WREM Response Surface Model and Simulation R Code

This document provides the R Code for:

- Generating a 13^6 component response surface array using WREM parameters derived from Study 2 mixed model regression coefficients
- Evaluating the response surface data to identify summary statistics and statistics by personality combination blocks
- Generating 10,000 replicates of a 3^6 component response surface array using WREM parameters derived from Study 2 mixed model regression coefficients with induced variance for independent variables
- Evaluating the stochastic response surface replicate data to identify summary statistics and statistics by personality combination blocks
- Evaluating stochastic response surface replicate data to identify probability of situational combinations producing maximum response values for each personality combination block
- Extracting optimized response data from the stochastic response surface replicate data
- Evaluating optimized response data to identify to identify summary statistics and statistics by personality combination blocks
- Comparing summary statistics for non-optimized and optimized response surface data by T-test

NOTES:

- No external data is required.
- All lines beginning with "#" character are ignored by the R console and may be pasted directly in the console with functional lines of code.
- This code was prepared and implemented in R version 3.6.1 (2019-07-05).
- See R project for details at https://www.r-project.org/.

```
## ESTABLISH DETERMINISITIC RESPONSE SURFACE ARRAY


# INITIATE R PACKAGES FOR MODELLING AND ANALYSIS

library(raster)

library(dplyr)


#Set Input Parameters


    q<-13 # SETS NUMBER OF SAMPLE POINTS ON PREDICTOR INTERVALS


    ## SET PREDICTOR INTERVALS AND SAMPLE UNIFORMLY ACROSS INTERVALS

    Pval<-seq(mean(P)-1.5*sd(P),mean(P)+1.5*sd(P),len=q) # THE Psych FACTOR
```

```
Eval<-seq(mean(E)-1.5*sd(E),mean(E)+1.5*sd(E),len=q) # THE Extrav FACTOR
Nval<-seq(mean(N)-1.5*sd(N),mean(N)+1.5*sd(N),len=q) # THE Neuro FACTOR
Cval<-seq(-1.5,1.5,len=q) # THE EnvVStim FACTOR
Sval<-seq(-1.5,1.5,len=q) # THE ProStruc FACTOR
Gval<-seq(-1.5,1.5,len=q) # THE SptGrp FACTOR

## GENERATE RESPONSE SURFACE ARRAY

DetDxEffectArray<-array(data = NA, dim = c(q,q,q,q,q,q), dimnames = NULL) # CREATES ARRAY

## SIMULATE FOR DETERMINISTIC RESPONSE SURFACE WITH q^6 DATA POINTS
# These Six Loops Ensure All Combinations Of Sampled Points For Each Factor Are Examined
for (i in 1:q){
    for (j in 1:q){
        for (k in 1:q){
          for (l in 1:q){
              for (m in 1:q){
                  for (n in 1:q){

DetDxEffectArray[i,j,k,l,m,n]<-(Study2MMreduxrpt2x[1] ## Intercept Term
+Study2MMreduxrpt2x[2]*Pval[i] ## P
+Study2MMreduxrpt2x[3]*Eval[j]## E
+Study2MMreduxrpt2x[4]*Nval[k]## N
+Study2MMreduxrpt2x[5]*Cval[l] ## C
+Study2MMreduxrpt2x[6]*Sval[m]## S
+Study2MMreduxrpt2x[7]*Gval[n]## G
+Study2MMreduxrpt2x[8]*Pval[i]*Cval[l]  #P*C
+Study2MMreduxrpt2x[9]*Pval[i]*Sval[m] #PS
```

```
                    +Study2MMreduxrpt2x[10]*Pval[i]*Gval[n]#PG
                    +Study2MMreduxrpt2x[11]*Eval[j]*Cval[l]#EC
                    +Study2MMreduxrpt2x[12]*Eval[j]*Gval[n]#EG
                    +Study2MMreduxrpt2x[13]*Nval[k]*Cval[l]#NC
                    +Study2MMreduxrpt2x[14]*Nval[k]*Gval[n] ## NG
                    +Study2MMreduxrpt2x[15]*Cval[l]*Gval[n] ## CG
                    +Study2MMreduxrpt2x[16]*Pval[i]*Cval[l]*Gval[n] ## PCG
                    +Study2MMreduxrpt2x[17]*Eval[j]*Cval[l]*Gval[n] ## ECG
                    +Study2MMreduxrpt2x[18]*Nval[k]*Cval[l]*Gval[n]) ## NCG
}}}}}} ##End Simulation

## IDENTIFY AND EVALUATE SUMMARY STATISTICS FOR RESPONSE VALUES
max(DetDxEffectArray)
min(DetDxEffectArray)
mean(DetDxEffectArray)
var(DetDxEffectArray)

## IDENTIFY ARRAY INDEX OF MAXIMUM & MINIMUM RESPONSE VALUES
which(DetDxEffectArray==max(DetDxEffectArray),arr.ind = T)
which(DetDxEffectArray==min(DetDxEffectArray),arr.ind = T)

## IDENTIFY MAXIMUM VALUES BY SITUATIONAL COMBINATION FOR EACH PERSONALITY BLOCK

## THIS IDENTIFIES THE ARRAY INDEX FOR MAXIMUM VALUES IDENITIFED IN ARRAY SUBSETS REPRESENTING PERSONALITY FACTOR COMBINATIONS
# ON THREE PERSONALITY SUB-INTERVALS

# ON THREE SUB-INTERVALS
# LOW STRENGTH SUB -INTERVALS ARE INDEXED FROM 1 TO 4
```

```
# MODERATE STRENGTH SUB -INTERVALS ARE INDEXED FROM 5 TO 9
# HIGH STRENGTH SUB -INTERVALS ARE INDEXED FROM 5 TO 9
maxrpt<-rbind(
which(DetDxEffectArray==max(DetDxEffectArray[1:4,1:4,1:4,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[1:4,5:9,1:4,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[1:4,10:13,1:4,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[1:4,1:4,5:9,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[1:4,5:9,5:9,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[1:4,10:13,5:9,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[1:4,5:9,10:13,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[1:4,10:13,10:13,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[5:9,10:13,1:4,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[5:9,10:13,5:9,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[5:9,10:13,10:13,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[1:4,1:4,10:13,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[5:9,1:4