**Dataset Key:**

1. RE\_DATA1\_OSF
   1. Social science dataset comprised of 500 true, clean responses
   2. Used to determine initial descriptive statistics and factor structure of indices of careless responding
2. RE\_DATA2\_OSF
   1. Dataset comprised of the 500 original true, clean responses, plus 100 **human BS** responses to the survey
   2. Used for structural regression analysis testing the three factor model on detection of human BS
3. RE\_DATA3\_OSF
   1. Dataset comprised of the 500 original true, clean responses, plus 100 **computer-normalized** responses to the survey
   2. Used for structural regression analysis testing the three factor model on detection of computer normal BS
4. RE\_DATA4\_OSF
   1. Dataset comprised of the 500 original true, clean responses, plus 100 **computer-uniform** responses to the survey
   2. Used for structural regression analysis testing the three factor model on detection of computer uniform BS
5. RE\_DATA5\_OSF
   1. Dataset comprised of the 500 original true, clean responses, plus 100 human BS, 100 computer-normalized, and 100 computer-uniform responses to the survey
   2. Used for multinomial logistic regression testing the predictive strength of CR indices for each type of BS data
6. REI\_DATA6\_OSF
   1. Social science dataset comprised of 302 true responses
   2. Used for all analyses in study 2

[Analyses that require R Studio are specified; all other analyses should be conducted in SPSS]

**STUDY 1**

**GENERATING BS DATA**

\*\*generate random uniform data

COMPUTE CB1=RV.UNIFORM(1,6).

COMPUTE CB2=RV.UNIFORM(1,6).

COMPUTE CB3=RV.UNIFORM(1,6).

COMPUTE CB4=RV.UNIFORM(1,6).

COMPUTE CB5=RV.UNIFORM(1,6).

COMPUTE CB6=RV.UNIFORM(1,6).

COMPUTE CB7=RV.UNIFORM(1,6).

COMPUTE CB8=RV.UNIFORM(1,6).

COMPUTE CB9=RV.UNIFORM(1,6).

COMPUTE CB10=RV.UNIFORM(1,6).

COMPUTE CB11=RV.UNIFORM(1,6).

COMPUTE CB12=RV.UNIFORM(1,6).

COMPUTE CB13=RV.UNIFORM(1,6).

COMPUTE CB14=RV.UNIFORM(1,6).

COMPUTE CB15=RV.UNIFORM(1,6).

COMPUTE CB16=RV.UNIFORM(1,6).

COMPUTE CB17=RV.UNIFORM(1,6).

COMPUTE CB18=RV.UNIFORM(1,6).

COMPUTE CB19=RV.UNIFORM(1,6).

COMPUTE CB20=RV.UNIFORM(1,6).

COMPUTE X=RV.UNIFORM(1,6).

EXECUTE.

\*\*generates random normal data

COMPUTE CB1=RV.NORMAL(3.5,1.25).

COMPUTE CB2=RV.NORMAL(3.5,1.25).

COMPUTE CB3=RV.NORMAL(3.5,1.25).

COMPUTE CB4=RV.NORMAL(3.5,1.25).

COMPUTE CB5=RV.NORMAL(3.5,1.25).

COMPUTE CB6=RV.NORMAL(3.5,1.25).

COMPUTE CB7=RV.NORMAL(3.5,1.25).

COMPUTE CB8=RV.NORMAL(3.5,1.25).

COMPUTE CB9=RV.NORMAL(3.5,1.25).

COMPUTE CB10=RV.NORMAL(3.5,1.25).

COMPUTE CB11=RV.NORMAL(3.5,1.25).

COMPUTE CB12=RV.NORMAL(3.5,1.25).

COMPUTE CB13=RV.NORMAL(3.5,1.25).

COMPUTE CB14=RV.NORMAL(3.5,1.25).

COMPUTE CB15=RV.NORMAL(3.5,1.25).

COMPUTE CB16=RV.NORMAL(3.5,1.25).

COMPUTE CB17=RV.NORMAL(3.5,1.25).

COMPUTE CB18=RV.NORMAL(3.5,1.25).

COMPUTE CB19=RV.NORMAL(3.5,1.25).

COMPUTE CB20=RV.NORMAL(3.5,1.25).

EXECUTE.

