer uses this information to format the date values in the output regions and to check the input date values for missing data. None of these specifications can be abbreviated. The default value for this parameter is Day.

Average, ignore, or normalize trading days in seasonality factor (a; i; n)

This parameter specifies how the transformer should deal with trading days while computing seasonality factors. The valid responses are `Average`, `Ignore`, and `Normalize`. These can be abbreviated to `A`, `I`, and `N`, respectively. When `average` is specified, the transformer uses the average number of trading days in periods across seasons to compute an adjustment factor that reduces the effects of trading days. When `normalize` is specified, the transformer uses the specified number of trading days in the adjustment factor. When `ignore` is specified, no adjustment for trading days is made and the trading day adjustment factors displayed in the output will be set to 1. The default response for this parameter is `ignore`.

Average number of trading days in a 'Normalized' period (1; 5; 20; 21; 62; 250;)

This parameter specifies the number of trading days used to calculate the adjustment factor when you request a normalized trading day adjustment in the `Average, ignore, or normalize trading days` parameter. If this parameter is blank, a default value based on your choice of `Date resolution` is used. The default number of trading day values for each period resolution is: Day, 1; Week, 5; QuadWeek, 20; Month, 21.75; Even, Odd or standard BiMonth, 43.5; Quarter, 65.25; and Year, 261. Any value specified for this parameter is ignored if you specify that trading days are to be averaged or ignored.

Add, remove, or ignore seasonality in 'Output 1' data (a; r; i)

This parameter specifies whether and how the transformer should apply the derived seasonality factors to the volume data in Input 2 before it is sent to Output 1. The valid responses are add, remove, and ignore; they can be abbreviated to `a`, `r`, and `i`, respectively. If `add` is specified, the seasonality factors are used to seasonalize the data in Input 2. If `remove` is specified, the seasonality factors are used to deseasonalize the data in Input 2. If `ignore` is specified, the data in Input 2 is moved to Output 1 with no seasonality modification. The default response for this parameter is `ignore`.

Add, remove, or ignore trading days in 'Output 1' data (a; r; i)

This parameter specifies whether and how the transformer should apply the derived trading day adjustment factors to the volume data in Input 2 before it is sent to Output 2. The valid responses are `add`, `remove`, and `ignore`; they can be abbreviated to `a`, `r`, and `i`, respectively. If `add` is specified, the trading day adjustment factors are used to add a trading day effect to the data in Input 2. If remove is specified, the trading day adjustment factors are used to remove any trading day effect from the data in Input 2. If ignore is specified, the data in Input 2 is moved to Output 1 with no trading day modification. The default response for this parameter is ignore.

Output titles on deseasonalized data, on seasonality factors (y; n)

This parameter specifies whether titles are added to Output 1 (the deseasonalized data) and Output 2 (the seasonality factors). The valid responses for each specification are Yes and No (Y and N). The default is N,N. In addition to adding labels to date and raw input columns, these parameters also enable descriptive titles of modified data. For example, when the transformer removes seasonality from volume, the modified volume column in Output 1 is labeled *Deseasonalized volume*. When the transformer adds seasonality to volume, the label is *Reseasonalized volume*.

Region Controls

The Display Data For field in the Transformer Controls window contains the following choices:

- Base (Input 1)
- Data (Input 2)
- Periods (Input 3)
- Results (Output 1)
- Factors (Output 2)
- Messages (Output 3)

When you click on any one of these choices, the data associated with that choice displays in the display area of the transformer.

Input Region Names

The Seasonality transformer has three input regions. Input 1 (Base) is optional; it can contain the historical series from which seasonality factors will be calculated, if you want to derive seasonality factors from one set of historical data and use that information to deseasonalize (or seasonalize) another set of data.

If Input 1 is used, it must contain two columns. The first column should contain valid Meta5 dates; the second column must contain the data or volume on which the seasonality factors are built. If the region contains additional columns, they are ignored. Input 1 can contain any number of header rows.

If the Input 1 region is empty, Input 2 (Data) must contain the data from which the seasonality factors are calculated and to which the seasonality factors are applied as well as valid Meta5 dates. In addition, Input 2 can contain columns with other data that can be sent to an output region without any modification.

The Input 3 (Periods) region of the Seasonality transformer must contain one column with a valid Meta5 date and one column with a period number. A period number is a sequential integer that corresponds to each period in a season and repeats from season to season. For all examples, months of January could be assigned a period number of 1, months of February could be assigned 2, and so on. Input 3 can also contain two other optional columns of data. The first, a sequence number column, contains a sequence of integers representing the number of periods in the series, starting with 1. The second, a trading days column, contains the number of trading days in each period.

Whenever data is read into input regions, the transformer checks Input 1 and Input 2 against Input 3 period table for missing periods. If data is missing, the transformer inserts the appropriate periods with corresponding zero value observations to preserve the data integrity. Although Input 1 and Input 2 can have missing data, the Input 3 period table must be complete and cannot have missing data rows. The Input 3 period table is used to determine whether data is missing in Input 1 and Input 2. Also, Input 1 and Input 2 cannot contain extra observations that are not recorded in Input 3.

Output Region Names

There are three output regions for the Seasonality transformer; none is limited in size.

Output 1 (Results), the deseasonalized region, consists of columns containing the following information.

- Date for each period
- The sequence number of each date
- The period number for each date
- The number of trading days in each period
- The original volume (the data field from Input 2)
- The computed seasonality factor
- The deseasonalized or reseasonalized value of volume (depending upon request)
- The trading day adjustment factor
- The final (seasonalized or deseasonalized, if requested) trading day-adjusted volume
- The moving average with the length of seasonality of the raw data
- If requested, additional columns copied from Input 2

The deseasonalized or seasonalized volumes are displayed in the same column of the output region, but optional titles indicate the adjustments made to the volume values. The Seasonality transformer does not estimate the beginning and ending points of the moving average of the raw data.

Output 2 (Factors), the seasonality factor region, consists of columns containing the following information for each period:

- The date of the first period after the first full year of data.

  This field is displayed only as a label for each of the period numbers. As a result, the year reported in this field is not significant.
- The period number
- The final seasonality factor for each period
- The average number of trading days in each period
- For each season, the initial seasonal factor statistic for each period in the season
- For each season, the original data values for each period in the season
- For each season, the deseasonalized data values for each period in the season

The last three columns are repeated for each season of data. Thus, the next column of data would contain the initial seasonal factor statistic for each period in the second season.

Output 3 (Messages) contains:

- Transformer run-time messages
- Warnings
- Error message
- A timestamp

Seasonality Transformer Examples

The following sections demonstrate how to remove or add seasonality using the Seasonality transformer.

Removing Seasonality

A chocolate manufacturing and marketing company knows that its shipment volume is very seasonal. Shipments seem to have several peaks in every year, with the largest peak occurring before Christmas. The seasonality effects are so dramatic that it is difficult to observe the general sales trend. To determine how to improve the company's performance, the company administration wants to know the seasonally adjusted sales statistics for the past two years.

An analyst uses the Seasonality transformer to remove the effects of seasonality from the last three and one-half years of monthly sales volumes. That information forms the content of the Input 2 region:

Meta5 Date	Ordered Cases	Meta5 Date	Ordered Cases

January, 1986	11,995	October, 1987	5,513
February, 1986	8,050	Ordered Cases	13,984
March, 1986	5,809	December, 1987	11,051
April, 1986	9,198	January, 1988	8,935
May, 1986	11,034	February, 1988	10,379
June, 1986	5,480	March, 1988	12,307
July, 1986	5,636	April, 1988	16,688
August, 1986	2,343	May, 1988	15,873
September, 1986	10,870	June, 1988	6,636
October, 1986	4,091	July, 1988	7,786
November, 1986	11,466	August, 1988	6,374
December, 1986	10,960	September, 1988	11,733
January, 1987	13,045	October, 1988	8,047
February, 1987	9,473	November, 1988	11,763
March, 1987	8,645	December, 1988	7,192
April, 1987	11,946	January, 1989	6,419
May, 1987	12,835	February, 1989	6,080
June, 1987	5,743	March, 1989	9,677
July, 1987	8,071	April, 1989	7,456
August, 1987	3,115	May, 1989	17,184
September, 1987	12,499	June, 1989	18,850

In this case, Input 1 was left blank, so the Seasonality transformer computes seasonality factors based on the data previously shown and then uses those factors to remove the seasonal effect from the same series. A portion of the period table that corresponds to the preceding data table is as follows:

Meta5 Date	Period Number	Sequence Number
January, 1986	1	1
February, 1986	2	2
March, 1986	3	3
April, 1986	4	4
May, 1986	5	5
June, 1986	6	6
.	.	.

.	.	.
January, 1989	1	37
February, 1989	2	38
March, 1989	3	39
April, 1989	4	40
May, 1989	5	41
June, 1989	6	42

The Seasonality transformer parameters for this analysis are set as follows:

Number of header rows
1, 1, 1

Length of seasonality, minimum number of seasons
12, 2

Use 'Input 2' for computing seasonality factors if 'Input 1' is empty?
yes

Column for date, volume for 'Input 1'
a, b

Column for date, volume for 'Input 2'
a, b

Column for date, period # for 'Input 3'
a, b, c

Date resolution
Month

Average, ignore, or normalize trading days in seasonality factor
Ignore

Add, remove, or ignore seasonality in 'Output 1' data
remove

Add, remove, or ignore trading days in 'Output 1' data
Ignore

Output titles on deseasonalized data, on seasonality factors
yes, yes

After the transformer runs, the following information is displayed in the Output 1.

```
                     Seq    Period    Trad    Original    Seas     De-Seas      Moving
Date                 #      #         Days    Volume      Factor   Volume       Average
                                                                                Volume
January, 1986         1       1        1      11,995      1.14     10,502.46
February, 1986        2       2        1       8,050      1.00      8,040.08
March, 1986           3       3        1       5,809      1.04      5,604.84
April, 1986           4       4        1       9,198      1.40      6,554.43
May, 1986             5       5        1      11,034      1.40      7,881.33
June, 1986            6       6        1       5,480      0.61      8,999.17
July, 1986            7       7        1       5,636      0.76      7,391.38      8,121.42
August, 1986          8       8        1       2,343      0.42      5,625.82      8,224.46
September, 1986       9       9        1      10,870      1.26      8,608.42      8,401.92
October, 1986        10      10        1       4,091      0.63      6,476.65      8,634.58
November, 1986       11      11        1      11,466      1.32      8,667.70      8,824.13
December, 1986       12      12        1      10,960      1.01     10,832.79      8,910.13
January, 1987        13       1        1      13,045      1.14     11,421.81      9,022.54
February, 1987       14       2        1       9,473      1.00      9,461.33      9,156.17
March, 1987          15       3        1       8,645      1.04      8,341.17      9,256.21
April, 1987          16       4        1      11,946      1.40      8,512.64      9,383.33
May, 1987            17       5        1      12,835      1.40      9,167.74      9,547.50
June, 1987           18       6        1       5,743      0.61      9,431.06      9,656.21
July, 1987           19       7        1       8,071      0.76     10,584.77      9,488.75
August, 1987         20       8        1       3,115      0.42      7,479.48      9,355.25
September, 1987      21       9        1      12,499      1.26      9,898.50      9,545.58
October, 1987        22      10        1       5,513      0.63      8,727.88      9,895.75
November, 1987       23      11        1      13,984      1.32     10,571.17     10,219.92
December, 1987       24      12        1      11,051      1.01     10,922.73     10,383.71
January, 1988        25       1        1       8,935      1.14      7,823.21     10,409.04
February, 1988       26       2        1      10,379      1.00     10,366.21     10,532.96
March, 1988          27       3        1      12,307      1.04     11,874.47     10,636.83
April, 1988          28       4        1      16,688      1.40     11,891.75     10,710.50
May, 1988            29       5        1      15,873      1.40     11,337.71     10,723.54
June, 1988           30       6        1       6,636      0.61     10,897.54     10,470.21
July, 1988           31       7        1       7,786      0.76     10,211.01     10,204.58
August, 1988         32       8        1       6,374      0.42     15,304.71      9,920.63
September, 1988      33       9        1      11,733      1.26      9,291.87      9,631.92
October, 1988        34      10        1       8,047      0.63     12,739.57      9,137.67
November, 1988       35      11        1      11,763      1.32      8,892.21      8,807.63
December, 1988       36      12        1       7,192      1.01      7,108.52      9,371.17
January, 1989        37       1        1       6,419      1.14      5,620.28
February, 1989       38       2        1       6,080      1.00      6,072.51
March, 1989          39       3        1       9,677      1.04      9,336.90
April, 1989          40       4        1       7,456      1.40      5,313.09
May, 1989            41       5        1      17,184      1.40     12,274.13
June, 1989           42       6        1      18,850      0.61     30,955.18
```

The columns containing the trading day factors and the final volume have been omitted from the preceding data set. Because the trading days for all of the periods are equal to 1, the trading day factors are also all 1. This output is the default when the number of trading days is not included in the period table or trading day effects are ignored in the output. As a result, the deseasonalized volume is equal to the final volume, because, when requested, the final volume is deseasonalized and adjusted for trading day. Because trading day effects are ignored in the output, the volumes are equal.

Because this type of information is much easier to interpret when it is plotted, the plot shown in Figure 53 was created. In that plot, it is apparent that the final volume does not have the extreme high and low fluctuations of the original volume.

Although there is still some variation in the final volume from month to month, it seems as though chocolate shipments were on a general upward trend at the beginning and middle parts of the series. However, that trend seems to have been reversed near the end of the series, starting with November, 1988. For example, in November 1987, there were 10,571 seasonally adjusted cases of chocolate shipped, but a year later there were only 8,892 seasonally adjusted cases of chocolate shipped.

Finally, although the last two months of the original shipment volume show a dramatic increase in chocolate shipments, the seasonally adjusted volume is even more exaggerated. Because of the seasonality methodology, the first and last several periods of seasonally adjusted volume should be interpreted with care.

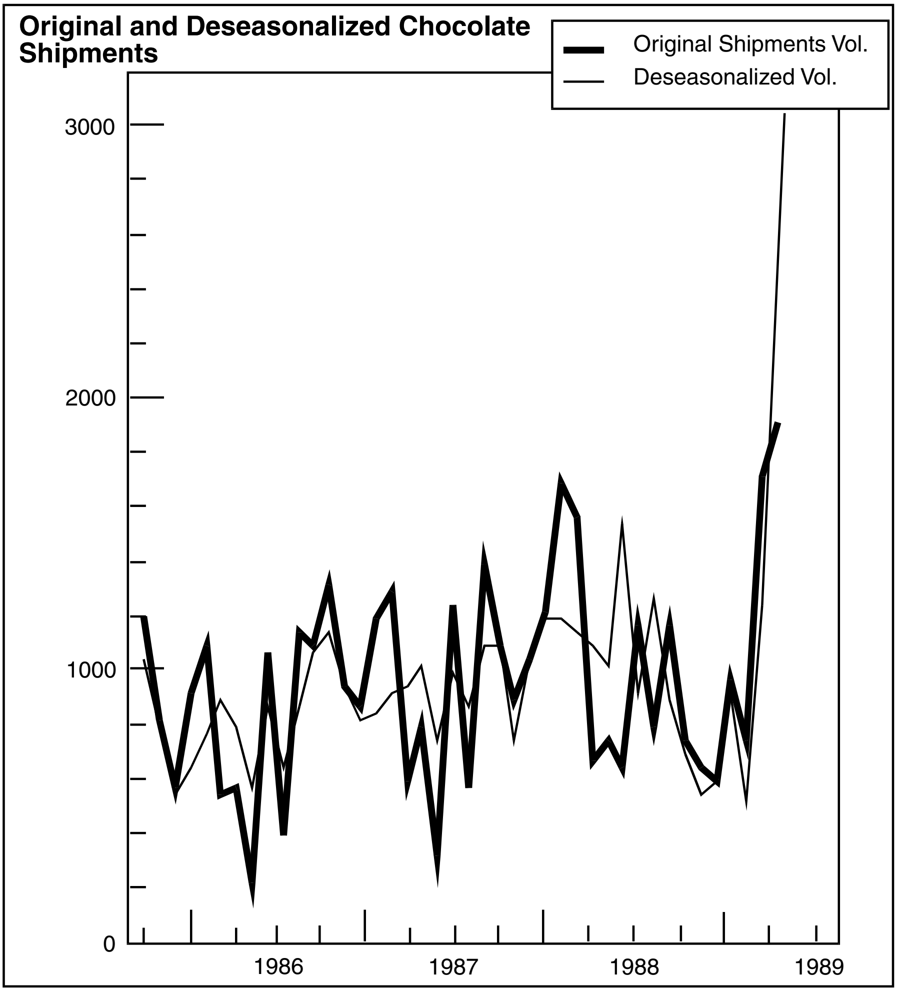
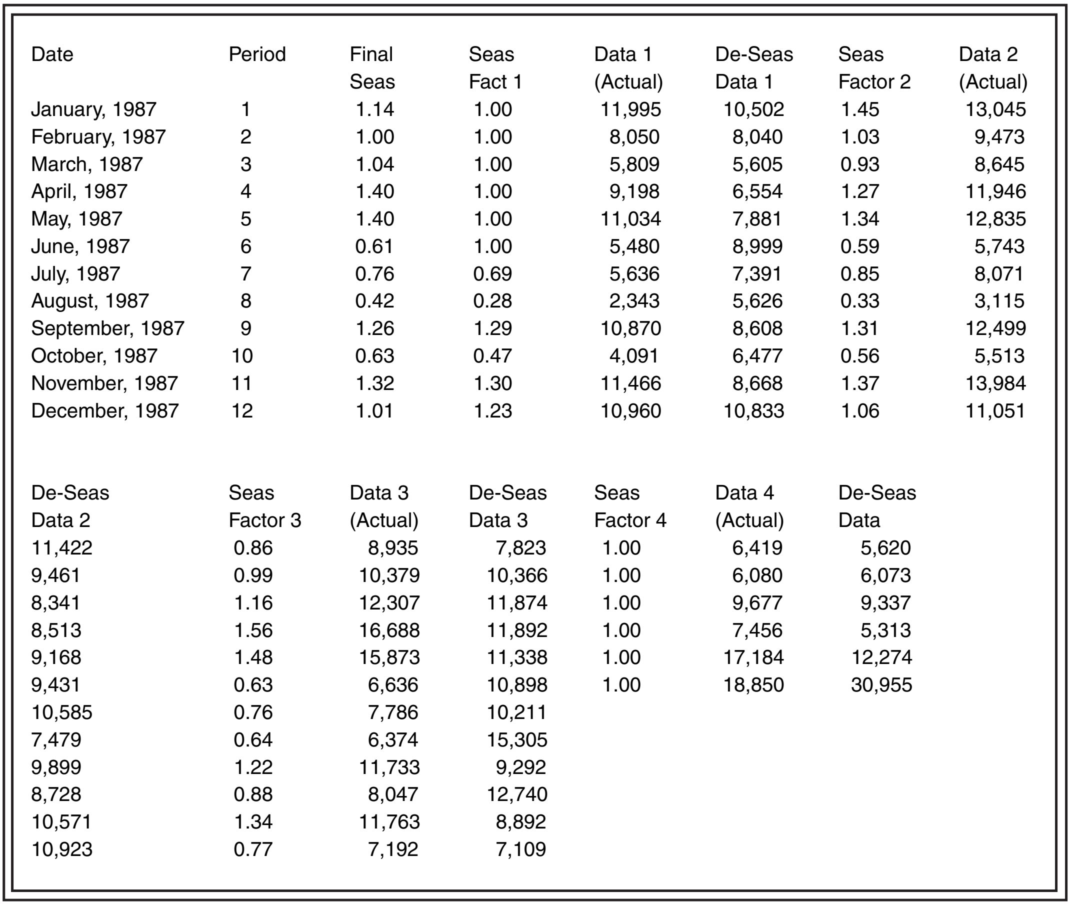


