SAS® Programming
Introduction: Basic Concepts

How-To Demonstrations

For SAS 8 and SAS® 9
Table of Contents

<table>
<thead>
<tr>
<th>Lesson</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Getting Started with SAS Programming</td>
<td>1-1</td>
</tr>
<tr>
<td>2</td>
<td>Understanding SAS Programming Basics</td>
<td>2-1</td>
</tr>
<tr>
<td>3</td>
<td>Navigating and Using the SAS Interface</td>
<td>3-1</td>
</tr>
<tr>
<td>4</td>
<td>Working with SAS Libraries, SAS Data Sets, and the Import Wizard</td>
<td>4-1</td>
</tr>
<tr>
<td>5</td>
<td>Using PROC PRINT to Create List Reports</td>
<td>5-1</td>
</tr>
<tr>
<td>6</td>
<td>Using PROC PRINT to Enhance List Reports</td>
<td>6-1</td>
</tr>
<tr>
<td>7</td>
<td>Using the DATA Step to Read Raw Data</td>
<td>7-1</td>
</tr>
<tr>
<td>8</td>
<td>Using Assignment Statements to Create Variables</td>
<td>8-1</td>
</tr>
<tr>
<td>9</td>
<td>Using Conditional Logic</td>
<td>9-1</td>
</tr>
<tr>
<td>10</td>
<td>Using PROC MEANS to Create Summary Reports</td>
<td>10-1</td>
</tr>
<tr>
<td>11</td>
<td>Using PROC FREQ to Create Frequency Reports</td>
<td>11-1</td>
</tr>
</tbody>
</table>
Lesson 1  Getting Started with SAS Programming
1. View the data and figure out the layout of the data file.
2. Read the data into SAS.
3. Create your reports.

1. View a copy of the survey form.
2. View the survey.dat plain text file in a text editor to determine the layout of the data file.
3. Write a SAS program to read the data.
4. Submit the program.
5. View the log to ensure that the program read the data into SAS successfully.
6. Add a statement to the program to print the data.
7. Submit the program.
8. View the report.
9. Alter the program to find the number and percentage of male and female survey respondents.
10. Submit the program.
11. View the report.
12. Alter the program to find the states in which the respondents live.
13. Submit the program.
14. View the report.
15. Alter the program to find the average number of years that respondents worked for their present employers.
16. Submit the program.
View the report.
1. Open SAS.

2. Submit the program to create the following reports:
   - Listing report of the survey data
   - Listing report of the survey data with the values in the column Profession changed to R and D for G and to Executive for A; the state name spelled out; the codes F and M changed to Female and Male; and the report grouped by gender
     - Listing report of the survey data distributed by gender and broken out by state
     - Listing report of the survey data distributed by state
     - Summary statistics of the survey data
     - Advanced statistics of the survey data including the mean number of years with an employer, median, mode, and extreme observations
   3. View the columns of the listing report. Notice the values for the column Profession.
   4. View the second report. Notice the values for Profession, the values for Executive, and the state name. Also notice that the listing report is grouped by gender.
   5. View the third report. Note the number of females and males and the custom note in the title.
   6. In this report, what percentage of Gender is female? 52.3%
   7. View the fourth report. Notice the distribution by state of birth.
   8. In this report, which state represents the highest percentage of the total? North Carolina
     Nine individuals come from North Carolina, making up 42.86% of our data.
   9. View the fifth report. Notice the mean, or average, number of years with the present employer.
   10. Which profession has the lowest mean number of years with the present employer? Executives The average number of years with the present employer is 1.33 for executives.
   11. View the sixth report. Notice the mean number of years with an employer.
   12. Scroll forward in the report. Notice the extreme observations with the five lowest and five highest values in the data.
   13. Look at the list of lowest values. How many individuals in the data have the lowest
number of years of service? Four
14. Submit a program to produce the following charts:
   - Distribution of Gender
   - Total Number of Years at Present Employer by State
   - Frequency of Profession
   - Frequency of Gender
   - Map – Number of Students per State
   - Listing Report in HTML
   - Frequency – Distribution of States in HTML
   - Frequency – Distribution of States in a Bar graph
15. View the first chart. Notice the distribution of gender in the horizontal bar chart.
16. View the second chart. Notice the sum of years with a present employer for each state.
17. View the third chart.
18. View the fourth chart. Note that the chart is broken out by gender and broken out by state.
19. What is the frequency count for males from Pennsylvania? One
20. View the fifth chart. Notice that the height of the state indicates the frequency count.
21. How many respondents are from Florida? One
22. View the sixth chart. Notice that this is the same listing report as previously viewed in HTML format.
23. View the seventh chart.
24. View the eighth chart. Notice that this report appears color coded with the state on one axis and gender on the other axis. Notice that when the mouse hovers over one of the bars, the values for the bar appear.
Lesson 2  Understanding SAS Programming Basics
Lesson 2: Understanding SAS Programming Basics

