CHAPTER 1

Navigation

Working in SAS puts a cornucopia of resources literally at our fingertips; a thorough tour of the nooks and crannies of this dynamic software will promote efficient navigation of the product and help us identify the tools that are best suited to any given task. In this chapter we begin with a guided tour of the five Base SAS windows, exploring their purpose and utility from the programmer's perspective. We then further explore those windows in the context of data migration. Finally, we touch upon SAS Enterprise Guide – comparing it with the SAS Windowing Environment and describing how the user can benefit from working in Enterprise Guide.

SAS Windowing Environment

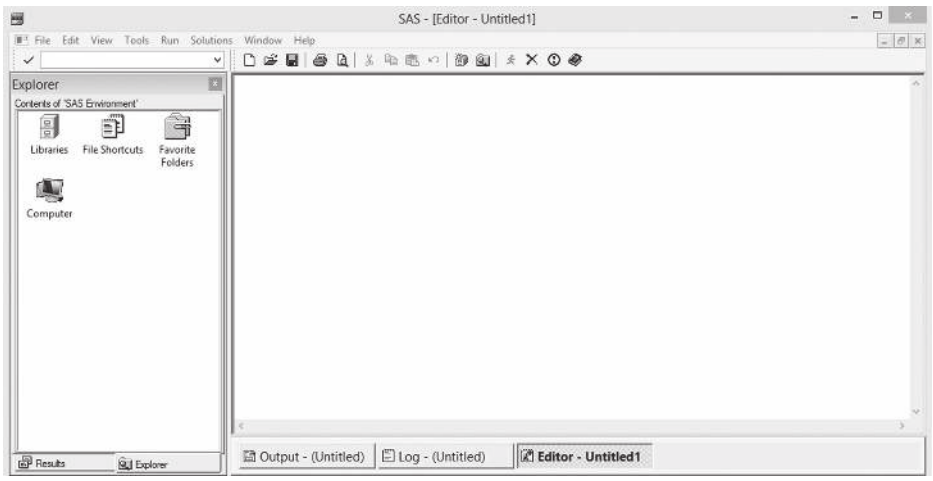
To begin our tour of the SAS Windows, let's open the application. A toolbar can easily be identified near the top of the screen (Figure 1.1) as well as two main work spaces (Figure 1.2a).

Figure 1.1 shows the standard SAS drop-down menus and toolbar. The File and Edit menu include common commands such as new, open, save, print, copy, and paste. File and Edit menu tasks can also be executed from the toolbar using standard icons. The View and Run menus are more unique to SAS. The View menu contains a list of SAS windows and folders; it allows instant

FIGURE 1.1. SAS Toolbar.

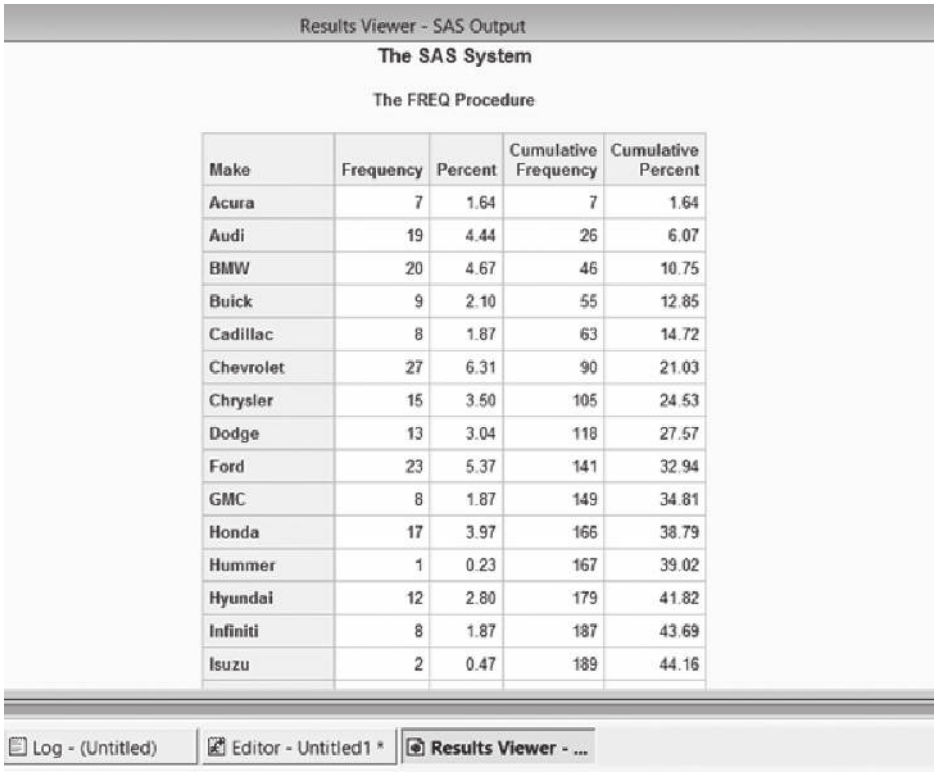
access to these windows (an important piece of information to remember should you accidentally close a window). Run allows the user to submit syntax. The more customary method for submitting programs is the Run icon, also known as the running person, which appears on the SAS toolbar.

