

```
/* To run this program, users will need to change the location where files
are saved; To locate lines where changes are needed, search the file for the
phase C: */
```

```
/* Note also that the program below includes all conditions and will take
several days (or longer) to run; */
```

```
/* If you simply wish to test the program, go to Step 2 below and change the
reps to a smaller number (e.g., 3), or reduce the number of conditions (e.g.,
1 to 4); */
```

```
/* Sending log files to logfiles folder */
```

```
proc printto log = 'C:\Users\pituchka\Documents\NEWSTUDY\UNIVARIATE
PAPER\REVISION FEB 2019\logfiles\log'
new; run;
```

```
/* Creating File Containing Study Conditions */
/* Step 1: Study conditions */
```

```
data conditions;
do ESY1 = 0, .2, .5, .8;
do ESY2 = 0, .2, .5, .8;
do N = 20, 40, 60;
do var1 = 1;
do r12 = -.8 to .8 by .2;
do var2 = 1;
output;
end;
end;
end;
end;
end;
end;
end;
run;
```

```
/* Ensuring that r12 = 0 & creating r21 */
```

```
Data conditions; set conditions;
if r12 gt -.10 and r12 lt .10 then r12 = 0;
r21 = r12;
run;
```

```
/* Step 2: Creating second file containing Reps, seed, number of Conditions
*/
```

```
Data condnum;
do reps = 10000;
do seed = 0;
do cond = 1 to 432;
output;
end;
```

```

end;
end;
run;

/* Step 3: Re-orders, merges, and saves resultant data */

Data allconds;
  retain cond var1 r12 r21 var2 esy1 esy2 N reps seed;
  merge condnum conditions;
  run;

/* Step 4 Add Survey Sample N */

Data allconds;
set allconds;
  if N = 20 then do;
    N1 = 3;
    N2 = 6;
  end;
  if N = 40 then do;
    N1 = 6;
    N2 = 12;
  end;
  if N = 60 then do;
    N1 = 9;
    N2 = 18;
  end;
run;

/* Saving file containing all study conditions to desired folder */

Data 'C:\Users\pituchka\Documents\NEWSTUDY\UNIVARIATE PAPER\REVISION FEB
2019\allconds'; set allconds;
run;

/*****
*****

          Simulating Two-Group Multivariate Data Using Parameter Values
Residing in Allconds Data Set

*****
*****/

/* Sending log files to logfiles folder */

proc printto log = 'C:\Users\pituchka\Documents\NEWSTUDY\UNIVARIATE
PAPER\REVISION FEB 2019\logfiles\log'

```

```

new; run;

/* Inputting study conditions that are contained in "parent" data set
Allconds */
/* Each line contains values used for simulation study */

libname OUT 'C:\Users\pituchka\Documents\NEWSTUDY\UNIVARIATE PAPER\REVISION
FEB 2019\SASOutput';

data allconds;
    set 'C:\Users\pituchka\Documents\NEWSTUDY\UNIVARIATE PAPER\REVISION FEB
2019\allconds.sas7bdat';
    where cond between 1 and 432;
run;

%macro ODSOff();
ods graphics off;
ods exclude all;
ods results off;
options nosource nosource2 nonotes errors=0;;
%mend;

%let datetime_start = %sysfunc(TIME()) ;
%put START TIME: %sysfunc(datetime(),datetime14.);

%ODSOFF

/* Macro to SCAN through allconds data file
Will read each line of data sequentially from allconds data file*/

%MACRO
SCANLOOP(Allconds,Field1,Field2,Field3,Field4,Field5,Field6,Field7,Field8,Field9,Field10,Field11,Field12);

/* First provide the number of records in allconds */
DATA _NULL_;
IF 0 THEN SET &allconds NOBS=X;
CALL SYMPUT('RECCOUNT',X);
STOP;
RUN;

/* Loop from one to number of records */
%DO I=1 %TO &RECCOUNT;

/* Advance to the Ith record */
DATA _NULL_;
SET &allconds (FIRSTOBS=&I OBS=&I);

/* Store the variables of interest in macro variables */
CALL SYMPUT('Cond',&Field1);

```

```

CALL SYMPUT('Var1',&Field2);
CALL SYMPUT('r12',&Field3);
CALL SYMPUT('r21',&Field4);
CALL SYMPUT('var2',&Field5);
CALL SYMPUT('EsY1',&Field6);
CALL SYMPUT('EsY2',&Field7);
CALL SYMPUT('N',&Field8);
CALL SYMPUT('reps', &Field9);
CALL SYMPUT('seed',&Field10);
CALL SYMPUT('N1',&Field11);
CALL SYMPUT('N2',&Field12);
STOP;
RUN;

```

```

/* Creates data set with 2 groups with Group size = N/2 and reps =
Numsamples, retains study parameters; */
/* Does this for each study condition */

```

```

DATA Group;
Cond=&Cond;
DO Reps = 1 to &reps;
  DO ID = 1 to &N;
    IF ID le &N/2 then T = 0;
    ELSE T=1;
    OUTPUT;
  END;
END;
RUN;

```

```

DATA Group; SET Group;
  IF T = 1 THEN DO;
    PredY1 = &EsY1;
    PredY2 = &EsY2;
  END;
  ELSE IF T = 0 THEN DO;
    PredY1 = 0;
    PredY2 = 0;
  END;
  OUTPUT;
RUN;

```

```

/* Obtaining residuals from multivariate normal distribution using values
from allconds */

```

```

PROC IML;
USE allconds;
READ POINT &I VAR {Var1 r21 r12 var2} INTO Vector;
Cov = SHAPE (vector, 2, 2);
CLOSE allconds;
mean = {0, 0};
USE allconds;
READ POINT &I VAR {N} INTO N;
READ POINT &I VAR {Reps} INTO Reps;
CLOSE allconds;

```

```

call randseed(&seed);
R = RANDNORMAL(N*Reps, Mean, Cov);
Repsy = COLVEC(repeat(T(1:Reps), 1, N)); /* 1,1,1,...,2,2,2,...,3,3,3,... */
Z = Repsy || R;
CREATE MVN FROM Z[c={"Reps" "R1" "R2" }];
APPEND FROM Z;
CLOSE MVN;
QUIT;

/* Merging fixed data set with residuals data set */
/* Creating Observed Y scores as predicted plus residual */
/* ALLDATA contains simulated data for each study conditions */
/* Setting up data analysis for 15 and 30% Missing */

Data ALLDATA;
merge Group MVN;
  Y1 = PredY1 + R1;
  Y2 = PredY2 + R2;
if T = 1 then newT = 0;
else newT = 1;
  Y1_15 = Y1;
  Y1_30 = Y1;
  Y2_15 = Y2;
  Y2_30 = Y2;
run;

/* Inserting 15% and 30% Missing Data Generation */
/* Obtaining Cumulative Percents for Y2 */

data temp; set alldata;
run;

proc freq data=temp noprint;
tables Y2 / outcum out=pct_cum;
by cond reps;
run;

data pct_cum; set pct_cum;
probmiss = 1-(CUM_PCT/100);
run;

proc sort data = alldata;
by cond reps y2;
run;

Data merged;
merge alldata (in=a)
      pct_cum(In =b);
By cond reps;
If a and b;
drop COUNT PERCENT CUM_FREQ;
run;

/*****
/***** MISSINGNESS *****/
/*****/

```

```
/* 15 Percent Missing */
```

```
PROC SURVEYSELECT DATA=merged (Where=(cum_Pct < 50))
  METHOD=pps N=&N1 OUT =miss_15;
  STRATA COND REPS;
  size probmiss;
RUN;

Data miss_15; set miss_15;
Y1 = Y1_15;
Y1_15 = .;
Run;
```

```
proc sort data=miss_15;
by Cond Reps ID;
run;
```

```
proc sort data=alldata;
by Cond Reps ID;
run;
```

```
data alldata; merge alldata miss_15;
by Cond Reps ID;
drop SelectionProb SamplingWeight;
run;
```

```
/* 30 Percent Missing */
```

```
PROC SURVEYSELECT DATA=merged (Where=(cum_Pct < 50))
  METHOD=pps N=&N2 OUT =miss_30;
  STRATA COND REPS;
  size probmiss;
RUN;
```

```
Data miss_30; set miss_30;
Y1_30 = .;
drop Y1_15;
Run;
```

```
proc sort data=miss_30;
by Cond Reps ID;
run;
```

```
proc sort data=alldata;
by Cond Reps ID;
run;
```

```
data alldata; merge alldata miss_30;
by Cond Reps ID;
drop SelectionProb SamplingWeight;
run;
```

```

/*****
*****

```

ANALYSIS: COMPLETE DATA

```
*****  
*****/
```

```
/*****  
*****
```

Traditional ANOVA

```
*****  
*****/
```

```
PROC GLM DATA = alldata;  
ods output OverallANOVA=ALLAnova ;  
ods output errorSSCP= eSSCP ;  
ods output partialcorr = corrs ;  
ods output Multstat = WILKS ;  
ods output cldiffs = cldiff ;  
ods output lsmeans = means ;  
ods output LSMeanCL= clmeans ;  
ods output means = descriptives ;  
BY Cond reps;  
CLASS T;  
MODEL Y1 Y2 = T;  
MANOVA H = T / PRINTE;  
MEANS T/ ALPHA = .05 CLDIFF LSD;  
lsmeans T/ CL tdiff;  
means T;  
RUN;  
QUIT;
```

```
data wilks (keep = cond reps singular );  
set wilks ;  
by cond reps;  
firstrep = first.reps;  
if firstrep NE 1 then delete;  
if numdf < 3 then singular = 0;  
else singular = 1;  
run;
```

```
/* Obtaining treatment and control means for Y1 and Y2, along with CI for  
each group mean */
```

```
data clmeansy1 clmeansy2 ; set clmeans ;  
if dependent = 'Y1 ' then output CLmeansY1 ;  
else if dependent = 'Y2 ' then output CLmeansY2 ;  
run;
```

```
Data TCLMeanY1 CCLmeanY1 ;  
set clmeansY1 ;  
if T = 1 then output TCLmeanY1 ;
```

```

else if T = 0 then output CCLmeanY1 ;
run;

Data TCLMeanY1 ;
Set TCLMeanY1 ;
label Lowercl = TCILowerMeanY1 ;
label lsmean = TMeanY1 ;
label Uppercl = TCIUpperMeanY1 ;
rename Lowercl = TCILowerMeanY1 ;
rename lsmean = TMeanY1 ;
rename Uppercl = TCIUpperMeanY1 ;
Drop effect dependent T;
run;

Data CCLMeanY1 ;
Set CCLMeanY1 ;
label Lowercl = CCILowerMeanY1 ;
label lsmean = CMeanY1 ;
label Uppercl = CCIUpperMeanY1 ;
rename Lowercl = CCILowerMeanY1 ;
rename lsmean = CMeanY1 ;
rename Uppercl = CCIUpperMeanY1 ;
Drop effect dependent T;
run;

Data TCLMeanY2 CCLmeanY2 ;
set clmeansY2 ;
if T = 1 then output TCLmeanY2 ;
else if T = 0 then output CCLmeanY2 ;
run;

Data TCLMeanY2 ;
Set TCLMeanY2 ;
label Lowercl = TCILowerMeanY2 ;
label lsmean = TMeanY2 ;
label Uppercl = TCIUpperMeanY2 ;
rename Lowercl = TCILowerMeanY2 ;
rename lsmean = TMeanY2 ;
rename Uppercl = TCIUpperMeanY2 ;
Drop effect dependent T;
run;

Data CCLMeanY2 ;
Set CCLMeanY2 ;
label Lowercl = CCILowerMeanY2 ;
label lsmean = CMeanY2 ;
label Uppercl = CCIUpperMeanY2 ;
rename Lowercl = CCILowerMeanY2 ;
rename lsmean = CMeanY2 ;
rename Uppercl = CCIUpperMeanY2 ;
Drop effect dependent T;
run;

Data outputc ;
merge TCLMeanY1 CCLMeanY1 TCLMeanY2 CCLmeanY2 ;
by cond reps;
run;

```



```

/* Creating separate data sets for ANOVAs of Y1 and Y2 */
/* Splits ANOVA Results into Y1 and Y2 data sets */

Data ANOVAY1 ANOVAY2 ;
set aLLANOVA ;
if dependent = 'Y1 ' then output ANOVAY1 ;
else if dependent = 'Y2 ' then output ANOVAY2 ;
run;

/* F test for Y1 */
DATA FtestY1 ; keep cond reps dependent FValue ProbF; set ANOVAY1 ;
if Source = 'Model' then output;
label ProbF = p-value FY1 ;
rename probF = PFY1 ;
label FValue = FY1 ;
rename FValue = FY1 ;
Run;

/* F test for Y2 */
DATA FtestY2 ; keep cond reps dependent Fvalue ProbF; set ANOVAY2 ;
if Source = 'Model' then output;
label ProbF = p-value FY2 ;
rename probF = PFY2 ;
label FValue = FY2 ;
rename FValue = FY2 ;
Run;

Data outputc ;
set outputc;
merge wilks FtestY1 FtestY2 ;
by cond reps;
drop dependent;
if PFY1 <= .05 then Sig_FY1 = 1; else Sig_FY1 = 0;
if PFY2 <= .05 then Sig_FY2 = 1; else Sig_FY2 = 0;
If PFY1 <= .025 or PFY2 <= .025 then Sig_SUTS = 1; else Sig_SUTS = 0;
run;

/* Mean Square Error for Y1 */
DATA MSEY1 ; keep cond reps MS; set ANOVAY1 ;
if source = 'Error' then output;
label MS = MSEY1 ;
rename MS = MSEY1 ;
Run;

/* Mean Square Error for Y2 */
DATA MSEY2 ; keep cond reps MS; set ANOVAY2 ;
if source = 'Error' then output;
label MS = MSEY2 ;
rename MS = MSEY2 ;
Run;

Data outputc ;
set outputc ;
merge MSEY1 MSEY2 ;
by cond reps;
run;

```

```

/* Creating Confidence Limits Data Set */
DATA CL ; SET Cldiff ;
if Comparison = '1 - 0' then output;
drop method comparison;
run;

Data CLY1 CLY2 ;
set CL ;
if dependent = 'Y1 ' then output CLY1 ;
else if dependent = 'Y2 ' then output CLY2 ;
run;

Data CLY1 ;
set CLY1 ;
label lowerCL= lowerCLY1 ;
rename lowerCL= lowerCLY1 ;
label upperCl = upperCLY1 ;
rename upperCl = upperCLY1 ;
label Difference = DiffY1 ;
rename Difference = DiffY1 ;
label Significance = sigY1 ;
rename Significance = sigY1 ;
drop Effect Dependent;
Run;

Data CLY2 ;
set CLY2 ;
label lowerCL= lowerCLY2 ;
rename lowerCL= lowerCLY2 ;
label upperCl = upperCLY2 ;
rename upperCl = upperCLY2 ;
label Difference = DiffY2 ;
rename Difference = DiffY2 ;
label Significance = sigY2 ;
rename Significance = sigY2 ;
drop Effect Dependent;
Run;

Data outputc ;
set outputc ;
merge CLY1 CLY2 ;
by cond Reps;
Run;

/* Keeping Pooled Correlations */

Data NewCorrs1 ; Set Corrs ;
by cond reps;
firstrep = first.reps;
if firstrep NE 1 then delete;
keep cond reps Y2 P_Y2 ;
rename Y2 = Corr12 ;
label Y2 = Corr12 ;
Rename P_Y2 = P_Val_CORR12 ;
label P_Y2 = P-Val_CORR12 ;
DROP Y1 Variable P_y1 ;

```

```

run;

Data outputc ;
set outputc ;
merge NewCorrs1 ;
by cond Reps;
Run;

/* Obtain group Ns and SDs below */

Data Tinfo Cinfo ; Set Descriptives ;
By reps;
if T = 0 then output Cinfo ;
else if T = 1 then output Tinfo ;
run;

Data Tinfo ; Set Tinfo ;
rename N = TreatN ;
label N = TreatN ;
rename SD_Y1 = TSD_Y1;
label SD_Y1 = TSD_Y1 ;
rename SD_Y2 = TSD_Y2;
label SD_Y2 = TSD_Y2 ;
Drop Mean_Y1 Mean_Y2 Effect T;
run;

Data Cinfo ; Set Cinfo ;
rename N = ControlN ;
label N = ControlN ;
rename SD_Y1 = CSD_Y1;
label SD_Y1 = CSD_Y1 ;
rename SD_Y2 = CSD_Y2 ;
label SD_Y2 = CSD_Y2 ;
Drop Mean_Y1 Mean_Y2 Effect T;
run;

Data outputc ;
set outputc ;
merge Tinfo Cinfo ;
By cond reps;
N_both = TreatN +ControlN ;
TVar_Y1 = sqrt(TSD_Y1);
TVar_Y2 = sqrt(TSD_Y2);
CVar_Y1 = sqrt(CSD_Y1);
CVar_Y2 = sqrt(TSD_Y2);
Covar12 = corrl2/sqrt(msey1)*sqrt(msey2);
Run;

/* Obtaining t tests for Y1 Y2 */
Data ttestY1 ttestY2 ;
set means ;
if dependent = 'Y1 ' then do;
    if T = 0 then output ttestY1 ;
    end;
else if dependent = 'Y2 ' then do;
    if T = 0 then output ttestY2 ;
    end;

```

```
Drop lsmean effect dependent T probtdiff;
run;
```

```
Data ttestY1 ; Set ttesty1 ;
  label tdiff = ttestY1_glm ;
  rename tdiff = ttestY1_glm ;
Run;
```

```
Data ttestY2 ; Set ttesty2 ;
  label tdiff = ttesty2_glm ;
  rename tdiff = ttesty2_glm ;
run;
```

```
Data outputc ; set outputc;
merge ttesty1 ttesty2 ;
By Cond reps;
Run;
```

```
/* SE of Y1 Y2 Diffs for OLS */
Data outputc ;
Set outputc ;
SEDiffY1_glm = abs(diffY1 )/abs(TtestY1_glm );
SEDiffY2_glm = abs(diffY2 )/abs(TtestY2_glm );
run;
```

```
/****** CONVERTING WIDE DATA SET TO LONG DATA SET *****/
/****** Wide Data Set *****/
data Wide ; set alldata;
run;
```

```
/****** Wide to Long *****/
data Long ; set Wide ;
INDEX=1; Y =Y1 ; R =R1; output;
INDEX=2; Y =Y2 ; R =R2; output;
Keep reps ID INDEX T NewT Y R ;
run;
```

```
/******
*****
```

```
Linear Mixed Model with FIML and Z Tests for Effects of Y1 and Y2: 15%
Missingness
```

```
*****
*****/
```

```
/* Syntax for FIML and Default Between-Within df
```

Z Tests for effects of Y1 and Y2 obtained below */

```
PROC MIXED DATA=long METHOD=ML NOCLPRINT COVTEST;
  BY reps;
  CLASS index ID;
  MODEL Y = index index*T/ CL NOINT SOLUTION;
  REPEATED index / SUBJECT = ID TYPE=UN R;
  ods output ConvergenceStatus = MVMM_Conv_FullML ;
  ods output FitStatistics = MVMM_Fit_FullML ;
  ods output CovParms = MVMM_Random_FullML ;
  ods output SolutionF = MVMM_Fixed_FullML ;
  RUN;

/** convergence for Full Model With ML**/
data MVMM_Conv_FullML ; set MVMM_Conv_FullML ;
drop Reason pdG;
label Status = Status_MVMM_FullML ;
rename Status = Status_MVMM_FullML ;
label pdH = R_MVMM_FullML ;
rename pdH = R_MVMM_FullML ;
run;

proc freq data=MVMM_Conv_FullML noprint;
tables Reps*Status_MVMM_FullML *R_MVMM_FullML / out=MVMM_Conv_FullML ;
run;

Data MVMM_Conv_FullML ; set MVMM_Conv_FullML ;
label count = Conv_MVMM_FullML_Count ;
rename count = Conv_MVMM_FullML_Count ;
label percent = Conv_MVMM_FullML_Percent ;
rename percent = Conv_MVMM_FullML_Percent ;
run;

Data outputc ; set outputc ;
Merge MVMM_Conv_FullML ;
BY REPS;
Run;

/** Fixed Effects From Full Model With ML **/
data MVMM_Fixed_FullML ; retain Reps Parm; set MVMM_Fixed_FullML ;
label index = Y ;
label Estimate = Est_MVMM_Fixed_FullML ;
label StdErr = SE_MVMM_Fixed_FullML ;
label Probt = P_MVMM_Fixed_FullML ;
label Lower = CIL_MVMM_Fixed_FullML ;
label Upper = CIU_MVMM_Fixed_FullML ;
rename index = Y ;
rename Estimate = Est_MVMM_Fixed_FullML ;
rename StdErr = SE_MVMM_Fixed_FullML ;
rename Probt = P_MVMM_Fixed_FullML ;
rename Lower = CIL_MVMM_Fixed_FullML ;
rename Upper = CIU_MVMM_Fixed_FullML ;
drop Alpha;
if INDEX = 1 then do;
  if Effect = 'INDEX' then Parm = 'INT(Y1) ';
```

```

        if Effect = 'T*INDEX' then Parm = 'd(Y1) ';
end;
if INDEX = 2 then do;
    if Effect = 'INDEX' then Parm = 'INT(Y2) ';
    if Effect = 'T*INDEX' then Parm = 'd(Y2) ';
end;
run;

/** Creating Separate Data Files: Intercepts **/
data INTY1_ML INTY2_ML ; set MVMM_Fixed_FullML ;
If EFFECT = 'INDEX' & Y = 1 then output INTY1_ML ;
Else if EFFECT = 'INDEX' & Y = 2 then output INTY2_ML ;
Run;

data INTY1_ML ; set INTY1_ML ;
label Est_MVMM_Fixed_FullML = INTY1_ML ;
label SE_MVMM_Fixed_FullML = SE_INTY1_ML ;
label P_MVMM_Fixed_FullML = P_INTY1_ML ;
label CIL_MVMM_Fixed_FullML = CIL_INTY1_ML ;
label CIU_MVMM_Fixed_FullML = CIU_INTY1_ML ;
label tvalue = t_INTY1_ML ;
rename Est_MVMM_Fixed_FullML = INTY1_ML ;
rename SE_MVMM_Fixed_FullML = SE_INTY1_ML ;
rename P_MVMM_Fixed_FullML = P_INTY1_ML ;
rename CIL_MVMM_Fixed_FullML = CIL_INTY1_ML ;
rename CIU_MVMM_Fixed_FullML = CIU_INTY1_ML ;
rename tvalue = t_INTY1_ML ;
DROP PARM EFFECT Y df;
RUN;

data INTY2_ML ; set INTY2_ML ;
label Est_MVMM_Fixed_FullML = INTY2_ML ;
label SE_MVMM_Fixed_FullML = SE_INTY2_ML ;
label P_MVMM_Fixed_FullML = P_INTY2_ML ;
label CIL_MVMM_Fixed_FullML = CIL_INTY2_ML ;
label CIU_MVMM_Fixed_FullML = CIU_INTY2_ML ;
label tvalue = t_INTY2_ML ;
rename Est_MVMM_Fixed_FullML = INTY2_ML ;
rename SE_MVMM_Fixed_FullML = SE_INTY2_ML ;
rename P_MVMM_Fixed_FullML = P_INTY2_ML ;
rename CIL_MVMM_Fixed_FullML = CIL_INTY2_ML ;
rename CIU_MVMM_Fixed_FullML = CIU_INTY2_ML ;
rename tvalue = t_INTY2_ML ;
DROP PARM EFFECT Y df;
RUN;

/** Creating Separate Data Files: Slopes **/
data DY1_ML DY2_ML ; set MVMM_Fixed_FullML ;
If EFFECT = 'T*INDEX' & Y = 1 then output DY1_ML ;
Else if EFFECT = 'T*INDEX' & Y = 2 then output DY2_ML ;
Run;

data DY1_ML ; set DY1_ML ;
label Est_MVMM_Fixed_FullML = DY1_ML ;
label SE_MVMM_Fixed_FullML = SE_DY1_ML ;