Creating and Locating Course Practice Files

1. Open the zip file containing the course practice files.
2. Extract the files to your My Documents\My SAS Files folder.
3. Verify that all the practice files were extracted correctly.

This demonstration shows you how to create and locate practice files for this course in the Windows file system.

Put the zip file on your desktop.

1. Now we need to perform a critical task for using this course: creating and locating the practice files you’ll use for SAS programming throughout the course. If you’re not familiar with extracting files from a ZIP file, pay close attention to this demonstration and feel free to play it over. But don’t worry — you’ll get step-by-step instructions in the upcoming practice.
2. Open SASIntro.zip file in WinZip.
3. Extract the files to My Documents. Make sure the Use Folder Names option is selected.
4. WinZip opens the MyDocuments\My SAS Files\SASIntro folder. Double-click PracticeFiles.
5. Notice that there are .dat files as well as SAS files.
6. Notice the .sas7bdat extension, which indicates a SAS data set.

For information about other operating environments, see Creating and Locating Course Practice Files in Appendix C or D.
This demonstration shows you how to view the raw data file `pilot dot dat`. Put `pilot.dat` in `MyDocuments\My SAS Files\SASIntro\PracticeFiles`

1. In Windows Explorer, verify that the `MyDocuments\My SAS Files\SASIntro\PracticeFiles` folder is open.
2. Double-click `Pilot.dat`.
3. Examine the raw data file in WordPad. Notice that the file contains only data, and that some of the columns look run together. For example, there appears to be a column of first names but it has some other column or columns immediately to its left. Other columns look freestanding.

For information about other operating environments, see `Opening Raw Data Files` in Appendix C or D.
This demonstration shows you how to open and view a SAS data set. Put `pilotdata.sas7bdat` in your `Work` folder.

1. Open `Work.Pilotdata` in SAS.
2. Examine the data portion of the Pilotdata data set in SAS. There are 6 columns. Scroll down to see that there are 50 observations, and then scroll back up.
3. What is the value for the variable `LastName` in observation 20? It is `ECHOLS`.
4. You can view the column attributes individually for each variable. For example, this is the `Column Attributes` window for the variable `LastName`.
5. Open the Properties window for the data set. Examine the General tab, the Details tab, and the Columns tab.

For information about other operating environments, see `Opening SAS Program Files` in Appendix C or D.
This demonstration shows you how to open and view a SAS program file. Put pilot.sas on your desktop.

1. Right-click pilot.sas and select Open With, and then WordPad to open the file in WordPad. Notice that the file contains only SAS programming statements, not any other formatting. Only the indentation is preserved.
Lesson 3  Navigating and Using the SAS Interface
This demonstration shows you how to start a SAS session.
1. Double-click the desktop icon for SAS. When SAS opens, we see the interactive windowing environment. Notice the Results and Explorer windows on the left and the Log and Editor windows on the right.
2. In the Log window, notice that there are some messages including copyright notes and the version of SAS. One important piece of information is the site number. If you contact SAS technical support by telephone or Web site for assistance, SAS requests this site number. You might want to make a note of that number once you start SAS.
3. What is the site number in the Log window shown here? It is 7857162208.
This demonstration shows you how to navigate SAS windows and explores some editing features of the Editor and Program Editor windows.
Make sure your Results formats are set to listing only.
The program is m212_3_d.sas.

