CHAPTER

2

Loading Data

Introduction

This chapter provides information that explains how SAS/SPECTRAVIEW loads data and how you can affect the results, and the chapter provides instructions on how to load data into SAS/SPECTRAVIEW .

Understanding the Volume Grid

When data is loaded into SAS/SPECTRAVIEW , the software creates a three-dimensional volume grid by plotting the values for the axis variables along the X, Y, and Z axes. Each intersection of an x,y,z coordinate is a data point in three-dimensional space. The shape and size of the volume grid is determined by the number of unique X, Y, and Z values.

The resulting total number of data points can be calculated by multiplying the number of unique X values * unique Y values * unique Z values. For example, if you have 10 X-axis values, 5 Y-axis values, and 2 Z-axis values, the result is 100 data points (10x5x2). If you have 10 values on each axis, the result is 1,000 data points (10x10x10).

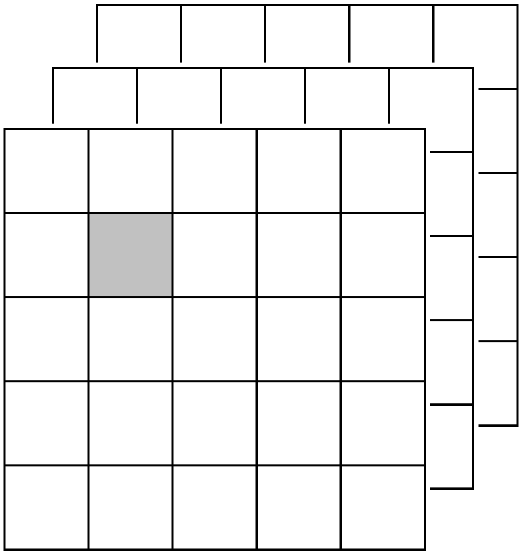
Loading Data Representing a Complete Grid

Data that represents a complete grid contains at least one set of x,y,z coordinates for each possible X, Y, Z variable combination. That is, when loading data, each time SAS/SPECTRAVIEW finds a unique axis value, the software creates a new grid intersection. For the grid to be complete, the data set must contain corresponding X, Y, and Z values for each possible intersection. The resulting number of data points would be the same as the number of observations in the data set, with the data points uniformly distributed in the volume grid, unless there are duplicate observations for a set of x,y,z coordinates. *Note that SAS/SPECTRAVIEW works best with data that represents a complete grid.*

Examples of data that would result in a complete grid is an air quality survey that includes a full grid of sample data from an entire area, scientific numerical models, medical images, or complete financial models like a mortgage table.

To have an idea of how much data is required for a complete grid, think of it like a three-dimensional spreadsheet where multiple sheets extend along the Z axis and where each cell on each sheet represents the values for one observation. Suppose the variables ROW represents X, COLUMN represents Y, and SHEET represents Z. The values ROW=2, COLUMN=2, and SHEET=1, which is one observation, would be located in the spreadsheet as shown in Figure 2.1 on page 20.

Figure 2.1 Three-Dimensional Spreadsheet



For a complete column 2, you would need these observations:

ROW	COLUMN	SHEET
1	2	1
2	2	1

3	2	1
4	2	1
5	2	1

For a complete sheet 1, you would need observations for all five columns:

ROW	COLUMN	SHEET
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
1	2	1
2	2	1
3	2	1
4	2	1
5	2	1
1	3	1
2	3	1
3	3	1
4	3	1
5	3	1
1	4	1
2	4	1
3	4	1
4	4	1
5	4	1
1	5	1
2	5	1
3	5	1
4	5	1
5	5	1

Finally, to complete the entire grid, you would need all those observations for sheet 2 and for sheet 3.

Loading Data Representing an Incomplete Grid

Data that represents less than a complete grid is data that does not have every possible combination but has at least one of the three values for X, Y, or Z. For example, data that represents an incomplete grid could be an air quality survey that consists of samples from random locations within a certain cubic area.

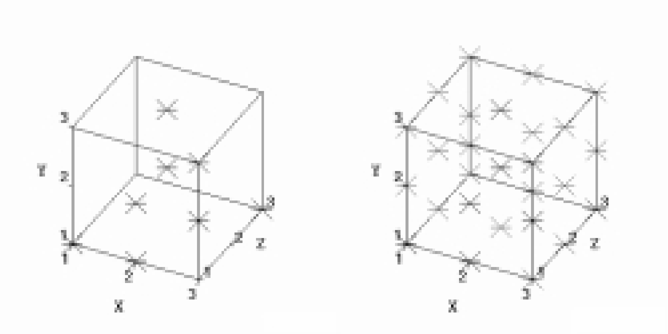
For an incomplete grid, when the software plots the actual axis values, any grid intersections without a data point are completed with software-generated *filler points* for the missing X, Y, or Z values to complete the grid.

For example, consider the following eight observations, which contain three unique values for each axis:

OBS	X	Y	Z	Response
1	1	1	1	111
2	2	1	1	211
3	3	1	3	313
4	3	2	1	321
5	3	3	1	331
6	2	2	1	221
7	2	2	2	222
8	2	3	2	232

The software would generate and plot 27 data points (3x3x3) — 8 actual data points representing the observations and 19 filler points as shown in Figure 2.2 on page 22. The first volume grid shows the actual data points; the second volume grid shows the actual data points and the filler points.

