Paper SAS257-2014

Managing Missing Data Using SAS® Enterprise Guide®

Elena Shtern and Matt Hall, SAS Institute Inc., Arlington, VA

ABSTRACT

Missing data is an ever-present issue, and analysts should exercise proper care when dealing with it. Depending on the data and analytical approach, this problem can be addressed by simply removing records with missing data. However, in most cases, this is not the best approach. In fact, this can potentially result in inaccurate or biased analyses.

The SAS® programming language offers many DATA step processes and functions for handling missing values. However, some analysts might not like or be comfortable with programming. Fortunately, SAS® Enterprise Guide® can provide those analysts with a number of simple built-in tasks to discover missing data and diagnose their distribution across fields. In addition, various techniques are available in SAS Enterprise Guide to impute missing values, varying from simple built-in tasks to more advanced tasks that may require some customized SAS code. The focus of this presentation is to demonstrate how SAS Enterprise Guide features such as Data Exploration, Query Builder, Summary Statistics, Standardize Data, and Create Time Series address missing data issues through the point-and-click interface. As an example of code integration, we demonstrate the use of a code node for more advanced handling of missing data. Specifically, this demonstration highlights the power and programming simplicity of PROC EXPAND (SAS/ETS® software) in imputing missing values for time series data.

INTRODUCTION

Anyone who ever worked with data in a business setting can testify that clean and ready-to-use data remains an abstract concept. Issues vary across data sets and include incomplete data, invalid values, censored data, and missing values, just to name few. While each of those issues is important and requires analysts' attention, *missing data* is likely to be the most common issue across data sets.

Whether a data user is an experienced statistician or business analyst, he or she must be able to assess the prevalence of missing data and to identify appropriate methods to address it. The purpose of this paper is to demonstrate how easily business analysts can assess and handle missing data in the SAS Enterprise Guide application.

DEMO DATA SET

For the purposes of this demonstration, the data set RETAIL in the SAS Library SASHELP[1] has been modified using the following code:

```
data retailTrain;
set sashelp.retail;
if year in (1980, 1984, 1992) and month in (4) then do;
SalesTrain = .;
Infilled = 1;
end;
else do;
SalesTrain=Sales;
Infilled=0;
end;
run;
```

The resulting data set RETAILTRAIN has three records with missing values. Note that the three records have been replaced with a '.', the SAS standard for missing numeric data. Missing character data is replaced with a space by SAS. Although there are some similarities in how SAS treats missing numeric and character data, we will focus on numeric data in this paper.

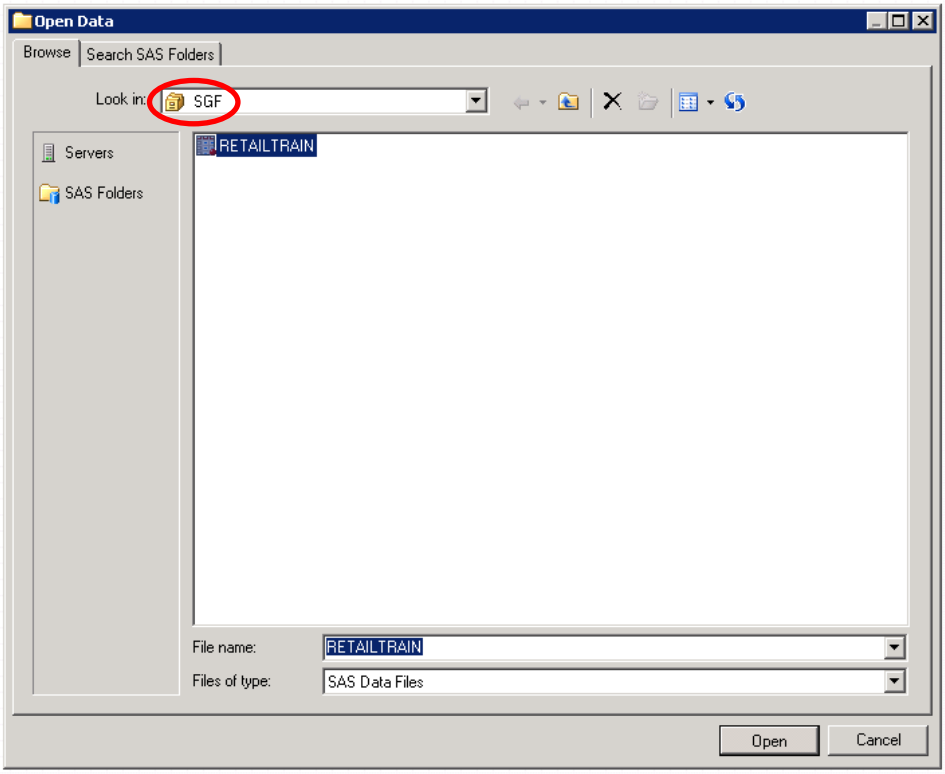
[1] The library SASHELP is installed by default with SAS Foundation. For details, see http://support.sas.com/documentation/onlinedoc/guide/tut51/en/m2_4.htm

HOW TO ASSESS MISSING DATA IN SAS ENTERPRISE GUIDE

SAS Enterprise Guide offers several ways to assess the prevalence of missing data as well as examine missing data. This section describes some of these tools that are ready to use without having to write a single line of code.

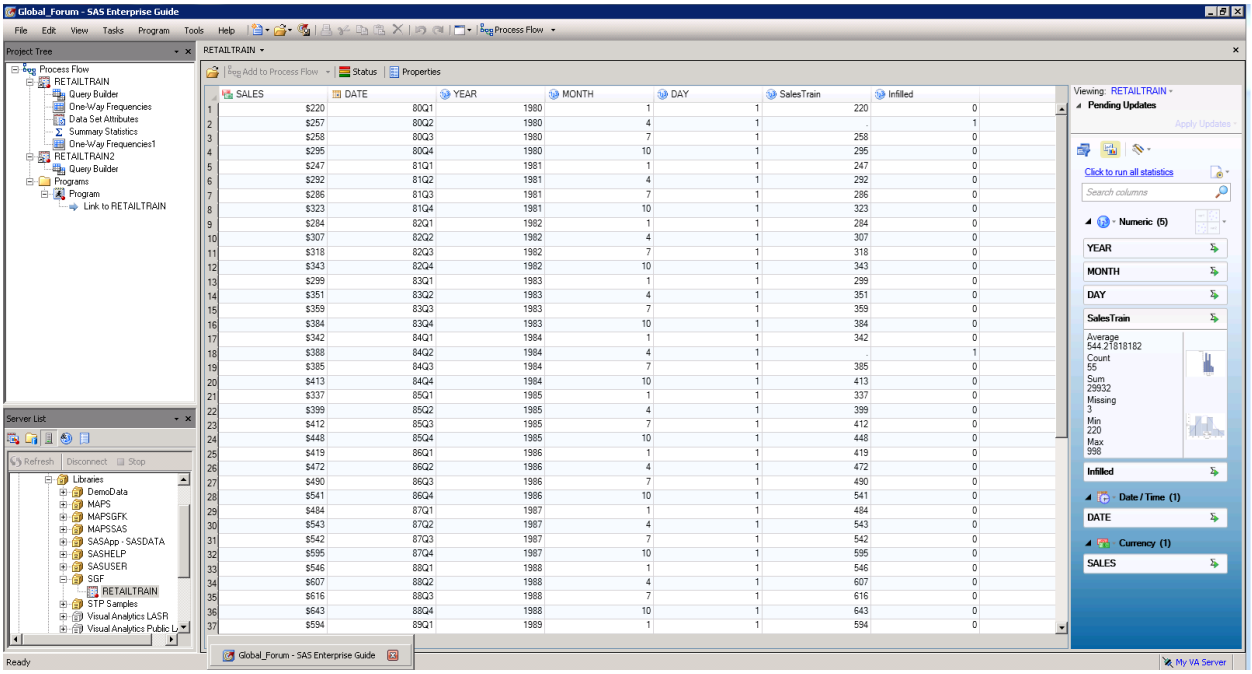
ASSESSING MISSING VALUES USING A DATA EXPLORATION

Beginning with SAS Enterprise Guide 5.1, SAS users have been able to explore a data set before making it part of a SAS Enterprise Guide project. To open a data exploration, select **File->Open->Data Exploration** and select the RetailTrain data set from a SAS library. Note that the data set must be in a SAS library for the data exploration task to work; in the example below, the RetailTrain data set is in the SGF library (red circle) in the Open Data window. Select **Open** to bring the data into the exploration.



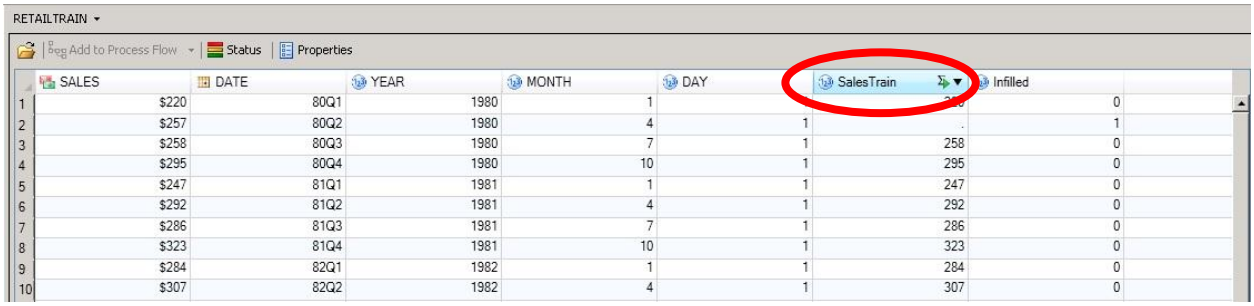
Display 1: Select library for data set to use in data exploration

In the SAS Enterprise Guide process flow window, the data set appears as shown below. Keep in mind that SAS Enterprise Guide will not make this data set part of your Process Flow until you want it; you are simply viewing the data and doing some basic manipulations. For our purposes, one of the observations provided by the exploration is the number of missing values.