Type this program into the **Editor** window and submit it.

```sas
data survey;
  infile 'survey.dat';
  input Initials $ Gender $ State $ Years Profession $;
run;
proc print data=survey;
  var gender profession years;
run;
```

1. After the program runs, the **Output** window moves to the front.
2. Now, let’s take a look at the code editors. Notice that the title bar is dimmed for an inactive window.
   Click the **Editor** window to activate it.
3. The **Editor** window provides many features to make coding easier. Notice in this code that the SAS program elements are color coded. To make your code easier to read, you can automatically indent the next line of your code when you press ENTER. You can view the overall flow of your SAS program or see each PROC and DATA set in detail. Just click the plus or minus icon.
4. Close the **Editor** window by clicking the **X** in the top-right corner of the window.
5. Open the **Program Editor** window by selecting **View⇒Program Editor**.
6. Paste the same program here.
7. You can set options to display line numbers or not, and you can toggle line numbers on and off.
   Using line numbers, we can edit code line by line. For example, we can type **D** to delete a line or **I** to
insert a line. Or we can type M to move a line, then type A or B on a line to indicate moving after or before that line.

8. Close the Program Editor window and activate the Log window.
9. Activate the Output window.
10. Activate the Results window.
11. Activate the Explorer window.

Leave everything as is for the next practice.
This demonstration shows you how to issue commands in different ways to clear the contents of your SAS windows.

1. Our SAS programming windows still contain information from the previous demonstration, where we submitted a SAS program. Now we’ll see how to issue commands using the toolbar, the menus, and the command bar. SAS provides several different ways to handle most tasks, and the same is true for issuing commands.

2. Activate the **Editor** window, and then the **Log** window, and then the **Output** window. To manage this information we might want to clear the contents of these windows periodically.

3. Activate the **Log** window and click the New icon.

4. Activate the **Output** window and click the New icon.

5. Activate the **Editor** window and click the New icon.

6. Activate the original **Editor** window.

7. Select **Edit ⇒ Clear All**.

8. Activate the new **Editor** window. Type this code:

   ```
   proc print data=survey;
   run;
   ```

9. Then, type clear in the command bar and press ENTER.

For information about other operating environments, see **Issuing SAS Commands** in **Appendix C** or **D**.
Assigning Function Keys

This demonstration shows you how to assign function keys using the KEYS window. SAS assigns default function key settings, but you can easily add, edit, or delete them. Let’s use the KEYS window to assign a command to one of our function keys.

1. Type `keys` in the command bar and press ENTER.
2. Notice the two columns in the Keys window – the Key column and the Definition column.
3. In the Definition column for the F-12 key, type `clear`. Then close the KEYS window to save the change.
4. Type `keys` in the command bar again. Verify that the new definition of F-12 has been saved. Close the KEYS window.

For information about other operating environments, see Issuing SAS Commands in Appendix C or D.
This demonstration shows you how to enter and submit a SAS program. Start with SAS windows cleared.

As you learned earlier, you can use the Editor window or the Program Editor window to enter your programming code. Here we’ll use the Editor window.

1. Type this program into the Editor window:

```sas
data pilotdata;
  infile 'pilot.dat';
  input EmployeeID $ 1 - 6
    FirstName $ 7 - 19
    LastName  $ 20 - 34
    JobCode   $ 35 - 41
    Salary    42 - 47
    Category  $ 48 - 50;
run;
proc print data=pilotdata;
run;
```

2. Zoom the Editor window to see the entire program.

3. When you write SAS programs, you can enter code in free format. That is, SAS statements can begin and end in any column, be on the same line as other statements, extend over several lines, can be in uppercase or lowercase. However, for readability, most SAS programs follow a suggested standard. Our program here follows the suggested standard. Notice that DATA, PROC, and RUN statements begin in or near column one; all other statements are indented by several columns for readability, a blank line separates steps, and only one statement appears on a line.
4. Look at the INPUT statement, which specifies the variables to create from the raw data file. This statement illustrates another suggested standard, which is to define each SAS variable on a separate line and to align the beginning and ending position numbers.

5. Click the running man tool on the toolbar.

6. Activate the Log window by clicking the Log button.

7. Scroll down to review the log.
This demonstration shows you how to review your program’s processing and check your code for errors.

1. Start with the program from the previous demo on the Editor window and all other windows clear.
2. Change the keyword for the INFILE statement to IN.
3. Submit the program.
4. Activate the log and notice the error message about the IN keyword.
5. Scroll down in the log until you reach another error message about the INFILE statement.
6. If SAS can figure out an error, it will try correcting it. For example, if we had specified I-N-F-L-I-E instead of INFILE, SAS would make an educated guess, and then give us this warning: **Assuming the symbol INFILE was misspelled as INFLIE.**
7. Next, notice the NOTE in blue saying that SAS stopped processing.
Finally, notice the WARNING in green saying that the output data set might be incomplete.
This demonstration shows you how to correct an error in a SAS program. Start with error message in the Log from the last demo.