Figure 53. Original and deseasonalized data comparison

After the transformer runs, the following information was sent to Output 2. (The output is wrapped to fit the format of this book.)



```
Date               Period   Final    Seas     Data 1    De-Seas   Seas      Data 2
                            Seas     Fact 1   (Actual)  Data 1    Factor 2  (Actual)
January, 1987        1      1.14     1.00     11,995    10,502    1.45      13,045
February, 1987       2      1.00     1.00      8,050     8,040    1.03       9,473
March, 1987          3      1.04     1.00      5,809     5,605    0.93       8,645
April, 1987          4      1.40     1.00      9,198     6,554    1.27      11,946
May, 1987            5      1.40     1.00     11,034     7,881    1.34      12,835
June, 1987           6      0.61     1.00      5,480     8,999    0.59       5,743
July, 1987           7      0.76     0.69      5,636     7,391    0.85       8,071
August, 1987         8      0.42     0.28      2,343     5,626    0.33       3,115
September, 1987      9      1.26     1.29     10,870     8,608    1.31      12,499
October, 1987       10      0.63     0.47      4,091     6,477    0.56       5,513
November, 1987      11      1.32     1.30     11,466     8,668    1.37      13,984
December, 1987      12      1.01     1.23     10,960    10,833    1.06      11,051


De-Seas            Seas     Data 3   De-Seas  Seas      Data 4    De-Seas
Data 2             Factor 3 (Actual) Data 3   Factor 4  (Actual)  Data
11,422              0.86     8,935    7,823   1.00       6,419     5,620
9,461               0.99    10,379   10,366   1.00       6,080     6,073
8,341               1.16    12,307   11,874   1.00       9,677     9,337
8,513               1.56    16,688   11,892   1.00       7,456     5,313
9,168               1.48    15,873   11,338   1.00      17,184    12,274
9,431               0.63     6,636   10,898   1.00      18,850    30,955
10,585              0.76     7,786   10,211
7,479               0.64     6,374   15,305
9,899               1.22    11,733    9,292
8,728               0.88     8,047   12,740
10,571              1.34    11,763    8,892
10,923              0.77     7,192    7,109
```

Figure 54. Seasonality transformer output

From this output region, the months that have higher than average chocolate shipments, as well as the months that have lower than average shipments, are apparent. From the final seasonality factor column, it is obvious that January, April, May, September, and November have higher than average shipments, and June, July, August, and October have lower than average shipments. Also, this table shows that from year to year there is a great deal of variability in shipments in certain months. For example, the final seasonality factor for December is 1.01, which is the combination of the first December seasonality factor of 1.23, a second factor of 1.06, and a final factor of 0.77.

Adding Seasonality

A financial services company has been having a difficult time over the past few years. Revenue dropped slightly between 1986 and 1987, increased only by about five percent between 1987 and 1988, and is projected to rise by about six percent this year. The company executives believe that business is about to pick up and have projected that revenues will increase by 15 percent next year. This rate of growth should translate into average revenues of $287,554 for each month.

Corporate revenues are very seasonal, with increases coinciding with the beginning of new calendar quarters. To develop a spending plan for the coming year, the finance department must know the seasonalized revenues. The past three full years of monthly revenues are used to develop seasonality factors for seasonalizing next year's monthly revenue estimates. Although there is a quarterly pattern to the data, the intensity of the spikes varies a great deal. Consequently, the length of seasonality is specified as 12. The data provided for Input 2 of the Seasonality transformer is as follows:

Date	$ Sales	Date	$ Sales
January, 1986	254,071	July, 1987	256,092
February, 1986	222,596	August, 1987	207,316
March, 1986	212,391	September, 1987	204,198
April, 1986	257,769	October, 1987	260,347
May, 1986	208,493	November, 1987	207,977
June, 1986	205,869	December, 1987	201,229
July, 1986	255,709	January, 1988	263,931
August, 1986	209,123	February, 1988	222,155
September, 1986	204,809	March, 1988	220,401
October, 1986	251,164	April, 1988	284,139
November, 1986	222,124	May, 1988	233,144
December, 1986	208,347	June, 1988	225,476
January, 1987	254,362	July, 1988	288,254
February, 1987	222,784	August, 1988	208,217
March, 1987	212,517	September, 1988	204,499
April, 1987	257,887	October, 1988	261,639
May, 1987	208,592	November, 1988	209,582
June, 1987	205,959	December, 1988	203,283

The average monthly revenue for 1990 is included in Input 1:

Date	Period	$ Sales
January, 1990	1	287,554
February, 1990	2	287,554
March, 1990	3	287,554
April, 1990	4	287,554
May, 1990	5	287,554
June, 1990	6	287,554
July, 1990	7	287,554
August, 1990	8	287,554
September, 1990	9	287,554
October, 1990	10	287,554
November, 1990	11	287,554
December, 1990	12	287,554

A portion of the date table that forms Input 3 is shown in the following example:

Meta5 Date	Period Number	Sequence Number
January, 1986	1	1
February, 1986	2	2
March, 1986	3	3
April, 1986	4	4
May, 1986	5	5
June, 1986	6	6
.	.	.
.	.	.
July, 1990	7	55
August, 1990	8	56
September, 1990	9	57
October, 1990	10	58
November, 1990	11	59
December, 1990	12	60

The Seasonality transformer parameters for this analysis are set as follows:

Number of header rows

1, 1, 1

Seasonality

Length of seasonality, minimum number of seasons
12, 2

Use 'Input 2' for computing seasonality factors if 'Input 1' is empty?
no

Column for date, volume for 'Input 1'
a, b

Column for date, volume for 'Input 2'
a, c

Column for date, period # for 'Input 3'
a, b, c

Date resolution
Month

Average, ignore, or normalize trading days in seasonality factor
Ignore

Add, remove, or ignore seasonality in 'Output 1' data
add

Add, remove, or ignore trading days in 'Output 1' data
Ignore

Output titles on deseasonalized data, on seasonality factors
yes, yes

After the transformer runs, the following information is in Output 1:

Meta5 Date	Seq	Period	Trad	Original	Seasonality	Re-Seas
	#	#	Days	$Volume	Factor	$Volume
January, 1990	49	1	1	287,554	1.12	323,082
February, 1990	50	2	1	287,554	0.97	280,347
March, 1990	51	3	1	287,554	0.94	269,141
April, 1990	52	4	1	287,554	1.17	336,456
May, 1990	53	5	1	287,554	0.95	274,211
June, 1990	54	6	1	287,554	0.93	268,299
July, 1990	55	7	1	287,554	1.13	325,252
August, 1990	56	8	1	287,554	0.92	264,271
September, 1990	57	9	1	287,554	0.90	259,232
October, 1990	58	10	1	287,554	1.12	323,158
November, 1990	59	11	1	287,554	0.94	270,572
December, 1990	60	12	1	287,554	0.89	256,627

The plot shown in Figure 55 was created with the reseasonalized data.

This plot and the output previously shown shows the importance of adjustment for seasonality. Although the average monthly income of the corporation is forecast to be $287,554, there is a lot of seasonal variation around that mean. At $256,627, December has the lowest anticipated revenue. At $336,456, April has the highest revenue. Although these are still estimates, the seasonalized revenue projections put the finance department in a much better position to create a spending plan in which spending coincides with revenue.

Because of the close similarity to the output shown in the first example, the content of Output 2 is not shown here.

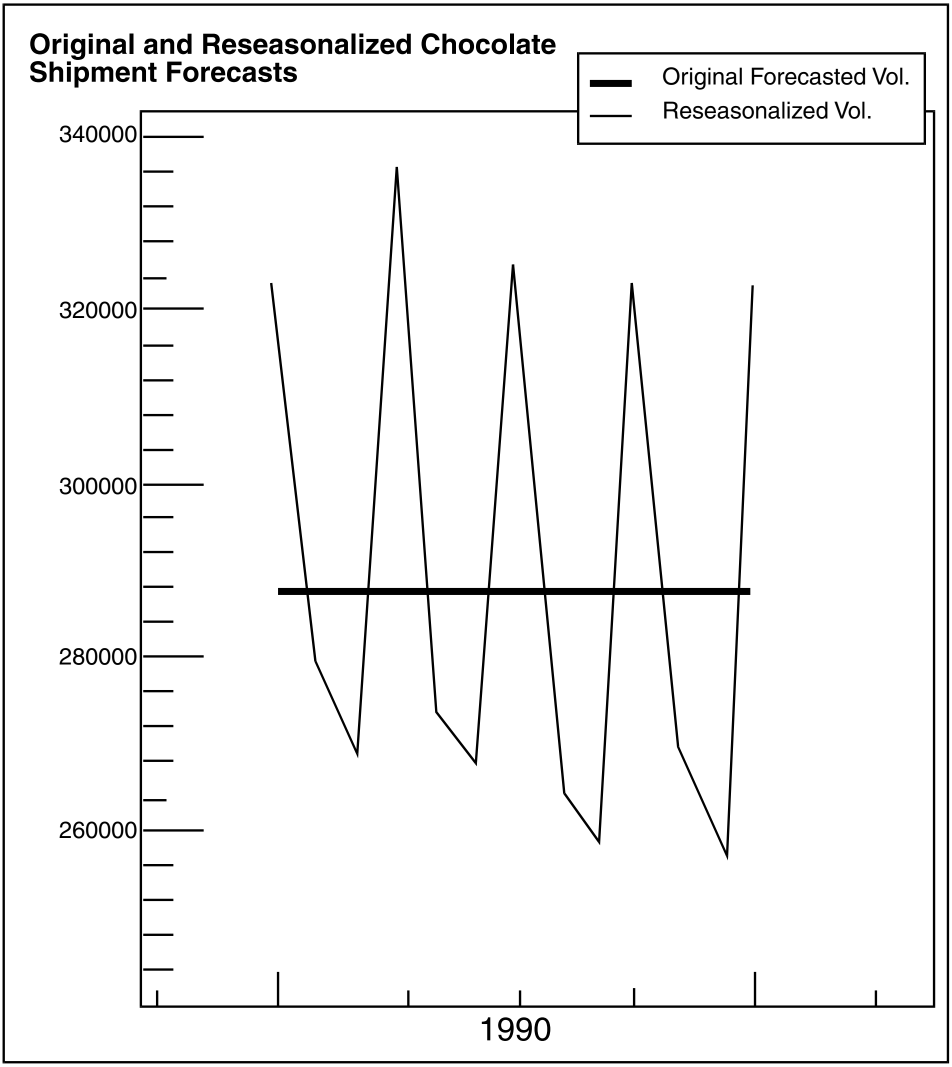


Figure 55. Original and reseasonalized data comparison

Identifying Seasonality Methods

The Seasonality transformer identifies the seasonal component in a series. After the seasonal component is identified, its effects can be removed from the series. A series that has had the seasonal component removed is called a deseasonalized series. Deseasonalization is important because the seasonal component in a series will confuse most of the techniques that identify and project a series' trend.

In addition to removing the effects of seasonality from a series, the Seasonality transformer can also add the effects of seasonality back into a series. This function is important because many of the techniques used to project a trend work best with deseasonalized data. Consequently, the resulting projections are deseasonalized. Adding the seasonal component to the projected series allows direct comparisons between the raw series and projected values.

Trading Days

The first operation performed in the seasonality analysis is adjustment for the number of trading days in each period. This step helps eliminate the number of trading days in a period as a cause of change in a series.

If requested, the transformer uses one of two methods to equalize the number of trading days in each period. In both methods, the observed volume is multiplied by a trading day adjustment factor. In the first method, which uses a normalized period, the adjustment factor is the number of days that should be in all trading periods divided by the number of days observed in the trading period. In the second method, which uses an average period length, the adjustment factor is the average number of trading days in each type period divided by the actual number of days in the period. The main difference in these methods is that normalizing adjusts for differences across periods in a year, whereas averaging adjusts for differences in the same period from year to year.

Seasonality Statistics

The Seasonality transformer uses a form of the census X-11 method. This method has several steps that isolate the seasonal component, adjust for differences in trading days, and eliminate outliers. These steps are as follows:

1. The transformer adjusts the series for the number of trading days (if requested) in each period as described in the preceding section.
2. After the data has been adjusted for trading days, the transformer calculates seasonality indicators. The first step in this process is to calculate a single moving average for the number of periods specified in each season; the result of this operation is the average of one entire season.
3. The volume for each period is divided by the moving average, yielding an index that indicates how much a given period's volume is above or below average.
4. The resulting indexes are lined up across seasons, and moving averages of the indexes are calculated for each period across seasons. For example, if a seasonality of 12 months is specified in analyzing data accumulated from 1984 through 1988, the transformer would calculate the moving average of the indexes for January 1984, January 1985, January 1986, January 1987, and January 1988.
5. The Seasonality transformer calculates the average for the months of February from 1984 through 1987, and so on. This operation provides the average seasonality index for each period across seasons.

6. The type of moving average used depends upon the amount of data that is available. If five or more seasons of data are available, a 3 x 3 moving average is calculated, if there are three to four seasons of data, a 2 x 2 moving average is calculated. If there are two or fewer seasons of data available, the transformer issues a warning and return seasonality factors of 1.
7. Whenever a moving average is computed, data is lost. For example, for a three month moving average, it takes the first three months of data to calculate the first moving average. The Seasonality transformer centers the moving average value; thus, the moving average for the first three points is assigned to the second observation. In taking averages of averages, as is done in calculating seasonality indexes, even more data is lost. To avoid losing data, the transformer uses a weighted averaging technique to estimate the beginning and ending moving average values to preserve the volume of data without affecting its integrity.
8. The transformer locates and removes outliers. An outlier is defined as any index that is more than two standard deviations from its moving average. If an index is beyond this level, it is replaced with the mean of the indexes from the season before and the season after. For example, if the index for January 1987 is more than two standard deviations above or below its moving average, it would be replaced by the average of the January 1986 and the January 1988 indexes.
9. The Seasonality transformer recalculates the moving averages and adjusts them so that the mean for any given season is 1. For this purpose, an adjustment factor is computed by dividing the number of periods in a season by the sum of the moving averages.
10. The moving averages are then multiplied by this adjustment factor to produce the final seasonal index. The seasonal indexes are indicators showing the amount by which each period is above or below the average for the entire season. They are also used to calculate the deseasonalized series.
11. To calculate the deseasonalized value of an observation, the raw value is divided by the seasonality factor.

