10 Quick Tips for Getting Tipsy with SAS



Richann

IS AN INDEPENDENT STATISTICAL PROGRAMMER AND CDISC CONSULTANT BASED IN OHIO WHO LOVES TO CODE AND IS VERY ACTIVE IN THE SAS USER GROUP COMMUNITY. WHEN RICHANN IS NOT BUSY CODING OR WRITING PUBLICATIONS IN THE SAS USER GROUP COMMUNITY, SHE IS SPENDING TIME WITH HER FAMILY AND CUTE BUT PSYCHO PUPPY, LOKI, OR DOING SOME OF HER FAVORITE CRAFTS SUCH AS CROCHETING OR SEWING.



Lisa

IS RESIDES IN OKINAWA, JAPAN AND HAS BEEN ACTIVE IN THE SAS USER COMMUNITY FOR MANY YEARS. SHE HAS MORE THAN 25 YEARS OF EXPERIENCE IN SAS PROGRAMMING, DATA SCIENCE, AND ANALYTICS. WHEN LISA ISN'T PRESENTING OR VOLUNTEERING, SHE ENJOYS SPENDING TIME WITH HER FAMILY, PHOTOGRAPHY, AND SCRAPBOOKING. LISA IS ALSO THE PHOTOGRAPHER FOR PHARMASUG JAPAN.

For reference only
Not for dissemination

Do not post on any shared sites or websites

Do not forward slides to others without express permission from the authors

Do not repurpose slides

Never Have I Ever

Been Mis"Quote"d







Adding Quotes Around Each Token in a String

- Determine how many tokens with COUNTW function
- Loop through each item adding quotations using QUOTE function
- Use CALL SYMPUTX subroutine to save temporary variable to a macro variable

```
%let fmtname = SUPPDMQNAM SUPPCMQNAM;

data _null_;
  fmtname = "&fmtname";
  length newfmtname $500;
  do i = 1 to countw(fmtname, ' ');
     newfmtname = catx(' ', newfmtname, quote(scan(fmtname, i)));
  end;
  call symputx('newfmtname', newfmtname);
run;
```

FMTNAME	NEWFMTNAME
SUPPDMQNAM SUPPCMQNAM	"SUPPDMQNAM" "SUPPCMQNAM"



Never Have I Ever

Been Turned Around





Column Headers with PROC TRANSPOSE



- Use data to determine column headers
- Create a variable that contains values that are used as labels

TRTPN	TRTP	PCNT	TRTPLBL	AVAL
1	Drug A	90	Drug A (N=90)	86
2	Drug A + Comp C	90	Drug A + Comp C (N=90)	75
3	Drug B	150	Drug B (N=150)	137
4	Drug B + Comp D	90	Drug B + Comp D (N=90)	.
5	Drug A Overall	180	Drug A Overall (N=180)	161
6	Drug B Overall	240	Drug B Overall (N=240)	137

```
proc transpose data = trt    out = t_trt (drop = _:)
              prefix = TRT_;
  var AVAL;
  id TRTPN;
  idlabel TRTPLBL;
run;
```

TRT_1	TRT_2	TRT_3	TRT_4	TRT_5	TRT_6
86	75	137	.	161	137





Column Headers with PROC TRANSPOSE



Name	Type	Length	Format	Informat	TRTPLBL
TRT_1	Numeric	8			Drug A`(N=90)
TRT_2	Numeric	8			Drug A + Comp C`(N=90)
TRT_3	Numeric	8			Drug B`(N=150)
TRT_4	Numeric	8			Drug B + Comp D`(N=90)
TRT_5	Numeric	8			Drug A Overall`(N=180)
TRT_6	Numeric	8			Drug B Overall`(N=240)

```
proc report data = t_trt split = "`"
        style(column) = [just = c cellwidth = 1.35in];
  columns _all_;
run;
```

Drug A (N=90)	Drug A + Comp C (N=90)	Drug B (N=150)	Drug B + Comp D (N=90)	Drug A Overall (N=180)	Drug B Overall (N=240)
86	75	137	.	161	137

Never Have I Ever

"Miss"ed Somthing





MISSING VALUES DEFAULT TO ZERO

- There is a column with a missing value
- For a static set of variables, this could be easily defaulted to zero with IF-THEN-ELSE, but dynamic approach is better
- Utilizing the ARRAY statement, we can loop through each variable in the transposed data set
- The COALESCEC function is used to return the first non-missing value in the list specified
- If it is null, then the 0 is returned

TRT_1	TRT_2	TRT_3	TRT_4	TRT_5	TRT_6
86	75	137	.	161	137

```
data t_trt2 (drop = i);
  set t_trt;
  array TRT_{&num_trt} 3;
  do i = 1 to dim(TRT_);
    TRT_{i} = coalesce(TRT_{i}, 0);
  end;
run;
```