Figure 1.2a includes the Editor, Log, and Output window tabs (bottom right-hand side of the screen). The Editor is where syntax (also referred to as code) is written; this is the way the programmer communicates with SAS software. Executing code is the primary way to accomplish all data manipulation and analysis in SAS. The Log window is where SAS software communicates with the user about its work; as syntax is processed, SAS prints an ongoing commentary in the log with respect to the progress of the tasks. Traditionally, the Output window is where any printed product of fully executed syntax (also known as output), such as tables and lists, can be found. In recent versions of SAS software, output is more typically found in the Results Viewer (Figure 1.2b). The Results Viewer, which displays output in HTML format, automatically opens when the first output of a SAS session is created.



(a)

FIGURE 1.2A. SAS Windowing Environment.



Make	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Acura	7	1.64	7	1.64
Audi	19	4.44	26	6.07
BMW	20	4.67	46	10.75
Buick	9	2.10	55	12.85
Cadillac	8	1.87	63	14.72
Chevrolet	27	6.31	90	21.03
Chrysler	15	3.50	105	24.53
Dodge	13	3.04	118	27.57
Ford	23	5.37	141	32.94
GMC	8	1.87	149	34.81
Honda	17	3.97	166	38.79
Hummer	1	0.23	167	39.02
Hyundai	12	2.80	179	41.82
Infiniti	8	1.87	187	43.69
Isuzu	2	0.47	189	44.16

(b)

FIGURE 1.2B. SAS Results Viewer.

On the bottom left-hand side of the SAS workspace there is another section where the Explorer and Results tabs are located (Figure 1.2a). The Explorer window allows the user to navigate through data libraries and the Results window provides a listing of what is available to view in the results viewer or output window. From the Explorer window, double clicking opens datasets in the Viewtable (Figure 1.3); from the Results window, double-clicking brings the programmer to a specific piece of output.

Editor Window

The editor window is where syntax is written. SAS (running on Microsoft Windows) offers a feature called the 'Enhanced Editor', which is designed

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	Engine Size (L)	Cylinders
1	Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	6
2	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2	4
3	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	4
4	Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,299	3.2	6
5	Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,014	3.5	6
6	Acura	3.5 RL w/Navigation 4dr	Sedan	Asia	Front	$46,100	$41,100	3.5	6
7	Acura	NSX coupe 2dr manual S	Sports	Asia	Rear	$89,765	$79,978	3.2	6
8	Audi	A4 1.8T 4dr	Sedan	Europe	Front	$25,940	$23,508	1.8	4
9	Audi	A41.8T convertible 2dr	Sedan	Europe	Front	$35,940	$32,506	1.8	4
10	Audi	A4 3.0 4dr	Sedan	Europe	Front	$31,840	$28,846	3	6
11	Audi	A4 3.0 Quattro 4dr manual	Sedan	Europe	All	$33,430	$30,366	3	6
12	Audi	A4 3.0 Quattro 4dr auto	Sedan	Europe	All	$34,480	$31,388	3	6
13	Audi	A6 3.0 4dr	Sedan	Europe	Front	$36,640	$33,129	3	6
14	Audi	A6 3.0 Quattro 4dr	Sedan	Europe	All	$39,640	$35,992	3	6
15	Audi	A4 3.0 convertible 2dr	Sedan	Europe	Front	$42,490	$38,325	3	6

FIGURE 1.3. SAS Viewtable.

to help the programmer debug their programming on the fly. Within the enhanced editor, text is shown in different colors that tell the programmer if their syntax is correct. Below (Example 1.1) is an example of how the enhanced editor works.

EXAMPLE 1.1. Reviewing the Enhanced Editor.

Syntax	Result
`libname x 'example1';`	✔
`libnm x 'example1';`	✖
`libname x example1';`	✖

The first line shows correct syntax. The word 'libname', when typed into the enhanced editor, will appear in blue because it is a recognized SAS command, and the path 'example 1' will appear in purple because it is fully enclosed by single quotes, telling SAS it is a character string. In the second line we see that 'libnm' is spelled incorrectly; the misspelled syntax will appear in red because it is not a recognized SAS command. In the third line the missing quote before *example 1* renders the path unidentifiable as a character string.