Figure 2.2 3x3x3 Volume Grid

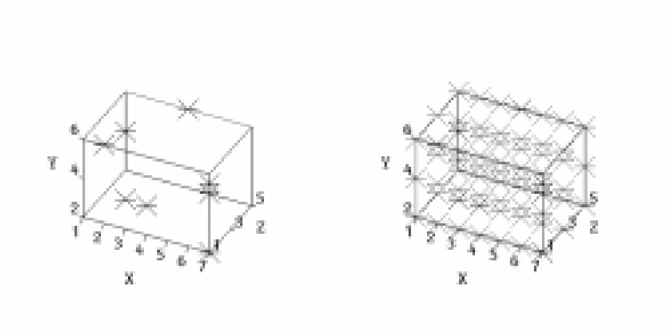


The larger the number of unique values for an axis, the larger the resulting number of data points. For example, consider the following eight observations, which contain 7 unique values for the X axis, and three unique values for the Y and Z axes.

OBS	X	Y	Z	Response
1	1	4	5	145
2	3	2	3	323
3	2	2	3	223
4	4	6	5	465
5	6	4	3	643
6	7	2	1	721
7	5	2	5	525
8	1	6	1	161

The software would generate and plot 63 data points (7x3x3) – .8 actual data points representing the observations and 55 filler points as shown in Figure 2.3 on page 22. The first volume grid shows the actual data points; the second volume grid shows the actual data points and the filler points.

Figure 2.3 7x3x3 Volume Grid



Loading Sparse Data

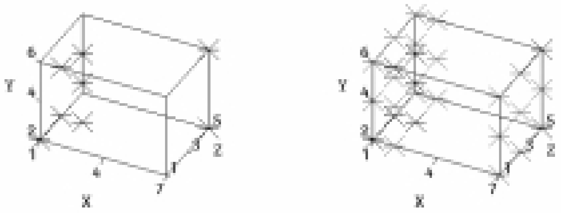
Data that does not contain at least one value for an x,y,z coordinate within the volume grid is referred to as *sparse* data. Generally, sparse data occurs when the unique values for an axis are widely distributed along the axis, for example, an air quality survey where an entire section of a test area was not sampled. And often, sparse data is not related spatially, for example, a data set where the X, Y, and Z values are height, weight, and age. *Note that sparse data can also result from subsetting.*

Unlike for locations having at least one value for x,y,z coordinate, the software does not replace non-existent x,y,z coordinates with filler points. Instead, the volume grid displays a visual gap indicating an area within the volume grid where no data is available. The actual data points appear to be non-uniformly distributed because of the gap in the data. Consider the following data, which contains three unique values for the axis variables:

OBS	X	Y	Z	Response
1	1	4	5	145
2	1	2	3	123
3	2	2	3	223
4	7	6	5	765
5	2	4	3	243
6	1	2	1	121
7	7	2	5	725
8	2	6	1	261

When the actual data values are plotted and the volume grid is completed, the actual data points are not uniformly distributed, resulting in a volume grid that appears to have gaps. The software would generate and plot 27 data points (3x3x3) – 8 actual data points representing the observations and 19 filler points as shown in Figure 2.4 on page 23. The first volume grid shows the actual data points; the second volume grid shows the actual data points, the filler points, and visual gaps:

Figure 2.4 Sparse Data Volume Grid



Note that when loading character data, gaps will not occur. The software assigns sequential numerical values to the character values, resulting in uniformly distributed data points.

Understanding Missing Values

A missing value is a value in the SAS System indicating that no data is stored for the variable in the current observation. In SAS/SPECTRAVIEW , any grid intersections with missing X, Y, or Z values or any x,y,z coordinate without an associated response value are completed with software-generated *filler points*. Filler points are handled as missing values.

Missing values, by default, have no color. If you want missing values to display in an image, you must use the color palette to assign a color as explained in “Assigning Color to Missing Values” on page 51.

If your data represents an incomplete grid or sparse data, the software may create many filler points. However, if your data represents a complete grid, displaying missing values lets you see holes, which may indicate a possible failure of the measuring equipment.

Assigning Librefs to Data

The SAS System requires that data stored in a SAS library or a specific SAS data set in a SAS library must have a *libref* (library reference) assigned to it. A libref is an arbitrary name that you make up to symbolically represent a SAS library. The SAS System uses a libref to associate a SAS library with its physical location.

The SAS System automatically provides the librefs SASUSER and WORK. SASUSER is for permanent SAS data sets; WORK is for temporary SAS data sets that are discarded at the end of a SAS session. You must assign librefs for any data library other than the libraries associated with SASUSER and WORK.

To assign a libref, issue the LIBNAME statement. You must assign a libref outside SAS/SPECTRAVIEW, for example, from the SAS PROGRAM EDITOR window. The most common form of the LIBNAME statement contains only the libref and the physical location for the SAS library, for example:

```
libname mylib 'mypath';
```