TRT_1	TRT_2	TRT_3	TRT_4	TRT_5	TRT_6
86	75	137	0	161	137

Never Have I Ever

Been Shorthanded





Shorthand for Dealing with Consecutive Variables and Data Sets with Similar Names





- Need to pool several data sets with similar naming convention
- Have to identify the source of the input data set in the master data set
- Need to allow for more data sets to be added

pets_lisa.sas7bdat

pets_louise.sas7bdat

pets_richann.sas7bdat

```
data allpetsorig;
  set TIPSY.PETS_LISA (in = lisa)
      TIPSY.PETS_LOUISE (in = louise)
      TIPSY.PETS_RICHANN (in = richann);
  length OWNER $20;
  if lisa then OWNER = 'Lisa';
  else if louise then OWNER = 'Louise';
  else if richann then OWNER = 'Richann';
run;
```

```
data allpets;
  set TIPSY.PETS_:
      indsname = source;
  length OWNER $20;
  OWNER = propcase(scan(source, -1, '_'));
run;
```

```
NOTE: There were 5 observations read from the data set TIPSY.PETS_LISA.
NOTE: There were 6 observations read from the data set TIPSY.PETS_LOUISE.
NOTE: There were 5 observations read from the data set TIPSY.PETS_RICHANN.
```

Never Have I Ever

Had "Abs"olut
(IYKYK)





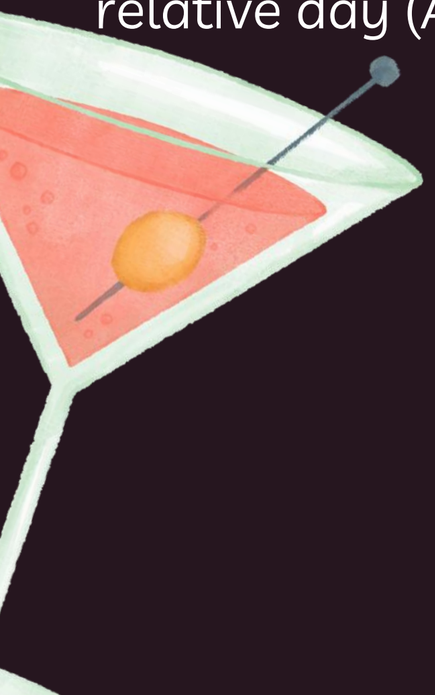




ABSOLUTE DIFFERENCE BETWEEN STUDY DAY AND TARGET DAY



- Within CDISC ADaM standards variables ending in DY represent the relative days and therefore there is no Day 0
- AWTDIFF represents the absolute difference between the target day (AWTARGET) and the relative day (ADY)



```
data dt_dy;
  set dts;
  if nmiss(ADT, TRTSDT) = 0 then ADY = ADT - TRTSDT + (ADT >= TRTSDT);
  if nmiss(ADY, AWTARGET) = 0 then
    AWTDIFF = abs(ADY - AWTARGET) - ((ADY*AWTARGET) < 0);
run;
```

AWTARGET	ADY	AWTDIFF	AWTDIFF*
1	-2	2	3
1	1	0	0
42	44	2	2

* Not adjusted for no Day 0

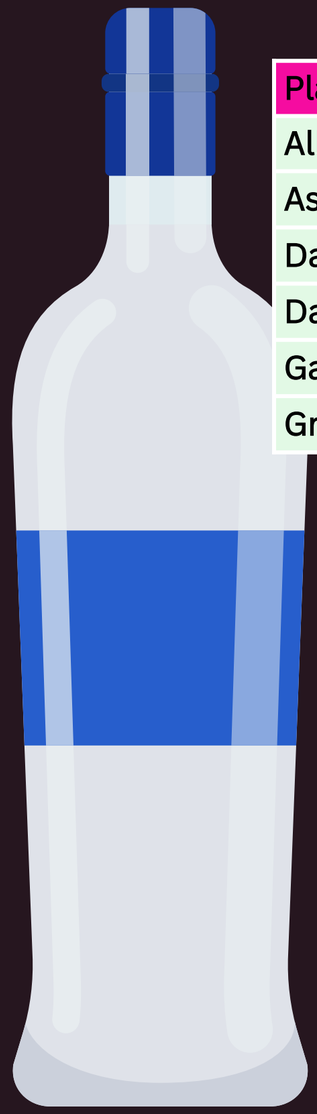
Never Have I Ever

Been in a SeQueL





Macro Variable List Using PROC SQL



Player's Name	Team at the End of 1986	Position(s) in 1986
Allanson, Andy	Cleveland	C
Ashby, Alan	Houston	C
Davis, Alan	Seattle	1B
Dawson, Andre	Montreal	RF
Galarraga, Andres	Montreal	1B
Griffin, Alfredo	Oakland	SS

```
proc sql noprint;
        select distinct Position
        into :Position_List separated by '*'
        from sashelp.baseball;
        %let n_vars = &SQLOBS;
quit;

%put &=n_vars &=position_list ;  /* print variable list in the log */
```