```

```

label P_MVMM_Fixed_FullML = P_DY1_ML ;
label CIL_MVMM_Fixed_FullML = CIL_DY1_ML ;
label CIU_MVMM_Fixed_FullML = CIU_DY1_ML ;
label tvalue = t_DY1_ML ;
label df = df_DY1_ML ;
rename Est_MVMM_Fixed_FullML = DY1_ML ;
rename SE_MVMM_Fixed_FullML = SE_DY1_ML ;
rename P_MVMM_Fixed_FullML = P_DY1_ML ;
rename CIL_MVMM_Fixed_FullML = CIL_DY1_ML ;
rename CIU_MVMM_Fixed_FullML = CIU_DY1_ML ;
rename tvalue = t_DY1_ML ;
rename df = df_DY1_ML ;
DROP PARM EFFECT Y ;
RUN;

data DY2_ML ; set DY2_ML ;
label Est_MVMM_Fixed_FullML = DY2_ML ;
label SE_MVMM_Fixed_FullML = SE_DY2_ML ;
label P_MVMM_Fixed_FullML = P_DY2_ML ;
label CIL_MVMM_Fixed_FullML = CIL_DY2_ML ;
label CIU_MVMM_Fixed_FullML = CIU_DY2_ML ;
label tvalue = t_DY2_ML ;
label df = df_DY2_ML ;
rename Est_MVMM_Fixed_FullML = DY2_ML ;
rename SE_MVMM_Fixed_FullML = SE_DY2_ML ;
rename P_MVMM_Fixed_FullML = P_DY2_ML ;
rename CIL_MVMM_Fixed_FullML = CIL_DY2_ML ;
rename CIU_MVMM_Fixed_FullML = CIU_DY2_ML ;
rename tvalue = t_DY2_ML ;
rename df = df_DY2_ML ;
DROP PARM EFFECT Y ;
RUN;

Data outputc ; set outputc ;
Merge INTY1_ML INTY2_ML DY1_ML DY2_ML ;
BY REPS;
Run;

/* Creating P values for Z tests of Y1 and Y2 */
Data outputc ; set outputc ;
Merge INTY1_ML INTY2_ML DY1_ML DY2_ML ;
BY REPS;
if t_DY1_ML < 0 then
do;
pval_Zy1 = 2*cdf("Normal", t_DY1_ML );
end;
else
do;
pval_Zy1 = 2*(1-cdf("Normal", t_DY1_ML ));
end;
if pval_Zy1 <= .05 then SigZy1 = 1;
else SigZy1 = 0;
if t_DY2_ML < 0 then
do;
pval_Zy2 = 2*cdf("Normal", t_DY2_ML );
end;
else

```

```

do;
    pval_Zy2 = 2*(1-cdf("Normal", t_DY2_ML ));
end;
if pval_Zy1 <= .05 then SigZy1 = 1;
else SigZy1 = 0;
if pval_Zy2 <= .05 then SigZy2 = 1;
else SigZy2 = 0;
if pval_Zy1 <= .025 or pval_Zy2 <= .025 then OVERALL_ML = 1;
Else OVERALL_ML = 0;
Run;

```

```

Data outputc ; set outputc ;
TMeanY1_ML = DY1_ML + INTY1_ML ;
TMeanY2_ML = DY2_ML + INTY2_ML ;
Run;

```

```

/* Random Effects */
Data VarY1_ML CoVar12_ML VarY2_ML ; Set Mvmm_random_fullML ;
If covparm = 'UN(1,1)' then output VarY1_ML ;
If covparm = 'UN(2,1)' then output CoVar12_ML ;
If covparm = 'UN(2,2)' then output VarY2_ML ;
run;

```

```

Data VarY1_ML ; set VarY1_ML ;
label Estimate = VarY1_ML ;
label StdErr = SE_VarY1_ML ;
label Probz = P_VarY1_ML ;
rename Estimate = VarY1_ML ;
rename StdErr = SE_VarY1_ML ;
rename Probz = P_VarY1_ML ;
Drop Subject Covparm ZValue;
Run;

```

```

Data VarY2_ML ; set VarY2_ML ;
label Estimate = VarY2_ML ;
label StdErr = SE_VarY2_ML ;
label Probz = P_VarY2_ML ;
rename Estimate = VarY2_ML ;
rename StdErr = SE_VarY2_ML ;
rename Probz = P_VarY2_ML ;
Drop Subject Covparm ZValue;
Run;

```

```

Data CoVar12_ML ; set CoVar12_ML ;
label Estimate = CoVar12_ML ;
label StdErr = SE_CoVar12_ML ;
label Probz = P_CoVar12_ML ;
rename Estimate = CoVar12_ML ;
rename StdErr = SE_CoVar12_ML ;
rename Probz = P_CoVar12_ML ;
Drop Subject Covparm ZValue;
Run;

```



```

Data outputc ; Set outputc ;
Merge VarY1_ML  CoVar12_ML  VarY2_ML  ;
run;

```

```

/*****
*****/

```

Linear Mixed Model with REML and Default Denominator df: 15%
Missingness

```

*****/
*****/

```

```

PROC MIXED DATA=long METHOD=REML NOCLPRINT COVTEST;
  BY  reps;
  CLASS index ID newT;
  MODEL Y = index newT*Index / NOINT SOLUTION CL;
  lsmeans newT*Index / slice = index cl diff;
  REPEATED index / SUBJECT = ID TYPE=UN R;
  ods output ConvergenceStatus = MVMM_Conv_def ;
  ods output FitStatistics = MVMM_Fit_def ;
  ods output CovParms = MVMM_Random_def ;
  ods output LSMeans = MVMM_Means_def ;
  ods output Diffs = MVMM_Diffs_def ;
  ods output SolutionF = MVMM_Fixed_def ;
  RUN;

/** convergence for Full Model KR**/
data MVMM_Conv_def ; set MVMM_Conv_def ;
drop Reason pdG;
label Status = Status_MVMM_def ;
rename Status = Status_MVMM_def ;
label pdH = R_MVMM_def ;
rename pdH = R_MVMM_def ;
run;

proc freq data = MVMM_Conv_def noprint;
tables Reps*Status_MVMM_def *R_MVMM_def / out=MVMM_Conv_def ;
run;

Data MVMM_Conv_def ; set MVMM_Conv_def ;
label count = Conv_MVMM_def_Count;
rename count = Conv_MVMM_def_Count;
label percent = Conv_MVMM_def_Percent;
rename percent = Conv_MVMM_def_Percent;
run;

/** Means From Full Model With Def df **/
data MVMM_Means_def ; retain Reps Parm; set MVMM_Means_def ;
label index = Y;

```

```

label Estimate = Means_def ;
label StdErr = SE_MEANS_def ;
label Lower = CIL_MEANS_def ;
label Upper = CIU_MEANS_def ;
rename index = Y;
rename Estimate = Est_MEANS_def ;
rename StdErr = SE_MEANS_def ;
rename Lower = CIL_MEANS_def ;
rename Upper = CIU_MEANS_def ;
drop Effect tvalue Alpha probt df;
if INDEX = 1 then do;
    if newT = 0 then Parm = 'TMEAN(Y1def)';
    if newT = 1 then Parm = 'CMEAN(Y1def)';
end;
if INDEX = 2 then do;
    if newT = 0 then Parm = 'TMEAN(Y2def)';
    if newT = 1 then Parm = 'CMEAN(Y2def)';
end;
run;

/** Creating Separate Data Files: Means */
data CMEANY1def TXMEANY1def CMEANY2def TXMEANY2def ; set MVMM_Means_def ;
    If Parm = 'TMEAN(Y1def)' then output TXMEANY1def ;
    If Parm = 'CMEAN(Y1def)' then output CMEANY1def ;
    If Parm = 'TMEAN(Y2def)' then output TXMEANY2def ;
    If Parm = 'CMEAN(Y2def)' then output CMEANY2def ;
Run;

Data TXMEANY1def ; Set TXMEANY1def ;
label EST_MEANS_def = TMeanY1_def ;
label SE_MEANS_def = SE_TMeanY1_def ;
label CIL_MEANS_def = CIL_TMEANY1_def ;
label CIU_MEANS_def = CUL_TMEANY1_def ;
rename EST_MEANS_def = TMeanY1_def ;
rename SE_MEANS_def = SE_TMeanY1_def ;
rename CIL_MEANS_def = CIL_TMEANY1_def ;
rename CIU_MEANS_def = CUL_TMEANY1_def ;
DROP PARM Y;
RUN;

Data CMEANY1def ; Set CMEANY1def ;
label EST_MEANS_def = CMeanY1_def ;
label SE_MEANS_def = SE_CMeanY1_def ;
label CIL_MEANS_def = CIL_CMEANY1_def ;
label CIU_MEANS_def = CUL_CMEANY1_def ;
rename EST_MEANS_def = CMeanY1_def ;
rename SE_MEANS_def = SE_CMeanY1_def ;
rename CIL_MEANS_def = CIL_CMEANY1_def ;
rename CIU_MEANS_def = CUL_CMEANY1_def ;
DROP PARM Y;
RUN;

Data TXMEANY2def ; Set TXMEANY2def ;
label EST_MEANS_def = TMeanY2_def ;
label SE_MEANS_def = SE_TMeanY2_def ;
label CIL_MEANS_def = CIL_TMEANY2_def ;

```

```

label CIU_MEANS_def = CUL_TMEANY2_def ;
rename EST_MEANS_def = TMeanY2_def ;
rename SE_MEANS_def = SE_TMeanY2_def ;
rename CIL_MEANS_def = CIL_TMEANY2_def ;
rename CIU_MEANS_def = CUL_TMEANY2_def ;
DROP PARM Y;
RUN;

Data CMEANY2def ; Set CMEANY2def ;
label EST_MEANS_def = CMeanY2_def ;
label SE_MEANS_def = SE_CMeanY2_def ;
label CIL_MEANS_def = CIL_CMEANY2_def ;
label CIU_MEANS_def = CUL_CMEANY2_def ;
rename EST_MEANS_def = CMeanY2_def ;
rename SE_MEANS_def = SE_CMeanY2_def ;
rename CIL_MEANS_def = CIL_CMEANY2_def ;
rename CIU_MEANS_def = CUL_CMEANY2_def ;
DROP PARM Y;
RUN;

DATA outputc ; set outputc ;
Merge MVMM_Conv_def TXMEANY1def CMEANY1def TXMEANY2def CMEANY2def ;
By Repts;
Run;

/* Fixed Effects From Full Model With def */

data MVMM_Fixed_def ; retain Repts Parm; set MVMM_Fixed_def ;
label index = Y;
label Estimate = Est_MVMM_Fixed_def ;
label StdErr = SE_MVMM_Fixed_def ;
label Probt = P_MVMM_Fixed_def ;
label Lower = CIL_MVMM_Fixed_def ;
label Upper = CIU_MVMM_Fixed_def ;
rename index = Y;
rename Estimate = Est_MVMM_Fixed_def ;
rename StdErr = SE_MVMM_Fixed_def ;
rename Probt = P_MVMM_Fixed_def ;
rename Lower = CIL_MVMM_Fixed_def ;
rename Upper = CIU_MVMM_Fixed_def ;
drop Alpha;
if INDEX = 1 then do;
    if Effect = 'INDEX' then Parm = 'INT(Y1def)';
    if NewT = 0 then Parm = 'd(Y1def)';
end;
if INDEX = 2 then do;
    if Effect = 'INDEX' then Parm = 'INT(Y2def)';
    if NewT = 0 then Parm = 'd(Y2def)';
end;
run;

/** Creating Separate Data Files: Intercepts */

data INTY1_def INTY2_def ; set MVMM_Fixed_def ;
If EFFECT = 'INDEX' & Y = 1 then output INTY1_def ;
Else if EFFECT = 'INDEX' & Y = 2 then output INTY2_def ;

```

Run;

```
data INTY1_def ; set INTY1_def ;
label Est_MVMM_Fixed_def = INTY1_def ;
label SE_MVMM_Fixed_def = SE_INTY1_def ;
label P_MVMM_Fixed_def = P_INTY1_def ;
label CIL_MVMM_Fixed_def = CIL_INTY1_def ;
label CIU_MVMM_Fixed_def = CIU_INTY1_def ;
label tvalue = t_INTY1_def ;
rename Est_MVMM_Fixed_def = INTY1_def ;
rename SE_MVMM_Fixed_def = SE_INTY1_def ;
rename P_MVMM_Fixed_def = P_INTY1_def ;
rename CIL_MVMM_Fixed_def = CIL_INTY1_def ;
rename CIU_MVMM_Fixed_def = CIU_INTY1_def ;
rename tvalue = t_INTY1_def ;
DROP PARM EFFECT Y NewT df;
RUN;
```

```
data INTY2_def ; set INTY2_def ;
label Est_MVMM_Fixed_def = INTY2_def ;
label SE_MVMM_Fixed_def = SE_INTY2_def ;
label P_MVMM_Fixed_def = P_INTY2_def ;
label CIL_MVMM_Fixed_def = CIL_INTY2_def ;
label CIU_MVMM_Fixed_def = CIU_INTY2_def ;
label tvalue = t_INTY2_def ;
rename Est_MVMM_Fixed_def = INTY2_def ;
rename SE_MVMM_Fixed_def = SE_INTY2_def ;
rename P_MVMM_Fixed_def = P_INTY2_def ;
rename CIL_MVMM_Fixed_def = CIL_INTY2_def ;
rename CIU_MVMM_Fixed_def = CIU_INTY2_def ;
rename tvalue = t_INTY2_def ;
DROP PARM EFFECT Y NewT ;
RUN;
```

/** Creating Separate Data Files: Slopes **/

```
Data DY1_def DY2_def ; set MVMM_Fixed_def ;
If Y = 1 & NewT = 0 then output DY1_def ;
Else if Y = 2 & NewT = 0 then output DY2_def ;
Run;
```

```
data DY1_def ; set DY1_def ;
label Est_MVMM_Fixed_def = DY1_def ;
label SE_MVMM_Fixed_def = SE_DY1_def ;
label P_MVMM_Fixed_def = P_DY1_def ;
label CIL_MVMM_Fixed_def = CIL_DY1_def ;
label CIU_MVMM_Fixed_def = CIU_DY1_def ;
label tvalue = t_DY1_ML;
label df = df_DY1_ML;
rename Est_MVMM_Fixed_def = DY1_def ;
rename SE_MVMM_Fixed_def = SE_DY1_def ;
rename P_MVMM_Fixed_def = P_DY1_def ;
rename CIL_MVMM_Fixed_def = CIL_DY1_def ;
rename CIU_MVMM_Fixed_def = CIU_DY1_def ;
rename tvalue = t_DY1_def ;
rename df = df_DY1_def ;
```

```
DROP PARM EFFECT Y;  
RUN;
```

```
data DY2_def ; set DY2_def ;  
label Est_MVMM_Fixed_def = DY2_def ;  
label SE_MVMM_Fixed_def = SE_DY2_def ;  
label P_MVMM_Fixed_def = P_DY2_def ;  
label CIL_MVMM_Fixed_def = CIL_DY2_def ;  
label CIU_MVMM_Fixed_def = CIU_DY2_def ;  
label tvalue = t_DY2_def ;  
label df = df_DY2_def ;  
rename Est_MVMM_Fixed_def = DY2_def ;  
rename SE_MVMM_Fixed_def = SE_DY2_def ;  
rename P_MVMM_Fixed_def = P_DY2_def ;  
rename CIL_MVMM_Fixed_def = CIL_DY2_def ;  
rename CIU_MVMM_Fixed_def = CIU_DY2_def ;  
rename tvalue = t_DY2_def ;  
rename df = df_DY2_def ;  
DROP PARM EFFECT Y;  
RUN;
```

```
Data outputc ; set outputc ;  
Merge INTY1_def INTY2_def DY1_def DY2_def ;  
BY REPS;  
Run;
```

```
Data outputc ; set outputc ;  
if P_DY1_def <= .05 then SigDY1_def = 1;  
else SigDY1_def = 0;  
if P_DY2_def <= .05 then SigDY2_def = 1;  
else SigDY2_def = 0;  
if P_DY1_def <= .025 or P_DY2_def <= .025 then OVERALL_def = 1;  
Else OVERALL_def = 0;  
run;
```

```
/* Random Effects */  
Data VarY1_def CoVar12_def VarY2_def ; Set Mvmm_random_def ;  
If covparm = 'UN(1,1)' then output VarY1_def ;  
If covparm = 'UN(2,1)' then output CoVar12_def ;  
If covparm = 'UN(2,2)' then output VarY2_def ;  
run;
```

```
Data VarY1_def ; set VarY1_def ;  
label Estimate = VarY1_def ;  
label StdErr = SE_VarY1_def ;  
label Probz = P_VarY1_def ;  
rename Estimate = VarY1_def ;  
rename StdErr = SE_VarY1_def ;  
rename Probz = P_VarY1_def ;  
Drop Subject Covparm ZValue;  
Run;
```

```
Data VarY2_def ; set VarY2_def ;  
label Estimate = VarY2_def ;  
label StdErr = SE_VarY2_def ;  
label Probz = P_VarY2_def ;
```

```

rename Estimate = VarY2_def ;
rename StdErr = SE_VarY2_def ;
rename Probz = P_VarY2_def ;
Drop Subject Covparm ZValue;
Run;

```

```

Data CoVar12_def ; set CoVar12_def ;
label Estimate = CoVar12_def ;
label StdErr = SE_CoVar12_def ;
label Probz = P_CoVar12_def ;
rename Estimate = CoVar12_def ;
rename StdErr = SE_CoVar12_def ;
rename Probz = P_CoVar12_def ;
Drop Subject Covparm ZValue;
Run;

```

```

Data outputc ; Set outputc ;
Merge VarY1_def VarY2_def CoVar12_def ;
by reps;
run;

```

```

/*****
*****

```

Linear Mixed Model with REML and KR Denominator df: 15% Missing

```

*****
*****/

```

```

/* Syntax for Full Model With REML and KR Denominator df */
PROC MIXED DATA=long METHOD=REML NOCLPRINT COVTEST;
  BY reps;
  CLASS index ID newT;
  MODEL Y = index newT*index / DDFM = KR NOINT SOLUTION CL;
  lsmeans newT*index / slice = index cl diff;
  REPEATED index / SUBJECT = ID TYPE=UN R;
  ods output ConvergenceStatus = MVMM_Conv_KR ;
  ods output FitStatistics = MVMM_Fit_KR ;
  ods output CovParms = MVMM_Random_KR ;
  ods output LSMeans = MVMM_Means_KR ;
  ods output Diffs = MVMM_Diffs_KR ;
  ods output SolutionF = MVMM_Fixed_KR ;
RUN;

```

```

/** convergence for Full Model KR**/
data MVMM_Conv_KR ; set MVMM_Conv_KR ;
drop Reason pdG;
label Status = Status_MVMM_KR ;
rename Status = Status_MVMM_KR ;
label pdH = R_MVMM_KR ;
rename pdH = R_MVMM_KR ;

```

```

run;

proc freq data = MVMM_Conv_KR noprint;
tables Reps*Status_MVMM_KR *R_MVMM_KR / out=MVMM_Conv_KR ;
run;

Data MVMM_Conv_KR ; set MVMM_Conv_KR ;
label count = Conv_MVMM_KR_Count ;
rename count = Conv_MVMM_KR_Count ;
label percent = Conv_MVMM_KR_Percent ;
rename percent = Conv_MVMM_KR_Percent ;
run;

/** Means From Full Model With KR */
data MVMM_Means_KR ; retain Reps Parm; set MVMM_Means_KR ;
label index = Y ;
label Estimate = Means_KR ;
label StdErr = SE_MEANS_KR ;
label Lower = CIL_MEANS_KR ;
label Upper = CIU_MEANS_KR ;
rename index = Y ;
rename Estimate = Est_MEANS_KR ;
rename StdErr = SE_MEANS_KR ;
rename Lower = CIL_MEANS_KR ;
rename Upper = CIU_MEANS_KR ;
drop Effect tvalue Alpha probt df;
if INDEX = 1 then do;
    if newT = 0 then Parm = 'TMEAN(Y1KR) ' ;
    if newT = 1 then Parm = 'CMEAN(Y1KR) ' ;
end;
if INDEX = 2 then do;
    if newT = 0 then Parm = 'TMEAN(Y2KR) ' ;
    if newT = 1 then Parm = 'CMEAN(Y2KR) ' ;
end;
run;

/** Creating Separate Data Files: Means */
data CMEANY1KR TXMEANY1KR CMEANY2KR TXMEANY2KR ; set MVMM_Means_KR ;
If Parm = 'TMEAN(Y1KR) ' then output TXMEANY1KR ;
If Parm = 'CMEAN(Y1KR) ' then output CMEANY1KR ;
If Parm = 'TMEAN(Y2KR) ' then output TXMEANY2KR ;
If Parm = 'CMEAN(Y2KR) ' then output CMEANY2KR ;
Run;

Data TXMEANY1KR ; Set TXMEANY1KR ;
label EST_MEANS_KR = TMeanY1_KR ;
label SE_MEANS_KR = SE_TMeanY1_KR ;
label CIL_MEANS_KR = CIL_TMEANY1_KR ;
label CIU_MEANS_KR = CUL_TMEANY1_KR ;
rename EST_MEANS_KR = TMeanY1_KR ;
rename SE_MEANS_KR = SE_TMeanY1_KR ;
rename CIL_MEANS_KR = CIL_TMEANY1_KR ;
rename CIU_MEANS_KR = CUL_TMEANY1_KR ;
DROP PARM Y ;
RUN;

Data CMEANY1KR ; Set CMEANY1KR ;

```

```

label EST_MEANS_KR = CMeanY1_KR ;
label SE_MEANS_KR = SE_CMeanY1_KR ;
label CIL_MEANS_KR = CIL_CMEANY1_KR ;
label CIU_MEANS_KR = CUL_CMEANY1_KR ;
rename EST_MEANS_KR = CMeanY1_KR ;
rename SE_MEANS_KR = SE_CMeanY1_KR ;
rename CIL_MEANS_KR = CIL_CMEANY1_KR ;
rename CIU_MEANS_KR = CUL_CMEANY1_KR ;
DROP PARM Y ;
RUN;

```

```

Data TXMEANY2KR ; Set TXMEANY2KR ;
label EST_MEANS_KR = TMeanY2_KR ;
label SE_MEANS_KR = SE_TMeanY2_KR ;
label CIL_MEANS_KR = CIL_TMEANY2_KR ;
label CIU_MEANS_KR = CUL_TMEANY2_KR ;
rename EST_MEANS_KR = TMeanY2_KR ;
rename SE_MEANS_KR = SE_TMeanY2_KR ;
rename CIL_MEANS_KR = CIL_TMEANY2_KR ;
rename CIU_MEANS_KR = CUL_TMEANY2_KR ;
DROP PARM Y ;
RUN;