The Seasonality transformer automatically calculates the seasonality factors, the deseasonalized observations, and the deseasonalized trading day-adjusted observations and sends them to an output region, with the raw series and other pertinent input information.

To yield reliable seasonality factors, the Seasonality transformer should have at least two full seasons of data to analyze; five seasons are recommended. The reliability of the seasonality analysis increases in direct proportion to the number of periods of data. Using data from less than two full seasons causes the Seasonality transformer to set all seasonality coefficients to 1.0.

Seasonality Transformer Formulas

Because the formulas used by the Seasonality transformer are very complex, a detailed discussion of them is beyond the scope of this document. Several of the formulas used to construct the most important statistics in the output regions are documented in this section. For more information on the calculation of moving averages, see “Moving” on page 440. The formulas in this section use the symbols defined in Table 66.

Table 66. Seasonality formula symbol definitions

Symbol	Definition
AT	Average number of trading days in all periods
D_i	Deseasonalized value for any given period in a series
DT_i	Deseasonalized, trading day-adjusted value for any given period
E_i	Seasonalized value for any given period in a series
MA_i	Single moving average with the length of one season whose value is assigned to the *i*th period
n	Number of periods in a season
N	Number of periods in a series
NS_i	Non-normalized final seasonal factor for any period
P_i	Preliminary seasonal factor for any given period
R_i	Raw value for any given period in a series
S_i	Normalized final seasonal factor for any period
ST	Sum of trading days specified in all periods
T_i	Trading day-adjusted value for any given period in a series
TD_i	Observed number of trading days in any given period
TF_i	Trading day factor for any given period

If a normalized trading day adjustment is requested, the trading day adjustment factor for any given period is calculated as:

$$TF_i = \frac{ST}{TD_i}$$

If an average trading day adjustment is requested, the trading day adjustment factor for any given period is calculated as:

$$\mathrm{TF_i} = \frac{\mathrm{AT'}}{\mathrm{TD_i}}$$

Where *AT'* is the user input in the `Average number of trading days in a 'Normalized' period` (not the same as the *AT* quantity in the average trading day adjustment calculations).

The trading day-adjusted value for any given period is calculated as:

$$T_i = R_i \times TF_i$$

A moving average with a length of the season (MA_i) is calculated for the values or trading day-adjusted values, if requested.

If no trading day adjustment is specified, the preliminary seasonality factor for any given period is calculated as follows:

$$P_i = \frac{R_i}{MA_i}$$

If a trading day adjustment is specified, the preliminary seasonality factor for any given period is calculated as:

$$P_i = \frac{T_i}{MA_i}$$

A 3 x 3 or 2 x 2 moving average of the preliminary indicators is calculated for each period, across seasons. If any preliminary indicator is more than two standard deviations from its corresponding moving average, its value is replaced with the mean of the indicators for the same period in the preceding and following seasons. After the outlier values are removed, the 3 x 3 or 2 x 2 moving average is recalculated. If more than one resulting moving average value for a given period results, they are averaged, forming the non-normalized final seasonal factors (NS_i).

The normalized final seasonal factor for a specified period is calculated as:

$$S_i = NS_i \times \left[\frac{n}{\sum_{i=1}^{n} NS_i} \right]$$

The deseasonalized value for any given period is calculated as:

$$D_i = \frac{R_i}{S_i}$$

The deseasonalized, trading day-adjusted value for any given date is calculated as:

$$DT_i = \frac{D_i}{TD_i}$$

The seasonalized value for any deseasonalized value is calculated as:

$$E_i = D_i \ast S_i$$

Using the Seasonality Transformer with the Forecast Transformer

The Seasonality and Forecast transformers were designed to work together to make it easy to develop an application using them. Generally, you start by using the Seasonality transformer to deseasonalize a series and then use the Forecast transformer to project the trend into the future.

To use the two transformers together:

1. Place the transformers in the same Capsule icon window.
2. Connect the Seasonality transformer Output 1 region (the deseasonalized output region) to the Forecast Input 2 region.
3. Connect the Seasonality transformer Output 2 region (the seasonality factors region) to the Forecast transformer Input 1 region.
4. Copy the period table used for the Seasonality transformer (Input 3) and connect it to the Input 3 region of the Forecast transformer.
5. You must provide the appropriate values for the transformer parameters. There are several parameters that are especially important when the two transformers are used together:
   - If output region column titles are requested in the Seasonality transformer, specify the appropriate number of header rows in the Forecast Transformer Controls window.
   - Make sure that the date resolution and seasonality length are the same in both transformers and correspond to the values in the date and period number columns of the period table (Input 3 for both transformers).

- Make sure that the column location specifications for the Forecast input regions correspond to the actual locations of various data in the Seasonality output regions.
- Specify whether to adjust the input and output forecast series for seasonality and trading days, based on whether you added, removed, or ignored seasonality and trading days effects to or from the Seasonality transformer output.

Chapter 6. Transformer Execution Language

The Transformer Execution Language (TXL) is a programming language created for the Function transformer. TXL is capable of performing simple tasks (such as entering column names), or complex functions for time-lag analysis.

TXL consists of elements and syntax. Elements are names, constants, functions, and operators. Names are either fixed ($A_) or positional (A$3). Constants are text (#STRA), number (#PI), special output strings (#NA), or computed positions (#ROW). Functions can be used on a range of data or on a single number. Examples of this are MIN or ABS. Operators are unary and binary, following specific precedence rules.

The syntax of a TXL program is a set of assignments, expressions, and transfer values, each separated by commas. Expressions are statements that describe data and processing. TXL can also contain expressions that can be in many forms and contain many elements.

Use the following guidelines when creating a TXL program:

- Specify column names with the one-letter or two-letter name for the column, just as you do in transformers. (Integers do not identify columns, except when used following $, ¢, or & characters. See “TXL Elements and Syntax” on page 498, and the examples that follow.)
- The basic mathematical operators, such as **+** and **-**, also work.
- Type numbers as expected. The decimal point is optional in whole numbers. Some functions and operators expect non-negative numbers.
- Use a dollar sign ($) followed by a column name to copy a column without changes.

You can create a wide variety of reports. For example, consider the following specification:

```
$A, b+c+d, e+f+g, (b+c+d)/(e+f+g)*100
```

This TXL specification creates four columns of output:

- The first column is a copy of input column A.
- The second column is the sum of input columns B, C, and D, which might be the sum of volume data for three markets, such as East, Midwest, and West, resulting in total volume.
- The third column is another sum of input columns E, F, and G, which might represent another set of three markets, or similar volume data for a competitor.
- The fourth output column is the first sum divided by the second sum, then multiplied by 100. If `e+f+g` is the competitor's volume data, this ratio

indicates whether your company is outselling a competitor, and by what percentage.

In addition to basic arithmetic expressions and a reference mechanism that provides access to values by position or calculated value, TXL provides:

- A wide variety of mathematical functions (log, floor, ceiling, exponentiation, square root)
- Built-in constants (e, pi)
- Spreadsheet procedures (sum, minimum, maximum, average)
- A complete set of logical operators (and, or, not)
- Relational operators (=<, <=, >, >=,!=).
- Sting manipulation functions (concat, left, right, mid, length)

Installing a TXL Program

You can place TXL programs in the transformer through any one of the five `Output specification n` parameter fields by typing directly into the field.

Alternatively, you can type any TXL program into a Text icon and then copy the text into one or more of the five `Output specification n` parameters fields. This method allows you to write programs that might be too long to be viewed in the parameter fields. Because TXL is a format-free language, you can use tabs, carriage returns, italics, boldface, and comments in the Text icon.

You can copy the TXL program into any of the five `Output specification n` parameter:

1. Write a TXL program in a Text icon.
2. Select the text in the Text icon, and press the Copy function key.
3. Move your cursor to any one of the five `Output specification n` parameter, and click to copy the text to that field.

TXL Elements and Syntax

This section contains the TXL elements and syntax as implemented in the Function transformer. Table 67 is an alphabetic list of the TXL elements used in the Function transformer, with examples and brief descriptions of each element.

Table 67. Alphabetic list of TXL elements

Element name	Syntax example	Description
Column Name	A	The numeric value of the named column. Non-numeric cells are converted to blank cells.

Table 67. Alphabetic list of TXL elements

Element name	Syntax example	Description
$ Column Name	$A	The contents of the named column as is. When used, a particular column is copied from input to output in its native data type. This is helpful when you are copying text strings and dates. Column names preceded by $ cannot be used in expressions.
Column Name $ Expression	A$3	A value in the cell at the current row Expression columns after the specified column.
Column Name ! Number	A!3	The value in the cell at the specified column of row Number.
Column Name @ Number	A@3	The value in the cell at the specified column Number rows before the current row.
$ Number	$1	Results of the expressions evaluated in the current row of output column Number.
#PR Number or ¢ Number	#PR 1 or ¢1	Results of the expressions evaluated in the previous row of output column Number.
Comments	[This is a comment]	Any text enclosed between square brackets.
Computed Position	#ROW, #NCOL-1	#ROW indicates current row number; #NCOL is the number of columns in the current row.
Constants	Numeric Format Text	Data that has fixed, predefined values.
Functions: mathematic, spreadsheet, #CCOL	#ABS(A, ,#CCOL(A)) a,b, #AVE(a:b)	Predefined mathematical, spreadsheet, and nonmathematical functions.
Mathematic Operators	Logical Binary Unary Relational	See Table 68 on page 509 for a list of mathematical operators and examples, and relational and logical operators.
Variable Assignment	var or #VAR:	A temporary storage location called a variable, where the value of the expression is placed in the named variable. See “Variable Assignment” on page 511 for details and examples.

Column, Row, and Cell Names

You can refer to a column of the input region by typing its one- or two-letter column name. Column names can be:

- Uppercase or lowercase letter from A (or a) to ZZ (or zz) (columns)
- The character $ followed by an expression (columns or rows)
- The column name, then @ or $, or ! followed by a number (cell)

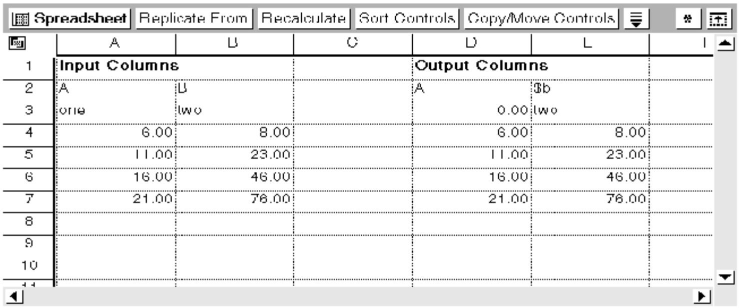
Expressions using column, row, and cell names include:

Column Name

The numeric value of a column. Non-numeric cells are converted to blank cells.

When the TXL syntax is *A,$b*, the data shown in the Input Column produces the data shown in the Output Column in the following example.

Please note that text input is, by default, read as the value 0.00.

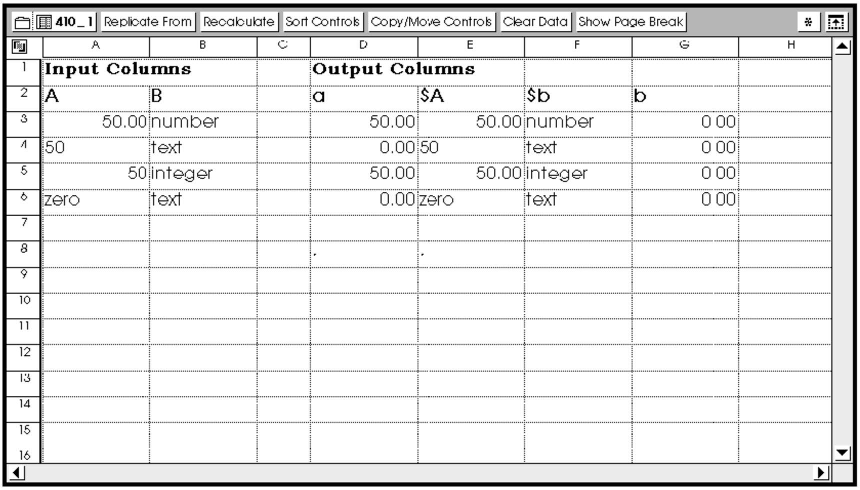


	A	B	C	D	E
1	**Input Columns**			**Output Columns**	
2	A	B		A	$b
3	one	two		0.00	two
4	6.00	8.00		6.00	8.00
5	11.00	23.00		11.00	23.00
6	16.00	46.00		16.00	46.00
7	21.00	76.00		21.00	76.00
8					
9					
10					

$ Column Name

This syntax indicates that a particular column should be copied from input to output in its native data type. This is used when copying text strings and dates.

When the TXL syntax is *a,$A,$b,b*, the data shown in the Input Column produces the data shown in the Output Column in the following example.



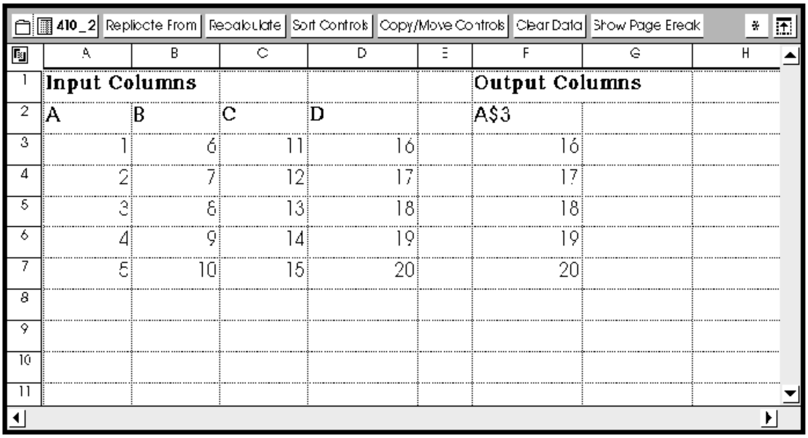
	A	B	C	D	E	F	G	H
1	**Input Columns**			**Output Columns**				
2	A	B		a	$A	$b	b	
3	50.00	number		50.00	50.00	number	0 00	
4	50	text		0.00	50	text	0 00	
5	50	integer		50.00	50.00	integer	0 00	
6	zero	text		0.00	zero	text	0 00	
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								

Column Name $ Expression

A value in the column at the current row `Expression` columns after the column `Column Name`. If `Expression` is negative, the column is `Expression` columns before `Column Name`.

When the TXL syntax is *A$3*, the data shown in the Input Column produces the data shown in the Output Column in the following example.

Do not use negative expressions. Negative strings such as *A$ -2* or *A$ -1* are meaningless because the results will be out of range.

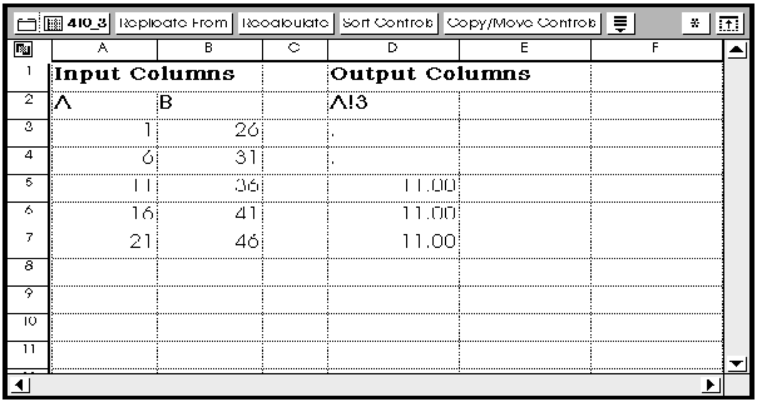


	A	B	C	D	E	F	G	H
1	**Input Columns**					**Output Columns**		
2	A	B	C	D		A$3		
3	1	6	11	16		16		
4	2	7	12	17		17		
5	3	8	13	18		18		
6	4	9	14	19		19		
7	5	10	15	20		20		
8								
9								
10								
11								

Column Name ! Number

The value in the cell at the column `Column Name` of row `Number`. The first row after the heading is row 1.

When the TXL syntax is *A!3*, the data shown in the Input Column produces the data shown in the Output Column in the following example.