,1:4,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[10:13,1:4,1:4,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[10:13,10:13,1:4,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[10:13,5:9,1:4,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[5:9,1:4,5:9,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[10:13,1:4,5:9,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[5:9,5:9,5:9,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[10:13,5:9,5:9,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[10:13,10:13,5:9,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[5:9,1:4,10:13,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[10:13,1:4,10:13,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[5:9,5:9,10:13,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[10:13,5:9,10:13,,,]),arr.ind = T),
```

```
which(DetDxEffectArray==max(DetDxEffectArray[10:13,10:13,10:13,,,]),arr.ind = T),
which(DetDxEffectArray==max(DetDxEffectArray[5:9,5:9,1:4,,,]),arr.ind = T)
) # Closes list for rbind

maxrpt # REPORTS INDEX FOR MAXIMUM VALUES ACROSS PERSONALITY COMBINATION ARRAY SUBSETS

## THIS EVALUATES THE MEAN VALUES FOR ARRAY SUBSETS REPRESENTING PERSONALITY FACTOR COMBINATIONS
# ON THREE PERSONALITY SUB-INTERVALS
DetSimBlockMeans<-c(
penMean<-mean(DetDxEffectArray[1:4,1:4,1:4,,,]),
pXnMean<-mean(DetDxEffectArray[1:4,5:9,1:4,,,]),
pEnMean<-mean(DetDxEffectArray[1:4,10:13,1:4,,,]),
peXMean<-mean(DetDxEffectArray[1:4,1:4,5:9,,,]),
pXXMean<-mean(DetDxEffectArray[1:4,5:9,5:9,,,]),
pEXMean<-mean(DetDxEffectArray[1:4,10:13,5:9,,,]),
peNMean<-mean(DetDxEffectArray[1:4,1:4,10:13,,,]),
pXNMean<-mean(DetDxEffectArray[1:4,5:9,10:13,,,]),
pENMean<-mean(DetDxEffectArray[1:4,10:13,10:13,,,]),
XenMean<-mean(DetDxEffectArray[5:9,1:4,1:4,,,]),
PenMean<-mean(DetDxEffectArray[10:13,1:4,1:4,,,]),
PEnMean<-mean(DetDxEffectArray[10:13,10:13,1:4,,,]),
PXnMean<-mean(DetDxEffectArray[10:13,5:9,1:4,,,]),
XeXMean<-mean(DetDxEffectArray[5:9,1:4,5:9,,,]),
PeXMean<-mean(DetDxEffectArray[10:13,1:4,5:9,,,]),
XXXMean<-mean(DetDxEffectArray[5:9,5:9,5:9,,,]),
PXXMean<-mean(DetDxEffectArray[10:13,5:9,5:9,,,]),
PEXMean<-mean(DetDxEffectArray[10:13,10:13,5:9,,,]),
XeNMean<-mean(DetDxEffectArray[5:9,1:4,10:13,,,]),
```

```
PeNMean<-mean(DetDxEffectArray[10:13,1:4,10:13,,,]),
XXNMean<-mean(DetDxEffectArray[5:9,5:9,10:13,,,]),
PXNMean<-mean(DetDxEffectArray[10:13,5:9,10:13,,,]),
PENMean<-mean(DetDxEffectArray[10:13,10:13,10:13,,,]),
XEnMean<-mean(DetDxEffectArray[5:9,10:13,1:4,,,]),
XXnMean<-mean(DetDxEffectArray[5:9,5:9,1:4,,,]),
XEXMean<-mean(DetDxEffectArray[5:9,10:13,5:9,,,]),
XENMean<-mean(DetDxEffectArray[5:9,10:13,10:13,,,]))

DetSimBlockMeans # REPORTS MEAN VALUES ACROSS PERSONALITY COMBINATION ARRAY SUBSETS

## THIS EVALUATES THE VARIANCE FOR ARRAY SUBSETS REPRESENTING PERSONALITY FACTOR COMBINATIONS
# ON THREE PERSONALITY SUB-INTERVALS
DetSimBlockVars<-c(
penVar<-var(DetDxEffectArray[1:4,1:4,1:4,,,]),
pXnVar<-var(DetDxEffectArray[1:4,5:9,1:4,,,]),
pEnVar<-var(DetDxEffectArray[1:4,10:13,1:4,,,]),
peXVar<-var(DetDxEffectArray[1:4,1:4,5:9,,,]),
pXXVar<-var(DetDxEffectArray[1:4,5:9,5:9,,,]),
pEXVar<-var(DetDxEffectArray[1:4,10:13,5:9,,,]),
peNVar<-var(DetDxEffectArray[1:4,1:4,10:13,,,]),
pXNVar<-var(DetDxEffectArray[1:4,5:9,10:13,,,]),
pENVar<-var(DetDxEffectArray[1:4,10:13,10:13,,,]),
XenVar<-var(DetDxEffectArray[5:9,1:4,1:4,,,]),
PenVar<-var(DetDxEffectArray[10:13,1:4,1:4,,,]),
PEnVar<-var(DetDxEffectArray[10:13,10:13,1:4,,,]),
PXnVar<-var(DetDxEffectArray[10:13,5:9,1:4,,,]),
XeXVar<-var(DetDxEffectArray[5:9,1:4,5:9,,,]),
```

```
PeXVar<-var(DetDxEffectArray[10:13,1:4,5:9,,,]),
XXXVar<-var(DetDxEffectArray[5:9,5:9,5:9,,,]),
PXXVar<-var(DetDxEffectArray[10:13,5:9,5:9,,,]),
PEXVar<-var(DetDxEffectArray[10:13,10:13,5:9,,,]),
XeNVar<-var(DetDxEffectArray[5:9,1:4,10:13,,,]),
PeNVar<-var(DetDxEffectArray[10:13,1:4,10:13,,,]),
XXNVar<-var(DetDxEffectArray[5:9,5:9,10:13,,,]),
PXNVar<-var(DetDxEffectArray[10:13,5:9,10:13,,,]),
PENVar<-var(DetDxEffectArray[10:13,10:13,10:13,,,]),
XEnVar<-var(DetDxEffectArray[5:9,10:13,1:4,,,]),
XXnVar<-var(DetDxEffectArray[5:9,5:9,1:4,,,]),
XEXVar<-var(DetDxEffectArray[5:9,10:13,5:9,,,]),
XENVar<-var(DetDxEffectArray[5:9,10:13,10:13,,,]))

DetSimBlockVars # REPORTS VARIANCE ACROSS PERSONALITY COMBINATION ARRAY SUBSETS

## THIS EVALUATES THE MAXIMUM VALUES FOR ARRAY SUBSETS REPRESENTING PERSONALITY FACTOR COMBINATIONS
# ON THREE PERSONALITY SUB-INTERVALS
DetSimBlockMax<-c(
penMax<-max(DetDxEffectArray[1:4,1:4,1:4,,,]),
pXnMax<-max(DetDxEffectArray[1:4,5:9,1:4,,,]),
pEnMax<-max(DetDxEffectArray[1:4,10:13,1:4,,,]),
peXMax<-max(DetDxEffectArray[1:4,1:4,5:9,,,]),
pXXMax<-max(DetDxEffectArray[1:4,5:9,5:9,,,]),
pEXMax<-max(DetDxEffectArray[1:4,10:13,5:9,,,]),
peNMax<-max(DetDxEffectArray[1:4,1:4,10:13,,,]),
pXNMax<-max(DetDxEffectArray[1:4,5:9,10:13,,,]),
pENMax<-max(DetDxEffectArray[1:4,10:13,10:13,,,]),
```

```
XenMax<-max(DetDxEffectArray[5:9,1:4,1:4,,,]),
PenMax<-max(DetDxEffectArray[10:13,1:4,1:4,,,]),
PEnMax<-max(DetDxEffectArray[10:13,10:13,1:4,,,]),
PXnMax<-max(DetDxEffectArray[10:13,5:9,1:4,,,]),
XeXMax<-max(DetDxEffectArray[5:9,1:4,5:9,,,]),
PeXMax<-max(DetDxEffectArray[10:13,1:4,5:9,,,]),
XXXMax<-max(DetDxEffectArray[5:9,5:9,5:9,,,]),