```

```

Data CMEANY2KR ; Set CMEANY2KR ;
label EST_MEANS_KR = CMeanY2_KR ;
label SE_MEANS_KR = SE_CMeanY2_KR ;
label CIL_MEANS_KR = CIL_CMEANY2_KR ;
label CIU_MEANS_KR = CUL_CMEANY2_KR ;
rename EST_MEANS_KR = CMeanY2_KR ;
rename SE_MEANS_KR = SE_CMeanY2_KR ;
rename CIL_MEANS_KR = CIL_CMEANY2_KR ;
rename CIU_MEANS_KR = CUL_CMEANY2_KR ;
DROP PARM Y ;
RUN;

```

```

DATA outputc ; set outputc ;
Merge MVMM_Conv_KR TXMEANY1KR CMEANY1KR TXMEANY2KR CMEANY2KR ;
By Reps;
Run;

```

```

Data MVMM_Fixed_KR ; retain Reps Parm; set MVMM_Fixed_KR ;
label index = Y_KR ;
label Estimate = Est_MVMM_Fixed_KR ;
label StdErr = SE_MVMM_Fixed_KR ;
label Probt = P_MVMM_Fixed_KR ;
label Lower = CIL_MVMM_Fixed_KR ;
label Upper = CIU_MVMM_Fixed_KR ;
rename index = Y_KR ;
rename Estimate = Est_MVMM_Fixed_KR ;
rename StdErr = SE_MVMM_Fixed_KR ;
rename Probt = P_MVMM_Fixed_KR ;
rename Lower = CIL_MVMM_Fixed_KR ;
rename Upper = CIU_MVMM_Fixed_KR ;
drop Alpha;
if INDEX = 1 then do;
    if Effect = 'INDEX' then Parm = 'INT(Y1KR)';

```



```

        if NewT = 0 then Parm = 'd(Y1KR)';
end;
if INDEX = 2 then do;
    if Effect = 'INDEX' then Parm = 'INT(Y2KR)';
    if NewT = 0 then Parm = 'd(Y2KR)';
end;
run;

/** Creating Separate Data Files: Intercepts **/

Data INTY1_KR INTY2_KR ; set MVMM_Fixed_KR ;
If EFFECT = 'INDEX' & Y_KR = 1 then output INTY1_KR ;
Else if EFFECT = 'INDEX' & Y_KR = 2 then output INTY2_KR ;
Run;

```

```

Data INTY1_KR ; set INTY1_KR ;
label Est_MVMM_Fixed_KR = INTY1_KR ;
label SE_MVMM_Fixed_KR = SE_INTY1_KR ;
label P_MVMM_Fixed_KR = P_INTY1_KR ;
label CIL_MVMM_Fixed_KR = CIL_INTY1_KR ;
label CIU_MVMM_Fixed_KR = CIU_INTY1_KR ;
label tvalue = t_INTY1_KR ;
rename Est_MVMM_Fixed_KR = INTY1_KR ;
rename SE_MVMM_Fixed_KR = SE_INTY1_KR ;
rename P_MVMM_Fixed_KR = P_INTY1_KR ;
rename CIL_MVMM_Fixed_KR = CIL_INTY1_KR ;
rename CIU_MVMM_Fixed_KR = CIU_INTY1_KR ;
rename tvalue = t_INTY1_KR ;
DROP PARM EFFECT Y_KR NewT df;
RUN;

```

```

Data INTY2_KR ; set INTY2_KR ;
label Est_MVMM_Fixed_KR = INTY2_KR ;
label SE_MVMM_Fixed_KR = SE_INTY2_KR ;
label P_MVMM_Fixed_KR = P_INTY2_KR ;
label CIL_MVMM_Fixed_KR = CIL_INTY2_KR ;
label CIU_MVMM_Fixed_KR = CIU_INTY2_KR ;
label tvalue = t_INTY2_KR ;
rename Est_MVMM_Fixed_KR = INTY2_KR ;
rename SE_MVMM_Fixed_KR = SE_INTY2_KR ;
rename P_MVMM_Fixed_KR = P_INTY2_KR ;
rename CIL_MVMM_Fixed_KR = CIL_INTY2_KR ;
rename CIU_MVMM_Fixed_KR = CIU_INTY2_KR ;
rename tvalue = t_INTY2_KR ;
DROP PARM EFFECT Y_KR NewT ;
RUN;

```

```

/** Creating Separate Data Files: Slopes **/

Data DY1_KR DY2_KR ; set MVMM_Fixed_KR ;
If Y_KR = 1 & NewT = 0 then output DY1_KR ;
Else if Y_KR = 2 & NewT = 0 then output DY2_KR ;
Run;

```

```

Data DY1_KR ; set DY1_KR ;
label Est_MVMM_Fixed_KR = DY1_KR ;
label SE_MVMM_Fixed_KR = SE_DY1_KR ;
label P_MVMM_Fixed_KR = P_DY1_KR ;
label CIL_MVMM_Fixed_KR = CIL_DY1_KR ;
label CIU_MVMM_Fixed_KR = CIU_DY1_KR ;
label tvalue = t_DY1_KR ;
label df = df_DY1_KR ;
rename Est_MVMM_Fixed_KR = DY1_KR ;
rename SE_MVMM_Fixed_KR = SE_DY1_KR ;
rename P_MVMM_Fixed_KR = P_DY1_KR ;
rename CIL_MVMM_Fixed_KR = CIL_DY1_KR ;
rename CIU_MVMM_Fixed_KR = CIU_DY1_KR ;
rename tvalue = t_DY1_KR ;
rename df = df_DY1_KR ;
DROP PARM EFFECT Y_KR ;
RUN;

```

```

Data DY2_KR ; set DY2_KR ;
label Est_MVMM_Fixed_KR = DY2_KR ;
label SE_MVMM_Fixed_KR = SE_DY2_KR ;
label P_MVMM_Fixed_KR = P_DY2_KR ;
label CIL_MVMM_Fixed_KR = CIL_DY2_KR ;
label CIU_MVMM_Fixed_KR = CIU_DY2_KR ;
label tvalue = t_DY2_KR ;
label df = df_DY2_KR ;
rename Est_MVMM_Fixed_KR = DY2_KR ;
rename SE_MVMM_Fixed_KR = SE_DY2_KR ;
rename P_MVMM_Fixed_KR = P_DY2_KR ;
rename CIL_MVMM_Fixed_KR = CIL_DY2_KR ;
rename CIU_MVMM_Fixed_KR = CIU_DY2_KR ;
rename tvalue = t_DY2_KR ;
rename df = df_DY2_KR ;
DROP PARM EFFECT Y_KR ;
RUN;

```

```

Data outputc ; set outputc ;
Merge INTY1_KR INTY2_KR DY1_KR DY2_KR ;
BY REPS;
Run;

```

```

Data outputc ; set outputc ;
if P_DY1_KR <= .05 then SigDY1_KR = 1;
else SigDY1_KR = 0;
if P_DY2_KR <= .05 then SigDY2_KR = 1;
else SigDY2_KR = 0;
if P_DY1_KR <= .025 or P_DY2_KR <= .025 then OVERALL_KR = 1;
Else OVERALL_KR = 0;
run;

```

```

/* Random Effects */
Data VarY1_KR CoVar12_KR VarY2_KR ; Set Mvmm_random_KR ;
If covparm = 'UN(1,1)' then output VarY1_KR ;
If covparm = 'UN(2,1)' then output CoVar12_KR ;
If covparm = 'UN(2,2)' then output VarY2_KR ;

```

```
run;
```

```
Data VarY1_KR ; set VarY1_KR ;  
label Estimate = VarY1_KR ;  
label StdErr = SE_VarY1_KR ;  
label Probz = P_VarY1_KR ;  
rename Estimate = VarY1_KR ;  
rename StdErr = SE_VarY1_KR ;  
rename Probz = P_VarY1_KR ;  
Drop Subject Covparm ZValue;  
Run;
```

```
Data VarY2_KR ; set VarY2_KR ;  
label Estimate = VarY2_KR ;  
label StdErr = SE_VarY2_KR ;  
label Probz = P_VarY2_KR ;  
rename Estimate = VarY2_KR ;  
rename StdErr = SE_VarY2_KR ;  
rename Probz = P_VarY2_KR ;  
Drop Subject Covparm ZValue;  
Run;
```

```
Data CoVar12_KR ; set CoVar12_KR ;  
label Estimate = CoVar12_KR ;  
label StdErr = SE_CoVar12_KR ;  
label Probz = P_CoVar12_KR ;  
rename Estimate = CoVar12_KR ;  
rename StdErr = SE_CoVar12_KR ;  
rename Probz = P_CoVar12_KR ;  
Drop Subject Covparm ZValue;  
Run;
```

```
Data lineconds;  
var1 = &var1;  
var2 = &var2;  
r12 = &r12;  
esy1 = &esy1;  
esy2 = &esy2;  
N = &N;  
run;
```

```
Data newlineconds (drop = i ); set lineconds;  
do i = 1 to &reps;  
output;  
end;  
run;
```

```
Data outputc ; Set outputc ;  
Merge VarY1_KR VarY2_KR CoVar12_KR newlineconds;  
run;
```

```
/* Forming Relative Bias Measures */
```

```

DATA outputc ; set outputc ;
P_Rel_Bias_MSEY1 = (MSEY1 - var1)/var1;
P_Rel_Bias_MSEY2 = (MSEY2 - var2)/var2;
P_Rel_Bias_COVAR12_GLM = (covar12 - r12)/r12;
P_Rel_Bias_VarY1_ML = (VarY1_ML - var1)/var1;
P_Rel_Bias_VarY2_ML = (VarY2_ML - var2)/var2;
P_Rel_Bias_COVAR12_ML = (CoVar12_ML - r12)/r12;
P_Rel_Bias_VarY1_KR = (VarY1_KR - var1)/var1;
P_Rel_Bias_VarY2_KR = (VarY2_KR - var2)/var2;
P_Rel_Bias_CoVar12_KR = (CoVar12_KR - r12)/r12;
P_Rel_Bias_TMeanY1_OLS = (TmeanY1 - EsY1)/EsY1;
P_Rel_Bias_TMeanY2_OLS = (TMeanY2 - EsY2)/EsY2;
P_Rel_Bias_TMeanY1_ML = (TMeanY1_ML - EsY1)/EsY1;
P_Rel_Bias_TMeanY2_ML = (TMeanY2_ML - EsY2)/EsY2;
P_Rel_Bias_TMeanY1_def = (TMeanY1_def - EsY1)/EsY1;
P_Rel_Bias_TMeanY2_def = (TMeanY2_def - EsY2)/EsY2;
P_Rel_Bias_TMeanY1_KR = (TMeanY1_KR - EsY1)/EsY1;
P_Rel_Bias_TMeanY2_KR = (TMeanY2_KR - EsY2)/EsY2;
P_Rel_Bias_OLSDiffY1 = (DiffY1 - EsY1)/EsY1;
P_Rel_Bias_OLSDiffY2 = (DiffY2 - EsY2)/EsY2;
P_Rel_Bias_MLDiffY1 = (DY1_ML - EsY1)/EsY1;
P_Rel_Bias_MLDiffY2 = (DY2_ML - EsY2)/EsY2;
P_Rel_Bias_DEFDiffY1 = (DY1_def - EsY1)/EsY1;
P_Rel_Bias_DEFDiffY2 = (DY2_def - EsY2)/EsY2;
P_Rel_Bias_KRDiffY1 = (dy1_KR - EsY1)/EsY1;
P_Rel_Bias_KRDiffY2 = (dy2_KR - EsY2)/EsY2;
Run;

```

```

/* Creating CI coverage variables */

```

```

DATA outputc ; SET outputc ;
if CCILowerMeanY1 <=0 and CCIUpperMeanY1 >= 0 then OLS_Capture_CONMEANY1 =
1;
else OLS_Capture_CONMEANY1 = 0;
if CCILowerMeanY2 <=0 and CCIUpperMeanY2 >= 0 then OLS_Capture_CONMEANY2 =
1;
else OLS_Capture_CONMEANY2 = 0;
if TCILowerMeanY1 <=EsY1 and TCIUpperMeanY1 >= EsY1 then
OLS_Capture_TMEANY1 = 1;
else OLS_Capture_TMEANY1 = 0;
if TCILowerMeanY2 <=EsY2 and TCIUpperMeanY2 >= EsY1 then
OLS_Capture_TMEANY2 = 1;
else OLS_Capture_TMEANY2 = 0;
if lowerCLY1 <=EsY1 and upperCLY1 >= EsY1 then OLS_Capture_DIFFY1 = 1;
else OLS_Capture_DIFFY1 = 0;
if lowerCLY2 <=EsY2 and upperCLY2 >= EsY2 then OLS_Capture_DIFFY2 = 1;
else OLS_Capture_DIFFY2 = 0;
if CIL_DY1_ML <=EsY1 and CIU_DY1_ML >= EsY1 then ML_Capture_DIFFY1 = 1;
else ML_Capture_DIFFY1 = 0;
if CIL_DY2_ML <=EsY2 and CIU_DY2_ML >= EsY2 then ML_Capture_DIFFY2 = 1;
else ML_Capture_DIFFY2 = 0;
if CIL_DY1_DEF <=EsY1 and CIU_DY1_DEF >= EsY1 then DEF_Capture_DIFFY1 =
1;
else DEF_Capture_DIFFY1 = 0;
if CIL_DY2_DEF <=EsY2 and CIU_DY2_DEF >= EsY2 then DEF_Capture_DIFFY2 = 1;

```

```

else DEF_Capture_DIFFY2 = 0;
if CIL_DY1_KR <= EsY1 and CIU_DY1_KR >= EsY1 then KR_Capture_DIFFY1 = 1;
else KR_Capture_DIFFY1 = 0;
if CIL_DY2_KR <= EsY2 and CIU_DY2_KR >= EsY2 then KR_Capture_DIFFY2 = 1;
else KR_Capture_DIFFY2 = 0;
run;

```

```

/*****
*****

```

ANALYSIS for 15% INCOMPLETE DATA

```

*****
*****/

```

```

/***** ANALYSIS *****/
/* Traditional MANOVA */

```

```

PROC GLM DATA = alldata;
ods output OverallANOVA=ALLAnova_15;
ods output errorSSCP= eSSCP_15;
ods output partialcorr = corrs_15;
ods output Multstat = WILKS_15;
ods output cldiffs = cldiff_15;
ods output lsmeans = means_15;
ods output LSMeanCL= clmeans_15;
ods output means = descriptives_15;
BY Cond reps;
CLASS T;
MODEL Y1_15 Y2_15 = T;
MANOVA H = T / PRINTE;
MEANS T/ ALPHA = .05 CLDIFF LSD;
lsmeans T/ CL tdiff;
means T;
RUN;
QUIT;

```

```

data wilks_15 (keep = cond reps singular_15);
set wilks_15;
by cond reps;
firstrep = first.reps;
if firstrep NE 1 then delete;
if numdf < 3 then singular_15 = 0;
else singular_15 = 1;
run;

```

```
/* Obtaining listwise deleted treatment and control means for Y1 and Y2,  
along with CI for each group mean */
```

```
data clmeansy1_15 clmeansy2_15; set clmeans_15;  
if dependent = 'Y1_15' then output CLmeansY1_15;  
else if dependent = 'Y2_15' then output CLmeansY2_15;  
run;
```

```
Data TCLMeanY1_15 CCLmeanY1_15;  
set clmeansY1_15;  
if T = 1 then output TCLmeanY1_15;  
else if T = 0 then output CCLmeanY1_15;  
drop effect dependent T;  
run;
```

```
Data TCLMeanY1_15;  
Set TCLMeanY1_15;  
label Lowercl = TCILowerMeanY1_15;  
label lsmean = TMeanY1_15;  
label Uppercl = TCIUpperMeanY1_15;  
rename Lowercl = TCILowerMeanY1_15;  
rename lsmean = TMeanY1_15;  
rename Uppercl = TCIUpperMeanY1_15;  
run;
```

```
Data CCLMeanY1_15;  
Set CCLMeanY1_15;  
label Lowercl = CCILowerMeanY1_15;  
label lsmean = CMeanY1_15;  
label Uppercl = CCIUpperMeanY1_15;  
rename Lowercl = CCILowerMeanY1_15;  
rename lsmean = CMeanY1_15;  
rename Uppercl = CCIUpperMeanY1_15;  
run;
```

```
Data TCLMeanY2_15 CCLmeanY2_15;  
set clmeansY2_15;  
if T = 1 then output TCLmeanY2_15;  
else if T = 0 then output CCLmeanY2_15;  
Drop effect dependent T;  
run;
```

```
Data TCLMeanY2_15;  
Set TCLMeanY2_15;  
label Lowercl = TCILowerMeanY2_15;  
label lsmean = TMeanY2_15;  
label Uppercl = TCIUpperMeanY2_15;  
rename Lowercl = TCILowerMeanY2_15;  
rename lsmean = TMeanY2_15;  
rename Uppercl = TCIUpperMeanY2_15;  
run;
```

```
Data CCLMeanY2_15;  
Set CCLMeanY2_15;  
label Lowercl = CCILowerMeanY2_15;  
label lsmean = CMeanY2_15;
```

```

label Uppercl = CCIUpperMeanY2_15;
rename Lowercl = CCILowerMeanY2_15;
rename lsmean = CMeanY2_15;
rename Uppercl = CCIUpperMeanY2_15;
run;

Data output_15;
merge TclmeanY1_15 CCLMeanY1_15 TCLMeanY2_15 CCLmeanY2_15 ;
by cond reps;
run;

/* Creating separate data sets for ANOVAs of Y1 and Y2 */
/* Splits ANOVA Results into Y1 and Y2 data sets */

Data ANOVAY1_15 ANOVAY2_15;
set aLLANOVA_15;
if dependent = 'Y1_15' then output ANOVAY1_15;
else if dependent = 'Y2_15' then output ANOVAY2_15;
run;

/* F test for Y1_15 */
DATA FtestY1_15; keep cond reps dependent FValue ProbF; set ANOVAY1_15;
if Source = 'Model' then output;
label ProbF = p-value FY1_15;
rename probF = PFY1_15;
label FValue = FY1_15;
rename FValue = FY1_15;
Run;

/* F test for Y2 */
DATA FtestY2_15; keep cond reps dependent Fvalue ProbF; set ANOVAY2_15;
if Source = 'Model' then output;
label ProbF = p-value FY2_15;
rename probF = PFY2_15;
label FValue = FY2_15;
rename FValue = FY2_15;
Run;

Data output_15; set output_15;
merge wilks_15 FtestY1_15 FtestY2_15 ;
by cond reps;
drop dependent;
if PFY1_15 <= .05 then Sig_FY1_15 = 1; else Sig_FY1_15 = 0;
if PFY2_15 <= .05 then Sig_FY2_15 = 1; else Sig_FY2_15 = 0;
If PFY1_15 <= .025 or PFY2_15 <= .025 then Sig_SUTS_15 = 1; else Sig_SUTS_15
= 0;
run;

/* Mean Square Error for Y1 */
DATA MSEY1_15; keep cond reps MS; set ANOVAY1_15;
if source = 'Error' then output;
label MS = MSEY1_15;
rename MS = MSEY1_15;
Run;

/* Mean Square Error for Y2 */
DATA MSEY2_15; keep cond reps MS; set ANOVAY2_15;

```

```

if source = 'Error' then output;
label MS = MSEY2_15;
rename MS = MSEY2_15;
Run;

Data output_15;
set output_15;
merge MSEY1_15 MSEY2_15;
by cond reps;
run;

/* Creating Confidence Limits Data Set */
DATA CL_15; SET Cldiff_15;
if Comparison = '1 - 0' then output;
run;

Data CLY1_15 CLY2_15 ;
set CL_15;
if dependent = 'Y1_15' then output CLY1_15;
else if dependent = 'Y2_15' then output CLY2_15;
drop method comparison Effect Dependent;
run;

Data CLY1_15;
set CLY1_15;
label lowerCL= lowerCLY1_15;
rename lowerCL= lowerCLY1_15;
label upperCl = upperCLY1_15;
rename upperCl = upperCLY1_15;
label Difference = DiffY1_15;
rename Difference = DiffY1_15;
label Significance = sigY1_15;
rename Significance = sigY1_15;
Run;

Data CLY2_15;
set CLY2_15;
label lowerCL= lowerCLY2_15;
rename lowerCL= lowerCLY2_15;
label upperCl = upperCLY2_15;
rename upperCl = upperCLY2_15;
label Difference = DiffY2_15;
rename Difference = DiffY2_15;
label Significance = sigY2_15;
rename Significance = sigY2_15;
Run;

Data output_15;
set output_15;
merge CLY1_15 CLY2_15;
by cond Reps;
Run;

/* Keeping Pooled Correlations */

Data NewCorrs1_15; Set Corrs_15;
by cond reps;

```



```

firstrep = first.reps;
if firstrep NE 1 then delete;
keep cond reps Y2_15 P_Y2_15 ;
label Y2_15 = Corr12_15;
rename Y2_15 = Corr12_15;
label P_Y2_15 = p-value CORR12_15;
Rename P_Y2_15 = P_Val_CORR12_15;
DROP Y1_15 Variable P_y1_15;
run;

Data output_15;
set output_15;
merge NewCorrs1_15 ;
by cond Reps;
Run;

/* Obtain group Ns and SDs below */

Data Tinfo_15 Cinfo_15; Set Descriptives_15;
By reps;
if T = 0 then output Cinfo_15;
else if T = 1 then output Tinfo_15;
run;

Data Tinfo_15; Set Tinfo_15;
rename N = TreatN_15;
label N = TreatN_15;
rename SD_Y1_15 = TSD_Y1_15;
label SD_Y1_15 = TSD_Y1_15;
rename SD_Y2_15 = TSD_Y2_15;
label SD_Y2_15 = TSD_Y2_15;
Drop Mean_Y1_15 Mean_Y2_15 Effect T;
run;

Data Cinfo_15; Set Cinfo_15;
rename N = ControlN_15;
label N = ControlN_15;
rename SD_Y1_15 = CSD_Y1_15;
label SD_Y1_15 = CSD_Y1_15;
rename SD_Y2_15 = CSD_Y2_15;
label SD_Y2_15 = CSD_Y2_15;
Drop Mean_Y1_15 Mean_Y2_15 Effect T;
run;

Data output_15;
set output_15;
merge Tinfo_15 Cinfo_15;
By cond reps;
N_both_15 = TreatN_15+ControlN_15;
TVar_Y1_15 = sqrt(TSD_Y1_15);
TVar_Y2_15 = sqrt(TSD_Y2_15);
CVar_Y1_15 = sqrt(CSD_Y1_15);
CVar_Y2_15 = sqrt(CSD_Y2_15);
Covar12_15 = corr12_15/sqrt(msey1_15)*sqrt(msey2_15);
Run;