1. Clear the **Log** window.
2. Clear the **Editor** window and paste in this code:

   ```sas
   data pilotdata;
     infile 'pilot.dat';
     input EmployeeID $ 1 - 6
         FirstName $ 7 - 19
         LastName $ 20 - 34
         JobCode $ 35 - 41
         Salary        42 - 47
         Category $ 48 - 50;
   run;
   ```

3. Add a pound sign to the `EmployeeID` variable, and submit the program.
4. Check the SAS Log and notice the error messages, the note, and the warning.
5. In the **Editor** window, fix the error and resubmit the program.
6. Check the Log again.
7. Clear the **Log** again.

For information about other operating environments, see **Recalling and Modifying SAS Program Code** in Appendix C or D.
This demonstration shows you how to save and how to retrieve a SAS program. Start from the ending state of the last demo, with the Log window activated.

1. Activate the Editor window.
2. Select File $\Rightarrow$ Save As, and then select \My SAS Files\SASIntro\PracticeFiles, type the name pilotdata.sas, and click Save.

For information about other operating environments, see Saving and Retrieving SAS Code in Appendix C or D.
Lesson 4  Working with SAS Libraries, SAS Data Sets, and the Import Wizard
This demonstration shows you how to view SAS libraries and their contents using the Explorer window. Have the Cargorev and Pilotdata data sets in your Sasuser library. Move other files (except Profile catalogs) out of Sasuser for the moment.

1. Activate the Explorer window. If the Explorer window isn’t open in your SAS windowing environment, you can display the window by selecting View ➔ Explorer.
2. In the Explorer window, double-click Libraries. Select the view menu to see the various display options such as large icons, small icons, or detailed list.
3. Double-click the Work library. Notice that it is empty. Click the Up One Level tool.
4. Double-click Sasuser. You see two data sets: Cargorev and Pilotdata. (You might have other data sets in your Sasuser library.)
5. Notice the Profile catalog in Sasuser.
6. Double-click Cargorev to view the data it contains.
7. Right-click the Cargorec data set icon and select Properties from the menu. Notice that the data set contains 50 rows and 4 columns.
8. Click the Columns tab to see information about the columns, such as their name, type, and length.
9. Close the windows and navigate back to the top level of the Explorer window.

For information about other operating environments, see Using the Explorer Window in Appendix C or D.
This demonstration shows you how to define a permanent SAS library using the New Library window. If you don’t work in the Windows operating environment, you can skip this demonstration.

1. Activate the **Explorer** window, and then click the **New** tool.
2. In the new window, click **Library**, and then **OK**.
3. Enter the libref **Mydocs**.
4. Select **Enable at Startup**.
5. Enter the path **C:\Documents and Settings**.
6. Click **OK**.

Double-click the **Libraries** icon and notice that **Mydocs** has been defined.
This demonstration shows you how to define a permanent SAS library by submitting a LIBNAME statement.
1. In the Editor window, type the following code:
   ```
   libname sasintro "C:\Documents and Settings";
   ```
2. Click the Submit tool.
3. Check the log. Notice that the libref was successfully assigned.
This demonstration shows you how to create a permanent SAS data set. 
Start with the LIBNAME statement from the previous demo in the Editor window, along with additional DATA step code:

```sas
data revenuedata;
  infile 'revenue.dat';
  input Flight $ 1 - 3
      Category $ 6 - 8
      PassengerRevenue 12 - 18
      CargoRevenue 22 - 27;
run;
```

1. Change the DATA statement to make the data set name a two-level name that includes SASIntro.
2. Highlight the code and click the Submit tool.
Check the log. Notice that the new permanent data set has been created.
This demonstration shows you how to import Microsoft Excel data using the SAS Import Wizard. Put DFWLAX.xls in your My Documents\My SAS Files\SASIntro\PracticeFiles folder.