PXXMax<-max(DetDxEffectArray[10:13,5:9,5:9,,,]),
PEXMax<-max(DetDxEffectArray[10:13,10:13,5:9,,,]),
XeNMax<-max(DetDxEffectArray[5:9,1:4,10:13,,,]),
PeNMax<-max(DetDxEffectArray[10:13,1:4,10:13,,,]),
XXNMax<-max(DetDxEffectArray[5:9,5:9,10:13,,,]),
PXNMax<-max(DetDxEffectArray[10:13,5:9,10:13,,,]),
PENMax<-max(DetDxEffectArray[10:13,10:13,10:13,,,]),
XEnMax<-max(DetDxEffectArray[5:9,10:13,1:4,,,]),
XXnMax<-max(DetDxEffectArray[5:9,5:9,1:4,,,]),
XEXMax<-max(DetDxEffectArray[5:9,10:13,5:9,,,]),
XENMax<-max(DetDxEffectArray[5:9,10:13,10:13,,,]))

DetSimBlockMax # REPORTS MAXIMUM VALUES ACROSS PERSONALITY COMBINATION ARRAY SUBSETS

## THIS EVALUATES THE MINIMUM VALUES FOR ARRAY SUBSETS REPRESENTING PERSONALITY FACTOR COMBINATIONS
# ON THREE PERSONALITY SUB-INTERVALS
DetSimBlockMin<-c(
penMin<-min(DetDxEffectArray[1:4,1:4,1:4,,,]),
pXnMin<-min(DetDxEffectArray[1:4,5:9,1:4,,,]),
pEnMin<-min(DetDxEffectArray[1:4,10:13,1:4,,,]),
peXMin<-min(DetDxEffectArray[1:4,1:4,5:9,,,]),
```

```
pXXMin<-min(DetDxEffectArray[1:4,5:9,5:9,,,]),
pEXMin<-min(DetDxEffectArray[1:4,10:13,5:9,,,]),
peNMin<-min(DetDxEffectArray[1:4,1:4,10:13,,,]),
pXNMin<-min(DetDxEffectArray[1:4,5:9,10:13,,,]),
pENMin<-min(DetDxEffectArray[1:4,10:13,10:13,,,]),
XenMin<-min(DetDxEffectArray[5:9,1:4,1:4,,,]),
PenMin<-min(DetDxEffectArray[10:13,1:4,1:4,,,]),
PEnMin<-min(DetDxEffectArray[10:13,10:13,1:4,,,]),
PXnMin<-min(DetDxEffectArray[10:13,5:9,1:4,,,]),
XeXMin<-min(DetDxEffectArray[5:9,1:4,5:9,,,]),
PeXMin<-min(DetDxEffectArray[10:13,1:4,5:9,,,]),
XXXMin<-min(DetDxEffectArray[5:9,5:9,5:9,,,]),
PXXMin<-min(DetDxEffectArray[10:13,5:9,5:9,,,]),
PEXMin<-min(DetDxEffectArray[10:13,10:13,5:9,,,]),
XeNMin<-min(DetDxEffectArray[5:9,1:4,10:13,,,]),
PeNMin<-min(DetDxEffectArray[10:13,1:4,10:13,,,]),
XXNMin<-min(DetDxEffectArray[5:9,5:9,10:13,,,]),
PXNMin<-min(DetDxEffectArray[10:13,5:9,10:13,,,]),
PENMin<-min(DetDxEffectArray[10:13,10:13,10:13,,,]),
XEnMin<-min(DetDxEffectArray[5:9,10:13,1:4,,,]),
XXnMin<-min(DetDxEffectArray[5:9,5:9,1:4,,,]),
XEXMin<-min(DetDxEffectArray[5:9,10:13,5:9,,,]),
XENMin<-min(DetDxEffectArray[5:9,10:13,10:13,,,]))

DetSimBlockMin # REPORTS MINIMUM VALUES ACROSS PERSONALITY COMBINATION ARRAY SUBSETS

## THIS COMBINES MEAN, VARIANCE, MAXIMUM AND MINIMUMS IN ONE REPORT MATRIX
DetSimBlockStats<-cbind(PersBlockList,DetSimBlockMeans,
```

```
DetSimBlockVars, DetSimBlockMax,DetSimBlockMin)
DetSimBlockStats # REPORTS MEAN, VARIANCE, MAXIMUM AND MINIMUMS

#############################################################################################
#############################################################################################

### GENERATE RESPONSE SURFACE REPLICATES WITH RANDOMIZED PREDICTORS
####Initialize Report Vectors and Sim Parameters
set.seed(1234)
#set.seed(5678)  # USE ONLY FOR SECOND SIMULATION FOR EXTRACTION OF OPTIMIZED DATA.
q<-3     ## LEVELS PER FACTOR
g<-10000 ## NUMBER OF RESPONSE SURFACE REPLICATES

DxEffectArray<-array(data = NA, dim = c(q,q,q,q,q,q), dimnames = NULL)          # THE RESPONSE SURFACE FOR THE LAST REPORTED REPLICATE
StochFullSimArray<-array(data = NA, dim = c(g,q,q,q,q,q,q), dimnames = NULL)    # THE ARRAY OF RESPONSE SURFACES FOR ALL REPLICATIONS

DxEffectRunMax<-DxEffectRunMin<-DxEffectRunVar<-DxEffectRunMean<-NULL # INITIALIZE VECTOR OF MAX VALUES

# INITIALIZE VECTORS OF MAX VALUES BY PERSONALITY COMBINATION
penMaxVector<-      pXnMaxVector<-      pEnMaxVector<-      peXMaxVector<-
pXXMaxVector<-      pEXMaxVector<-      peNMaxVector<-      pXNMaxVector<-
pENMaxVector<-      XenMaxVector<-      PenMaxVector<-      PEnMaxVector<-
PXnMaxVector<-      XeXMaxVector<-      PeXMaxVector<-      XXXMaxVector<-
PXXMaxVector<-      PEXMaxVector<-      XeNMaxVector<-      PeNMaxVector<-
XXNMaxVector<-      PXNMaxVector<-      PENMaxVector<-      XEnMaxVector<-
XXnMaxVector<-      XEXMaxVector<-      XENMaxVector<-NULL
```

```
#INITIALIZE VECTORS OF MEAN VALUES BY PERSONALITY COMBINATION
penMeanVector<-      pXnMeanVector<-      pEnMeanVector<-      peXMeanVector<-
pXXMeanVector<-      pEXMeanVector<-      peNMeanVector<-      pXNMeanVector<-
pENMeanVector<-      XenMeanVector<-      PenMeanVector<-      PEnMeanVector<-
PXnMeanVector<-      XeXMeanVector<-      PeXMeanVector<-      XXXMeanVector<-      PXXMeanVector<-
PEXMeanVector<-      XeNMeanVector<-      PeNMeanVector<-      XXNMeanVector<-
PXNMeanVector<-      PENMeanVector<-XEnMeanVector<-     XXnMeanVector<-
XEXMeanVector<-      XENMeanVector<-NULL

#INITIALIZE VECTORS OF MODAL VALUES BY PERSONALITY COMBINATION
# THESE RECORD HOW OFTEN MAXIMUM VALUES ARE PRODUCED BY EACH COMBINATION
penModalVector<-     pXnModalVector<-     pEnModalVector<-     peXModalVector<-
pXXModalVector<-     pEXModalVector<-     peNModalVector<-     pXNModalVector<-
pENModalVector<-XenModalVector<-   PenModalVector<-     PEnModalVector<-     PXnModalVector<-
XeXModalVector<-     PeXModalVector<-     XXXModalVector<-     PXXModalVector<-
PEXModalVector<-     XeNModalVector<-     PeNModalVector<-     XXNModalVector<-
PXNModalVector<-     PENModalVector<-
XEnModalVector<-     XXnModalVector<-     XEXModalVector<-     XENModalVector<-NULL

# INITIALIZE MATRIX TO REPORT STATISITICS BY PERSONLITY COMBINATION
penMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
pXnMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
pEnMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
peXMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
pXXMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
pEXMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
peNMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
pXNMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
```

```
pENMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
XenMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
PenMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
PEnMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
PXnMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
XeXMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
PeXMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
XXXMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
PXXMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
PEXMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
XeNMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
PeNMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
XXNMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
PXNMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
PENMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
XEnMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
XXnMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
XEXMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)
XENMaxMatrixStoch<-matrix(, nrow = 1, ncol = 6, byrow = FALSE)

## INITIATE SIMULATION
# INITIALIZE FIXED FACTORS IN THREE SUB-INTERVALS
        Pval<-seq(mean(P)-sd(P),mean(P)+sd(P),len=q)
        Eval<-seq(mean(E)-sd(E),mean(E)+sd(E),len=q)
        Nval<-seq(mean(N)-sd(N),mean(N)+sd(N),len=q)
        Cval<-seq(-1,1,len=q)
        Sval<-seq(-1,1,len=q)
        Gval<-seq(-1,1,len=q)
```

```
# THIS LOOP INDUCES VARIANCE FOR ALL FACTORS WITHIN THREE SUBINTERVALS AND GENERATES 10000 REPLICATES OF THE
#RESPONSE SURFACE
for (h in 1:g){ #0
        for (i in 1:q){ #1
                for (j in 1:q){  #2
                        for (k in 1:q){ #3
                          for (l in 1:q){ #4
                                  for (m in 1:q){ #5
                                          for (n in 1:q){ #6
                                                  Prep<-Pval+runif(1,-0.5*sd(P),0.5*sd(P))
                                                  Erep<-Eval+runif(1,-0.5*sd(E),0.5*sd(E))
                                                  Nrep<-Nval+runif(1,-0.5*sd(N),0.5*sd(N))
                                                  Crep<-Cval+runif(1,-0.5,0.5)
                                                  Srep<-Sval+runif(1,-0.5,0.5)
                                                  Grep<-Gval+runif(1,-0.5,0.5)

                DxEffectArray[i,j,k,l,m,n]<-(Study2MMreduxrpt2x[1] # THIS SERIES OF COMMANDS APPLIES PREDICTION PARMAETERS
                +Study2MMreduxrpt2x[2]*Prep[i] ## P
                +Study2MMreduxrpt2x[3]*Erep[j]## E
                +Study2MMreduxrpt2x[4]*Nrep[k]## N
                +Study2MMreduxrpt2x[5]*Crep[l] ## C
                +Study2MMreduxrpt2x[6]*Srep[m]## S
                +Study2MMreduxrpt2x[7]*Grep[n]## G
                +Study2MMreduxrpt2x[8]*Prep[i]*Crep[l]  #P*C
                +Study2MMreduxrpt2x[9]*Prep[i]*Srep[m] #PS
                +Study2MMreduxrpt2x[10]*Prep[i]*Grep[n]#PG
                +Study2MMreduxrpt2x[11]*Erep[j]*Crep[l]#EC
```

```
                +Study2MMreduxrpt2x[12]*Erep[j]*Grep[n]#EG
                +Study2MMreduxrpt2x[13]*Nrep[k]*Crep[l]#NC
                +Study2MMreduxrpt2x[14]*Nrep[k]*Grep[n] ## NG
                +Study2MMreduxrpt2x[15]*Crep[l]*Grep[n] ## CG
                +Study2MMreduxrpt2x[16]*Prep[i]*Crep[l]*Grep[n] ## PCG
                +Study2MMreduxrpt2x[17]*Erep[j]*Crep[l]*Grep[n] ## ECG
                +Study2MMreduxrpt2x[18]*Nrep[k]*Crep[l]*Grep[n]) ## NCG
        StochFullSimArray[h,i,j,k,l,m,n]<-DxEffectArray[i,j,k,l,m,n] # THIS POPULATES THE MAIN RESPONSE SURFACE ARRAY
} #6
} #5
} #4
} #3
} #2
} #1#End Replication

# THE FOLLOWING COMMANDS INCREMENTALLY POPULATE REPORT VECTORS FOR ALL STATISTICS OF INTEREST

DxEffectRunMax<-c(DxEffectRunMax,max(DxEffectArray))
DxEffectRunMin<-c(DxEffectRunMin,min(DxEffectArray))
DxEffectRunMean<-c(DxEffectRunMean,mean(DxEffectArray))
DxEffectRunVar<-c(DxEffectRunVar,var(DxEffectArray))

penMaxVector<-c(penMaxVector,max(DxEffectArray[1,1,1,,,]))
pXnMaxVector<-c(pXnMaxVector,max(DxEffectArray[1,2,1,,,]))#
pEnMaxVector<-c(pEnMaxVector,max(DxEffectArray[1,3,1,,,]))#
peXMaxVector<-c(peXMaxVector,max(DxEffectArray[1,1,2,,,]))
pXXMaxVector<-c(pXXMaxVector,max(DxEffectArray[1,2,2,,,]))
pEXMaxVector<-c(pEXMaxVector,max(DxEffectArray[1,3,2,,,]))
```

```
peNMaxVector<-c(peNMaxVector,max(DxEffectArray[1,1,3,,,]))
pXNMaxVector<-c(pXNMaxVector,max(DxEffectArray[1,2,3,,,]))
pENMaxVector<-c(pENMaxVector,max(DxEffectArray[1,3,3,,,]))
XenMaxVector<-c(XenMaxVector,max(DxEffectArray[2,1,1,,,]))
PenMaxVector<-c(PenMaxVector,max(DxEffectArray[3,1,1,,,]))
PEnMaxVector<-c(PEnMaxVector,max(DxEffectArray[3,3,1,,,]))
PXnMaxVector<-c(PXnMaxVector,max(DxEffectArray[3,2,1,,,]))
XeXMaxVector<-c(XeXMaxVector,max(DxEffectArray[2,1,2,,,]))
PeXMaxVector<-c(PeXMaxVector,max(DxEffectArray[3,1,2,,,]))