```

```

/* Obtaining t tests for Y1 Y2 */
Data ttestY1_15 ttestY2_15 ;
set means_15 ;
if dependent = 'Y1_15 ' then do;
    if T = 0 then output ttestY1_15 ;
end;
else if dependent = 'Y2_15 ' then do;
    if T = 0 then output ttestY2_15 ;
end;
Drop lsmean effect dependent T probtdiff;
run;

Data ttestY1_15 ; Set ttesty1_15 ;
label tdiff = ttestY1_glm_15 ;
rename tdiff = ttestY1_glm_15 ;
Run;

Data ttestY2_15 ; Set ttesty2_15 ;
label tdiff = ttesty2_glm_15 ;
rename tdiff = ttesty2_glm_15 ;
run;

Data output_15 ; set output_15;
merge ttesty1_15 ttesty2_15 ;
By Cond reps;
Run;

Data output_15;
Set output_15;
SEdiffY1_glm_15 = abs(diffy1_15)/abs(TtestY1_glm_15);
SEdiffY2_glm_15 = abs(diffy2_15)/abs(TtestY2_glm_15);
run;

/* ANOVA FOR Y2 USING PAIRWISE DELETION: HERE, ALL CASES USED FOR Y2
Note that ANOVA for Y1 is obtained with the MANOVA procedure above */

PROC GLM DATA = alldata;
ods output OverallANOVA=Y2AnovaPAIR_15;
ods output cldiffs = Y2cldiffPAIR_15;
ods output Means = Y2PAIR_15;
ods output lsmeans = means_pair_15;
ods output LSMeanCL = Y2clmeanspair_15;
By cond reps;
CLASS T;
MODEL Y2_15 = T;
MEANS T/ ALPHA = .05 CLDIFF LSD;
lsmeans T/ CL tdiff;
means T;
RUN;
QUIT;

/* Obtaining pairwise deleted treatment and control means for Y2, along with
CI for each group mean */

Data TCLMeanY2pair_15 CCLmeanY2pair_15;
set Y2clmeanspair_15;

```

```

if T = 1 then output TCLmeanY2pair_15;
else if T = 0 then output CCLmeanY2pair_15;
run;

```

```

Data TCLMeanY2pair_15;
Set TCLMeanY2pair_15;
label Lowercl = TCILowerMeanY2pair_15;
label lsmean = TMeanY2pair_15;
label Uppercl = TCIOpperMeanY2pair_15;
rename Lowercl = TCILowerMeanY2pair_15;
rename lsmean = TMeanY2pair_15;
rename Uppercl = TCIOpperMeanY2pair_15;
Drop effect dependent T;
run;

```

```

Data CCLMeanY2pair_15;
Set CCLMeanY2pair_15;
label Lowercl = CCILowerMeanY2pair_15;
label lsmean = CMeanY2pair_15;
label Uppercl = CCIUpperMeanY2pair_15;
rename Lowercl = CCILowerMeanY2pair_15;
rename lsmean = CMeanY2pair_15;
rename Uppercl = CCIUpperMeanY2pair_15;
Drop effect dependent T;
run;

```

```

Data output_15;
set output_15;
merge TCLMeanY2pair_15 CCLmeanY2Pair_15;
by cond reps;
run;

```

```

/* F test for Y2: PAIRWISE DELETION */
DATA FtestY2pair_15; keep cond reps Fvalue ProbF; set Y2ANOVAPAIR_15;
if Source = 'Model' then output;
label ProbF = p-value FY2PAIR_15;
rename probF = PFY2_PAIR_15;
label FValue = FY2PAIR_15;
rename FValue = FY2_PAIR_15;
Run;

```

```

/* Mean Square Error for Y2 PAIRWISE DELETION: ALL CASES */
DATA MSEY2_PAIR_15; keep cond reps MS; set Y2ANOVAPAIR_15;
if source = 'Error' then output;
label MS = MSEY2_PAIR_15;
rename MS = MSEY2_PAIR_15;
Run;

```

```

/* Obtaining CL limits for Y2 PAIRWISE DELETION */
DATA CL_y2_pair_15; SET Y2CLDIFFPAIR_15;
if Comparison = '1 - 0' then output;
label lowerCL= lowerCLY2_PAIR_15;
rename lowerCL= lowerCLY2_PAIR_15;
label upperCl = upperCLY2_PAIR_15;
rename upperCl = upperCLY2_PAIR_15;
label difference = diff_Y2_PAIR_15;

```

```

rename difference = diff_Y2_PAIR_15;
drop effect dependent comparison method significance;
run;

```

```

Data Cinfo_Y2PAIR_15 Tinfo_Y2Pair_15; set y2pair_15;
By cond reps;
if T = 0 then output Cinfo_Y2PAIR_15;
else if T = 1 then output Tinfo_Y2PAIR_15;
run;

```

```

Data Tinfo_Y2Pair_15; Set Tinfo_Y2Pair_15;
rename N = TreatN_Y2PAIR_15;
label N = TreatN_Y2PAIR_15;
rename SD_Y2_15 = TSD_Y2_PAIR_15;
label SD_Y2_15 = TSD_Y2_PAIR_15;
rename Mean_Y2_15 = TMean_Y2_PAIR_15;
label Mean_Y2_15 = TMean_Y2_PAIR_15;
Drop Effect T;
run;

```

```

Data Cinfo_Y2Pair_15; Set Cinfo_Y2Pair_15;
rename N = ControlN_Y2Pair_15;
label N = ControlN_Y2Pair_15;
rename SD_Y2_15 = CSD_Y2_Pair_15;
label SD_Y2_15 = CSD_Y2_Pair_15;
rename Mean_Y2_15 = CMean_Y2_PAIR_15;
label Mean_Y2_15 = CMean_Y2_PAIR_15;
Drop Effect T;
run;

```

```

/* Obtaining t test for Y2 pairwise */
Data ttestY2_pair_15 ;
set means_pair_15 ;
    if T = 0 then output ttestY2_pair_15 ;
Drop lsmean effect dependent T probtdiff;
    label tdiff = ttesty2_glm_pair_15 ;
    rename tdiff = ttesty2_glm_pair_15 ;
run;

```

```

Data output_15 ; set output_15;
merge ttestY2_pair_15 ;
Run;

```

```

Data output_15;
set output_15;
merge TCLMeanY2pair_15 CCLmeanY2Pair_15 FtestY2pair_15 MSEY2_PAIR_15
CL_Y2_PAIR_15 Tinfo_Y2Pair_15 Cinfo_Y2Pair_15;
by cond reps;
if PFY2_PAIR_15 <= .05 then Sig_FY2_PAIR_15 = 1; else Sig_FY2_PAIR_15 = 0;
SEDiffY2_PAIR_15 = abs(diff_Y2_PAIR_15)/abs(TtestY2_glm_Pair_15);
If PFY1_15 <= .025 or PFY2_PAIR_15 <= .025 then Sig_SUTS_PAIR_15 = 1; else
Sig_SUTs_PAIR_15 = 0;
run;

```

```

/***** CONVERTING WIDE DATA SET TO LONG DATA SET *****/
/***** Wide Data Set *****/
data Wide_15; set alldata;
run;

/***** Wide to Long *****/
data Long_15; set Wide_15;
INDEX=1; Y_15=Y1_15; R_15=R1; output;
INDEX=2; Y_15=Y2_15; R_15=R2; output;
Keep reps ID INDEX T NewT Y_15 R_15;
run;

/*****
*****

Linear Mixed Model with FIML and Z Tests for Effects of Y1 and Y2: 15%
Missingness

*****
*****/

/* Syntax for FIML and Default Between-Within df
Z Tests for effects of Y1 and Y2 obtained below */

PROC MIXED DATA=long_15 METHOD=ML NOCLPRINT COVTEST;
  BY reps;
  CLASS index ID;
  MODEL Y_15 = index index*T/ CL NOINT SOLUTION;
  REPEATED index / SUBJECT = ID TYPE=UN R;
  ods output ConvergenceStatus = MVMM_Conv_FullML_15;
  ods output FitStatistics = MVMM_Fit_FullML_15;
  ods output CovParms = MVMM_Random_FullML_15;
  ods output SolutionF = MVMM_Fixed_FullML_15;
  RUN;

/** convergence for Full Model With ML**/
data MVMM_Conv_FullML_15; set MVMM_Conv_FullML_15;
drop Reason pdG;
label Status = Status_MVMM_FullML_15;
rename Status = Status_MVMM_FullML_15;
label pdH = R_MVMM_FullML_15;
rename pdH = R_MVMM_FullML_15;
run;

proc freq data=MVMM_Conv_FullML_15 noprint;
tables Reps*Status_MVMM_FullML_15*R_MVMM_FullML_15/ out=MVMM_Conv_FullML_15;
run;

```

```

Data MVMM_Conv_FullML_15; set MVMM_Conv_FullML_15;
label count = Conv_MVMM_FullML_Count_15;
rename count = Conv_MVMM_FullML_Count_15;
label percent = Conv_MVMM_FullML_Percent_15;
rename percent = Conv_MVMM_FullML_Percent_15;
run;

Data output_15; set output_15;
Merge MVMM_Conv_FullML_15;
BY REPS;
Run;

/** Fixed Effects From Full Model With ML */
data MVMM_Fixed_FullML_15; retain Repls Parm; set MVMM_Fixed_FullML_15;
label index = Y_15;
label Estimate = Est_MVMM_Fixed_FullML_15;
label StdErr = SE_MVMM_Fixed_FullML_15;
label Probt = P_MVMM_Fixed_FullML_15;
label Lower = CIL_MVMM_Fixed_FullML_15;
label Upper = CIU_MVMM_Fixed_FullML_15;
rename index = Y_15;
rename Estimate = Est_MVMM_Fixed_FullML_15;
rename StdErr = SE_MVMM_Fixed_FullML_15;
rename Probt = P_MVMM_Fixed_FullML_15;
rename Lower = CIL_MVMM_Fixed_FullML_15;
rename Upper = CIU_MVMM_Fixed_FullML_15;
drop Alpha;
if INDEX = 1 then do;
    if Effect = 'INDEX' then Parm = 'INT(Y1)_15';
    if Effect = 'T*INDEX' then Parm = 'd(Y1)_15';
end;
if INDEX = 2 then do;
    if Effect = 'INDEX' then Parm = 'INT(Y2)_15';
    if Effect = 'T*INDEX' then Parm = 'd(Y2)_15';
end;
run;

/** Creating Separate Data Files: Intercepts */
data INTY1_ML_15 INTY2_ML_15 ; set MVMM_Fixed_FullML_15;
If EFFECT = 'INDEX' & Y_15 = 1 then output INTY1_ML_15;
Else if EFFECT = 'INDEX' & Y_15 = 2 then output INTY2_ML_15;
Run;

data INTY1_ML_15 ; set INTY1_ML_15;
label Est_MVMM_Fixed_FullML_15 = INTY1_ML_15;
label SE_MVMM_Fixed_FullML_15 = SE_INTY1_ML_15;
label P_MVMM_Fixed_FullML_15 = P_INTY1_ML_15;
label CIL_MVMM_Fixed_FullML_15 = CIL_INTY1_ML_15;
label CIU_MVMM_Fixed_FullML_15 = CIU_INTY1_ML_15;
label tvalue = t_INTY1_ML_15;
rename Est_MVMM_Fixed_FullML_15 = INTY1_ML_15;
rename SE_MVMM_Fixed_FullML_15 = SE_INTY1_ML_15;
rename P_MVMM_Fixed_FullML_15 = P_INTY1_ML_15;
rename CIL_MVMM_Fixed_FullML_15 = CIL_INTY1_ML_15;

```

```

rename CIU_MVMM_Fixed_FullML_15 =CIU_INTY1_ML_15;
rename tvalue = t_INTY1_ML_15;
DROP PARM EFFECT Y_15 df;
RUN;

```

```

data INTY2_ML_15 ; set INTY2_ML_15;
label Est_MVMM_Fixed_FullML_15 = INTY2_ML_15;
label SE_MVMM_Fixed_FullML_15 = SE_INTY2_ML_15;
label P_MVMM_Fixed_FullML_15 = P_INTY2_ML_15;
label CIL_MVMM_Fixed_FullML_15 = CIL_INTY2_ML_15;
label CIU_MVMM_Fixed_FullML_15 = CIU_INTY2_ML_15;
label tvalue = t_INTY2_ML_15;
rename Est_MVMM_Fixed_FullML_15 = INTY2_ML_15;
rename SE_MVMM_Fixed_FullML_15 = SE_INTY2_ML_15;
rename P_MVMM_Fixed_FullML_15 = P_INTY2_ML_15;
rename CIL_MVMM_Fixed_FullML_15 = CIL_INTY2_ML_15;
rename CIU_MVMM_Fixed_FullML_15 =CIU_INTY2_ML_15;
rename tvalue = t_INTY2_ML_15;
DROP PARM EFFECT Y_15 df;
RUN;

```

```

/** Creating Separate Data Files: Slopes */
data DY1_ML_15 DY2_ML_15 ; set MVMM_Fixed_FullML_15;
If EFFECT = 'T*INDEX' & Y_15 = 1 then output DY1_ML_15;
Else if EFFECT = 'T*INDEX' & Y_15 = 2 then output DY2_ML_15;
Run;

```

```

data DY1_ML_15 ; set DY1_ML_15;
label Est_MVMM_Fixed_FullML_15 = DY1_ML_15;
label SE_MVMM_Fixed_FullML_15 = SE_DY1_ML_15;
label P_MVMM_Fixed_FullML_15 = P_DY1_ML_15;
label CIL_MVMM_Fixed_FullML_15 = CIL_DY1_ML_15;
label CIU_MVMM_Fixed_FullML_15 = CIU_DY1_ML_15;
label tvalue = t_DY1_ML_15;
label df = df_DY1_ML_15;
rename Est_MVMM_Fixed_FullML_15 = DY1_ML_15;
rename SE_MVMM_Fixed_FullML_15 = SE_DY1_ML_15;
rename P_MVMM_Fixed_FullML_15 = P_DY1_ML_15;
rename CIL_MVMM_Fixed_FullML_15 = CIL_DY1_ML_15;
rename CIU_MVMM_Fixed_FullML_15 =CIU_DY1_ML_15;
rename tvalue = t_DY1_ML_15;
rename df = df_DY1_ML_15;
DROP PARM EFFECT Y_15;
RUN;

```

```

data DY2_ML_15 ; set DY2_ML_15;
label Est_MVMM_Fixed_FullML_15 = DY2_ML_15;
label SE_MVMM_Fixed_FullML_15 = SE_DY2_ML_15;
label P_MVMM_Fixed_FullML_15 = P_DY2_ML_15;
label CIL_MVMM_Fixed_FullML_15 = CIL_DY2_ML_15;
label CIU_MVMM_Fixed_FullML_15 = CIU_DY2_ML_15;
label tvalue = t_DY2_ML_15;
label df = df_DY2_ML_15;
rename Est_MVMM_Fixed_FullML_15 = DY2_ML_15;
rename SE_MVMM_Fixed_FullML_15 = SE_DY2_ML_15;
rename P_MVMM_Fixed_FullML_15 = P_DY2_ML_15;

```

```

rename CIL_MVMM_Fixed_FullML_15 = CIL_DY2_ML_15;
rename CIU_MVMM_Fixed_FullML_15 = CIU_DY2_ML_15;
rename tvalue = t_DY2_ML_15;
rename df = df_DY2_ML_15;
DROP PARM EFFECT Y_15;
RUN;

Data output_15; set output_15;
Merge INTY1_ML_15 INTY2_ML_15 DY1_ML_15 DY2_ML_15;
BY REPS;
Run;

/* Creating P values for Z tests of Y1 and Y2 */
Data output_15; set output_15;
Merge INTY1_ML_15 INTY2_ML_15 DY1_ML_15 DY2_ML_15 ;
BY REPS;
if t_DY1_ML_15 < 0 then
  do;
    pval_Zy1_15 = 2*cdf("Normal", t_DY1_ML_15);
  end;
else
  do;
    pval_Zy1_15 = 2*(1-cdf("Normal", t_DY1_ML_15));
  end;
if pval_Zy1_15 <= .05 then SigZy1_15 = 1;
else SigZy1_15 = 0;
if t_DY2_ML_15 < 0 then
  do;
    pval_Zy2_15 = 2*cdf("Normal", t_DY2_ML_15);
  end;
else
  do;
    pval_Zy2_15 = 2*(1-cdf("Normal", t_DY2_ML_15));
  end;
if pval_Zy1_15 <= .05 then SigZy1_15 = 1;
else SigZy1_15 = 0;
if pval_Zy2_15 <= .05 then SigZy2_15 = 1;
else SigZy2_15 = 0;
if pval_Zy1_15 <= .025 or pval_Zy2_15 <= .025 then OVERALL_ML_15 = 1;
Else OVERALL_ML_15 = 0;
Run;

Data output_15; set output_15;
TMeanY1_ML_15 = DY1_ML_15 + INTY1_ML_15;
TMeanY2_ML_15 = DY2_ML_15 + INTY2_ML_15;
Run;

/* Random Effects */
Data VarY1_ML_15 CoVar12_ML_15 VarY2_ML_15 ; Set Mvmm_random_fullML_15;
If covparm = 'UN(1,1)' then output VarY1_ML_15;
If covparm = 'UN(2,1)' then output CoVar12_ML_15;
If covparm = 'UN(2,2)' then output VarY2_ML_15;
run;

Data VarY1_ML_15; set VarY1_ML_15;

```



```

label Estimate = VarY1_ML_15;
label StdErr = SE_VarY1_ML_15;
label Probz = P_VarY1_ML_15;
rename Estimate = VarY1_ML_15;
rename StdErr = SE_VarY1_ML_15;
rename Probz = P_VarY1_ML_15;
Drop Subject Covparm ZValue;
Run;

```

```

Data VarY2_ML_15; set VarY2_ML_15;
label Estimate = VarY2_ML_15;
label StdErr = SE_VarY2_ML_15;
label Probz = P_VarY2_ML_15;
rename Estimate = VarY2_ML_15;
rename StdErr = SE_VarY2_ML_15;
rename Probz = P_VarY2_ML_15;
Drop Subject Covparm ZValue;
Run;

```

```

Data CoVar12_ML_15; set CoVar12_ML_15;
label Estimate = CoVar12_ML_15;
label StdErr = SE_CoVar12_ML_15;
label Probz = P_CoVar12_ML_15;
rename Estimate = CoVar12_ML_15;
rename StdErr = SE_CoVar12_ML_15;
rename Probz = P_CoVar12_ML_15;
Drop Subject Covparm ZValue;
Run;

```

```

Data output_15; Set output_15;
Merge VarY1_ML_15 CoVar12_ML_15 VarY2_ML_15 ;
run;

```

```

/*****
****

```

```

Linear Mixed Model with REML and Default Denominator df: 15%
Missingness

```

```

****
****/

```

```

PROC MIXED DATA=long_15 METHOD=REML NOCLPRINT COVTEST;
  BY reps;
  CLASS index ID newT;
  MODEL Y_15 = index newT*Index / NOINT SOLUTION CL;
  lsmeans newT*Index / slice = index cl diff;
  REPEATED index / SUBJECT = ID TYPE=UN R;

```

```

ods output ConvergenceStatus = MVMM_Conv_def_15;
ods output FitStatistics = MVMM_Fit_def_15;
ods output CovParms = MVMM_Random_def_15;
ods output LSMeans = MVMM_Means_def_15;
ods output Diffs = MVMM_Diffs_def_15;
ods output SolutionF = MVMM_Fixed_def_15;
RUN;

/** convergence for Full Model KR**/
data MVMM_Conv_def_15; set MVMM_Conv_def_15;
drop Reason pdG;
label Status = Status_MVMM_def_15;
rename Status = Status_MVMM_def_15;
label pdH = R_MVMM_def_15;
rename pdH = R_MVMM_def_15;
run;

proc freq data = MVMM_Conv_def_15 noprint;
tables Reps*Status_MVMM_def_15*R_MVMM_def_15 / out=MVMM_Conv_def_15;
run;

Data MVMM_Conv_def_15; set MVMM_Conv_def_15;
label count = Conv_MVMM_def_15_Count;
rename count = Conv_MVMM_def_15_Count;
label percent = Conv_MVMM_def_15_Percent;
rename percent = Conv_MVMM_def_15_Percent;
run;

/** Means From Full Model With Def_15 df **/
data MVMM_Means_def_15; retain Reps Parm; set MVMM_Means_def_15;
label index = Y;
label Estimate = Means_def_15;
label StdErr = SE_MEANS_def_15;
label Lower = CIL_MEANS_def_15;
label Upper = CIU_MEANS_def_15;
rename index = Y;
rename Estimate = Est_MEANS_def_15;
rename StdErr = SE_MEANS_def_15;
rename Lower = CIL_MEANS_def_15;
rename Upper = CIU_MEANS_def_15;
drop Effect tvalue Alpha probt df;
if INDEX = 1 then do;
    if newT = 0 then Parm = 'TMEAN(Y1def_15)';
    if newT = 1 then Parm = 'CMEAN(Y1def_15)';
end;
if INDEX = 2 then do;
    if newT = 0 then Parm = 'TMEAN(Y2def_15)';
    if newT = 1 then Parm = 'CMEAN(Y2def_15)';
end;
run;

/** Creating Separate Data Files: Means **/
data CMEANY1def_15 TXMEANY1def_15 CMEANY2def_15 TXMEANY2def_15 ; set
MVMM_Means_def_15;
If Parm = 'TMEAN(Y1def_15)' then output TXMEANY1def_15;
If Parm = 'CMEAN(Y1def_15)' then output CMEANY1def_15;

