

To Macro or Not To Macro?

Mike Atkinson and Steve Gibbs

To Macro...

What is a Macro?

- ▶ Essentially a program which can be activated by a 'call' within the same program or from a different program.
- ▶ Another defining characteristic is that it can have macro variables which can be specified in the macro call.

- ▶ Layout:
- ▶ `%macro msuave1(mvar1 =,mvar2 =,mvar3);`
- ▶ Code ...
- ▶ `%mend msuave1;`

- ▶ To call this macro:
- ▶ `%msuave1(mvar1 =age,mvar2 =gender,mvar3 =group);`

When can macros be most useful?

- ▶ When you have to run a program multiple times with slight differences.
- ▶ When you have a number of colleagues needing to run comparable analyses and interpretation might be a problem (e.g. working on different protocols for a study drug utilising programming staff around the globe).
- ▶ When you have a repetitive task to do within a program (e.g. upcasing variable names; output to e.g. .csv files).

Running a program multiple times with slight differences:

- ▶ Using Clinical Trials in the Pharmaceutical Industry as an example, where reporting Safety Laboratory (lab) data (descriptive stats).
- ▶ One lab test may need to be reported to a different number of decimal places than another.

Laboratory Data Report

Lab Test		Treatment A	Placebo
Glucose	n	49	51
	Mean (SD)	96.4 (11.06)	97.1 (10.54)
	Median	96.1	98.3
	Min, Max	78, 113	79, 115
Potassium	n	49	51
	Mean (SD)	4.59 (0.617)	4.63 (0.598)
	Median	4.53	4.61
	Min, Max	3.6, 5.7	3.6, 5.7
Total Bilirubin	n	49	51
	Mean (SD)	0.944 (0.4026)	1.016 (0.3998)
	Median	0.885	1.046
	Min, Max	0.35, 1.60	0.31, 1.59

Case 1

- ▶ Descriptive statistics:
 - ▶ Mean, Standard Deviation, Median, Min & Max need to be found for 3 lab tests.
 - ▶ These are BILIRUBIN, GLUCOSE and POTASSIUM
 - ▶ Each needs to be reported to a different number of decimal places though...

Macros can also be useful for standardising coding practices.

- ▶ An example a global pharmaceutical company might like to have PK/ECG data analysed the same way between users working on related projects.
- ▶ In this way a macro-based approach can help to minimise user differences.

- ▶ Similarly macros can help to minimise within user differences for similar coding tasks over time.
- ▶ When you have a repetitive task across a number of different programs.
- ▶ e.g....

Case 2

- ▶ Making sure all datasets to be kept have upcase variables.
- ▶ e.g. suave.demog dataset:

<u>subjid</u>	<u>AGE</u>	<u>rAcE</u>	<u>bwt</u>	<u>BHt</u>
1000001	79	1	224	116
1000002	77	3	197	3
1000003	71	5	148	59
1000004	3	2	54	57
1000005	46	5	15	163
1000006	76	4	34	61
1000007	38	5	179	40
1000008	44	2	5	174
1000009	59	3	200	61
1000010	51	3	179	154

```
*****;  
* Problem 2 *;  
* Variable names are of mixed case in datasets, please upcase them all for required datasets *;  
*****;
```

```
□ %macro upcasevariables(lib=, dsn=);
```

```
  %let lib= %upcase(&lib.);
```

```
  %let dsn= %upcase(&dsn.);
```

```
  proc sql noprint ;
```

```
    select count(*) into :numrows from (select distinct name from dictionary.columns  
    where libname = "&lib."      and memname= "&dsn.");
```

```
    %let numrows=&numrows;
```

```
    %put number of variables: &numrows ;
```

```
    select distinct name into :vars1-:vars&numrows from dictionary.columns where libname = "&lib."  
    and memname= "&dsn.";
```

```
quit;
```

```
  %do i = 1 %to &numrows.;
```

```
    %if &&vars&i ne %upcase(&&vars&i) %then %do;
```

```
      proc datasets nolist lib=&lib.;
```

```
      modify &dsn.;
```

```
      rename &&vars&i=%upcase( &&vars&i);
```

```
      run;
```

```
      quit;
```

```
    %end;
```

```
  %end;
```

```
%mend;
```

```
%upcasevariables(lib=suave,dsn=demog);
```

```
%upcasevariables(lib=suave,dsn=handed);
```

```
%upcasevariables(lib=suave,dsn=labs);
```

```
%upcasevariables(lib=work,dsn=labs2);
```

<u>SUBJID</u>	<u>AGE</u>	<u>RACE</u>	<u>BWT</u>	<u>BHT</u>
1000001	79	1	224	116
1000002	77	3	197	3
1000003	71	5	148	59
1000004	3	2	54	57
1000005	46	5	15	163
1000006	76	4	34	61
1000007	38	5	179	40
1000008	44	2	5	174
1000009	59	3	200	61
1000010	51	3	179	154

