PharmaSUG China 2016 - Paper 34

ESTIMATE/CONTRAST Statement too Complex? Tell the Effect Level and Let SAS Do It for You

Aide ZHOU, Sanofi, Beijing, China

Eric LIAO, Sanofi, Beijing, China

ABSTRACT

Write ESTIMATE/CONTRAST statement will be challenging when the model is complex or there are many levels for the effect, especially for the freshman. We develop a SAS macro to use 2-step approach, the 1st PROC MIXED step to get the coefficient data set, and then use the coefficient data set to automatically built the ESTIMATE/CONTRAST statement, what you need to do is just tell the macro which level of the effect (the actual data value in your data set) that you want to estimate/compare, then this macro will select/create the correct column (coefficient) used for the level of this effect and create the ESTIMATE/CONTRAST statement for you, then use the statement in the 2ND PROC MIXED step to do the analysis. This macro is easy to use and can minimize mistakes when we do the analysis.

INTRODUCTION

The key to write the ESTIMATE/CONTRAST statement is how to write coefficient, normally the biostatistician will set up the statistical model, and the statistical programmer need to build the ESTIMATE/CONTRAST statement to get the LS-means estimation and comparison. The syntax of writing the ESTIMATE or CONTRAST statements will depend on the internal SAS coding for the classification variables, an incorrect result may be obtained without understanding the internal SAS coding.

It is very important to understand the order of the parameters within effects that have multiple parameters (such as CLASS variables and interactions of CLASS variables). The ordering typically depends on the order in which the variables are specified in the CLASS statement and the setting of the ORDER= option in the PROC or CLASS statement.

In this paper, we illustrate an approach in PROC MIXED to tell SAS the actual data value in your data set which you want to estimate or contrast, and then apply 2-step approach to automatically generate the statement. It is more straightforward and easier to use. It can also be used to validate the statement.

THE DESIGN OF THIS MACRO

A macro named as %statement_estimate_contrast (see appendix) was developed.

Here are the 5 steps for the design of the macro.

1 Get the coefficient data set in the 1st PROC MIXED step using LSMEANS statement and ODS OUTPUT statement.

2 Identify or create the correct column for the desired effect.

3 Identify and delete the no needed blocks for the desired effect.

4 Concatenate the variable of EFFECT and the specific coefficient column to generate the statement and store it in a data set

5 Using %syscall set function to call all the ESTIMATE/CONTRAST statements in the 2nd PROC MIXED to do the analysis.

Take an MMRM model as an example to show how to manage it.

MODEL chg = rdstrf1n rdstrf2n countryc base base*aweek armn*aweek.

Where chg is the Change from baseline, rdstrf1n is the stratification one with 2 levels, rdstrf2n is the stratification two with 2 levels, aweek is the analysis week with 3 levels (week 8, 16 and 24), armn is the treatment group number with 2 levels (1, 2), base is the baseline (used as the covariate), countryc is the country name with 25 levels, and each subject is repeatedly tested at the 3 analysis weeks (week 8, 16, and 24) .

We want to estimate the LS-means at week 24 for each treatment groups, and the difference between the two treatment groups.

STEP 1, GET THE COEFFICIENT DATA SET IN THE 1ST PROC MIXED STEP USING LSMEANS STATEMENT AND ODS OUTPUT STATEMENT

Use the following code to get the coefficient data set (see table 1).

```
ODS OUTPUT Coef=_coef;
proc mixed data=eff method=reml;
  class subjid rdstrf1n rdstrf2n countryc armn aweek;
  model chg = rdstrf1n rdstrf2n countryc base
              base*aweek armn*aweek / ddfm=kr;
  repeated  aweek / subject=subjid type=UN;
  lsmeans   armn*aweek/e ;
run;
```

STEP 2, IDENTIFY OR CREATE THE CORRECT COLUMN FOR THE DESIRED EFFECT

One difficulty is to identify the correct column corresponding to desired effect levels, which depends on the order of the variables in the class statement, it is very important to write ESTIMATE statement, otherwise you may get the wrong result.

In this example, as armn is before the aweek in the class statement, thus aweek variable is nested inside of the armn. Here is the correspondence for column and the effect level (see table 2). Row3 for effect when ARMN=1 and AWEEK=24, and Row6 for effect when ARMN=2 and AWEEK=24.

If armn is after aweek in the CLASS statement using the code below, then armn variable is nested inside of aweek, the correspondence for column and the effect level (see table 3) would be - Row5 for effect when ARMN=1 and AWEEK=24, and Row6 for effect when ARMN=2 and AWEEK=24.

```
ODS output Coef=_coef;
proc mixed data=eff method=reml;
  class subjid rdstrf1n rdstrf2n countryc aweek armn;
  model chg = rdstrf1n rdstrf2n countryc base
              base*aweek armn*aweek / ddfm=kr;
  repeated  aweek / subject=subjid type=UN;
  lsmeans   armn*aweek/e ;
run;
```

But if you take a closer look at the 2 data sets, you will find that no matter which order is used in the CLASS statement, there is only one cell =1 for the specific level combination of the aweek and armn (see table 4 and 5), e.g. for the rows with armn=1 and aweek=24, only one cell=1 among Row1 to Row6, thus we can use SAS code to identify the correct column for the specific effect level.