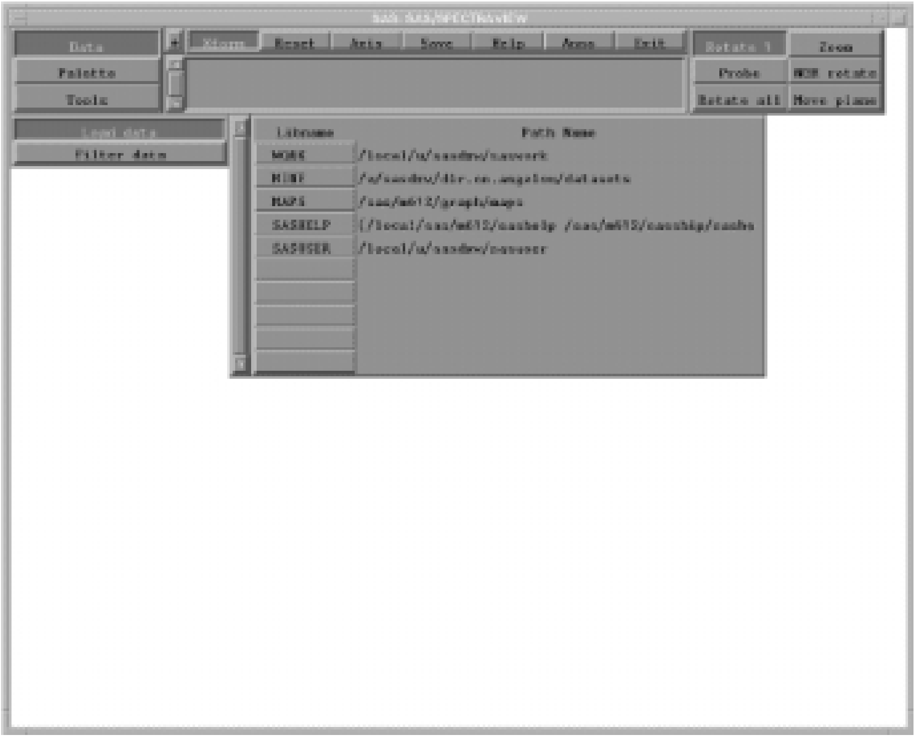
Note: You can include LIBNAME statements in an autoexec file, which is an external file containing SAS statements that are executed automatically when the SAS System is invoked. Then each time you bring up the SAS System, your librefs are automatically assigned. △

Loading the Data

The first step in the visualization process is selecting and reading your data into SAS/SPECTRAVIEW. The interface guides you through the process.

When you first invoke SAS/SPECTRAVIEW, Data is selected by default, ready for you to load data. Note that you can load data at any time during a SAS/SPECTRAVIEW session by reselecting Data.

Display 2.1 Loading Data



Libname	Path Name
WORK	/local/u/sasdmv/saswork
MINE	/u/sasdmv/dir.on.angelou/datasets
MAPS	/sas/m612/graph/maps
SASHELP	(/local/sas/m612/sashelp /sas/m612/sasshlp/sashe
SASUSER	/local/u/sasdmv/sasuser

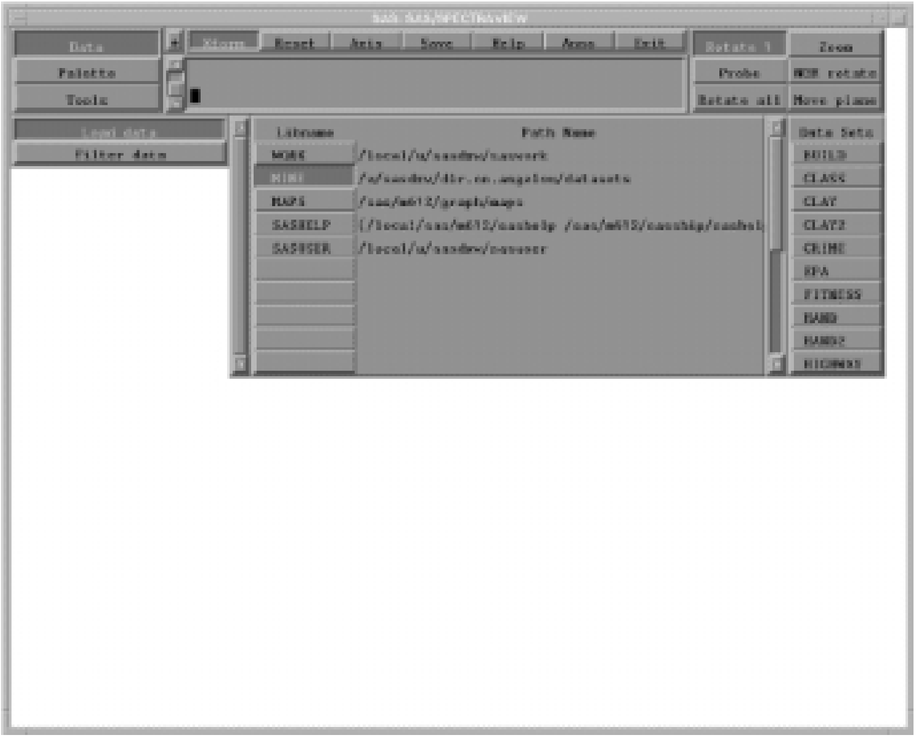
Selecting a Libref

To display the session's assigned librefs:

1 Select Load data . The software displays the assigned librefs under the label **`Libname.`** *To assign an additional libref for a session, you can do so from the SAS PROGRAM EDITOR window (if you invoked SAS/SPECTRAVIEW with a command), then refresh the session's librefs for SAS/SPECTRAVIEW by reselecting the Data and Load data buttons.*

2 Select the libref containing the data set that you want to load. Use the scroll bar if there are more than 10. Once you select the libref, the software displays the data sets associated with the libref.

