

Generating Least Square Means, Standard Error, Observed Mean, Standard Deviation and Confidence Intervals for Treatment Differences using Proc Mixed

Richann Watson

ABSTRACT

Have you ever wanted to calculate the confidence intervals for treatment differences or calculate the least square means using a mixed model but can't always recall the correct options or layout of the model for PROC MIXED? If so, the macro presented in this paper will appeal to you. The DOMIXED macro allows for the calculation of least square means, standard error, observed mean, standard deviation and confidence intervals for treatment difference. The macro will also calculate p-values.

MOTIVATION

The motivating factor for the DOMIXED macro was the generation of multiple tables that required treatment comparisons and the p-values for different dependent variables as well as calculating the least square means, standard error, observed mean and standard deviation. The table required 1) observed mean and standard deviation, 2) least square means and standard error, 3) p-value for the Analysis of Covariance model with treatment as one of the factors and a fixed effect as a covariate, 4) confidence intervals for treatment difference, and 5) p-values for treatment comparisons. Furthermore, additional outputs were incorporated into the macro. For example, the estimates and standard error of treatment differences.

SPECIFIC EXAMPLE

The following is an example of a table layout that will be used throughout this paper.

Systolic Blood Pressure Mean Change from Baseline Analysis

	Treat 1	Treat 2	Treat 3	Treat 4	P-Value
Obs. Mean	##.#	##.#	##.#	##.#	#####
Obs. Std Deviation	###.###	###.###	###.###	###.###	
LS Mean	##.#	##.#	##.#	##.#	
Standard Error	###.###	###.###	###.###	###.###	
Treatment Difference	Estimate (Std Error)	95% Confidence Intervals		P-Values	
1 vs. 2	###.### (###.###)	(###.### - ###.###)		#####	
1 vs. 3	###.### (###.###)	(###.### - ###.###)		#####	
1 vs. 4	###.### (###.###)	(###.### - ###.###)		#####	
2 vs. 3	###.### (###.###)	(###.### - ###.###)		#####	
2 vs. 4	###.### (###.###)	(###.### - ###.###)		#####	
3 vs. 4	###.### (###.###)	(###.### - ###.###)		#####	

PREPARE THE INPUT DATA

Before the DOMIXED macro can be applied, the input data set will need to be created. The data set needs a fixed effect(s) variable and a dependent variable for each record. For example, if the analysis is to be based on the mean change from baseline and the baseline value is the fixed effect, then the baseline value will need to be incorporated into each record and the baseline record removed. Then the change from baseline will need to be calculated for each record.

DATA SET	<i>Original</i>	<i>Original</i>	→	<i>Modified</i>
OBS NUMBER	1	2		1
TREAT	1	1		1
SITEGRP	1	1		1
SUBJID	001	001		001
VISIT	Baseline	Final		
SYSBP	105	109		
<i>BASESYS</i>				<i>105</i>
<i>CHGSYS</i>				<i>4</i>

Note: The variables that are ***red bold italics*** are variables that have been added. In addition, the VISIT and SYSBP variables have been dropped.

MACRO VARIABLES USED

There are several macro variables used in the DOMIXED macro that need to be defined in the call. An explanation of these macro variables and sample calls based on the example are in parenthesis provided below.

INDSN*: Data set which contains the variables you wish to analyze (e.g., newvital)--this defaults to the last used data set

Y: Dependent variable in the model (e.g., chgsys)

X: Fixed effects in the model (e.g., treat sitegrp basesys) **CLASS**: Class variables in the model (e.g., treat sitegrp)

LSMEANS*: Variables that the least squares mean and standard error needs to be determined for (should be in the model). This is also the variable for which the mean and standard deviation should be calculated. The differences between groups as well as the confidence intervals will be calculated (e.g., treat).

NOTE: If this is not specified lsmeans, standard error, mean and standard deviation will NOT be calculated.