Table 1. The original coefficient data set

LMatrix	Effect	RDSTRF1N	RDSTRF2N	COUNTRYC	ARMN	AWEEK	Row1	Row2	Row3	Row4	Row5	Row6
1	Intercept	_	_		_	_	1	1	1	1	1	1
1	RDSTRF1N	1	_		_	_	0.5	0.5	0.5	0.5	0.5	0.5
1	RDSTRF1N	2	_		_	_	0.5	0.5	0.5	0.5	0.5	0.5
1	RDSTRF2N	_	1		_	_	0.5	0.5	0.5	0.5	0.5	0.5
1	RDSTRF2N	_	2		_	_	0.5	0.5	0.5	0.5	0.5	0.5
1	COUNTRYC[a]	_	_	COUNTRY 1	_	_	0.04	0.04	0.04	0.04	0.04	0.04
1	COUNTRYC[a]	_	_	…	_	_	0.04	0.04	0.04	0.04	0.04	0.04
1	COUNTRYC[a]	_	_	COUNTRY 25	_	_	0.04	0.04	0.04	0.04	0.04	0.04
1	BASE	_	_		_	_	7.5884	7.5884	7.5884	7.5884	7.5884	7.5884
1	BASE*AWEEK	_	_		_	8	7.5884	0	0	7.5884	0	0
1	BASE*AWEEK	_	_		_	16	0	7.5884	0	0	7.5884	0
1	BASE*AWEEK	_	_		_	24	0	0	7.5884	0	0	7.5884
1	ARMN*AWEEK	_	_		1	8	1	0	0	0	0	0
1	ARMN*AWEEK	_	_		1	16	0	1	0	0	0	0
1	ARMN*AWEEK	_	_		1	24	0	0	1	0	0	0
1	ARMN*AWEEK	_	_		2	8	0	0	0	1	0	0
1	ARMN*AWEEK	_	_		2	16	0	0	0	0	1	0
1	ARMN*AWEEK	_	_		2	24	0	0	0	0	0	1

[a] Not all the country levels are shown here to simplify the output, all the coefficient is the same for all the country

Table 2, the coefficient data with illustration when armn is before aweek in the CLASS statement

LMatrix	Effect	RDSTRF1N	RDSTRF2N	COUNTRYC	ARMN	AWEEK	Row1	Row2	Row3	Row4	Row5	Row6
ARMN [b]							1	1	1	2	2	2
AWEEK [b]							8	16	24	8	16	24
1	Intercept	_	_		_	_	1	1	1	1	1	1
1	RDSTRF1N	1	_		_	_	0.5	0.5	0.5	0.5	0.5	0.5
1	RDSTRF1N	2	_		_	_	0.5	0.5	0.5	0.5	0.5	0.5
1	RDSTRF2N	_	1		_	_	0.5	0.5	0.5	0.5	0.5	0.5
1	RDSTRF2N	_	2		_	_	0.5	0.5	0.5	0.5	0.5	0.5
1	COUNTRYC [a]	_	_	COUNTRY 1	_	_	0.04	0.04	0.04	0.04	0.04	0.04
1	COUNTRYC [a]	_	_	…	_	_	0.04	0.04	0.04	0.04	0.04	0.04
1	COUNTRYC [a]	_	_	COUNTRY 25	_	_	0.04	0.04	0.04	0.04	0.04	0.04
1	BASE	_	_		_	_	7.5884	7.5884	7.5884	7.5884	7.5884	7.5884
1	BASE*AWEEK	_	_		_	8	7.5884	0	0	7.5884	0	0
1	BASE*AWEEK	_	_		_	16	0	7.5884	0	0	7.5884	0
1	BASE*AWEEK	_	_		_	24	0	0	7.5884	0	0	7.5884
1	ARMN*AWEEK	_	_		1	8	1	0	0	0	0	0
1	ARMN*AWEEK	_	_		1	16	0	1	0	0	0	0
1	ARMN*AWEEK	_	_		1	24	0	0	1	0	0	0
1	ARMN*AWEEK	_	_		2	8	0	0	0	1	0	0
1	ARMN*AWEEK	_	_		2	16	0	0	0	0	1	0
1	ARMN*AWEEK	_	_		2	24	0	0	0	0	0	1

[a] Not all the country levels are shown here to simplify the output, all the coefficient is the same for all the country
[b] The first 2 observations are added for illustration, which is not in the original coefficient data

Table 3, the coefficient data with illustration when armn is after aweek in the CLASS statement

