Building Macros and Tracking Their Use

Richard Koopmann Jr., Capella University

Business Requirements

Starting with SASHELP.SHOES, calculate a new variable defined as

```
(Sales - Returns) / Stores
```

Identify the top & bottom three products per region (there are ten regions) based on new variable.

Save the results to individual data sets and generate combined listing output.

Hard-Coded Approach

Subset statements are hard-coded into code.

Develop one complete, working ‘branch’.

Duplicate and adjust for each remaining region.

```
*** Africa ***;

title1 'Region: Africa';
proc summary data=sashelp.shoes nway;
    where region = 'Africa';
    class region product;
    var stores sales returns;
    output out=shoes_Africa(drop=_:)
        n(stores)=
        sum(sales returns)=;
run;
data shoes_Africa;
    set shoes_Africa;
    format slrps dollar12.;
    label slrps ='Sales (Less Returns) Per Store';
    slrps = (sales - returns) / stores;
run;

title2 'Top/Bottom 3 Products';
proc sort data=shoes_Africa;
    by descending slrps;
run;
data shoes_Africa_top;
    set shoes_Africa(obs=3);
run;
...
```

```
proc sort data=shoes_Africa;
    by slrps;
run;
data shoes_Africa_bottom;
    set shoes_Africa(obs=3);
run;
proc sort data=shoes_Africa_bottom;
    by descending slrps;
run;

data Africa;
    set shoes_Africa_top(rename=(product=Top_Product
slrps=Top_slrps));
    set
shoes_Africa_bottom(rename=(product=Bottom_Product
slrps=Bottom_slrps));
    keep top_: bottom_:;
run;
proc print data=Africa label;
    var top_: bottom_:;
run;

proc datasets library=work nodetails nolist;
    delete shoes_Africa:;
    run;
quit;
```

Nearly 500 lines

Convert to Parameterized Code

Isolate parameters.

Add parameter assignment statement near top.

Replace hard-coded parameter values with macro variables.

```
*** Central America/Caribbean ***;
%let Region=Central America/Caribbean;
data _null_;
    format dsn $19.;
    dsn = compress("&REGION", , 'ka');
    call symput('regiondsn', trim(dsn));
run;

title1 "Region: &REGION";
proc summary data=sashelp.shoes nway;
    where region = "&REGION";
    class region product;
    var stores sales returns;
    output out=shoes_&REGIONDSN(drop=_:)
        n(stores)=
        sum(sales returns)=;
run;
data shoes_&REGIONDSN;
    set shoes_&REGIONDSN;
    format slrps dollar12.;
    label slrps ='Sales (Less Returns) Per Store';
    slrps = (sales - returns) / stores;
run;

title2 'Top/Bottom 3 Products';
proc sort data=shoes_&REGIONDSN;
    by descending slrps;
run;
...
```

```
data shoes_&REGIONDSN._top;
    set shoes_&REGIONDSN(obs=3);
run;
proc sort data=shoes_&REGIONDSN;
    by slrps;
run;
data shoes_&REGIONDSN._bottom;
    set shoes_&REGIONDSN(obs=3);
run;
proc sort data=shoes_&REGIONDSN._bottom;
    by descending slrps;
run;

data &REGIONDSN;
    set shoes_&REGIONDSN._top(rename=(product=Top_Product
slrps=Top_slrps));
    set
    shoes_&REGIONDSN._bottom(rename=(product=Bottom_Product
slrps=Bottom_slrps));
    keep top_: bottom_:;
run;
proc print data=&REGIONDSN label;
    var top_: bottom_:;
run;

proc datasets library=work nodetails nolist;
    delete shoes_&REGIONDSN:;
    run;
quit;
```

Over 550 lines

Convert to Parameterized Macro

Wrap parameterized code in macro definition.