(log output)

```
N_VARS=25
POSTION_LIST=13*1B*1O*23*2B*2S*32*3B*3O*3S*C*CD*CF*CS*DH*DO*LF*O1*OD*OF*OS*RF*S3*SS*UT
```

Never Have I Ever

"Multi"plied





USING THE XLSX ENGINE TO READ IN MULTIPLE EXCEL WORKSHEETS

- SAS XLSX engine allows you to read and write Microsoft Excel files as if they were data sets in a library
- In the libname statement, you point to where the Excel workbook is located
- Creates structure (library and data set) only, need to populate data

```
Options validvarname=any;
```



Never Have I Ever

Failed to "Exist"







CHECK FOR FILE EXISTENCE OR SUCCESSFUL ASSIGNMENT OF A LIBNAME

- FILEEXIST:
  - checks for the existence of a file
  - Returns a value of 1 if found and a value of 0 if not found
- LIBREF:
  - Checks for the successful assignment of a LIBNAME
  - Returns a value of 0 if successful and a non-zero if not successful
- Both functions are invoked using %SYSFUNC

```
%sysfunc(libref(&libfile))
%sysfunc(fileexist(&libfile))
```



CHECK FOR FILE EXISTENCE OR SUCCESSFUL ASSIGNMENT OF A LIBNAME



```
42      %lfexist(libfile = C:\Users\gonza\Desktop\Conferences\Drafts\Tipsy\Pet Data.xlsx, lfcheck = file)
C:\Users\gonza\Desktop\Conferences\Drafts\Tipsy\Pet Data.xlsx WAS FOUND!
ADDITIONAL SAS CODE WOULD BE ENTERED HERE TO CONTINUE PROCESSING
43      %lfexist(libfile = C:\Users\gonza\Desktop\Conferences\Drafts\Tipsy\Pet_Data.xlsx, lfcheck = file)
ERROR: C:\Users\gonza\Desktop\Conferences\Drafts\Tipsy\Pet_Data.xlsx WAS NOT FOUND!!!
ERROR: Execution terminated by an %ABORT statement.
44      %lfexist(libfile = tipsy)
tipsy WAS FOUND!
ADDITIONAL SAS CODE WOULD BE ENTERED HERE TO CONTINUE PROCESSING
45      %lfexist(libfile = tips2)
ERROR: tips2 WAS NOT FOUND!!!
ERROR: Execution terminated by an %ABORT statement.
```

SAS Log



Never Have I Ever

Been "DIVIDE"d





Dividing Numbers Using the DIVIDE Function

```
data div;
  set num_den;
  DIV = NUM / DEN;
run;
```

```
data div;
  set num_den;
  DIV = divide(NUM, DEN);
run;
```

NOTE: Division by zero detected at line 31 column 14.
NUM=39 DEN=0 DIV=. _ERROR_=1 _N_=3
NOTE: Mathematical operations could not be performed at the following places. The results of the operations have been set to missing values.

NUM	DEN	DIV	DIV*
34	3	11.33333	11.33333
125	10	12.5	12.5
39	0	.	.
0	-3	0	0
-44	-4	11	11

* Uses the DIVIDE function



Never Have I Ever

Been "Anebriated"...