\*\*recodes randomly selected values of 7 or greater to 6 and randomly selected values of 0 as 1

RECODE CB1 CB2 CB3 CB4 CB5 CB6 CB7 CB8 CB9 CB10 CB11 CB12 CB13 CB14 CB15 CB16 CB17 CB18 CB19 CB20

(0=1) (1=1) (2=2) (3=3) (4=4) (5=5) (6=6) (7=6) (8 = 6).

EXECUTE.

**CONSTRUCT AND NORMALIZE VARIABLES**

[use with dataset 1]

\*\*creates race variable

IF (ASIAN = 1 and BLACK = 0 and LATINO = 0 and NATIVE = 0 and WHITE = 0)

RACE2 = 1.

IF (ASIAN = 0 and BLACK = 1 and LATINO = 0 and NATIVE = 0 and WHITE = 0)

RACE2 = 2.

IF (ASIAN = 0 and BLACK = 0 and LATINO = 1 and NATIVE = 0 and WHITE = 0)

RACE2 = 3.

IF (ASIAN = 0 and BLACK = 0 and LATINO = 0 and NATIVE = 0 and WHITE = 1)

RACE2 = 4.

EXECUTE.

RECODE RACE2 (1=1) (2=2) (3=3) (4=4) (SYSMIS=5).

EXECUTE.

IF (ASIAN = 1 and BLACK = 0 and LATINO = 0 and NATIVE = 0 and WHITE = 0)

RACE3 = 1.

IF (ASIAN = 0 and BLACK = 1 and LATINO = 0 and NATIVE = 0 and WHITE = 0)

RACE3 = 2.

IF (ASIAN = 0 and BLACK = 0 and LATINO = 1 and NATIVE = 0 and WHITE = 0)

RACE3 = 3.

IF (ASIAN = 0 and BLACK = 0 and LATINO = 0 and NATIVE = 0 and WHITE = 1)

RACE3 = 4.

EXECUTE.

[use with datasets 1,2,3, and 4]

**RECODE COBRAS**

\*\*reverse score CB items

RECODE CB2 CB4 CB5 CB6 CB8 CB11 CB12 CB15 CB17 CB20 (1=6) (2=5) (3=4) (4=3) (5=2) (6=1) INTO

CB2r CB4r CB5r CB6r CB8r CB11r CB12r CB15r CB17r CB20r.

EXECUTE.

COMPUTE CoBRAS=CB1 + CB2r + CB3 + CB4r + CB5r + CB6r + CB7 + CB8r + CB9 + CB10 + CB11r + CB12r + CB13 + CB14 + CB15r + CB16 + CB17r + CB18 + CB19 + CB20r.

EXECUTE.

**CALCULATE VARIABLES**

\*\*calculates MD

REGRESSION

/MISSING LISTWISE

/STATISTICS COEFF OUTS R ANOVA

/DEPENDENT X

/METHOD=ENTER CB1 CB2r CB3 CB4r CB5r CB6r CB7 CB8r CB9 CB10 CB11r CB12r CB13 CB14 CB15r CB16 CB17r CB18 CB19 CB20r

/CASEWISE PLOT(ZRESID) OUTLIERS(3)

/SAVE MAHAL (MD).

exe.

\*\*calculates REI

COUNT VALUE1tally=CB1, CB2, CB3, CB4, CB5, CB6, CB7, CB8, CB9, CB10, CB11, CB12, CB13, CB14, CB15, CB16, CB17, CB18, CB19, CB20 (1).

COUNT VALUE2tally= CB1, CB2, CB3, CB4, CB5, CB6, CB7, CB8, CB9, CB10, CB11, CB12, CB13, CB14, CB15, CB16, CB17, CB18, CB19, CB20 (2).

COUNT VALUE3tally= CB1, CB2, CB3, CB4, CB5, CB6, CB7, CB8, CB9, CB10, CB11, CB12, CB13, CB14, CB15, CB16, CB17, CB18, CB19, CB20 (3).

COUNT VALUE4tally= CB1, CB2, CB3, CB4, CB5, CB6, CB7, CB8, CB9, CB10, CB11, CB12, CB13, CB14, CB15, CB16, CB17, CB18, CB19, CB20 (4).

COUNT VALUE5tally= CB1, CB2, CB3, CB4, CB5, CB6, CB7, CB8, CB9, CB10, CB11, CB12, CB13, CB14, CB15, CB16, CB17, CB18, CB19, CB20 (5).