Display 2.2 Selecting a Libref



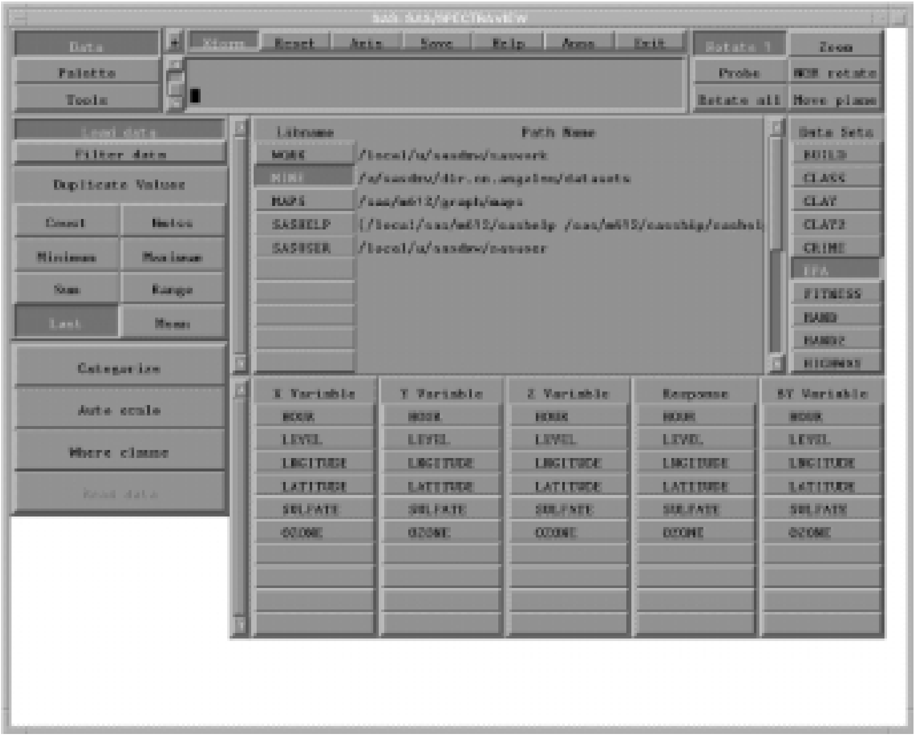
Libname	Path Name
WORK	/local/u/sasdmv/saswork
MINI	/u/sasdmv/dir.on.[illegible]/datasets
MAPS	/sas/m612/graph/maps
SASHELP	(/local/sas/m612/sashelp /sas/m612/saschip/sashel
SASUSER	/local/u/sasdmv/sasuser

Selecting a Data Set

SAS/SPECTRAVIEW works as well with small data sets (such as 20 observations) as it does with large data sets (such as a quarter million observations). The SAS data set that you select must have at least four variables to be specified for the three axis variables and the response variable, the response variable must be numeric, and each variable specified for SAS/SPECTRAVIEW must contain at least two unique values. If you want to use a BY variable, the data set must have a fifth variable as well. *To load a data set that has only three variables, see "Loading a Data Set with Only Three Variables" on page 34.*

Select the input data set from the list of names. Use the scroll bar if there are more than 10. Once you select the input data set, the software lists the data set's variables in columns from which you can select SAS/SPECTRAVIEW variables.

Display 2.3 Selecting a Data Set



Libname	Path Name
WORK	/local/u/sasdmv/saswork
MINE	/u/sasdmv/dir.on.angeles/datasets
MAPS	/sas/m612/graph/maps
SASHELP	(/local/sas/m612/sashelp /sas/m612/sashlp/sashel
SASUSER	/local/u/sasdmv/sasuser

X Variable	Y Variable	Z Variable	Response	BY Variable
HOUR	HOUR	HOUR	HOUR	HOUR
LEVEL	LEVEL	LEVEL	LEVEL	LEVEL
LNGITUDE	LNGITUDE	LNGITUDE	LNGITUDE	LNGITUDE
LATITUDE	LATITUDE	LATITUDE	LATITUDE	LATITUDE
SULFATE	SULFATE	SULFATE	SULFATE	SULFATE
OZONE	OZONE	OZONE	OZONE	OZONE

Specifying SAS/SPECTRAVIEW Variables

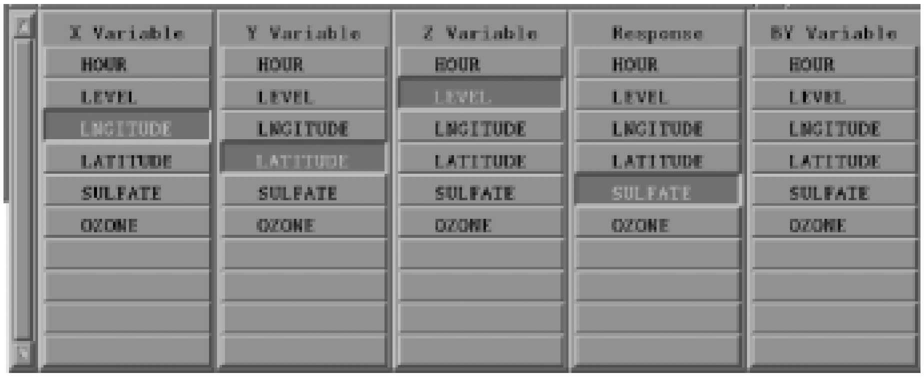
You must specify a different data set variable for each SAS/SPECTRAVIEW variable. That is, you must select a different variable from each of the **`X Variable`**, **`Y Variable`**, **`Z Variable`**, and **`Response variable`** columns. The axis variables can be either numeric or character, but the response variable must be numeric.