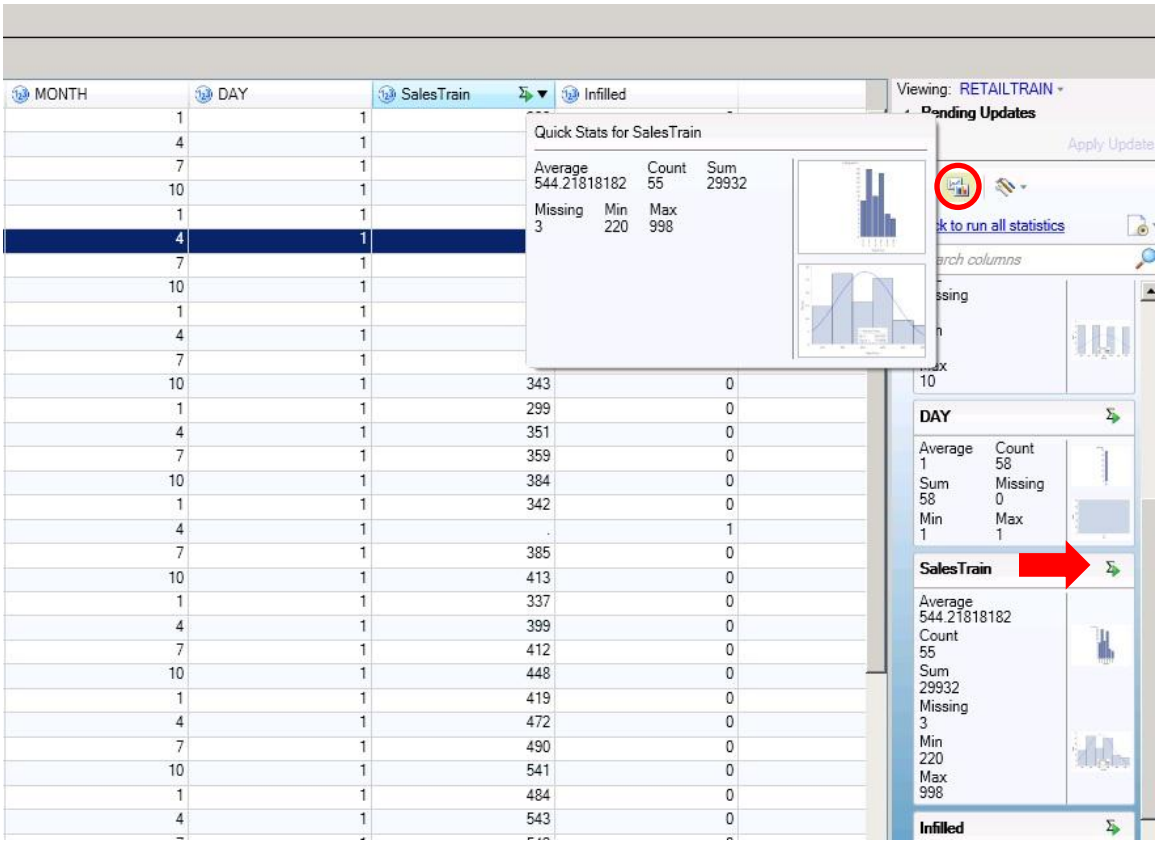
Display 2: Data Exploration window

Hovering over the column heading for the variable of interest will expose the summary statistics icon as shown below.



Display 3: Summary Statistics icon

Clicking on the icon will bring up a Quick Stats window for the variable, including the number of missing values (Missing).

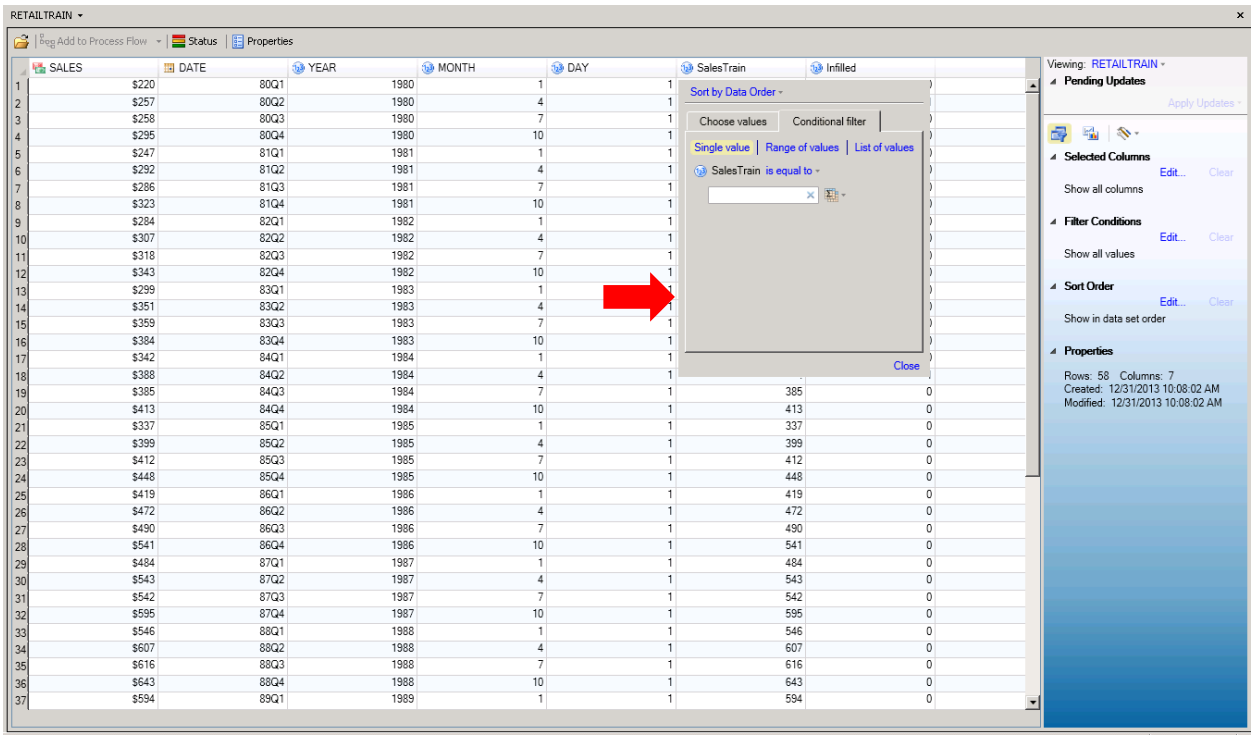


Display 4: Quick Stats window displaying number of missing values

Note that the same statistic appears in the right pane. This is accomplished by selecting the Quick Stats button (circled in the image above) and then selecting the Quick Stats icon next to the variable of interest (arrow in the image above).

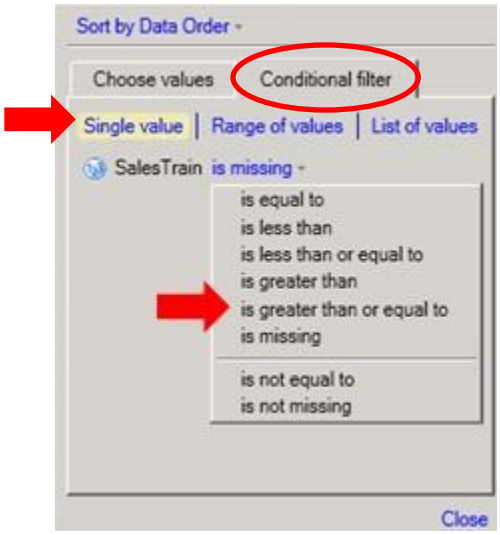
Once you know the number of missing values for variables, discovering which rows contain missing values may be of interest. This is especially true if missing values will significantly impact your results, as in the example data set. Since the data are quarterly sales totals, a single missing value can have a significant impact within a year or series of years. The good news is that you can use the same SAS Enterprise Guide exploration to filter for observations containing missing values.

Clicking on the column header for the variable of interest (continuing with SalesTrain for this example) opens the Filter and Sort window (see below). In that window, under the Choose Values tab, select SalesTrain from the list of variables.



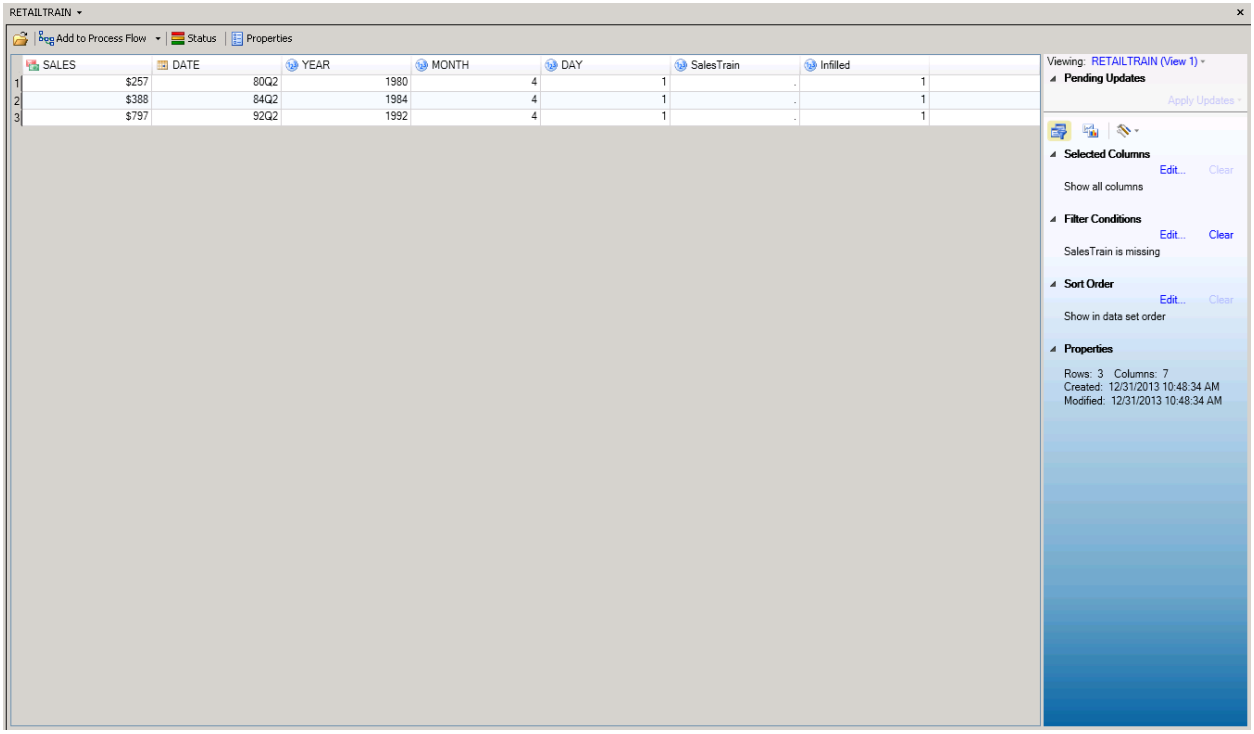
Display 5: Filter and Sort window in Data Exploration

Then move to the Conditional Filter tab, where you can choose to filter on a Single value, Range of values, or List of values. For our purposes, a single value filter will be used for SalesTrain. By clicking on the small arrow next to the variable, a number of conditional filters can be selected, including “is missing” (see below).



Display 6: Conditional Filter tab in Filter and Sort window

The filter runs and shows the observations where SalesTrain is missing.