Lmatrix	Effect	RDSTRF1N	RDSTRF2N	COUNTRYC	AWEEK	ARMN	Row1	Row2	Row3	Row4	Row5	Row6
AWEEK[b]							8	8	16	16	24	24
ARMN[b]							1	2	1	2	1	2
1	Intercept	_	_		_	_	1	1	1	1	1	1
1	RDSTRF1N	1	_		_	_	0.5	0.5	0.5	0.5	0.5	0.5
1	RDSTRF1N	2	_		_	_	0.5	0.5	0.5	0.5	0.5	0.5
1	RDSTRF2N	_	1		_	_	0.5	0.5	0.5	0.5	0.5	0.5
1	RDSTRF2N	_	2		_	_	0.5	0.5	0.5	0.5	0.5	0.5
1	COUNTRYC[a]	_	_	COUNTRY 1	_	_	0.04	0.04	0.04	0.04	0.04	0.04
1	COUNTRYC[a]	_	_	…	_	_	0.04	0.04	0.04	0.04	0.04	0.04
1	COUNTRYC[a]	_	_	COUNTRY 25	_	_	0.04	0.04	0.04	0.04	0.04	0.04
1	BASE	_	_		_	_	7.5884	7.5884	7.5884	7.5884	7.5884	7.5884
1	BASE*AWEEK	_	_		8	_	7.5884	7.5884	0	0	0	0
1	BASE*AWEEK	_	_		16	_	0	0	7.5884	7.5884	0	0
1	BASE*AWEEK	_	_		24	_	0	0	0	0	7.5884	7.5884
1	AWEEK*ARMN	_	_		8	1	1	0	0	0	0	0
1	AWEEK*ARMN	_	_		8	2	0	1	0	0	0	0
1	AWEEK*ARMN	_	_		16	1	0	0	1	0	0	0
1	AWEEK*ARMN	_	_		16	2	0	0	0	1	0	0
1	AWEEK*ARMN	_	_		24	1	0	0	0	0	1	0
1	AWEEK*ARMN	_	_		24	2	0	0	0	0	0	1

[a] Not all the country levels are shown here to simplify the output, all the coefficient is the same for all the country
[b] The first 2 observations are added for illustration, which is not in the original coefficient data

Table 4 the coefficient data with illustration when armn is before aweek in the CLASS statement and identify the column for the desired effect armn=1 and aweek=24 and the desired effect where armn=2 and aweek=24.

LMatrix	Effect	RDSTRF1N	RDSTRF2N	COUNTRYC	ARMN	AWEEK	Row1	Row2	Row3	Row4	Row5	Row6
ARMN[b]							1	1	1	2	2	2
AWEEK[b]							8	16	24	8	16	24
1	Intercept	_	_		_	_	1	1	1	1	1	1
1	RDSTRF1N	1	_		_	_	0.5	0.5	0.5	0.5	0.5	0.5
1	RDSTRF1N	2	_		_	_	0.5	0.5	0.5	0.5	0.5	0.5
1	RDSTRF2N	_	1		_	_	0.5	0.5	0.5	0.5	0.5	0.5
1	RDSTRF2N	_	2		_	_	0.5	0.5	0.5	0.5	0.5	0.5
1	COUNTRYC[a]	_	_	COUNTRY 1	_	_	0.04	0.04	0.04	0.04	0.04	0.04
1	COUNTRYC[a]	_	_	…	_	_	0.04	0.04	0.04	0.04	0.04	0.04
1	COUNTRYC[a]	_	_	COUNTRY 25	_	_	0.04	0.04	0.04	0.04	0.04	0.04
1	BASE	_	_		_	_	7.5884	7.5884	7.5884	7.5884	7.5884	7.5884
1	BASE*AWEEK	_	_		_	8	7.5884	0	0	7.5884	0	0
1	BASE*AWEEK	_	_		_	16	0	7.5884	0	0	7.5884	0
1	BASE*AWEEK	_	_		_	24	0	0	7.5884	0	0	7.5884
1	ARMN*AWEEK	_	_		1	8	1	0	0	0	0	0
1	ARMN*AWEEK	_	_		1	16	0	1	0	0	0	0
1	ARMN*AWEEK	_	_		1	24	0	0	1	0	0	0
1	ARMN*AWEEK	_	_		2	8	0	0	0	1	0	0
1	ARMN*AWEEK	_	_		2	16	0	0	0	0	1	0
1	ARMN*AWEEK	_	_		2	24	0	0	0	0	0	1

[a] Not all the country levels are shown here to simplify the output, all the coefficient is the same for all the country
[b] The first 2 observations are added for illustration, which is not in the original coefficient data

Table 5 the coefficient data with illustration when armn is after aweek in the CLASS statement and identify the column for the desired effect armn=1 and aweek=24 and desired effect where armn=2 and aweek=24.

Lmatrix	Effect	RDSTRF1N	RDSTRF2N	COUNTRYC	AWEEK	ARMN	Row1	Row2	Row3	Row4	Row5	Row6
AWEEK[b]							8	8	16	16	24	24
ARMN[b]							1	2	1	2	1	2
1	Intercept	_	_		_	_	1	1	1	1	1	1
1	RDSTRF1N	1	_		_	_	0.5	0.5	0.5	0.5	0.5	0.5
1	RDSTRF1N	2	_		_	_	0.5	0.5	0.5	0.5	0.5	0.5
1	RDSTRF2N	_	1		_	_	0.5	0.5	0.5	0.5	0.5	0.5
1	RDSTRF2N	_	2		_	_	0.5	0.5	0.5	0.5	0.5	0.5
1	COUNTRYC[a]	_	_	COUNTRY 1	_	_	0.04	0.04	0.04	0.04	0.04	0.04
1	COUNTRYC[a]	_	_	…	_	_	0.04	0.04	0.04	0.04	0.04	0.04
1	COUNTRYC[a]	_	_	COUNTRY 25	_	_	0.04	0.04	0.04	0.04	0.04	0.04
1	BASE	_	_		_	_	7.5884	7.5884	7.5884	7.5884	7.5884	7.5884
1	BASE*AWEEK	_	_		8	_	7.5884	7.5884	0	0	0	0
1	BASE*AWEEK	_	_		16	_	0	0	7.5884	7.5884	0	0
1	BASE*AWEEK	_	_		24	_	0	0	0	0	7.5884	7.5884
1	AWEEK*ARMN	_	_		8	1	1	0	0	0	0	0
1	AWEEK*ARMN	_	_		8	2	0	1	0	0	0	0
1	AWEEK*ARMN	_	_		16	1	0	0	1	0	0	0
1	AWEEK*ARMN	_	_		16	2	0	0	0	1	0	0
1	AWEEK*ARMN				24	1	0	0	0	0	1	0
1	AWEEK*ARMN				24	2	0	0	0	0	0	1