Call macro for each Region.

or

Create control data set to call macro automatically.

```
%macro TopBottomShoes(Region);
*%let Region=Central America/Caribbean;
data _null_;
    format dsn $19.;
    dsn = compress("&REGION", , 'ka');
    call symput('regiondsn', trim(dsn));
run;

title1 "Region: &REGION";
proc summary data=sashelp.shoes nway;
    where region = "&REGION";
    class region product;
    var stores sales returns;
    output out=shoes_&REGIONDSN(drop=_:)
        n(stores)=
        sum(sales returns)=;
run;

...

proc datasets library=work nodetails nolist;
    delete shoes_&REGIONDSN:;
    run;
quit;
%mend;
```

```
/* explicit call for each region */;
ods listing file='TopBottomShoes by Region.txt';
%TopBottomShoes(Africa);
%TopBottomShoes(Asia);
%TopBottomShoes(Canada);
%TopBottomShoes(Central America/Caribbean);
%TopBottomShoes(Eastern Europe);
%TopBottomShoes(Middle East);
%TopBottomShoes(Pacific);
%TopBottomShoes(South America);
%TopBottomShoes(United States);
%TopBottomShoes(Western Europe);
%TopBottomShoes(South America);
%TopBottomShoes(United States);
ods listing;
```

```
/* generate control file for all regions */;
proc freq data=sashelp.shoes;
    table Region / noprint out=Regions;
run;
data _null_;
    set regions;
    calltext = cats('%', 'TopBottomShoes(', Region, ');');
    file 'TopBottomShoes by Region.sas';
    put calltext;
run;

ods listing file='TopBottomShoes by Region.txt';
%inc 'TopBottomShoes by Region.sas';
ods listing;
```

Business Requirements Change

Rather than Top & Bottom products based on one metric, provide Top products based on two metrics

```
(Sales - Returns) / Stores

(Inventory - Sales + Returns) / Stores
```

```
%macro TopSalesDebtShoes(Region);
data _null_;
   format dsn $19.;
   dsn = compress("&REGION", , 'ka');
   call symput('regiondsn', trim(dsn));
run;

title1 "Region: &REGION";
proc summary data=sashelp.shoes nway;
   where region = "&REGION";
   class region product;
   var stores sales returns inventory;
   output out=shoes_&REGIONDSN(drop=_:)
      n(stores)=
      sum(sales returns inventory)=;
run;
data shoes_&REGIONDSN;
   set shoes_&REGIONDSN;
   format slrps isrps dollar12.;
   label slrps='Sales (Less Returns) Per Store'
      isrps='Debt per Store';
   slrps = (sales - returns) / stores;
   isrps = (inventory - sales + returns) / stores;
run;

title2 'Top 3 Product Sales/Debt';
proc sort data=shoes_&REGIONDSN;
   by descending slrps;
run;
data shoes_&REGIONDSN._sales;
   set shoes_&REGIONDSN(obs=3);
run;
...
```

```
proc sort data=shoes_&REGIONDSN;
   by descending isrps;
run;
data shoes_&REGIONDSN._debt;
   set shoes_&REGIONDSN(obs=3);
run;
data &REGIONDSN;
   set shoes_&REGIONDSN._sales(rename=(product=Top_Product_Sales slrps=Top_Sales));
   set shoes_&REGIONDSN._debt(rename=(product=Top_Product_Debt isrps=Top_Debt));
   keep top_:;
run;
proc print data=&REGIONDSN label;
   var top_:;
run;

proc datasets library=work nodetails nolist;
   delete shoes_&REGIONDSN:;
   run;
quit;
%mend;

/* generate control file for all regions */;
proc freq data=sashelp.shoes;
   table Region / noprint out=Regions;
run;
data _null_;
   set regions;
   calltext = cats('%', 'TopSalesDebtShoes(', Region, ');');
   file 'Shoe Sales & Debt.sas';
   put calltext;
run;
ods listing file='Shoe Sales & Debt.txt';
%inc 'Shoe Sales & Debt.sas';
ods listing;
```

Track Macro Usage

The macro has been adopted by several other users.

Determine how frequently the macro is called in a given week and who the top users are.

Extend Macro Functionality

Add tracking code.