SSTYPE*: Type of sum of square (types available are TESTS1, TESTS2, TESTS3) (e.g., 1) -- defaults to 3

BYVAR*: number of analyses that are to be performed (e.g., sitegrp -- there are 4 sitegrps -- 4 analyses will be done -- one for each sitegrp)

ALPHA*: Level of confidence used to determine the confidence intervals (e.g., 0.10) -- defaults to 0.05

RANDOM*: Random effects in the model (e.g., treat)

ESTIMATES*+: Specific estimate statements that the user wants to calculated (e.g.,

```
%str(estimate "1 vs. 2,3,4" treat 1 -.33 -.33 -.34 / cl;  
estimate "2 vs. 1,3,4" treat -.33 1 -.33 -.34 /cl;  
estimate "3 vs. 1,2,4" treat -.33 -.33 1 -.34 /cl;  
estimate "4 vs. 1,2,3" treat -.33 -.33 -.34 1 /cl;)
```

CONTRASTS*+: Specific contrast statements that the user wants to calculated (e.g.,

```
%str(contrast "1 vs. 2,3,4" treat 1 -.33 -.33 -.34;
```

```
contrast "2 vs. 1,3,4" treat -.33 1 -.33 -.34;
contrast "3 vs. 1,2,4" treat -.33 -.33 1 -.34;
contrast "4 vs. 1,2,3" treat -.33 -.33 -.34 1; )
```

CIMNFORM*: Format for the mean and confidence interval in the diffs1 data set (e.g., 5.2) -- defaults to 6.2

STDFORM*: Format for the standard error in the diffs1 data set (e.g., 6.3) - defaults to 7.3

TRANSPOSE*: Transpose the allmeans and the diffs1 data sets so that all the groups are in one record (e.g., N) -- defaults to Y.

NOTE: 4 treatments, all 4-treatment means are in one record, all 4 treatment stderr are in one record, etc.) It also transposes the fitstatistics so that all stats are on one record

* Identifies macro variables that can be left null if not applicable or if default is to be used.

+ The sample calls are not in the paper example.

INTERDEPENDENCIES OF MACRO VARIABLES

The macro variables LSMEANS, X, CLASS, ESTIMATES and CONTRASTS are interdependent in the DOMIXED macro. LSMEANS must be included in the CLASS and X list of variables. If the ESTIMATES or the CONTRASTS macro variables are used, then the fixed effect used to define the estimates and/or the contrasts statements must be contained in the X list of variables.

MACRO CALL FOR SPECIFIC EXAMPLE

The following is an example of the SAS code used to call the DOMIXED macro. This call corresponds with the example Systolic Blood Pressure Mean Change from Baseline Analysis used throughout this paper.

```
%DOMIXED (INDSN=NEWVITAL,
          Y = CHGSYS,
          X = TREAT SITEGRP BASESYS,
          CLASS = TREAT SITEGRP,
          LSMEANS = TREAT,
          CIMNFORM = 5.2,
          STDFORM = 6.3);
```

For the above macro call the following default values are being used:

```
SSTYPE = 3
ALPHA = .05
TRANSPOSE = Y
```

THE DOMIXED MACRO

The overall goal of the DOMIXED macro is to calculate the least square means, standard error, observed mean, standard deviation, confidence intervals for treatment difference and p-values. In addition, it can calculate the solutions for the fixed effects, the fit statistics, the solution for the random effects*, the estimates*, and the contrasts*. The * items will be calculated only if the appropriate macro variable is specified.

The macro can create up to 13 data sets. The main data sets are FITSTATS, TESTS#, SOLUTIONF, LSMEANS, MEANS and DIFFS. However, the macro will combine the LSMEANS and MEANS data set into one data set called ALLMEANS this will allow the observed mean and the least squares mean to be in the same data set. If the transpose variable is Y, the ALLMEANST, DIFFST and FITSTATT data sets will be created. The purpose of the transpose is so that the calculated variables for each treatment are in the same record. For example, the least square means for all the treatments are in one record instead of 4 records. This will allow for easier treatment comparison. Additional data sets are SOLUTIONR,

CONTRASTS and ESTIMATES. Even though some data sets will contain duplicate information, all data sets were kept so that the user could choose the data set structure that is either most appropriate for their table or that the user is most comfortable with. A description of each data set is in Appendix 1.

The code for the DOMIXED macro is available in Appendix 2. The following is a description of the logic used in the macro.

STEP 1. DETERMINE WHAT DATA SETS ARE TO BE GENERATED

The first step is to determine which data sets will be generated from the proc mixed model. The SOLUTIONF, TESTS# and FITSTATS will automatically be generated. If the LSMEANS macro variable is specified, then the LSMEANS and DIFFS data sets will be generated. If the RANDOM macro variable is specified, then the SOLUTIONR data set will be created. If the CONTRASTS/ESTIMATES macro variable(s) is specified, then the CONTRASTS/ESTIMATES data set will be created.

STEP 1A. GENERATE THE DATA SETS

Based on the data sets determined in Step 1, PROC MIX will generate the data sets for the specified model.