Log Window

Although the enhanced editor is helpful, it is not foolproof. Often, simple spelling errors such as those in Example 1.1 are not initially identified by the programmer. Fortunately, the log is another place where we can and should frequently check our work.

SAS uses the log to tell the programmer how things are going. It is a documentation of everything that happens during the SAS session. Periodically checking the log can save a great deal of time later. The three most common types of messages are NOTE, ERROR, and WARNING. Line numbers are included in some messages to indicate the exact point in the program where an error occurred.

"Note" appears in blue. It describes system processing times and information describing the datasets involved in a particular process. This is a useful place to check the efficiency of your program (by looking at the processing time) and to make sure your dataset has the correct number of observations and variables (Output 1.1).

Output 1.1. SAS Log: Example of Note Message.

```
NOTE: There were 15 observations read from the data set SASUSER.HOUSES.
NOTE: The data set WORK.X has 15 observations and 6 variables.
NOTE: DATA statement used (Total process time):
      real time           0.20 seconds
      cpu time            0.01 seconds
```

"Error" appears in red. It identifies a problem that caused SAS to stop running. Output 1.2 indicates a filename problem; usually this type of error indicates a problem with the path that is specified on the filename statement.

Output 1.2. SAS Log: Example of Error Message.

```
ERROR: Error in the FILENAME statement.
```

Output 1.3 shows an example where a dataset 'work.xyz' has been called upon but does not exist at the specified location.

Output 1.3. SAS Log: Example of Error Message.

```
ERROR: File WORK.XYX.DATA does not exist.
```

"Warning" appears in green. This alerts the user to potential hazards or missteps in data handling. The warning statement should be read carefully; common warnings are due to nonmatching variable lengths, incomplete datasets, and the halt of data steps before completion. Each message gives the user a reference to where the message comes from and describes its meaning and possible options for correcting the issue.

Output 1.4 indicates a warning explaining that the dataset work practice may be incomplete, noting that the dataset has zero observations and zero variables. The warning message makes it clear that the data step needs to be debugged.

Output 1.4. SAS Log: Example of Warning.

```
WARNING: The data set WORK.PRACTICE may be incomplete.  When this step was stopped there were
         0 observations and 0 variables.
```

Output 1.5 shows a product expiration issue. The SAS system operates in 'warning mode' when the product license is close to its expiration date.

Output 1.5. SAS Log: Example of Warning.

```
WARNING: The Base SAS Software product with which RESULTS is associated will be expiring soon,
WARNING: and is currently in warning mode to indicate this upcoming expiration. Please run
WARNING: PROC SETINIT to obtain more information on your warning period.
```

Log messages are particularly important because they offer information that may not be obvious. Often a new dataset is created or output is produced but it is inaccurate or incomplete due to a programming error. Log messages can tell the programmer when this is the case.

Explorer Window and Viewtable

The explorer window allows us to navigate through SAS libraries and catalogs as well as the computer's libraries, drives, and devices (explained in greater detail in Chapter 3). Double-clicking on each library reveals included SAS datasets. While the Explorer Window is active, the view menu allows the user to toggle between various versions of list and thumbnail views. The 'Up one level' button is also available when the Explorer Window is active. This button

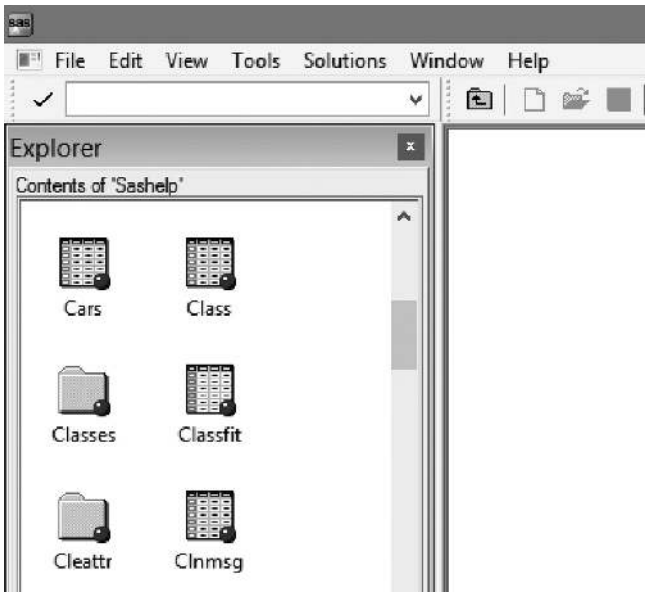


FIGURE 1.4. 'Up One Level' Button.

allows the user to navigate out of one library and into another (Figure 1.4). Double-clicking on a dataset from the Explorer Window opens that dataset in the Viewtable (Figure 1.3). If the explorer window is not already open (or it is accidentally closed), reopen the window by toggling to 'view' on the toolbar and select 'contents only'. Further discussions of investigating data in the Viewtable can be found in Chapter 2.

