



# Quick Start to the SAS Macro Language

Allison Saito



# SAS Macro Language = Fancy Find-and-Replace

```
title "Revenue from [redacted]s";  
proc means data=sashelp.snacks;  
  where upcase(Product) contains "[redacted]";  
run;
```



```
title "Revenue from Puffs";  
proc means data=sashelp.snacks;  
  where upcase(Product) contains "PUFF";  
run;
```

# Why 'fancy' find-and-replace?

- Replace text with ....
  - Given text
  - Text a function has been applied to
  - A value that was calculated by SAS
- Also ...
  - IF/THEN/ELSE to control what text is input
  - Loops

# Demo : Transform a static program into a macro

# Code that was presented , provided for reference

- The following is based on the demo. It shows how a SAS program can be transformed into a macro program.
- Note on copy/pasting from these slides: The copy seems to grab characters in a nonstandard way. Before pasting to a program editor, paste 'text only' into a Word document, then copy/paste the result of the 'text only' paste into the program editor

# Starter Code

Summarize and subset data, then graph the result

```
proc sql;  
create table work.snacks as  
  select put(date, yymmd.) as Month  
        , product  
        , sum(qtySold*price) as Revenue  
from sashelp.snacks  
where upcase(product) contains "PRETZEL"  
group by Month  
        , product  
;  
quit;  
  
title "Revenue from Pretzels";  
proc sgplot data=work.snacks;  
  series x=Month Y=Revenue / group=Product;  
run;
```

# Generalize

Make flexible with macro variable &variety

```
%let variety = Pretzel;
```

```
proc sql;
create table work.snacks as
  select put(date, yymmd.) as Month
        , product
        , sum(qtySold*price) as Revenue
  from sashelp.snacks
  where upcase(product) contains "&variety"
  group by Month
        , product
;
quit;

title "Revenue from &variety.s";
proc sgplot data=work.snacks;
  series x=Month Y=Revenue / group=Product;
run;
```



# Generalize

## Standardize capitalization of macro variable's value

```
%let variety = Pretzel;

proc sql;
create table work.snacks as
  select put(date, yymmd.) as Month
        , product
        , sum(qtySold*price) as Revenue
  from sashelp.snacks
  where upcase(product) contains "%sysFunc(upcase(&variety))"
  group by Month
        , product
;
quit;

title "Revenue from %sysFunc(propcase(&variety))s";
proc sgplot data=work.snacks;
  series x=Month Y=Revenue / group=Product;
run;
```

# Put into Macro Program

```
%let variety = Pretzel;
```

```
%macro snackGraph(variety);  
proc sql;  
create table work.snacks as  
  select put(date, yymmd.) as Month  
        , product  
        , sum(qtySold*price) as Revenue  
from sashelp.snacks  
where upcase(product) contains "%sysFunc(upcase(&variety))"  
group by Month  
        , product  
;  
quit;  
  
title "Revenue from %sysFunc(propcase(&variety))s";  
proc sgplot data=work.snacks;  
  series x=Month Y=Revenue / group=Product;  
run;  
%mEnd snackGraph;
```

# Check number of rows before graphing

```
%macro snackGraph(variety);  
proc sql;  
...  
quit;  
  
%if &sqlObs = 0 %then %do;  
  %put WARNING: No rows have the value &variety in their Product value.;  
%end;  
%else %do;  
  title "Revenue from %sysFunc(propcase(&variety))s";  
  proc sgplot data=work.snacks;  
    series x=Month Y=Revenue / group=Product;  
  run;  
%end;  
%mEnd snackGraph;
```

# Full code, including comments

```
/* %snackGraph
*   Purpose: Summarize data to revenue per month, and subset to only
*             products of a given variety
*   Input: variety parameter (single word, used to subset products)
*   Output: work.snack and a line graph of revenue , both summarized and subset
*/
%macro snackGraph(variety);
  %macro a; %mEnd a; /* restores color coding in Enterprise Guide */

  proc sql;
  create table work.snacks as
  select put(date, yymmd.) as Month /* PUT function formats a value, then returns the formatted value as a string */
         , product
         , sum(qtySold*price) as Revenue
  from sashelp.snacks
  where upcase(product) contains "%sysFunc(upcase(&variety))" /* %sysFunc allows macro to use a standard function */
  group by Month
         , product      ;
quit;

  /* If zero rows in WORK.SNACKS, put warning in log.  Otherwise, graph it.  ;
  %if &sqlObs = 0 %then %do;
    %put WARNING: No rows have the value &variety in their Product value.; /* puts warning into log */
  %end;
  %else %do;
    title "Revenue from %sysFunc(propcase(&variety))s"; /* %sysFunc allows macro to use a standard function */
    proc sgplot data=work.snacks;
      series x=Month Y=Revenue / group=Product;
    run;
  %end;
%mEnd snackGraph;
```

# Stay in touch



SAS QuickStart  
YouTube Playlist



Allison's  
LinkedIn

**sas innovate**  
2025