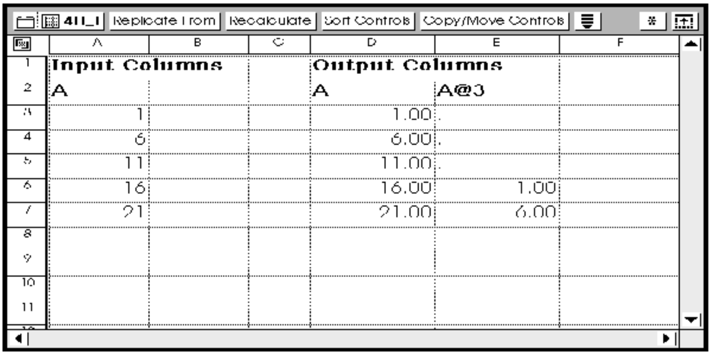


	A	B	C	D	E	F
1	Input Columns			Output Columns		
2	A	B		A!3		
3	1	26		.		
4	6	31		.		
5	11	36		11.00		
6	16	41		11.00		
7	21	46		11.00		
8						
9						
10						
11						

Column Name @ Number

The value in the cell at the column `Column NameNumber` rows before the current row.

When the TXL syntax is *A,A@3*, the data shown in the Input Column produces the data shown in the Output Column in the following example.

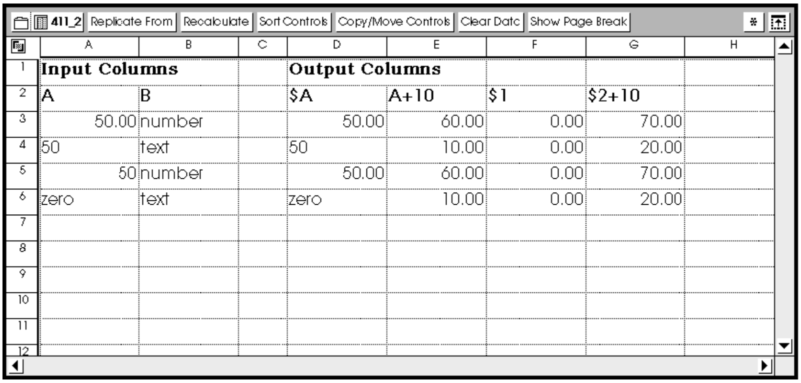


	A	B	C	D	E	F
1	Input Columns			Output Columns		
2	A			A	A@3	
3	1			1.00	.	
4	6			6.00	.	
5	11			11.00	.	
6	16			16.00	1.00	
7	21			21.00	6.00	
8						
9						
10						
11						

$ Number

These values are the result of the expression evaluated in output column `Number`. If the expression has not been evaluated, or its `Number` is greater than the number of expressions, the register contains 0.0. Also, if the output column contains text values, the register contains 0.0.

When the TXL syntax is *$A,A+10,$1,$2+10*, the data shown in the Input Column produces the data shown in the Output Column in the following example. Error cells have been generated where non-numeric cells were used in mathematical calculations.

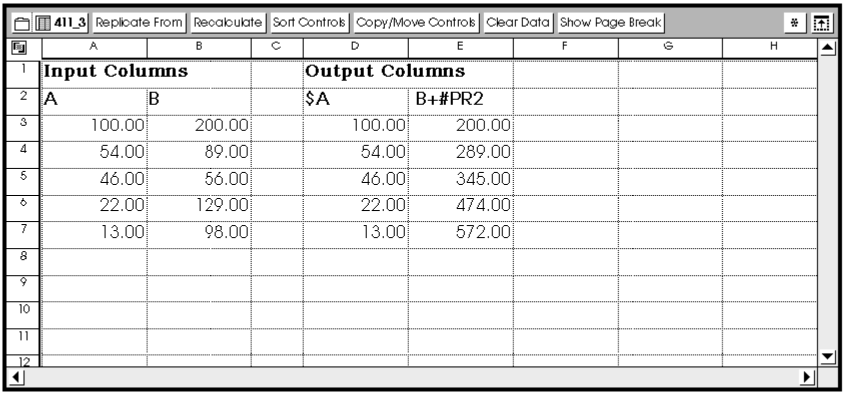


	A	B	C	D	E	F	G	H
1	Input Columns			Output Columns				
2	A	B		$A	A+10	$1	$2+10	
3	50.00	number		50.00	60.00	0.00	70.00	
4	50	text		50	10.00	0.00	20.00	
5	50	number		50.00	60.00	0.00	70.00	
6	zero	text		zero	10.00	0.00	20.00	
7								
8								
9								
10								
11								
12								

#PR or ¢ Number

These values are the result of the expressions evaluated in the previous row of output column `Number`.

When the TXL syntax is *$A,B+#PR2*, the data shown in the Input Column produces the data shown in the Output Column in the following example.



	A	B	C	D	E	F	G	H
1	Input Columns			Output Columns				
2	A	B		$A	B+#PR2			
3	100.00	200.00		100.00	200.00			
4	54.00	89.00		54.00	289.00			
5	46.00	56.00		46.00	345.00			
6	22.00	129.00		22.00	474.00			
7	13.00	98.00		13.00	572.00			
8								
9								
10								
11								
12								

In the preceding example, the values in column *B+#PR2* in the Output Columns are:

- The value 200 is the value in column B row 1.
- The value 289 is the sum of row 1 in column B+#PR2 (200) + row 2 (89) in column B.
- The value 345 is the sum of row 2 in column B+#PR2 (289) + row 3 (56) in column B.

- The value 474 is the sum of row 3 in column B+#PR2 (345) + row 4 (129) in column B.
- The value 572 is the sum of row 4 in column B+#PR2 (468) + row 5 (98) in column B.

Comment

Any text enclosed by square brackets is assumed to be a comment and is ignored by the transformer. Comments cannot be nested; the first closing bracket terminates the comment. However, more than one comment can be displayed in sequence. For example:

```
[This is a comment]
[and so is this]
```

Constants

Constant names are case-sensitive and must be in all uppercase letters.

Three types of built-in constants (data that have fixed, predefined values) are available:

Numeric constants

The numeric constant specifications and their values are:

#E

2.71828

#PI

3.14159

#RBP

+99,999,999 (Large Positive Number)

#RBN

-99,999,999 (Large Negative Number)

When you use #E, #PI, #RBP, and #RBN constants, the value of the constants is returned.

When the TXL syntax is A*#E, A*#PI, the data shown in the Input Column produces the data shown in the Output Column in the following example.

Input Column	Output Columns	
A	A*#E	A*#PI
1.00	2.72	3.14
6.00	16.31	18.85
11.00	29.90	34.56
16.00	43.49	50.27

Formatting constants

The formatting constant specifications and their values are:

#BLANK
Prints a blank cell

#ERROR
Prints "=Error”

#FALSE
Prints "=False”

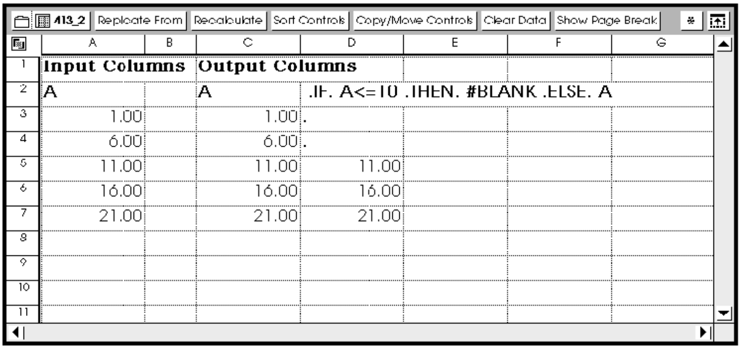
#NA
Prints "=N/A”

#NC
Ensures that nothing is output for this column (No Column)

#TRUE
Prints "=True”

When you use #BLANK, #ERROR, #FALSE, #NA, #NC, and #TRUE constants, the values of all the constants are returned, except for #NC, which ensures that nothing is output for that column.

When the TXL syntax is *A,A<=10?#BLANK:A*, the data shown in the Input Column produces the data shown in the Output Column in the following example. See “.IF. expression1.THEN. expression2.ELSE. expression3” on page 512 for a complete explanation of this TXL element.



	A	B	C	D	E	F	G
1	Input Columns		Output Columns				
2	A		A	.IF. A<=10 .THEN. #BLANK .ELSE. A			
3	1.00		1.00	.			
4	6.00		6.00	.			
5	11.00		11.00	11.00			
6	16.00		16.00	16.00			
7	21.00		21.00	21.00			
8							
9							
10							
11							

Text constants

The text constant specifications and their values are:

#STRA

Prints the contents of the STRA parameter

#STRB

Prints the contents of the STRB parameter

#STRC

Prints the contents of the STRC parameter

#STRD

Prints the contents of the STRD parameter

#STRE

Prints the contents of the STRE parameter

The parameter contents can be @-variables.

Computed Position

The computed position specifications and their values are:

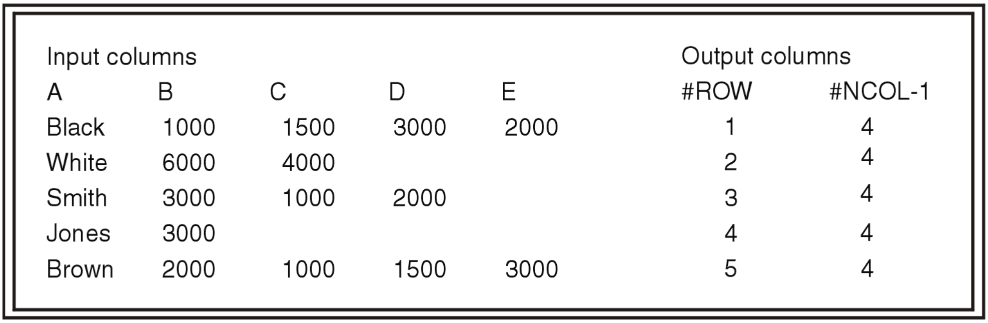
#ROW

Displays the current row number. Row 1 is the first row after the header.

#NCOL

Displays the number of columns in current row. Column count starts with 1.

When the TXL syntax is #ROW,#NCOL-1, the data shown in the Input Column produces the data shown in the Output Column in the following example.



Input columns					Output columns	
A	B	C	D	E	#ROW	#NCOL-1
Black	1000	1500	3000	2000	1	4
White	6000	4000			2	4
Smith	3000	1000	2000		3	4
Jones	3000				4	4
Brown	2000	1000	1500	3000	5	4

In the previous example, the data in column A is a list of the sales representatives, and the remaining columns are individual sale for the month. Output columns represent the number of sales per representative.

Functions

The predefined mathematical functions are:

#ABS (expression)
Absolute value of expression.

#CEIL (expression)
Ceiling value of expression.

#EXP (expression)
Exponentiation of expression, where the expression is valid for exponents up to +79.

#FLOOR (expression)
Floor value of expression.

#LOG (expression)
Logarithm (base 10) of expression, where expression is valid for positive numbers.

#LN (expression)
Natural logarithm (base e) of expression, where expression is valid for positive numbers only.

#SQR (expression)
Square value of expression.

#SQRT (expression))
Square root of expression, where expression is valid for positive numbers only.

The argument to a function can be any valid expression. These functions might be unreliable with very large or very small arguments when overflow or underflow is a possibility.

When the TXL syntax is #CEIL(A),#FLOOR(A),#SQR(A), the data shown in the Input Column produces the data shown in the Output Columns in the following example.

A	B	C	D
Input Column	Output Columns		
A	#CEILA	#FLOOR(A)	#SQR(A)
1.20	2.00	1.00	1.44
6.40	7.00	6.00	40.96
11.80	12.00	11.00	139.24
16.60	17.00	16.00	275.56

The following nonmathematical, predefined function returns the numeric value of the specified column name:

```
#CCOL (COLUMN)
#CCOL(D) = 4
```

Internally, column numbers start with *a = 1*, *b = 2*, *, aa = 27*.

The Spreadsheet functions are:

#AVE(a : b)
Average value in columns a through b

#MAX(a : b)
Largest value in columns a through b

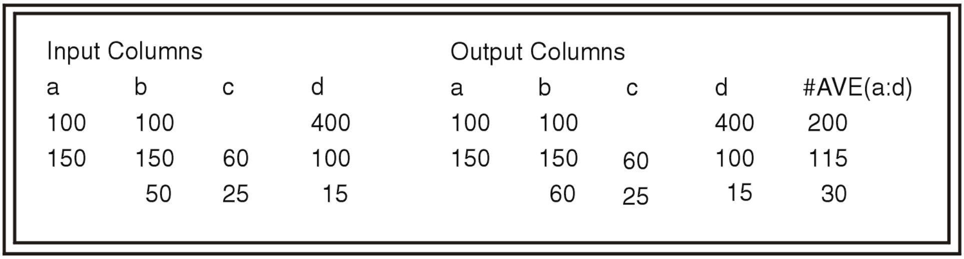
#MIN(a : b)
Smallest value in columns a through b

#SUM(a : b)
Sum of all values columns a through b

a and *b* can be any valid expressions or column names. If *a* or *b* is an expression, its result is converted into a column name. If a cell does not exist, the function ignores references to it. Therefore, the average will be the true average of the existing cells.

If *a* or *b* is an expression whose result cannot be converted to a column name, an error message is displayed, and the affected output cells are set to Error.

When the TXL syntax is `a,b,c,d,#AVE(a:d)`, the data shown in the Input Column produces the data shown in the Output Columns in the following example.



Input Columns				Output Columns				
a	b	c	d	a	b	c	d	#AVE(a:d)
100	100		400	100	100		400	200
150	150	60	100	150	150	60	100	115
	50	25	15		60	25	15	30

In the previous example, the first row in column c and third row in column a are ignored for the #AVE calculation.

Mathematical Operators

The mathematical operators are shown in Table 68 in order of increasing precedence.

Table 68. Mathematical operators in order of precedence

Operator	Type	Operation
|, ||, .OR.	Binary logical	or
&,&&,.AND.	Binary logical	and
>, .GT.	Binary relational	greater than
>=, .GE.	Binary relational	greater than or equal to
<, .LT.	Binary relational	less than
<=, .LE.	Binary relational	less than or equal to
==,.EQ.	Binary relational	exactly equal to
!=, .NE.	Binary relational	not equal to
-	Binary mathematic	subtraction
+	Binary mathematic	addition
%	Binary mathematic	modulus/division remainder
/	Binary mathematic	division
*	Binary mathematic	multiplication
-	Unary mathematic	negation
+	Unary mathematic	plus (no operation)
˜, Ø, .NOT.	Unary logical	negation

Table 68. Mathematical operators in order of precedence

Operator	Type	Operation
^	Binary mathematic	exponentiation

Mathematical operators follow conventional rules of algebra. Exponentiation expects a non-negative base value.

Relational operators compare two expressions. If the comparison is true, the relational expression is equal to 1.0; if false, the expression is equal to 0.0. Two values are considered equal if the difference between them is less than +0.0001.

The logical operators use the standard definitions of mathematical logic. Logical operators interpret 0.0 as false and any nonzero result (at least 0.0001 different from 0) as true.

The following values have precedence lower than .OR.:

.IF.

.THEN.

.ELSE.

?

:

The following operators have precedence higher than **Ø**:

@, **!**, **$**

Operator Examples

If an expression includes parentheses, the parentheses define the order of operation.

For example, when the TXL syntax is `(A+3)*2`, which means the results of column A plus 3 is multiplied by 2, the data shown in the Input Column results in the data shown in the Output Column.

```
Input Column                    Output Columns
A                               (A+3)*2
  1.00                            8.00
  6.00                           18.00
 11.00                           28.00
 16.00                           38.00
 21.00                           48.00
```

If an operator is used between two expressions, the operator is applied to the result of each expression.

For example, when the TXL syntax is `A+B`, which means the value of column A is added to column B, the data shown in the Input Column results in the data shown in the Output Column.

Input Column		Output Columns
A	B	A+B
1.00	2.00	3.00
6.00	7.00	13.00
11.00	12.00	23.00
16.00	17.00	33.00
21.00	22.00	43.00

Variable Assignment

A variable assignment saves the result of any expression into a temporary storage location called a *variable*. In an assignment, the value of the expression is placed in the named variable. Thus, an assignment is Variable=Expression. A variable can be used as an expression any place that a number is legal. Any name can be a variable, provided that all the characters in the name are alphabetic.

For example, you can direct that whenever var or #VAR: is encountered by a user, the symbol var can be assigned to translate to numeric 33. When the TXL syntax is `A=33,A,A*A`, the data shown in the Input Column produces the data shown in the Output Columns in the following example.

Input Columns	Output Columns	
A	A	A*£A
1.00	1.00	33.00
6.00	6.00	198.00
11.00	11.00	363.00
16.00	16.00	528.00
21.00	21.00	693.00

No-Row Test

The No-Row Test allows suppression of an output row. If the expression in the test evaluates to false (value less than 0.0001), the row is not written to the output after it has been completely computed. Because the No-Row Test does not produce an output column, the test is the only allowable expression in a particular specification when it is used.

For example, if the data looks like this:

A	B	C	D
Product	Units Sold	Dollars Revenue	Cost
Gadget	5000	10,000	5000
Widget	7500	12,000	7500
Toy	200	100	200
Box	3500	17,500	7000

the TXL syntax is `$A,b,c,c-d,#NR(B>1000)`, which returns the units, dollars, and profit for all products with more than 1000 units sold:

$A	B	C	C-D
Gadget	5000	10,000	5000
Widget	7500	12,000	4500
Box	3500	17,500	10,500

In the previous example, the toy row is not displayed in the output because units sold were fewer than 1000.