Case 3

- ▶ Spec received recently:
- ▶ “One other query, is it possible to add the Double Delta for ECG parameters (HR, PRI, QTF, QT, QTB, RR, QTS) for Parts A, B and C please?”
- ▶ “The double delta value would be the difference between DHR and the DHR_PBO values (time-matched).”

```

* Macro Version *;
%macro mdd(dsnin111=,dsnout111=,c1=,c2=,v1=,m=);
proc sort data=&dsnin111 out=&m.1;
  by nsid ntpd &c1;
run;

data &m.2 (rename=(&c1=&c1._pbo));
  set &m.1 (where=(lowercase(trtg)="&v1" and &c1 ne . and ntpd gt 0 ));
  by nsid ntpd &c1;
  if first.&c1;
run;

data &m.3;
  merge &m.1 (in=a) &m.2 (keep=nsid ntpd &c1._pbo);
  by nsid ntpd;
  if a;
run;

data &dsnout111;
  length &c2 8. ;
  set &m.3;
  if nmiss(&c1,&c1._pbo)=0 then &c2=(&c1-&c1._pbo);
run;
%mend mdd;

```

* Part A *;

```
%mdd(dsnin111=pkecg_a,dsnout111=z5e,c1=dhr,c2=ddhr,v1=placebo,m=mdd1_);  
%mdd(dsnin111=z5e,dsnout111=z5f,c1=dpri,c2=ddpri,v1=placebo,m=mdd2_);  
%mdd(dsnin111=z5f,dsnout111=z5g,c1=dqtf,c2=ddqtf,v1=placebo,m=mdd3_);  
%mdd(dsnin111=z5g,dsnout111=z5h,c1=dqt,c2=ddqt,v1=placebo,m=mdd4_);  
%mdd(dsnin111=z5h,dsnout111=z5i,c1=dqtb,c2=ddqtb,v1=placebo,m=mdd5_);  
%mdd(dsnin111=z5i,dsnout111=z5j,c1=dr,r,c2=ddr,r,v1=placebo,m=mdd6_);  
%mdd(dsnin111=z5j,dsnout111=z5k,c1=dqts,c2=ddqts,v1=placebo,m=mdd7_);
```

* Part B *;

```
%mdd(dsnin111=pkecg_b,dsnout111=z6a,c1=dhr,c2=ddhr,v1=pooled_placebo,m=mdd8_);  
%mdd(dsnin111=z6a,dsnout111=z6b,c1=dpri,c2=ddpri,v1=pooled_placebo,m=mdd9_);  
%mdd(dsnin111=z6b,dsnout111=z6c,c1=dqtf,c2=ddqtf,v1=pooled_placebo,m=mdd10_);  
%mdd(dsnin111=z6c,dsnout111=z6d,c1=dqt,c2=ddqt,v1=pooled_placebo,m=mdd11_);  
%mdd(dsnin111=z6d,dsnout111=z6e,c1=dqtb,c2=ddqtb,v1=pooled_placebo,m=mdd12_);  
%mdd(dsnin111=z6e,dsnout111=z6f,c1=dr,r,c2=ddr,r,v1=pooled_placebo,m=mdd13_);  
%mdd(dsnin111=z6f,dsnout111=z6g,c1=dqts,c2=ddqts,v1=pooled_placebo,m=mdd14_);
```

* Part C *;

```
%mdd(dsnin111=pkecg_c,dsnout111=z7a,c1=dhr,c2=ddhr,v1=cohort_6:_placebo,m=mdd15_);  
%mdd(dsnin111=z7a,dsnout111=z7b,c1=dpri,c2=ddpri,v1=cohort_6:_placebo,m=mdd16_);  
%mdd(dsnin111=z7b,dsnout111=z7c,c1=dqtf,c2=ddqtf,v1=cohort_6:_placebo,m=mdd17_);  
%mdd(dsnin111=z7c,dsnout111=z7d,c1=dqt,c2=ddqt,v1=cohort_6:_placebo,m=mdd18_);  
%mdd(dsnin111=z7d,dsnout111=z7e,c1=dqtb,c2=ddqtb,v1=cohort_6:_placebo,m=mdd19_);  
%mdd(dsnin111=z7e,dsnout111=z7f,c1=dr,r,c2=ddr,r,v1=cohort_6:_placebo,m=mdd20_);  
%mdd(dsnin111=z7f,dsnout111=z7g,c1=dqts,c2=ddqts,v1=cohort_6:_placebo,m=mdd21_);
```

Tips to help when using macros.

- ▶ 1. Switch on extra log information to help with debugging: options mprint mlogic symbolgen;
- ▶ 2. Avoid overwriting work datasets by using the same name for each data step work dataset throughout the code (also applies to coding generally) as makes debugging more difficult.
- ▶ 3. Beware trying to get one macro to do too much. They are often at their most beneficial and efficient when performing robust functions within reliable parameters. Sometimes, like Economies of Scale in a big company, a macro which tries to do too much in the realm of the specifics, can create more cost than benefit.