I mean Abbreviated

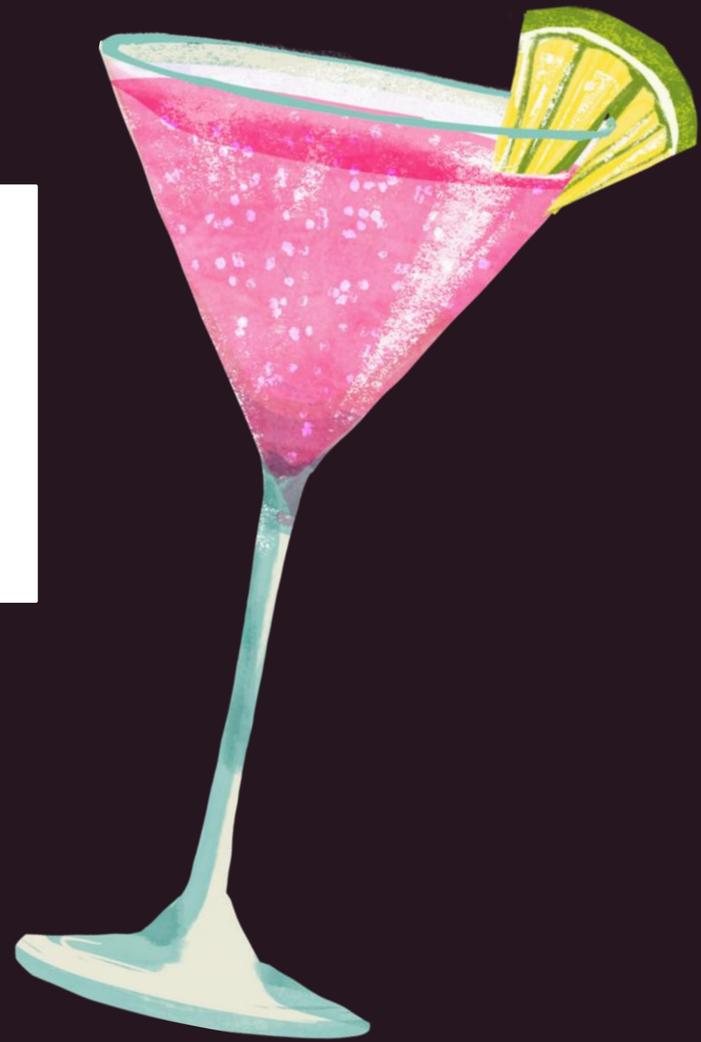






Abbreviations For Frequently Used SAS Code

```
ods output chisq = chi_oall;
ods output fishersexact = fis_oall;
proc freq data = INDSN noprint order = data;
     tables VAR1 * VAR2 / out = OUTDSN
                          chisq cmh fisher expected
                          plots = freqplot (twoway = stacked);
run;
```



