B5.1 - Enhancing Report with Titles, Footnotes, and Labels

Now that data access, validation, and manipulation are behind us, we are finally ready to address what may be considered the peak of the programming process: analyzing and reporting on the data. Analyzing your data can mean a lot of different things. It could be basic summarization to examine what has happened in the past, or it could be complex data mining or machine learning algorithms to predict what will happen in the future. In this lesson, we concentrate on summarizing data. Specifically, you learn more about using the procedures that we touched on for exploration: PRINT, MEANS, and FREQ. You also learn how to use titles, column labels, footnotes, and macro variables to enhance your reports and make them more meaningful.

Using Titles and Footnotes

First let's learn about some additional SAS statements that can be used with any procedure to enhance a report.

TITLE is a global statement that establishes a permanent title for all reports created in your SAS session. The syntax is just the keyword TITLE followed by the title text enclosed in quotation marks. You can have up to 10 titles. You specify a number 1 through 10 after the keyword TITLE to indicate the line number. TITLE and TITLE1 are equivalent.

You can also add footnotes to any report with the FOOTNOTE statement. The same rules for titles apply to footnotes.

Using Titles and Footnotes

TITLE<n> "title-text";   FOOTNOTE<n> "footnote-text";

```
title1 "Class Report";
title2 "All Students";
footnote1 "School Use Only";

proc print data=pg1.class_birthdate;
run;
```

Class Report
All Students

Obs	Name	Sex	Age	Height	Weight	Birthdate
1	Alfred	M	14	69.0	112.5	16370
2	Alice	F	13	56.5	84.0	16756
3	Barbara	F	13	65.3	98.0	16451
4	Carol	F	14	62.8	102.5	16256
5	Henry	M	14	63.5	102.5	16406
6	James	M	12	57.3	83.0	16967
7	Jane	F	12	59.8	84.5	16873
8	Janet	F	15	62.5	112.5	15797
9	Jeffrey	M	13	62.5	84.0	16552
10	John	M	12	59.0	99.5	17036
11	Joyce	F	11	51.3	50.5	17169
12	Judy	F	14	64.3	90.0	16410
13	Louise	F	12	56.3	77.0	17021
14	Mary	F	15	66.5	112.0	15790
15	Philip	M	16	72.0	150.0	15665
16	Robert	M	12	64.8	128.0	16958
17	Ronald	M	15	67.0	133.0	15992
18	Thomas	M	11	57.5	85.0	17243
19	William	M	15	66.5	112.0	16067

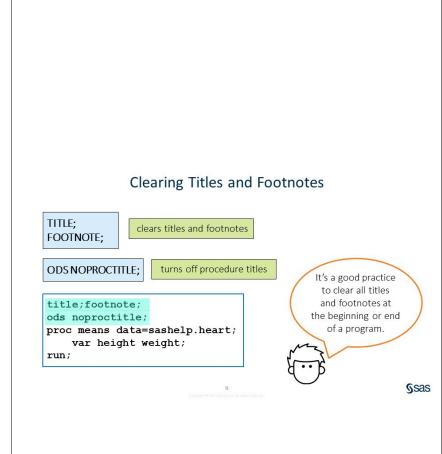School Use Only

4

p105d01   §sas

Activity 5.01

Open **p105a01.sas** from the **activities** folder and perform the following tasks:

1. In the program, notice that there is a TITLE statement followed by two procedures. Run the program. Where does the title appear in the output?
2. Add a TITLE2 statement above PROC MEANS to print a second line: **Summary Statistics for MaxWind and MinPressure**
3. Add another TITLE2 statement above PROC FREQ with this title: **Frequency Report for Basin**
4. Run the program. Which titles appear above each report?

Click here for Solution.

Activity 5.02

Open **p105a02.sas** from the **activities** folder. Notice that there are no TITLE statements in the code. Run the program. Does the report have the same titles assigned in the previous activity? Yes or No
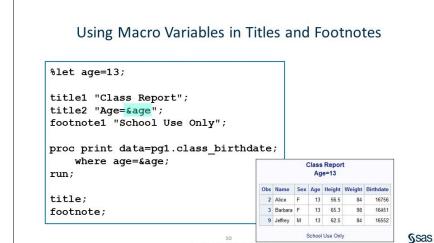
> Click here for Solution.