STEP 2. CALCULATE THE OBSERVED MEANS AND THE STANDARD DEVIATION, IF APPLICABLE (LSMEANS NE .)

If the LSMEANS macro variable is specified, then the observed mean and the standard deviation will be calculated and the MEANS data set will be created.

STEP 2A. CREATE THE ALLMEANS DATA SET, IF APPLICABLE (LSMEANS NE .).

If the LSMEANS, created in Step 1, and MEANS, created in Step 2, data sets exist, then they will be combined into the ALLMEANS data set.

STEP 2B. CREATE DIFFS1 DATA SET.

Using the DIFFS data set, created in Step 1, the DIFFS1 data set will be created with 3 new variables—label, ci and est_std. The label will be composed of the effect difference that is being calculated. For example, treatment 1 – treatment 2 is being calculated, then the label would be “1 vs. 2”. The ci is the confidence interval for the effect difference. It will combine the lower and the upper range into one variable. The estimate of the effect difference and the standard error will be combined to create the est_std variable.

STEP 3. DETERMINE IF ANY OF THE DATA SETS SHOULD BE TRANSPOSED (TRANPOSE = Y).

Need to determine if any data sets should be transposed. By default, the ALLMEANS (if it exists), DIFFS1 (if it exists) and the FITSTATS data sets will be transposed. Note: transposing the TESTS#, SOLUTIONF, SOLUTIONR, CONTRASTS or ESTIMATES have no added benefit.

STEP 3A. TRANSPOSE THE MEANS AND STANDARD DEVIATION/ERROR TO CREATE ALLMEANST.

The data set ALLMEANS created in Step 2a is transposed so that all the observed means, all the least square means, all the standard deviation and all the standard error are on individual records for each effect (i.e. there will be 4 records for each effect).

STEP 3B. TRANSPOSE THE ESTIMATES AND CONFIDENCE INTERVALS FOR EFFECT DIFFERENCES TO CREATE DIFFST.

The data set DIFFS1 created in Step 2b is transposed so that all the estimates and standard errors for each effect are on one record and all the effect difference confidence intervals are on one record (i.e. there will be 2 records for each effect).

STEP 3C. TRANSPOSE THE FIT STATISTICS TO CREATE FITSTATT.

The data set FITSTATS created in Step 1 is transposed so that all the fit statistics are on one record (i.e. there will be only one record).

CREATE SPECIFIC EXAMPLE OUTPUT

The output data sets created by the DOMIXED macro can be used with PROC REPORT to create output files.

Note: To produce the layout in the example, two separate PROC REPORTS will need to be executed, the first to produce the means analysis and the second to produce the difference analysis. The ALLMEANST data set needs to have the SOLUTIONF data set merged in (need to only keep the treatment effect on the first record) and the records need to be put in the correct order in which they are to appear in the report (i.e. if the least squares calculations are to appear first then no reordering is necessary, if the observed calculations are to appear first then a sort variable will need to be created) (e.g. create newmeans data set).

In order to produce the first section of the desired report, the allmeanst and tests3 data sets were combined to create the newmeans data set. NOTE: Tests3 contains the p-value for all the fixed effects as well as the intercept. The newmeans data set only contains the p-value for the effect being tested (i.e. newtreat). See example below:

NEWMEANS data set:

OBS 1:	OBS 2:	OBS 3:	OBS 4:
COL1 = ###.##	COL1 = ###.##	COL1 = ###.##	COL1 = ###.##
COL2 = ###.##	COL2 = ###.##	COL2 = ###.##	COL2 = ###.##
COL3 = ###.##	COL3 = ###.##	COL3 = ###.##	COL3 = ###.##
COL4 = ###.##	COL4 = ###.##	COL4 = ###.##	COL4 = ###.##
LABEL = Observed Mean	LABEL = Standard Deviation	LABEL = Least Square Mean	LABEL = Standard Error
INDEX = 1	INDEX = 1	INDEX = 1	INDEX = 1
PROBF = #.###	PROBF = null	PROBF = null	PROBF = null
ORDER = 1	ORDER = 2	ORDER = 3	ORDER = 4

```
%let LONGLINE="%sysfunc(repeat(, %eval(100)))";
proc report data=newmeans;
  column (&LONGLINE "" index order label col1 col2 col3 col4 probf);
  define index / order noprint;
  define order / order noprint;
  define label / display left width=25 " ";
  define col1 / display left width=15 "Treat 1";
  define col2 / display left width=15 "Treat 2";
  define col3 / display left width=15 "Treat 3";
  define col4 / display left width=15 "Treat 4";
  define probf / display left width=10 "P-value";

  compute before index;
    line &LONGLINE;
  endcomp;
run;
```

```

proc report data=diffs1 split='^';
  columns (label est_std ci probt);
  define label / order width=20 "Treatment Difference^ ^";
  define est_std / display left width=15 "Estimate^(Std. Error) ^ ^";
  define ci / display left width=25 "95% Confidence^Intervals^ ^";
  define probt / display left width=10 "P-value^ ^";

  compute after;
    line
      &LONGLINE;
  endcomp;
run;