```

```
If Parm = 'TMEAN(Y2def_15)' then output TXMEANY2def_15;  
If Parm = 'CMEAN(Y2def_15)' then output CMEANY2def_15;  
Run;
```

```
Data TXMEANY1def_15; Set TXMEANY1def_15 ;  
label EST_MEANS_def_15 = TMeanY1_def_15;  
label SE_MEANS_def_15 = SE_TMeanY1_def_15;  
label CIL_MEANS_def_15 = CIL_TMEANY1_def_15;  
label CIU_MEANS_def_15 = CUL_TMEANY1_def_15;  
rename EST_MEANS_def_15 = TMeanY1_def_15;  
rename SE_MEANS_def_15 = SE_TMeanY1_def_15;  
rename CIL_MEANS_def_15 = CIL_TMEANY1_def_15;  
rename CIU_MEANS_def_15 = CUL_TMEANY1_def_15;  
DROP PARM Y;  
RUN;
```

```
Data CMEANY1def_15; Set CMEANY1def_15 ;  
label EST_MEANS_def_15 = CMeanY1_def_15;  
label SE_MEANS_def_15 = SE_CMeanY1_def_15;  
label CIL_MEANS_def_15 = CIL_CMEANY1_def_15;  
label CIU_MEANS_def_15 = CUL_CMEANY1_def_15;  
rename EST_MEANS_def_15 = CMeanY1_def_15;  
rename SE_MEANS_def_15 = SE_CMeanY1_def_15;  
rename CIL_MEANS_def_15 = CIL_CMEANY1_def_15;  
rename CIU_MEANS_def_15 = CUL_CMEANY1_def_15;  
DROP PARM Y;  
RUN;
```

```
Data TXMEANY2def_15; Set TXMEANY2def_15 ;  
label EST_MEANS_def_15 = TMeanY2_def_15;  
label SE_MEANS_def_15 = SE_TMeanY2_def_15;  
label CIL_MEANS_def_15 = CIL_TMEANY2_def_15;  
label CIU_MEANS_def_15 = CUL_TMEANY2_def_15;  
rename EST_MEANS_def_15 = TMeanY2_def_15;  
rename SE_MEANS_def_15 = SE_TMeanY2_def_15;  
rename CIL_MEANS_def_15 = CIL_TMEANY2_def_15;  
rename CIU_MEANS_def_15 = CUL_TMEANY2_def_15;  
DROP PARM Y;  
RUN;
```

```
Data CMEANY2def_15; Set CMEANY2def_15 ;  
label EST_MEANS_def_15 = CMeanY2_def_15;  
label SE_MEANS_def_15 = SE_CMeanY2_def_15;  
label CIL_MEANS_def_15 = CIL_CMEANY2_def_15;  
label CIU_MEANS_def_15 = CUL_CMEANY2_def_15;  
rename EST_MEANS_def_15 = CMeanY2_def_15;  
rename SE_MEANS_def_15 = SE_CMeanY2_def_15;  
rename CIL_MEANS_def_15 = CIL_CMEANY2_def_15;  
rename CIU_MEANS_def_15 = CUL_CMEANY2_def_15;  
DROP PARM Y;  
RUN;
```

```
DATA output_15; set output_15;  
Merge MVMM_Conv_def_15 TXMEANY1def_15 CMEANY1def_15 TXMEANY2def_15  
CMEANY2def_15 ;  
By Reps;
```

```

Run;

/* Fixed Effects From Full Model With def_15 */

data MVMM_Fixed_def_15; retain  Reps Parm; set MVMM_Fixed_def_15;
label index = Y;
label Estimate = Est_MVMM_Fixed_def_15;
label StdErr = SE_MVMM_Fixed_def_15;
label Probt = P_MVMM_Fixed_def_15;
label Lower = CIL_MVMM_Fixed_def_15;
label Upper = CIU_MVMM_Fixed_def_15;
rename index = Y;
rename Estimate = Est_MVMM_Fixed_def_15;
rename StdErr = SE_MVMM_Fixed_def_15;
rename Probt = P_MVMM_Fixed_def_15;
rename Lower = CIL_MVMM_Fixed_def_15;
rename Upper = CIU_MVMM_Fixed_def_15;
drop Alpha;
if INDEX = 1 then do;
    if Effect = 'INDEX' then Parm = 'INT(Y1def_15)';
    if NewT = 0 then Parm = 'd(Y1def_15)';
end;
if INDEX = 2 then do;
    if Effect = 'INDEX' then Parm = 'INT(Y2def_15)';
    if NewT = 0 then Parm = 'd(Y2def_15)';
end;
run;

/** Creating Separate Data Files: Intercepts **/

data INTY1_def_15 INTY2_def_15 ; set MVMM_Fixed_def_15;
If EFFECT = 'INDEX' & Y = 1 then output INTY1_def_15;
Else if EFFECT = 'INDEX' & Y = 2 then output INTY2_def_15;
Run;

data INTY1_def_15 ; set INTY1_def_15;
label Est_MVMM_Fixed_def_15 = INTY1_def_15;
label SE_MVMM_Fixed_def_15 = SE_INTY1_def_15;
label P_MVMM_Fixed_def_15 = P_INTY1_def_15;
label CIL_MVMM_Fixed_def_15 = CIL_INTY1_def_15;
label CIU_MVMM_Fixed_def_15 = CIU_INTY1_def_15;
label tvalue = t_INTY1_def_15;
rename Est_MVMM_Fixed_def_15 = INTY1_def_15;
rename SE_MVMM_Fixed_def_15 = SE_INTY1_def_15;
rename P_MVMM_Fixed_def_15 = P_INTY1_def_15;
rename CIL_MVMM_Fixed_def_15 = CIL_INTY1_def_15;
rename CIU_MVMM_Fixed_def_15 = CIU_INTY1_def_15;
rename tvalue = t_INTY1_def_15;
DROP PARM EFFECT Y NewT df;
RUN;

data INTY2_def_15 ; set INTY2_def_15;
label Est_MVMM_Fixed_def_15 = INTY2_def_15;
label SE_MVMM_Fixed_def_15 = SE_INTY2_def_15;
label P_MVMM_Fixed_def_15 = P_INTY2_def_15;
label CIL_MVMM_Fixed_def_15 = CIL_INTY2_def_15;
label CIU_MVMM_Fixed_def_15 = CIU_INTY2_def_15;

```

```

label tvalue = t_INTY2_def_15;
rename Est_MVMM_Fixed_def_15 = INTY2_def_15;
rename SE_MVMM_Fixed_def_15 = SE_INTY2_def_15;
rename P_MVMM_Fixed_def_15 = P_INTY2_def_15;
rename CIL_MVMM_Fixed_def_15 = CIL_INTY2_def_15;
rename CIU_MVMM_Fixed_def_15 = CIU_INTY2_def_15;
rename tvalue = t_INTY2_def_15;
DROP PARM EFFECT Y Newt ;
RUN;

/** Creating Separate Data Files: Slopes **/

Data DY1_def_15 DY2_def_15 ; set MVMM_Fixed_def_15;
If Y = 1 & NewT = 0 then output DY1_def_15;
Else if Y = 2 & NewT = 0 then output DY2_def_15;
Run;

data DY1_def_15 ; set DY1_def_15;
label Est_MVMM_Fixed_def_15 = DY1_def_15;
label SE_MVMM_Fixed_def_15 = SE_DY1_def_15;
label P_MVMM_Fixed_def_15 = P_DY1_def_15;
label CIL_MVMM_Fixed_def_15 = CIL_DY1_def_15;
label CIU_MVMM_Fixed_def_15 = CIU_DY1_def_15;
label tvalue = t_DY1_ML;
label df = df_DY1_ML;
rename Est_MVMM_Fixed_def_15 = DY1_def_15;
rename SE_MVMM_Fixed_def_15 = SE_DY1_def_15;
rename P_MVMM_Fixed_def_15 = P_DY1_def_15;
rename CIL_MVMM_Fixed_def_15 = CIL_DY1_def_15;
rename CIU_MVMM_Fixed_def_15 = CIU_DY1_def_15;
rename tvalue = t_DY1_def_15;
rename df = df_DY1_def_15;
DROP PARM EFFECT Y;
RUN;

data DY2_def_15 ; set DY2_def_15;
label Est_MVMM_Fixed_def_15 = DY2_def_15;
label SE_MVMM_Fixed_def_15 = SE_DY2_def_15;
label P_MVMM_Fixed_def_15 = P_DY2_def_15;
label CIL_MVMM_Fixed_def_15 = CIL_DY2_def_15;
label CIU_MVMM_Fixed_def_15 = CIU_DY2_def_15;
label tvalue = t_DY2_def_15;
label df = df_DY2_def_15;
rename Est_MVMM_Fixed_def_15 = DY2_def_15;
rename SE_MVMM_Fixed_def_15 = SE_DY2_def_15;
rename P_MVMM_Fixed_def_15 = P_DY2_def_15;
rename CIL_MVMM_Fixed_def_15 = CIL_DY2_def_15;
rename CIU_MVMM_Fixed_def_15 = CIU_DY2_def_15;
rename tvalue = t_DY2_def_15;
rename df = df_DY2_def_15;
DROP PARM EFFECT Y;
RUN;

Data output_15; set output_15;
Merge INTY1_def_15 INTY2_def_15 DY1_def_15 DY2_def_15;

```

```

BY REPS;
Run;

Data output_15; set output_15;
if P_DY1_def_15 <= .05 then SigDY1_def_15 = 1;
else SigDY1_def_15 = 0;
if P_DY2_def_15 <= .05 then SigDY2_def_15 = 1;
else SigDY2_def_15 = 0;
if P_DY1_def_15 <= .025 or P_DY2_def_15 <= .025 then OVERALL_def_15 = 1;
Else OVERALL_def_15 = 0;
run;

/* Random Effects */
Data VarY1_def_15 CoVar12_def_15 VarY2_def_15 ; Set Mvmm_random_def_15;
If covparm = 'UN(1,1)' then output VarY1_def_15;
If covparm = 'UN(2,1)' then output CoVar12_def_15;
If covparm = 'UN(2,2)' then output VarY2_def_15;
run;

Data VarY1_def_15; set VarY1_def_15;
label Estimate = VarY1_def_15;
label StdErr = SE_VarY1_def_15;
label Probz = P_VarY1_def_15;
rename Estimate = VarY1_def_15;
rename StdErr = SE_VarY1_def_15;
rename Probz = P_VarY1_def_15;
Drop Subject Covparm ZValue;
Run;

Data VarY2_def_15; set VarY2_def_15;
label Estimate = VarY2_def_15;
label StdErr = SE_VarY2_def_15;
label Probz = P_VarY2_def_15;
rename Estimate = VarY2_def_15;
rename StdErr = SE_VarY2_def_15;
rename Probz = P_VarY2_def_15;
Drop Subject Covparm ZValue;
Run;

Data CoVar12_def_15; set CoVar12_def_15;
label Estimate = CoVar12_def_15;
label StdErr = SE_CoVar12_def_15;
label Probz = P_CoVar12_def_15;
rename Estimate = CoVar12_def_15;
rename StdErr = SE_CoVar12_def_15;
rename Probz = P_CoVar12_def_15;
Drop Subject Covparm ZValue;
Run;

Data output_15; Set output_15;
Merge VarY1_def_15 VarY2_def_15 CoVar12_def_15;
by reps;
run;

```

```
/******  
*****
```

Linear Mixed Model with REML and KR Denominator df: 15% Missing

```
*****  
*****/
```

```
/* Syntax for Full Model With REML and KR Denominator df */  
PROC MIXED DATA=long_15 METHOD=REML NOCLPRINT COVTEST;  
  BY reps;  
  CLASS index ID newT;  
  MODEL Y_15 = index newT*index / DDFM = KR NOINT SOLUTION CL;  
  lsmeans newT*index / slice = index cl diff;  
  REPEATED index / SUBJECT = ID TYPE=UN R;  
  ods output ConvergenceStatus = MVMM_Conv_KR_15;  
  ods output FitStatistics = MVMM_Fit_KR_15;  
  ods output CovParms = MVMM_Random_KR_15;  
  ods output LSMeans = MVMM_Means_KR_15;  
  ods output Diffs = MVMM_Diffs_KR_15;  
  ods output SolutionF = MVMM_Fixed_KR_15;  
  RUN;  
  
/** convergence for Full Model KR**/  
data MVMM_Conv_KR_15; set MVMM_Conv_KR_15;  
drop Reason pdG;  
label Status = Status_MVMM_KR_15;  
rename Status = Status_MVMM_KR_15;  
label pdH = R_MVMM_KR_15;  
rename pdH = R_MVMM_KR_15;  
run;  
  
proc freq data = MVMM_Conv_KR_15 noprint;  
tables Reps*Status_MVMM_KR_15*R_MVMM_KR_15 / out=MVMM_Conv_KR_15;  
run;  
  
Data MVMM_Conv_KR_15; set MVMM_Conv_KR_15;  
label count = Conv_MVMM_KR_Count_15;  
rename count = Conv_MVMM_KR_Count_15;  
label percent = Conv_MVMM_KR_Percent_15;  
rename percent = Conv_MVMM_KR_Percent_15;  
run;  
  
/** Means From Full Model With KR **/  
data MVMM_Means_KR_15; retain Reps Parm; set MVMM_Means_KR_15;  
label index = Y_15;  
label Estimate = Means_KR_15;  
label StdErr = SE_MEANS_KR_15;  
label Lower = CIL_MEANS_KR_15;  
label Upper = CIU_MEANS_KR_15;  
rename index = Y_15;  
rename Estimate = Est_MEANS_KR_15;  
rename StdErr = SE_MEANS_KR_15;  
rename Lower = CIL_MEANS_KR_15;
```

```

rename Upper = CIU_MEANS_KR_15;
drop Effect tvalue Alpha probt df;
if INDEX = 1 then do;
    if newT = 0 then Parm = 'TMEAN(Y1KR)_15';
    if newT = 1 then Parm = 'CMEAN(Y1KR)_15';
end;
if INDEX = 2 then do;
    if newT = 0 then Parm = 'TMEAN(Y2KR)_15';
    if newT = 1 then Parm = 'CMEAN(Y2KR)_15';
end;
run;

/** Creating Separate Data Files: Means */
data CMEANY1KR_15 TXMEANY1KR_15 CMEANY2KR_15 TXMEANY2KR_15 ; set
MVMM_Means_KR_15;
    If Parm = 'TMEAN(Y1KR)_15' then output TXMEANY1KR_15;
    If Parm = 'CMEAN(Y1KR)_15' then output CMEANY1KR_15;
    If Parm = 'TMEAN(Y2KR)_15' then output TXMEANY2KR_15;
    If Parm = 'CMEAN(Y2KR)_15' then output CMEANY2KR_15;
Run;

Data TXMEANY1KR_15; Set TXMEANY1KR_15 ;
label EST_MEANS_KR_15 = TMeanY1_KR_15;
label SE_MEANS_KR_15 = SE_TMeanY1_KR_15;
label CIL_MEANS_KR_15 = CIL_TMEANY1_KR_15;
label CIU_MEANS_KR_15 = CUL_TMEANY1_KR_15;
rename EST_MEANS_KR_15 = TMeanY1_KR_15;
rename SE_MEANS_KR_15 = SE_TMeanY1_KR_15;
rename CIL_MEANS_KR_15 = CIL_TMEANY1_KR_15;
rename CIU_MEANS_KR_15 = CUL_TMEANY1_KR_15;
DROP PARM Y_15;
RUN;

Data CMEANY1KR_15; Set CMEANY1KR_15 ;
label EST_MEANS_KR_15 = CMeanY1_KR_15;
label SE_MEANS_KR_15 = SE_CMeanY1_KR_15;
label CIL_MEANS_KR_15 = CIL_CMEANY1_KR_15;
label CIU_MEANS_KR_15 = CUL_CMEANY1_KR_15;
rename EST_MEANS_KR_15 = CMeanY1_KR_15;
rename SE_MEANS_KR_15 = SE_CMeanY1_KR_15;
rename CIL_MEANS_KR_15 = CIL_CMEANY1_KR_15;
rename CIU_MEANS_KR_15 = CUL_CMEANY1_KR_15;
DROP PARM Y_15;
RUN;

Data TXMEANY2KR_15; Set TXMEANY2KR_15 ;
label EST_MEANS_KR_15 = TMeanY2_KR_15;
label SE_MEANS_KR_15 = SE_TMeanY2_KR_15;
label CIL_MEANS_KR_15 = CIL_TMEANY2_KR_15;
label CIU_MEANS_KR_15 = CUL_TMEANY2_KR_15;
rename EST_MEANS_KR_15 = TMeanY2_KR_15;
rename SE_MEANS_KR_15 = SE_TMeanY2_KR_15;
rename CIL_MEANS_KR_15 = CIL_TMEANY2_KR_15;
rename CIU_MEANS_KR_15 = CUL_TMEANY2_KR_15;
DROP PARM Y_15;
RUN;

```



```

Data CMEANY2KR_15; Set CMEANY2KR_15 ;
label EST_MEANS_KR_15 = CMeanY2_KR_15;
label SE_MEANS_KR_15 = SE_CMeanY2_KR_15;
label CIL_MEANS_KR_15 = CIL_CMEANY2_KR_15;
label CIU_MEANS_KR_15 = CUL_CMEANY2_KR_15;
rename EST_MEANS_KR_15 = CMeanY2_KR_15;
rename SE_MEANS_KR_15 = SE_CMeanY2_KR_15;
rename CIL_MEANS_KR_15 = CIL_CMEANY2_KR_15;
rename CIU_MEANS_KR_15 = CUL_CMEANY2_KR_15;
DROP PARM Y_15;
RUN;

```

```

DATA output_15; set output_15;
Merge MVMM_Conv_KR_15 TXMEANY1KR_15 CMEANY1KR_15 TXMEANY2KR_15 CMEANY2KR_15;
By Reps;
Run;

```

```

Data MVMM_Fixed_KR_15; retain Reps Parm; set MVMM_Fixed_KR_15;
label index = Y_KR_15;
label Estimate = Est_MVMM_Fixed_KR_15;
label StdErr = SE_MVMM_Fixed_KR_15;
label Probt = P_MVMM_Fixed_KR_15;
label Lower = CIL_MVMM_Fixed_KR_15;
label Upper = CIU_MVMM_Fixed_KR_15;
rename index = Y_KR_15;
rename Estimate = Est_MVMM_Fixed_KR_15;
rename StdErr = SE_MVMM_Fixed_KR_15;
rename Probt = P_MVMM_Fixed_KR_15;
rename Lower = CIL_MVMM_Fixed_KR_15;
rename Upper = CIU_MVMM_Fixed_KR_15;
drop Alpha;
if INDEX = 1 then do;
    if Effect = 'INDEX' then Parm = 'INT(Y1KR_15)';
    if NewT = 0 then Parm = 'd(Y1KR_15)';
end;
if INDEX = 2 then do;
    if Effect = 'INDEX' then Parm = 'INT(Y2KR_15)';
    if NewT = 0 then Parm = 'd(Y2KR_15)';
end;
run;

```

/** Creating Separate Data Files: Intercepts **/

```

Data INTY1_KR_15 INTY2_KR_15 ; set MVMM_Fixed_KR_15;
If EFFECT = 'INDEX' & Y_KR_15 = 1 then output INTY1_KR_15;
Else if EFFECT = 'INDEX' & Y_KR_15 = 2 then output INTY2_KR_15;
Run;

```

```

Data INTY1_KR_15 ; set INTY1_KR_15;
label Est_MVMM_Fixed_KR_15 = INTY1_KR_15;
label SE_MVMM_Fixed_KR_15 = SE_INTY1_KR_15;
label P_MVMM_Fixed_KR_15 = P_INTY1_KR_15;
label CIL_MVMM_Fixed_KR_15 = CIL_INTY1_KR_15;
label CIU_MVMM_Fixed_KR_15 = CIU_INTY1_KR_15;

```

```

label tvalue = t_INTY1_KR_15;
rename Est_MVMM_Fixed_KR_15 = INTY1_KR_15;
rename SE_MVMM_Fixed_KR_15 = SE_INTY1_KR_15;
rename P_MVMM_Fixed_KR_15 = P_INTY1_KR_15;
rename CIL_MVMM_Fixed_KR_15 = CIL_INTY1_KR_15;
rename CIU_MVMM_Fixed_KR_15 = CIU_INTY1_KR_15;
rename tvalue = t_INTY1_KR_15;
DROP PARM EFFECT Y_KR_15 NewT df;
RUN;

Data INTY2_KR_15 ; set INTY2_KR_15;
label Est_MVMM_Fixed_KR_15 = INTY2_KR_15;
label SE_MVMM_Fixed_KR_15 = SE_INTY2_KR_15;
label P_MVMM_Fixed_KR_15 = P_INTY2_KR_15;
label CIL_MVMM_Fixed_KR_15 = CIL_INTY2_KR_15;
label CIU_MVMM_Fixed_KR_15 = CIU_INTY2_KR_15;
label tvalue = t_INTY2_KR_15;
rename Est_MVMM_Fixed_KR_15 = INTY2_KR_15;
rename SE_MVMM_Fixed_KR_15 = SE_INTY2_KR_15;
rename P_MVMM_Fixed_KR_15 = P_INTY2_KR_15;
rename CIL_MVMM_Fixed_KR_15 = CIL_INTY2_KR_15;
rename CIU_MVMM_Fixed_KR_15 = CIU_INTY2_KR_15;
rename tvalue = t_INTY2_KR_15;
DROP PARM EFFECT Y_KR_15 NewT ;
RUN;

/** Creating Separate Data Files: Slopes **/

Data DY1_KR_15 DY2_KR_15 ; set MVMM_Fixed_KR_15;
If Y_KR_15 = 1 & NewT = 0 then output DY1_KR_15;
Else if Y_KR_15 = 2 & NewT = 0 then output DY2_KR_15;
Run;

Data DY1_KR_15 ; set DY1_KR_15;
label Est_MVMM_Fixed_KR_15 = DY1_KR_15;
label SE_MVMM_Fixed_KR_15 = SE_DY1_KR_15;
label P_MVMM_Fixed_KR_15 = P_DY1_KR_15;
label CIL_MVMM_Fixed_KR_15 = CIL_DY1_KR_15;
label CIU_MVMM_Fixed_KR_15 = CIU_DY1_KR_15;
label tvalue = t_DY1_KR_15;
label df = df_DY1_KR_15;
rename Est_MVMM_Fixed_KR_15 = DY1_KR_15;
rename SE_MVMM_Fixed_KR_15 = SE_DY1_KR_15;
rename P_MVMM_Fixed_KR_15 = P_DY1_KR_15;
rename CIL_MVMM_Fixed_KR_15 = CIL_DY1_KR_15;
rename CIU_MVMM_Fixed_KR_15 = CIU_DY1_KR_15;
rename tvalue = t_DY1_KR_15;
rename df = df_DY1_KR_15;
DROP PARM EFFECT Y_KR_15;
RUN;

Data DY2_KR_15 ; set DY2_KR_15;
label Est_MVMM_Fixed_KR_15 = DY2_KR_15;
label SE_MVMM_Fixed_KR_15 = SE_DY2_KR_15;
label P_MVMM_Fixed_KR_15 = P_DY2_KR_15;
label CIL_MVMM_Fixed_KR_15 = CIL_DY2_KR_15;

```

```

label CIU_MVMM_Fixed_KR_15 = CIU_DY2_KR_15;
label tvalue = t_DY2_KR_15;
label df = df_DY2_KR_15;
rename Est_MVMM_Fixed_KR_15 = DY2_KR_15;
rename SE_MVMM_Fixed_KR_15 = SE_DY2_KR_15;
rename P_MVMM_Fixed_KR_15 = P_DY2_KR_15;
rename CIL_MVMM_Fixed_KR_15 = CIL_DY2_KR_15;
rename CIU_MVMM_Fixed_KR_15 = CIU_DY2_KR_15;
rename tvalue = t_DY2_KR_15;
rename df = df_DY2_KR_15;
DROP PARM EFFECT Y_KR_15;
RUN;

```

```

Data output_15; set output_15;
Merge INTY1_KR_15 INTY2_KR_15 DY1_KR_15 DY2_KR_15 ;
BY REPS;
Run;

```

```

Data output_15; set output_15;
if P_DY1_KR_15 <= .05 then SigDY1_KR_15 = 1;
else SigDY1_KR_15 = 0;
if P_DY2_KR_15 <= .05 then SigDY2_KR_15 = 1;
else SigDY2_KR_15 = 0;
if P_DY1_KR_15 <= .025 or P_DY2_KR_15 <= .025 then OVERALL_KR_15 = 1;
Else OVERALL_KR_15 = 0;
run;

```

```

/* Random Effects */

```

```

Data VarY1_KR_15 CoVar12_KR_15 VarY2_KR_15 ; Set Mvmm_random_KR_15;
If covparm = 'UN(1,1)' then output VarY1_KR_15;
If covparm = 'UN(2,1)' then output CoVar12_KR_15;
If covparm = 'UN(2,2)' then output VarY2_KR_15;
run;

```

```

Data VarY1_KR_15; set VarY1_KR_15;
label Estimate = VarY1_KR_15;
label StdErr = SE_VarY1_KR_15;
label Probz = P_VarY1_KR_15;
rename Estimate = VarY1_KR_15;
rename StdErr = SE_VarY1_KR_15;
rename Probz = P_VarY1_KR_15;
Drop Subject Covparm ZValue;
Run;

```

```

Data VarY2_KR_15; set VarY2_KR_15;
label Estimate = VarY2_KR_15;
label StdErr = SE_VarY2_KR_15;
label Probz = P_VarY2_KR_15;
rename Estimate = VarY2_KR_15;
rename StdErr = SE_VarY2_KR_15;
rename Probz = P_VarY2_KR_15;
Drop Subject Covparm ZValue;
Run;

```

```

Data CoVar12_KR_15; set CoVar12_KR_15;
label Estimate = CoVar12_KR_15;
label StdErr = SE_CoVar12_KR_15;
label Probz = P_CoVar12_KR_15;
rename Estimate = CoVar12_KR_15;
rename StdErr = SE_CoVar12_KR_15;
rename Probz = P_CoVar12_KR_15;
Drop Subject Covparm ZValue;
Run;

```

```

Data output_15; Set output_15;
Merge VarY1_KR_15 VarY2_KR_15 CoVar12_KR_15 newlineconds;
run;