[a] Not all the country levels are shown here to simplify the output, all the coefficient is the same for all the country
[b] The first 2 observations are added for illustration, which is not in the original coefficient data

Table 6 Final coefficient data set

LMatrix	Effect	RDSTRF1N	RDSTRF2N	COUNTRYC	ARMN	AWEEK	Row1	Row2	Row3	Row4	Row5	Row6	Compare
1	Intercept	_	_		_	_	1	1	1	1	1	1	0
1	RDSTRF1N	1	_		_	_	0.5	0.5	0.5	0.5	0.5	0.5	0
1	RDSTRF1N	2	_		_	_	0.5	0.5	0.5	0.5	0.5	0.5	0
1	RDSTRF2N	_	1		_	_	0.5	0.5	0.5	0.5	0.5	0.5	0
1	RDSTRF2N	_	2		_	_	0.5	0.5	0.5	0.5	0.5	0.5	0
1	COUNTRYC[a]	_	_	COUNTRY 1	_	_	0.04	0.04	0.04	0.04	0.04	0.04	0
1	COUNTRYC[a]	_	_	…	_	_	0.04	0.04	0.04	0.04	0.04	0.04	0
1	COUNTRYC[a]	_	_	COUNTRY 25	_	_	0.04	0.04	0.04	0.04	0.04	0.04	0
1	BASE	_	_		_	_	7.5884	7.5884	7.5884	7.5884	7.5884	7.5884	0
1	BASE*AWEEK	_	_		_	8	7.5884	0	0	7.5884	0	0	0
1	BASE*AWEEK	_	_		_	16	0	7.5884	0	0	7.5884	0	0
1	BASE*AWEEK	_	_		_	24	0	0	7.5884	0	0	7.5884	0
1	ARMN*AWEEK	_	_		1	8	1	0	0	0	0	0	0
1	ARMN*AWEEK	_	_		1	16	0	1	0	0	0	0	0
1	ARMN*AWEEK	_	_		1	24	0	0	1	0	0	0	-1
1	ARMN*AWEEK	_	_		2	8	0	0	0	1	0	0	0
1	ARMN*AWEEK	_	_		2	16	0	0	0	0	1	0	0
1	ARMN*AWEEK	_	_		2	24	0	0	0	0	0	1	1

We can use the following code applying array statement to select the corresponding column and put the column number into a macro variable.

```
data  __test __control;
      set  _coef;
      if armn=2 and aweek=24 then output __test;
      if armn=1 and aweek=24 then output __control;
run;

data _null_;
      set __test;
      array tempv {*} row:;
      do i=1 to dim(tempv);
          if tempv {i}=1 then
          call symput("testrown",put(i,3.-L));
      end;
run;

data _null_;
      set __control;
      array tempv {*} row:;
      do i=1 to dim(tempv);
          if tempv {i}=1 then
          call symput("controlrown",put(i,3.-L));
      end;
run;
```

Then we can calculate the coefficient for the difference between treatment groups at week 24.

```
data __coef_final;
       set _coef;
       compare= row&testrown  -row&controlrown;
run;
```

Now we get the final coefficient data set (see table 6).

STEP 3, IDENTIFY AND DELETE THE NO NEEDED BLOCKS IN THE COEFFICIENT DATA SET FOR THE DESIRED EFFECT

The following two kinds of blocks in ESTIMATE/CONTRAST statement can be deleted.

1. Zero block

   The zero blocks means that for the same EFFECT values in the coefficient data set, all values are zero for the desired effect. For example, in the above Table 6, the observation rows for EFFECT = 'RDSTRF1N' for column compare is a zero block.

2. Balanced distribution block (for class variables only)

   The balanced distribution block means for the same EFFECT values in the coefficient data set, all values are identical for the desired effect. For example, in the above table 6, the observation for EFFECT = 'COUNTRYC' for column Row3 is a balanced distribution block.

Thus for treatment armn=2 and aweek=24 (see Row6), and armn=1 and aweek=24 (see Row3) the EFFECT for RDSTRF1N, RDSTRF2N and COUNTRYC can be deleted, and for the difference between the treatment groups (see row compare), only the effect ARMN*AWEEK need to be kept.