```

NOTE: If the layout for the difference is

	Treat1-Treat2	Treat1-Treat3	...	Treat3-Treat4
Estimate (Std. Error)	### (#####)	### (#####)	...	### (#####)
Confidence Interval	### - ###	### - ###	...	### - ###
P-value	####	####	...	####

Then the DIFFST data set will produce this layout.

CONCLUSION

The DOMIXED macro was designed to be robust under many circumstances. I have found this to be true. The macro is extremely helpful in determining the least square means, standard error, observed mean, standard deviation, confidence intervals and p-values for different kinds of simple models. The only disadvantage I have found is that the macro will not handle more complicated models such as split-plot, models with multiple random statements, weighted models or models with repeated effects at this time. In addition, if there are multiple variables specified for the lsmeans macro variable, they will need to be of the same type (i.e. all numeric or all character).

ACKNOWLEDGMENTS

Thanks to Lara Guttadauro for her advice in preparing this paper.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

APPENDIX 1: DATA SETS

FITSTATS: contains the values for the different fit statistics in separate records — variables: DESCR, VALUE
DESCR: XXXXXXXXXXXXXXXXXXXXXXXX VALUE: ####.#

FITSTAT*: contains the values for the different fit statistics in one record — variables: _NAME_, _2_RES_LOG_LIKELIHOOD, AIC_SMALLER_IS_BETTER, AICC_SMALLER_IS_BETTER, BIC_SMALLER_IS_BETTER

TESTS#: the # is dependent upon the type of sum of squares requested, contains the type # tests of fixed effects — variables: EFFECT, NUMDF, DENDF, FVALUE, PROBF (contains p-values for each fixed effect)
EFFECT: XXXXXXXX NUMDF: ### DENDF: ### FVALUE: #.## PROBF: #.####

SOLUTIONF: contains the fixed effects solution vector (i.e. estimates and standard error for each level of the fixed effect(s)) — variables: EFFECT, MODEL SPECIFIC, ESTIMATE, STDERR, DF, TVALUE, PROBT
EFFECT: XXXXXXXX MODEL SPECIFIC VARIABLES: XXXXXXXX ESTIMATE: #.####
STDERR: #.#### DF: ### TVALUE: #.## PROBT: #.####

SOLUTIONR: contains the random effects solution vector (i.e. estimates and standard error for each level of the random effect(s)) — variables: EFFECT, MODEL SPECIFIC, ESTIMATE, STDERRPRED, DF, TVALUE, PROBT
EFFECT: XXXXXXXX MODEL SPECIFIC VARIABLES: XXXXXXXX ESTIMATE: #.####
STDERRPRED: #.#### DF: ### TVALUE: #.## PROBT: #.####

LSMEANS: contains the least squares means estimate and standard error — variables: EFFECT, MODEL SPECIFIC, ESTIMATE, STDERR, DF, TVALUE, PROBT, ALPHA, LOWER, UPPER
EFFECT: XXXXXXXX MODEL SPECIFIC VARIABLES: XXXXXXXX ESTIMATE: #.####
STDERR: #.#### DF: ### TVALUE: #.## PROBT: #.####
ALPHA: #.## LOWER: #.#### UPPER: #.####

MEANS: contains the observed mean and standard deviation — variables: MODEL SPECIFIC, _TYPE_, _FREQ_, OBS_MEAN, OBS_STD
MODEL SPECIFIC VARIABLES: XXXXXXXX _TYPE_: ### _FREQ_: ### OBS_MEAN: #.#### OBS_STD: #.####