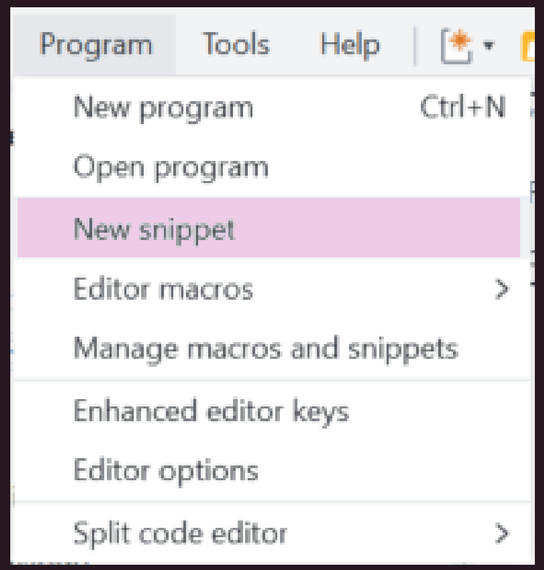


Abbreviations For Frequently Used SAS Code

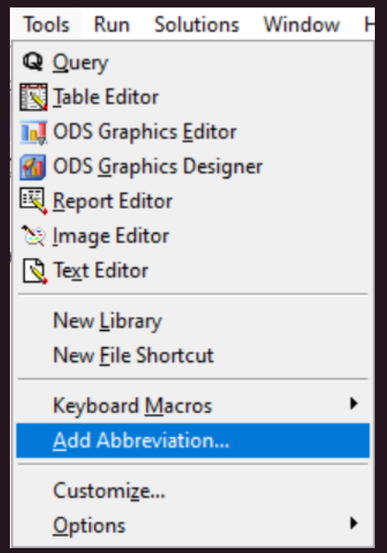


STEP 1

SAS Enterprise Guide



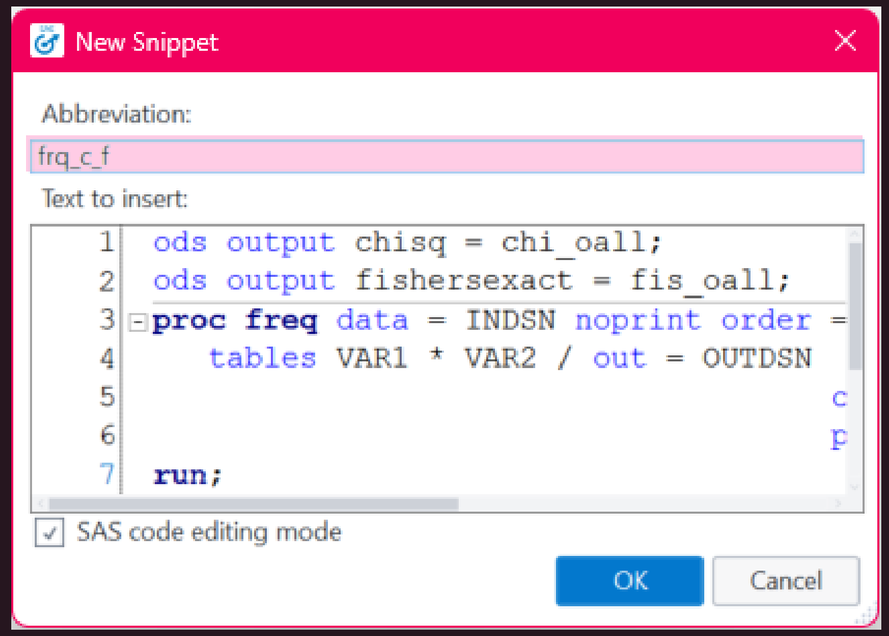
SAS Display Manager





Abbreviations For Frequently Used SAS Code

STEP 2



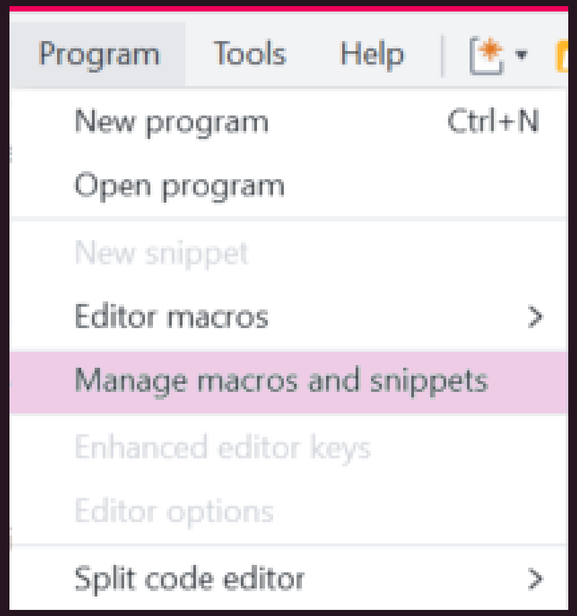




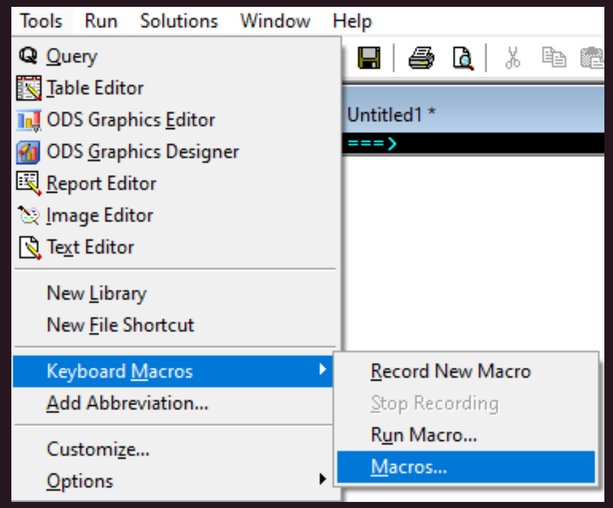
Abbreviations For Frequently Used SAS Code