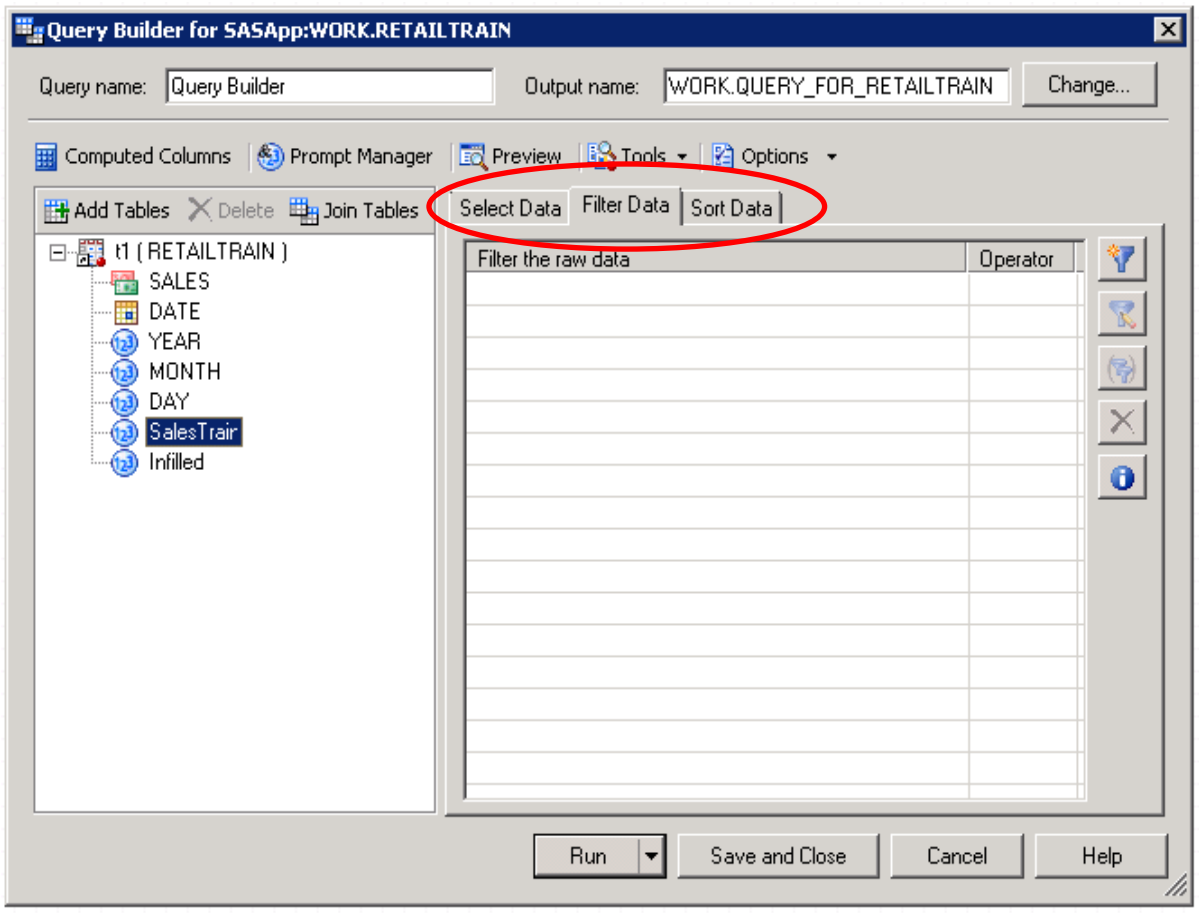
Display 7: Results showing observations with missing values for SalesTrain

Alternatively, if you wish to eliminate rows with missing data for SalesTrain, choose “is not missing” from the Conditional Filter tab.

ASSESSING MISSING VALUES USING THE QUERY BUILDER

Once data are part of a SAS Enterprise Guide project, a number of tools are available to manipulate data just as easily as in a Data Exploration. One of the most useful and flexible tools is the Query Builder. The Query Builder does exactly what the name implies; constructs queries using specific data selections, filters, and sorts that you select. The Query Builder allows you to filter the data set based on a variety of criteria, including missing values.

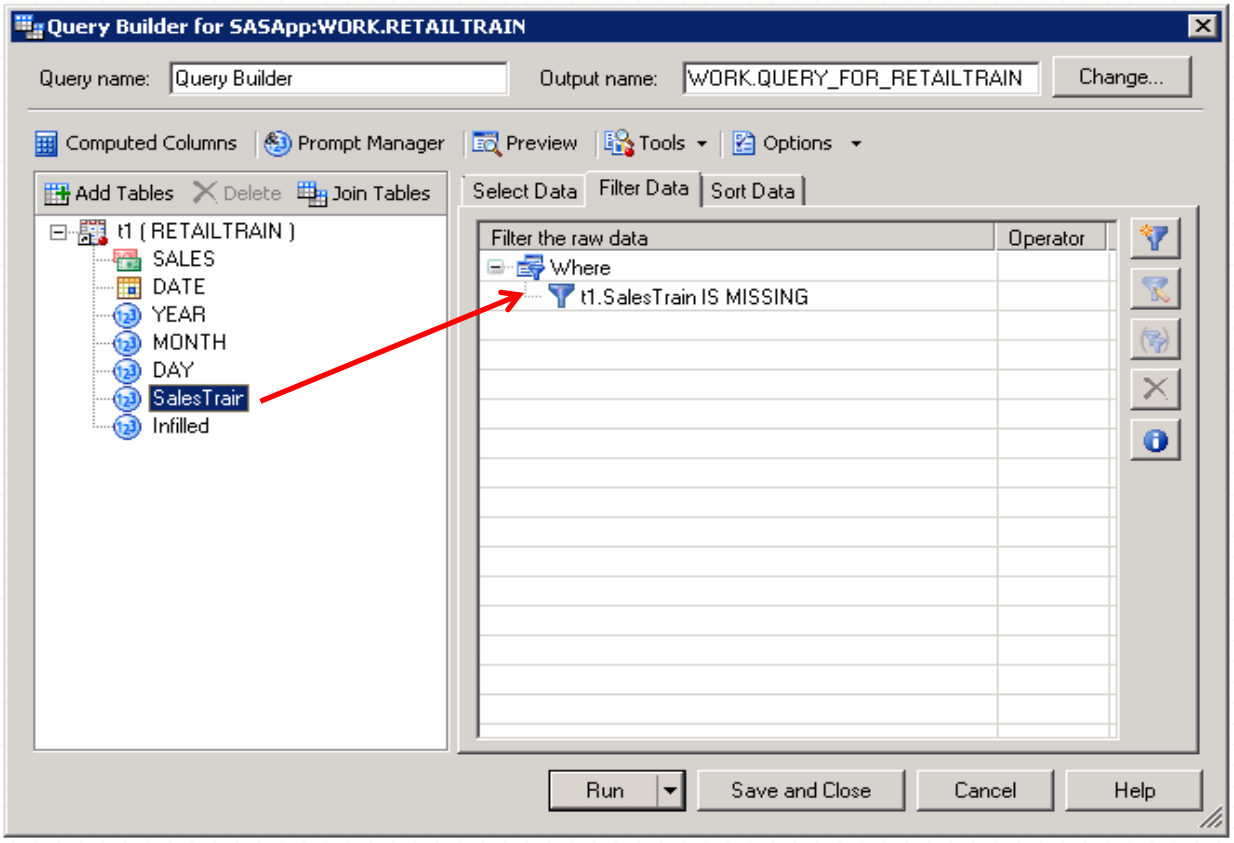
When you open the Query Builder from an open data set in SAS Enterprise Guide, the following window appears:



Display 8: Enterprise Guide Query Builder

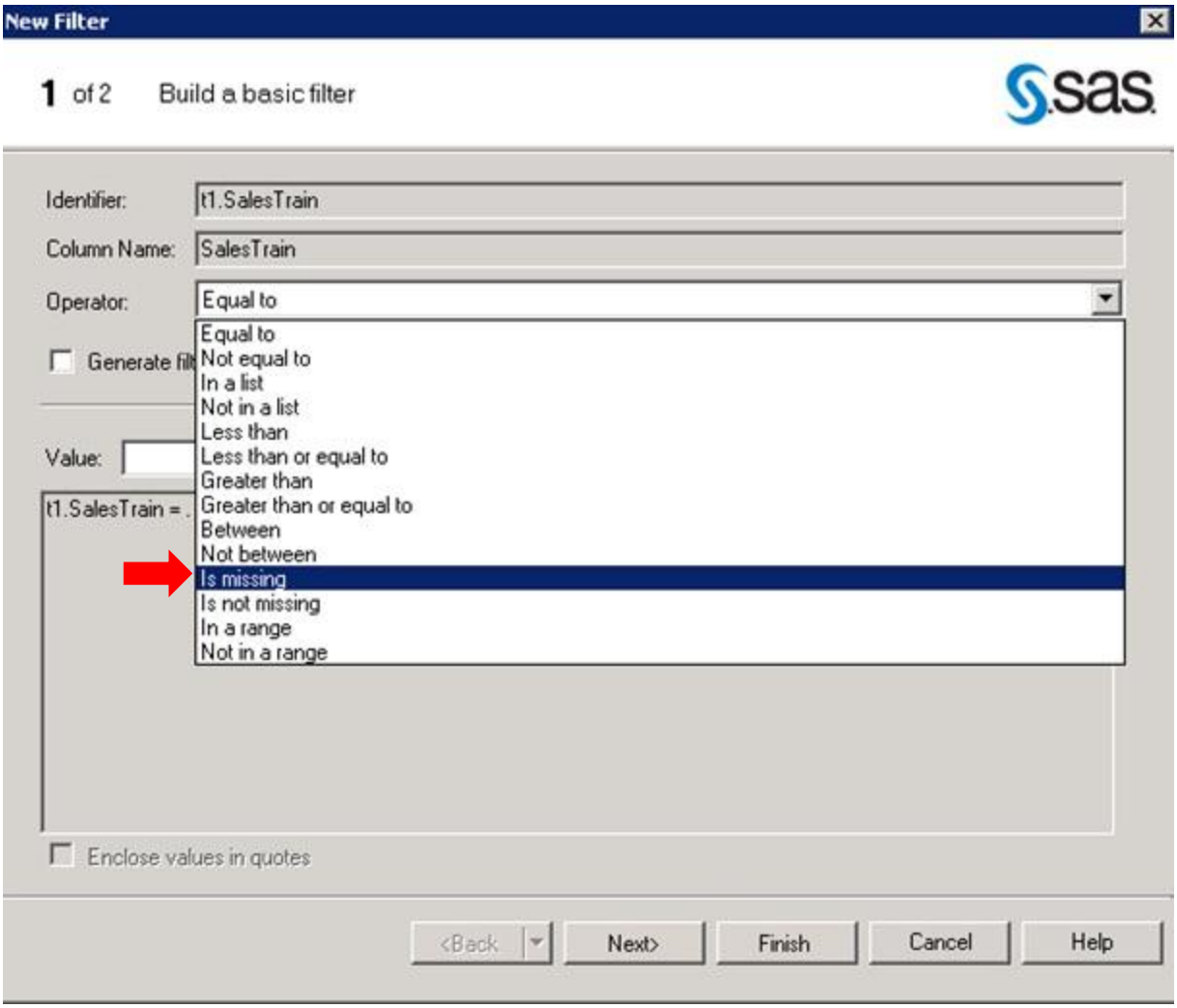
The three tabs indicated in Display 8 provide different building blocks of a SAS query: **Select Data**, **Filter Data**, and **Sort Data**.

First, you must select the variables you wish to include in your output data set. You must move at least one variable from the list on the left into the Select Data tab. Next, you can select the variable you want to filter on, in this case SalesTrain, and move it into the Filter tab.