```

```

/* Forming Relative Bias Measures */

```

```

DATA output_15; set output_15;
P_Rel_Bias_MSEY1_15 = (MSEY1_15 - var1)/var1;
P_Rel_Bias_MSEY2_15 = (MSEY2_15 - var2)/var2;
P_Rel_Bias_MSEY2_PAIR_15 = (MSEY2_PAIR_15 - var2)/var2;
P_Rel_Bias_COVAR12_GLM_15 = (covar12_15 - r12)/r12;
P_Rel_Bias_VarY1_ML_15 = (VarY1_ML_15 - var1)/var1;
P_Rel_Bias_VarY2_ML_15 = (VarY2_ML_15 - var2)/var2;
P_Rel_Bias_COVAR12_ML_15 = (CoVar12_ML_15 - r12)/r12;
P_Rel_Bias_VarY1_KR_15 = (VarY1_KR_15 - var1)/var1;
P_Rel_Bias_VarY2_KR_15 = (VarY2_KR_15 - var2)/var2;
P_Rel_Bias_CoVar12_KR_15 = (CoVar12_KR_15 - r12)/r12;
P_Rel_Bias_TMeanY1_OLS_15 = (TmeanY1_15 - EsY1)/EsY1;
P_Rel_Bias_TMeanY2_OLS_15 = (TMeanY2_15 - EsY2)/EsY2;
P_Rel_Bias_TMeanY2_PAIR_15 = (TMean_Y2_PAIR_15 - EsY2)/EsY2;
P_Rel_Bias_TMeanY1_ML_15 = (TMeanY1_ML_15 - EsY1)/EsY1;
P_Rel_Bias_TMeanY2_ML_15 = (TMeanY2_ML_15 - EsY2)/EsY2;
P_Rel_Bias_TMeanY1_def_15 = (TMeanY1_def_15 - EsY1)/EsY1;
P_Rel_Bias_TMeanY2_def_15 = (TMeanY2_def_15 - EsY2)/EsY2;
P_Rel_Bias_TMeanY1_KR_15 = (TMeanY1_KR_15 - EsY1)/EsY1;
P_Rel_Bias_TMeanY2_KR_15 = (TMeanY2_KR_15 - EsY2)/EsY2;
P_Rel_Bias_OLSDiffY1_15 = (DiffY1_15 - EsY1)/EsY1;
P_Rel_Bias_OLSDiffY2_15 = (DiffY2_15 - EsY2)/EsY2;
P_Rel_Bias_DiffY2_PAIR_15 = (Diff_Y2_PAIR_15 - EsY2)/EsY2;
P_Rel_Bias_MLDiffY1_15 = (DY1_ML_15 - EsY1)/EsY1;
P_Rel_Bias_MLDiffY2_15 = (DY2_ML_15 - EsY2)/EsY2;
P_Rel_Bias_DEFDiffY1_15 = (DY1_def_15 - EsY1)/EsY1;
P_Rel_Bias_DEFDiffY2_15 = (DY2_def_15 - EsY2)/EsY2;
P_Rel_Bias_KRDiffY1_15 = (dy1_KR_15 - EsY1)/EsY1;
P_Rel_Bias_KRDiffY2_15 = (dy2_KR_15 - EsY2)/EsY2;
Run;

```

```

/* Creating CI coverage variables */

```

```

DATA Output_15; SET output_15;
if CCIUpperMeanY2pair_15 <= 0 and CCIUpperMeanY2pair_15 >= 0 then
OLS_Capture_CONMEANY2pair_15 = 1;
else OLS_Capture_CONMEANY2pair_15 = 0;

```

```

if TCILowerMeanY2pair_15 <=EsY2 and TCIOpperMeanY2pair_15 >= EsY2 then
OLS_Capture_TMEANY2pair_15 = 1;
else OLS_Capture_TMEANY2pair_15 = 0;
if CCIILowerMeanY1_15 <=0 and CCIUpperMeanY1_15 >= 0 then
OLS_Capture_CONMEANY1_15 = 1;
else OLS_Capture_CONMEANY1_15 = 0;
if CCIILowerMeanY2_15 <=0 and CCIUpperMeanY2_15 >= 0 then
OLS_Capture_CONMEANY2_15 = 1;
else OLS_Capture_CONMEANY2_15 = 0;
if TCILowerMeanY1_15 <=EsY1 and TCIOpperMeanY1_15 >= EsY1 then
OLS_Capture_TMEANY1_15 = 1;
else OLS_Capture_TMEANY1_15 = 0;
if TCILowerMeanY2_15 <=EsY2 and TCIOpperMeanY2_15 >= EsY1 then
OLS_Capture_TMEANY2_15 = 1;
else OLS_Capture_TMEANY2_15 = 0;
if lowerCLY1_15 <=EsY1 and upperCLY1_15 >= EsY1 then OLS_Capture_DIFFY1_15 =
1;
else OLS_Capture_DIFFY1_15 = 0;
if lowerCLY2_15 <=EsY2 and upperCLY2_15 >= EsY2 then OLS_Capture_DIFFY2_15 =
1;
else OLS_Capture_DIFFY2_15 = 0;
if lowerCLY2_PAIR_15 <=EsY2 and upperCLY2_PAIR_15 >= EsY2 then
OLS_Capture_DIFFY2_PAIR_15 = 1;
else OLS_Capture_DIFFY2_PAIR_15 = 0;
if CIL_DY1_ML_15 <=EsY1 and CIU_DY1_ML_15 >= EsY1 then ML_Capture_DIFFY1_15
= 1;
else ML_Capture_DIFFY1_15 = 0;
if CIL_DY2_ML_15 <=EsY2 and CIU_DY2_ML_15 >= EsY2 then ML_Capture_DIFFY2_15 =
1;
else ML_Capture_DIFFY2_15 = 0;
if CIL_DY1_DEF_15 <=EsY1 and CIU_DY1_DEF_15 >= EsY1 then
DEF_Capture_DIFFY1_15 = 1;
else DEF_Capture_DIFFY1_15 = 0;
if CIL_DY2_DEF_15 <=EsY2 and CIU_DY2_DEF_15 >= EsY2 then
DEF_Capture_DIFFY2_15 = 1;
else DEF_Capture_DIFFY2_15 = 0;
if CIL_DY1_KR_15 <=EsY1 and CIU_DY1_KR_15 >= EsY1 then KR_Capture_DIFFY1_15
= 1;
else KR_Capture_DIFFY1_15 = 0;
if CIL_DY2_KR_15 <= EsY2 and CIU_DY2_KR_15 >= EsY2 then
KR_Capture_DIFFY2_15 = 1;
else KR_Capture_DIFFY2_15 = 0;
run;

```

```

/*****

```

ANALYSIS for 30% INCOMPLETE DATA

```

*****/

```

```

/***** ANALYSIS *****/
/* Traditional MANOVA */

PROC GLM DATA = alldata;
ods output OverallANOVA=ALLAnova_30;
ods output errorSSCP= eSSCP_30;
ods output partialcorr = corrs_30;
ods output Multstat = WILKS_30;
ods output cldiffs = cldiff_30;
ods output lsmeans = means_30;
ods output LSmeanCL= clmeans_30;
ods output means = descriptives_30;
BY Cond reps;
CLASS T;
MODEL Y1_30 Y2_30 = T;
MANOVA H = T / PRINTE;
MEANS T/ ALPHA = .05 CLDIFF LSD;
lsmeans T/ CL tdiff;
means T;
RUN;
QUIT;

data wilks_30 (keep = cond reps singular_30);
set wilks_30;
  by cond reps;
  firstrep = first.reps;
  if firstrep NE 1 then delete;
  if numdf < 3 then singular_30 = 0;
  else singular_30 = 1;
run;

/* Obtaining listwise deleted treatment and control means for Y1 and Y2,
along with CI for each group mean */

data clmeansy1_30 clmeansy2_30; set clmeans_30;
if dependent = 'Y1_30' then output CLmeansY1_30;
else if dependent = 'Y2_30' then output CLmeansY2_30;
run;

Data TCLMeanY1_30 CCLmeanY1_30;
set clmeansY1_30;
if T = 1 then output TCLmeanY1_30;
else if T = 0 then output CCLmeanY1_30;
drop effect dependent T;
run;

Data TCLMeanY1_30;
Set TCLMeanY1_30;
label Lowercl = TCILowerMeanY1_30;
label lsmean = TMeanY1_30;
label Uppercl = TCIUpperMeanY1_30;
rename Lowercl = TCILowerMeanY1_30;

```

```

rename lsmean = TMeanY1_30;
rename Uppercl = TCIUpperMeanY1_30;
run;

Data CCLMeanY1_30;
Set CCLMeanY1_30;
label Lowercl = CCILowerMeanY1_30;
label lsmean = CMeanY1_30;
label Uppercl = CCIUpperMeanY1_30;
rename Lowercl = CCILowerMeanY1_30;
rename lsmean = CMeanY1_30;
rename Uppercl = CCIUpperMeanY1_30;
run;

Data TCLMeanY2_30 CCLmeanY2_30;
set clmeansY2_30;
if T = 1 then output TCLmeanY2_30;
else if T = 0 then output CCLmeanY2_30;
Drop effect dependent T;
run;

Data TCLMeanY2_30;
Set TCLMeanY2_30;
label Lowercl = TCILowerMeanY2_30;
label lsmean = TMeanY2_30;
label Uppercl = TCIUpperMeanY2_30;
rename Lowercl = TCILowerMeanY2_30;
rename lsmean = TMeanY2_30;
rename Uppercl = TCIUpperMeanY2_30;
run;

Data CCLMeanY2_30;
Set CCLMeanY2_30;
label Lowercl = CCILowerMeanY2_30;
label lsmean = CMeanY2_30;
label Uppercl = CCIUpperMeanY2_30;
rename Lowercl = CCILowerMeanY2_30;
rename lsmean = CMeanY2_30;
rename Uppercl = CCIUpperMeanY2_30;
run;

Data output_30;
merge Tclmeany1_30 CCLMeanY1_30 TCLMeanY2_30 CCLmeanY2_30 ;
by cond reps;
run;

/* Creating separate data sets for ANOVAs of Y1 and Y2 */
/* Splits ANOVA Results into Y1 and Y2 data sets */

Data ANOVAY1_30 ANOVAY2_30;
set aLLANOVA_30;
if dependent = 'Y1_30' then output ANOVAY1_30;
else if dependent = 'Y2_30' then output ANOVAY2_30;
run;

/* F test for Y1_30 */

```

```

DATA FtestY1_30; keep cond reps dependent FValue ProbF; set ANOVAY1_30;
if Source = 'Model' then output;
label ProbF = p-value FY1_30;
rename probF = PFY1_30;
label FValue = FY1_30;
rename FValue = FY1_30;
Run;

/* F test for Y2 */
DATA FtestY2_30; keep cond reps dependent FValue ProbF; set ANOVAY2_30;
if Source = 'Model' then output;
label ProbF = p-value FY2_30;
rename probF = PFY2_30;
label FValue = FY2_30;
rename FValue = FY2_30;
Run;

Data output_30; set output_30;
merge wilks_30 FtestY1_30 FtestY2_30 ;
by cond reps;
drop dependent;
if PFY1_30 <= .05 then Sig_FY1_30 = 1; else Sig_FY1_30 = 0;
if PFY2_30 <= .05 then Sig_FY2_30 = 1; else Sig_FY2_30 = 0;
If PFY1_30 <= .025 or PFY2_30 <= .025 then Sig_SUTS_30 = 1; else Sig_SUTS_30
= 0;
run;

/* Mean Square Error for Y1 */
DATA MSEY1_30; keep cond reps MS; set ANOVAY1_30;
if source = 'Error' then output;
label MS = MSEY1_30;
rename MS = MSEY1_30;
Run;

/* Mean Square Error for Y2 */
DATA MSEY2_30; keep cond reps MS; set ANOVAY2_30;
if source = 'Error' then output;
label MS = MSEY2_30;
rename MS = MSEY2_30;
Run;

Data output_30;
set output_30;
merge MSEY1_30 MSEY2_30;
by cond reps;
run;

/* Creating Confidence Limits Data Set */
DATA CL_30; SET Cldiff_30;
if Comparison = '1 - 0' then output;
run;

Data CLY1_30 CLY2_30 ;
set CL_30;
if dependent = 'Y1_30' then output CLY1_30;
else if dependent = 'Y2_30' then output CLY2_30;
drop method comparison Effect Dependent;

```



```

run;

Data CLY1_30;
set CLY1_30;
label lowerCL= lowerCLY1_30;
rename lowerCL= lowerCLY1_30;
label upperCl = upperCLY1_30;
rename upperCl = upperCLY1_30;
label Difference = DiffY1_30;
rename Difference = DiffY1_30;
label Significance = sigY1_30;
rename Significance = sigY1_30;
Run;

Data CLY2_30;
set CLY2_30;
label lowerCL= lowerCLY2_30;
rename lowerCL= lowerCLY2_30;
label upperCl = upperCLY2_30;
rename upperCl = upperCLY2_30;
label Difference = DiffY2_30;
rename Difference = DiffY2_30;
label Significance = sigY2_30;
rename Significance = sigY2_30;
Run;

Data output_30;
set output_30;
merge CLY1_30 CLY2_30;
by cond Reps;
Run;

/* Keeping Pooled Correlations */

Data NewCorrs1_30; Set Corrs_30;
by cond reps;
firstrep = first.reps;
if firstrep NE 1 then delete;
keep cond reps Y2_30 P_Y2_30 ;
label Y2_30 = Corr12_30;
rename Y2_30 = Corr12_30;
label P_Y2_30 = p-value CORR12_30;
Rename P_Y2_30 = P_Val_CORR12_30;
DROP Y1_30 Variable P_y1_30;
run;

Data output_30;
set output_30;
merge NewCorrs1_30 ;
by cond Reps;
Run;

/* Obtain group Ns and SDs below */

Data Tinfo_30 Cinfo_30; Set Descriptives_30;
By reps;
if T = 0 then output Cinfo_30;

```

```

else if T = 1 then output Tinfo_30;
run;

Data Tinfo_30; Set Tinfo_30;
rename N = TreatN_30;
label N = TreatN_30;
rename SD_Y1_30 = TSD_Y1_30;
label SD_Y1_30 = TSD_Y1_30;
rename SD_Y2_30 = TSD_Y2_30;
label SD_Y2_30 = TSD_Y2_30;
Drop Mean_Y1_30 Mean_Y2_30 Effect T;
run;

Data Cinfo_30; Set Cinfo_30;
rename N = ControlN_30;
label N = ControlN_30;
rename SD_Y1_30 = CSD_Y1_30;
label SD_Y1_30 = CSD_Y1_30;
rename SD_Y2_30 = CSD_Y2_30;
label SD_Y2_30 = CSD_Y2_30;
Drop Mean_Y1_30 Mean_Y2_30 Effect T;
run;

Data output_30;
set output_30;
merge Tinfo_30 Cinfo_30;
By cond reps;
N_both_30 = TreatN_30+ControlN_30;
TVar_Y1_30 = sqrt(TSD_Y1_30);
TVar_Y2_30 = sqrt(TSD_Y2_30);
cVar_Y1_30 = sqrt(CSD_Y1_30);
CVar_Y2_30 = sqrt(CSD_Y2_30);
Covar12_30 = corr12_30/sqrt(msey1_30)*sqrt(msey2_30);
Run;

/* Obtaining t tests for Y1 Y2 */
Data ttestY1_30 ttestY2_30 ;
set means_30 ;
if dependent = 'Y1_30 ' then do;
    if T = 0 then output ttestY1_30 ;
end;
else if dependent = 'Y2_30 ' then do;
    if T = 0 then output ttestY2_30 ;
end;
Drop lsmean effect dependent T probtdiff;
run;

Data ttestY1_30 ; Set ttesty1_30 ;
label tdiff = ttestY1_glm_30 ;
rename tdiff = ttestY1_glm_30 ;
Run;

Data ttestY2_30 ; Set ttesty2_30 ;
label tdiff = ttesty2_glm_30 ;
rename tdiff = ttesty2_glm_30 ;
run;

```

```

Data output_30 ; set output_30;
merge ttesty1_30  ttesty2_30 ;
By Cond reps;
Run;

```

```

Data output_30;
Set output_30;
SEDiffY1_glm_30 = abs(diffy1_30)/abs(TtestY1_glm_30);
SEDiffY2_glm_30 = abs(diffy2_30)/abs(TtestY2_glm_30);
run;

```

```

/* ANOVA FOR Y2 USING PAIRWISE DELETION: HERE, ALL CASES USED FOR Y2
   Note that ANOVA for Y1 is obtained with the MANOVA procedure above */

```

```

PROC GLM DATA = alldata;
ods output OverallANOVA=Y2AnovaPAIR_30;
ods output cldiffs = Y2cldiffPAIR_30;
ods output Means = Y2PAIR_30;
ods output lsmeans = means_pair_30;
ods output LSMeanCL = Y2clmeanspair_30;
By cond reps;
CLASS T;
MODEL Y2_30 = T;
MEANS T/ ALPHA = .05 CLDIFF LSD;
lsmeans T/ CL tdiff;
means T;
RUN;
QUIT;

```

```

/* Obtaining pairwise deleted treatment and control means for Y2, along with
   CI for each group mean */

```

```

Data TCLMeanY2pair_30 CCLmeanY2pair_30;
set Y2clmeanspair_30;
if T = 1 then output TCLmeanY2pair_30;
else if T = 0 then output CCLmeanY2pair_30;
run;

```

```

Data TCLMeanY2pair_30;
Set TCLMeanY2pair_30;
label Lowercl = TCILowerMeanY2pair_30;
label lsmean = TMeanY2pair_30;
label Uppercl = TCIOpperMeanY2pair_30;
rename Lowercl = TCILowerMeanY2pair_30;
rename lsmean = TMeanY2pair_30;
rename Uppercl = TCIOpperMeanY2pair_30;
Drop effect dependent T;
run;

```

```

Data CCLMeanY2pair_30;
Set CCLMeanY2pair_30;
label Lowercl = CCILowerMeanY2pair_30;
label lsmean = CMeanY2pair_30;
label Uppercl = CCIUpperMeanY2pair_30;

```

```

rename Lowercl = CCILowerMeanY2pair_30;
rename lsmean = CMeanY2pair_30;
rename Uppercl = CCIUpperMeanY2pair_30;
Drop effect dependent T;
run;

Data output_30;
set output_30;
merge TCLMeanY2pair_30 CCLmeanY2Pair_30;
by cond reps;
run;

/* F test for Y2: PAIRWISE DELETION */
DATA FtestY2pair_30; keep cond reps Fvalue ProbF; set Y2ANOVAPAIR_30;
if Source = 'Model' then output;
label ProbF = p-value FY2PAIR_30;
rename probF = PFY2_PAIR_30;
label FValue = FY2PAIR_30;
rename FValue = FY2_PAIR_30;
Run;

/* Mean Square Error for Y2 PAIRWISE DELETION: ALL CASES */
DATA MSEY2_PAIR_30; keep cond reps MS; set Y2ANOVAPAIR_30;
if source = 'Error' then output;
label MS = MSEY2_PAIR_30;
rename MS = MSEY2_PAIR_30;
Run;

/* Obtaining CL limits for Y2 PAIRWISE DELETION */
DATA CL_y2_pair_30; SET Y2CLDIFFPAIR_30;
if Comparison = '1 - 0' then output;
label lowerCL= lowerCLY2_PAIR_30;
rename lowerCL= lowerCLY2_PAIR_30;
label upperCl = upperCLY2_PAIR_30;
rename upperCl = upperCLY2_PAIR_30;
label difference = diff_Y2_PAIR_30;
rename difference = diff_Y2_PAIR_30;
drop effect dependent comparison method significance;
run;

Data Cinfo_Y2PAIR_30 Tinfo_Y2Pair_30; set y2pair_30;
By cond reps;
if T = 0 then output Cinfo_Y2PAIR_30;
else if T = 1 then output Tinfo_Y2PAIR_30;
run;

Data Tinfo_Y2Pair_30; Set Tinfo_Y2Pair_30;
rename N = TreatN_Y2PAIR_30;
label N = TreatN_Y2PAIR_30;
rename SD_Y2_30 = TSD_Y2_PAIR_30;
label SD_Y2_30 = TSD_Y2_PAIR_30;
rename Mean_Y2_30 = TMean_Y2_PAIR_30;
label Mean_Y2_30 = TMean_Y2_PAIR_30;
Drop Effect T;
run;

Data Cinfo_Y2Pair_30; Set Cinfo_Y2Pair_30;

```

```

rename N = ControlN_Y2Pair_30;
label N = ControlN_Y2Pair_30;
rename SD_Y2_30 = CSD_Y2_Pair_30;
label SD_Y2_30 = CSD_Y2_Pair_30;
rename Mean_Y2_30 = CMean_Y2_PAIR_30;
label Mean_Y2_30 = CMean_Y2_PAIR_30;
Drop Effect T;
run;

/* Obtaining t test for Y2 pairwise */
Data ttestY2_pair_30 ;
set means_pair_30 ;
    if T = 0 then output ttestY2_pair_30 ;
Drop lsmean effect dependent T probtdiff;
    label tdiff = ttesty2_glm_pair_30 ;
    rename tdiff = ttesty2_glm_pair_30 ;
run;

Data output_30 ; set output_30;
merge ttestY2_pair_30 ;
Run;

Data output_30;
set output_30;
merge TCLMeanY2pair_30 CCLmeanY2Pair_30 FtestY2pair_30 MSEY2_PAIR_30
CL_Y2_PAIR_30 Tinfo_Y2Pair_30 Cinfo_Y2Pair_30;
by cond reps;
if PFY2_PAIR_30 <= .05 then Sig_FY2_PAIR_30 = 1; else Sig_FY2_PAIR_30 = 0;
SEDiffY2_PAIR_30 = abs(diff_Y2_PAIR_30)/abs(TtestY2_glm_Pair_30);
If PFY1_30 <= .025 or PFY2_PAIR_30 <= .025 then Sig_SUTS_PAIR_30 = 1; else
Sig_SUTs_PAIR_30 = 0;
run;

/***** CONVERTING WIDE DATA SET TO LONG DATA SET *****/
/***** Wide Data Set *****/
data Wide_30; set alldata;
run;

/***** Wide to Long *****/
data Long_30; set Wide_30;
INDEX=1; Y_30=Y1_30; R_30=R1; output;
INDEX=2; Y_30=Y2_30; R_30=R2; output;
Keep reps ID INDEX T NewT Y_30 R_30;
run;

/*****
*****