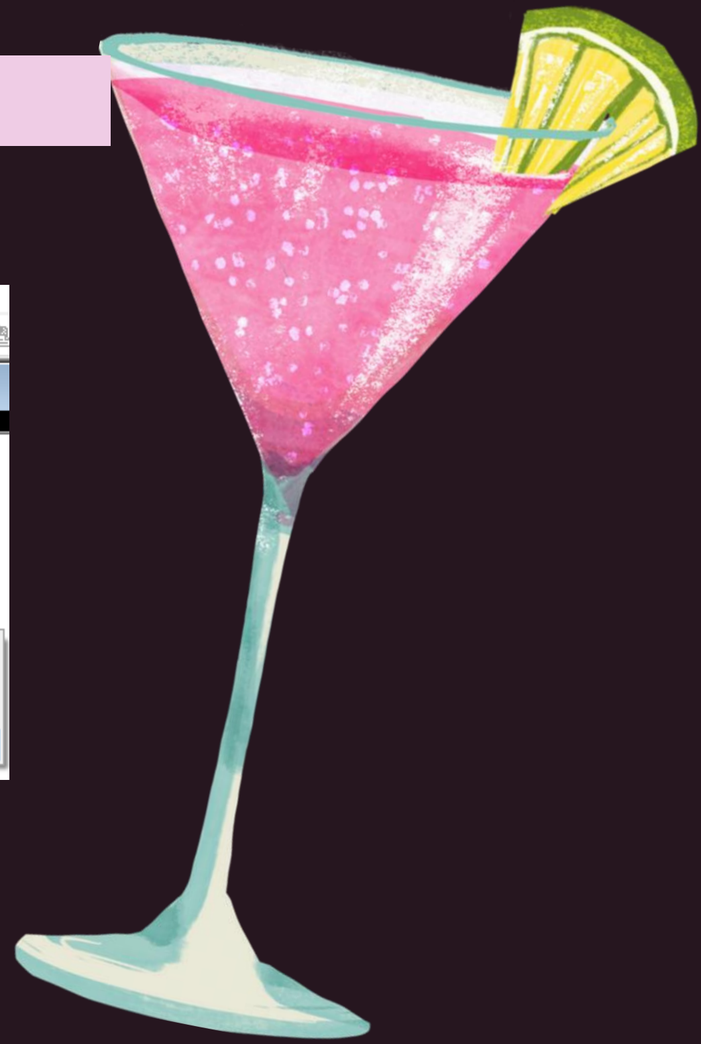
STEP 3

SAS Enterprise Guide



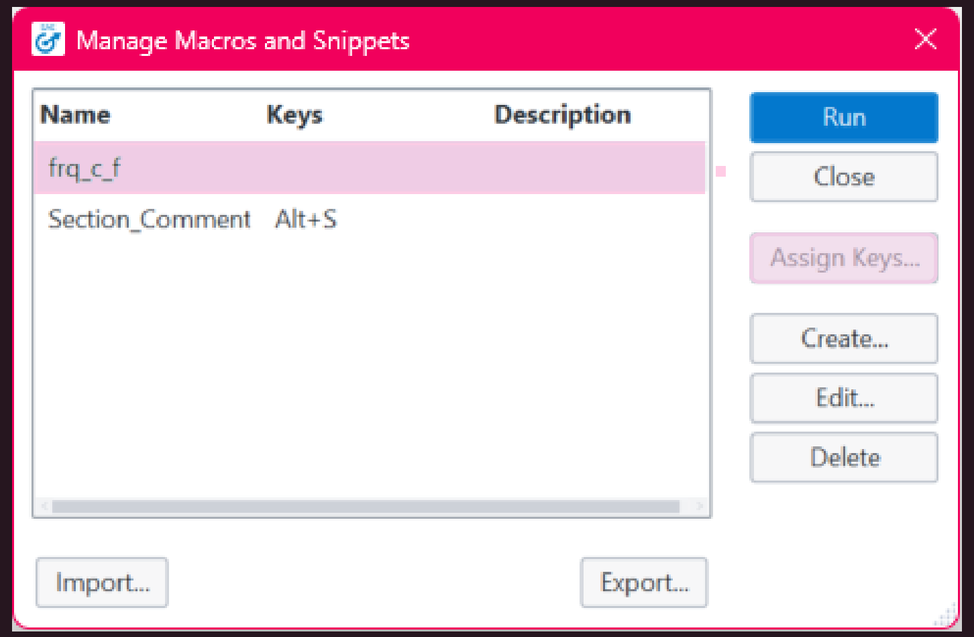
SAS Display Manager





Abbreviations For Frequently Used SAS Code

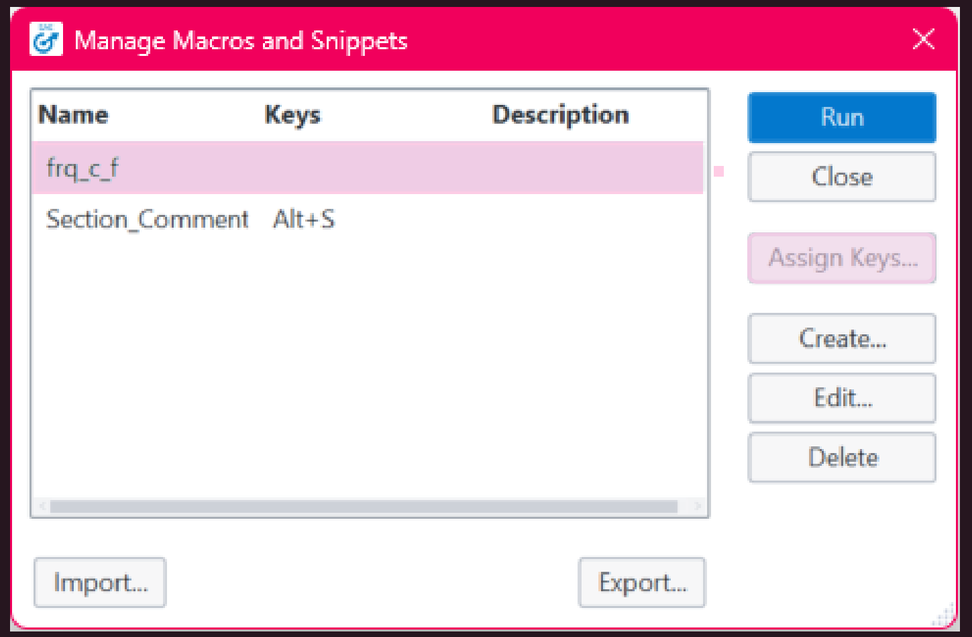
STEP 4





Abbreviations For Frequently Used SAS Code

STEP 5







Abbreviations For Frequently Used SAS Code

STEP 6

Press new shortcut key:

Ctrl+Shift+F

Currently assigned to:

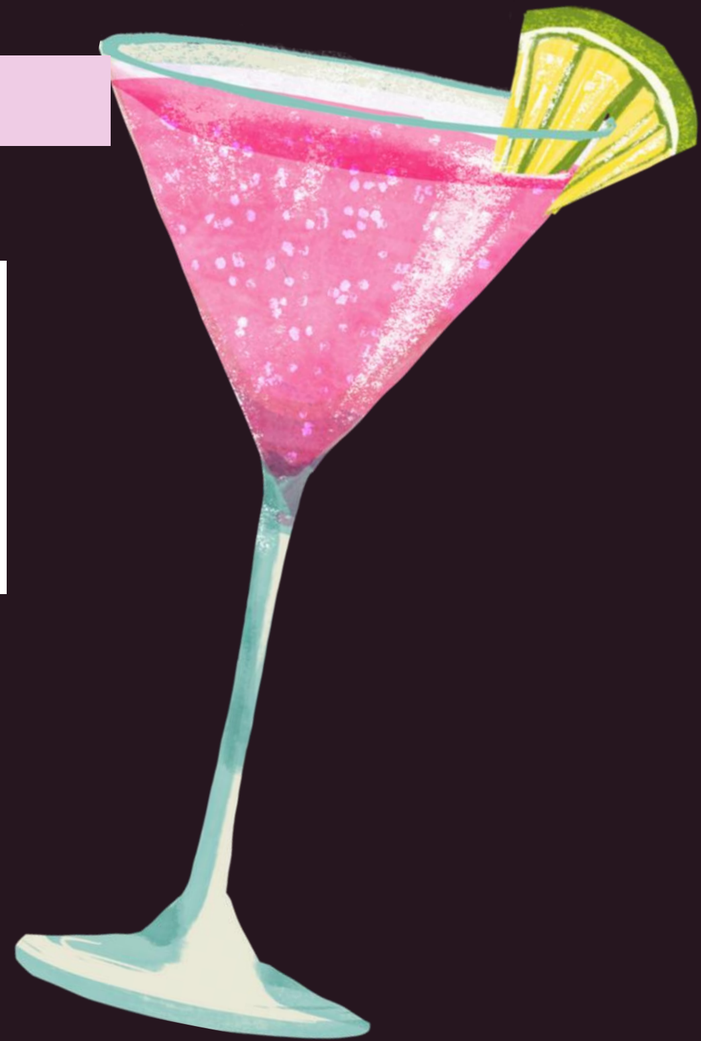
Autocomplete format

Press new shortcut key:

Alt+Shift+F

Currently assigned to:

None





Abbreviations For Frequently Used SAS Code

```
frq_c_f
ods output chisq = chi_oall; od...(Abbrev)
```

```
ods output chisq = chi_oall;
ods output fishersexact = fis_oall;
proc freq data = INDSN noprint order = data;
   tables VAR1 * VAR2 / out = OUTDSN
                                chisq cmh fisher expected
                                plots = freqplot (twoway = stacked);
run;
```