Display 9: Select Data in the Query Builder

At this point, a New Filter window appears as shown in Display 10 below. From here, simply selecting “Is missing” as the operator will output a SAS data set that only contains rows where the value of SalesTrain is “.”.



Display 10: New Filter window

Finally, your query will look like the table in Output 1 below. When the query is run, the resulting data set contains the three rows where the value of SalesTrain is missing. Since the data are quarterly sales records, three missing values could have a high impact on any subsequent analysis.

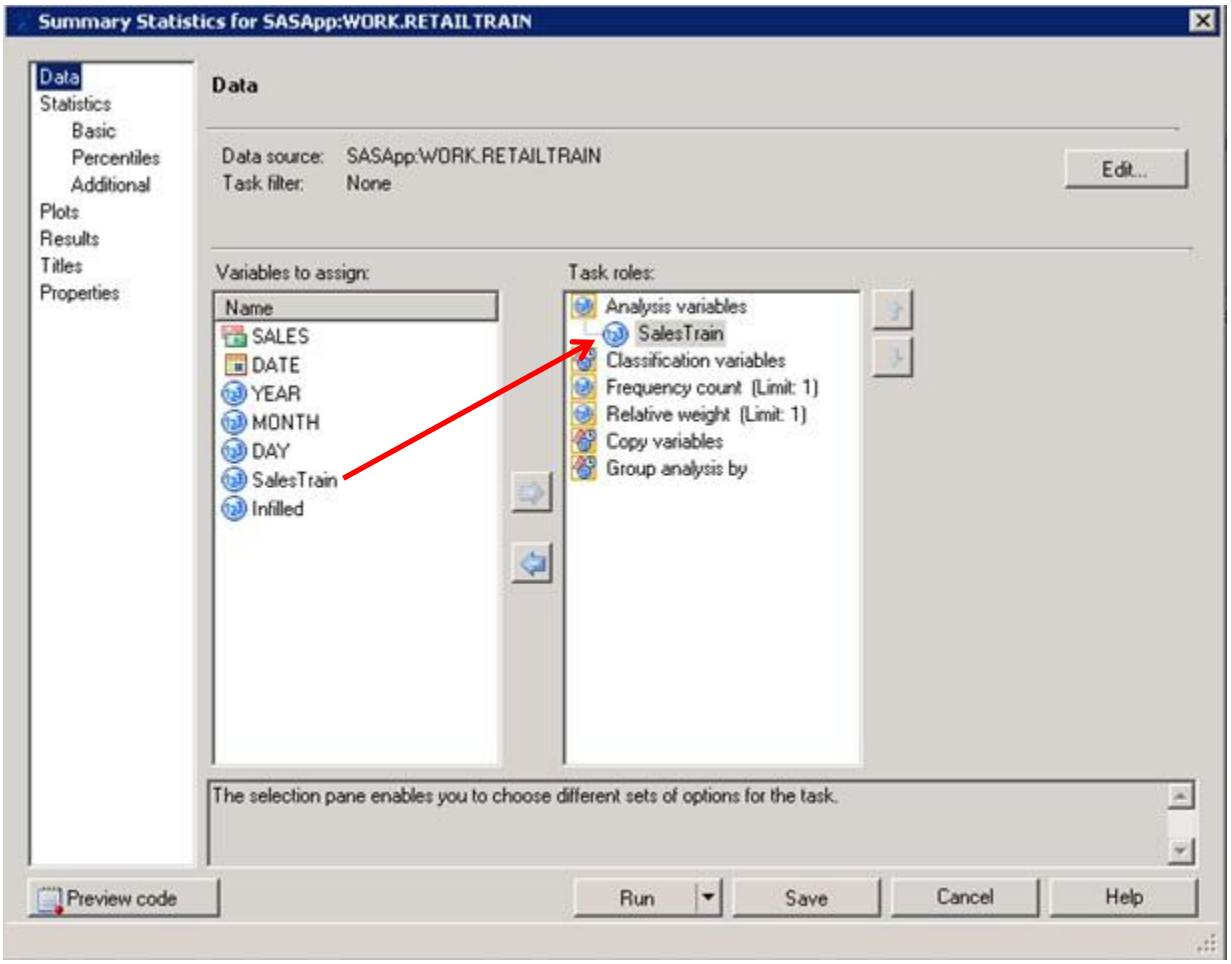
	SALES	DATE	YEAR	MONTH	DAY	SalesTrain	Infilled
1	$257	80Q2	1980	4	1	.	1
2	$388	84Q2	1984	4	1	.	1
3	$797	92Q2	1992	4	1	.	1

Output 1: Results showing observations with missing values for SalesTrain

ASSESSING MISSING VALUES USING SUMMARY STATISTICS

Another way to determine how many missing values are in a data set that is part of a SAS Enterprise Guide project is to use the Summary Statistics task. The Summary Statistics task is most useful for determining the number of missing values for a variable if the release of SAS Enterprise Guide is older than 5.1 (in which case, Data Exploration is not available). Also, when reporting summary statistics as part of a SAS Enterprise Guide business intelligence project, the number of missing values for any variable can be output to the report.

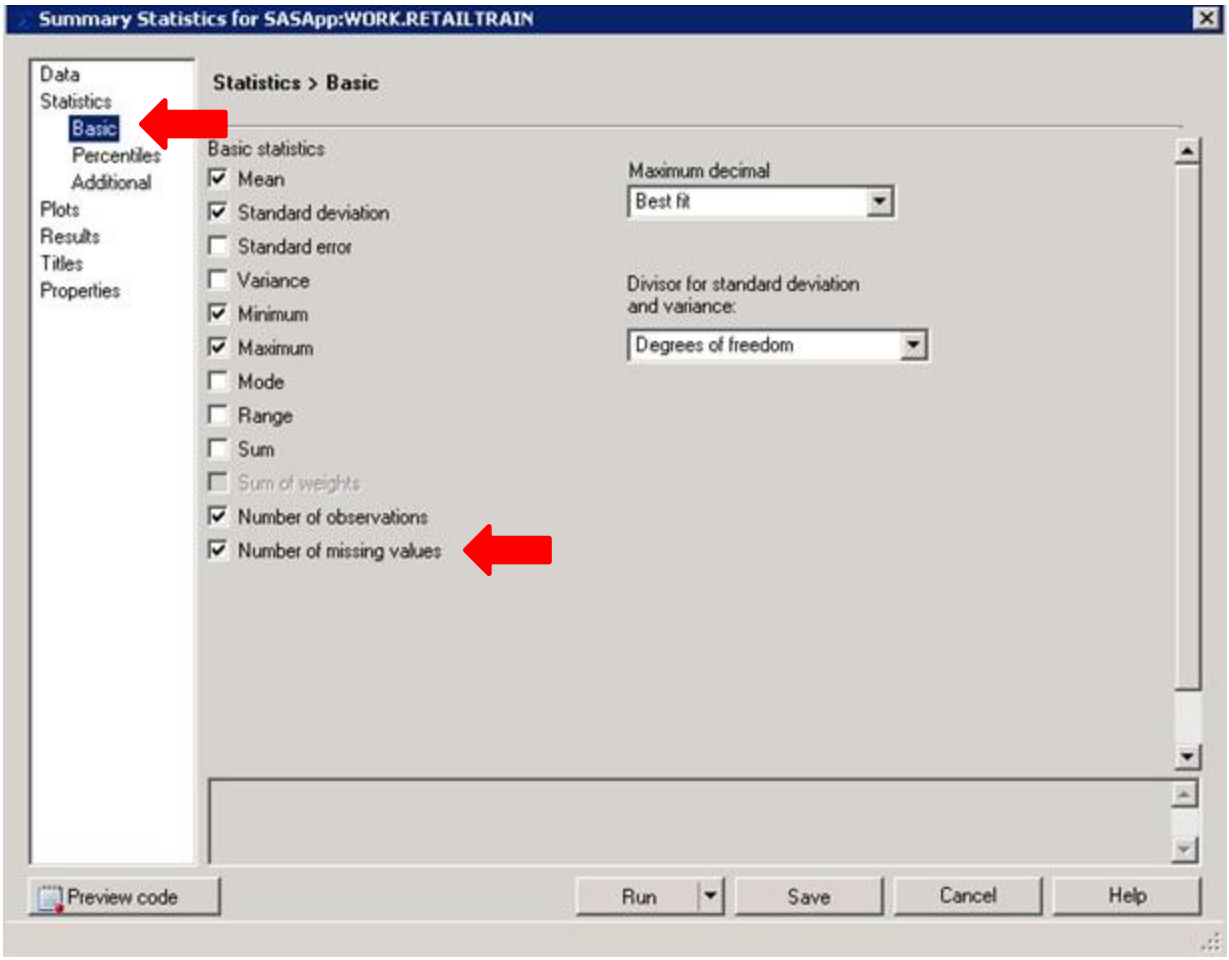
To report the number of missing values as part of a summary statistics report, bring the data set into the project window. Then, select Summary Statistics under Describe in the tool bar or under Describe in the Tools menu from the menu bar. The Summary Statistics window opens as shown in Display 11 below.



Display 11: Results showing observations with missing values for SalesTrain

With Data selected in the left pane of the Summary Statistics window, drag the variables of interest from the **Variables to assign** pane to the **Task Roles** pane under Analysis variables. In the image above, the variable SalesTrain has been selected as the analysis variable.

Next, select **Basic** under Statistics in the left pane. Be sure to select the **Number of missing values** check box (it is not selected by default).



Display 12: Selecting missing values in Summary Statistics window

When the Summary Statistics task is run, the report includes the number of missing values (N Miss) for the selected variables as shown Output 2. Note that the number of values (N) for the variable, in this case 55 for the variable SalesTrain, does not include missing values. Missing values are dropped for summary statistics calculations.