XXXMaxVector<-c(XXXMaxVector,max(DxEffectArray[2,2,2,,,]))
PXXMaxVector<-c(PXXMaxVector,max(DxEffectArray[3,2,2,,,]))
PEXMaxVector<-c(PEXMaxVector,max(DxEffectArray[3,3,2,,,]))
XeNMaxVector<-c(XeNMaxVector,max(DxEffectArray[2,1,3,,,]))
PeNMaxVector<-c(PeNMaxVector,max(DxEffectArray[3,1,3,,,]))
XXNMaxVector<-c(XXNMaxVector,max(DxEffectArray[2,2,3,,,]))
PXNMaxVector<-c(PXNMaxVector,max(DxEffectArray[3,2,3,,,]))
PENMaxVector<-c(PENMaxVector,max(DxEffectArray[3,3,3,,,]))
XEnMaxVector<-c(XEnMaxVector,max(DxEffectArray[2,3,1,,,]))
XXnMaxVector<-c(XXnMaxVector,max(DxEffectArray[2,2,1,,,]))
XEXMaxVector<-c(XEXMaxVector,max(DxEffectArray[2,3,2,,,]))
XENMaxVector<-c(XENMaxVector,max(DxEffectArray[2,3,3,,,]))

penMeanVector<-c(penMeanVector,mean(DxEffectArray[1,1,1,,,]))
pXnMeanVector<-c(pXnMeanVector,mean(DxEffectArray[1,2,1,,,]))
pEnMeanVector<-c(pEnMeanVector,mean(DxEffectArray[1,3,1,,,]))#
peXMeanVector<-c(peXMeanVector,mean(DxEffectArray[1,1,2,,,]))
pXXMeanVector<-c(pXXMeanVector,mean(DxEffectArray[1,2,2,,,]))
pEXMeanVector<-c(pEXMeanVector,mean(DxEffectArray[1,3,2,,,]))
```

```
peNMeanVector<-c(peNMeanVector,mean(DxEffectArray[1,1,3,,,]))
pXNMeanVector<-c(pXNMeanVector,mean(DxEffectArray[1,2,3,,,]))
pENMeanVector<-c(pENMeanVector,mean(DxEffectArray[1,3,3,,,]))
XenMeanVector<-c(XenMeanVector,mean(DxEffectArray[2,1,1,,,]))
PenMeanVector<-c(PenMeanVector,mean(DxEffectArray[3,1,1,,,]))
PEnMeanVector<-c(PEnMeanVector,mean(DxEffectArray[3,3,1,,,]))
PXnMeanVector<-c(PXnMeanVector,mean(DxEffectArray[3,2,1,,,]))
XeXMeanVector<-c(XeXMeanVector,mean(DxEffectArray[2,1,2,,,]))
PeXMeanVector<-c(PeXMeanVector,mean(DxEffectArray[3,1,2,,,]))
XXXMeanVector<-c(XXXMeanVector,mean(DxEffectArray[2,2,2,,,]))
PXXMeanVector<-c(PXXMeanVector,mean(DxEffectArray[3,2,2,,,]))
PEXMeanVector<-c(PEXMeanVector,mean(DxEffectArray[3,3,2,,,]))
XeNMeanVector<-c(XeNMeanVector,mean(DxEffectArray[2,1,3,,,]))
PeNMeanVector<-c(PeNMeanVector,mean(DxEffectArray[3,1,3,,,]))
XXNMeanVector<-c(XXNMeanVector,mean(DxEffectArray[2,2,3,,,]))
PXNMeanVector<-c(PXNMeanVector,mean(DxEffectArray[3,2,3,,,]))
PENMeanVector<-c(PENMeanVector,mean(DxEffectArray[3,3,3,,,]))
XEnMeanVector<-c(XEnMeanVector,mean(DxEffectArray[2,3,1,,,]))
XXnMeanVector<-c(XXnMeanVector,mean(DxEffectArray[2,2,1,,,]))
XEXMeanVector<-c(XEXMeanVector,mean(DxEffectArray[2,3,2,,,]))
XENMeanVector<-c(XENMeanVector,mean(DxEffectArray[2,3,3,,,]))

penModalVector<-c(penModalVector,which(DxEffectArray==max(DxEffectArray[1,1,1,,,])))
pXnModalVector<-c(pXnModalVector,which(DxEffectArray==max(DxEffectArray[1,2,1,,,])))#
pEnModalVector<-c(pEnModalVector,which(DxEffectArray==max(DxEffectArray[1,3,1,,,])))#
peXModalVector<-c(peXModalVector,which(DxEffectArray==max(DxEffectArray[1,1,2,,,])))
pXXModalVector<-c(pXXModalVector,which(DxEffectArray==max(DxEffectArray[1,2,2,,,])))
pEXModalVector<-c(pEXModalVector,which(DxEffectArray==max(DxEffectArray[1,3,2,,,])))
```

```
peNModalVector<-c(peNModalVector,which(DxEffectArray==max(DxEffectArray[1,1,3,,,])))
pXNModalVector<-c(pXNModalVector,which(DxEffectArray==max(DxEffectArray[1,2,3,,,])))
pENModalVector<-c(pENModalVector,which(DxEffectArray==max(DxEffectArray[1,3,3,,,])))
XenModalVector<-c(XenModalVector,which(DxEffectArray==max(DxEffectArray[2,1,1,,,])))
PenModalVector<-c(PenModalVector,which(DxEffectArray==max(DxEffectArray[3,1,1,,,])))
PEnModalVector<-c(PEnModalVector,which(DxEffectArray==max(DxEffectArray[3,3,1,,,])))
PXnModalVector<-c(PXnModalVector,which(DxEffectArray==max(DxEffectArray[3,2,1,,,])))
XeXModalVector<-c(XeXModalVector,which(DxEffectArray==max(DxEffectArray[2,1,2,,,])))
PeXModalVector<-c(PeXModalVector,which(DxEffectArray==max(DxEffectArray[3,1,2,,,])))
XXXModalVector<-c(XXXModalVector,which(DxEffectArray==max(DxEffectArray[2,2,2,,,])))
PXXModalVector<-c(PXXModalVector,which(DxEffectArray==max(DxEffectArray[3,2,2,,,])))
PEXModalVector<-c(PEXModalVector,which(DxEffectArray==max(DxEffectArray[3,3,2,,,])))
XeNModalVector<-c(XeNModalVector,which(DxEffectArray==max(DxEffectArray[2,1,3,,,])))
PeNModalVector<-c(PeNModalVector,which(DxEffectArray==max(DxEffectArray[3,1,3,,,])))
XXNModalVector<-c(XXNModalVector,which(DxEffectArray==max(DxEffectArray[2,2,3,,,])))
PXNModalVector<-c(PXNModalVector,which(DxEffectArray==max(DxEffectArray[3,2,3,,,])))
PENModalVector<-c(PENModalVector,which(DxEffectArray==max(DxEffectArray[3,3,3,,,])))
XEnModalVector<-c(XEnModalVector,which(DxEffectArray==max(DxEffectArray[2,3,1,,,])))
XXnModalVector<-c(XXnModalVector,which(DxEffectArray==max(DxEffectArray[2,2,1,,,])))
XEXModalVector<-c(XEXModalVector,which(DxEffectArray==max(DxEffectArray[2,3,2,,,])))
XENModalVector<-c(XENModalVector,which(DxEffectArray==max(DxEffectArray[2,3,3,,,])))

penMaxMatrixStoch<-rbind(penMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[1,1,1,,,]),arr.ind = T))
pXnMaxMatrixStoch<-rbind(pXnMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[1,2,1,,,]),arr.ind = T))#
pEnMaxMatrixStoch<-rbind(pEnMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[1,3,1,,,]),arr.ind = T))#
peXMaxMatrixStoch<-rbind(peXMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[1,1,2,,,]),arr.ind = T))
pXXMaxMatrixStoch<-rbind(pXXMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[1,2,2,,,]),arr.ind = T))