```

Linear Mixed Model with FIML and Z Tests for Effects of Y1 and Y2: 30% Missingness

```
*****  
*****/
```

```
/* Syntax for FIML and Default Between-Within df  
   Z Tests for effects of Y1 and Y2 obtained below */
```

```
PROC MIXED DATA=long_30 METHOD=ML NOCLPRINT COVTEST;  
  BY reps;  
  CLASS index ID;  
  MODEL Y_30 = index index*T/ CL NOINT SOLUTION;  
  REPEATED index / SUBJECT = ID TYPE=UN R;  
  ods output ConvergenceStatus = MVMM_Conv_FullML_30;  
  ods output FitStatistics = MVMM_Fit_FullML_30;  
  ods output CovParms = MVMM_Random_FullML_30;  
  ods output SolutionF = MVMM_Fixed_FullML_30;  
  RUN;
```

```
/** convergence for Full Model With ML**/
```

```
data MVMM_Conv_FullML_30; set MVMM_Conv_FullML_30;  
drop Reason pdG;  
label Status = Status_MVMM_FullML_30;  
rename Status = Status_MVMM_FullML_30;  
label pdH = R_MVMM_FullML_30;  
rename pdH = R_MVMM_FullML_30;  
run;
```

```
proc freq data=MVMM_Conv_FullML_30 noprint;  
tables Reps*Status_MVMM_FullML_30*R_MVMM_FullML_30/ out=MVMM_Conv_FullML_30;  
run;
```

```
Data MVMM_Conv_FullML_30; set MVMM_Conv_FullML_30;  
label count = Conv_MVMM_FullML_Count_30;  
rename count = Conv_MVMM_FullML_Count_30;  
label percent = Conv_MVMM_FullML_Percent_30;  
rename percent = Conv_MVMM_FullML_Percent_30;  
run;
```

```
Data output_30; set output_30;  
Merge MVMM_Conv_FullML_30;  
BY REPS;  
Run;
```

```
/** Fixed Effects From Full Model With ML **/
```

```
data MVMM_Fixed_FullML_30; retain Reps Parm; set MVMM_Fixed_FullML_30;  
label index = Y_30;  
label Estimate = Est_MVMM_Fixed_FullML_30;  
label StdErr = SE_MVMM_Fixed_FullML_30;  
label Probt = P_MVMM_Fixed_FullML_30;  
label Lower = CIL_MVMM_Fixed_FullML_30;
```

```

label Upper = CIU_MVMM_Fixed_FullML_30;
rename index = Y_30;
rename Estimate = Est_MVMM_Fixed_FullML_30;
rename StdErr = SE_MVMM_Fixed_FullML_30;
rename Probt = P_MVMM_Fixed_FullML_30;
rename Lower = CIL_MVMM_Fixed_FullML_30;
rename Upper = CIU_MVMM_Fixed_FullML_30;
drop Alpha;
if INDEX = 1 then do;
    if Effect = 'INDEX' then Parm = 'INT(Y1)_30';
    if Effect = 'T*INDEX' then Parm = 'd(Y1)_30';
end;
if INDEX = 2 then do;
    if Effect = 'INDEX' then Parm = 'INT(Y2)_30';
    if Effect = 'T*INDEX' then Parm = 'd(Y2)_30';
end;
run;

/** Creating Separate Data Files: Intercepts **/
data INTY1_ML_30 INTY2_ML_30 ; set MVMM_Fixed_FullML_30;
If EFFECT = 'INDEX' & Y_30 = 1 then output INTY1_ML_30;
Else if EFFECT = 'INDEX' & Y_30 = 2 then output INTY2_ML_30;
Run;

data INTY1_ML_30 ; set INTY1_ML_30;
label Est_MVMM_Fixed_FullML_30 = INTY1_ML_30;
label SE_MVMM_Fixed_FullML_30 = SE_INTY1_ML_30;
label P_MVMM_Fixed_FullML_30 = P_INTY1_ML_30;
label CIL_MVMM_Fixed_FullML_30 = CIL_INTY1_ML_30;
label CIU_MVMM_Fixed_FullML_30 = CIU_INTY1_ML_30;
label tvalue = t_INTY1_ML_30;
rename Est_MVMM_Fixed_FullML_30 = INTY1_ML_30;
rename SE_MVMM_Fixed_FullML_30 = SE_INTY1_ML_30;
rename P_MVMM_Fixed_FullML_30 = P_INTY1_ML_30;
rename CIL_MVMM_Fixed_FullML_30 = CIL_INTY1_ML_30;
rename CIU_MVMM_Fixed_FullML_30 = CIU_INTY1_ML_30;
rename tvalue = t_INTY1_ML_30;
DROP PARM EFFECT Y_30 df;
RUN;

data INTY2_ML_30 ; set INTY2_ML_30;
label Est_MVMM_Fixed_FullML_30 = INTY2_ML_30;
label SE_MVMM_Fixed_FullML_30 = SE_INTY2_ML_30;
label P_MVMM_Fixed_FullML_30 = P_INTY2_ML_30;
label CIL_MVMM_Fixed_FullML_30 = CIL_INTY2_ML_30;
label CIU_MVMM_Fixed_FullML_30 = CIU_INTY2_ML_30;
label tvalue = t_INTY2_ML_30;
rename Est_MVMM_Fixed_FullML_30 = INTY2_ML_30;
rename SE_MVMM_Fixed_FullML_30 = SE_INTY2_ML_30;
rename P_MVMM_Fixed_FullML_30 = P_INTY2_ML_30;
rename CIL_MVMM_Fixed_FullML_30 = CIL_INTY2_ML_30;
rename CIU_MVMM_Fixed_FullML_30 = CIU_INTY2_ML_30;
rename tvalue = t_INTY2_ML_30;
DROP PARM EFFECT Y_30 df;
RUN;

```

```

    /** Creating Separate Data Files: Slopes **/
data DY1_ML_30 DY2_ML_30 ; set MVMM_Fixed_FullML_30;
If EFFECT = 'T*INDEX' & Y_30 = 1 then output DY1_ML_30;
Else if EFFECT = 'T*INDEX' & Y_30 = 2 then output DY2_ML_30;
Run;

```

```

data DY1_ML_30 ; set DY1_ML_30;
label Est_MVMM_Fixed_FullML_30 = DY1_ML_30;
label SE_MVMM_Fixed_FullML_30 = SE_DY1_ML_30;
label P_MVMM_Fixed_FullML_30 = P_DY1_ML_30;
label CIL_MVMM_Fixed_FullML_30 = CIL_DY1_ML_30;
label CIU_MVMM_Fixed_FullML_30 = CIU_DY1_ML_30;
label tvalue = t_DY1_ML_30;
label df = df_DY1_ML_30;
rename Est_MVMM_Fixed_FullML_30 = DY1_ML_30;
rename SE_MVMM_Fixed_FullML_30 = SE_DY1_ML_30;
rename P_MVMM_Fixed_FullML_30 = P_DY1_ML_30;
rename CIL_MVMM_Fixed_FullML_30 = CIL_DY1_ML_30;
rename CIU_MVMM_Fixed_FullML_30 = CIU_DY1_ML_30;
rename tvalue = t_DY1_ML_30;
rename df = df_DY1_ML_30;
DROP PARM EFFECT Y_30;
RUN;

```

```

data DY2_ML_30 ; set DY2_ML_30;
label Est_MVMM_Fixed_FullML_30 = DY2_ML_30;
label SE_MVMM_Fixed_FullML_30 = SE_DY2_ML_30;
label P_MVMM_Fixed_FullML_30 = P_DY2_ML_30;
label CIL_MVMM_Fixed_FullML_30 = CIL_DY2_ML_30;
label CIU_MVMM_Fixed_FullML_30 = CIU_DY2_ML_30;
label tvalue = t_DY2_ML_30;
label df = df_DY2_ML_30;
rename Est_MVMM_Fixed_FullML_30 = DY2_ML_30;
rename SE_MVMM_Fixed_FullML_30 = SE_DY2_ML_30;
rename P_MVMM_Fixed_FullML_30 = P_DY2_ML_30;
rename CIL_MVMM_Fixed_FullML_30 = CIL_DY2_ML_30;
rename CIU_MVMM_Fixed_FullML_30 = CIU_DY2_ML_30;
rename tvalue = t_DY2_ML_30;
rename df = df_DY2_ML_30;
DROP PARM EFFECT Y_30;
RUN;

```

```

Data output_30; set output_30;
Merge INTY1_ML_30 INTY2_ML_30 DY1_ML_30 DY2_ML_30;
BY REPS;
Run;

```

```

/* Creating P values for Z tests of Y1 and Y2 */
Data output_30; set output_30;
Merge INTY1_ML_30 INTY2_ML_30 DY1_ML_30 DY2_ML_30 ;
BY REPS;
if t_DY1_ML_30 < 0 then
do;
    pval_Zy1_30 = 2*cdf("Normal", t_DY1_ML_30);
end;
else

```



```

do;
    pval_Zy1_30 = 2*(1-cdf("Normal", t_DY1_ML_30));
end;
if pval_Zy1_30 <= .05 then SigZy1_30 = 1;
else SigZy1_30 = 0;
if t_DY2_ML_30 < 0 then
do;
    pval_Zy2_30 = 2*cdf("Normal", t_DY2_ML_30);
end;
else
do;
    pval_Zy2_30 = 2*(1-cdf("Normal", t_DY2_ML_30));
end;
if pval_Zy1_30 <= .05 then SigZy1_30 = 1;
else SigZy1_30 = 0;
if pval_Zy2_30 <= .05 then SigZy2_30 = 1;
else SigZy2_30 = 0;
if pval_Zy1_30 <= .025 or pval_Zy2_30 <= .025 then OVERALL_ML_30 = 1;
Else OVERALL_ML_30 = 0;
Run;

Data output_30; set output_30;
TMeanY1_ML_30 = DY1_ML_30 + INTY1_ML_30;
TMeanY2_ML_30 = DY2_ML_30 + INTY2_ML_30;
Run;

/* Random Effects */
Data VarY1_ML_30 CoVar12_ML_30 VarY2_ML_30 ; Set Mvmm_random_fullML_30;
If covparm = 'UN(1,1)' then output VarY1_ML_30;
If covparm = 'UN(2,1)' then output CoVar12_ML_30;
If covparm = 'UN(2,2)' then output VarY2_ML_30;
run;

Data VarY1_ML_30; set VarY1_ML_30;
label Estimate = VarY1_ML_30;
label StdErr = SE_VarY1_ML_30;
label Probz = P_VarY1_ML_30;
rename Estimate = VarY1_ML_30;
rename StdErr = SE_VarY1_ML_30;
rename Probz = P_VarY1_ML_30;
Drop Subject Covparm ZValue;
Run;

Data VarY2_ML_30; set VarY2_ML_30;
label Estimate = VarY2_ML_30;
label StdErr = SE_VarY2_ML_30;
label Probz = P_VarY2_ML_30;
rename Estimate = VarY2_ML_30;
rename StdErr = SE_VarY2_ML_30;
rename Probz = P_VarY2_ML_30;
Drop Subject Covparm ZValue;
Run;

Data CoVar12_ML_30; set CoVar12_ML_30;

```

```

label Estimate = CoVar12_ML_30;
label StdErr = SE_CoVar12_ML_30;
label Probz = P_CoVar12_ML_30;
rename Estimate = CoVar12_ML_30;
rename StdErr = SE_CoVar12_ML_30;
rename Probz = P_CoVar12_ML_30;
Drop Subject Covparm ZValue;
Run;

```

```

Data output_30; Set output_30;
Merge VarY1_ML_30 CoVar12_ML_30 VarY2_ML_30 ;
run;

```

```

/*****
*****

```

```

Linear Mixed Model with REML and Default Denominator df: 30%
Missingness

```

```

*****
*****/

```

```

PROC MIXED DATA=long_30 METHOD=REML NOCLPRINT COVTEST;
  BY reps;
  CLASS index ID newT;
  MODEL Y_30 = index newT*Index / NOINT SOLUTION CL;
  lsmeans newT*Index / slice = index cl diff;
  REPEATED index / SUBJECT = ID TYPE=UN R;
  ods output ConvergenceStatus = MVMM_Conv_def_30;
  ods output FitStatistics = MVMM_Fit_def_30;
  ods output CovParms = MVMM_Random_def_30;
  ods output LSMeans = MVMM_Means_def_30;
  ods output Diffs = MVMM_Diffs_def_30;
  ods output SolutionF = MVMM_Fixed_def_30;
  RUN;

```

```

/** convergence for Full Model KR**/
data MVMM_Conv_def_30; set MVMM_Conv_def_30;
drop Reason pdG;
label Status = Status_MVMM_def_30;
rename Status = Status_MVMM_def_30;
label pdH = R_MVMM_def_30;
rename pdH = R_MVMM_def_30;
run;

```

```

proc freq data = MVMM_Conv_def_30 noprint;
tables Reps*Status_MVMM_def_30*R_MVMM_def_30 / out=MVMM_Conv_def_30;
run;

```

```

Data MVMM_Conv_def_30; set MVMM_Conv_def_30;
label count = Conv_MVMM_def_30_Count;
rename count = Conv_MVMM_def_30_Count;
label percent = Conv_MVMM_def_30_Percent;
rename percent = Conv_MVMM_def_30_Percent;
run;

/** Means From Full Model With Def_30 df */
data MVMM_Means_def_30; retain Reps Parm; set MVMM_Means_def_30;
label index = Y;
label Estimate = Means_def_30;
label StdErr = SE_MEANS_def_30;
label Lower = CIL_MEANS_def_30;
label Upper = CIU_MEANS_def_30;
rename index = Y;
rename Estimate = Est_MEANS_def_30;
rename StdErr = SE_MEANS_def_30;
rename Lower = CIL_MEANS_def_30;
rename Upper = CIU_MEANS_def_30;
drop Effect tvalue Alpha probt df;
if INDEX = 1 then do;
    if newT = 0 then Parm = 'TMEAN(Y1def_30)';
    if newT = 1 then Parm = 'CMEAN(Y1def_30)';
end;
if INDEX = 2 then do;
    if newT = 0 then Parm = 'TMEAN(Y2def_30)';
    if newT = 1 then Parm = 'CMEAN(Y2def_30)';
end;
run;

/** Creating Separate Data Files: Means */
data CMEANY1def_30 TXMEANY1def_30 CMEANY2def_30 TXMEANY2def_30 ; set
MVMM_Means_def_30;
    If Parm = 'TMEAN(Y1def_30)' then output TXMEANY1def_30;
    If Parm = 'CMEAN(Y1def_30)' then output CMEANY1def_30;
    If Parm = 'TMEAN(Y2def_30)' then output TXMEANY2def_30;
    If Parm = 'CMEAN(Y2def_30)' then output CMEANY2def_30;
Run;

Data TXMEANY1def_30; Set TXMEANY1def_30 ;
label EST_MEANS_def_30 = TMeanY1_def_30;
label SE_MEANS_def_30 = SE_TMeanY1_def_30;
label CIL_MEANS_def_30 = CIL_TMEANY1_def_30;
label CIU_MEANS_def_30 = CUL_TMEANY1_def_30;
rename EST_MEANS_def_30 = TMeanY1_def_30;
rename SE_MEANS_def_30 = SE_TMeanY1_def_30;
rename CIL_MEANS_def_30 = CIL_TMEANY1_def_30;
rename CIU_MEANS_def_30 = CUL_TMEANY1_def_30;
DROP PARM Y;
RUN;

Data CMEANY1def_30; Set CMEANY1def_30 ;
label EST_MEANS_def_30 = CMeanY1_def_30;
label SE_MEANS_def_30 = SE_CMeanY1_def_30;
label CIL_MEANS_def_30 = CIL_CMEANY1_def_30;
label CIU_MEANS_def_30 = CUL_CMEANY1_def_30;

```

```

rename EST_MEANS_def_30 = CMeanY1_def_30;
rename SE_MEANS_def_30 = SE_CMeanY1_def_30;
rename CIL_MEANS_def_30 = CIL_CMEANY1_def_30;
rename CIU_MEANS_def_30 = CUL_CMEANY1_def_30;
DROP PARM Y;
RUN;

```

```

Data TXMEANY2def_30; Set TXMEANY2def_30 ;
label EST_MEANS_def_30 = TMeanY2_def_30;
label SE_MEANS_def_30 = SE_TMeanY2_def_30;
label CIL_MEANS_def_30 = CIL_TMEANY2_def_30;
label CIU_MEANS_def_30 = CUL_TMEANY2_def_30;
rename EST_MEANS_def_30 = TMeanY2_def_30;
rename SE_MEANS_def_30 = SE_TMeanY2_def_30;
rename CIL_MEANS_def_30 = CIL_TMEANY2_def_30;
rename CIU_MEANS_def_30 = CUL_TMEANY2_def_30;
DROP PARM Y;
RUN;

```

```

Data CMEANY2def_30; Set CMEANY2def_30 ;
label EST_MEANS_def_30 = CMeanY2_def_30;
label SE_MEANS_def_30 = SE_CMeanY2_def_30;
label CIL_MEANS_def_30 = CIL_CMEANY2_def_30;
label CIU_MEANS_def_30 = CUL_CMEANY2_def_30;
rename EST_MEANS_def_30 = CMeanY2_def_30;
rename SE_MEANS_def_30 = SE_CMeanY2_def_30;
rename CIL_MEANS_def_30 = CIL_CMEANY2_def_30;
rename CIU_MEANS_def_30 = CUL_CMEANY2_def_30;
DROP PARM Y;
RUN;

```

```

DATA output_30; set output_30;
Merge MVMM_Conv_def_30 TXMEANY1def_30 CMEANY1def_30 TXMEANY2def_30
CMEANY2def_30 ;
By Repts;
Run;

```

```

/* Fixed Effects From Full Model With def_30 */

```

```

data MVMM_Fixed_def_30; retain Repts Parm; set MVMM_Fixed_def_30;
label index = Y;
label Estimate = Est_MVMM_Fixed_def_30;
label StdErr = SE_MVMM_Fixed_def_30;
label Probt = P_MVMM_Fixed_def_30;
label Lower = CIL_MVMM_Fixed_def_30;
label Upper = CIU_MVMM_Fixed_def_30;
rename index = Y;
rename Estimate = Est_MVMM_Fixed_def_30;
rename StdErr = SE_MVMM_Fixed_def_30;
rename Probt = P_MVMM_Fixed_def_30;
rename Lower = CIL_MVMM_Fixed_def_30;
rename Upper = CIU_MVMM_Fixed_def_30;
drop Alpha;
if INDEX = 1 then do;
    if Effect = 'INDEX' then Parm = 'INT(Y1def_30)';
    if NewT = 0 then Parm = 'd(Y1def_30)';

```

```

end;
if INDEX = 2 then do;
    if Effect = 'INDEX' then Parm = 'INT(Y2def_30)';
    if NewT = 0 then Parm = 'd(Y2def_30)';
end;
run;

/** Creating Separate Data Files: Intercepts **/

data INTY1_def_30 INTY2_def_30 ; set MVMM_Fixed_def_30;
If EFFECT = 'INDEX' & Y = 1 then output INTY1_def_30;
Else if EFFECT = 'INDEX' & Y = 2 then output INTY2_def_30;
Run;

data INTY1_def_30 ; set INTY1_def_30;
label Est_MVMM_Fixed_def_30 = INTY1_def_30;
label SE_MVMM_Fixed_def_30 = SE_INTY1_def_30;
label P_MVMM_Fixed_def_30 = P_INTY1_def_30;
label CIL_MVMM_Fixed_def_30 = CIL_INTY1_def_30;
label CIU_MVMM_Fixed_def_30 = CIU_INTY1_def_30;
label tvalue = t_INTY1_def_30;
rename Est_MVMM_Fixed_def_30 = INTY1_def_30;
rename SE_MVMM_Fixed_def_30 = SE_INTY1_def_30;
rename P_MVMM_Fixed_def_30 = P_INTY1_def_30;
rename CIL_MVMM_Fixed_def_30 = CIL_INTY1_def_30;
rename CIU_MVMM_Fixed_def_30 = CIU_INTY1_def_30;
rename tvalue = t_INTY1_def_30;
DROP PARM EFFECT Y NewT df;
RUN;

data INTY2_def_30 ; set INTY2_def_30;
label Est_MVMM_Fixed_def_30 = INTY2_def_30;
label SE_MVMM_Fixed_def_30 = SE_INTY2_def_30;
label P_MVMM_Fixed_def_30 = P_INTY2_def_30;
label CIL_MVMM_Fixed_def_30 = CIL_INTY2_def_30;
label CIU_MVMM_Fixed_def_30 = CIU_INTY2_def_30;
label tvalue = t_INTY2_def_30;
rename Est_MVMM_Fixed_def_30 = INTY2_def_30;
rename SE_MVMM_Fixed_def_30 = SE_INTY2_def_30;
rename P_MVMM_Fixed_def_30 = P_INTY2_def_30;
rename CIL_MVMM_Fixed_def_30 = CIL_INTY2_def_30;
rename CIU_MVMM_Fixed_def_30 = CIU_INTY2_def_30;
rename tvalue = t_INTY2_def_30;
DROP PARM EFFECT Y NewT ;
RUN;

/** Creating Separate Data Files: Slopes **/

Data DY1_def_30 DY2_def_30 ; set MVMM_Fixed_def_30;
If Y = 1 & NewT = 0 then output DY1_def_30;
Else if Y = 2 & NewT = 0 then output DY2_def_30;
Run;

data DY1_def_30 ; set DY1_def_30;
label Est_MVMM_Fixed_def_30 = DY1_def_30;
label SE_MVMM_Fixed_def_30 = SE_DY1_def_30;

```

```

label P_MVMM_Fixed_def_30 = P_DY1_def_30;
label CIL_MVMM_Fixed_def_30 = CIL_DY1_def_30;
label CIU_MVMM_Fixed_def_30 = CIU_DY1_def_30;
label tvalue = t_DY1_ML;
label df = df_DY1_ML;
rename Est_MVMM_Fixed_def_30 = DY1_def_30;
rename SE_MVMM_Fixed_def_30 = SE_DY1_def_30;
rename P_MVMM_Fixed_def_30 = P_DY1_def_30;
rename CIL_MVMM_Fixed_def_30 = CIL_DY1_def_30;
rename CIU_MVMM_Fixed_def_30 = CIU_DY1_def_30;
rename tvalue = t_DY1_def_30;
rename df = df_DY1_def_30;
DROP PARM EFFECT Y;
RUN;

data DY2_def_30 ; set DY2_def_30;
label Est_MVMM_Fixed_def_30 = DY2_def_30;
label SE_MVMM_Fixed_def_30 = SE_DY2_def_30;
label P_MVMM_Fixed_def_30 = P_DY2_def_30;
label CIL_MVMM_Fixed_def_30 = CIL_DY2_def_30;
label CIU_MVMM_Fixed_def_30 = CIU_DY2_def_30;
label tvalue = t_DY2_def_30;
label df = df_DY2_def_30;
rename Est_MVMM_Fixed_def_30 = DY2_def_30;
rename SE_MVMM_Fixed_def_30 = SE_DY2_def_30;
rename P_MVMM_Fixed_def_30 = P_DY2_def_30;
rename CIL_MVMM_Fixed_def_30 = CIL_DY2_def_30;
rename CIU_MVMM_Fixed_def_30 = CIU_DY2_def_30;
rename tvalue = t_DY2_def_30;
rename df = df_DY2_def_30;
DROP PARM EFFECT Y;
RUN;

Data output_30; set output_30;
Merge INTY1_def_30 INTY2_def_30 DY1_def_30 DY2_def_30;
BY REPS;
Run;

Data output_30; set output_30;
if P_DY1_def_30 <= .05 then SigDY1_def_30 = 1;
else SigDY1_def_30 = 0;
if P_DY2_def_30 <= .05 then SigDY2_def_30 = 1;
else SigDY2_def_30 = 0;
if P_DY1_def_30 <= .025 or P_DY2_def_30 <= .025 then OVERALL_def_30 = 1;
Else OVERALL_def_30 = 0;
run;

/* Random Effects */
Data VarY1_def_30 CoVar12_def_30 VarY2_def_30 ; Set Mvmm_random_def_30;
If covparm = 'UN(1,1)' then output VarY1_def_30;
If covparm = 'UN(2,1)' then output CoVar12_def_30;
If covparm = 'UN(2,2)' then output VarY2_def_30;
run;

Data VarY1_def_30; set VarY1_def_30;
label Estimate = VarY1_def_30;