ALLMEANS: contains the data in the LSMEANS and MEANS data sets — variables: same variables in LSMEANS and MEANS data sets
EFFECT: XXXXXXXX MODEL SPECIFIC VARIABLES: XXXXXXXX LSM_EST: #.####
LSM_STD: #.#### DF: ### TVALUE: #.## PROBT: #.####
ALPHA: #.## LOWER: #.#### UPPER: #.#### _TYPE_: ###
FREQ: ### OBS_MEAN: #.#### OBS_STD: #.#### LSM_EST_: X.XXXX
LSM_STD_: X.XXXX OBS_MEAN_: X.XXXX OBS_STD_: X.XXXX

ALLMEANST*: contains the LS-mean estimate for each effect in one record, the standard error for each effect in one record, the observed mean for each effect in one record and the standard deviation for each effect in one record — variables: MODEL SPECIFIC, LABEL, INDEX

DIFFS1: contains the differences of the LS-means as well as the created variables for the combined data of the estimate and standard error and of the lower and upper range for the confidence limit — variables: EFFECT, MODEL SPECIFIC, ESTIMATE, STDERR, DF, TVALUE, PROBT, ALPHA, LOWER, UPPER, LABEL, CI, EST_STD
EFFECT: XXXXXXXX MODEL SPECIFIC VARIABLES: XXXXXXXX ESTIMATE: #.####
STDERR: #.#### DF: ### TVALUE: #.## PROBT: #.####
ALPHA: #.## LOWER: #.#### UPPER: #.#### LABEL: XXXXXXXXXXXXX
CI: #.## - #.## EST_STD: #.##(#.###)

DIFFST*: contains the confidence interval for each effect difference in one record, the LS-means difference estimate and standard error for each effect difference in one record — variables: MODEL SPECIFIC, LABEL

CONTRASTS: contains the results from the contrast statements — variables: LABEL, NUMDF, DENDF, FVALUE, PROBF
LABEL: XXXXXXXX NUMDF: ### DENDF: ### FVALUE: #.## PROBF: #.####

ESTIMATES: contains the results from the estimate statements — variables: LABEL, ESTIMATE, STDERR, DF, TVALUE, PROBT, LOWER, UPPER
LABEL: XXXXXXXX ESTIMATE: #.#### STDERR: #.#### DF: ###
TVALUE: #.## PROBT: #.#### LOWER: #.#### UPPER: #.####

* Transpose of previous data set.