.IF. expression$_1$.THEN. expression$_2$.ELSE. expression$_3$

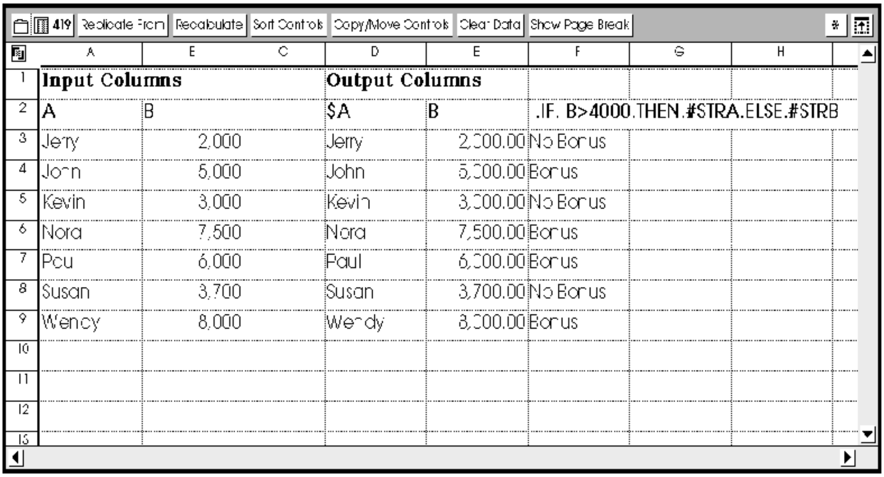
If expression$_1$ is nonzero, the overall expression evaluates to expression$_2$. If expression$_1$ is 0, the overall expression evaluates to expression$_3$. Nonzero is defined as greater than 0.0001 in absolute value.

If expression$_2$ or expression$_3$ is a column name, only numeric cells will be displayed.

If the data looks like that shown in the Input Column and the TXL syntax is:

```
$A,B, .IF. B>4000 .THEN. #STRA .ELSE. #STRB
```

and #STRA contains Bonus and #STRB contains No Bonus, the output will look as shown here:



	A	B	C	D	E	F	G	H
1	**Input Columns**			**Output Columns**				
2	A	B		$A	B	.IF. B>4000.THEN.#STRA.ELSE.#STRB		
3	Jerry	2,000		Jerry	2,000.00	No Bonus		
4	John	5,000		John	5,000.00	Bonus		
5	Kevin	3,000		Kevin	3,000.00	No Bonus		
6	Nora	7,500		Nora	7,500.00	Bonus		
7	Paul	6,000		Paul	6,000.00	Bonus		
8	Susan	3,700		Susan	3,700.00	No Bonus		
9	Wendy	8,000		Wendy	8,000.00	Bonus		
10								
11								
12								
13								

In the TXL syntax:

```
$A,B, .IF. B>4000 .THEN. #STRA .ELSE. #STRB
```

one space is required before and after each period in the TXL.

Expression$_1$? Expression$_2$: Expression$_3$

This is a condensed version of the .IF./.THEN./.ELSE. construct. If expression$_1$ is nonzero, the overall expression evaluates to the result of expression$_2$. If expression$_1$ is 0, the overall expression evaluates to expression$_3$. Nonzero is defined as greater than 0.0001 in absolute value.

If expression$_2$ or expression$_3$ is a column name, only numeric cells will be displayed.

For example, the expression:

```
B>4000?#STRA:#STRB
```

works the same way as

```
.IF.B>4000.THEN.#STRA.ELSE.#STRB
```

described in the previous example.

Blank spaces before and after periods are not required for this TXL element. For an example of this TXL, see “Constants” on page 504.

String Manipulation Functions

#CONCAT

Example: **CONCAT(a:d)**

This expression allows the concatenation of consecutive columns **only**. For example, the expression above would concatenate column a to d. The result is always a string column. The expression concatenates any column type, string column, date column, or number column. This is the accepted format for this expression. To do concatenation of nonconsecutive column, use the following syntax:

a$ + b$ + d$

The dollar sign in front of a column indicates that the column is to be considered as a string column. The dollar signs differentiate between addition and concatenation of different columns.

Important: The concatenation command will only work on column data. It also can not be combined with other string commands. In order to take advantage of both concatenate and other string commands you must separate them using two different function transformers. It is recommended that you perform string manipulation commands in the first function transformer and then combine them using concatenation in the second.

Valid syntax:

```
#CONCAT(a:b)
 $a + $b + $c
#MID(a,2,3),#STRA,#LEFT(a,3)
```

Invalid syntax:

```
#CONCAT(#RIGHT(a,2):b)
 $a+#MID(a,2,3)
#MID(a,2,3)+#STRA + #LEFT(a,3)
```

#RIGHT

This expression returns a specific number of characters starting from the right end of a string in a column.

The first argument can be a column, a string or certain expression that returns either a string or a column number.

Expressions allowed:

$a+$b+$d
#STRA

#STRB
#STRC
#STRD
#STRE

The second argument specifies the number of characters to return. The second argument can be a number, a column, or an expression that returns a numeric value. When the second argument is a column, the data in the column must be numeric; otherwise the expression returns an error.

SUM
AVE
LEN
A column
Addition
Subtraction, addition, multiplication, and division of columns, or numbers

Examples:

- #RIGHT(a,2) - returns the last two characters of column a.

- #RIGHT(a,b) - returns a number of characters from a text string in column a.

  The number of characters to return is the numeric value in column b. The expression returns an error for every cell that contains non-numeric data type in column b. The same goes for column a. The expression returns an error for every cell that contains non-string data type.

#LEFT

This expression returns a specific number of characters starting from the left end of a string in a column.

The first argument can be a column, a string or an expression that returns either a string or a column number.

Expression allowed:

$a+$b+$d
#STRA
#STRB
#STRC
#STRD
#STRE

The second specify the number of characters to return. The second argument can be a number, a column, or an expression that returns a number. When the second argument is a column, the data must be numeric, otherwise it return an error.

SUM
AVE
LEN
A column
Addition
Subtraction, addition, multiplication, and division of columns, or numbers.

Examples:

- #LEFT(a,2) - returns the first two characters of column a.
- #LEFT(a,b) - returns a number characters from a text string in column a.

  The number of characters is decided from the numeric value in column b. The expression returns an error for every cell that contains non-numeric data type in column b. The same goes for column a. The expression returns an error for every cell that contains non-string data type.
- #LEFT($b+$a, 3) - returns the first 3 characters from combining a cell in column ’b’ to a cell in column ’a’.

#MID

Example: **#MID(a, 3, 3)**

This expression extracts a number of characters from a specific start position in a string.

The first argument can be a column, a string or certain expression that returns either a string or a column number.

Expression allowed:

$a+$b+$d

#STRA

#STRB

#STRC

#STRD

#STRE

The second argument specifies the starting position in a string. The second argument can be a number, a column name, or an expression that returns a numeric value. When the second argument is a column, the data must be a numeric type. Otherwise the expression returns an error.

Expressions allowed:

SUM
AVE
LEN
A column
Addition
Subtraction, addition, multiplication, and division of columns, or numbers

The third argument specifies the number of characters to return. It can be a number, or a column name. When the third argument is a column, the data must be numeric. Otherwise the expression returns an error.

Examples:

- #MID(a, 2,2) - returns the first two characters starting after the second character of a string in column a.
- #MID(a,b,3) - returns three characters from a string in column a starting at the position indicated by the numeric value in column b. The expression returns an error for every cell that contains non-numeric data type in column b. The same goes for column a. It returns an error for every cell that contains non-string data type.
- #MID($b+$a, 3,4) - returns four characters, the result of combining a cell in column b to a cell in column a, starting at position 3 in the new string.

#LEN

This expression returns the length of a string.

The argument can be a column, a string or an expression that returns either a string or a column number.

Expression allowed:
SUM
AVE
LEN
A column
Subtraction, addition, multiplication, and division of numeric columns, or actual numbers

$a+$b
#MID
#RIGHT
#LEFT
#CONCAT

If the argument is a string, it returns the length of that string. Otherwise if the argument is a column, it returns the length for each cell in that column. The data in the cell must be a string.

Examples:

- #LEN(a) - returns the length of each cell in column a.
- #LEN($a+$b) - returns the length of cell a concatenate with b.

String Comparison Functions

The relational expressions or operators compare two strings. If the relation is true, the relational expression is equal to 1, otherwise it's equal to 0.

Expressions	Example
>, .GT.	.IF. #CONCAT(a:b) .GT. #LEFT(a,2) .THEN.#CONCAT(a:a) .ELSE. b
>=, .GE.	.IF. #CONCAT(a:b) .GE. #LEFT(a,2) .THEN.#CONCAT(a:a) .ELSE. b
<, .LT.	.IF. #CONCAT(a:b) .LT. #LEFT(a,2) .THEN. #CONCAT(a:a) .ELSE. b
<=, .LE.	.IF. #CONCAT(a:b) .LE. #LEFT(a,2) THEN. #CONCAT(a:a) .ELSE. b
==, .EQ.	.IF. #CONCAT(a:b) .EG. #LEFT(a,2) .THEN.#CONCAT(a:a) .ELSE. b
!=, .NE.	.IF. #CONCAT(a:b) .NE. #LEFT(a,2) .THEN.#CONCAT(a:a) .ELSE. b

The relational expression accepts two types of data: numeric or string. Both arguments must be the same type.

Important: The concatenation command will only work on column data. It also can not be combined with other string commands. In order to take advantage of both concatenate and other string commands you must separate them using two different function transformers. It is recommended that you perform string manipulation commands in the first function transformer and then combine them using concatenation in the second.

Valid syntax:

```
#CONCAT(a:b)
 $a + $b + $c
#MID(a,2,3),#STRA,#LEFT(a,3)
```

Invalid syntax:

```
#CONCAT(#RIGHT(a,2):b)
 $a+#MID(a,2,3)
#MID(a,2,3)+#STRA + #LEFT(a,3)
```

Chapter 7. Time Series Analysis

Many people are interested in predicting events that will happen in the future. Stock analysts estimate stock prices, marketers predict product sales, and financial analysts approximate future earnings and expenses. Before someone can make assertions about the future, it is important that they understand what has happened in the past. Time-series analysis is a method of studying past events, isolating patterns in past observations, and projecting those patterns into the future. In this type of analysis, a series is a sequence of observations that span a given amount of time. An example of a series is the price of a company's stock recorded each day the market was open over the past five years.

In classical time-series analysis; a series is broken down into four components that determine the value of any one observation in a series. This process of dissecting a series is known as decomposition analysis. The four components of a series are seasonal, trend, cycle, and randomness.

The seasonal component in a series represents repetitive changes that take place over a given time span. The time span can vary from a week to a year. An example of seasonality is retail toy sales. From year to year, toy sales increase in the fall, peak in December, and then drop dramatically in January.

The trend component in a series represents the long-term upward or downward change. An example of a trend in a series is the general increase in the number of people employed in the United States. While employment might decrease in the short term, it increases over the long term.

The cycle component in a series is like the seasonal component in that it is composed of up-and-down fluctuations. However, the duration of these changes generally spans years and varies a great deal. An example of the cycle component is the growth of the U.S. economy. The U.S. economy has gone though many cycles of growth and decline that have varied a great deal in their duration and intensity.

The last component of a series is randomness or irregularity. It is composed of variation in a series that cannot be assigned to the three previously described components. This component cannot be predicted. An example of randomness is the large increase in inflation caused by OPEC's dramatic increase in oil prices in the 1970s.

It is usually assumed that these components have a multiplicative relationship. In other words, to summarize a series as a combination of these components, these components should be multiplied as follows:

$$X_t = S_t \times T_t \times C_t \times R_t$$

In this formula, *X* is the value of the series at time t, and *S* is the seasonality component, *T* is the trend component, *C* is the cycle component, *R* is the randomness component, all measured at the time *t*.

Glossary

This glossary defines terms that are used in this book and throughout the Meta5 library. If you do not find the term you are looking for, see the index of this book.

@-value. The value assigned to an @-variable in the capsule's User Input Controls window.

@-variable. A control variable used to pass information from one tool to another within a Capsule icon. You can use an @-variable (at-sign variable) to represent a value in any tool that can be run inside a capsule application.

alignment. The method by which the Moving Average transformer assigns moving average and rolling sum values to observations used in a calculation. The three alignments are:

- **First** places the computed value with the first observation used in the calculation
- **Last** places the computed value with the last observation used in the calculation
- **Center** places the computed value with the middle observation

alpha. A number between one and 0 that is used by the exponential smoothing methods of the Forecast transformer. Alpha is a coefficient that determines the relative importance of past values in calculating forecasts. When alpha is near 0, observations in the more distant past are given more importance and the resulting forecast is relatively smooth. When alpha is near one, observations in the recent past are given more importance and the forecasts are more sensitive to swings in the data.

alphanumeric. Pertaining to a character set that contains letters, digits, and usually other characters, such as punctuation marks.

alternative hypothesis. A hypothesis that is accepted when the null hypothesis is rejected.

analysis of variance (ANOVA). A method for determining whether the means of several different groups are equal. This method of analysis shows the relationship between a continuous dependent variable and one or more nominal independent variables.

ANOVA. Analysis of variance.

Append transformer. The *transformer* that combines up to 10 tables of data into one table by appending the data in the listed sequence of the tables.

ARIMA. Autoregressive integrated moving average.

arrow. A symbol that connects icons within capsule applications and directs the flow of data. The arrow can be selected or deleted, and its settings can be changed in the Arrow Options window.

association. A relationship between two variables where knowing the value of one variable helps in estimating the value of the second variable. If two variables are not associated, they are independent. The chi-square test identifies whether there is an association between two variables.

autocorrelation coefficient. A measure of a linear relationship between values of an observation and prior values of the same observation. Autocorrelation coefficients can be used to detect the presence of trend and seasonality components in a time series. They can also be used to determine the lags that should be incorporated in an autoregression model. The AutoCorrelation and AutoRegression transformers supply autocorrelations.

autoregression. A method for measuring and modeling the relationship between values and their lags. Autoregression summarizes the relationships between variable values and past values of the same variable. Compare with regression.

autoregressive integrated moving average. A method of describing both stationary and nonstationary time series.

backward regression method. A method for constructing multiple regression models that starts with all independent variables in a model, but removes the least-important independent variables one at a time until only significant independent variables remain in the model.

beta. A number between one and 0 that is used by the Holt's exponential smoothing method of the Forecast transformer. Beta is a coefficient that determines the relative importance of past values in a series. When beta is near 0, observations in the more distant past are given more importance and the resulting forecast is relatively smooth. When beta is near one, observations in the recent past are given more importance and the forecast contains more swings. In the Holt's method, beta is used to control the effect of trend on forecasts while *alpha* is used to control the effect of randomness.

beta weight. A statistic provided by the Regression transformer that measures the relative importance of each independent variable. The variable with the highest absolute beta weight has the most power in explaining variation in the dependent variable. The variable with lowest absolute beta weight is the least important predictor.

bimonth. A two-month period.

blank transformer. A transformer that is not configured to perform a particular function.

Boolean value. A value that is limited to one of two possible values, such as true or false, yes or no, 1 or 0.

capsule. A tool for building an automated processing sequence by joining other tools using their icons.

capsule application. The automated processing sequence built using the capsule tools and run by clicking on an icon.

cell. The intersection of a row and a column. Cells are identified by the labels of the column and row that form them. For example, the cell in column C and row 4 is identified as c4. Cells exist in *cross-tabulation tables* as well as in raw data tables. In a cross-tabulation table, a cell contains either a count or sum of observations with a given combination of grouping variable values. The CrossTab transformer can create cross-tabulation tables.

checksum. A numeric value used to compare the degree of similarity between two Text documents. A checksum is calculated based on the binary values of each alphanumeric character. If the checksums for two Text documents do not match, the documents are different. If the checksums match, the documents are probably identical. The checksum is created by the Word Count transformer.

chi-square goodness-of-fit test. A test to determine whether the number of observations in several categories is the same as the expected number of observations in each category. In this test, the observed category counts are in the data, but the expected counts are supplied. If the probability associated with the chi-square statistic is near 0, the expected distribution is not equal to the actual distribution.

ChiSquare transformer. The *transformer* that helps users study associations between pairs of variables and differences in distribution between an observed sample and a theoretical sample.

chi-square statistic. A test statistic that summarizes differences between observed and expected cell values, based on the sample size. Because the distribution of the chi-square statistic is known, it is possible to calculate the probability of finding a chi-square value that is at least as large as the observed chi-square. If the probability of a chi-square is near 0, the difference between the observed and expected values is significant.

chi-square test. A test that determines whether two variables that form the row and column values of a cross-tabulation table are independent. It compares the actual cell counts with expected cell counts. The expected cell counts are estimated under the assumption that the two variables are independent. If the probability of the observed chi-square is near 0, the actual and expected cell counts are different, because the two variables are associated, not independent.