We can use SQL to first identify these blocks and delete them for the desired effect.

```
****identify 0 blocks***;
proc sql;
      create table look0 as
      select distinct effect
      from  __coef_final
      group by effect
      having max(&dsvar)=0
      ;
quit;

**** identify Balanced distributed blocks**;
proc sql;
      create table look1 as
      select distinct effect
      from  __coef_final
      group by effect
      having  count(*)>=2 and min(&dsvar)=max(&dsvar)
      ;
quit;

**** delete the 2 blocks identified****;

data __coef_final2;
      if _N_ = 1 then do;
      if 0 then set look0 ;
            declare hash lookup(dataset: "look0");
            rc=lookup.definekey('effect');
            rc=lookup.definedone();

      if 0 then set look1 ;
            declare hash lookup1(dataset: "look1");
            rc=lookup1.definekey('effect');
            rc=lookup1.definedone();
      end;
      set __coef_final;
      if lookup.find()=0 or lookup1.find()=0 then delete;
run;
```

STEP 4, CONCATENATE THE VARIABLE OF EFFECT AND THE SPECIFIC COEFFICIENT VARIABLE TO GENERATE THE STATEMENT AND STORE IT IN A DATA SET

The following codes are used applying catx function and retain statement.

```
data _null_;
      length estvar $500;
      set __coef_final2 end=eof;
      by Effect NOTSORTED;
      retain estvar ;

      if _n_=1 then do;
           estvar=catx(' ', "estimate", "ARM 1 AT WEEK 24" );
      end;

      if first.Effect then do;
            estvar=catx(' ', estvar,effect,row&testrown);
      end;

      else do;
           estvar=catx(' ', estvar,row&testrown);
      end;

      if eof then call symput
(cats("statement"),strip(estvar)||strip("/CL")||";");
run;
```

Do the similar step for control replacing variable row&testrown with row&controlrown, and the difference replacing variable row&testrown with compare, and store all the macro variables into a data set named as statement.

STEP 5, USING %SYSCALL SET FUNCTION TO CALL ALL THE ESTIMATE/CONTRAST STATEMENT IN THE 2ND PROC MIXED TO DO THE ANALYSIS.

Apply a small macro do_estimate_contrast, where parameter statement_dsin is the data set name used to store the estimate/contrast statement.

```
%macro do_estimate_contrast(statement_dsin=);
       %let dsid=%sysfunc(open(&statement_dsin,I));
           %if &dsid gt 0 %then %do;
            %syscall set(dsid);
                  %do %while (%sysfunc(fetch(&dsid)) eq 0);
                  &statement;
                  %end;
                  %end;
       %let rc=%sysfunc(close(&dsid));
%mend do_estimate_contrast;
```

```
proc mixed data=eff;
      class subjid armn rdstrf1n rdstrf2n countryc aweek;
      model chg = rdstrf1n rdstrf2n countryc base
                  base*aweek
                  armn*aweek / ddfm=kr;
      repeated  aweek / subject=subjid type=UN;
     %do_estimate_contrast(statement_dsin=statement);
run;
```

THE DESCRIPTION FOR EACH PARAMETER OF THIS MACRO

Parameter	Description and value to be used	Required	Example
statement_dsout	The data set name to be used to store the statement, valid SAS data name	Y	Statement
coefdsin	The coef data set to be used, valid SAS data name	Y	_coef
estimate_contrast	Specify the statement you want to do, estimate or contrast	Y	estimate
label	The model effect label to be used in the estimate or contrast statement, any word	N	drug A
test	The test effect level data point, SAS code for the data point	Y	ARMN=1
control	The control level data point, SAS code for the data point or NULL	Y (when you do contrast or estimate the difference)	ARMN=0
start	The first estimate or contrast statement, Y or NULL	N	Y
options	Other options to be add for this statement, valid option for the corresponding ESTIMATE or CONTRAST statement	N	CL
multiple	multiple cell means to be tested, Y or NULL	N	Y

HOW TO USE THIS MACRO

There are 3 steps to use this macro:

1. The 1st PROC MIXED with LSMEANS statement to get the coefficient data set.

2. Specify the all desired effect (actual data point level) you want to estimate/contrast to generate the statement and put the statement into a data set.

3. Using the statements in the 2nd PROC MIXED step by calling %do_estimate_contrast with the dataset generated in step 2.

Here is three examples show how to use this macro:

Example 1: Single arms estimation and straight comparison between two arms

Example 2: Pool effect using plus (+) to calculate pooled effect

Example 3: CONTAST using F test with more than 2 LS-means

EXAMPLE 1 SINGLE ARMS ESTIMATION AND STRAIGHT COMPARISON BETWEEN TWO ARMS

For the example show above, here are the steps to use the macro for the analysis.