APPENDIX 2: THE DOMIXED MACRO

```
/******  
INDSN = Data set which contains the variables you wish to analyze.  
This defaults to the last used dataset.  
  
Y = Dependent variable in the model.  
  
X = Fixed effects in the model  
  
CLASS = Class variables in the model (e.g. treat site)  
  
LSMEANS = Variable that the least squares mean and standard error needs to  
be determined for (should be in the model) this is also the  
variables that the mean and standard deviation should be calc  
The differences between groups as well as the confidence  
intervals will be calculated (optional)  
NOTE: if this is not specified lsmeans, standard error, mean  
and standard deviation will NOT be calculated  
  
SSTYPE = type of sum of square  
(e.g. htype 1 (TESTS1); htype 2 (TESTS2); htype 3 (TESTS3))  
default to 3  
  
BYVAR = number of analyses that are to be performed (e.g. there are 4  
sitegrps -- 4 analyses will be done -- one for each sitegrp) (optional)  
  
ALPHA = Level of confidence used to determine the confidence intervals  
default to .05  
  
RANDOM = Random effects in the model (optional)  
  
ESTIMATES= specific estimate statements that the user wants to calculated (optional)  
  
CONTRASTS= specific contrast statements that the user wants to calculated (optional)  
  
CIMNFORM = format for the mean & confidence interval in the diffs1 dataset  
default to 6.2  
  
STDFORM = format for the standard error in the diffs1 dataset  
default to 7.3  
  
TRANSPPOSE= transpose the allmeans and the diffs1 datasets so that all the  
groups are in one record (e.g. 4 treatments all 4 treatment  
means are in one record, all 4 treatment stderr are in one  
record, etc.) default to Y  
It also transposes the fitstatistics so that all stats are on  
one record  
*****/  
  
%macro domixed(indsn=_LAST_, y= , x= , class= , lsmeans= , sstype=3, byvar= ,  
alpha=.05, random= , estimates= , contrasts= , cimnform=6.2,  
stdform=7.3, transpose = Y);  
  
/* using proc mixed to get the least square means and standard error, */  
/* confidence intervals and p-values for the model specified */  
  
/* use ods to generate temporary datasets */  
/* create datasets that have p-values, confidence intervals and lsmeans to be used later */  
/* this will only be created if the user request that they are created */  
ods output fitstatistics = fitstats;  
ods output tests&sstype = tests&sstype;
```

```

/*****NEED TO ADD A CONDITION TO LOOK FOR EITHER '(' OR '!'*****/
/*****IF THIS IS IN THE X VARIABLE THEN THE SOLUTIONF DS *****/
/*****WILL NOT BE CREATED *****/
/****!!!!!!!!!!!!!!!!!!!!!! BEGIN SECTION TO DETERMINE WHAT DATA SETS ARE TO BE GENERATED !!!!!!!!!!!!!!!!!!!!!!!****/
ods output solutionf = solutionf;
%if &random ne %then %do;
  ods output solutionr = solutionr;
%end;
%if &lsmeans ne %then %do;
  ods output lsmeans = lsmeans;
  ods output diffs = diffs;
%end;
%if &contrasts ne %then %do;
  ods output contrasts = contrasts;
%end;
%if &estimates ne %then %do;
  ods output estimates = estimates;
%end;
/****!!!!!!!!!!!!!!!!!!!!!! END SECTION TO DETERMINE WHAT DATA SETS ARE TO BE GENERATED !!!!!!!!!!!!!!!!!!!!!!!****/

%if &byvar ne %then %do;
  proc sort data=&indsn;
    by &byvar;
  run;
%end;

/****!!!!!!!!!!!!!!!!!!!!!! BEGIN SECTION TO GENERATE THE DATA SETS !!!!!!!!!!!!!!!!!!!!!!!****/
proc mixed data=&indsn;
  %if &class ne %then %do;
    class &class;
  %end;

  /* if random is specified then need to use the new class variable */
  /* random statement if the random effect is specified */
  /* this will treat the specified variable as a random */
  /* variable as well as create the estimates */
  %if &random ne %then %do;
    /* need to determine if the type of sum of squares probabilities are to be calculated */
    /* htype=3 says that we want type III sum squares probabilities */
    /* htype=2 says that we want type II sum squares probabilities */
    /* htype=1 says that we want type I sum squares probabilities */
    model &y = &x / htype=&sstype solution;
    random &random / solution;
  %end;
  %else %do;
    model &y = &x / htype=&sstype solution;
  %end;

  %if &byvar ne %then %do;
    by &byvar;
  %end;

  /* get the least square means, standard error and p-values only if a variable is specified */
  /* also creates the estimates, standard error and p-value for the differences */
  /* it also creates the alpha level confidence intervals for the lsmeans and differences */
  %if &lsmeans ne %then %do;
    lsmeans &lsmeans / diff cl alpha=&alpha;
  %end;

  /* creates the estimates and standard error for the user specified estimate statement(s) */
  &estimates;

  /* creates the contrasts for the user specified contrast statement(s) */
  &contrasts;

run;
/****!!!!!!!!!!!!!!!!!!!!!! END SECTION TO GENERATE THE DATA SETS !!!!!!!!!!!!!!!!!!!!!!!****/

```

```

/****!!!!!!!!!!!!!!!!!!!!!! BEGIN SECTION TO CALCULATE THE OBSERVED MEANS AND THE STANDARD DEVIATION !!!!!!!!!!!!!!!!!!!!!!!****/
/* if the lsmeans and mean are calculated then they need to be combined into one dataset */
%if &lsmeans ne %then %do;
/* using proc means to get the observed mean change and standard error for each fixed effect */
proc means data=&indsn;
  class &lsmeans;
  var &y;
  %if &byvar ne %then %do;
    by &byvar;
  %end;
  output out=means mean=obs_mean std=obs_std;
  ways 1; /* this does NOT allow for all conditional means -- does each unconditionally/independently */
run;
/****!!!!!!!!!!!!!!!!!!!!!! END SECTION TO CALCULATE THE OBSERVED MEANS AND THE STANDARD DEVIATION !!!!!!!!!!!!!!!!!!!!!!!****/

/****!!!!!!!!!!!!!!!!!!!!!! BEGIN SECTION TO CREATE THE ALLMEANS DATA SET !!!!!!!!!!!!!!!!!!!!!!!****/
/* determine the number of variables that the lsmeans/means are being calculated for */
%let i = 1;
%do %while(%scan(&lsmeans, &i) ne);
  %let i = %eval(&i + 1);
%end;
%let numlsmeans = %eval(&i - 1);

proc sort data=means;
  by &lsmeans;
run;

%LET GVAR=;
%DO I = 1 %TO &numlsmeans;
  %LET GVAR = &GVAR a.&i;
%END;

data _null_;
  set &indsn;
  array ls {*} &lsmeans;
  %let temp = &lsmeans ;
  %DO I = 1 %TO &numlsmeans;
    CALL SYMPUT("a&i", vformat(ls{&i})); /* determine the format of the lsmeans variables in the original data set */
    %let z&i = %scan(&temp,&i); /* create macro variables that contain the lsmeans variables */
  %END;
run;

/* redefine the lsmeans variable lengths based on what was in the original data set */
data lsmeans;
  set lsmeans;
  %do k = 1 %to &numlsmeans;
    format &z&k &&a&k;
  %end;
run;

/* if the data type is numeric then we need to redefine the */
/* missing values from ._ to . so that it will merge with the */
/* means data set */
data lsmeans;
  set lsmeans;
  array ls {*} &lsmeans;
  array x {*} $ x1 - x&numlsmeans;
  do j = 1 to &numlsmeans;
    x{j} = vtype(ls{j}); /* this will determine what the data type for each variable is */
    if x{j} = 'N' and ls{j} = ._ then ls{j} = .;
    if x{j} = 'C' then ls{j} = left(trim(ls{j}));
  end;
run;

proc sort data=lsmeans;
  by &lsmeans;
run;