Summary Statistics for Retail Train Dataset

The MEANS Procedure

Analysis Variable : SalesTrain					
Mean	Std Dev	Minimum	Maximum	N	N Miss
544.2181818	213.4069049	220.0000000	998.0000000	55	3

Output 2: Results showing number of missing values as part of summary statistics

HOW TO TREAT MISSING DATA IN SAS ENTERPRISE GUIDE

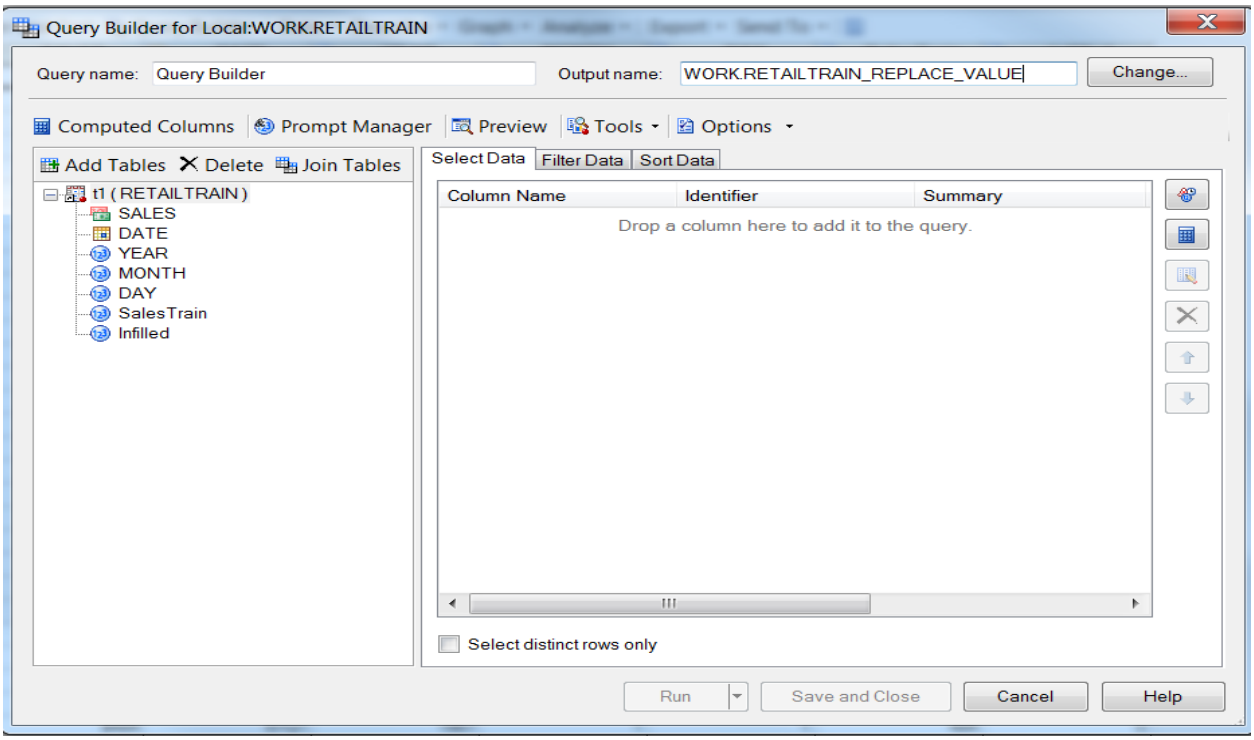
Removing entire records with missing data in one or more fields might be appropriate in certain situations. However, in most cases the records are likely to contain important information in other non-missing fields and should be

preserved. This section discusses four different techniques, ranging from simple to more advanced, for treating missing data using SAS Enterprise Guide.

REPLACE MISSING WITH USER SPECIFIED VALUES

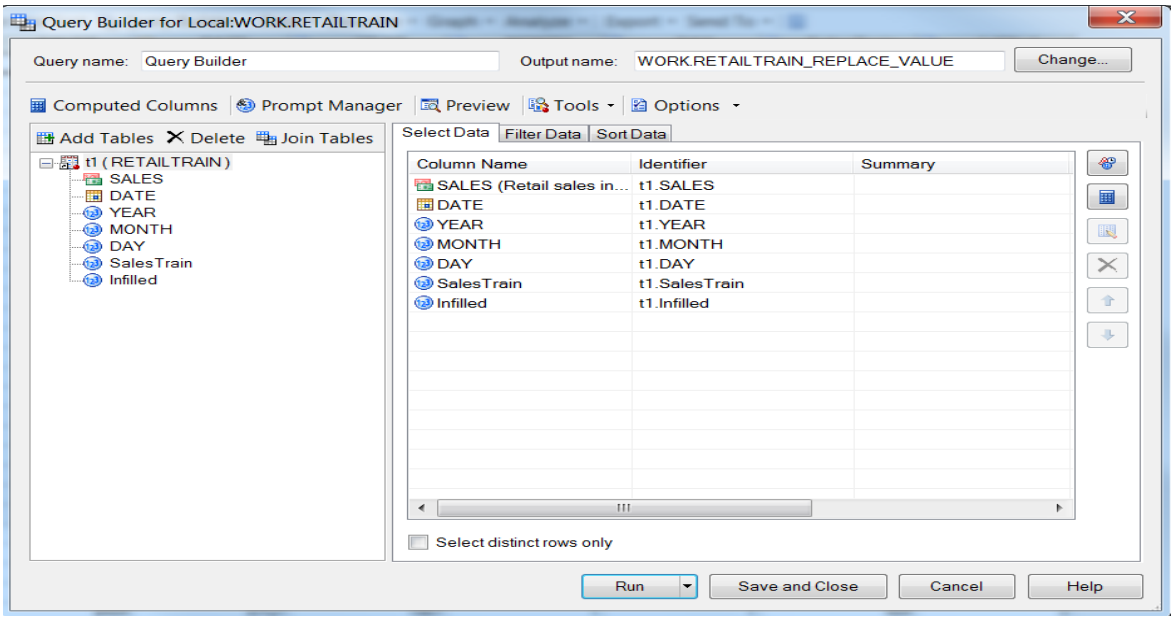
With the help of the Query Builder, missing values can be easily replaced with any value. When using this approach, it is expected that analysts have a good understanding of data, and can support the choice of the replacement value.

When the Query Builder is invoked, the following window appears:



Display 13: Query builder window

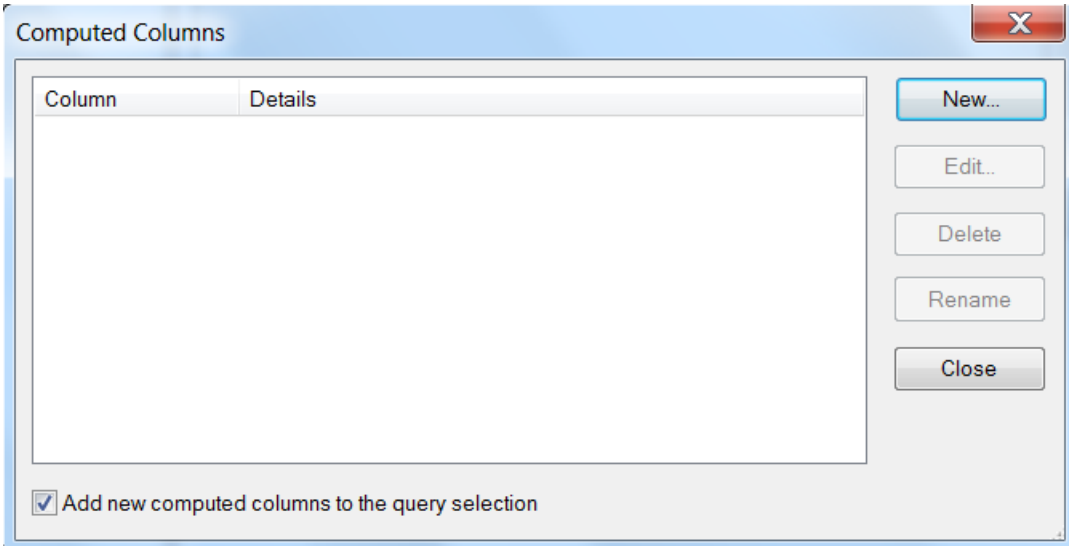
You should select the fields from the original data set that you want to retain in the new data set. To accomplish it, you should be in the **Select Data** tab and move the desired fields into the right panel as shown below.



Display 14: Query builder window with selected variables

The next step is to create a new field SalesTrainReplaced that will be equal to non-missing values of the field SalesTrain, and will be equal to 200[2] when the field SalesTrain is missing. This can be accomplished through the use of the **Computed Column** feature of the Query Builder.

To invoke the **Computed Column** feature, click on the **Computed Column** icon in the top-left panel of the Query Builder window. A new window appears on the screen.



Display 15: Computed Columns Window

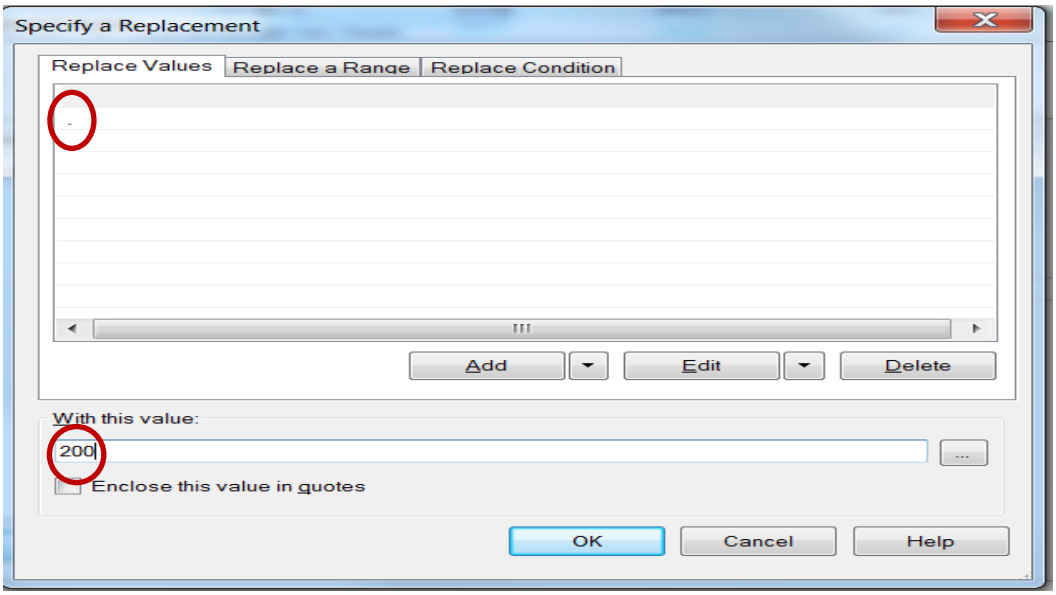
Click the **New** button, and the **New Computed Column Wizard** opens. Follow the steps below to create a new computed field.

Step 1. Select the **Recorded Column** type.

[2] The value '200' was arbitrarily chosen.