```

```

label StdErr = SE_VarY1_def_30;
label Probz = P_VarY1_def_30;
rename Estimate = VarY1_def_30;
rename StdErr = SE_VarY1_def_30;
rename Probz = P_VarY1_def_30;
Drop Subject Covparm ZValue;
Run;

```

```

Data VarY2_def_30; set VarY2_def_30;
label Estimate = VarY2_def_30;
label StdErr = SE_VarY2_def_30;
label Probz = P_VarY2_def_30;
rename Estimate = VarY2_def_30;
rename StdErr = SE_VarY2_def_30;
rename Probz = P_VarY2_def_30;
Drop Subject Covparm ZValue;
Run;

```

```

Data CoVar12_def_30; set CoVar12_def_30;
label Estimate = CoVar12_def_30;
label StdErr = SE_CoVar12_def_30;
label Probz = P_CoVar12_def_30;
rename Estimate = CoVar12_def_30;
rename StdErr = SE_CoVar12_def_30;
rename Probz = P_CoVar12_def_30;
Drop Subject Covparm ZValue;
Run;

```

```

Data output_30; Set output_30;
Merge VarY1_def_30 VarY2_def_30 CoVar12_def_30;
by reps;
run;

```

```

/*****
*****/

```

Linear Mixed Model with REML and KR Denominator df: 30% Missing

```

*****/

```

```

/* Syntax for Full Model With REML and KR Denominator df */
PROC MIXED DATA=long_30 METHOD=REML NOCLPRINT COVTEST;
  BY reps;
  CLASS index ID newT;
  MODEL Y_30 = index newT*index / DDFM = KR NOINT SOLUTION CL;
  lsmeans newT*index / slice = index cl diff;
  REPEATED index / SUBJECT = ID TYPE=UN R;
  ods output ConvergenceStatus = MVMM_Conv_KR_30;
  ods output FitStatistics = MVMM_Fit_KR_30;
  ods output CovParms = MVMM_Random_KR_30;

```

```

ods output LSMeans = MVMM_Means_KR_30;
ods output Diffs = MVMM_Diffs_KR_30;
ods output SolutionF = MVMM_Fixed_KR_30;
RUN;

/** convergence for Full Model KR**/
data MVMM_Conv_KR_30; set MVMM_Conv_KR_30;
drop Reason pdG;
label Status = Status_MVMM_KR_30;
rename Status = Status_MVMM_KR_30;
label pdH = R_MVMM_KR_30;
rename pdH = R_MVMM_KR_30;
run;

proc freq data = MVMM_Conv_KR_30 noprint;
tables Reps*Status_MVMM_KR_30*R_MVMM_KR_30 / out=MVMM_Conv_KR_30;
run;

Data MVMM_Conv_KR_30; set MVMM_Conv_KR_30;
label count = Conv_MVMM_KR_Count_30;
rename count = Conv_MVMM_KR_Count_30;
label percent = Conv_MVMM_KR_Percent_30;
rename percent = Conv_MVMM_KR_Percent_30;
run;

/** Means From Full Model With KR **/
data MVMM_Means_KR_30; retain Reps Parm; set MVMM_Means_KR_30;
label index = Y_30;
label Estimate = Means_KR_30;
label StdErr = SE_MEANS_KR_30;
label Lower = CIL_MEANS_KR_30;
label Upper = CIU_MEANS_KR_30;
rename index = Y_30;
rename Estimate = Est_MEANS_KR_30;
rename StdErr = SE_MEANS_KR_30;
rename Lower = CIL_MEANS_KR_30;
rename Upper = CIU_MEANS_KR_30;
drop Effect tvalue Alpha probt df;
if INDEX = 1 then do;
    if newT = 0 then Parm = 'TMEAN(Y1KR)_30';
    if newT = 1 then Parm = 'CMEAN(Y1KR)_30';
end;
if INDEX = 2 then do;
    if newT = 0 then Parm = 'TMEAN(Y2KR)_30';
    if newT = 1 then Parm = 'CMEAN(Y2KR)_30';
end;
run;

/** Creating Separate Data Files: Means **/
data CMEANY1KR_30 TXMEANY1KR_30 CMEANY2KR_30 TXMEANY2KR_30 ; set
MVMM_Means_KR_30;
If Parm = 'TMEAN(Y1KR)_30' then output TXMEANY1KR_30;
If Parm = 'CMEAN(Y1KR)_30' then output CMEANY1KR_30;
If Parm = 'TMEAN(Y2KR)_30' then output TXMEANY2KR_30;
If Parm = 'CMEAN(Y2KR)_30' then output CMEANY2KR_30;
Run;

```



```

Data TXMEANY1KR_30; Set TXMEANY1KR_30 ;
label EST_MEANS_KR_30 = TMeanY1_KR_30;
label SE_MEANS_KR_30 = SE_TMeanY1_KR_30;
label CIL_MEANS_KR_30 = CIL_TMEANY1_KR_30;
label CIU_MEANS_KR_30 = CUL_TMEANY1_KR_30;
rename EST_MEANS_KR_30 = TMeanY1_KR_30;
rename SE_MEANS_KR_30 = SE_TMeanY1_KR_30;
rename CIL_MEANS_KR_30 = CIL_TMEANY1_KR_30;
rename CIU_MEANS_KR_30 = CUL_TMEANY1_KR_30;
DROP PARM Y_30;
RUN;

```

```

Data CMEANY1KR_30; Set CMEANY1KR_30 ;
label EST_MEANS_KR_30 = CMeanY1_KR_30;
label SE_MEANS_KR_30 = SE_CMeanY1_KR_30;
label CIL_MEANS_KR_30 = CIL_CMEANY1_KR_30;
label CIU_MEANS_KR_30 = CUL_CMEANY1_KR_30;
rename EST_MEANS_KR_30 = CMeanY1_KR_30;
rename SE_MEANS_KR_30 = SE_CMeanY1_KR;
rename CIL_MEANS_KR_30 = CIL_CMEANY1_KR_30;
rename CIU_MEANS_KR_30 = CUL_CMEANY1_KR_30;
DROP PARM Y_30;
RUN;

```

```

Data TXMEANY2KR_30; Set TXMEANY2KR_30 ;
label EST_MEANS_KR_30 = TMeanY2_KR_30;
label SE_MEANS_KR_30 = SE_TMeanY2_KR_30;
label CIL_MEANS_KR_30 = CIL_TMEANY2_KR_30;
label CIU_MEANS_KR_30 = CUL_TMEANY2_KR_30;
rename EST_MEANS_KR_30 = TMeanY2_KR_30;
rename SE_MEANS_KR_30 = SE_TMeanY2_KR_30;
rename CIL_MEANS_KR_30 = CIL_TMEANY2_KR_30;
rename CIU_MEANS_KR_30 = CUL_TMEANY2_KR_30;
DROP PARM Y_30;
RUN;

```

```

Data CMEANY2KR_30; Set CMEANY2KR_30 ;
label EST_MEANS_KR_30 = CMeanY2_KR_30;
label SE_MEANS_KR_30 = SE_CMeanY2_KR_30;
label CIL_MEANS_KR_30 = CIL_CMEANY2_KR_30;
label CIU_MEANS_KR_30 = CUL_CMEANY2_KR_30;
rename EST_MEANS_KR_30 = CMeanY2_KR_30;
rename SE_MEANS_KR_30 = SE_CMeanY2_KR_30;
rename CIL_MEANS_KR_30 = CIL_CMEANY2_KR_30;
rename CIU_MEANS_KR_30 = CUL_CMEANY2_KR_30;
DROP PARM Y_30;
RUN;

```

```

DATA output_30; set output_30;
Merge MVMM_Conv_KR_30 TXMEANY1KR_30 CMEANY1KR_30 TXMEANY2KR_30 CMEANY2KR_30;
By Reps;
Run;

```

```

Data MVMM_Fixed_KR_30; retain Reps Parm; set MVMM_Fixed_KR_30;
label index = Y_KR_30;
label Estimate = Est_MVMM_Fixed_KR_30;

```

```

label StdErr = SE_MVMM_Fixed_KR_30;
label Probt = P_MVMM_Fixed_KR_30;
label Lower = CIL_MVMM_Fixed_KR_30;
label Upper = CIU_MVMM_Fixed_KR_30;
rename index = Y_KR_30;
rename Estimate = Est_MVMM_Fixed_KR_30;
rename StdErr = SE_MVMM_Fixed_KR_30;
rename Probt = P_MVMM_Fixed_KR_30;
rename Lower = CIL_MVMM_Fixed_KR_30;
rename Upper = CIU_MVMM_Fixed_KR_30;
drop Alpha;
if INDEX = 1 then do;
    if Effect = 'INDEX' then Parm = 'INT(Y1KR_30)';
    if NewT = 0 then Parm = 'd(Y1KR_30)';
end;
if INDEX = 2 then do;
    if Effect = 'INDEX' then Parm = 'INT(Y2KR_30)';
    if NewT = 0 then Parm = 'd(Y2KR_30)';
end;
run;

/** Creating Separate Data Files: Intercepts */

Data INTY1_KR_30 INTY2_KR_30 ; set MVMM_Fixed_KR_30;
If EFFECT = 'INDEX' & Y_KR_30 = 1 then output INTY1_KR_30;
Else if EFFECT = 'INDEX' & Y_KR_30 = 2 then output INTY2_KR_30;
Run;

Data INTY1_KR_30 ; set INTY1_KR_30;
label Est_MVMM_Fixed_KR_30 = INTY1_KR_30;
label SE_MVMM_Fixed_KR_30 = SE_INTY1_KR_30;
label P_MVMM_Fixed_KR_30 = P_INTY1_KR_30;
label CIL_MVMM_Fixed_KR_30 = CIL_INTY1_KR_30;
label CIU_MVMM_Fixed_KR_30 = CIU_INTY1_KR_30;
label tvalue = t_INTY1_KR_30;
rename Est_MVMM_Fixed_KR_30 = INTY1_KR_30;
rename SE_MVMM_Fixed_KR_30 = SE_INTY1_KR_30;
rename P_MVMM_Fixed_KR_30 = P_INTY1_KR_30;
rename CIL_MVMM_Fixed_KR_30 = CIL_INTY1_KR_30;
rename CIU_MVMM_Fixed_KR_30 = CIU_INTY1_KR_30;
rename tvalue = t_INTY1_KR_30;
DROP PARM EFFECT Y_KR_30 NewT df;
RUN;

Data INTY2_KR_30 ; set INTY2_KR_30;
label Est_MVMM_Fixed_KR_30 = INTY2_KR_30;
label SE_MVMM_Fixed_KR_30 = SE_INTY2_KR_30;
label P_MVMM_Fixed_KR_30 = P_INTY2_KR_30;
label CIL_MVMM_Fixed_KR_30 = CIL_INTY2_KR_30;
label CIU_MVMM_Fixed_KR_30 = CIU_INTY2_KR_30;
label tvalue = t_INTY2_KR_30;
rename Est_MVMM_Fixed_KR_30 = INTY2_KR_30;
rename SE_MVMM_Fixed_KR_30 = SE_INTY2_KR_30;
rename P_MVMM_Fixed_KR_30 = P_INTY2_KR_30;
rename CIL_MVMM_Fixed_KR_30 = CIL_INTY2_KR_30;

```

```

rename CIU_MVMM_Fixed_KR_30 =CIU_INTY2_KR_30;
rename tvalue = t_INTY2_KR_30;
DROP PARM EFFECT Y_KR_30 Newt ;
RUN;

```

```

/** Creating Separate Data Files: Slopes */

```

```

Data DY1_KR_30 DY2_KR_30 ; set MVMM_Fixed_KR_30;
If Y_KR_30 = 1 & NewT = 0 then output DY1_KR_30;
Else if Y_KR_30 = 2 & NewT = 0 then output DY2_KR_30;
Run;

```

```

Data DY1_KR_30 ; set DY1_KR_30;
label Est_MVMM_Fixed_KR_30 = DY1_KR_30;
label SE_MVMM_Fixed_KR_30 = SE_DY1_KR_30;
label P_MVMM_Fixed_KR_30 = P_DY1_KR_30;
label CIL_MVMM_Fixed_KR_30 = CIL_DY1_KR_30;
label CIU_MVMM_Fixed_KR_30 = CIU_DY1_KR_30;
label tvalue = t_DY1_KR_30;
label df = df_DY1_KR_30;
rename Est_MVMM_Fixed_KR_30 = DY1_KR_30;
rename SE_MVMM_Fixed_KR_30 = SE_DY1_KR_30;
rename P_MVMM_Fixed_KR_30 = P_DY1_KR_30;
rename CIL_MVMM_Fixed_KR_30 = CIL_DY1_KR_30;
rename CIU_MVMM_Fixed_KR_30 =CIU_DY1_KR_30;
rename tvalue = t_DY1_KR_30;
rename df = df_DY1_KR_30;
DROP PARM EFFECT Y_KR_30;
RUN;

```

```

Data DY2_KR_30 ; set DY2_KR_30;
label Est_MVMM_Fixed_KR_30 = DY2_KR_30;
label SE_MVMM_Fixed_KR_30 = SE_DY2_KR_30;
label P_MVMM_Fixed_KR_30 = P_DY2_KR_30;
label CIL_MVMM_Fixed_KR_30 = CIL_DY2_KR_30;
label CIU_MVMM_Fixed_KR_30 = CIU_DY2_KR_30;
label tvalue = t_DY2_KR_30;
label df = df_DY2_KR_30;
rename Est_MVMM_Fixed_KR_30 = DY2_KR_30;
rename SE_MVMM_Fixed_KR_30 = SE_DY2_KR_30;
rename P_MVMM_Fixed_KR_30 = P_DY2_KR_30;
rename CIL_MVMM_Fixed_KR_30 = CIL_DY2_KR_30;
rename CIU_MVMM_Fixed_KR_30 = CIU_DY2_KR_30;
rename tvalue = t_DY2_KR_30;
rename df = df_DY2_KR_30;
DROP PARM EFFECT Y_KR_30;
RUN;

```

```

Data output_30; set output_30;
Merge INTY1_KR_30 INTY2_KR_30 DY1_KR_30 DY2_KR_30 ;
BY REPS;
Run;

```

```

Data output_30; set output_30;

```

```

if P_DY1_KR_30 <= .05 then SigDY1_KR_30 = 1;
else SigDY1_KR_30 = 0;
if P_DY2_KR_30 <= .05 then SigDY2_KR_30 = 1;
else SigDY2_KR_30 = 0;
if P_DY1_KR_30 <= .025 or P_DY2_KR_30 <= .025 then OVERALL_KR_30 = 1;
Else OVERALL_KR_30 = 0;
run;

```

```

/* Random Effects */

```

```

Data VarY1_KR_30 CoVar12_KR_30 VarY2_KR_30 ; Set Mvmm_random_KR_30;
If covparm = 'UN(1,1)' then output VarY1_KR_30;
If covparm = 'UN(2,1)' then output CoVar12_KR_30;
If covparm = 'UN(2,2)' then output VarY2_KR_30;
run;

```

```

Data VarY1_KR_30; set VarY1_KR_30;
label Estimate = VarY1_KR_30;
label StdErr = SE_VarY1_KR_30;
label Probz = P_VarY1_KR_30;
rename Estimate = VarY1_KR_30;
rename StdErr = SE_VarY1_KR_30;
rename Probz = P_VarY1_KR_30;
Drop Subject Covparm ZValue;
Run;

```

```

Data VarY2_KR_30; set VarY2_KR_30;
label Estimate = VarY2_KR_30;
label StdErr = SE_VarY2_KR_30;
label Probz = P_VarY2_KR_30;
rename Estimate = VarY2_KR_30;
rename StdErr = SE_VarY2_KR_30;
rename Probz = P_VarY2_KR_30;
Drop Subject Covparm ZValue;
Run;

```

```

Data CoVar12_KR_30; set CoVar12_KR_30;
label Estimate = CoVar12_KR_30;
label StdErr = SE_CoVar12_KR_30;
label Probz = P_CoVar12_KR_30;
rename Estimate = CoVar12_KR_30;
rename StdErr = SE_CoVar12_KR_30;
rename Probz = P_CoVar12_KR_30;
Drop Subject Covparm ZValue;
Run;

```

```

Data output_30; Set output_30;
Merge VarY1_KR_30 VarY2_KR_30 CoVar12_KR_30 newlineconds;
run;

```

```

/* Forming Relative Bias Measures */

```

```

DATA output_30; set output_30;
P_Rel_Bias_MSEY1_30 = (MSEY1_30 - var1)/var1;
P_Rel_Bias_MSEY2_30 = (MSEY2_30 - var2)/var2;

```

```

P_Rel_Bias_MSEY2_PAIR_30 = (MSEY2_PAIR_30 - var2)/var2;
P_Rel_Bias_COVAR12_GLM_30 = (covar12_30 - r12)/r12;
P_Rel_Bias_VarY1_ML_30 = (VarY1_ML_30 - var1)/var1;
P_Rel_Bias_VarY2_ML_30 = (VarY2_ML_30 - var2)/var2;
P_Rel_Bias_COVAR12_ML_30 = (CoVar12_ML_30 - r12)/r12;
P_Rel_Bias_VarY1_KR_30 = (VarY1_KR_30 - var1)/var1;
P_Rel_Bias_VarY2_KR_30 = (VarY2_KR_30 - var2)/var2;
P_Rel_Bias_CoVar12_KR_30 = (CoVar12_KR_30 - r12)/r12;
P_Rel_Bias_TMeanY1_OLS_30 = (TmeanY1_30 - EsY1)/EsY1;
P_Rel_Bias_TMeanY2_OLS_30 = (TMeanY2_30 - EsY2)/EsY2;
P_Rel_Bias_TMeanY2_PAIR_30 = (TMean_Y2_PAIR_30 - EsY2)/EsY2;
P_Rel_Bias_TMeanY1_ML_30 = (TMeanY1_ML_30 - EsY1)/EsY1;
P_Rel_Bias_TMeanY2_ML_30 = (TMeanY2_ML_30 - EsY2)/EsY2;
P_Rel_Bias_TMeanY1_def_30 = (TMeanY1_def_30 - EsY1)/EsY1;
P_Rel_Bias_TMeanY2_def_30 = (TMeanY2_def_30 - EsY2)/EsY2;
P_Rel_Bias_TMeanY1_KR_30 = (TMeanY1_KR_30 - EsY1)/EsY1;
P_Rel_Bias_TMeanY2_KR_30 = (TMeanY2_KR_30 - EsY2)/EsY2;
P_Rel_Bias_OLSDiffY1_30 = (DiffY1_30 - EsY1)/EsY1;
P_Rel_Bias_OLSDiffY2_30 = (DiffY2_30 - EsY2)/EsY2;
P_Rel_Bias_DiffY2_PAIR_30 = (Diff_Y2_PAIR_30 - EsY2)/EsY2;
P_Rel_Bias_MLDiffY1_30 = (DY1_ML_30 - EsY1)/EsY1;
P_Rel_Bias_MLDiffY2_30 = (DY2_ML_30 - EsY2)/EsY2;
P_Rel_Bias_DEFDiffY1_30 = (DY1_def_30 - EsY1)/EsY1;
P_Rel_Bias_DEFDiffY2_30 = (DY2_def_30 - EsY2)/EsY2;
P_Rel_Bias_KRDiffY1_30 = (dy1_KR_30 - EsY1)/EsY1;
P_Rel_Bias_KRDiffY2_30 = (dy2_KR_30 - EsY2)/EsY2;
Run;

```

```

/* Creating CI coverage variables */

```

```

DATA Output_30; SET output_30;
if CCILowerMeanY2pair_30 <=0 and CCIUpperMeanY2pair_30 >= 0 then
OLS_Capture_CONMEANY2pair_30 = 1;
else OLS_Capture_CONMEANY2pair_30 = 0;
if TCILowerMeanY2pair_30 <=EsY2 and TCIUpperMeanY2pair_30 >= EsY2 then
OLS_Capture_TMEANY2pair_30 = 1;
else OLS_Capture_TMEANY2pair_30 = 0;
if CCILowerMeanY1_30 <=0 and CCIUpperMeanY1_30 >= 0 then
OLS_Capture_CONMEANY1_30 = 1;
else OLS_Capture_CONMEANY1_30 = 0;
if CCILowerMeanY2_30 <=0 and CCIUpperMeanY2_30 >= 0 then
OLS_Capture_CONMEANY2_30 = 1;
else OLS_Capture_CONMEANY2_30 = 0;
if TCILowerMeanY1_30 <=EsY1 and TCIUpperMeanY1_30 >= EsY1 then
OLS_Capture_TMEANY1_30 = 1;
else OLS_Capture_TMEANY1_30 = 0;
if TCILowerMeanY2_30 <=EsY2 and TCIUpperMeanY2_30 >= EsY1 then
OLS_Capture_TMEANY2_30 = 1;
else OLS_Capture_TMEANY2_30 = 0;
if lowerCLY1_30 <=EsY1 and upperCLY1_30 >= EsY1 then OLS_Capture_DIFFY1_30 =
1;
else OLS_Capture_DIFFY1_30 = 0;
if lowerCLY2_30 <=EsY2 and upperCLY2_30 >= EsY2 then OLS_Capture_DIFFY2_30 =
1;
else OLS_Capture_DIFFY2_30 = 0;

```

```

if lowerCLY2_PAIR_30 <=EsY2 and upperCLY2_PAIR_30 >= EsY2 then
OLS_Capture_DIFFY2_PAIR_30 = 1;
else OLS_Capture_DIFFY2_PAIR_30 = 0;
if CIL_DY1_ML_30 <=EsY1 and CIU_DY1_ML_30 >= EsY1 then ML_Capture_DIFFY1_30
= 1;
else ML_Capture_DIFFY1_30 = 0;
if CIL_DY2_ML_30 <=EsY2 and CIU_DY2_ML_30 >= EsY2 then ML_Capture_DIFFY2_30 =
1;
else ML_Capture_DIFFY2_30 = 0;
if CIL_DY1_DEF_30 <=EsY1 and CIU_DY1_DEF_30 >= EsY1 then
DEF_Capture_DIFFY1_30 = 1;
else DEF_Capture_DIFFY1_30 = 0;
if CIL_DY2_DEF_30 <=EsY2 and CIU_DY2_DEF_30 >= EsY2 then
DEF_Capture_DIFFY2_30 = 1;
else DEF_Capture_DIFFY2_30 = 0;
if CIL_DY1_KR_30 <=EsY1 and CIU_DY1_KR_30 >= EsY1 then KR_Capture_DIFFY1_30
= 1;
else KR_Capture_DIFFY1_30 = 0;
if CIL_DY2_KR_30 <= EsY2 and CIU_DY2_KR_30 >= EsY2 then
KR_Capture_DIFFY2_30 = 1;
else KR_Capture_DIFFY2_30 = 0;
run;

```

```

/* Merging all output */
Data outputmerged;
Merge Outputc Output_15 Output_30;
Run;

```

```

%put END TIME: %sysfunc(datetime(),datetime14.);
%put PROCESSING TIME: %sysfunc(putn(%sysvalf(%sysfunc(TIME())-
&datetime_start.),mmss.)) (mm:ss);
%put &cond;

```

```

proc append base=OUT.alloutput data=outputmerged; /* accumulate statistics */
run; quit;

```

```

%END;
%MEND SCANLOOP;

```

```

/* Call SCANLOOP macro */
%SCANLOOP(Allconds,Cond,Var1,r12,r21,Var2,Esy1,Esy2,N, reps, seed,N1,N2);
RUN;

```

```

/*****

```

END OF PROGRAM

```

*****/

```