Clean transformer. The *transformer* that removes columns or rows that contain blanks, N/As, zeros, or other null values.

Clear Contents transformer. The *transformer* that removes the contents of a specified container.

coefficient. A report of statistics computed for each variable (input data column), its autocorrelation, and partial autocorrelation. The statistics include count, mean, standard error, maximum and minimum values, and first-half and second-half means.

coefficient of determination. A statistic (R^2) that measures the proportion of variation in the dependent variable that is explained by a regression model. Thus, an R^2 of 0.69 indicates that the regression model explains 69% of the variation in the dependent variable.

coefficient of variation. A statistic provided by the Elementary transformer that measures the amount of variation in a group relative to the mean. It is calculated by dividing a standard deviation by its mean. This statistic can be very helpful for comparing the distributions of two variables that have different means.

column. A vertical field within a data table or spreadsheet. Data is organized in columns and rows, with columns identified by letters or text strings at the top of the display area.

comment. In a programming language, text that is included as documentation rather than as instructions. Comments are intended to explain certain aspects of a program and have no effect on how the program runs.

comparison period. A portion of a time series that is not taken into account while the Forecast transformer creates initial forecasts for that portion of the series. The transformer calculates predicted values for the comparison period and compares them with the actual values. The forecast methods are then ranked according to the accuracy with which they predicted values in the comparison period.

Compress transformer. The *transformer* that combines or deletes two or more columns or rows of data based on the parameters you specify.

concordant pair. A pair of observations where the value of two variables for one observation are either greater than or less than the value of the two variables for the other. For example, a concordant pair exists if one observation has the values of 1 and 2 for variables A and B, and another observation has values of 3 and 4 for the same variables. The number of concordant pairs in a sample is a component of the Kendall's tau correlation coefficient.

confidence level. The probability that the null hypothesis is not true.

confidence limits. The upper and lower boundaries of a range that should contain a given percentage of observations in a distribution. For example, 95% confidence limits should include 95% of the observations in a normal distribution.

controls area. The top portion of the transformer window, which contains the transformer name and options for controlling what data is shown in the display area.

Copy Icon transformer. The *transformer* that copies an icon to a folder.

correlation coefficient. A measure of the linear relationship between two variables. A correlation coefficient of 1 or -1 indicates a perfect linear relationship; a correlation coefficient of 0 indicates no linear relationship.

covariance. A measure in which two variables vary together in a linear manner. Like the *correlation coefficient*, covariance can be negative or positive; unlike the correlation coefficient, covariance does not have an upper or lower limit. Thus, covariance should not be used to compare the relationship between two variables with the relationship between two other variables that are measured on a different scale.

CrossTab transformer. The *transformer* that constructs contingency tables that are used to study distributions of cross-classified data.

cross-tabulation table. A table used to organize values of one or more variables. If only one variable is summarized, the table contains a single column; its rows correspond to the values of the variable. If two variables are summarized, the table contains several rows that correspond to the values of the first variable, and several columns that correspond to the values of the second variable. The contents of the cells can take two forms: a count of the observations with variable values that match the row and column values, or the value of a third variable summed for all of the observations that fit in the cell.

cumulative frequency. The type of distribution that the Kolmogorov-Smirnov test compares to determine whether two samples differ or whether an observed sample differs from a hypothetical sample. This distribution shows how many items are "less than" or "more than" given values in a grouped data. In a cumulative frequency distribution, observed values are sorted in ascending order; the original values are replaced by the sum of the values up through the current value, thus generating cumulative values. These cumulative values are then divided by the sum of all the observed values.

cycle. One of the four components that comprise a time series. Cycle is composed of irregular up-and-down fluctuations that often span years.

D test statistic. The test statistic of the Kolmogorov-Smirnov test. D is the maximum difference between the cumulative distributions of two samples. The *KSTest transformer* automatically converts D into a probability, which is much easier to interpret. If the probability is near 0, it can be assumed that the two samples have different distributions or the observed sample is different from the theoretical sample.

Data Entry tool. The tool used to enter, retrieve, delete, and update data on a database server, or to set @-*variables* in the User Input Controls window of a capsule application.

date format. The internal storage representation of dates with a particular date resolution.

date resolution. The way in which the interval between dates is interpreted. For example, if the date resolution is Week, the interval between dates is 7 days, but if the date resolution is QuadWeek, the interval is 28 days (4 weeks). Meta5 date resolutions include Day, Week, QuadWeek, Month, EvenBiMonth, OddBiMonth, Quarter, and Year.

decimal precision. The number of digits past the decimal that are used to express the fractional part of a number. More digits result in greater precision.

decomposition analysis. An analytic approach in which the phenomena under study are broken down into various components that can be studied separately. This approach is commonly used in forecasting. Decomposition analysis allows you to break down a time series into trend, seasonality, cycle, and random components, which can be forecast separately and later reassembled.

default value. A value that is assumed when no value has been specified for a parameter.

degree of autoregression. Synonym for *order of autoregression*.

degrees of freedom. The parameter of the F, t, and chi-square distributions that determines the probability of getting a value that is at least as large as the observed F-statistic, t-statistic, or chi-square statistic.

dependent variable. The variable that can be affected by the other variables under study. For example, in *regression analysis*, the values of the dependent variable are estimated based on the values of the independent variables.

deseasonalized series. Time series data that has had the seasonality component removed.

DESKTOP. A Meta5 keyword that causes a search for a particular icon to begin at the desktop level. Used in Meta5 path names.

desktop. The screen space occupied by Meta5.

destination icon. An icon that receives information from other icons.

differencing. A method for removing seasonality and trend effects from a series before autocorrelation or autoregression statistics are computed. A differenced series is the result of subtracting lag values from the values of the original series.

dimension. A column whose data values are used to create the column and row headings for a report.

discordant pair. A pair of observations where the values of one are not always greater or smaller than the values of the other. For example, if the first observation has a higher value for variable A and a lower value for variable B, the pair is discordant.

display area. The lower portion of the transformer window, which displays the input and output data.

distribution. The overall dispersion or scattering of observed data.

double exponential smoothing. A forecasting method that starts with the same calculations used by the *single exponential smoothing* model but uses additional calculations to estimate the trend component, resulting in double exponential smoothing. The value of alpha should be significantly smaller than the alpha used in the single exponential model, in most cases falling between 0.1 and 0.3. This model is most appropriate for forecasting series that show a significant upward or downward linear trend.

Durbin-Watson statistic. A measure of the correlation between residual values and a sequence variable, which measures the passage of time. If the Durbin-Watson statistic is less than 1.5 or above 2.5, the model might not adequately explain the effect of time on the dependent variable. The Durbin-Watson statistic is provided by the Forecast and Regression transformers to measure the adequacy of models.

dynamic buffer. A data storage area that can expand and contract as needed. The Concatenate transformer can be used as a dynamic buffer to contain data from multiple data sources that could exhaust the size limitations of the Spreadsheet tool.

even bimonth. A two-month period ending in an even-numbered month; for example, January and February.

expected value. The value expected in a crosstab cell if there is no association between the variables that define the table columns and rows. This value is subtracted from the actual value in the cell, and the difference is used to calculate the chi-square statistic generated by either the CrossTab transformer or the ChiSquare transformer.

exponential smoothing. A group of forecasting methods that use historical values and residual values to smooth a time series or forecast future values of a series. The Forecast transformer implements four exponential smoothing methods: single, double, triple, and Holt's. Within the set of exponential smoothing models, each model is computed differently and, as a result, predicts different types of trends. Though they use different calculations, all models use a coefficient called alpha. Alpha is a number ranging from 0 to one that represents the weight given to different values in a series when a new series is estimated.

expression. A designation of a symbolic mathematical form, such as an equation.

fact. A column whose data values are used in the body of a report.

field. An area in a window where the user enters information.

file server. The workstation on which the file service runs.

forward regression method. A regression model construction method that starts with a model containing no independent variables. Independent variables are added one at a time until there are no independent variables with sufficiently large *F-statistic* values that are not in the model.

frequency distribution. A table or other form of arrangement that shows the classes into which a set of data is grouped (together with the corresponding frequencies) or the number of items falling into each class.

F-statistic. A test of whether a statistic is important. The value of F increases for statistics that represent stable relationships and decreases for statistics that might have occurred by chance. The significance value associated with the F-statistic is often easier to interpret than the F. If the *significance level* is near 0, it is unlikely that the associated statistic happened by chance.

F to Enter. A criterion for selecting variables in a *multiple regression* model. An independent variable must have an *F-statistic* value that is greater than or equal to this value for that variable to be included in a regression model with the forward or stepwise regression methods.

F to Remove. A criterion for excluding variables from a *multiple regression* model. If an independent variable has an *F-statistic* value that is less than or equal to this value, it will be excluded from a regression model by the backward or stepwise regression methods.

full model regression method. A method that constructs a regression model in one step. This model contains all of the independent variables. It makes no attempt to determine which independent variables are the most or least important predictors of the dependent variable.

goodness-of-fit test. A method of determining whether a distribution is equal to a theoretical distribution specified by the user. If the probability associated with the Kolmogorov-Smirnov test statistic D is near 0, the observed sample is not equal to the specified distribution. This type of analysis is completed by the KSTest transformer.

group column. A single data column containing information that determines the group to which a particular data element belongs. For example, if the first column of the input data contains grouping information, the entry for column A would be a.

grouping. A set of elements or observations that possess one or more characteristics in common. Each group is treated as a distinct set of data, and a set of correlation tables is generated for every requested group. For example, the Correlation transformer can calculate a correlation coefficient that summarizes the linear relationship between price and sales in each of the sales regions of the country.

H test statistic. The Kruskal-Wallis test statistic that measures differences among sample distributions. When samples have different distributions, H is large. The probability value associated with H is often easier to interpret than H. If the probability is near 0, there are significant differences among the sample distributions.

heading row. One or more rows at the top of an input data table that are designated as column titles in the output data.

Holt's two-parameter exponential smoothing. A forecasting method that is similar to the double exponential smoothing technique in that it identifies the trend component and uses it in the forecast. However, as the name implies, in addition to the alpha coefficient used by the other exponential smoothing techniques, it uses a second coefficient, beta. The values of beta and alpha are analogous; beta is used in the equations to estimate the trend, whereas alpha is used to smooth the most recent values of the series and thus reduce randomness. Although this model has the disadvantage of requiring two parameters, it is useful for certain types of series for which different weights should be assigned to the randomness and trend components.

independence. A relationship between two variables where knowing the value of one of the variables does not help predict the value of the second.

independent sample. A sample that is randomly chosen from two or more populations. No attempt is made to ensure that the samples are similar or dissimilar. Each sample can have a different number of subjects.

independent t-test. A test that uses the *t-statistic* to determine whether two independent samples have different means. The IPMean transformer calculates the t-statistic and automatically converts it into a probability. If the probability is near 0, the two samples have different means.

independent variable. In experimental research, a variable whose value can be controlled by the researcher (for example, the type of commercial shown to different market research samples). However, this kind of control is not possible in many forms of data analysis. In those cases, an independent variable is a variable that is believed to affect a dependent variable. For example, regression can be used to analyze sales volumes based upon levels of advertising, price, and prevailing rates. Synonymous with *predictor variable*.

input region. A storage area in a tool, for data transferred from elsewhere. Data can either be copied directly into an input region or, in a capsule application, transferred to the input region by means of an arrow.

integer number. A number that does not contain a decimal place, such as 5, 60, or 912.

interaction term. An *ANOVA* term that summarizes the variation in the dependent variable that is due to the combination of two or more independent variables.

intercept. In multiple regression, the value of the dependent variable when all of the independent variables are 0 or are extrapolated to 0. In effect, the intercept is the predicted value of the dependent variable without the effects of the independent variables.

interval data. Data measured on an interval scale in which the distance between values can be measured and the zero point has been set in an arbitrary fashion. An example of interval data is Fahrenheit temperature. Also referred to as interval-level data.

IPMean transformer. The *transformer* that performs the t-test to measure differences in the distribution of a variable between two paired or independent samples.

Join transformer. The *transformer* that joins the data from two tables.

Kendall's tau correlation coefficient. A nonparametric correlation coefficient that can be interpreted much like the standard correlation coefficient. It ranges from minus one to one; 0 indicates no relationship between the variables. A value of one indicates that if one variable increases, the other variable always increases. A value of minus one indicates that as the value of one variable increases, the value of the other variable always decreases.

Kolmogorov-Smirnov test. A nonparametric test that determines whether two samples have equal distributions. If the probability of the test statistic D is near 0, the samples are different.

Kruskal transformer. The *transformer* that performs the Kruskal-Wallis nonparametric test to determine whether different samples have different distributions of a variable.

Kruskal-Wallis test. A nonparametric test that determines whether several different groups have different distributions. If the probability associated with the test statistic H is near 0, there are differences among the group distributions. The multiple-comparisons analysis and contrast-analysis capabilities provided by the Kruskal transformer help identify exactly which groups differ from other groups.

KSTest transformer. The *transformer* that performs the Kolmogorov-Smirnov nonparametric test to determine whether an observed distribution differs from an expected distribution or whether the distribution of a variable differs between two groups.

Label transformer. The *transformer* that creates mailing labels for form letters.

lag. A time series value that occurred a given number of observations before the current observation. For example, in a time series with monthly data, the sixth lag of the January, 1990 value is the value observed for July, 1989. The sixth lag for February, 1990 is the value observed for August, 1989.

least squares methods. A group of forecasting methods that incorporate the least squares technique used in regression analysis. Least squares method specifies a line in an equation that is closest to all observations. This method uses statistical information that fits a certain trend to time series and then forecasts future values.

Lilliefors test. Variant of the Kolmogorov-Smirnov test using the mean and standard deviation.

linear contrast analysis. An *ANOVA* technique used to join several groups together mathematically before means are compared. For example, if the means of three groups, A, B, and C are being compared, contrast analysis allows ANOVA to determine whether the mean of group A is significantly different from the mean of the combined B and C distribution.

linear relationship. A measure between two variables indicating that as the values in one variable increase, the values of the second variable increase or decrease in a linear fashion. When the values of one variable are plotted against another, the data points should fall into a long, narrow band. If they form some other pattern, the relationship between the two variables is probably not linear. Standard regression analysis assumes that all predictors have a linear relationship with the dependent variable.

long-term differencing. Synonym for *seasonal differencing*.

MakeSecure transformer. The *transformer* that secures all data access icons within a specified container as if each icon was individually secured in its own Icon Options window.

Mann-Whitney test. A nonparametric test that determines whether two independent samples have different distributions. The NPIndependent transformer uses the sum of the ranks to calculate a test statistic U, which is converted into a Z-score and a significance level. If the significance is close to 0, the two samples have significantly different distributions.

matched pair t-test. An analysis method that uses the *t-statistic* to determine whether two matched samples have different means. The IPMean transformer calculates the t-statistic and automatically converts it to a probability. If the probability is near 0, the two samples have different means.

maximum. The largest value of a variable.

mean. The average value of a variable.

median. The midpoint of a sample when values are sorted in ascending order. One half of the values are above and one half are below the median value.

Merge transformer. The *transformer* that combines text with incoming data to produce a report or form letters.

minimum. The smallest value of a variable.

model. A theory (usually expressed mathematically) that attempts to describe the inherent structure of selected aspects of a phenomenon or process and generates an observed data. For example, an equation that expresses a relationship among pertinent variables of a model is referred to as a model equation.

moving average. A measuring method where a new average is computed by dropping the oldest observation and including the most recent observation in a time series data set.

moving average ratio. A measure that represents the distance of each original observation value from its moving average. A value of less than one indicates that the observation is less than the moving average; a value of more than one indicates that the observation is above the moving average.

multicolinearity. A condition in which the independent variables in a multiple regression model are significantly related to one another. This condition is detected with the tolerance measure and can result in a regression model that inadequately explains the dependent variable. In cases of severe multicollinearity, it is mathematically impossible to derive a regression model.

MultiJoin transformer. The *transformer* that joins the data from up to five tables.

multiple comparisons analysis. A technique used by the Kruskal transformer to compare each sample to every other sample to find all differences among sample distributions.

multiple regression. A method for predicting the value of a dependent variable using information from more than one independent variable.

naive method II. A forecasting method supported by the Forecast transformer that uses the last deseasonalized historical value as the value for future periods. This method is provided as a benchmark. If other forecasting methods do not forecast better than the naive II method, the data might not be well suited to the available forecasting methods.

nominal-level data. Data measured on the nominal scale, in which unique values can be determined, but no specific order is implied. Gender is an example of nominal-level data.

nonparametric tests. A set of statistical tests that make no assumptions about the underlying distribution of the variables being analyzed. Most statistical techniques assume that the activity being studied is normally distributed.

normal deviate. Synonym for *Z-score*.

normal distribution. A distribution where the resulting plot is bell-shaped when the values of a variable are plotted. In a normal distribution, most observations occur near the center. Many statistical methods, called parametric statistics, are built around normal distribution.

normalize trading days in seasonality factor. A seasonality analysis method that adjusts the number of trading days in each period. The adjustment factor is the number of days in all trading periods divided by the number of days observed in that trading period. Normalizing trading days adjusts for given differences across many periods in a given year rather than the same period from year to year.