To help you select appropriate variables, you can place your cursor on a variable name, and the software will display a short description of it in the text window. For example, for the EPA data set, which contains the variables HOUR, LEVEL, LNGITUDE, LATITUDE, SULFATE, and OZONE, their descriptions provide the following information:

- □ All the variables are numeric. Specifically, the description for SULFATE is **`Type: Num, Label: Sulfate (ppm)`**.
- □ SULFATE and OZONE specify that their values are in ppm (parts per million). SULFATE and OZONE are good candidates for the Response variable, since you usually want a variable that is observed or generated in various quantities. A Response variable is one that contains the values that are of most interest.
- □ Variables LEVEL, LNGITUDE, and LATITUDE are described as RADM Model layer, RADM Cell X coordinate, and RADM Cell Y coordinate. Their values are most likely not sampled or generated but represent where SULFATE and OZONE values are located or from what types of data the response values were generated. Therefore, LEVEL, LNGITUDE, and LATITUDE are good candidates as axis variables, since they can be used to generate grid locations to display the response values.
- □ HOUR contains hour values. This type of variable is useful to generate groups of observations by assigning it as a BY variable as explained in “Grouping Observations with a BY Variable” on page 28.

Note that any variable that is appropriate as a Response variable is not a valid choice as an axis variable, and any variable that is appropriate as an axis variable is not a valid choice for a Response variable. Attempting to read a data set with inappropriate variables selected could result in the data set failing to load. You want to specify variables that are the best ones as the axis variables to build as complete a volume grid with actual data points as possible. And you want to avoid specifying axis variables that are sparsely valued or have continuous data.

Display 2.4 Specifying SAS/SPECTRAVIEW Variables



Once you select the four required variables, the software highlights Read data, but you still have the option of specifying BY variable processing, duplicate values handling, data categorizing, automatic axis scaling, and data subsetting with a WHERE clause, which are discussed in the following sections.

Grouping Observations with a BY Variable

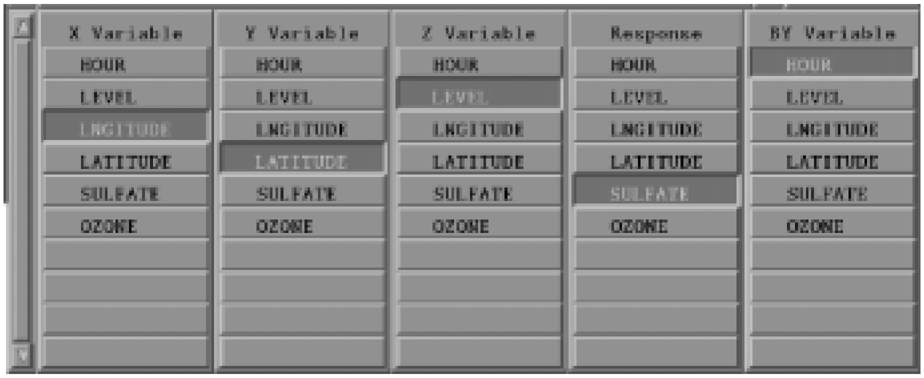
In addition to the four required variables, you have the option of specifying a fifth variable as a *BY variable.* The values of a BY variable define groups of observations, such as hour, month, or year. Specifying a BY variable allows you to animate an image so that you can see how response values change according to some grouping, like over time.

A BY variable can be either character or numeric. BY data usually includes multiple response values for a single data point.

For example, in the EPA data set, the variable HOUR contains hour values, which would be useful as a BY variable. If you imagine that the first four variables would generate a cube of data values, then specifying a BY variable would generate a sequence of cubes of data values that can be cycled through to determine how response values change over time (in this case).

If you select LNGITUDE, LATITUDE, and LEVEL as the axis variables, SULFATE as the Response variable, then HOUR as the BY variable, you will create a sequence of volumes of data to be displayed and analyzed.

Display 2.5 Specifying a BY Variable



X Variable	Y Variable	Z Variable	Response	BY Variable
HOUR	HOUR	HOUR	HOUR	[✓ HOUR]
LEVEL	LEVEL	[✓ LEVEL]	LEVEL	LEVEL
[✓ LNGITUDE]	LNGITUDE	LNGITUDE	LNGITUDE	LNGITUDE
LATITUDE	[✓ LATITUDE]	LATITUDE	LATITUDE	LATITUDE
SULFATE	SULFATE	SULFATE	[✓ SULFATE]	SULFATE
OZONE	OZONE	OZONE	OZONE	OZONE

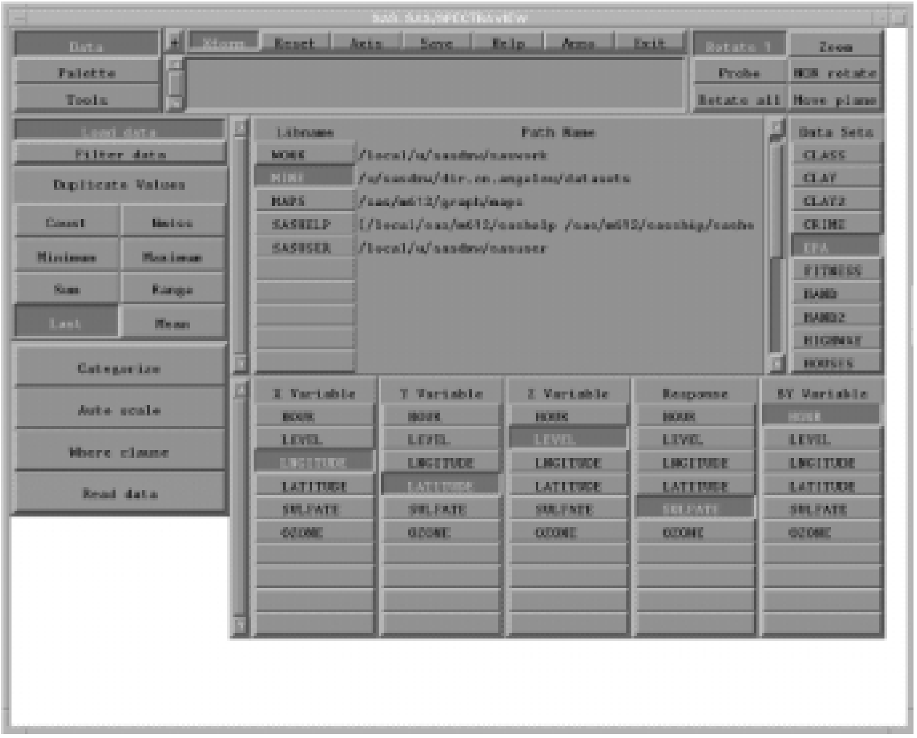
Note: If you do not specify a BY variable but your data contains BY data (like a time variable), you may receive a message in the text window after loading the data. The message warns that there is more than one response value for an x,y,z coordinate. When this occurs, the software handles the response values according to the setting on the **`Duplicate Values`** panel. △