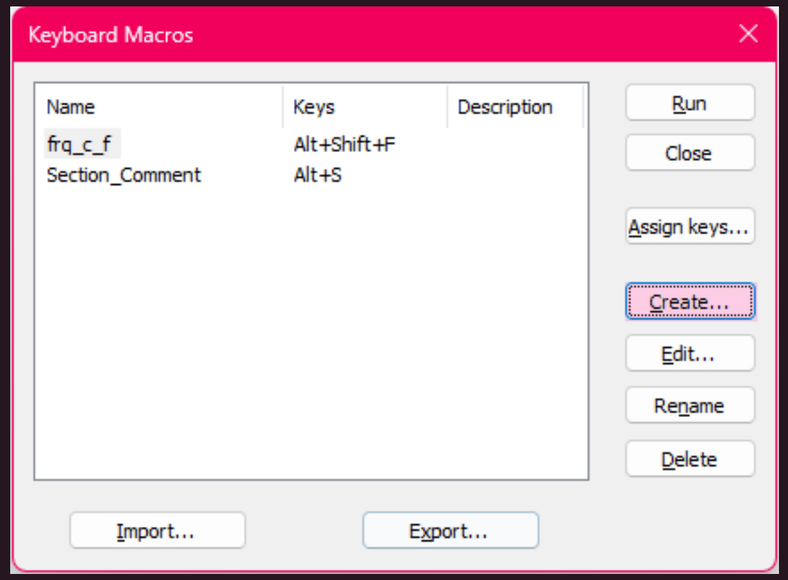




Keyboard Macros For Frequently Used SAS Code



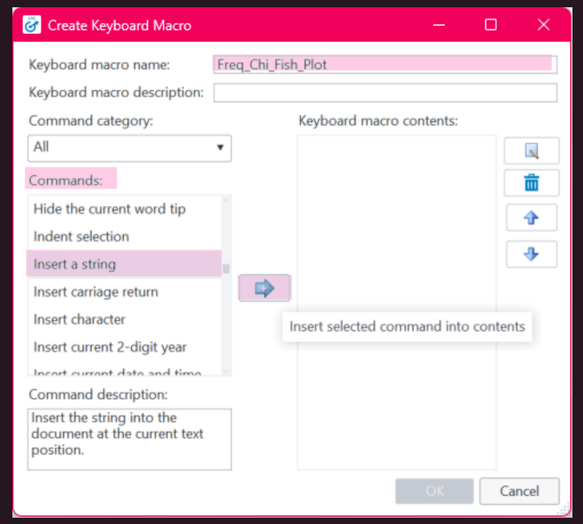
STEP 1

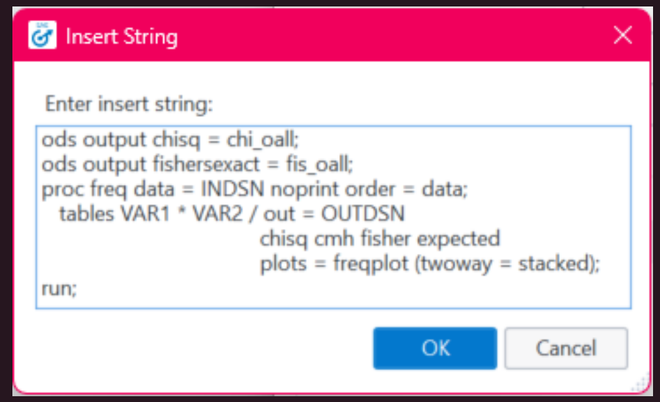


Keyboard Macros For Frequently Used SAS Code



STEP 2



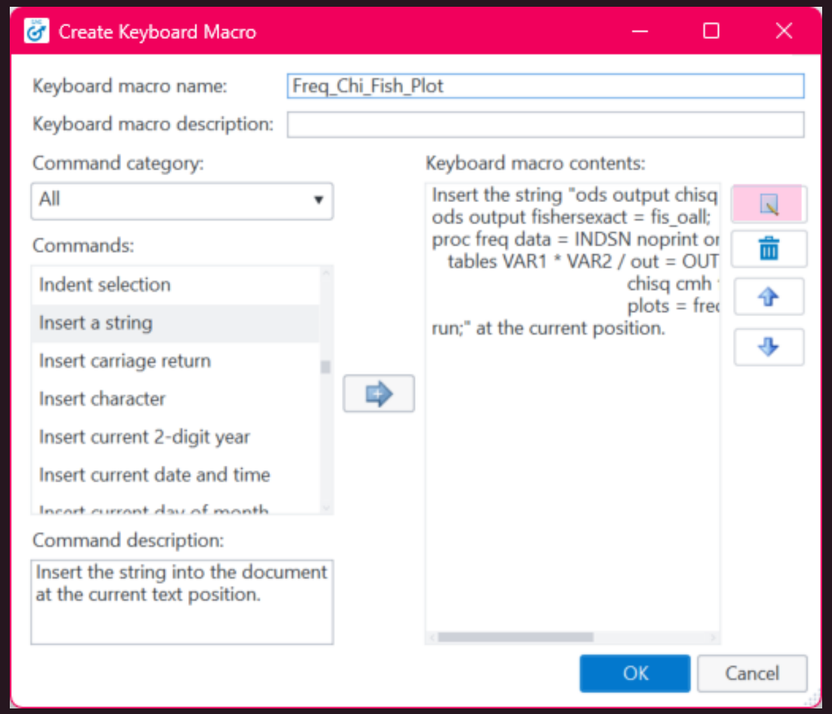






Keyboard Macros For Frequently Used SAS Code

STEP 3





MORE TASTY LIBATIONS IN THE BAR (OR PAPER)

- Column Headers with PROC TRANSPOSE
- Shorthand for Dealing with Consecutive variables and Data Sets with Similar Names
- Macro Variable List Using PROC SQL
- Dividing Numbers Using the DIV Function
- Abbreviations For Frequently Used SAS Code



Contact Information

RICHANN JEAN WATSON
DataRich Consulting
richann.watson@datarichconsulting.com

LISA MENDEZ
sasebmendez@gmail.com