```

```

/* redefine the lsmeans variable lengths based on what was in the original data set */
data means;
  set means;
  %do k = 1 %to &numlsmeans;
    format &z&k &a&k;
  %end;
run;

/* if the data type is numeric then we need to redefine the */
/* missing values from ._ to . so that it will merge with the */
/* means data set */
data means;
  set means;
  array ls {*} &lsmeans;
  array x {*} $ x1 - x&numlsmeans;
  do j = 1 to &numlsmeans;
    x{j} = vtype(ls{j}); /* this will determine what the data type for each variable is */
    if x{j} = 'N' and ls{j} = ._ then ls{j} = .;
    if x{j} = 'C' then ls{j} = left(trim(ls{j}));
  end;
run;

proc sort data=means;
  by &lsmeans;
run;

/* combine the lsmeans data with the observed means data */
data allmeans;
  merge lsmeans means;
  format obs_mean estimate &cimnform obs_std stderr &stdform;
  by &lsmeans &byvar;
  rename estimate = lsm_est
        stderr = lsm_std;
run;
/****!!!!!!!!!!!!!!!!!!!!!! END SECTION TO CREATE THE ALLMEANS DATA SET !!!!!!!!!!!!!!!!!!!!!!!*****/

/****!!!!!!!!!!!!!!!!!!!!!! BEGIN SECTION TO CREATE THE DIFFS1 DATA SET !!!!!!!!!!!!!!!!!!!!!!!*****/
/* need to create an array for the lsmean variables for the diffs */
/* in the diffs dataset the variables that are being compared are */
/* distinguished with the original variable name and the variable */
/* name with an "_" (i.e. newtreat and _newtreat) -- need to */
/* create the "_" variables */
proc sort data=diffs out=temp nodupkey;
  by effect;
run;

data _null_;
  set temp;
  retain _vars;
  length _vars $40.;
  _vars = compress(_vars) || "_" || effect;
  put "vars =" _vars;
  call symput('_lsmeans', _vars);
run;