Handling Duplicate Values

Duplicate values occur when the data has more than one observation for the same x,y,z coordinate, which could result in more than one response value for a data point. Note that if you also categorize the data or if you have specified a BY variable, the instances of duplicate values may increase.

You determine how the software handles duplicate values by selecting one of the choices under the label **`Duplicate Values.`** The default is Last, which means that the last response value encountered for a data point is used as that location's response value.

Display 2.6 Handling Duplicate Values



Libname	Path Name
WORK	/local/u/sasdmw/saswork
MINE	/u/sasdmw/dir.on.angeles/datasets
MAPS	/sas/m612/graph/maps
SASHELP	(/local/sas/m612/sashelp /sas/m612/sasship/sashe
SASUSER	/local/u/sasdmw/sasuser

To specify how the software handles duplicate values, select one of the following options:

Count

For each unique x,y,z location, the software counts the number of observations and uses that count as the response value. For example, if there are three observations that specify the x,y,z location 1,1,1, the response value is 3, regardless of the actual response values in the data.

When you load data, each response value for the resulting data points represents a count of the observations for that location. If there are no duplicate observations for a particular x,y,z location, the response value is 1, indicating that only one observation was found for that location. Similarly, if the data includes no observations for a particular x,y,z location, the response value would be 0, meaning that the data point is missing. Count allows you to find the number of response values that were used to calculate other values, for example, Mean or Sum. If you load data with Mean, you may want to know how many values were used to calculate the mean value shown at a particular x,y,z location. You can load again using Count, then probe the data to reveal the number used for the mean.

Nmiss

For each unique x,y,z location, the software counts the number of observations with missing response values. For example, if an x,y,z location has two observations and both have a valid response value, the result is a response value of 0, meaning no observations with a missing response value were found for that location.

With Nmiss specified, every data point has a response value indicating how many missing response values were encountered for that location. If a valid data point has five observations and only three had response values, then that data point's response value is 2, meaning two observations were found missing a response value for that location. Nmiss only counts valid data points having no response value. It does not count filler points generated by the software. If the data does not contain an observation for an x,y,z location, the software inserts a data point that has a missing response value. This means that if you load a data set, display it as a point cloud, and discover there are several missing values in the volume grid, you can reload the data with Nmiss selected and determine which missing values are caused by missing response values as opposed to missing axis values.

Minimum

If there are two or more response values for the same x,y,z location, the software uses the minimum value as the response value.

Maximum

If there are two or more response values for the same x,y,z location, the software uses the maximum value as the response value.

Sum

If there are two or more response values for the same x,y,z location, the software uses the sum as the response value.

Range

If the data contains at least two response values for each x,y,z location, the software uses the range as the response value. The range is calculated by subtracting the minimum response value from the maximum response value. If there is only one value for a location, the response value is set to missing.

Last

If the data contains two or more response values for the same x,y,z location, the software uses the last response value as the response value. This is the default.

Mean

If the data contains two or more response values for the same x,y,z location, the software uses the mean as the response value.

Categorizing Data

Categorizing data is an option that groups numeric data to create distinct ranges (called categories) for each axis. *You cannot categorize character variables.* The result is a reduced number of data points in the volume grid. By categorizing all three axes, you can set exactly how many data points the software will create. Categorizing data is useful

- for data that has a large number of unique values for one or more axes, which would result in a large number of data points making the data difficult to analyze or possibly unable to be loaded.
- when it is simply not important that each axis value be distinct. In many cases, measurements are more precise than necessary for successful analysis.

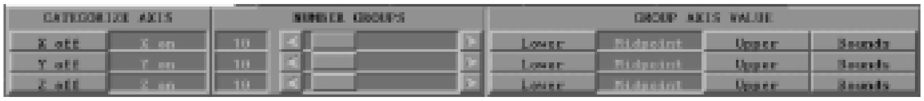
Continuous data (containing few gaps that vary slightly over a large range like weight and height) are a good candidate for categorizing. For example, to analyze a group of people's heart rate based on their age, activity level, and weight, the weight values, which would be in pounds like 139.5, 143.6, would be considered continuous. That is, it is not likely that any two people (let alone several) would have the same weight but a different age and activity level. Categorizing the weight values by creating weight categories for ranges of weight with one value to represent each category would make the data clearer and easier to use.

Discrete data (containing natural gaps like patient IDs and years) would probably not be as useful to categorize. But discrete data such as hour could be categorized into groups if the degree of precision can be reduced without losing data integrity.