Able to track macro use and offer end users help.

```
%macro TopSalesDebtShoes(Region);
%local macro;
%let macro=&SYSMACRONAME;

%if %upcase("&REGION") eq "HELP" %then %do;
    %put;
    %put //// &SYSMACRONAME. information;
    %put Extracts top products for Sales and Debt for specified region from
SASHELP.SHOES.;
    %put;
    %put //// Positional Parameters (in this order):;
    %put 1. Region: Course section start date begin in date9 format. If null, first of
current month;
    %put;
    %put //// Optional Keyword Parameters (in any order):;
    %put -none-;
    %put;
    %put //// Notes:;
    %put -none-;
    %put;
    dm log 'show';
    %goto ByeBye;
%end;

*** use of this macro will be tracked ***;
%tracker(Macro, &MACRO);

...

%ByeBye:;
%mend;
```

```
%TopSalesDebtShoes(HELP);
```

```
2237  ods listing;
2238 %TopSalesDebtShoes(HELP);

//// TOPSALESDEBTSHOES information
Lists top products Sales and Debt for specified region from SASHELP.SHOES.


//// Positional Parameters (in this order):
1. Region: Which region to extract information for.


//// Optional Keyword Parameters (in any order):
-none-


//// Notes:
-none-
```

```
%macro TopSalesDebtShoes(Region);
%local macro;
%let macro=&SYSMACRONAME;

%if %upcase("&REGION") eq "HELP" %then %do;
    %put;
    %put //// &SYSMACRONAME. information;
    %put Extracts top products for Sales and Debt for specified region from
SASHELP.SHOES.;
    %put;
    %put //// Positional Parameters (in this order):;
    %put 1. Region: Course section start date begin in date9 format. If null, first of
current month;
    %put;
    %put //// Optional Keyword Parameters (in any order):;
    %put -none-;
    %put;
    %put //// Notes:;
    %put -none-;
    %put;
    dm log 'show';
    %goto ByeBye;
%end;

*** use of this macro will be tracked ***;
%tracker(Macro, &MACRO);

...

%ByeBye:;
%mend;
```

```
%macro tracker(
  action
, description
);
%local macro;
%let macro=&SYSMACRONAME;

%if %upcase(&ACTION) eq HELP %then %do;
  %put;
  %put // &SYSMACRONAME. information;
  %put Tracks actions and events for usage-tracking.;
  %put;
  %put // Positional Parameters (in this order):;
  %put 1. ACTION: Brief description of what kind of action was taken by user.;
  %put 2. DESCRIPTION: Description of this action, if needed.;
  %put;
  %put //Optional Keyword Parameters (in any order):;
  %put -none-;
  %put;
  dm log 'show';
  %goto ByeBye;
%end;

option nonotes;
proc sql;
insert into sasuser.tracker
  set sysuserid=upcase("&SYSUSERID")
, Action=upcase("&ACTION")
, Description=upcase("&DESCRIPTION")
, datetime=dhms(today(),0,0,time())
;
quit;
option notes;

%ByeBye:;

%mend;
```

```
proc means data=sys.tracker n;
    format description sysUserID $20.;
    where action eq 'MACRO'
    and description like 'TOP%SHOES'
    and datepart(datetime) ge "&SYSDATE9."d-7;
    class sysUserID Description;
    var datetime;
    types sysUserID Description;
run;
```

```
    Analysis Variable : DateTime

Description                N Obs         N
-----------------------------------------
TOPBOTTOMSHOES                 2         2

TOPSALESDEBTSHOES             10        10
-----------------------------------------


    Analysis Variable : DateTime

SysUserID                  N Obs         N
-----------------------------------------
RKOOPMANN                     12        12
-----------------------------------------
```

```
/* add this to autoexec.sas to transfer client tracker records to shared library */;

option nonotes;
libname sys 'path-to-shared-drive';


proc append base=sys.tracker data=sasuser.tracker;
run;

data sasuser.tracker;
   set sys.tracker;
   stop;
run;

libname sys clear;
option notes;
dm log 'clear;';
```

Contact Information

Richard Koopmann Jr.
Sr. Research Analyst
Capella University
225 S. 6th St, 9th Fl.
Minneapolis, MN 55402
rkoopmann@capella.edu
http://j.mp/sasmacrotracker