HW 11: Macro Variables

Level 1

1. **Displaying Automatic Macro Variables** 
   1. Use the %PUT statement to list all automatic macro variables in the SAS log.

Level 2

1. **Using Automatic Macro Variables** 
   1. Using the SORT procedure, sort the data set **orion.customer** by **Country**. Use the OUT= option in the PROC SORT statement to create a dataset in the work library so that you do not overwrite the original data set.
   2. Using the PRINT procedure and an automatic macro variable, print the most recently created data set and display the data set name in the title.
   3. Submit the program and examine the results.
2. **Using Automatic Macro Variables** 
   1. What is the value of the automatic macro variable SYSLAST after the following DATA step is submitted? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**data new;**

**set orion.customer;**

**run;**

* 1. What is the value of the automatic macro variable SYSLAST after the following PROC PRINT step is submitted? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

**run;**

Level 3

1. **Using SAS Date Constants**
   1. Copy theprogram shown below into the Editor window.

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

**format Birth\_Date Employee\_Hire\_Date date9.;**

**run;**

* 1. Modify the program so that it subsets the data to return only the employees hired between   
     January 1, 2007, and today. Use the automatic macro variable SYSDATE9 to return today’s date.

PROC PRINT Output

Employee\_ Birth\_ Employee\_ Employee\_ Marital\_

Obs Employee\_ID Gender Salary Date Hire\_Date Term\_Date Status Dependents

310 121034 M 27110 23AUG1988 01JAN2007 . S 0

361 121085 M 32235 12NOV1986 01JAN2007 . S 0

364 121088 M 27240 10JUN1988 01JAN2007 . S 0

Level 1

1. **Defining and Using Macro Variables for Character Substitution**
   1. Copy the program shown below into the Editor window. Submit the program and examine the output that it creates.

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

**var 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. Precede the program with a %LET statement to assign the value Gold to TYPE. Submit the program. Varify that it produces the same output as before.
  2. Include the appropriate system option to display resolved values of macro variables in the SAS log. Resubmit the program and examine the log.
  3. Modify the value of TYPE to Internet. Resubmit the program and examine the log.
  4. Turn off the system option from part **c** above.

Level 2

1. **Defining and Using Macro Variables for Numeric Substitution** 
   1. Copy the program shown below into the Editor window. Edit the program to display only the Gold level customers between the ages of 30 to 45.

**🖉** Customer ages range from 19 to 73.

**%let type=Gold;**

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

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

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

**title "&type Customers";**

**run;**

SAS Output

Gold Customers between 30 and 45

Customer\_ Customer\_

Obs Customer\_Name Gender Age

3 Cornelia Krahl F 33

11 Oliver S. Füßling M 43

32 James Klisurich M 38

35 Viola Folsom F 38

40 Kyndal Hooks F 43

57 Rita Lotz F 43

75 Angel Borwick F 38

* 1. Modify the program so that the values 30 and 45 are replaced by references to the macro variables AGE1 and AGE2, respectively.
  2. Include the appropriate system option to display resolved values of macro variables in the SAS log. Resubmit the program and examine the log.
  3. Modify the values of AGE1 and AGE2, assign the value 25 to AGE1 and the value 40 to AGE2. Resubmit the program and examine the log.
  4. Turn off the system option from part **c** above.

1. **Deleting Macro Variables** 
   1. Copy the program shown below into the Editor window. Submit the program to create the macro variables.

**%let pet1=Paisley;**

**%let pet2=Sitka;**

* 1. Add the statement to delete the user-defined macro variables, pet1 and pet2.
  2. Use the %PUT statement to verify that the macro variable deletion was successful.

Level 1

1. **Consecutive Macro Variables References**
   1. Copy the program shown below into the Editor window. Submit the program and examine the output that it creates.

**proc print data=orion.employee\_payroll;  
 where Employee\_Hire\_Date='01AUG2006'd;  
 id Employee\_ID;  
 var Employee\_Gender Employee\_Hire\_Date;  
 title 'Personal Information for Employees Hired in AUG 2006';  
run;**

* 1. Modify the program so that the two occurrences of AUG and 2006 are replaced by references   
     to the macro variables MONTH and YEAR, respectively. Precede the program with a %LET statement to assign the value AUG to MONTH and the value 2006 to YEAR. Submit the program. It produces the same output as before.
  2. Modify the value of MONTH to JUL and YEAR to 2003. Resubmit the program.