To categorize data:

1 Select Categorize. The software displays a group of sliders and buttons at the bottom of the interface.

Display 2.7 Categorizing Data



2 Under the label **`CATEGORIZE AXIS`**, specify which axis you want to categorize. By default, all three are turned on for categorizing. Use the on/off buttons to turn categorizing on or off for a particular axis. For example, selecting X on turns on categorizing for the X axis, and selecting Y off turns off categorizing for the Y axis.

3 Under **`NUMBER GROUPS`**, use the sliders to specify the number of categories you want for each axis. You can specify between two and 100 categories for each, with 10 being the default.

4 Under **`GROUP AXIS VALUE`**, for each categorized axis, specify the axis tick mark value:

Lower Uses the lower bound value in each range.

Midpoint Uses the midpoint value in each range. This is the default setting.

Upper Uses the upper bound value in each range.

Bounds Uses both the upper and lower bound values in each range. The values display as a range, for example, **`125.1–225.1`** for each major tick mark.

Effect on Duplicate Values Handling

Categorizing data makes it more likely that the software encounters more than one response value for a given x,y,z coordinate. (Uncategorized data usually contain only one response value for each x,y,z coordinate.) When one or more of the axes are categorized, some of the data points become duplicates within a group, which could result in more than one response value for a single data point.

For example, suppose values for the X variable are integers from 1 to 100. If you categorize the X values into groups of 10 values, 1-10 would be a single category. The data points 1,1,1 and 2,1,1 and 3,1,1 and so forth are viewed by the software as the same data point in the volume grid, because they would all have the same X, Y, and Z values.

The response values for the 10 data points would appear to be 10 different response values for the same data point. The response values for the duplicate locations are handled according to the method specified for duplicate values handling, with the default being to use the last response value found as the category's response value.

Automatically Scaling Axes

By selecting Auto scale, you can automatically scale the volume's three axes to the same length. The default is that the length of each axis is determined by the range of axis values. For example, an axis with values from 1 to 100 is ten times as long as an axis with values from 1 to 10.

Note: Once a data set is loaded, Auto scale is deselected. To load a subsequent data set with automatic scaling, you must select Auto scale again. △

Subsetting Data with a WHERE Clause

Optionally, you can specify a subset of data to be loaded into SAS/SPECTRAVIEW by specifying condition(s) that observations must meet. You can subset response values by specifying criteria for the response variable, and you can subset data points by specifying criteria for the axis variables.

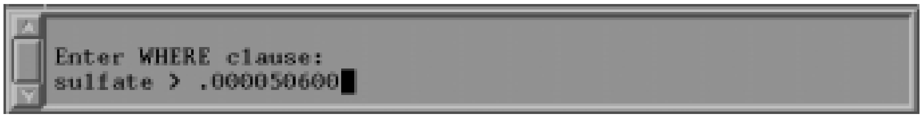
Subsetting can change the size and shape of the volume grid. For example, subsetting data can create holes that are replaced with filler points, or subsetting can remove holes in data.

Prior to selecting Read data, you can specify subsetting conditions using a SAS WHERE clause:

1 Select Where clause.

2 In the text window, type a SAS WHERE clause, without the keyword WHERE and no ending semicolon. A condition consists of a variable name, an operator (such as EQ, NE, LT), and a value, such as **`sulfate > .00005060`**.

Display 2.8 Subsetting Data



```
Enter WHERE clause:
sulfate > .000050600
```

3 Press Enter.

For details on specifying conditions, see the appropriate WHERE clause documentation. *Note that before you invoke SAS/SPECTRAVIEW, you can create a smaller SAS data set containing only the values that you want to use. For example, you could choose certain ranges of axis values or specific response values.*

Reading the Data Set

To have the software read the data, select Read data.

The software loads the input data, applying any optional specifications. For example, if a WHERE clause is specified, the software loads only those observations meeting the criteria, and if categorizing is specified, the software changes the number of data points accordingly. Once the data set is loaded, the variable list disappears, and the software is ready for you to

- □ apply optional data filtering, which is explained in Chapter 3, “Adjusting Data with Filters,” on page 39.
- □ customize colors, which is explained in Chapter 4, “Setting Response Value Colors for Images,” on page 47.
- □ request the visualization techniques that produce the images, which is explained in Chapter 5, “Exploring Data with Visualization Techniques,” on page 55.

If you have loading problems, see “Resolving Data Loading Problems” on page 33.

Resolving Data Loading Problems

The following topics provide suggestions on how to resolve possible data loading problems. Note that if a data set fails to load, the software displays an error message in the text window.

Loading a Data Set with Only Three Variables

SAS/SPECTRAVIEW requires four variables in order to load a SAS data set. However, with the following procedure, it is possible to load a data set that has only three variables.

1 Create a temporary SAS data set with the following DATA step code:

```
data temp;
   set yourdatasetname;
   dummy=1;
   output;
   dummp =2;
   output;
run;
```

2 Load the temporary data set TEMP into SAS/SPECTRAVIEW.

3 Select the X and Y axis variables that you are interested in, then select DUMMY as the Z variable.

4 Select the Response variable that you want.

5 Select Read data.

6 Use the data for your analysis.

Note that the Z plane will have two identical planes (z=1 and z=2). You can ignore the second one.

Changing Axis Variables

Sometimes data will load with certain axes and response variables specified but will not with different ones due to memory constraints. You want to specify variables that are the *best* ones as the axis variables to build as complete a volume grid with actual data points as possible. That is, you want to avoid specifying axis variables that are sparsely valued or have continuous data.