Remember that TITLE and FOOTNOTE are global statements, and they remain active as long as your SAS session is active. If you want to clear the titles and footnotes you've specified, you can use the keyword TITLE or FOOTNOTE with no text. That's called a null TITLE statement. The null TITLE statement clears all the titles that you've specified on any line. It's a good idea to do this at the end of your program. Client applications such as SAS Studio submit a null TITLE statement for you at the end of your code, but it's a good idea to get in the habit of submitting the statement yourself.
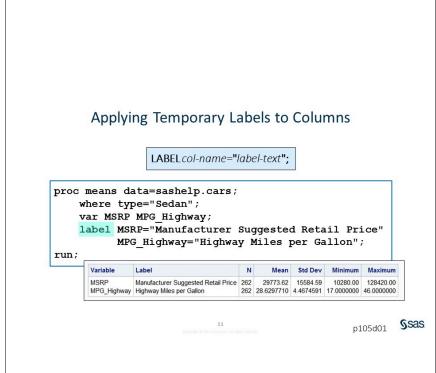
Some procedures include the name of the procedure in a title above the results. You can turn this off my submitting an ODS statement with the NOPROCTITLE option. You do more with ODS in another lesson.

Using Macro Variables in Titles and Footnotes

Using Macro Variables in Titles and Footnotes

```
%let age=13;

title1 "Class Report";
title2 "Age=&age";
footnote1 "School Use Only";

proc print data=pg1.class_birthdate;
    where age=&age;
run;

title;
footnote;
```

Class Report
Age=13

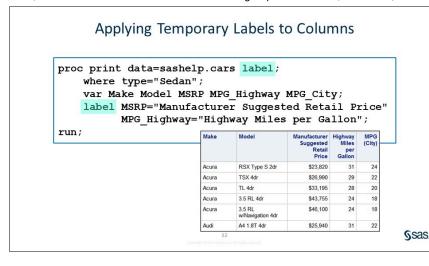| Obs	Name	Sex	Age	Height	Weight	Birthdate
2	Alice	F	13	56.5	84	16756
3	Barbara	F	13	65.3	98	16451
9	Jeffrey	M	13	62.5	84	16552

School Use Only

10

Macro variables are very helpful in titles. Suppose we have a %LET statement to create a macro variable for age that we're using in a filter in the PRINT procedure. It would be helpful to include the filter criteria in a title so that it is clear what filter has been applied. We can use the macro variable in the title to do just that. Remember that you must enclose the macro variable in double quotation marks so that SAS replaces the macro variable reference with the stored text.

Temporary Labels

Applying Temporary Labels to Columns

LABEL *col-name*="*label-text*";

```
proc means data=sashelp.cars;
    where type="Sedan";
    var MSRP MPG_Highway;
    label MSRP="Manufacturer Suggested Retail Price"
          MPG_Highway="Highway Miles per Gallon";
run;
```

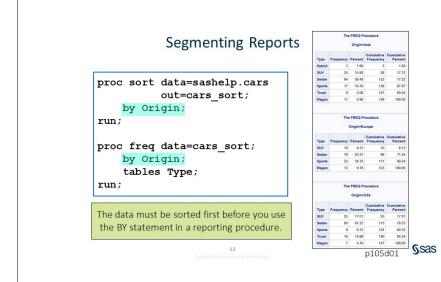| Variable	Label	N	Mean	Std Dev	Minimum	Maximum
MSRP	Manufacturer Suggested Retail Price	262	29773.62	15584.59	10280.00	128420.00
MPG_Highway	Highway Miles per Gallon	262	28.6297710	4.4674591	17.0000000	46.0000000

11

p105d01

Column names must adhere to particular naming conventions, but that means sometimes the names might be a bit difficult to interpret, especially for someone that isn't familiar with the data. Labels are an easy way to enhance a report with more descriptive column headings. A label can be any text string up to 256 characters, including spaces and special characters.

You can use the LABEL statement in procedures to improve your reports. You start with the keyword LABEL, and then list the column name, equal sign, and the label string enclosed in quotation marks. You can label more than one column in a single LABEL statement.

