

Work on Repetitive Tasks using SAS Macro

Presented by Yanguo Qin

The Forestry Corp.

October 11, 2007

Why SAS Macro?

- The program can become reusable, short, and easy to follow.
- The repetitive tasks can be accomplished quickly and efficiently.
- More.

(From SAS Macro Programming Made Easy, by Michele M. Burlew)

SAS Macro in Data Import - Example 1

- Purpose: import 10 dbf files to create one SAS file
- Data
 - ✓ plot1.dbf
 - ✓ plot2.dbf
 - ✓
 - ✓ plot10.dbf

```
%Macro Combinefiles;
```

```
  %do num = 1 %to 10;
```

```
    filename avi "c:\projects\plot&num..dbf";
```

```
    proc dbf db4=avi out=plots_&num;
```

```
    run;
```

```
  %end;
```

```
data allplots;
```

```
  set
```

```
  %do num = 1 %to 10;
```

```
    plots_&num
```

```
  %end;
```

```
  ;
```

```
run;
```

```
%Mend;
```

```
%Combinefiles
```

SAS Macro in Data Import - Example 2

- Purpose: import 12 dbf files to create one SAS file
- Data
 - ✓ psp_srd.dbf
 - ✓ psp_mwfp.dbf
 - ✓
 - ✓ psp_apl.dbf

```
%let pathname = c:\projects;

%let file1 = psp_srd.dbf;
%let file2 = psp_mwfp.dbf;
.....
%let file12 = psp_apl.dbf;

%let numfile = 12;

%Macro GetDataDictionary;
  %do f = 1 %to &numfile;
proc import out=tempdata&f
  datafile= "&pathname.\&&&file&f"
  dbms=dbf replace;
  getdeleted=no;
run;
  %end;
```

```
data allplots;  
  set  
    %do f = 1 %to &numfile;  
      tempdata&f  
    %end;  
  ;  
run;
```

```
%Mend;  
%GetDataDictionary
```

SAS Macro in Data Import - Example 3

- Purpose: create 7 permanent SAS files
- Data: 7 tables in file scup_v2_for_sas_import.mdb
 - ✓ blocks
 - ✓ plots
 - ✓
 - ✓ ecosite

```
libname p407 '\\silver\';
```

```
%let table1=blocks ;
```

```
%let table2=plots ;
```

```
%let table3=trees ;
```

```
%let table4=tallies ;
```

```
%let table5=exttrees ;
```

```
%let table6=line ;
```

```
%let table7=ecosite ;
```

```
%let survlib=p407;
```

```
%let n=7;
```

```
%Macro GetData;  
  %do n = 1 %to &n;  
proc import out=&survlib.&&table&n  
  datatable= "&&table&n"  
  dbms=access2000 replace;  
  database="\ \silver\scup_v2_for_sas_import.mdb";  
run;  
proc print data=&survlib.&&table&n(obs=10);  
run;  
  %end;  
  
%Mend;  
%GetData
```

SAS Macro in Data Export - Example 4

- Purpose: create 10 dbf files with meaningful names
- Results: 10 output files
 - ✓ tph400forest1.dbf
 - ✓ tph400forest2.dbf
 - ✓
 - ✓ tph1200forest2.dbf

```
%let tph1 =400;
%let tph2 =600;
.....
%let tph2 =1200;
%let numtph =5;

%let forest1 =1;
%let forest2 =2;
%let numfor =2;

%Macro Getlists;
  %do t = 1 %to &numtph;
    %do f = 1 %to &numfor;

data data&t&f;
  do tree=1 to &&&tph&t*50;
```

```
x=round(ranuni(0)*(sqrt(50*10000)), .01);
y=round(ranuni(0)*(sqrt(50*10000)), .01);
output;
end;
run;

proc export data=data&t&f
  outfile=
  "c:\projects\tph&&&tph&t..forest&&&forest&f...dbf"
  dbms=dbf replace;
run;
  %mend;
  %mend;

%Mend Getlists;
%Getlists
```

SAS Macro in Data Export - Example 5

- Purpose: put 8 outputs in 8 worksheets in an Excel file
- Results: one Excel file containing 8 worksheets
 - ✓ AP
 - ✓ AS
 -
 - ✓ SW

```

%let stratum1=AP    ;
%let stratum2=AS    ;
.....
%let stratum8=SW    ;
%let n=8;

%let pathname = c:\projects\yc_chart_template.xls;

%Macro Getdata(dsname);
  %do p = 1 %to &n;
proc export data= &dsname(where=(stratum="&&stratum&p"))
  outfile= "&pathname"
  dbms=excel2000 replace; sheet="&&stratum&p";
run;
%end;

%Mend;
%Getdata(gy_yc_data)

```

SAS Macro in Statistical Test - Example 6

- Purpose: test 15 response variables
- Results: 15 test outputs
 - ✓ HT_DEC_RATE
 - ✓ HT_CON_RATE
 -
 - ✓ DBH_CON_RATE

```

%let y1=HT_DEC_RATE;
%let y2=HT_CON_RATE;

.....
%let y15=DBH_CON_RATE;
%let n=15;

%Macro MakeTests;
  %do n=1 %to &n;
proc glm data=plot;
title "test variable: &&y&n";
  by aspen_stand ;
  class ecoregion;
model &&y&n = ecoregion bh_age_ave / solution;
lsmeans ecoregion / stderr pdiff cov out=adjmeans;
run;
  %end;
%Mend MakeTests;
%MakeTests

```

SAS Macro in Making Graphs - Example 7

- Purpose: create 5 graphs for 5 response variables
- Results: 5 graphs
 - ✓ TOPHT vs. Age
 - ✓ AVE_HT_CON vs. Age
 -
 - ✓ AVE_HT_DEC vs. Age

```
axis1 label=(a=90 r=0 f=swissl h=0.5 cm 'height (m)')  
    order=(0 to 40 by 10)  
    value=(f=swissl h=0.5 cm);  
axis2 label=(f=swissl h=0.5 cm 'age (years)')  
    order=(0 to 200 by 20 ) value=(f=swissl h=0.5 cm);
```

```
%let Height1=TOPHT;  
%let Height2=AVE_HT_CON;  
.....  
%let Height5=AVE_HT_DEC;  
  
%let n=5;
```

```

%Macro MakeGraphs;
  %do n=1 %to &n;
title j=c f=swiss1 h=2 "&&Height&n vs. Age";

proc gplot data=plot1 ;
  plot &&Height&n*Bh_age=x
    /overlay grid vaxis=axis1 haxis=axis2 nolegend
    autohref autovref frame lvref=2 lhref=2;
symbol1 v=dot c=green h=0.2 cm i=join r=3000;
run;
  %end;

%Mend MakeGraphs;
%MakeGraphs

```

Questions?

Thanks!

Tel. (780) 452-5878

yanguo_qin@forcorp.com