1. Suppose we have a Microsoft Excel spreadsheet of flight data that we want to be able to process in SAS. We want to import that data from Microsoft Excel into a SAS data set, so we’ll use the SAS Import Wizard. The name of the Excel file is DFWLAX.xls. We want to create a temporary SAS data set named DFWLAX in the Work library.
2. In SAS, select File ➔ Import Data. Make sure Standard data source is selected, and that Microsoft Excel 97, 2000, or 2002 Workbook is selected.
3. Click Next.
4. Click Browse to locate DFWLAX.xls.
5. Click Open and then OK.
6. DFWLAX.xls is listed as the table to import. Click Next.
7. Leave Work as the library and type DFWLAX as the member name. Click Next.
8. Browse to the location where you want to save the SAS code. Save the code as import.sas in My Documents\My SAS Files\SASIntro\PracticeFiles. Make sure that Replace file, if exists, is checked and click Finish.
9. View the log to check the import process. Notice that Work.DFWLAX was created.

Use the Explorer window to open the Work library and the new DFWLAX data set.
This demonstration shows you how to specify listing results, HTML results, or both. Have Survey data set in Work.
Make sure only listing output is selected.
Make sure Normal style is selected for HTML output.
Use this code to create reports:

```sas
proc print data=survey;
title 'The PRINT Procedure';
footnote;
run;
proc freq data=survey;
title1;
footnote;
tables state;
run;
proc means data=survey mean std;
title1;
footnote;
var years;
class profession;
run;
```

1. Submit the program.
2. Scroll through output in the Output window.
3. Select Tools ➔ Options ➔ Preferences and click the Results tab.
4. Deselect Create listing and select Create HTML.
5. Click OK.
6. Resubmit the program and view output in the Results Viewer window.
7. Clear all content in the Results window.
8. Navigate to the **Results** tab again. Leave **Create HTML** selected and select **Create listing**. Click **OK**.

9. Resubmit the program again.

10. Expand the tree in the **Results** window to show the two types of output for each PROC step. Click each report icon (first listing, and then HTML) for each PROC step to show reports.
Lesson 5  Using PROC PRINT to Create List Reports
This demonstration shows you how to create a list report using the PROC PRINT step.  
1. Type this program:  
   ```sas
   proc print data=sasuser.pilotdata;
   run;
   ```
2. Submit the program and check the log. 
3. Examine the new list report in the **Output** window.
This demonstration shows you how to modify your list report by using the VAR statement in the PROC PRINT step.
Start with the list report from the last demo in the **Output** window.

1. Activate the **Editor** window to show the program from the last demo. Add the following line of code between the PROC PRINT statement and the RUN statement:
   ```
   var employeeid firstname lastname jobcode;
   ```
2. Submit the program and check the log.
3. Examine the new list report in the **Output** window.
Lesson 6 Using PROC PRINT to Enhance List Reports
This demonstration shows you how to sequence a SAS data set using PROC SORT. Start with this code in the Editor window:

```sas
proc print data=sasuser.pilotdata;
  var employeedid firstname lastname jobcode;
run;
```

1. Add the following lines of code before the PROC PRINT step:

```sas
proc sort data=sasuser.pilotdata out=sortedpilotdata;
  by lastname firstname;
run;
```
2. Change the PROC PRINT statement to this:

```sas
proc print data=sortedpilotdata;
```
3. Submit the program and check the log.
4. Examine the new list report in the Output window.
This demonstration shows you how to enhance a list report by adding formats and labels to PROC PRINT code. Start with the code from the previous demo.

1. Change the VAR statement as follows:
   ```
   var employeeid firstname lastname salary;
   ```

2. Add a LABEL statement to label EmployeeID as ID Number, FirstName as First Name, LastName as Last Name, and Salary as Salary.
   ```
   label EmployeeID='ID Number'
   FirstName='First Name'
   LastName='Last Name'
   Salary='Salary';
   ```

3. Add the SPLIT= option to the PROC PRINT statement:
   ```
   proc print data=sortedpilotdata split=' ';
   ```

4. Add a FORMAT statement to format the Salary variable to represent dollar amounts.
   ```
   format salary dollar8.;
   ```

5. Submit the program and check the log.

6. Examine the new list report in the **Output** window. Notice the column labels and how those labels are split. Notice the Salary column.
Lesson 7  Using the DATA Step to Read Raw Data
This demonstration shows you how to locate and view the raw data file you’ll read. Start with SAS session where SASIntro has already been defined as C:\Documents and Settings. Have Windows Explorer open to MyDocuments\My SAS Files\SASIntro\PracticeFiles and have pilot.dat in this folder.