COUNT VALUE6tally= CB1, CB2, CB3, CB4, CB5, CB6, CB7, CB8, CB9, CB10, CB11, CB12, CB13, CB14, CB15, CB16, CB17, CB18, CB19, CB20 (6).

EXECUTE.

COMPUTE VALUE1prop=VALUE1tally / (VALUE1tally + VALUE2tally + VALUE3tally + VALUE4tally +

VALUE5tally + VALUE6tally).

COMPUTE VALUE2prop=VALUE2tally / (VALUE1tally + VALUE2tally + VALUE3tally + VALUE4tally +

VALUE5tally + VALUE6tally).

COMPUTE VALUE3prop=VALUE3tally / (VALUE1tally + VALUE2tally + VALUE3tally + VALUE4tally +

VALUE5tally + VALUE6tally).

COMPUTE VALUE4prop=VALUE4tally / (VALUE1tally + VALUE2tally + VALUE3tally + VALUE4tally +

VALUE5tally + VALUE6tally).

COMPUTE VALUE5prop=VALUE5tally / (VALUE1tally + VALUE2tally + VALUE3tally + VALUE4tally +

VALUE5tally + VALUE6tally).

COMPUTE VALUE6prop=VALUE6tally / (VALUE1tally + VALUE2tally + VALUE3tally + VALUE4tally +

VALUE5tally + VALUE6tally).

EXECUTE.

COMPUTE REI= - ((VALUE1prop \* LG10(VALUE1prop)) + (VALUE2prop \* LG10(VALUE2prop)) + (VALUE3prop \* LG10(VALUE3prop)) + (VALUE4prop \* LG10(VALUE4prop)) + (VALUE5prop \* LG10(VALUE5prop))+ VALUE6prop \* LG10(VALUE6prop)).

EXECUTE.

ADD FILES FILE=\* /DROP=VALUE1tally,VALUE2tally,VALUE3tally,VALUE4tally,VALUE5tally, VALUE6tally, VALUE6prop, VALUE1prop,VALUE2prop,VALUE3prop,VALUE4prop,VALUE5prop.

EXECUTE.

\*\*calculates LS

VECTOR V = CB1 TO CB20.

COMPUTE #RUN = 1.

COMPUTE MAXRUN = 1.

LOOP #i = 2 TO 21.

DO IF V(#i) EQ V(#i-1) and V(#i) EQ 1.

COMPUTE #RUN = #RUN + 1.

COMPUTE MAXRUN1 = MAX(MAXRUN, #RUN).

ELSE.

COMPUTE #RUN = 1.

END IF.

END LOOP.

EXECUTE.

VECTOR V = CB1 TO CB20.

COMPUTE #RUN = 1.

COMPUTE MAXRUN = 1.

LOOP #i = 2 TO 21.

DO IF V(#i) EQ V(#i-1) and V(#i) EQ 1 + 1.

COMPUTE #RUN = #RUN + 1.

COMPUTE MAXRUN2 = MAX(MAXRUN, #RUN).

ELSE.

COMPUTE #RUN = 1.

END IF.

END LOOP.

EXECUTE.

VECTOR V = CB1 TO CB20.

COMPUTE #RUN = 1.

COMPUTE MAXRUN = 1.

LOOP #i = 2 TO 21.

DO IF V(#i) EQ V(#i-1) and V(#i) EQ 1 + 2.

COMPUTE #RUN = #RUN + 1.

COMPUTE MAXRUN3 = MAX(MAXRUN, #RUN).

ELSE.

COMPUTE #RUN = 1.

END IF.

END LOOP.

EXECUTE.

VECTOR V = CB1 TO CB20.

COMPUTE #RUN = 1.

COMPUTE MAXRUN = 1.

LOOP #i = 2 TO 21.

DO IF V(#i) EQ V(#i-1) and V(#i) EQ 1 + 3.

COMPUTE #RUN = #RUN + 1.

COMPUTE MAXRUN4 = MAX(MAXRUN, #RUN).

ELSE.

COMPUTE #RUN = 1.

END IF.

END LOOP.

EXECUTE.

VECTOR V = CB1 TO CB20.

COMPUTE #RUN = 1.

COMPUTE MAXRUN = 1.

LOOP #i = 2 TO 21.

DO IF V(#i) EQ V(#i-1) and V(#i) EQ 1 + 4.

COMPUTE #RUN = #RUN + 1.

COMPUTE MAXRUN5 = MAX(MAXRUN, #RUN).