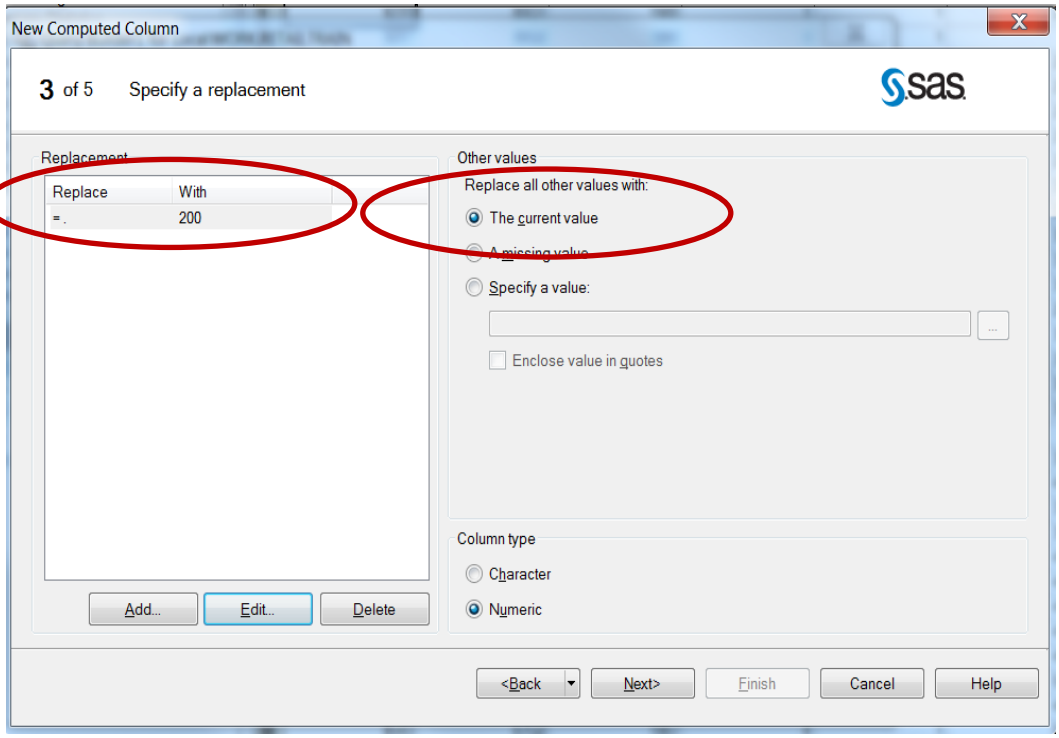
Step 2. Select a column with missing values you want to replace. In our example this is the SalesTrain field. Click **Next**.

Step 3. Select the **Replace Value** tab. Click **Add** in the lower-left corner, and type '.' and the replacement value in the provided spaces as shown below. Click **OK** when done.



Display 16: Specify a replacement value window

Now Step 3 **New Computed Column** wizard window looks like this:



Display 17: Step 3 of 5 of New Computed Column Wizard

Step 4. Provide name and identifier for the new computed field. Click **Next**.

Step 5. Review the summary of selected choices and click **Finish**.

The new data set SalesTrain_Replace_Value contains the same number of records and the same fields as the original data set plus one additional field SalesTrainReplaced.

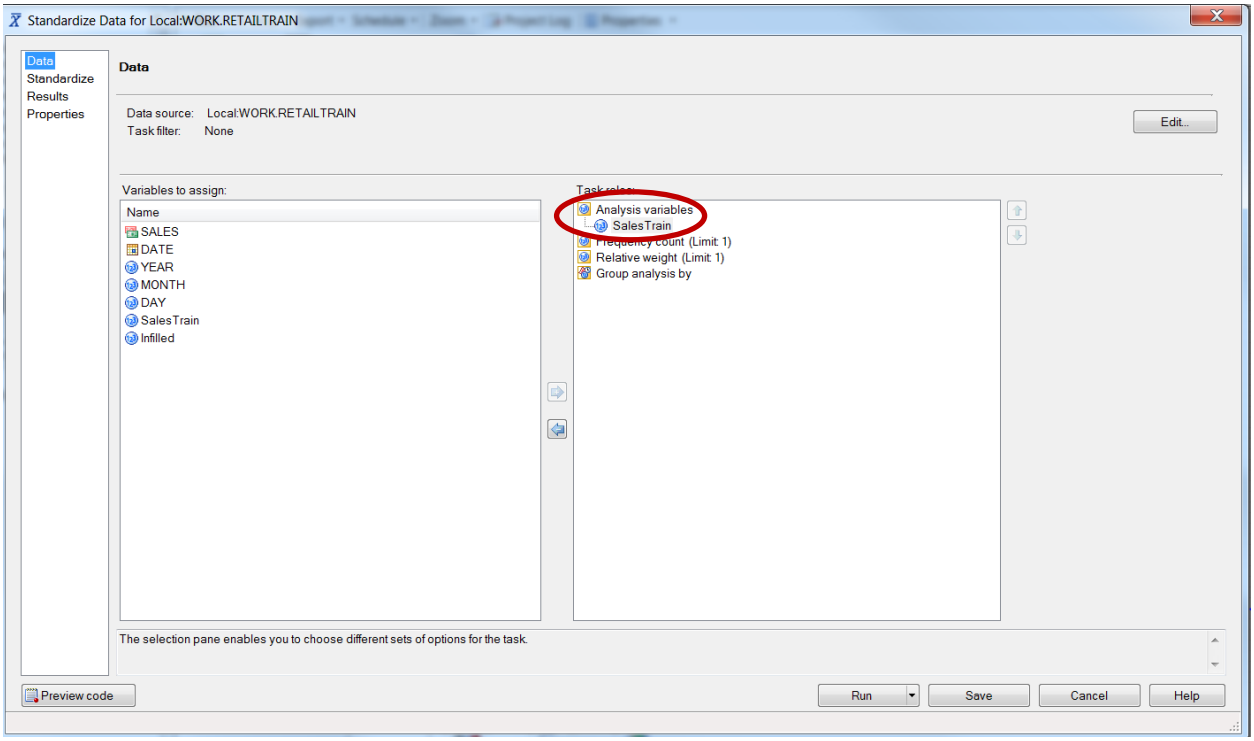
	SALES	DATE	YEAR	MONTH	DAY	SalesTrain	Infilled	SalesTrainReplaced
1	$220	80Q1	1980	1	1	220	0	220
2	$257	80Q2	1980	4	1	.	1	200
3	$258	80Q3	1980	7	1	258	0	258
4	$295	80Q4	1980	10	1	295	0	295
5	$247	81Q1	1981	1	1	247	0	247
6	$292	81Q2	1981	4	1	292	0	292
7	$286	81Q3	1981	7	1	286	0	286
8	$323	81Q4	1981	10	1	323	0	323
9	$284	82Q1	1982	1	1	284	0	284
10	$307	82Q2	1982	4	1	307	0	307
11	$318	82Q3	1982	7	1	318	0	318
12	$343	82Q4	1982	10	1	343	0	343
13	$299	83Q1	1983	1	1	299	0	299
14	$351	83Q2	1983	4	1	351	0	351
15	$359	83Q3	1983	7	1	359	0	359
16	$384	83Q4	1983	10	1	384	0	384
17	$342	84Q1	1984	1	1	342	0	342
18	$388	84Q2	1984	4	1	.	1	200

Table 1: Updated table

REPLACE MISSING WITH STANDARDIZED VALUES

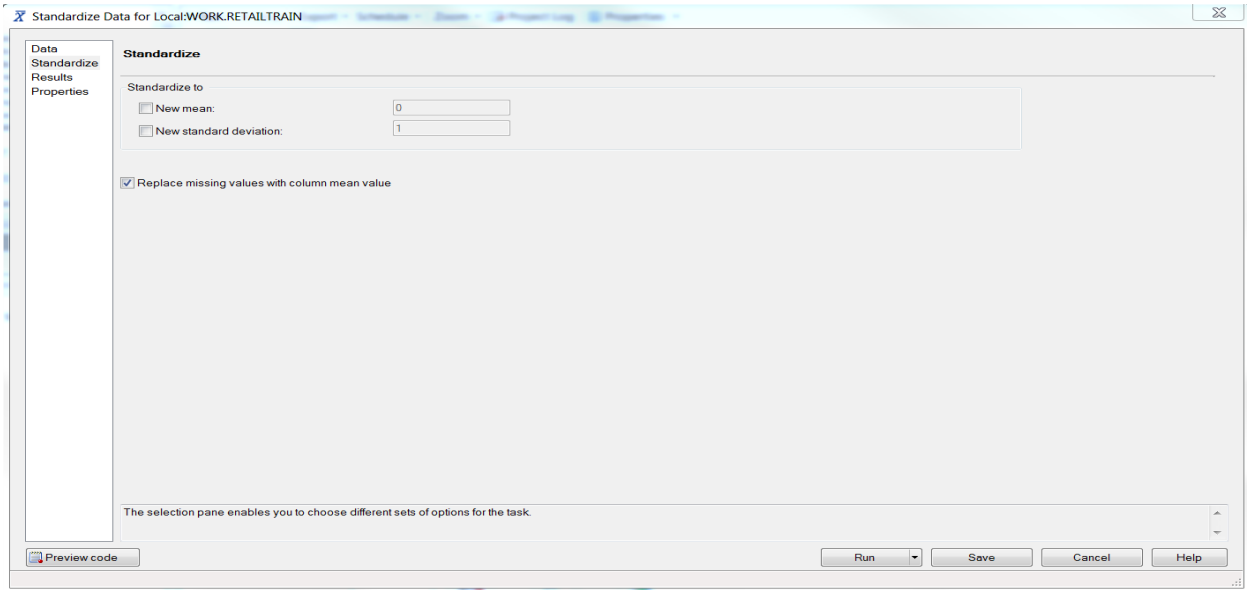
In some situations analysts might find it appropriate to replace missing data with the average value of this field. This can be easily accomplished in SAS Enterprise Guide using the **Standardize Data** functionality as described below.

Once you open the original data set in the SAS Enterprise Guide window, click the **Data** tab and select **Standardize Data**. The **Standardize Data** window will appear on the screen. Select SalesTrain as Analysis variable.



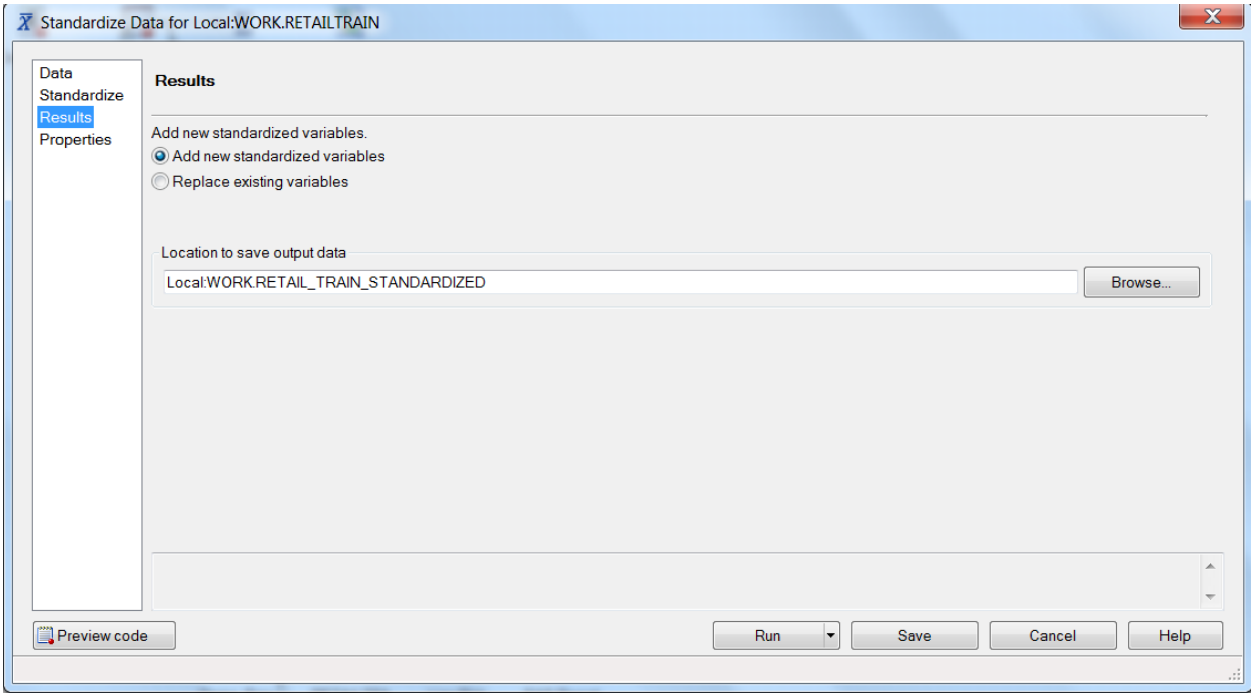
Display 18: Data tab window

Next, click **Standardize** in the selection pane on the left and select the **Replace missing values with column mean value** check box as shown below.