Applying Temporary Labels to Columns

```
proc print data=sashelp.cars label;
    where type="Sedan";
    var Make Model MSRP MPG_Highway MPG_City;
    label MSRP="Manufacturer Suggested Retail Price"
          MPG_Highway="Highway Miles per Gallon";
run;
```

Make	Model	Manufacturer Suggested Retail Price	Highway Miles per Gallon	MPG (City)
Acura	RSX Type S 2dr	$23,820	31	24
Acura	TSX 4dr	$26,990	29	22
Acura	TL 4dr	$33,195	28	20
Acura	3.5 RL 4dr	$43,755	24	18
Acura	3.5 RL w/Navigation 4dr	$46,100	24	18
Audi	A4 1.8T 4dr	$25,940	31	22

12

§sas

PROC MEANS, PROC FREQ, and most other procedures automatically display the labels in the results. PROC PRINT is an exception. Because the main purpose of PROC PRINT is to examine the data, column names are always displayed by default. To display labels instead, you must add the LABEL option in the PROC PRINT statement.

Segmenting Reports

Segmenting Reports

```
proc sort data=sashelp.cars
          out=cars_sort;
    by Origin;
run;

proc freq data=cars_sort;
    by Origin;
    tables Type;
run;
```

The data must be sorted first before you use the BY statement in a reporting procedure.

The FREQ Procedure

Origin=Asia

Type	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Hybrid	3	1.90	3	1.90
SUV	25	15.82	28	17.72
Sedan	94	59.49	122	77.22
Sports	17	10.76	139	87.97
Truck	8	5.06	147	93.04
Wagon	11	6.96	158	100.00

The FREQ Procedure

Origin=Europe

Type	Frequency	Percent	Cumulative Frequency	Cumulative Percent
SUV	10	8.13	10	8.13
Sedan	78	63.41	88	71.54
Sports	23	18.70	111	90.24
Wagon	12	9.76	123	100.00

The FREQ Procedure

Origin=USA

Type	Frequency	Percent	Cumulative Frequency	Cumulative Percent
SUV	25	17.01	25	17.01
Sedan	90	61.22	115	78.23
Sports	9	6.12	124	84.35
Truck	16	10.88	140	95.24
Wagon	7	4.76	147	100.00

13

p105d01

§sas

We can use the BY statement in a reporting procedure to segment a report based on the unique values of one or more columns. For example, what if I want to generate a separate frequency report for each value of **Origin**? I must sort the table by **Origin** first, and then use the BY statement in PROC FREQ. Then SAS treats the rows for each value of **Origin** as a separate table and runs the frequency report.
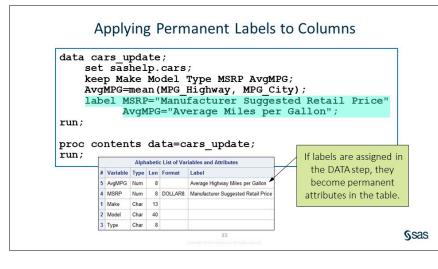
Demo: Enhancing Reports

5_1 - Demo - Enhancing Reports.pdf

(https://clemson.instructure.com/courses/237270/files/23073936/download?wrap=1) ⬇

(https://clemson.instructure.com/courses/237270/files/23073936/download?download_frd=1) (i)

Permanent Labels

Applying Permanent Labels to Columns

```
data cars_update;
    set sashelp.cars;
    keep Make Model Type MSRP AvgMPG;
    AvgMPG=mean(MPG_Highway, MPG_City);
    label MSRP="Manufacturer Suggested Retail Price"
          AvgMPG="Average Miles per Gallon";
run;

proc contents data=cars_update;
run;
```

If labels are assigned in the DATA step, they become permanent attributes in the table.

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Label
5	AvgMPG	Num	8		Average Highway Miles per Gallon
4	MSRP	Num	8	DOLLAR8.	Manufacturer Suggested Retail Price
1	Make	Char	13		
2	Model	Char	40		
3	Type	Char	8		

If we take the same LABEL statement and put it in a DATA step, labels are assigned to the designated columns as permanent attributes in the descriptor portion of the table. When procedures create reports using that data, labels are automatically displayed. However, you still need the LABEL option in PROC PRINT.

Activity 5.04

Open **p105a04.sas** from the **activities** folder and perform the following tasks:

1. Modify the LABEL statement in the DATA step to label the **Invoice** column as **Invoice Price**.

2. Run the program. Why do the labels appear in the PROC MEANS report but not in the PROC PRINT report? Fix the program and run it again.

Click here for Solution.