Level 3

1. **Macro Variable References with Multiple Delimiters** 
   1. Copy the program shown below into the Editor window. This program analyzes the **orion.staff** data to find the employee with the most seniority within a job title. Submit the program and examine the output that it creates.

**proc sort data=orion.staff out=staffhires;  
 by Job\_Title Emp\_Hire\_Date;  
run;  
data FirstHired;**

**set staffhires;  
 by Job\_Title;  
 if First.Job\_Title;**

**run;  
proc print data=FirstHired;**

**id Job\_Title;  
 var Employee\_ID Emp\_Hire\_Date;  
 title "First Employee Hired within Each Job Title";**

**run;**

* 1. Using a macro variable, modify the program to return the employees with the most or least amount of seniority. Be sure to make any necessary modifications to the title.

Partial PROC PRINT Output

First Employee Hired within Each Job Title

Emp\_Hire\_

Job\_Title Employee\_ID Date

Account Manager 120746 01APR2002

Accountant I 120752 01AUG1975

Accountant II 120771 01DEC1976

Accountant III 120757 01JAN1974

Administration Manager 120104 01JAN1981

Applications Developer I 120797 01DEC1977

Applications Developer II 120796 01MAR1983

Applications Developer IV 120802 01JAN1978

Partial PROC PRINT Output

Last Employee Hired within Each Job Title

Emp\_Hire\_

Job\_Title Employee\_ID Date

Account Manager 120746 01APR2002

Accountant I 120761 01JUL2006

Accountant II 120754 01MAY2006

Accountant III 120755 01AUG1983

Administration Manager 121000 01DEC1993

Applications Developer I 120801 01JUL1999

Applications Developer II 120812 01AUG2001

Applications Developer IV 120794 01JUL2003

Level 1

1. **Using the %SUBSTR and %SCAN Functions**
   1. Submit a %LET statement to assign the value Anthony Miller to a macro variable named FULLNAME.
   2. Extract the first initial and last name, putting them together into a new macro variable as A. Miller. Use the %PUT statement to display the results.
2. **Using the %SYSFUNC Function**

Use the %PUT statement and the %SYSFUNC function to display the current date and time. Format the date with the MMDDYYP10.format and the time with TIMEAMPM. format.

Level 2

1. **Protecting Special Characters** 
   1. Copy the program shown below into the Editor window. Submit the program and examine the output that it creates.

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

**where Product\_Name contains "Jacket";**

**var Product\_Name Product\_ID Supplier\_Name;**

**title "Product Names Containing 'Jacket'";**

**run;**

* 1. Submit a %LET statement to assign the value R&D to a macro variable named PRODUCT. Use the new macro variable in the WHERE statement and the TITLE statement. Submit the modified program.

PROC PRINT Output

Product Names Containing 'R&D'

Obs Product\_Name Product\_ID Supplier\_Name

393 Top Men's R&D Ultimate Jacket 240300300070 Top Sports Inc

395 Top R&D Long Jacket 240300300090 Top Sports Inc

* 1. Add the current date and time to a title. Format the date with the MMDDYYP10. format and the time with the TIMEAMPM. format. Remove any extra blanks in the title.

PROC PRINT Output

Product Names Containing 'R&D'

Report generated at 1:55:17 PM

on 02.08.2008

Obs Product\_Name Product\_ID Supplier\_Name

393 Top Men's R&D Ultimate Jacket 240300300070 Top Sports Inc

395 Top R&D Long Jacket 240300300090 Top Sports Inc

Level 3

1. **Verifying a Data Set Name**

Verify that the first character of a data set name is a valid value. Begin by creating a macro variable named DSN with a value of work.test and extract the first letter of the data set name into a macro variable named FIRST. With the aid of the SAS Help facility, specifically the Macro Language Dictionary, locate an autocall macro that will verify the value of FIRST against a list of valid values.

Recall that Data set names can begin with a letter or underscore and are not case sensitive.

* If the value is **correct** the output should be as follows:

SAS Log

The value of ckbegin=0

* If the value is **incorrect** the output should be as follows:

SAS Log

The value of ckbegin=1