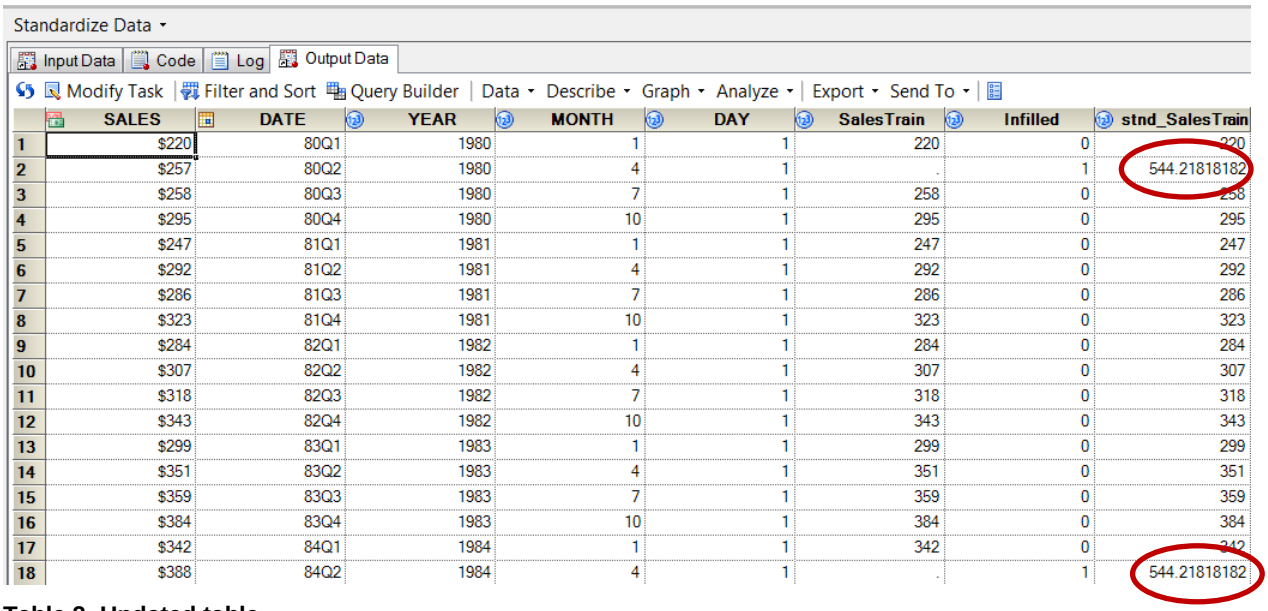
Display 19: Standardize tab window

To overwrite a system-generated name for the output data set, click **Results** in the selection pane on the left and type the name in the provided space. Click **Run**.



Display 20: Results tab window

The new data set RetailTrain_STANDARDIZED has the same number of records and the same number of fields as the original data set with the exception of one additional variable stnd_SalesTrain.



Standardize Data

Input Data | Code | Log | Output Data

Modify Task | Filter and Sort | Query Builder | Data | Describe | Graph | Analyze | Export | Send To

	SALES	DATE	YEAR	MONTH	DAY	SalesTrain	Infilled	stnd_SalesTrain
1	$220	80Q1	1980	1	1	220	0	220
2	$257	80Q2	1980	4	1	.	1	544.21818182
3	$258	80Q3	1980	7	1	258	0	258
4	$295	80Q4	1980	10	1	295	0	295
5	$247	81Q1	1981	1	1	247	0	247
6	$292	81Q2	1981	4	1	292	0	292
7	$286	81Q3	1981	7	1	286	0	286
8	$323	81Q4	1981	10	1	323	0	323
9	$284	82Q1	1982	1	1	284	0	284
10	$307	82Q2	1982	4	1	307	0	307
11	$318	82Q3	1982	7	1	318	0	318
12	$343	82Q4	1982	10	1	343	0	343
13	$299	83Q1	1983	1	1	299	0	299
14	$351	83Q2	1983	4	1	351	0	351
15	$359	83Q3	1983	7	1	359	0	359
16	$384	83Q4	1983	10	1	384	0	384
17	$342	84Q1	1984	1	1	342	0	342
18	$388	84Q2	1984	4	1	.	1	544.21818182

Table 2: Updated table

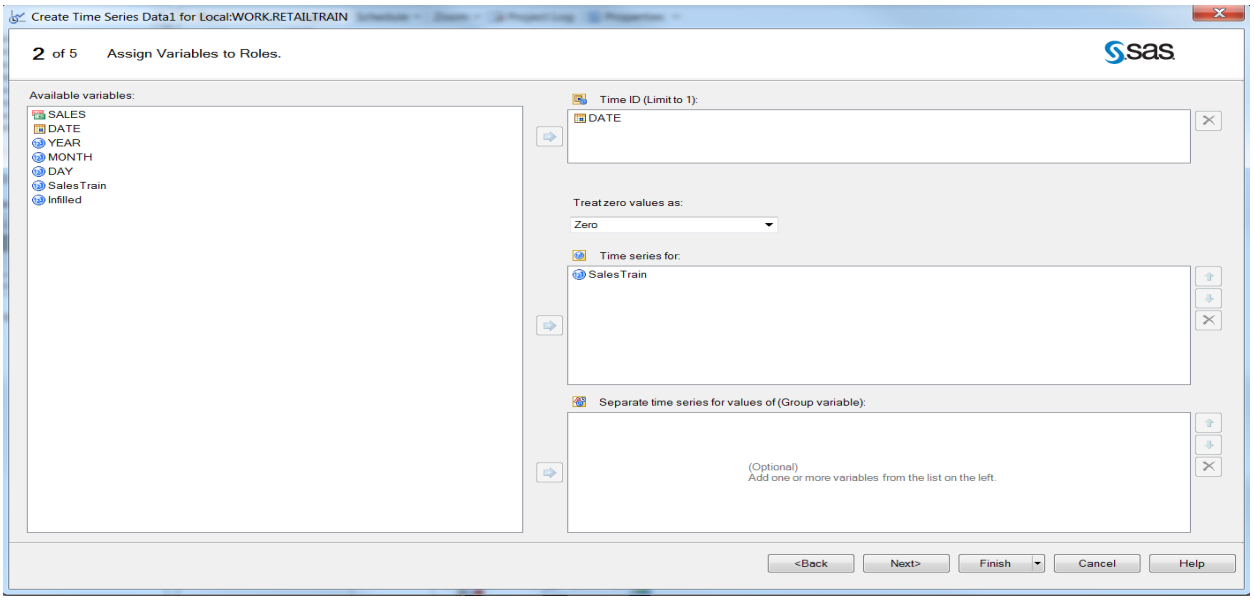
USE THE CREATE TIME SERIES DATA WIZARD IN SAS ENTERPRISE GUIDE

If your data is a time series, that is, data that was collected at successive time intervals, and you have SAS/ETS software installed, you can take advantage of time series features available in SAS Enterprise Guide to address missing values in numerical fields.

Our demo data set RetailTrain is a time series data as it captures quarterly sales numbers. After you open the data set in the SAS Enterprise Guide window, click **Analyze** in the top panel and select **Create Time Series Data**. This invokes the **Create Time Series Data** wizard that will guide you through a series of steps described below:

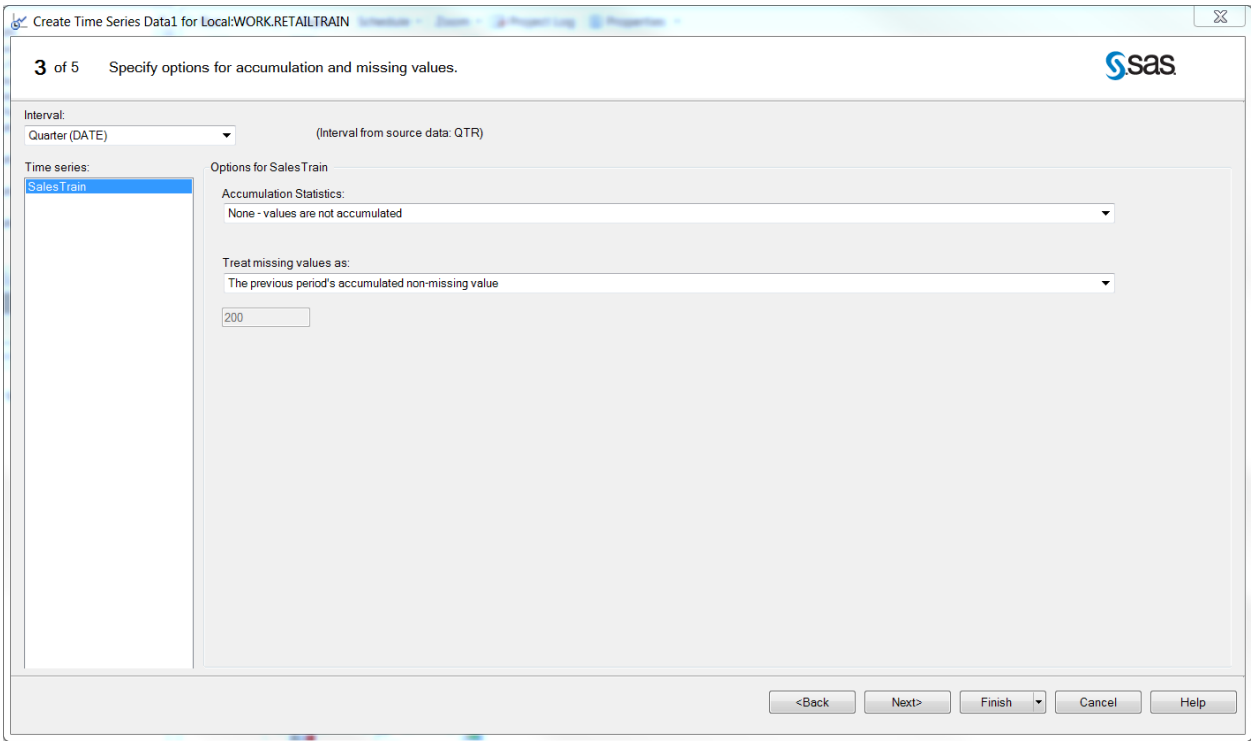
Step 1. Verify your input data.