1. Open the SASIntro library. Remember that you want to name your data set Pilotdata and store it in the SASIntro library.
2. Navigate to the raw data file pilot.dat.
3. Open the raw data file. Examine the raw data. If you try to figure out the data in this file, you can see some first names and last names. You might also be able to figure out the job codes. But beyond that, based only on what you see, you would have to guess. That’s why the person that created the raw data file needs to give you more information about the layout, including the field names and locations.
This demonstration shows you how to submit a DATA step and examine your results. Start with this DATA step in the **Editor** window:

```sas
data sasintro.pilotdata;
   infile 'pilot.dat';
   input EmployeeID $ 1 - 6
       FirstName $ 7 - 19
       LastName  $ 20 - 34
       JobCode   $ 35 - 41
       Salary    42 - 47
       Category  $ 48 - 50;
run;
```

1. **Submit the DATA step.**
2. **Check the SAS log.** You see the code that you submitted, and then a reference to the raw data file and its file characteristics. Then you see that the new data set was created.
3. **Navigate to the SASIntro library and open it.** Scroll to find the **Pilotdata** data set. Open the data set to verify that the data was read correctly.
This demonstration shows you how to create a new variable using an assignment statement. Start with this DATA step in the Editor window:

```sas
data sasintro.pilotdata;
  infile 'pilot.dat';
  input EmployeeID $ 1 – 6
    FirstName $ 7 – 19
    LastName $ 20 – 34
    JobCode $ 35 – 41
    Salary 42 – 47
    Category $ 48 - 50;
run;
```

1. Add the following assignment statement after the INPUT statement:
   ```sas
   Bonus=salary*0.10;
   ```
2. Submit the revised DATA step and check the log.
3. Notice that the log shows that the DATA step ran successfully and created the data set Pilotdata. Notice the number of observations in the data set.
4. Add a PROC PRINT step:
   ```sas
   proc print data=sasintro.pilotdata;
   run;
   ```
5. Submit the step and view the report.
Lesson 9   Using Conditional Logic
This demonstration shows you how to create a new variable using IF-THEN/ELSE statements.

Start with this DATA step in the Editor window:

```sas
data sasintro.pilotdata;
  infile 'pilot.dat';
  input EmployeeID $ 1 – 6
     FirstName $ 7 – 19
     LastName $ 20 – 34
     JobCode $ 35 – 41
     Salary 42 – 47
     Category $ 48 - 50;
  Bonus=salary*0.10;
run;
```

1. Add the following IF_THEN_ELSE statements after the assignment statement for Bonus:
   ```sas
   if JobCode = 'PILOT1' then
     NewSalary = Salary * 1.05;
   else if JobCode = 'PILOT2' then
     NewSalary = Salary * 1.07;
   else if JobCode = 'PILOT3' then
     NewSalary = Salary * 1.09;
   
   run;
   ```

2. Submit the program and check the log.
3. Notice that the log shows that the Pilotdata SAS data set was created and contains eight variables.
4. Add a PROC PRINT step to print the data set and submit it.
   ```sas
   proc print data=sasintro.pilotdata;
run;
   ```
5. View the report and notice the new variable, which appears last in the data set.
6. But let’s make one change to the DATA step. Three variables—Salary, Bonus, and New Salary—represent dollar amounts. So let’s add a FORMAT statement to format these three variables with a dollar sign and two decimal places. You learned about the FORMAT statement in the lesson on PROC PRINT. If we add the FORMAT statement in the PROC PRINT step, it only applies to this report. But if we add it in the DATA step, the format applies permanently to these variables.
7. Add a FORMAT statement to the DATA step to format Salary, Bonus, and NewSalary with a dollar sign to two decimal places.
   
   format salary bonus newsalary dollar12.2;
8. Submit the DATA step again, and then run the PROC PRINT step again.
9. Look at the revised report.
Lesson 10 Using PROC MEANS to Create Summary Reports
Demonstration

Creating a Default Summary Report Using PROC MEANS

This demonstration shows you how to create a default summary report using PROC MEANS. Start with this PROC MEANS step in the Editor window:

```
proc means data=sasintro.pilotdata;
run;
```

1. Submit the program.
2. Look at the output: it shows a statistical report of the three numerical variables: **Salary**, **Bonus**, and **NewSalary**. For each, we see the statistics N for frequency, mean, standard deviation, minimum, and maximum.
This demonstration shows you how to use a VAR statement to limit the variables in a summary report created by PROC MEANS.