ELSE.

COMPUTE #RUN = 1.

END IF.

END LOOP.

EXECUTE.

VECTOR V = CB1 TO CB20.

COMPUTE #RUN = 1.

COMPUTE MAXRUN = 1.

LOOP #i = 2 TO 21.

DO IF V(#i) EQ V(#i-1) and V(#i) EQ 1 + 5.

COMPUTE #RUN = #RUN + 1.

COMPUTE MAXRUN6 = MAX(MAXRUN, #RUN).

ELSE.

COMPUTE #RUN = 1.

END IF.

END LOOP.

EXECUTE.

COMPUTE LS=MAX(MAXRUN1,MAXRUN2,MAXRUN3,MAXRUN4, MAXRUN5, MAXRUN6).

EXECUTE.

ADD FILES FILE=\* /DROP=MAXRUN, MAXRUN1, MAXRUN2, MAXRUN3, MAXRUN4, MAXRUN5, MAXRUN6.

EXECUTE.

RECODE LS (SYSMIS=1) (2=2) (3=3) (4=4) (5=5) (6=6) (7=7) (8=8) (9=9) (10=10) (11=11)

(12=12) (13=13) (14=14) (15=15) (16=16) (17=17) (18=18) (19=19) (20=20).

EXECUTE.

\*\*calculates PAI based on correlations without reverse scored items

COMPUTE PScrossProd=(CB1 \* CB6) + (CB4 \* CB16) + (CB5 \* CB7) + (CB7 \* CB12) + (CB7 \* CB11).

EXECUTE.

COMPUTE PSavgA=(CB1 + CB4 + CB5 + CB7 + CB7) / 5.

EXECUTE.

COMPUTE PSsumAsq=(CB1 \*\* 2 + CB4 \*\* 2 + CB5 \*\* 2 + CB7 \*\* 2 + CB7 \*\* 2).

EXECUTE.

COMPUTE PSavgB=(CB6 + CB16 + CB7 + CB12 + CB11) / 5.

EXECUTE.

COMPUTE PSsumBsq=(CB6 \*\* 2 + CB16 \*\* 2 + CB7 \*\* 2 + CB12 \*\* 2 + CB11 \*\* 2).

EXECUTE.

COMPUTE PSNum=PScrossProd - (5 \* PSavgA \* PSavgB).

EXECUTE.

COMPUTE PSD1=PSsumAsq - (5 \* (PSavgA \*\* 2)).

EXECUTE.

COMPUTE PSD2=PSsumBsq - (5 \* (PSavgB \*\* 2)).

EXECUTE.

COMPUTE PSDen=SQRT(PSD1 \* PSD2).

EXECUTE.

COMPUTE PAI=PSNum / PSDen.

EXECUTE.

ADD FILES FILE=\* /DROP= PScrossProd, PSavgA, PSsumAsq, PSavgB, PSsumBsq, PSNum, PSD1, PSD2, PSDen.

EXECUTE.

\*\*calculates PSI based on correlations with reverse scored items

CORRELATIONS

/VARIABLES=CB1 CB2r CB3 CB4r CB5r CB6r CB7 CB8r CB9 CB10 CB11r CB12r CB13 CB14 CB15r CB16 CB17r

CB18 CB19 CB20r

/PRINT=TWOTAIL NOSIG

/MISSING=PAIRWISE.

OUTPUT MODIFY

/REPORT PRINTREPORT=NO

/SELECT TABLES

/IF COMMANDS=[LAST] SUBTYPES="Correlations"

/TABLECELLS SELECT=[CORRELATION] SELECTDIMENSION=ROWS SELECTCONDITION="Abs(x)>=0.575"

BACKGROUNDCOLOR=RGB(255, 255, 0) APPLYTO=CELL.

COMPUTE PScrossProd=(CB7 \* CB5r) + (CB12r \* CB5r) + (CB7 \* CB12r) + (CB8r \* CB12r) + (CB11r \* CB12r) + (CB15r \* CB12r) + (CB19 \* CB7) + (CB20r \* CB15r) + (CB20r \* CB12r) + (CB20r \* CB8r).

EXECUTE.

COMPUTE PSavgA=(CB7 + CB12r + CB7 + CB8r + CB12r + CB15r + CB19 + CB20r + CB20r + CB20r) / 10.

EXECUTE.