NPCorrelation transformer. The *transformer* that computes Spearman Rank and Kendall Tau nonparametric correlation coefficients to summarize the relationships among two or more variables.

NPIndependent transformer. The *transformer* that performs the Mann-Whitney and Wilcoxon Rank-Sum nonparametric test to determine whether two independent samples have different distributions of a variable.

NPPaired transformer. The *transformer* that performs the Wilcoxon Signed Rank and Sign nonparametric test to determine whether two paired samples have different distributions of a variable.

null hypothesis. A reference point for determining what a statistical test proves. In most cases, an analyst looks for a relationship or association among variables or differences among samples. The belief that this relationship exists is a research hypothesis. The null hypothesis is its opposite; it represents the theory that there are no associations or relationships among the variables or differences among groups.

numeric data format. The representation of a date as the number of days before or after January 1, 1970. For example, 0 represents January 1, 1970, 1 represents January 2, 1970, and -1 represents December 31, 1969. Dates are typically stored in a database in numeric date format.

numerical tolerance. The amount of variance allowed when two numbers are compared.

odd bimonth. A two-month period ending in an odd-numbered month; for example, December and January.

one-tailed significance level. The probability that one distribution is either higher than or lower than another distribution. Also called one-tailed probability level.

order of autoregression. The number of *lags* that are incorporated into an autoregression model. The order of an autoregression is often represented as AR_n where *n* represents the number of lags or terms in the model. For example, an AR_2 model includes two lags. Synonymous with *degree of autoregression*.

ordinal data. Data measured on the ordinal scale, in which unique values can be determined and an order is implied, but the distances among values cannot be measured. An example is customer satisfaction that ranges from very satisfied to very dissatisfied. Many statistics provided by the Significance and Sample Testing transformers are well-suited for analyzing ordinal-level data.

outlier. Any value that is very different from other values in a distribution. The Regression transformer can identify outliers by creating a full model and predicting the values of the dependent variable. Any observation that has a predicted dependent variable value that is more than the specified number of Z-scores from the mean predicted value is called an outlier.

output. Data that has been manipulated as directed by the controls set in the Transformer Controls window.

output region. (1) In a tool, a storage area for data (such as processing results) that is to be transferred elsewhere. (2) A range of spreadsheet cells that accepts data transferred from elsewhere.

paired sample. Two samples consisting of comparable data such as ages of husbands and wives, or the weights of individuals before and after a diet. Paired samples always have equal numbers of subjects. For example, paired samples can be composed of randomly chosen married couples, with wives in one sample and husbands in the other.

PARENT. A Meta5 keyword that causes a search for an icon to begin at a level above the container that holds the transformer. Used in capsule applications to specify Meta5 path names.

partial autocorrelation coefficient. A measure of correlation used to identify the extent of the relationship between current values of a variable and earlier values, while holding the effects of all other *lags* constant.

Pearson correlation coefficient.

Synonym for *correlation coefficient*.

period table. A table of date information, including a column of dates, and period numbers. The Seasonality transformer and the Forecast transformer each require a period table as an input region.

Pivot transformer. The *transformer* that designs a report by rearranging the columns and rows of data.

predictive model. A model that can predict the value of a dependent variable with a given probability for a given value of the independent variable.

predictor variable. Synonym for *independent variable*.

primary grouping. A grouping feature that allows users to categorize a data file into groups that are presented as a series of rows in the cross-tabulation output. Primary grouping forms the rows in a table created by the CrossTab transformer.

probability. The likelihood of an event occurring. Probability is usually measured on a scale of 0 to one. A value near one indicates that an event will almost definitely occur; a value near 0 indicates that an event will almost definitely not occur.

proportion. The product of dividing the sum of values up to the current observation by the sum of all values in the sample.

quadratic. A sampling device in the form of a square lattice. The device can be a framework that can be placed on the ground. Dividing a plot into sub-plots or a square grid of transparencies for superimposition on a map are examples of quadratics.

quadweek. Exactly four weeks, starting on a Sunday.

query. A request for data from a database, based on specified conditions.

randomness. One of the components of a time series identified in *decomposition analysis*. Randomness is basically any variation in the series that cannot be explained or assigned to the other components in a series. An example of randomness is a plant strike and its accompanying effects on product sales.

random number. A number generated by a mechanism that produces irregularity.

range. A group of cells to which a common name is assigned.

rank value. A measure assigned by sorting observations in ascending order. Each observation is given a new value. The lowest raw value receives a rank of 1, the second lowest receives a rank of 2, and so on. Ranks are the basis for most of the nonparametric statistics provided by the Significance and Sample Testing transformers.

ratio data. A quantity that can be measured on the ratio scale where the distance between values can be measured and there is a meaningful zero point. Ratio data allows two different values to be compared as ratios. For example, because the price of an item is measured on a ratio scale, it is possible to say that fifty cents is one half of one dollar.

real number. A number with a decimal point, such as 2.1 or 0.003.

recode. To replace certain values of a variable or column with different specific values. Both the Replace and the Substitute transformers provide facilities for recoding data.

regression. A statistical method of summarizing the relationship between one variable and other variables.

residual. The field in the Output Region Names area that contains an analysis of the autoregression residuals, including time lag, autocorrelation coefficient, the t-value, upper confidence limit value, lower confidence limit value, partial autocorrelation coefficient, chi-square of the autocorrelations, and probability of the observed chi-square for the first specified number of lags.

residual value. A value denoting the difference between the actual and expected or predicted values of a crosstab cell. The residual value is used to compute the chi-square statistic.

result. The value in a spreadsheet cell that represents the outcome of a calculation.

rolling sum. A measure of cumulative values of a series. For example, in a 12-period rolling sum, the first sum would be the cumulative value of the first through the 12th observations and the second rolling sum would be the cumulative value of the second through 13th observations.

root mean squared error. A measure of the accuracy of a predictive model that can be interpreted as the average distance between each predicted value and its actual value.

row. (1) The horizontal component of a table. A row contains one value for each column of the table. (2) A record, or single instance, in a database table. (3) A horizontal field in a spreadsheet, identified by a row number.

row- and column-format. A type of format in which each data field resides in a unique cell that is defined by columns and rows. Most Meta5 tools, such as the Query, SQL Entry, and Spreadsheet tools, provide data in row- and column-format.

row containing title. One of the *header rows* that contains information that the transformer can use to label the output.

seasonal differencing. A differencing method used by the AutoRegression and AutoCorrelation transformers to eliminate seasonality from a time series. To create a seasonally differenced value, the transformer subtracts the value of the period in a previous season from the current value. For example, the value of July, 1989 would be subtracted from the value of July, 1990.

seasonality. One of the components of a time series that represents repetitive changes that take place over a given period. The time frame can vary from one week to a year. The Seasonality transformer can help you identify and remove the seasonal component of a series.

secondary grouping. A grouping feature that categorizes a data file into subgroups that are presented as a series of columns in the cross-tabulation output. Secondary grouping forms the columns in a table created by the CrossTab transformer.

selection rule. A row in the Clean transformer that informs the transformer of the data columns, logical operators, and comparison values that should be used while considering data for selection.

Select transformer. The *transformer* that sends specified data columns from one input region to as many as 10 output regions.

short-term differencing. Synonym for *trend differencing*.

sign test. A nonparametric test that determines whether two paired samples have different distributions. If the probability of the Z-score associated with the sign test is near 0, the two samples have significantly different distributions.

significance level. A probability level calculated for many types of statistics that is used to determine whether the statistic could have happened by chance or represents a stable relationship that could be replicated if the analysis were conducted again under the same circumstances. If the significance level is near 0, it is unlikely that the statistic happened by chance and it is likely that the measure represents a stable relationship. If the significance level is near one, it is very likely that the statistic happened by chance.

single exponential smoothing. A forecasting method that estimates a value in a series based on the previous forecast value and the previous observation. The previous forecast is multiplied by one minus alpha and added to the product of multiplying alpha times the previous observed value. Thus, each predicted value carries information from previous values, so that the influence of earlier values decreases as the forecast proceeds. Single exponential smoothing is most appropriate for forecasting series with little or no trend.

slope. A measure of the linear relationship between a dependent variable and the independent variable. In a regression model with only one independent variable, the slope represents the amount of change in the dependent variable that arises out of a one-unit change in the independent variable.

smoothing. The process of removing fluctuations in an ordered series so that the first differences are regular and the higher order differences small. The smoothing process approximates true values and reduces errors in observation.

Sort transformer. The *transformer* that rearranges data in alphabetic, numeric, or chronological order.

Spearman rank correlation coefficient.

A nonparametric correlation coefficient that can be interpreted like the standard correlation coefficient. The Spearman rank correlation coefficient ranges from minus one to one; 0 indicates no relationship between the pair of variables. A value of 1 indicates that if one variable increases, the other variable always increases; a value of negative one or –1 indicates that as the value of one variable increases, the value of the other variable always decreases.

Spreadsheet. A Meta5 tool used to perform calculations and analyses, schedule projects, and perform iteration in embedded capsules.

spreadsheet-formatted data. Synonym for row-formatted and column-formatted data.

SQL. Structured Query Language.

SQL Entry. A Meta5 tool used to run Structured Query Language programs.

Structured Query Language (SQL). A command language for use with relational databases. The language consists of statements to insert, update, delete, query, and protect data. Meta5 tools use SQL to communicate with database servers.

standard deviation. A measure of a distribution's dispersion about the mean. Unlike the variance, standard deviation is measured on the same scale as the variable it summarizes.

standard error. The standard deviation of a sampling distribution, which is the theoretical distribution of all possible variable values in a population.

stationary series. A time series that shows no evidence of trend or seasonality. In order for the AutoRegression transformer to derive a valid model, the series must be stationary.

step. Addition or removal of a variable to or from a model.

stepwise regression method. A method for constructing a regression model that starts with no independent variables in the model. If independent variables meet the *F to Enter* criterion, the strongest are added to the model one at a time. As soon as two or more variables are in the model, they can be removed one at a time if they have F-statistics that are less than the specified *F to Remove* value. This process continues with variables included and considered for exclusion until all of the variables not in the model do not have sufficiently high F-statistics to be added to the model and all of the variables in the model have F-statistics that are too high to allow them to be excluded from the model.

substitution rule. A row in of the Substitute transformer that specifies to the transformer the data column, original value, and substitution value to be used while replacing data values.

sum of squares. A calculation that is a component of *ANOVA* and other statistical techniques. This number is the sum of the squared values of the differences between the observations and their mean. In ANOVA, the sums of squares are used to assign all variation in a dependent variable to one source or another.

table. In a database, a single file consisting of data organized in columns and rows.

text-formatted data. Data that is delimited by standard text separation characters such as tabs and spaces.

time series analysis. A method of studying past events, isolating patterns in past observations, and projecting those patterns into the future. In this type of analysis, a series is a sequence of observations (such as trading days) that span a given amount of time. An example of time series analysis is the price of a company's stock recorded each day the market was open over the past five years.

tolerance. A measure of the extent to which an independent variable is related to other independent variables. If this value is near 0, the independent variable is strongly related to the other independent variables. If the value is near one, the independent variable is not related to the other independent variables. This value can be calculated by the Regression transformer and can be a criterion for variable inclusion or exclusion when the transformer builds a regression model.

tool. A Meta5 desktop program, such as Text, Reporter, or Query. Every tool is represented by an icon.

trading days. The number of days in a period that the business being studied could occur. This information is especially important for studying an activity that can only take place during specific dates or times, such as stock trading. The Seasonality transformer can automatically measure and adjust a time series for the effects of trading days.

transformer. A Meta5 tool that can be used to perform various types of data manipulation either interactively or in a capsule application.

transpose. The operation of interchanging columns and rows in a table of data.

trend. The long-term upward or downward movement of a time series. Under the decomposition framework, trend can be identified and projected into the future using the Forecast transformer. Also, the AutoRegression and AutoCorrelation transformers can remove trend from a series by applying trend differencing.

trend differencing. A method of differencing that removes the effects of trend from a time series. Trend differencing involves subtracting the previous value of a series from the current value to create a differenced series. If a trend is still apparent, the differenced series can be differenced again. The AutoRegression and AutoCorrelation transformers can do trend differencing before doing their analyses.

triple exponential smoothing. A forecasting method that is an extension of the double exponential smoothing model. Together, these smoothing techniques can account for quadratic trends. The values of alpha usually vary between 0.1 and 0.3. Whereas the equation used by the linear smoothing model translates into a straight line, the equation used in the quadratic model translates into a parabola.

t-statistic. A test that determines whether an independent variable in a regression equation is not equal to 0. As this value gets larger, it is more likely that the regression coefficient is not 0 and the independent variable is a significant predictor of the dependent variable. Because this value is the square root of the *F-statistic*, the significance of F is identical to the significance of t. Thus, if the significance of F is near 0, it is very likely that the regression coefficient is not equal to 0.

t-test. A parametric test used to determine whether two paired or independent samples have different means and thus different distributions.

two-tailed significance level. The probability that one observed distribution is not equal to the other sample under consideration. Also called two-tailed probability level.

U test statistic. The test statistic of the Mann-Whitney test of independent samples. As the value of U increases, the probability associated with it decreases. When the probability of U is near 0, the two samples have different distributions.

value. A number used within a table (such as 3,109.00) or in text (such as a year or zip code).

variance. A measure of a distribution's dispersal around a mean.

Wilcoxon signed rank test. The nonparametric test performed by the NPPaired transformer that is used to determine whether two paired samples have different distributions. When the probability associated with this test is near 0, the two samples have significantly different distributions.

W test statistic. The test statistic of the Wilcoxon rank-sum test of independent samples. As the value of W increases, the probability associated with it decreases. When the probability of W is near 0, the two samples have different distributions.

Yates' correction. A method of correction applied by the ChiSquare transformer. Whenever the expected value of crosstab cells are less than five, the resulting chi-square value could be inflated. Yates' correction adjusts for that inflation by subtracting 0.5 from the absolute difference between the expected and the observed cell values.

Z-score. The distance between an observation and the mean as measured in standard deviations. Because this measure is standardized, it facilitates the comparison of one distribution to another. The Regression transformer uses the Z-score as a way of identifying outliers. Synonymous with normal deviate.

Meta5 Publications

This section lists Meta5 publications. To order copies of the books listed here, or to get more information about a book, see your Meta5, Inc. representative.

- **Meta5 Volume 1**

  Getting Started with the Meta5 Developer's Desktop

  PC Integration Tools
- **Meta5 Volume 2**

  Spreadsheet User's Guide

  Plot User's Guide
- **Meta5 Volume 3**

  Text User's Guide

  Layout User's Guide
- **Meta5 Volume 4**

  Capsule User's Guide

  BASIC Tool User's Guide
- **Meta5 Volume 5**

  Data Access Tools User's Guide
- **Meta5 Volume 6**

  Forms User's Guide
- **Meta5 Volume 7**

  Transformers Guide
- **Meta5 Volume 8**

  Error Messages and Codes
- **Meta5 Volume 9**

  Installing Meta5 LAN Components

  System Administration Guide and Reference
- **Meta5 Volume 10**

  Database Gateway Services Guide

  Administering Databases with Meta5
- **Meta5 Volume 11**

  Installing and Configuring Meta5 Open Clients

  Developing Applications with Open Data Access Service
- **Meta5 Volume 12**

Administering Host Services for MVS:DB2 and Cooperative Application Services

Planning and Installing Host-to-LAN Communications for MVS

Index

Special Characters

@-keywords 60
  using in headers 10
@-variables 71, 103, 120, 126, 131, 163
  in Write SQL transformer 209
  using as icon names 10

A

adjusted R2 469
alpha
  in Forecast transformer 431, 433
analysis of variance 212, 218
ANOVA
  input 214, 265
  one-way
    see one-way ANOVA 218
  output 214, 265
  parameters 213
  statistics 218
  three-way
    see three-way ANOVA 218
  two-way
    see two-way ANOVA 218
ANOVA region 459
ANOVA statistics 219
ANOVA test
  and Kruskal-Wallis test 286
ANOVA transformer 211, 212, 218
Append 16, 17
  description 15
  example 17
  input regions 17
  output region 17
  parameters 16
Append Message transformer 101, 103, 132
ARIMA 362
ARIMA model 380
ASCII format conversion 118
autocorrelation 346, 360, 473
  coefficient 346, 363, 380
AutoCorrelation Transformer
  input 349
AutoCorrelation transformer 345, 346, 360
  formulas 360, 362
  output 350
  parameters 347
autoregression
  orders 385
autoregression analysis
  and regression analysis 365
AutoRegression Transformer
  input 368, 369
AutoRegression transformer 345, 365, 379
  formulas 382, 385
  output 370
  parameters 366

B

backward method 471
base line 468
BASIC tool, creating a transformer with 13
beta
  in Forecast transformer 432, 433
beta weight 469
  in Forecast transformer 432
Brown's one-parameter
  linear exponential smoothing 432, 437
  quadratic exponential smoothing 432, 438
button
  Run 2
  Show Controls 2
  Special 2