Step 1, The 1st PROC MIXED with LSMEANS statement to get the coefficient data set.

```
ods output Coef= _coef;
proc mixed data=eff method=reml;
      class PATID TRT STRAT1 STRAT2 COUNTRY WEEK;
  model chg = rdstrf1n rdstrf2n countryc base
              base*aweek
              armn*aweek / ddfm=kr;
  repeated  aweek / subject=subjid type=UN;
  lsmeans   armn*aweek/e ;
run;
```

Step 2, Specify the all the effect levels (actual data point level) you want to ESTIMATE/CONTRAST to generate the statement and put the statement into a data set.

```
%statment_estimate_contrast
          (coefdsin         =_coef
          , estimate_contrast=estimate
          , label           =%str(Drug A)
          , test            =%str(armn=2 and aweek=24)
          , control         =
          , options         =cl
          , statement_dsout =statement
          , start           =Y);

%statment_estimate_contrast
          (coefdsin         =_coef
          , estimate_contrast=estimate
          , label           =%str(Drug B )
          , test            =%str(armn=1 and aweek=24)
          , control         =
          , options         =cl
          , statement_dsout =statement
          , start           =);

%statment_estimate_contrast
          (coefdsin         =_coef
          , estimate_contrast=estimate
          , label           =%str(Drug A vs drug B)
          , test            =%str(armn=2 and aweek=24)
          , control         =%str(armn=1 and aweek=24)
          , options         =cl
          , statement_dsout =statement
          , start           =);
```

Step 3, using the statements in the 2nd PROC MIXED step by calling %do_estimate_contrast with the dataset generated in step 2.

```
Options mprint;
proc mixed data=eff;
      class subjid armn rdstrf1n rdstrf2n countryc aweek;
      model chg = rdstrf1n rdstrf2n countryc base
                  base*aweek
                  armn*aweek / ddfm=kr;
      repeated  aweek / subject=subjid type=UN;
     %do_estimate_contrast(statement_dsin=statement);
run;
```

Note that you can use SAS OPTION (OPTIONS MPRINT) to put the statement in SAS log for validation.

EXAMPLE 2 POOL EFFECT USING PLUS (+) TO CALCULATE POOLED EFFECT

MODEL chg = base armn rdstrf1n rdstrf2n subgrp subgrp*armn;

Where chg is the Change from baseline, rdstrf1n is the stratification 1 with 2 levels , rdstrf2n is the stratification 2 with 2 levels, armn is the treatment group number with 2 levels (1, 2), base is the baseline (used as the covariate), subgrp has 4 levels(A,B,C,D).

We want to do some custom hypothesis to see the effect of the difference between the 2 treatment groups for subgroup A+B and A+B+C

1 1st PROC MIXED with LSMEANS statement to get the coefficient data set.

```
proc mixed data=EFF method=reml;
  class armn rdstrf1n rdstrf2n subgrp;
  model chg = base armn rdstrf1n rdstrf2n subgrp subgrp*armn /ddfm=kr;
  lsmeans subgrp*armn/e;
  ods output coef=_coef;
run;
```

2. Specify the all the effect levels (actual data point level) you want to estimate/contrast to generate the statement and put the statement into a data set.

```
%statment_estimate_contrast(coefdsin=_coef
    , estimate_contrast=estimate
    , label=%str(A+B: ARMN 2 VS 1)
    , test=%str((subgrp=1 and armn=2)+(subgrp=2 and armn=2))
    , control=%str((subgrp=1 and armn=1)+(subgrp=2 and armn=1))
    , options=cl
    , statement_dsout=statement
    , start=Y);

%statment_estimate_contrast(coefdsin=_coef
    , estimate_contrast=estimate
    , label=%str(A+B+C: ARMN 2 VS 1)
    , test=%str((subgrp=1 and armn=2)+(subgrp=2 and armn=2)+(subgrp=3
and armn=2))
    , control=%str((subgrp=1 and armn=1)+(subgrp=2 and armn=1) +
(subgrp=3 and armn=1))
    , options=cl
    , statement_dsout=statement
    , start=);
```

3. Using the statements in the 2nd PROC MIXED step by calling %do_estimate_contrast with the dataset generated in step 2.

```
*** ANCOVA results;
proc mixed data=EFF method=reml;
  class armn rdstrf1n rdstrf2n subgrp;
  model chg = base armn rdstrf1n rdstrf2n subgrp subgrp*armn /ddfm=kr;
  %do_estimate_contrast(statement_dsin=statement);
run;
```

EXAMPLE 3 CONTAST USING F TEST WITH MORE THAN 2 LS-MEANS

MODEL chg = base armn rdstrf1n rdstrf2n subgrp aweek armn* aweek;

Where chg is the change from baseline, rdstrf1n is the stratification 1 with 2 levels, rdstrf2n is the stratification 2 with 3 levels, armn is the treatment group number with 3 levels (1, 2, and 3), base is the baseline (used as the covariate), aweek is the analysis week with 3 levels (week 2, 4, 8 and 12) , and we want to get the overall F test to see if LS-means for all the treatment groups are the same.

