Homework 12, Defining Macros

1. **Defining and Calling a Macro**
   1. Copy the program shown below into the Editor window.

**proc print data=orion.customer\_dim;**

**var Customer\_Group Customer\_Name Customer\_Gender Customer\_Age;**

**where Customer\_Group contains 'Gold';**

**title 'Gold Customers';**

**run;**

* 1. Modify the program so that the two occurrences of Gold are replaced by references to the macro variable TYPE.
  2. Convert the program into a macro named CUSTOMERS. Set the appropriate system option to display a note in the SAS log when a macro definition has compiled. Submit the macro definition and examine the log.
  3. Submit a %LET statement to assign the value Gold to the macro variable TYPE. Call the macro and examine the log.
  4. Change the value of TYPE to Internet.
  5. Activate the appropriate system option to display source code received by the SAS compiler. Call the macro again and examine the log.

1. **Controlling Case-Sensitivity in a Macro** 
   1. Copy the program shown below into the Editor window.

**proc means data=orion.staff sum mean;**

**var Salary;**

**class Job\_Title;**

**where Job\_Title contains "Sales";**

**title "Summary Statistics for the Sales Group";**

**run;**

* 1. Convert the program into a macro named SUMSTATS. Replace occurrences of Sales with a macro variable reference. Modify the SUMSTATS macro so that the macro variable and the data set variable in the WHERE statement are not case-sensitive. Compile and execute the macro using sales as the value for the macro variable.
  2. Alter the macro variable reference in the TITLE statement to display the value in proper case.

1. **Calling a Macro in a TITLE Statement** 
   1. Following the steps outlined below, create a macro named FUTURE and use that macro to generate the text in the TITLE statement. The title should contain today’s date and the date one year from today.
      1. Define a macro named FUTURE.
      2. Create the macro variable DAYMON by extracting the day and month from the macro variable SYSDATE9.
      3. Create the macro variable YEAR by extracting the year from the macro variable SYSDATE9 and adding 1 to that value.
      4. Create the macro variable FUTUREDATE by concatenating DAYMON and YEAR.
      5. Reference the macro variables SYSDATE9 and FUTUREDATE.
      6. End the macro definition.
   2. Copy the program shown below into the Editor window and call the macro in the TITLE statement.

**proc print data=orion.customer\_dim(obs=10);**

**var Customer\_Name Customer\_Group;**

**title "Date range of ";**

1. Defining and Using Macro Parameters
   1. Copy the program shown below into the Editor window.

**%macro customer;**

**proc print data=orion.customer\_dim;**

**var Customer\_Name Customer\_Gender Customer\_Age;**

**where Customer\_Group contains "&type";**

**title "&type Customers";**

**run;**

**%mend customer;**

* 1. Convert this program into a macro with a positional parameter. Name the parameter based on macro variable references within the program. Set the appropriate system option to display a note in the SAS log when a macro definition has compiled. Submit the macro definition to compile the macro.
  2. Call the macro defined in the previous step with a value of Gold for the parameter.
  3. Call the macro again, but with a parameter value of Catalog.
  4. Change the positional parameter to a keyword parameter with a default value of Club. Submit the revised macro definition to compile the macro.
  5. Call the macro defined in the previous step with a value of Internet for the keyword parameter.
  6. Call the macro again, but allow the macro to use its default parameter value.

1. **Using a Macro to Generate SAS/GRAPH Code**

Consider the following two examples of producing bar charts with proc sgplot:

**proc** **sgplot** data=orion.customer\_dim;

vbar Customer\_Age\_Group/

fillattrs=( color=red transparency=**.25**);

**run**;

**proc** **sgplot** data=orion.customer\_dim;

hbar Customer\_Group/

fillattrs=( color=pink transparency=**.5**);

**run**;

Notice that you can produce both vertical (VBAR) and horizontal (HBAR) charts and also control color and transparency of the fill for each bar.

* 1. Copy the following program into the Editor window.

**proc** **sgplot** data=orion.customer\_dim;

vbar Customer\_Age\_Group/

fillattrs=( color=red transparency=**.25**);

run;

* 1. Create a macro with keyword parameters that generalize the code so that the following chart attributes are controlled by macro variables. All parameters should have default values so that   
     the code can execute properly.

Bar orientation – VBAR or HBAR

Bar Color

Bar Transparency

* 1. Execute the macro using the default parameter values.
  2. Call the macro again, but override all of the default parameter values.
  3. Call the macro again, but override the default parameter values except for the bar orientation   
     and color.

1. Using Parameters That Contain Special Characters
   1. Copy the program shown below into the Editor window. Submit the program to compile the macro.

**%macro specialchars(name);**

**proc print data=orion.employee\_addresses;**

**where upcase(Employee\_Name)="%upcase(&name)";**

**var Employee\_ID Street\_Number Street\_Name City State**

**Postal\_Code;**

**title "Data for &name";**

**run;**

**%mend specialchars;**

* 1. Execute the macro with a value of Abbott, Ray for the parameter.