C

capsule 103, 126, 141, 163, 164
  branching 194, 196
capsule, using a transformer in 9, 13
  @-variables 9
  arrow options 11
  column alignment 9
  data removal from the last input region 9
  positioning
    Clear Contents 24
    Copy Icon 36, 38
  processing sequence 9
cells 232
census X-11 decomposition method 474, 491
checksum value 202
chi-square
  goodness-of-fit test 231, 238, 239, 240
    value 260
chi-square goodness-of-fit test
  formulas 240, 241
chi-square goodness-of-fit value 260
chi-square test 238, 241, 243
  and K-S test 290
  formulas 242, 243
ChiSquare transformer 211, 231
  data columns and rows 244, 245
  grouping features 244
  input region names 234
  output 235
  output region names 234
  parameters 232
Clean 20, 22
  description 15
  example 22
  parameters 20
Clean transformer 172
Clear Contents 23, 31
  benefits 23
  description 15
  output icon 24

- output region 24
- parameters 23
- positioning in a capsule 24
- window 24

coefficient
- of determination 469

Column Name @ Number 500
Column Name $ Expression 500
Column Name! Number 500
Column names 499
comment 504
common data 43, 73
Compress
- benefits 31
- description 15
- example 33
- parameters 33

computed position 506
Concatenate Transformer 104
Concatenate transformer 101
- input region names 105
- output region names 105

concatenating data 104
concordant pair 316
Conditional Clean transformer 101, 106
- input region names 108
- operators 111
- output region names 109
- selection rules 111

constants
- constant value 504

contingency tables 231
controlCode 120
controls area 1, 2
Controls window, Transformer 3
Copy Icon 35, 43
- benefits 35
- description 15
- input region 36
- output icon 36
- output region 36
- parameters 35

copyright messages 141
correlation 309, 315
- coefficient 386
- coefficient r 385

Correlation transformer 345, 346, 385, 394
- data columns 397
- formulas 394, 396
- output 388

count
- in CrossTab transformer 259

cross-tabulation 245, 258
cross-tabulation tables 231, 258
CrossTab Transformer
- input 387, 444, 477

CrossTab transformer 211, 232, 245, 258
- formulas 260, 263
- groupings 248
- input region names 252
- output region names 252
- parameters 250
- statistics computed 245, 247, 259, 260

cycle
- component in time series 521

D

data
- common 43, 73
- partitioning 94

data columns 143, 154
- breaking up 143
- printing 150, 155
- transposing 201, 202

data filtering 106, 110
Data Format transformer 101, 112
- input region names 115
- output region names 115

data rows 143, 147, 150, 154, 172, 177, 181
- printing 150, 155
- removing 172, 177
- selecting 177
- selection rule language 182
- splitting 183, 185
- transposing 201, 202

data types
- in Elementary transformer 400
- in Text To Spreadsheet transformer 199, 201
- in Write SQL transformer 209

data values
- substituting 188

data, selecting 106, 110
database period tables 155
Date Format transformer 101, 121
- input region names 124
- output region names 124
- parameters 122

date formats 121, 124, 137, 163
- alternate 162, 163

date resolution 162
date tables 155, 160
date translation 137
date values
- generating 155, 160

dates 121, 124
- converting 167
- formatting 167
- reformatting 121, 124

debugging 141
degrees of freedom 239
dependent variable 467
DESKTOP keyword 26, 29, 31, 38, 41, 43
differencing 362, 364, 380, 381
dimension, definition 82
discordant pair 316
display area 1, 3
- input region 7
- output region 7

document heading 150
double linear exponential smoothing 432
double moving average 441
Durbin-Watson value 473
dynamic buffer 104

E

Elementary transformer 345, 398, 408
  descriptive statistics of 398
  formulas 409, 411
  input 405
  output 405
  parameters 403
Error Log 1 Transformer
  input region names 128
Error Log 1 transformer 101, 126
  error codes 130
  output region names 129
  parameters 127
Error Log 2 transformer 101, 126
  error codes 130
  parameters 127
error message
  parts 131
error recovery
  Row Select transformer 183
evaluator 132
examples
  changing region names 6
  of parameters 4, 5
exception reporting 106, 110
exponential smoothing 431
Expression 510
expression 132, 511
expression definitions 497
Expression Operator Expression 510
Expression1 ? Expression2
  Expression3 513

F

F-statistic 469, 471
fact, definition 82
find and replace 186, 188
Forecast transformer 155, 345, 411, 428
  and Seasonality transformer 495
  comparison statistics 439, 440
  formulas 433
  input 415, 416
  models 430, 433
  output 417
  parameters 412
  procedure 429
forecasting 365, 379, 411, 428
  Seasonality transformer 474, 489
formatting constants 504
forward method 470
full model method 470
function 507
Function transformer 101, 132, 510
  date translations 137
  input region names 135
  output region names 135
Function transformer parameters 133

G

goodness-of-fit test 231, 238, 239, 240, 290, 299

H

header 183
header buttons 2
Header transformer 101, 137
  input region names 139
  output region names 139
  parameters 138
  text formatting options 140
Henderson's moving averages 441
Holt's two-parameter linear exponential smoothing 432, 438

I

Important Message 143
independent t-test 264, 268, 269
independent variable 467
  in Regression transformer 467
inner join 49, 77
input region 7
installing TXL programs 498
intercept 468
IPIndependent transformer
  input region names 323
  output region names 323
IPMean Transformer
  parameters 454
IPMean transformer 211, 263, 268
  formulas 269, 276
  grouping features 277
  independent t-test formulas 269, 274
  matched pair t-test formulas 274, 276
  output 265
  parameters 277
iterative process 471

J

Join 43, 51
  description 15
  input regions 49
  output region 49
  parameters 43, 49
join types
  left 76
  right 76

K

K-S test, see also Kolmogorov-Smirnov test 300
Kendall's tau correlation 309, 318
  formulas 319
  statistics 316
Kendall's tau correlation formulas 320
Kolmogorov-Smirnov test 290, 299
Kolmogorov-Smirnov test, see also K-S test 300
Kruskal transformer 211, 277, 286
  contrast analysis 290
  decision making 288
  formulas 288, 290
  input region names 279
  output 280
  output region names 280

parameters 278
Kruskal-Wallis statistics 286
Kruskal-Wallis test 277
contrast analysis 287
multiple comparison analysis 287
KSTest transformer 211, 290, 299
input region names 293
one-sample
formulas 301, 303
output 293
output region names 293
parameters 291
two-sample 303
formulas 304, 309

L

Label 56
description 15
example 56
input region 56
Label 52
output region 56
parameters 52
lag 346, 347, 359, 363
least squares 430, 431, 436
left join 47, 76
Lilliefors statistic 291, 301
linear
least squares 430, 431, 435
linear contrast analysis 212, 219
linear contrast expressions 230, 231
linear relationship 468
logic
for Switch and Text Switch transformers 194
long-term differencing 363, 364, 380, 381

M

mailing labels, constructing 52
adding text to an address 52
Mann-Whitney statistics 326, 327
Mann-Whitney test 303, 321, 326
formulas 329, 330
marginal counts 232
marginals 232
matched pair t-test 264, 269
mathematical expressions 132
mathematical functions 507
mathematical operators 509
mean
in CrossTab transformer 259
Merge 58, 71
@-variables 71
description 15
example 71
input regions 67
instructions 61, 67
after change 61, 63
at first row 61, 62
at last row 61, 67
before change 61, 65
every row 61, 64
example 62
guidelines 61, 62
when nonblank 61, 65
output 58
output region 67
parameters 67
templates
creating 60
example 69
templates, creating 61
Message transformer 101, 141
messages 126, 141
copyright 141
moving average 440, 448, 491
formulas 449, 451
moving average ratio 448
Moving Average transformer
formulas 451
output
alignment 452, 453
parameters 441
Moving transformer 345, 440, 448
input 443
multicollinearity 472
MultiJoin 73, 81
description 15
example 81
input regions 78
output region 78
parameters 78
sorting the output 73
multiple linear regression analysis 454, 467
multiple regression models 467
Multiple Select transformer 101, 143
input region names 144
output region names 144
substitution values 143
Multiple Split transformer 101, 147, 150
input region names 148
output region names 148

N

N/A cell 167
naive method II 430, 434
NCorrelation transformer
data columns 320
No-Row Test 512
nominal data 239
non-data values
Error 172
N/A 172
removing rows 172, 177
white space 172
nonparametric
correlation 310, 315
tests 321, 326
nonstationary series 347, 380
normal deviate 469
NPCorrelation transformer 211, 309, 315
data columns 320
grouping features 320, 321
input 311
output region names 311
parameters 321

NPIndependent transformer 211, 321, 326
grouping features 330, 331
output 323
parameters 331
NPPaired transformer 211, 331, 338
grouping features 344
input region names 333
output 333
output region names 333
parameters 331, 344
sign test
formulas 340, 342
Wilcoxon signed rank test
formulas 343
Wilcoxon signed-rank test
formulas 342
null hypothesis 469
numeric constants 504
numeric tolerance 107

O

one-sample K-S test 300, 303
one-sample sign test 339
one-tailed significance level 327, 338
one-way analysis of variance 218
one-way ANOVA 218, 222
formulas 222
table format 222
options
Program Controls 4
Region Names 6
ordinal data 239
outer join 47, 75
outliers 473, 491
output region 7

P

Page Break transformer 101, 150, 155
dimensions 150
document heading 150
input region names 153
output region names 154
parameters
Clean columns 20
Clean rows 20
Column headings 83
Columns to check 20
Columns to combine 32
Columns to compress 52
Columns to delete 33
Columns to keep 92
Common source container 35
Compute 83
examples 4
Facts 83
format 4
Headings to sort 83
Insert template 61
Join columns in first table 73, 74
Join columns in second table 73, 74
Join type 46, 49, 75
inner 49, 77
left 47, 76
outer 47, 75
right 48, 76
Joining columns in first table 43, 45
Joining columns in second table 43, 45
Language 94
Name of container(s) to clear 23
Name of destination folder 35
Name of icon(s) to copy 35
Number of columns per page 52
Number of heading rows 16, 20, 45, 52, 61, 74, 94
Number of labels per column 52
Number of lines per label 52
Number of partitions 94
Number of rows to output 94
Overwrite? Y/N 35
Rank columns 94
Replacement for missing values 83
Replacement value for nulls 49
Row headings 83
Rows to combine 32
Rows to delete 32
Section headings 83
Sort columns 94
Value of blanks, N/As, and errors 94
sorting 94
PARENT keyword 26, 27, 29, 38, 40, 41
partial autocorrelation 346, 347
partial regression residuals 473
partitioning data 94
path names, entering 26, 31, 38, 43
searching
current level 26, 27, 38, 40
desktop level 29, 31, 41, 43
parent level 27, 29, 40, 41
pattern definition 182
Pearson correlation 315
percentage statistics
in CrossTab transformer 259
period aggregations 448
Period Table transformer 101, 111, 155, 160, 164
input region names 159
output region names 159
parameters 156
period tables 155, 160
Pivot 82, 90
description 15
input regions 89
column order 89
guidelines 90
input data 89
row order 89
output region 89
Plot tool
and Regression transformer 454
Post Message transformer 101, 132, 163, 164
input 163
output 163
pound sign 189
predictor variables
in Regression transformer 467

printing
- data columns 150, 155
- data rows 150, 155
- Query tool 150, 155
- Spreadsheet 150, 155
- SQL Entry icon 150, 155

product-moment correlation 315
Program Controls option 4

Q

Query tool
- printing 150, 155
- with Elementary transformer 398

R

R2 value 469
Random Number
- output region names 166

Random Number transformer 101, 164, 166
- input region names 165
- parameters 165

random numbers 164
randomness
- component in time series 429, 521

ranking columns 94
record formatting 119
record-oriented I/O 119
reformatting dates 121, 124
region names 6
regression
- analysis 454, 467
- equation 468

Regression transformer 345, 454, 467
- input 458
- output 458, 460
- parameters 454
- specifying columns 474
- statistics and concepts 468, 470

removing rows with non-data values 172, 177
Replace transformer 101, 166, 172
- input region names 170
- output region names 171
- parameters 169

reportKey 120
reportKey code 120
residual mean squared (RMS) error 469
residuals 467
right join 48, 76
rolling sum 440, 448
- formulas 451
- statistics 448

Row Clean
- output region names 176

Row Clean transformer 101, 172, 177
- input region names 176
- parameters 173

Row Select
- input region names 179
- output region names 179

Row Select transformer 101, 177, 181
- error recovery 183
- parameters 178

rowNumber 120
rowNumber code 120
Run button 2

S

S-curve least squares 436
search and replace, see find and replace 188
seasonal
- component in time series 429
- differencing 363, 364, 380, 381
- effect 363, 380, 381

seasonality 440, 474, 489
- component in time series 429

seasonality component
- see also seasonal 521

Seasonality transformer 155, 345, 415, 416, 474, 489
- and Forecast transformer 495
- and Moving Average transformer 440
- formulas 492, 495
- input 478
- methods 490, 492
- output 479
- statistics 491, 492

see one-way ANOVA 218
seed value 164, 165
Select 92, 93
- description 15
- example 93
- input region 93
- output region 93
- parameters 92

Select transformer 143
selecting data 106, 110
selecting data rows 177
selection rule language 182
selection rules 106, 111
serial correlation, see autocorrelation 346
short-term
- differencing 363, 364, 380, 381
- forecast 431

short-term differencing 363, 364
Show Controls button 2
sign test 331, 338
- statistics 338, 340

sign test statistics 339
significance level 316
- of correlation coefficient 386

single exponential smoothing 432, 437
single moving average 441
slope 468
Sort 94, 99
- description 15
- input region 99
- output region 99
- parameters 94

Spearman rank correlation 309
- formulas 318, 319
- statistics 315

Spearman rank correlation statistics 316, 317
Special button 2
Spencer's 15-term moving average 441
Split Header transformer 101, 147, 183, 185

header 183
input region names 184
output region names 184
Split Header transformer parameters 183
Spreadsheet 106, 117
printing 150, 155
SQL
code conversion 119, 120, 204, 208
control codes 120
syntax support 209
SQL Entry icon
printing 150, 155
standard deviation 341
stationary
time series 362, 366, 380
time series data set 347
stationary series, see stationary time series 362
stepwise selection method 471
Substitute transformer 101, 186
input region names 187
output region names 187
parameters 186
substitution rules 188
substitution values 143, 144
sum
in CrossTab transformer 259
Switch transformer 101, 194
input region names 195
logic 194, 195
output region names 195

T

t-statistic 263, 469
t-test
and K-S test 290
tau correlation
see Kendall's tau correlation 317
terminating a capsule 142
text constants 504
Text Switch transformer 101, 194, 196
input region names 195
logic 194
output region names 195
parameters 194
Text To Spreadsheet transformer 101, 196, 199
data type conversion 199, 201
input region names 198
output region names 198
parameters 197
three-way analysis of variance, see three-way ANOVA 218
three-way ANOVA 218, 219, 228
time series analysis 474, 489
autocorrelation 346, 360
autoregression 365, 379
time-series analysis transformers 155
trading days 474, 491
transformer
Append 15, 16, 17
Clean 15, 20, 22
Clear Contents 15, 23, 31
Compress 15, 31, 33
configuring a blank 13
Copy Icon 15, 35, 43
in a capsule 9
Join 15, 43, 51
Label 15, 52, 56
Merge 15, 58, 71
MultiJoin 15, 73, 81
Pivot 15, 82, 90
Select 15, 92
Sort 15, 94, 99
Transformer Controls window 3
Transformer Execution Language 132, 497, 510
transformers
controls area 1, 2
display area 1, 3
moving data from 8
moving data into 8
preconfigured 5
table requirements 8
Transpose transformer 101, 201, 202
input region names 201
output region names 201
trapping errors 126, 130
trend
component in time series 429, 521
detecting 380
differencing 363, 364, 380, 381
triple exponential smoothing 432
two-dimensional cross-tabulation 245, 258
two-sample sign test 331, 339, 340
two-tailed significance 327
two-tailed significance level 328, 338, 340
two-way analysis of variance, see two-way ANOVA 218
two-way ANOVA 218, 219, 223, 228
formulas 228
TXL 497, 510
TXL elements and syntax 498
TXL specification 497

V

variable 511
dependent 467
independent 467
variable assignment 511

W

Wilcoxon rank-sum test 321, 326
formulas 328
statistics 326, 327
Wilcoxon signed-rank test 331, 338
statistics 339, 340
windows
controls area 1, 2
display area 1, 3
header buttons 2
Transformer Controls 3
Word Count transformer 101, 202, 203
definitions 203
input region names 203
output region names 203
Write SQL transformer 101, 204, 208
and Data Format transformer 119

data type handling 209
input region names 206
output region names 207
Write SQL transformer parameter 204

Y

Yates' correction 243

Z

Z-score 469
zero value 173, 412