1 1st PROC MIXED with LSMEANS statement to get the coefficient data set.

```
ods output coef=_coef;
PROC MIXED DATA=eff;
    CLASS subjid aweek armn rdstrf1n rdstrf2n;
    MODEL HABLCHG = base armn rdstrf1n rdstrf2n subgrp aweek armn* aweek;
    REPEATED aweek/ SUB = SUBJID TYPE = UN;
    LSMEANS ARMN* aweek /e;
RUN;
```

2 Specify the all the effect levels (actual data point level) you want to estimate/contrast to generate the statement and put the statement into a data set. Note that here we need to use multiple=Y.

```
%statment_estimate_contrast(coefdsin=_coef
, estimate_contrast=contrast,label=%str(F test)
, test=%str((ARMN=2 and visn=8))
, control=%str((ARMN=3 and visn=8))
, options=
, statement_dsout=statement
, start=Y);

%statment_estimate_contrast(coefdsin=_coef
, estimate_contrast=contrast,label=%str(F test)
, test=%str((ARMN=3 and visn=8))
, control=%str((ARMN=1 and visn=8))
, options=
, statement_dsout=statement
, start=
, multiple=Y);

%statment_estimate_contrast(coefdsin=_coef
, estimate_contrast=contrast,label=%str(F test)
, test=%str((ARMN=1 and visn=8))
, control=%str((ARMN=2 and visn=8))
, options=
, statement_dsout=statement
, start=
, multiple=Y);
```

3. Using the statements in the 2nd PROC MIXED step by calling %do_estimate_contrast with the dataset generated in step 2.

```
PROC MIXED DATA=eff;
    CLASS subjid aweek armn rdstrf1n rdstrf2n;
    MODEL HABLCHG = base armn rdstrf1n rdstrf2n subgrp aweek armn* aweek;
    REPEATED aweek/ SUB = SUBJID TYPE = UN;
    %do_estimate_contrast(statement_dsin=statement);
RUN;
```

CONCLUSION

This macro can help you to write the ESTIMATE/CONTRAST statement in a more straightforward way and can be used in the most of the cases, and you do not need to know so much about the internal SAS code, which has been done by the macro. You just need to specify the related the data point for the desired effect level. And you no need to pay attention to the variables order in the class statement, when the model is very complex or when you are not sure how to wire the statement, you can think about to use this macro. This macro can be used in SAS STAT PROC which support LSEMANS & ESTIMATE/contrast statement, such as GLM, MIXED, GENMOD. Although there are other new statements, such as LSMESTIMATE which can be easily used to do estimate and comparison, but ESTIMATE/CONTRAST statement is the most flexible one which can be used for any of liner combination of your model.

REFERENCES

Qi, Eric. 2002. “Two Step Approach to Automatically Generate ESTIMATE Statements for Pair-wise Treatment Comparisons”. Available at http://www.lexjansen.com/pharmasug/2002/proceed/Stats/sp09.pdf

Kiernan, Kathleen. 2011. “CONTRAST and ESTIMATE Statements Made Easy: The LSMESTIMATE Statement”. *Proceedings of the SAS Global Forum 2011 Conference,* Available at https://support.sas.com/resources/papers/proceedings11/351-2011.pdf
s

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Aide Zhou
Enterprise: Sanofi
Address: 2/F, HNA Building, No. 108, Jianguo Road, Chaoyang District, 100022, Beijing, China
E-mail: Aide.zhou@sanofi.com