For example, the sample data set MORTGAGE loads without problems if YEARS, RATE, and AMOUNT are specified as the axis variables. However, if you specify PAYMENT for an axis and either YEARS, RATE, or AMOUNT as the response variable, the data may not load, because there are 16,400 unique values for PAYMENT. *Note that if a data set fails to load, the error message in the text window specifies the number of unique values found for each axis.*

See “Specifying SAS/SPECTRAVIEW Variables” on page 27 for details on specifying variables and determining which variables are best.

Categorizing Data

One of the main reasons that a data set will not load is that the data does not represent a complete grid, which most often occurs with random data or if the axis values are continuous rather than discrete. The data set may fail to load due to memory constraints, even when a larger data set loaded successfully. The problem is the number of resulting data points in the volume grid, not the number of observations.

Memory requirements for a data set depend on the number of unique X, Y, and Z values, which determines the number of data points that are created. If the number of data points becomes large, the data set may fail to load without additional memory. Of course, it takes thousands and thousands of data points to cause data loading problems.

To make the data clearer and easier to use in SAS/SPECTRAVIEW, you can *categorize* the data, which groups numeric data to create distinct ranges (called categories) for each axis. Instructions on how to categorize data are in "Categorizing Data" on page 31.

Changing Duplicate Values Handling

Specifying how the software handles duplicate values can cause data not to load. For example, if you select either Count or Nmiss under the label **Duplicate Values** and the data you want to load comprises a complete grid having no missing x,y,z locations and no duplicate observations for the same x,y,z location, the data would fail to load. That is,

- With Count specified, the response value for every data point would be 1. The data would fail to load because Count requires at least two different response values for an x,y,z location.
- With Nmiss specified, the response value for every data point would be 0. The data would fail to load because Nmiss requires at least two different response values for an x,y,z location.

Instructions for specifying how the software handles duplicate values are in "Handling Duplicate Values" on page 29.

Removing BY Variable Specification

Removing the BY variable specification will cut the amount of storage required by the number of BY groups in the data set.

To calculate storage requirements for a BY variable, multiply the number of unique values for each axis variable by the number of BY groups. For example, if you have five BY groups, you would need five times as much storage, because a grid is created for each value of the BY variable.

More information on BY variable processing is in "Grouping Observations with a BY Variable" on page 28.

Using G4GRID Procedure to Create a Complete Grid

You can run the G4GRID procedure on data to create a data set that represents a complete grid. For example, if your data is random in nature, PROC G4GRID may be a good choice. The procedure produces data that is derived from the original data. The amount of time it takes to produce the new data set is based on the number of observations in the data set and the size of the requested output grid.

PROC G4GRID enables the loading of a data set that could not otherwise be loaded due to memory constraints. By using PROC G4GRID, you can fill in missing values with interpolated values or resize the data set as required. PROC G4GRID is useful when

- the response values were sampled at discrete locations, for example, measurements of air pollution.
- the response data is functionally related to the axis variables. That is, the response is either analytically or physically a function of the axis variables. Air pollution measurements are a function of discrete locations identified by axis values, but a stock's price is not a function of a stock's name. That is, just because Granny's Kitchen stock price is high does not mean Gerry's Garage stock price is high even though they fall next to each other in the grid. Smoothing with PROC G4GRID would lower Granny's stock and raise Gerry's stock because they would be assumed to influence each other.

- □ you want a complete grid of values and can accept some changes from your original values.

Complete documentation for PROC G4GRID is in Appendix 1, “The G4GRID Procedure.”

Calculating Volume Grid Storage Requirements

To understand how to calculate storage requirements, compare the following two DATA step examples.

The first example produces 9,261 observations and would load with no problems. In fact, it is a relatively small data set by SAS/SPECTRAVIEW standards. There are 21 unique values for each axis, which results in a grid that has 9,261 data points (21x21x21). Each data point requires approximately four bytes of storage on most machines. Therefore, it requires 4x9,261=~36KB of storage for the grid.

```
data load;
  drop a b c;
  a=0.3;
  b=0.2;
  c=0.1;
  do x = -1 to 1 by 0.1;
    do y = -1 to 1 by 0.1;
       do z = -1 to 1 by 0.1;
       response = x**2/a**2 + y**2/b**2 + z**2/c**2;
       output;
       end;
    end;
  end;
run;
```

The second example, however, may not load, even though it has only 100 observations. The number of unique X, Y, and Z values is unknown, but by using the RANUNI function, it can be assumed that it will be close to 100 for each variable. The grid, therefore, requires 100x100x100=1,000,000 data points or about 108 times (~4MB) the storage requirement as compared to the first example.

```
data noload;
   drop seed I a b c;
   seed = -1;
   a = 0.3;
   b = 0.2;
   c = 0.1;
      do I = 1 to 100;
         x = 2.0*ranuni(seed) - 1.0;
         y = 2.0*ranuni(seed) - 1.0;
         z = 2.0*ranuni(seed) - 1.0;
         response = x**2/a**2 + y**2/b**2 + z**2/c**2;
         output;
      end;
run;
```

Specifying Larger Memory Size

To specify a larger memory size, invoke the SAS System and specify the system option MEMSIZE, which controls how much memory the SAS System uses, with a larger memory size. For example,

`-memsize 100m`.

Note that SAS/SPECTRAVIEW also requires additional memory for overhead, some of which is proportional to the size of the data set. It is possible that, while there is enough memory to build the grid, some other area may not succeed, which will prevent the SAS data set from loading.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/SPECTRAVIEW® Software User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS/SPECTRAVIEW® Software User's Guide, Version 8

ISBN 1–58025–462–4

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

The Institute is a private company devoted to the support and further development of its software and related services.