pEXMaxMatrixStoch<-rbind(pEXMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[1,3,2,,,]),arr.ind = T))
```

```
peNMaxMatrixStoch<-rbind(peNMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[1,1,3,,,]),arr.ind = T))
pXNMaxMatrixStoch<-rbind(pXNMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[1,2,3,,,]),arr.ind = T))
pENMaxMatrixStoch<-rbind(pENMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[1,3,3,,,]),arr.ind = T))
XenMaxMatrixStoch<-rbind(XenMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[2,1,1,,,]),arr.ind = T))
PenMaxMatrixStoch<-rbind(PenMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[3,1,1,,,]),arr.ind = T))
PEnMaxMatrixStoch<-rbind(PEnMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[3,3,1,,,]),arr.ind = T))
PXnMaxMatrixStoch<-rbind(PXnMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[3,2,1,,,]),arr.ind = T))
XeXMaxMatrixStoch<-rbind(XeXMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[2,1,2,,,]),arr.ind = T))
PeXMaxMatrixStoch<-rbind(PeXMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[3,1,2,,,]),arr.ind = T))
XXXMaxMatrixStoch<-rbind(XXXMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[2,2,2,,,]),arr.ind = T))
PXXMaxMatrixStoch<-rbind(PXXMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[3,2,2,,,]),arr.ind = T))
PEXMaxMatrixStoch<-rbind(PEXMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[3,3,2,,,]),arr.ind = T))
XeNMaxMatrixStoch<-rbind(XeNMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[2,1,3,,,]),arr.ind = T))
PeNMaxMatrixStoch<-rbind(PeNMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[3,1,3,,,]),arr.ind = T))
XXNMaxMatrixStoch<-rbind(XXNMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[2,2,3,,,]),arr.ind = T))
PXNMaxMatrixStoch<-rbind(PXNMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[3,2,3,,,]),arr.ind = T))
PENMaxMatrixStoch<-rbind(PENMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[3,3,3,,,]),arr.ind = T))
XEnMaxMatrixStoch<-rbind(XEnMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[2,3,1,,,]),arr.ind = T))
XXnMaxMatrixStoch<-rbind(XXnMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[2,2,1,,,]),arr.ind = T))
XEXMaxMatrixStoch<-rbind(XEXMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[2,3,2,,,]),arr.ind = T))
XENMaxMatrixStoch<-rbind(XENMaxMatrixStoch,which(DxEffectArray==max(DxEffectArray[2,3,3,,,]),arr.ind = T))

} #0 #End Simulation

## EVALUATE SIMULATION DATA AND GENERATE REPORTS
# GENERATE SUMMARY STATISITICS ACROSS ALL REPLICATIONS
mean(StochFullSimArray)
var(StochFullSimArray)
```

```
max(StochFullSimArray)
min(StochFullSimArray)
mean(DxEffectRunMean) # MEAN OF MEAN VALUES
var(DxEffectRunMean) # VARIANCE OF MEAN VALUES #[1] 0.01912326

## EVALUATE SUMMARY STATISITICS FOR EACH REPLICATION

DxEffectRunMax # VECTOR OF MAXIMUM VALUES BY REPLICATION
mean(DxEffectRunMax
max(DxEffectRunMax)
min(DxEffectRunMax)
var(DxEffectRunMax)

# EVALUATE STATISITICS FOR MINIMUM VALUES IN EACH REPLICATION
DxEffectRunMin # VECTOR OF MINIMUM VALUES BY REPLICATION
mean(DxEffectRunMin)
max(DxEffectRunMin)
min(DxEffectRunMin)
var(DxEffectRunMin)

DxEffectRunMean# VECTOR OF MEAN VALUES BY REPLICATION
mean(DxEffectRunMean)## 0.7685012
max(DxEffectRunMean)##1.360714
min(DxEffectRunMean)##0.247747
var(DxEffectRunMean)##0.01324133

DxEffectRunVar # VECTOR VARIANCE BY REPLICATION
mean(DxEffectRunVar)##0.9037814
```

```
max(DxEffectRunVar)##1.904446
min(DxEffectRunVar)## 0.6594548
var(DxEffectRunVar)## 0.05061463

## ESTABLISH MATRIX OF MODAL VECTORS

StochModalVector<-rbind(
penModalVector,     pXnModalVector,     pEnModalVector,     peXModalVector,
pXXModalVector,     pEXModalVector,     peNModalVector,     pXNModalVector,
pENModalVector,     XenModalVector,     PenModalVector,     PEnModalVector,
PXnModalVector,     XeXModalVector,     PeXModalVector,     XXXModalVector,
PXXModalVector,     PEXModalVector,     XeNModalVector,     PeNModalVector,
XXNModalVector,     PXNModalVector,     PENModalVector,     XEnModalVector,
XXnModalVector,     XEXModalVector,     XENModalVector)

## ESTABLISH VECTOR WITH MAX VALUES FOR EACH PERSONALITY COMBINATION ACROSS REPLICATIONS

BlockMax<-
c(max(penMaxVector),max(pXnMaxVector),max(pEnMaxVector),max(peXMaxVector),max(pXXMaxVector),max(pEXMaxVector),max(peNMaxVector),max(pXN
MaxVector),max(pENMaxVector),max(XenMaxVector),max(PenMaxVector),max(PEnMaxVector),max(PXnMaxVector),max(XeXMaxVector),max(PeXMaxVector),
max(XXXMaxVector),max(PXXMaxVector),max(PEXMaxVector),max(XeNMaxVector),max(PeNMaxVector),max(XXNMaxVector),max(PXNMaxVector),max(PENM
axVector),max(XEnMaxVector),max(XXnMaxVector),max(XEXMaxVector),max(XENMaxVector))

# EVALUATE SUMMARY STATSITICS FOR MAXIMUM VALUES BY PERSONALITY COMBINATION

mean(BlockMax)

var(BlockMax)

# ESTABLISH VECTOR OF MEAN VALUES FOR EACH PERSONALITY COMBINATION ACROSS REPLICATIONS

StochFullMeanVector<-c(
```

```
mean(StochFullSimArray[,1,1,1,,,]), #pen/LHL
mean(StochFullSimArray[,1,2,1,,,]),
mean(StochFullSimArray[,1,3,1,,,]),
mean(StochFullSimArray[,1,1,2,,,]),
mean(StochFullSimArray[,1,2,2,,,]),
mean(StochFullSimArray[,1,3,2,,,]),
mean(StochFullSimArray[,1,1,3,,,]),
mean(StochFullSimArray[,1,2,3,,,]),
mean(StochFullSimArray[,1,3,3,,,]),
mean(StochFullSimArray[,2,1,1,,,]),
mean(StochFullSimArray[,3,1,1,,,]),
mean(StochFullSimArray[,3,3,1,,,]),