COMPUTE PSsumAsq=( CB7\*\*2 + CB12r\*\*2 + CB7\*\*2 + CB8r\*\*2 + CB12r\*\*2 + CB15r\*\*2 + CB19\*\*2 + CB20r\*\*2 + CB20r\*\*2 + CB20r\*\*2).

EXECUTE.

COMPUTE PSavgB=(CB5r + CB5r + CB12r + CB12r + CB12r + CB11r + CB7 + CB15r + CB12r + CB8r) / 10.

EXECUTE.

COMPUTE PSsumBsq=(CB5r\*\*2 + CB5r\*\*2 + CB12r\*\*2 + CB12r\*\*2 + CB12r\*\*2 + CB11r\*\*2 + CB7\*\*2 + CB15r\*\*2 + CB12r\*\*2 + CB8r\*\*2).

EXECUTE.

COMPUTE PSNum=PScrossProd - (10 \* PSavgA \* PSavgB).

EXECUTE.

COMPUTE PSD1=PSsumAsq - (10 \* (PSavgA \*\* 2)).

EXECUTE.

COMPUTE PSD2=PSsumBsq - (10 \* (PSavgB \*\* 2)).

EXECUTE.

COMPUTE PSDen=SQRT(PSD1 \* PSD2).

EXECUTE.

COMPUTE PSI=PSNum / PSDen.

EXECUTE.

ADD FILES FILE=\* /DROP= PScrossProd, PSavgA, PSsumAsq, PSavgB, PSsumBsq, PSNum, PSD1, PSD2, PSDen.

EXECUTE.

\*\*calculates EOI with reverse scored items

COMPUTE PScrossProd=(CB1 \* CB2r) + (CB3 \* CB4r) + (CB5r \* CB6r) + (CB7 \* CB8r) + (CB9 \* CB10) + (CB11r \* CB12r) + (CB13 \* CB14) + (CB15r \* CB16) + (CB17r \* CB18) + (CB19 \* CB20r).

EXECUTE.

COMPUTE PSavgA=(CB1 + CB3 + CB5r + CB7 + CB9 + CB11r + CB13 + CB15r + CB17r + CB19) / 10.

EXECUTE.

COMPUTE PSsumAsq=(CB1 \*\* 2 + CB3 \*\* 2 + CB5r \*\* 2 + CB7 \*\* 2 + CB9 \*\* 2 + CB11r \*\* 2 + CB13 \*\* 2 + CB15r \*\* 2 + CB17r \*\* 2 + + CB19 \*\* 2).

EXECUTE.

COMPUTE PSavgB=(CB2r + CB4r + CB6r+ CB8r + CB10 + CB12r + CB14 + CB16 + CB18 + CB20r) / 10.

EXECUTE.

COMPUTE PSsumBsq=(CB2r \*\* 2 + CB4r \*\* 2 + CB6r \*\* 2 + CB8r \*\* 2 + CB10 \*\* 2 + CB12r \*\* 2 + CB14 \*\* 2 + CB16 \*\* 2 + CB18 \*\* 2 + CB20r \*\* 2).

EXECUTE.

COMPUTE PSNum=PScrossProd - (10 \* PSavgA \* PSavgB).

EXECUTE.

COMPUTE PSD1=PSsumAsq - (10 \* (PSavgA \*\* 2)).

EXECUTE.

COMPUTE PSD2=PSsumBsq - (10 \* (PSavgB \*\* 2)).

EXECUTE.

COMPUTE PSDen=SQRT(PSD1 \* PSD2).

EXECUTE.

COMPUTE EOI1=PSNum / PSDen.

EXECUTE.

ADD FILES FILE=\* /DROP= PScrossProd, PSavgA, PSsumAsq, PSavgB, PSsumBsq, PSNum, PSD1, PSD2, PSDen.

EXECUTE.

\*\*adjusts EOI score using Spearman-Brown Prediction Formula and recoded values less than -1 to -1

COMPUTE EOI2=2 \* EOI1 / (1 + (2 - 1) \* EOI1).

EXECUTE.

RECODE EOI2 (Lowest thru -1=-1) (ELSE=Copy) INTO EOI.

EXECUTE.

ADD FILES FILE=\* /DROP= EOI1, EOI2.

EXECUTE.

\*\*calculates STDEV

COMPUTE STDEV=VARIANCE(CB1,CB2r,CB3,CB4r,CB5r,CB6r,CB7,CB8r,CB9,CB10,CB11r,CB12r,CB13,CB14,CB15r,CB16,CB17r, CB18,CB19,CB20r).