Accessing Data for This Book

To further demonstrate the Explorer and Viewtable windows, let's take a look at the datasets we will be using for examples throughout this book. First, navigate to the Explorer tab, double-click on the 'Libraries' icon, and then open the folder labeled 'Sashelp.' Scroll through the listing of datasets. Locate the datasets labeled 'Cars', 'Shoes', and 'Class'. We will begin with 'Cars'; double-click on it. The Viewtable should open in the right-hand workspace of the SAS window and the 'Cars' dataset should be visible.

Simple SAS Tips for New Programmers

- *Syntax is not case sensitive. However, case specification is necessary when calling upon the value of a character variable or specifying the path to an external file.*
- *Check the log! The log is an invaluable part of the SAS framework; it saves time and gives the programmer an idea of how SAS is interpreting the code.*
- *Punctuation is important. ALL statements end with a semicolon. A mark as simple as a period can change the meaning of text.*

Data Migration

Data stored in alternative formats (e.g., Microsoft Access, Microsoft Excel, text files) must be converted to SAS format before any manipulation can be done using SAS software. When data manipulation is complete, many datasets must be exported to these alternative formats. There are a few common ways of importing and exporting data using SAS. The import and export wizards provide simple, point-and-click avenues, while composing and executing syntax using the import and export procedures (PROC IMPORT, PROC EXPORT) provides a more hands-on approach. In this chapter we will describe the Import wizard and the IMPORT procedure. Similar techniques apply to exporting data using the Export wizard and EXPORT procedure. In Chapter 8, we will discuss another option when we present JMP software as an alternative for importing data with unusual formats.

The Wizard

Using the wizard to import data files is an easy and efficient way of bringing external data into SAS. Files can be imported from various formats such as Microsoft Access, Microsoft Excel, and text files. Open the 'File' menu on the top left of the SAS toolbar and select 'import data.' The following screen appears (Figure 1.5); the wizard would like to know "What type of data do you wish to import?"

After choosing the type of file to be imported, the wizard prompts for the location of the file using the 'browse' button. In the next step a SAS destination is selected, the default is the 'work' folder, and 'Member' indicates how SAS should name the dataset.

FIGURE 1.5. Import Wizard.

The next screen asks the question of whether or not you would like to save your syntax. If the program you are creating is going to be run again, then yes, it is always a good idea to save the syntax. You can choose any location you like to save the information. One option is to maintain a single SAS program for this purpose and append all imports to this single program. Once the import process is complete and syntax is saved, it can be copied into the current program. If the import will never be used again, then saving the syntax may be a useless task. The decision is completely up to the programmer and their future needs.

The wizard will exit after completion of the import process; if import has been successful, then a new dataset is available. The log is a good place to determine the status of the import (Output 1.6). It will reveal whether the dataset was created and the path to its location.

OUTPUT 1.6. SAS Log: Data Set Creation.

```
NOTE: WORK.SAMPLE_DATA data set was successfully created.
NOTE: The data set WORK.SAMPLE DATA has 4 observations and 3 variables.
```

The dataset can be opened and viewed by navigating through the explorer tab (bottom left) and double-clicking on the appropriate icon as described earlier in this chapter.

Writing Your Own Import

Depending on the type and complexity of the data file you wish to import, it is sometimes necessary to write your own syntax. Three of the most commonly imported file types are Microsoft Excel, Microsoft Access, and text files. The IMPORT procedure can be used for all three, with just a few variations in syntax (some file formats require additional software, i.e., SAS/ACCESS to import data).

The import procedure works in the same way as the import wizard. In Example 1.2 OUT = names the new SAS dataset, DATAFILE = references the location of the file, DBMS = specifies the type of external file, SHEET = names the sheet within the excel file for SAS to read, and GETNAMES = instructs SAS to read the column headings (i.e., the first row of data) as variable names.

Example 1.2. Importing Data from Microsoft Excel Using the IMPORT Procedure.

```
PROC IMPORT OUT= WORK.Samp1e
DATAFILE= "C:\Desktop\exceldata_ch1P2.xlsx"
DBMS=xlsx REPLACE;
SHEET="sheet1";
GETNAMES=YES;
RUN;
```

Similar to Example 1.2, the syntax shown in Example 1.3 imports a text file. The one difference is the inclusion of "DATAROW," which tells SAS to start reading data from the specified row number.

Example 1.3 Importing Data from a Text File Using the IMPORT Procedure.

```
PROC IMPORT OUT= WORK.Sample
DATAFILE= "C:\Desktop\txtsas.txt";
DBMS=TAB REPLACE;
GETNAMES=YES;
DATAROW=2;
RUN;
```