mean(StochFullSimArray[,3,2,1,,,]),
mean(StochFullSimArray[,2,1,2,,,]),
mean(StochFullSimArray[,3,1,2,,,]),
mean(StochFullSimArray[,2,2,2,,,]),
mean(StochFullSimArray[,3,2,2,,,]),
mean(StochFullSimArray[,3,3,2,,,]),
mean(StochFullSimArray[,2,1,3,,,]),
mean(StochFullSimArray[,3,1,3,,,]),
mean(StochFullSimArray[,2,2,3,,,]),
mean(StochFullSimArray[,3,2,3,,,]),
mean(StochFullSimArray[,3,3,3,,,]),
mean(StochFullSimArray[,2,3,1,,,]),
mean(StochFullSimArray[,2,2,1,,,]),
mean(StochFullSimArray[,2,3,2,,,]),
mean(StochFullSimArray[,2,3,3,,,]))
```

```
mean(StochFullMeanVector) # REPORTS THE MEAN OF MEANS ACROSS REPLICATIONS

# ESTABLISH VECTOR OF VARIANCE FOR EACH PERSONALITY COMBINATION ACROSS REPLICATIONS

StochFullVarVector<-c(
var(StochFullSimArray[,1,1,1,,,]),
var(StochFullSimArray[,1,2,1,,,]),
var(StochFullSimArray[,1,3,1,,,]),
var(StochFullSimArray[,1,1,2,,,]),
var(StochFullSimArray[,1,2,2,,,]),
var(StochFullSimArray[,1,3,2,,,]),
var(StochFullSimArray[,1,1,3,,,]),
var(StochFullSimArray[,1,2,3,,,]),
var(StochFullSimArray[,1,3,3,,,]),
var(StochFullSimArray[,2,1,1,,,]),
var(StochFullSimArray[,3,1,1,,,]),
var(StochFullSimArray[,3,3,1,,,]),
var(StochFullSimArray[,3,2,1,,,]),
var(StochFullSimArray[,2,1,2,,,]),
var(StochFullSimArray[,3,1,2,,,]),
var(StochFullSimArray[,2,2,2,,,]),
var(StochFullSimArray[,3,2,2,,,]),
var(StochFullSimArray[,3,3,2,,,]),
var(StochFullSimArray[,2,1,3,,,]),
var(StochFullSimArray[,3,1,3,,,]),
var(StochFullSimArray[,2,2,3,,,]),
var(StochFullSimArray[,3,2,3,,,]),
var(StochFullSimArray[,3,3,3,,,]),
```

```
var(StochFullSimArray[,2,3,1,,,]),
var(StochFullSimArray[,2,2,1,,,]),
var(StochFullSimArray[,2,3,2,,,]),
var(StochFullSimArray[,2,3,3,,,]))

StochFullVarVector # REPORTS THE VARIANCE ACROSS REPLICATIONS BY PERSONALITY COMBINATION

# ESTABLISH VECTOR OF MAXIMUM VALUES FOR EACH PERSONALITY COMBINATION ACROSS REPLICATIONS

StochFullMaxVector<-c(
max(StochFullSimArray[,1,1,1,,,]),
max(StochFullSimArray[,1,2,1,,,]),
max(StochFullSimArray[,1,3,1,,,]),
max(StochFullSimArray[,1,1,2,,,]),
max(StochFullSimArray[,1,2,2,,,]),
max(StochFullSimArray[,1,3,2,,,]),
max(StochFullSimArray[,1,1,3,,,]),
max(StochFullSimArray[,1,2,3,,,]),
max(StochFullSimArray[,1,3,3,,,]),
max(StochFullSimArray[,2,1,1,,,]),
max(StochFullSimArray[,3,1,1,,,]),
max(StochFullSimArray[,3,3,1,,,]),
max(StochFullSimArray[,3,2,1,,,]),
max(StochFullSimArray[,2,1,2,,,]),
max(StochFullSimArray[,3,1,2,,,]),
max(StochFullSimArray[,2,2,2,,,]),
max(StochFullSimArray[,3,2,2,,,]),
max(StochFullSimArray[,3,3,2,,,]),
```

```
max(StochFullSimArray[,2,1,3,,,]),
max(StochFullSimArray[,3,1,3,,,]),
max(StochFullSimArray[,2,2,3,,,]),
max(StochFullSimArray[,3,2,3,,,]),
max(StochFullSimArray[,3,3,3,,,]),
max(StochFullSimArray[,2,3,1,,,]),
max(StochFullSimArray[,2,2,1,,,]),
max(StochFullSimArray[,2,3,2,,,]),
max(StochFullSimArray[,2,3,3,,,]))

#StochFullMaxVector

# ESTABLISH VECTOR OF MINIMUM VALUES FOR EACH PERSONALITY COMBINATION ACROSS REPLICATIONS

StochFullMinVector<-c(
min(StochFullSimArray[,1,1,1,,,]),
min(StochFullSimArray[,1,2,1,,,]),
min(StochFullSimArray[,1,3,1,,,]),
min(StochFullSimArray[,1,1,2,,,]),
min(StochFullSimArray[,1,2,2,,,]),
min(StochFullSimArray[,1,3,2,,,]),
min(StochFullSimArray[,1,1,3,,,]),
min(StochFullSimArray[,1,2,3,,,]),
min(StochFullSimArray[,1,3,3,,,]),
min(StochFullSimArray[,2,1,1,,,]),
min(StochFullSimArray[,3,1,1,,,]),
min(StochFullSimArray[,3,3,1,,,]),
min(StochFullSimArray[,3,2,1,,,]),
```

```
min(StochFullSimArray[,2,1,2,,,]),
min(StochFullSimArray[,3,1,2,,,]),
min(StochFullSimArray[,2,2,2,,,]),
min(StochFullSimArray[,3,2,2,,,]),
min(StochFullSimArray[,3,3,2,,,]),
min(StochFullSimArray[,2,1,3,,,]),
min(StochFullSimArray[,3,1,3,,,]),
min(StochFullSimArray[,2,2,3,,,]),
min(StochFullSimArray[,3,2,3,,,]),
min(StochFullSimArray[,3,3,3,,,]),
min(StochFullSimArray[,2,3,1,,,]),
min(StochFullSimArray[,2,2,1,,,]),
min(StochFullSimArray[,2,3,2,,,]),
min(StochFullSimArray[,2,3,3,,,]))

#StochFullMinVector

## CONCATENATE VECTORS AS A REPORT MATRIX
StochFullBlockStats<-cbind(PersBlockList,StochFullMeanVector, StochFullVarVector,StochFullMaxVector,StochFullMinVector)

StochFullBlockStats # REPORTS SUMMARY STATS ACROSS REPLICATIONS

########################## EXTRACT DATA BY BLOCK WITH ALL SITUATIONAL COMBINATIONS

mean(StochFullSimArray[,2,2,1,1,1,1]) #XXn/HLL
mean(StochFullSimArray[,1,1,3,3,3,1]) #peN/HHL
```

```
ObjVarVector<-c(
var(StochFullSimArray[,1,1,1,1,3,1]), #pen/LHL
var(StochFullSimArray[,1,2,1,1,3,1]), #pXn/LHL
var(StochFullSimArray[,1,3,1,1,3,1]), #pEn/LHL
var(StochFullSimArray[,1,1,2,1,3,1]), #peX/LHL
var(StochFullSimArray[,1,2,2,1,3,1]), #pXX/LHL
var(StochFullSimArray[,1,3,2,1,3,1]), #pEX/LHL
var(StochFullSimArray[,1,1,3,1,3,1]), #peN/LHL
var(StochFullSimArray[,1,2,3,1,3,1]), #pXN/LHL
var(StochFullSimArray[,1,3,3,1,3,1]), #pEN/LHL
var(StochFullSimArray[,2,1,1,3,1,1]), #Xen/HLL
var(StochFullSimArray[,3,1,1,3,1,1]), #Pen/HLL
var(StochFullSimArray[,3,3,1,3,1,1]), #PEn/HLL
var(StochFullSimArray[,3,2,1,3,1,1]), #PXn/HLL
var(StochFullSimArray[,2,1,2,3,1,1]), #XeX/HLL
var(StochFullSimArray[,3,1,2,3,1,1]), #PeX/HLL
var(StochFullSimArray[,2,2,2,3,1,1]), #XXX/HLL
var(StochFullSimArray[,3,2,2,3,1,1]), #PXX/HLL
var(StochFullSimArray[,3,3,2,3,1,1]), #PEX/HLL
var(StochFullSimArray[,2,1,3,3,1,1]), #XeN/HLL
