# Macro Overview

Mihaela Simion

# Macro Facility Overview

Definition :

The SAS Macro Facility is a tool within base SAS software that contains the essential elements which enables you to use the macros.

# Properties of the MF

- The MF contains a macro processor that translate macro code into statements which can be used by SAS System, and the macro language.
- The macro language ML provides the means to communicate with the macro processor.
- The macro language consists of its own set of commands, options, syntax and compiler.
- There are syntax differences between the SAS Macro and SAS Base.

# Benefits of using the MF

The Macro language provides tools that:

- Pass information between SAS steps.
- Dynamically create code at the execution time.
- Conditionally execute DATA or PROC steps.
- Create generalization and flexible code.

# A Program Example

- This program is a great example because will show you how some of this tools are working, in order to access MF.
- Using Macro variable and subroutines I was able to provide a great automatization to a long process, full of many repetitive tasks.
- The objective was to provide multiple reports (<50) to different branch managers by e-mail or saving them in a share drive.

# Options

```
options nomprint nomlogic nosymbolgen;
*options mprint mlogic symbolgen;
```

***Debugging a macro can be, under the best of conditions, difficult.***

***You could use these two options statement  at the beginning of the program, that contains:***

***MPRINT= displays the test or SAS statements that are generated by macro executions, one statement per line, with macro variable references solved.***

***MLOGIC=traces the macro logic and follows the  pattern of execution.***

***SYMBOLGEN=print a message in the LOG***

***whenever a macro variable is resolved.***

# Step 1

```
data _null_;
format yer yer1 4. mth mth1 2.;
 dat=“&sysdate”d;
 mth =month(dat);
 yer=year(dat);
if mth=1 then do;
    yer=yer-1;
    mth=12;
 end;
 else do;
   mth1=mth-1;
    yer1=yer;
  end;
   dat2=mdy(mth,1,yer1);
call
      symput(‘prevmth’,put(dat2,monyy5.
      ));
run;
```

- ***Create a Macro variable for the previous month date using a data _null_ step.***
- ***In this step we use one of the most important macro function: &sysdate***
- ***In order to create the macro variable I called the symput subroutine:***

  ***call symput(‘prevmth’,put(dat2,monthyy7.));***

- ***The macro variable prevmth, that will be solved like simple text  will be used in the title of the report.***

# Log

```
102
103     options mprint mlogic symbolgen;
104
105      data _null_;
106      format yer yer1 4. mth mth1 2.;
107      dat="&sysdate"d;
SYMBOLGEN:  Macro variable SYSDATE resolves to 15SEP11
108      mth =month(dat);
109       yer=year(dat);
110      if mth=1 then do;
111          yer=yer-1;
112          mth=12;
113       end;
114       else do;
115         mth1=mth-1;
116          yer1=yer;
117          end;
118         dat2=mdy(mth,1,yer1);
119      call symput('prevmth',put(dat2,monyy5.));
120      run;

NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds
```

# Step2

```
%macro  totrep(data1,list,var1);
```

Open the macro function totrep (data1. list, var1) where:

data1=input data set.

list =the list of the control parameters.

By example in my case was transit#, district, region.

They could be used for the sorting purpose.

Could be missed when it is not need for special sorting.

Var1= the variable used for the creation of the reports (In my example was transit#).

When there are  multiple parameters, the comma separates their values.

# Step2

```
proc sort data=&data1:
 by &var1;
 run;
 data _null_;
 set &data1:
 by &var1;
 if first..&var1 then do:
 i+1;
 ii=left(put(i,2));
 call symput(
     ‘var1a’||ii,put(&var1,4.)));
end;
 cal symput( ‘total’,ii);
run;
```

***Create a macro variable for each value of the variable var1 :var1a1,var1a2,…***

***call symput(‘var1a’||ii,put(&var1,4.)));***

***Create a macro variable for the number of the reports we want to edit, named: total***

***Observe how is used the first variable in the if statement: double dots***

**if first..&var1 then do.**

***For the efficiency purpose I limited the number of the created macro variables to 50.***

# Step 3

```
%do i=1 %to %total:
 proc sort data=&data1:
 by &list;
 run;
 data  report ;
 set &data1:
 if var1 in (&&var1a&i);
 call symput(“var1b”,put(
     left(&var1),4.));
 run;
```

***Create  the loop that will create for each I to the total( number of values for the variable var1)***

- ***In the first part of the loop the data was sorted by list .We sort by list if it is another order needed.***
- ***Second part of the loop is a data step that***

  ***creates a temporary dataset name report and a macro variable:  var1b, used in the name of excel report.***

# A Macro Exercise

```
ods listing close ;
filename result1
      “H:\temp\audit&var1b..xls”;
ods html body=results1;
proc report data=report nowindows
headline headskip  split=‘*’ misssing out=result1;
 style( report)={ Background =white} style(header)= {foreground
      =dark blue background =whitte
 font_weight=bold font_size=1}
Style( column ) = { background=white font_size=1};
 @ line @1 “ region” @11 region 5. @17 “_”  @18 regname $28.;
 @ line @1 “ district” @11  dis_name 5. @17 “_”  @18 regname
      $28.;
 @ line @1 “ transit” @11transit 5. @17 “_”  @18 trn_name $28.;
 endcomp:
 compute after nat;
line @ 9 115*”_”; @38 Lnamount .sum dollar 17.; @58
      prbalos.sum doll1ar1.;
endcomp;
  title1;title2;title3;
  title4 “Monthly Report as at “&prevmth”;
 footnote1;
 footnote2 “The Report is addressed to the to the manager of
      the branch &var1b”;
```

- **We are still in the macro torep**
- **I open an ods (output delivery system) to build the report in *html* format or an excel file.**

# A Macro Exercise

```
ods html close;run;
file name mymail1 email “&var1”;
subject =“Monthly report for &var1”
attach =‘H:\temp\audit& var1b..xls”;
data _null_;
put mymail1; put ;
put “This is the monthly report
    for..”;run;
%end; %mend totrep;
```

- Close the ods for the report.
- Send the e-mail, inclosing the report.
- Close the loop %end;
- Close the macro %mend ;

# A Macro Exercise

**Calling the macro:**

```
%totrep(data1=plcdata, list= region district transit,var1=transit);
```

**or**

```
%totrep(plcdata,region district transit,transit);
```

**If we have an enumeration in one of  the parameters in the macro don’t use the comma between the components.**

**See the *list* enumeration in the example.**

# Some suggestions for improvements and application

- The proc report could be changed with PROC TEMPLATE.
- You could create a list with e-mails and transit numbers to be sent.
- You could also use a data set containing the data and the months that you could use instead of the previous month prevmth macro variable.

# Bibliography

- 1. Carpenter’s Complete Guide to the SAS Macro Language.
- 2. Advanced Macro Languages-SAS Institute.

*Have fun!*