A Novel Task

- Using 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', then copy & paste, change subset values.

Advantages

- Explicit code is very easy to follow. This is often less intimidating for newer SAS analysts.
- Illusion of productivity. More effort goes in to generating 500 lines of code than 50 lines of code, but effort $\neq$ productivity.

Disadvantages

- Explicit code is very verbose. Easy to make consistency & syntactical errors.
- Changes must be made in many places. This example calculated a single new variable. What about adding 10 (or more) variables?
- Not scalable. This example was limited to 10 different global regions. What if this was by US state or county (or zip code)?
- Nearly impossible to track how often code is used and who uses it.

```
ods listing file='c:\temp\Shoes.txt';

*** 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;


*** Asia ***;

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

title2 'Top/Bottom 3 Products';
proc sort data=shoes_Asia;
     by descending slrps;
run;
data shoes_Asia_top;
     set shoes_Asia(obs=3);
run;
proc sort data=shoes_Asia;
     by slrps;
run;
data shoes_Asia_bottom;
     set shoes_Asia(obs=3);
run;
proc sort data=shoes_Asia_bottom;
     by descending slrps;
run;

data Asia;
     set shoes_Asia_top(rename=(product=Top_Product slrps=Top_slrps));
     set  shoes_Asia_bottom(rename=(product=Bottom_Product slrps=Bottom_slrps));
     keep top_: bottom_:;
run;
proc print data=Asia label;
     var top_: bottom_:;
run;

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


*** repeat for remaining 8 Regions (500 lines of code total) ***;

ods listing;
title;
```

Parameterize Code

Isolate parameters. Add parameter assignment statement near top, and replace hard-coded parameter values with macro variables.

Advantages

- Slightly abstracted code is still easy to follow. Though macro variables may be intimidating for newer SAS analysts.
- Only marginally better than Hard-Coded approach.

Disadvantages

- Code is still verbose, though now more abstracted. Still easy to make errors.
- Changes must be made in many places. This example calculated a single new variable. What about adding 10 (or more) variables?
- Not scalable. This example was limited to 10 different global regions. What if this was by US state or county (or zip code)?

```
ods listing file='c:\temp\Shoes.txt';

*** 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;



*** Eastern Europe ***;
%let Region=Eastern Europe;
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_Prod slrps=Top_slrps));
     set shoes_&REGIONDSN._bottom(rename=(product=Bottom_Prod 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;

*** repeat for remaining 8 Regions (550 lines of code total) ***;

ods listing;
title;
```

Convert to a Macro

Wrap parameterized code in macro definition and call macro for each Region.

Advantages

- Scalable to many different levels (parameter values).
- Virtually impossible for inconsistencies between calls.
- Less chance for syntactical errors.

Disadvantages

- Difficult to know how often code is being called and who is using it.
- Macros may be intimidating to newer SAS users.

```
%macro TopBottomShoes(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;
     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_Prod slrps=Top_slrps));
     set shoes_&REGIONDSN._bottom(rename=(product=Bottom_Prod 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;
%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;
title;
```

```
/* 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 file='TopBottomShoes by Region.txt';
title;
```

Changed Requirements

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

- `(Sales – Returns) / Stores`
- `(inventory - sales + returns) / stores`

Modify Macro Workflow

Implement requirement changes in code only once.

Advantages

- Easy to implement changes to repetitive tasks. Changing reporting from region to country would be fairly trivial.
- Easy to add functionality in a consistent, scalable way. Adding additional metrics for the Top lists would be easy.

Disadvantages

- Making edits to macro definition may cause unexpected "issues" to arise in existing working code.

```
%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_Prod_Sales slrps=Top_Sales));
     set shoes_&REGIONDSN._debt(rename=(product=Top_Prod_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 'TopSalesDebtShoes by Region.sas';
     put calltext;
run;
ods listing file='TopSalesDebtShoes by Region.txt';
%inc 'TopSalesDebtShoes by Region.sas';
ods listing;
title;
```

Track Macro Users and 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.

All code available for download from

http://j.mp/sasmacrotracker

Extend Macro Functionality
Able to track macro use and offer end users help.

Advantages
- Able to add help without requiring end users to open the source code and search throughout for comments.
- Able to track who executes the macro code and when. Able to demonstrate value of macro work.

Disadvantages
- ?

```
/* 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;';
```

```
/* add this inside macro definition to track the macro use */;

%local macro;
%let macro=&SYSMACRONAME;

%tracker(Macro, &MACRO);
```

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

%if %upcase("&REGION") eq "HELP" %then %do;
     %put;
     %put //// &SYSMACRONAME. information;
     %put List top products Sales and Debt for specified region from SASHELP.SHOES.;
     %put;
     %put //// Positional Parameters (in this order):;
     %put 1. Region: Which region to extract information for.;
     %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);

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_Prod_Sales slrps=Top_Sales));
     set  shoes_&REGIONDSN._debt(rename=(product=Top_Prod_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;
%ByeBye:;
%mend;

%TopSalesDebtShoes(HELP);

/* 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;
title;

proc means data=sasuser.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;
```

```
%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;
```

```
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-
```

```
%TopSalesDebtShoes(Africa);
%TopSalesDebtShoes(Asia);
%TopSalesDebtShoes(Canada);
%TopSalesDebtShoes(Central America/Caribbean);
%TopSalesDebtShoes(Eastern Europe);
%TopSalesDebtShoes(Middle East);
%TopSalesDebtShoes(Pacific);
%TopSalesDebtShoes(South America);
%TopSalesDebtShoes(United States);
%TopSalesDebtShoes(Western Europe);
```

```
Region: Africa
Top 3 Product Sales/Debt

                                Sales (Less
                                   Returns)                                Debt per
Obs    Top_Prod_Sales           Per Store    Top_Prod_Debt                 Store

 1     Men's Casual              $109,337    Men's Casual                $180,335
 2     Women's Casual            $101,417    Women's Casual              $171,479
 3     Women's Dress              $45,462    Slipper                     $112,240



Region: Asia
Top 3 Product Sales/Debt

                               Sales (Less
      Top_Prod_                   Returns)                               Debt per
Obs   Sales                     Per Store    Top_Prod_Debt                 Store

 1    Women's Dress              $76,343    Slipper                     $168,059
 2    Slipper                    $74,482    Men's Dress                  $77,882
 3    Men's Dress                $58,436    Women's Dress                $64,285


...
```

```
 Analysis Variable : DateTime

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

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


    Analysis Variable : DateTime

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