var(StochFullSimArray[,3,1,3,3,1,1]), #PeN/HLL
var(StochFullSimArray[,2,2,3,3,1,1]), #XXN/HLL
var(StochFullSimArray[,3,2,3,3,1,1]), #PXN/HLL
var(StochFullSimArray[,3,3,3,3,1,1]), #PEN/HLL
var(StochFullSimArray[,2,3,1,1,1,1]), #XEn/LLL
var(StochFullSimArray[,2,2,1,3,1,1]), #XXn/HLL
var(StochFullSimArray[,2,3,2,1,1,1]), #XEX/LLL
var(StochFullSimArray[,2,3,3,1,1,1])) #XEN/LLL
```

```
var(ObjVarVector) #[1] 0.1678017

#########################################################################
# THIS EXTRACTS AND EVALUATES AN OPTIMIZED SUBSET OF THE STOCHASTIC RESPONSE SURFACE ARRAY
# ACROSS REPLICATIONS TO SUPPORT COMPARISON TO THE FULL ARRAY
# NOTE THAT THE COMPARISON OF FULL DATA TO EXTRACTED DATA IS INVALID WITHOUT RE-RUNNING THE SIMULATION WITH A NEW RANDOM SEED
#################################################
# THIS EXTRACTS FROM THE STOCHASTIC ARRAY ALL RESPONSE DATA FROM
# THE ESTIMATED OPTIMAL SITUATIONAL COMBINATIONS BY PERSONALITY COMBINATION
OptBlockResults<-rbind(
penOptArray<-FullSimArray[,1,1,1,3,1,1],
pXnOptArray<-FullSimArray[,1,2,1,1,3,1],
pEnOptArray<-FullSimArray[,1,3,1,1,3,1],
peXOptArray<-FullSimArray[,1,1,2,1,3,1],
pXXOptArray<-FullSimArray[,1,2,2,1,3,1],
pEXOptArray<-FullSimArray[,1,3,2,1,3,1],
peNOptArray<-FullSimArray[,1,1,3,3,3,1],
pXNOptArray<-FullSimArray[,1,2,3,1,3,1],
pENOptArray<-FullSimArray[,1,3,3,1,3,1],
XenOptArray<-FullSimArray[,2,1,1,3,1,1],
PenOptArray<-FullSimArray[,3,1,1,3,1,1],
PEnOptArray<-FullSimArray[,3,3,1,3,1,1],
PXnOptArray<-FullSimArray[,3,2,1,3,1,1],
XeXOptArray<-FullSimArray[,2,1,2,3,1,1],
PeXOptArray<-FullSimArray[,3,1,2,3,1,1],
XXXOptArray<-FullSimArray[,2,2,2,3,1,1],
PXXOptArray<-FullSimArray[,3,2,2,3,1,1],
PEXOptArray<-FullSimArray[,3,3,2,3,1,1],
```

```
XeNOptArray<-FullSimArray[,2,1,3,3,1,1],
PeNOptArray<-FullSimArray[,3,1,3,3,1,1],
XXNOptArray<-FullSimArray[,2,2,3,3,1,1],
PXNOptArray<-FullSimArray[,3,2,3,3,1,1],
PENOptArray<-FullSimArray[,3,3,3,3,1,1],
XEnOptArray<-FullSimArray[,2,3,1,1,1,1],
XXnOptArray<-FullSimArray[,2,2,1,3,1,1],
XEXOptArray<-FullSimArray[,2,3,2,1,1,1],
XENOptArray<-FullSimArray[,2,3,3,1,1,1]
)

# THIS EVALUATES SUMMARY STATISITICS OF THE OPTIMIZED DATA ACROSS REPLICATIONS AND PERSONALITY COMBINATIONS
mean(OptBlockResults)
max(OptBlockResults)
min(OptBlockResults)
sd(OptBlockResults)^2 # Var function doesn't work on OptBlockResults so it is evaluated as sd^2

## THIS PERFORMS A T-TEST COMPARISON OF THE FULL (UNOPTIMIZED) STOCHASTIC ARRAY TO THE OPTIMIZED SUBSET
t.test(FullSimArray,OptBlockResults)

## THIS ESTABLISHES A REPORT OF SUMMARY STATISTICS OPTIMIZED BLOCK MEANS

# THIS CREATES A VECTOR OF MEAN VALUES FOR PERSONALITY COMBINATIONS IN THE OPTIMIZED SUBSET
OptBlockMeans<-rbind(
mean(penOptArray),mean(pXnOptArray),mean(pEnOptArray),
mean(peXOptArray),mean(pXXOptArray),mean(pEXOptArray),
mean(peNOptArray),mean(pXNOptArray),mean(pENOptArray),
mean(XenOptArray),mean(PenOptArray),mean(PEnOptArray),
```

```
mean(PXnOptArray),mean(XeXOptArray),mean(PeXOptArray),
mean(XXXOptArray),mean(PXXOptArray),mean(PEXOptArray),
mean(XeNOptArray),mean(PeNOptArray),mean(XXNOptArray),
mean(PXNOptArray),mean(PENOptArray),mean(XEnOptArray),
mean(XXnOptArray),mean(XEXOptArray),mean(XENOptArray))

# THIS CREATES A VECTOR VARIANCES FOR PERSONALITY COMBINATIONS IN THE OPTIMIZED SUBSET
OptBlockVars<-rbind(
var(penOptArray),var(pXnOptArray),var(pEnOptArray),
var(peXOptArray),var(pXXOptArray),var(pEXOptArray),
var(peNOptArray),var(pXNOptArray),var(pENOptArray),
var(XenOptArray),var(PenOptArray),var(PEnOptArray),
var(PXnOptArray),var(XeXOptArray),var(PeXOptArray),
var(XXXOptArray),var(PXXOptArray),var(PEXOptArray),
var(XeNOptArray),var(PeNOptArray),var(XXNOptArray),
var(PXNOptArray),var(PENOptArray),var(XEnOptArray),
var(XXnOptArray),var(XEXOptArray),var(XENOptArray))

# THIS CREATES A VECTOR OF MAXIMUM VALUES FOR PERSONALITY COMBINATIONS IN THE OPTIMIZED SUBSET
OptBlockMax<-rbind(
max(penOptArray),max(pXnOptArray),max(pEnOptArray),
max(peXOptArray),max(pXXOptArray),max(pEXOptArray),
max(peNOptArray),max(pXNOptArray),max(pENOptArray),
max(XenOptArray),max(PenOptArray),max(PEnOptArray),
max(PXnOptArray),max(XeXOptArray),max(PeXOptArray),
max(XXXOptArray),max(PXXOptArray),max(PEXOptArray),
max(XeNOptArray),max(PeNOptArray),max(XXNOptArray),
max(PXNOptArray),max(PENOptArray),max(XEnOptArray),
```

```
max(XXnOptArray),max(XEXOptArray),max(XENOptArray))


# THIS CREATES A VECTOR OF MINIMUM VALUES FOR PERSONALITY COMBINATIONS IN THE OPTIMIZED SUBSET
OptBlockMin<-rbind(
min(penOptArray),min(pXnOptArray),min(pEnOptArray),
min(peXOptArray),min(pXXOptArray),min(pEXOptArray),
min(peNOptArray),min(pXNOptArray),min(pENOptArray),
min(XenOptArray),min(PenOptArray),min(PEnOptArray),
min(PXnOptArray),min(XeXOptArray),min(PeXOptArray),
min(XXXOptArray),min(PXXOptArray),min(PEXOptArray),
min(XeNOptArray),min(PeNOptArray),min(XXNOptArray),
min(PXNOptArray),min(PENOptArray),min(XEnOptArray),
min(XXnOptArray),min(XEXOptArray),min(XENOptArray))

# THIS COMBINES MEAN, VARIANCE, MAXIMUM AND MINIMUM VECTOR FOR PERSONALITY COMBINATIONS IN THE OPTIMIZED SUBSET
OptimizedBlockStats<-cbind(PersBlockList,OptBlockMeans,OptBlockVars,OptBlockMax,OptBlockMin)

OptimizedBlockStats # THIS REPORTS THE SUMMARY STAITICS FOR PERSONALITY COMBINATIONS IN THE OPTIMIZED SUBSET
```