Start with this PROC MEANS step in the Editor window:

```
proc means data=sasintro.pilotdata;
run;
```

1. Add the following VAR statement to select `Salary` and `NewSalary`.

   ```
   var salary newsalary;
   ```

2. Submit the program and examine the report.
3. Look at the Log window. See the code that you submitted and the number of observations that were read.
This demonstration shows you how to group data in a summary report using PROC MEANS and a CLASS statement.

Start with this PROC MEANS step in the Editor window:

```sas
proc means data=sasintro.pilotdata;
   var salary newsalary;
run;
```

1. Add the following CLASS statement:

```sas
   class category;
```
2. Submit the program.
3. Examine the new report, which now gives statistics for groups within Salary and NewSalary.
This demonstration shows you how to select observations for a report.

Start with this PROC MEANS step in the Editor window:

```sas
proc means data=sasintro.pilotdata;
  var salary newsalary;
  class category;
run;
```

1. Copy and paste the step to create three separate PROC MEANS steps.
2. After each CLASS statement, add WHERE statements to each of the steps as follows:
   - for the first step: `where JobCode='PILOT2';`
   - for the second step: `where Category='DOM';`
   - for the third step: `where JobCode='PILOT2' and Category='DOM';`
3. Add a title for each step as follows:
   - for the first step: `title1 'Level-Two Pilots';`
   - for the second step: `title1 'Domestic Pilots';`
   - for the third step: `title1 'Level-Two Domestic Pilots';`
4. Submit the first PROC MEANS step and view the report.
5. Submit the second step and view the report.
6. Submit the third step and view the report.
Lesson 11   Using PROC FREQ to Create Frequency Reports
This demonstration shows you how to run a default one-way frequency report with PROC FREQ. Write a PROC FREQ step.

```
proc freq data=sasintro.pilotdata;
run;
```

2. Submit the step and view the output.
3. Scroll through the output to see all the employee IDS, all the first names, all the last names, and so on.
4. **JobCode** is more interesting, with only three unique values. Here you can see how many pilots are in each job category.
   **Category** is another useful variable, with only two unique values.
This demonstration shows you how to create a simple frequency report using PROC FREQ and the TABLES statement.

Start with this PROC FREQ step in the Editor window:

```sas
proc freq data=sasintro.pilotdata;
run;
```

1. Add the following TABLES statement to specify the variable **Category**:
   ```sas
tables category;
```
2. Clear the log and output windows.
3. Submit the program
4. Examine the new output.
5. Go back to the Editor and revise the TABLES statement so that it includes two variables: **Category** and **JobCode** (separated with a space).
   ```sas
tables category jobcode;
```
6. Submit the code again and view the output.
7. Now you see two tables, one for **Category**, and one for **JobCode**.
This demonstration shows you how to create a two-way frequency report with PROC FREQ. Start with this PROC FREQ step in the Editor window:

```
proc freq data=sasintro.pilotdata;
  tables category jobcode;
run;
```

1. Add an asterisk between the two variable names:
```
tables category*jobcode;
```
2. Submit the program.
3. Examine the output. First, notice the legend in the top-left corner. This explains the statistics in the cells of the table. Look at the table cell for domestic flights and **PILOT1**.
   a. The first statistic is frequency—the number of domestic flights flown by pilots with the job code **PILOT1**. In this case, **PILOT1** pilots flew 10 domestic flights.
   b. Next is percent of the total—10 flights represent 20% of all flights.
   c. Next is the row percent—these 10 flights are 52.63% of all domestic flights.

Finally, you have column percent—these 10 flights are 100% of all flights for **PILOT1** pilots. So pilots at Level One are flying only domestic flights, not international flights.
This demonstration shows you how to enhance PROC FREQ output by adding titles and footnotes. Start with this PROC FREQ step in the Editor window:

```sas
proc freq data=sasintro.pilotdata;
  tables category jobcode;
  title1 'First Title Line for Two-Way Table';
  title2 'Second Title Line for Two-Way Table';
  footnote1 'First Footnote Line for Two-Way Table';
  footnote2 'Second Footnote Line for Two-Way Table';
run;
```

1. Submit the program and view the output.
2. Notice the two footnotes at the bottom of the page.
3. Notice the two titles at the top of the output.
4. In the Editor window, change the title statements to a null title statement: `title;`.
5. Then, change the footnote statements to a null footnote statement: `footnote;`.
6. Submit the program again and view the output. There are no titles or footnotes except the title that PROC FREQ displays.