Step 2. You need to assign variables to appropriate roles. In our demo, the field DATE is a Time ID variable and the field Sales Train is a Time Series variable.



Display 21: Step 2 of Create Time Series Data Wizard

Step 3. Select a method for treating missing values. The Create Time Series Data Wizard offers a number of methods to address missing values, ranging from calculating the average value of accumulated series to calculating the previous period accumulated non-missing value, just to name a few. To select an appropriate method, analysts need to understand their data and its expected behavior or trend over time. For the purpose of this demonstration, we selected **the previous period's accumulated non-missing value** option as shown below.

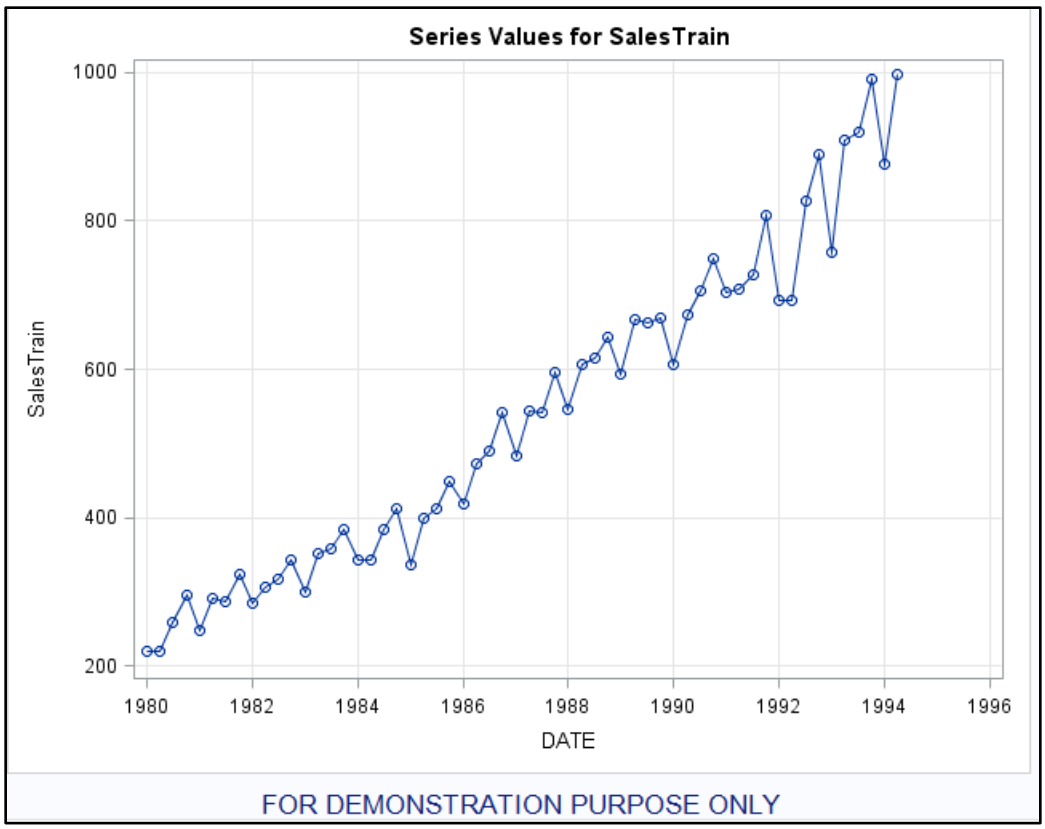


Display 22: Step 3 of Create Time Series Data Wizard

Step 4. In this step SAS Enterprise Guide prompts users to select graph and output options. It is always a good idea to select a graph option as it helps to see how the missing values were interpolated and what the overall trend looks like.

Step 5. SAS Enterprise Guide asks you to supply title and footnote. Once you are done, click **Finish**.

The graph below is generated by SAS Enterprise Guide and enables users to visualize the series trend after missing values were filled in for the periods 80Q2, 84Q2, and 92Q2. Given how easy it is to run the Create Time Series Data wizard, users are encouraged to try different methods and review graphs prior to choosing a particular method.



Output 3: Interpolated time series

USE THE POWER OF PROC EXPAND FOR TIME SERIES DATA

When working with time series data, analysts can benefit from the rich functionalities of the EXPAND procedure, assuming that SAS/ETS software is installed. This particular procedure can be easily integrated into the existing SAS Enterprise Guide program through the use of a code node, and can equip analysts with a toolkit that includes numerous methods for treating missing values.

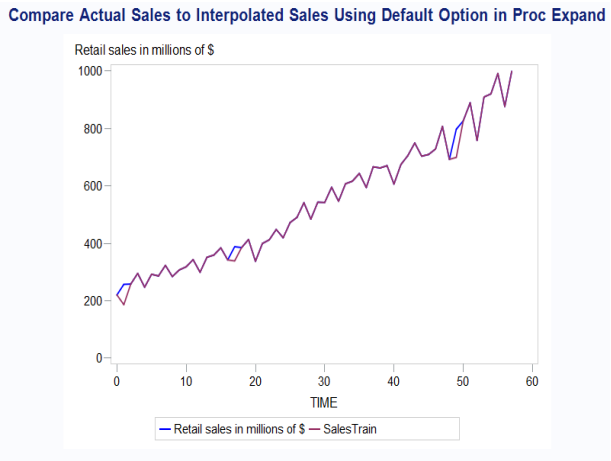
To add a code node to your SAS Enterprise Guide project, select **File** > **New** > **Program**. A new program window will open in SAS Enterprise Guide, and a program node will appear in the Process Flow diagram. The program node can be linked to the data set of interest in the program flow, or a LIBNAME statement for the library containing the data set can be written into the code. For more detailed information on the latter and SAS programming in general, please refer to the Cody reference at the end of this paper or visit support.sas.com.

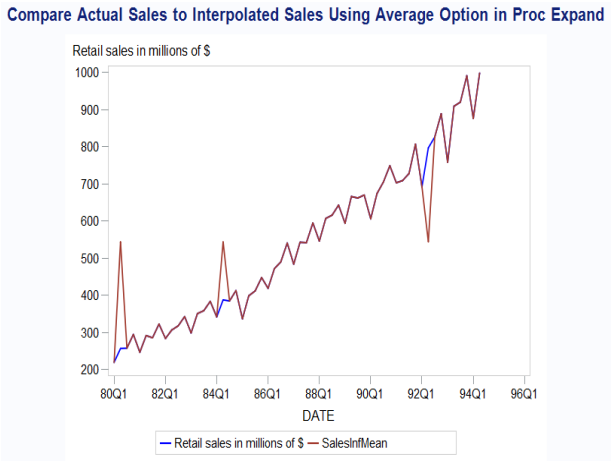
Sample code with **PROC EXPAND** is presented below.

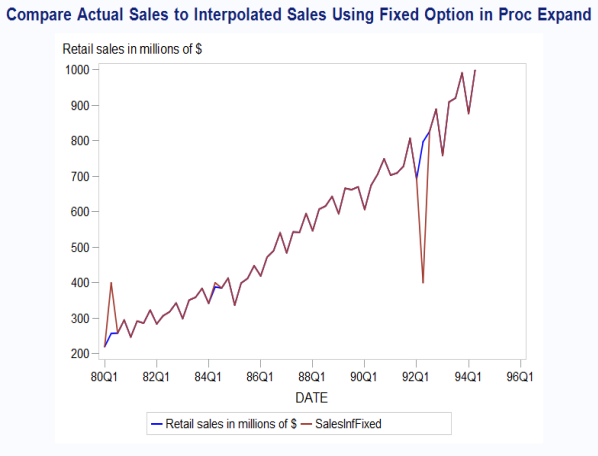
```
proc expand data=retailTrain(keep=Date Sales SalesTrain) out=retailTrainDflt;
 run;

proc expand data=retailTrain out=retailTrainEWMA;
 id Date;
 convert SalesTrain=SalesInfEWMA/transformin=(missonly ewma 0.3);
run;

proc expand data=retailTrain out=retailTrainFixed;
 id Date;
 convert SalesTrain=SalesInfFixed/transformin=(setmiss 400);
run;

proc expand data=retailTrain out=retailTrainMean;
 id Date;
 convert SalesTrain=SalesInfMean/transformin=(missonly mean);
run;
```

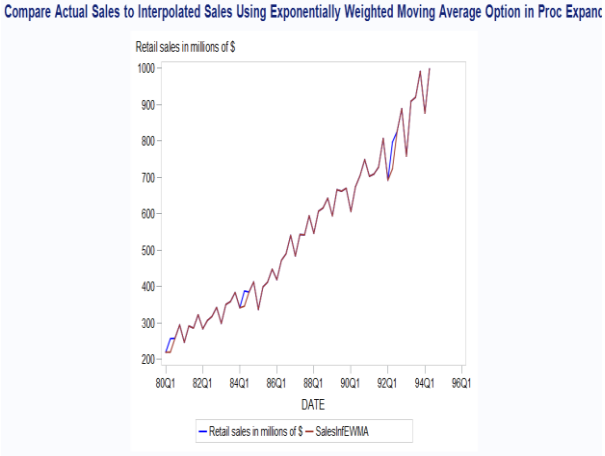
The script demonstrates four different methods of interpolating missing values. The first example is using a default method, a cubic spine function. The second one is using exponentially weighted moving average. The third replaces all missing values with a fixed value of 400. The fourth is setting missing values to the average of the series.

The graphs below compare the original series, Retail Sales, to the interpolated series. The exponentially weighted moving average does a better job interpolating missing values for this series, followed by the cubic spine method.









Output 4: Interpolated results using PROC EXPAND

CONCLUSION

As this paper demonstrates, SAS Enterprise Guide offers numerous functionalities to help analysts quickly and easily assess prevalence of missing data in data sets, and identify the most appropriate method of handling them. Most tasks described in the paper require no prior programming background, and can be implemented through a very intuitive point-and-click interface. For good measure, an example of how to add a code node to the project has been included. Analysts working with data in SAS Enterprise Guide have all the right tools in their arsenal to perform a thorough and sound analysis of data with missing values.

REFERENCES

Cody, R. 2007. *Learning SAS® by Example: A Programmer's Guide*. 54-55. Cary, NC. SAS Institute, Inc.

Chitale, A., and L. Clover. 2012. "Up close and personal with SAS Enterprise Guide 5.1". *Proceedings of the 2012 SAS Global Forum*. Cary, NC. SAS Institute, Inc. Available at http://support.sas.com/resources/papers/proceedings12/302-2012.pdf

SAS Institute, Inc. 2013. *SAS/ETS® 13.1 User's Guide*. Cary, NC: SAS Institute, Inc. Available at http://support.sas.com/documentation/cdl/en/etsug/66840/HTML/default/viewer.htm#etsug_expand_overview.htm

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors:

Elena Shtern
1530 Wilson Blvd, Suite 800
Arlington, VA 22209
SAS Institute Inc.
Elena.Shtern@sas.com
http://www.sas.com

Matt Hall
1530 Wilson Blvd, Suite 800
Arlington, VA 22209
SAS Institute Inc.
Matt.Hall@sas.com
http://www.sas.com