APPENDIX

```
%macro statment_estimate_contrast(coefdsin=,
estimate_contrast=estimate,label=, test=, control=,options=, multiple=N,
statement_dsout=, start=);


%if &start=Y %then %do;
proc sql;
      create table &statement_dsout
  (
   estimate_contrast char(8),
   label char(200),
   test char(200),
   control char(200),
   options char(200),
   statement char(2000)
  );
quit;
%end;


**find which column to be used in the statement***;
            %if %nrbquote(&control) ne %then %do;
            %let dsvar=compare;

          %if %index(&test,+) %then %do;

                %let count=%eval(%sysfunc(count(&test,+))+1);
                %do iii=1 %to &count;

                     %let test1=%scan(&test,&iii,+);
                     %let control1=%scan(&control,&iii,+);

                           data  __test  __control;
                                 set &coefdsin;

                                 if &test1 then output __test;

                                 if &control1 then output __control;
                           run;

                  data _null_;
                        set __test;
                        array tempv {*} row:;
                        do i=1 to dim(tempv);
                             if tempv {i}=1 then
                             call symput("testrown&iii",put(i,3.-L));
                        end;
                  run;
                  %put >>>>>>>&&testrown&iii;

                 data _null_;
                      set __control;
                      array tempv {*} row:;
                      do i=1 to dim(tempv);
                           if tempv {i}=1 then
```

```
                              call symput("controlrown&&iii",put(i,3.-L));
                          end;
                    run;

                    %put >>>>>>>&&controlrown&iii;


                    %if &iii=1 %then %do;
                       data __coef_final;
                             set &coefdsin;
                                 ****FOR THE MOST CASE, for compare the
treatment effect ***;
                             compare= row&&testrown&iii  -
row&&controlrown&iii;
                        run;
                     %end;
                     %else %do;
                      data __coef_final;
                             set __coef_final;
                             ****FOR THE MOST CASE, for compare the treatment
effect ***;
                             compare= compare+ row&&testrown&iii  -
row&&controlrown&iii;
                       run;
                       %end;

                  %end;

              %end;


               %if %index(&test,+)=0 %then %do;

                    data  __test  __control;
                          set &coefdsin;
                          if &test then output __test;
                          if &control then output __control;
                     run;

                    data _null_;
                          set __test;
                          array tempv {*} row:;
                          do i=1 to dim(tempv);
                                if tempv {i}=1 then
                                     call symput("testrown",put(i,3.-L));
                          end;
                   run;

                   %put >>>>>>>&testrown;

                   data _null_;
                         set __control;
                         array tempv {*} row:;
                         do i=1 to dim(tempv);
                                if tempv {i}=1 then
                                call symput("controlrown",put(i,3.-L));
                          end;
```

```
                    run;

                    %put >>>>>>>&controlrown;

                     data __coef_final;
                           set &coefdsin;
                           ****FOR THE MOST CASE, for compare the treatment
effect ***;
                            compare= row&testrown  -row&controlrown;
                     run;

                  %end; ***end %index(&test,+)=0**;

            %end; ***end of if &control ne;


                %if %nrbquote(&control) eq %then %do;

                %if %index(&test,+) %then %do;
                %let count=%eval(%sysfunc(count(&test,+))+1);
                %do iii=1 %to &count;
                          %let test1=%scan(&test,&iii,+);

                               data  __test  ;
                                     set &coefdsin;
                                     if &test1 then output __test;
                               run;

                     data _null_;
                           set __test;
                           array tempv {*} row:;
                           do i=1 to dim(tempv);
                                if tempv {i}=1 then
                                call symput("testrown&iii",put(i,3.-L));
                           end;
                     run;
                     %put >>>>>>>&&testrown&iii;

                    %let dsvar=test;

                    %if &iii=1 %then %do;
                       data __coef_final;
                             set &coefdsin;
                             test= row&&testrown&iii;
                       run;
                     %end;
                     %else %do;
                      data __coef_final;
                             set __coef_final;
                             test= test+ row&&testrown&iii;
                      run;
                      %end;
                  %end; ***end of to ***;
            %end; **** end of test +;
                %if %index(&test,+)=0 %then %do;
             data __coef_final;
                   set &coefdsin;
```

```
            run;

            data  __test;
                  set &coefdsin;
                  if &test then output __test;
            run;

            data _null_;
                  set __test;
                  array tempv {*} row:;
                  do i=1 to dim(tempv);
                        if tempv {i}=1 then
                        call symput('testrown',put(i,3.-L));
                  end;
             run;

             %put >>>>>>>&testrown;

             %let dsvar=row&testrown;
                  %end;
            %end; ***end of if &control eq ;



****identify 0 blocks***;
proc sql;
      create table look0 as
      select distinct effect
      from  __coef_final
      group by effect
      having max(&dsvar)=0
      ;
quit;

**** identify Balanced distributed blocks**;
proc sql;
      create table look1 as
      select distinct effect
      from  __coef_final
      group by effect
      having  count(*)>=2 and min(&dsvar)=max(&dsvar)
      ;
quit;

*****decide divisor***;
%let divisor_no=1;


%if &multiple ne Y %then %do;
%let divisor_no=%eval(%sysfunc(count(&test,+))+1);
%end;

**** delete the 2 blocks identified****;
data __coef_final2;
      if _N_ = 1 then do;
      if 0 then set look0 ;
            declare hash lookup(dataset: "look0");
```

```
            rc=lookup.definekey('effect');
            rc=lookup.definedone();

      if 0 then set look1 ;
            declare hash lookup1(dataset: "look1");
            rc=lookup1.definekey('effect');
            rc=lookup1.definedone();
      end;
      set __coef_final;
      if lookup.find()=0 or lookup1.find()=0 then delete;
run;


data _null_;
      length estvar __coef $500;
      set __coef_final2 end=eof;
      by Effect NOTSORTED;
      retain estvar __coef;

      if _n_=1 then do;
           estvar=catx(' ', strip("&estimate_contrast"),
"'"||(strip("&label")||"'" ));
            __coef=' ';
      end;

      if first.Effect then do;
            estvar=catx(' ', estvar,effect,&dsvar);
            __coef=catx(' ', __coef,effect,&dsvar);
      end;

      else do;
            estvar=catx(' ', estvar,&dsvar);
            __coef=catx(' ', __coef,&dsvar);
      end;

      *****add divisor options***;
      %if &divisor_no = 1 %then %do;
      if eof then call symput
(cats("statement"),strip(estvar)||strip("/&options")||";");
      %end;
    %else %do;
      if eof then call symput
(cats("statement"),strip(estvar)||strip("/&options")||"
divisor=&divisor_no;");
      %end;
      if eof then call symput ("__coef",__coef);
run;
%put >>>> &statement;
%put >>>> __coef=&__coef;


%if &estimate_contrast=contrast and &multiple=Y and &start ne Y %then %do;
 data &statement_dsout;
      set &statement_dsout;
      where estimate_contrast="&estimate_contrast" and label="&label";
      statement=catx(', ', substr(statement,1,find(statement,'/')-1),
"&__coef");
```

```
      test=catx(', ', test, "&test");
      control=catx(', ', control, "&control");
run;
%end;

%else %do;
proc sql;
      insert into &statement_dsout
      set statement="&statement", estimate_contrast="&estimate_contrast",
label="&label", test="&test",
    control="&control", options="&options"
 ;
quit;
%end;

%mend statment_estimate_contrast;
*************************************************************************;
*                      End of Code                                      *;
*************************************************************************;
```