```

```

/* create a confidence interval (combine lower and upper) */
/* create a mean and standard error variable */
data diffs1;
set diffs;
array vars {*} &lsmeans;
array _vars {*} &_lsmeans;
length label ci est_std $30.;
format probt 7.3;
do i = 1 to dim(vars);
if vars{i} ne . then do;
label = compress(vars{i})|| ' vs. ' || compress(_vars{i});
ci = compress(put(lower, &cimnform)) || ' - ' || compress(put(upper, &cimnform));
est_std = compress(put(estimate, &cimnform)) || ' (' || compress(put(stderr, &stdform)) || ')';
output;
end;
end;
run;

/* get rid of miscellaneous records */
data diffs1;
set diffs1;
if label ne ' _ vs. _';
run;
/****!!!!!!!!!!!!!!!!!!!!!! END SECTION TO CREATE THE DIFFS1 DATA SET !!!!!!!!!!!!!!!!!!!!!!!****/
%end;

/****!!!!!!!!!!!!!!!!!!!!!! BEGIN SECTION TO DETERMINE IF ANY OF THE DATA SETS SHOULD BE TRANSPOSED !!!!!!!!!!!!!!!!!!!!!!!****/
/* transpose the data so that all the treat lsmeans are on one record, */
/* all the stderr are on one record, all the observed means are on one */
/* record, all the standard deviations are on one record, all the ci */
/* for diffs are on one record, all the diffs mean and stderr are on */
/* one record -- makes for easier treatment to treatment comparison */
/* also always for printing treatments across rather than down a page */
%if &transpose = Y %then %do;

/****!!!!!!!!!!!!!!!!!!!!!! BEGIN SECTION TO TRANSPOSE THE MEANS AND STD/ERR TO CREATE ALLMEANST !!!!!!!!!!!!!!!!!!!!!!!****/
%if &lsmeans ne %then %do;
%if &byvar ne %then %do;
proc sort data=allmeans;
by &byvar;
run;
%end;

/* make the estimate/mean and the standard deviation/error */
/* into character variables so that they keep the format */
/* when they are transposed */
data allmeans;
set allmeans;
length lsm_est_ lsm_std_ obs_mean_ obs_std_ $20.;
lsm_est_ = left(trim(put(lsm_est, &cimnform)));
lsm_std_ = left(trim(put(lsm_std, &stdform)));
obs_mean_ = left(trim(put(obs_mean, &cimnform)));
obs_std_ = left(trim(put(obs_std, &stdform)));
run;

proc sort data=allmeans;
by effect &byvar;
run;

/* transpose the data so that all the obs means, ls means, all std and all ste are on individual records for each effect */
proc transpose data=allmeans out=allmeanst;
var lsm_est_ lsm_std_ obs_mean_ obs_std_;
by effect &byvar;
run;

```

```

/* create a label and an index variable that will be used to merge the p-value in */
data allmeanst; set
  allmeanst; length
  label $20.;
  if _NAME_ = 'obs_mean_' then label = 'Observed Mean'; if
  _NAME_ = 'obs_std_' then label = 'Standard Deviation'; if
  _NAME_ = 'lsm_est_' then label = 'Least Square Mean'; if
  _NAME_ = 'lsm_std_' then label = 'Standard Error'; index
  = 1;
run;
/****!!!!!!!!!!!!!!!!!!!!!! END SECTION TO TRANSPOSE THE MEANS AND STD/ERR TO CREATE ALLMEANST !!!!!!!!!!!!!!!!!!!!!****/

/****!!!!!!!!!!!!!!!!!!!!!! BEGIN SECTION TO TRANSPOSE THE EST & C.I. FOR EFFECT DIFF TO CREATE DIFFST !!!!!!!!!!!!!!!!!!!!!****/
proc sort data=diffs1;
  by effect &byvar;
run;

proc transpose data=diffs1 out=diffst prefix = label;
  var ci est_std probt;
  id label;
  by effect &byvar;

run;

data diffst;
  set diffst;
  length label $30.;
  if _NAME_ = 'label' then label = 'Label';
  if _NAME_ = 'ci' then label = 'Confidence Interval';
  if _NAME_ = 'est_std' then label = 'Estimate (Std. Error)';
  if _NAME_ = 'Probt' then label = 'P-Value';
  drop _NAME_ _LABEL_;
run;
%end;
/****!!!!!!!!!!!!!!!!!!!!!! END SECTION TO TRANSPOSE THE EST & C.I. FOR EFFECT DIFF TO CREATE DIFFST !!!!!!!!!!!!!!!!!!!!!****/

/****!!!!!!!!!!!!!!!!!!!!!! BEGIN SECTION TO TRANSPOSE THE FIT STATISTICS TO CREATE FITSTATT !!!!!!!!!!!!!!!!!!!!!****/
%if &byvar ne %then %do;
  proc sort data=fitstats;
    by &byvar;
  run;
%end;

proc transpose data=fitstats out=fitstatt;
  var value;
  id descr;
  %if &byvar ne %then %do;
    by &byvar;
  %end;
run;
/****!!!!!!!!!!!!!!!!!!!!!! END SECTION TO TRANSPOSE THE FIT STATISTICS TO CREATE FITSTATT !!!!!!!!!!!!!!!!!!!!!****/
%end;
/****!!!!!!!!!!!!!!!!!!!!!! END SECTION TO DETERMINE IF ANY OF THE DATA SETS SHOULD BE TRANSPOSED !!!!!!!!!!!!!!!!!!!!!****/

%mend;

```