EXECUTE.

**PERSON FIT STATISTICS**

[computation for person fit statistics conducted in R studio (guttman errors, guttman errors normed, and standardized log likelihood variables are already included in SPSS datasets 1- 6)]

\*\*open PerFit package

\*\*create new file for only CB items

\*\*recode to include 0

RECODE CB1 CB2 CB3 CB4 CB5 CB6 CB7 CB8 CB9 CB10 CB11 CB12 CB13 CB14 CB15 CB16 CB17 CB18 CB19 CB20

(6=5) (5=4) (4=3) (3=2) (2=1) (1=0).

EXECUTE.

\*\*save as csv with no headings and import to R

\*\*human datafile

human\_matrix<- as.matrix(human)

sink("output.human.csv")

Gpoly(human\_matrix, Ncat=6,NA.method = "Pairwise", Save.MatImp = FALSE,IP = NULL, IRT.PModel = "GRM", Ability = NULL, Ability.PModel = "EAP")

lzpoly(human\_matrix, Ncat=6,NA.method = "Pairwise", Save.MatImp = FALSE,IP = NULL, IRT.PModel = "GRM", Ability = NULL, Ability.PModel = "EAP")

sink()

**NORMALIZE VARIABLES**

DESCRIPTIVES VARIABLES=MD REI LS PAI PSI EOI STDEV GUTT GUTTN LZ

/STATISTICS=MEAN STDDEV MIN MAX KURTOSIS SKEWNESS.

\*\*add constant to PAI and LZ since there are negative values

COMPUTE PAIp10=PAI + 10.

COMPUTE LZp10=LZ + 10.

EXECUTE.

\*\*computes logs for skewed variables

COMPUTE MD\_log=LG10(MD).

COMPUTE LS\_log=LG10(LS).

COMPUTE PAI\_log=LG10(PAIp10).

COMPUTE GUTT\_log=LG10(GUTT).

COMPUTE GUTTN\_log=LG10(GUTTN).

COMPUTE LZ\_log=LG10(LZp10).

EXECUTE.

FREQUENCIES VARIABLES=MD\_log LS\_log PAI\_log GUTT\_log GUTTN\_log LZ\_log

/FORMAT=NOTABLE

/STATISTICS=SKEWNESS SESKEW KURTOSIS SEKURT

/HISTOGRAM NORMAL

/ORDER=ANALYSIS.

\*\* creates square root transformation and examines distributions and skewness and kurtosis

COMPUTE LS\_sq=SQRT(LS).

COMPUTE PAI\_sq=SQRT(PAIp10).

EXECUTE.

FREQUENCIES VARIABLES= LS\_sq PAI\_sq

/FORMAT=NOTABLE

/STATISTICS=SKEWNESS SESKEW KURTOSIS SEKURT

/HISTOGRAM NORMAL

/ORDER=ANALYSIS.

\*\* creates cube root transformation and examines distributions and skewness and kurtosis

COMPUTE LS\_cube=LS\*\* (1/3).

COMPUTE PAI\_cube=PAIp10\*\* (1/3).

EXECUTE.

FREQUENCIES VARIABLES=LS\_cube PAI\_cube

/FORMAT=NOTABLE

/STATISTICS=SKEWNESS SESKEW KURTOSIS SEKURT

/HISTOGRAM NORMAL

/ORDER=ANALYSIS.

\*\* creates reciprocal transformation and examines distributions and skewness and kurtosis

COMPUTE LS\_recip=1/LS.

COMPUTE PAI\_recip=1/PAIp10.

EXECUTE.

FREQUENCIES VARIABLES=LS\_recip PAI\_recip

/FORMAT=NOTABLE

/STATISTICS=SKEWNESS SESKEW KURTOSIS SEKURT

/HISTOGRAM NORMAL

/ORDER=ANALYSIS.

\*\* creates square and cube transformations and examines distributions and skewness and kurtosis

COMPUTE REI\_2=REI \*\* 2.

COMPUTE LZ\_2=LZ\*\*2.

COMPUTE LZ\_3=LZ\*\*3.

EXECUTE.

FREQUENCIES VARIABLES= REI\_2 LZ\_2 LZ\_3

/FORMAT=NOTABLE

/STATISTICS=SKEWNESS SESKEW KURTOSIS SEKURT

/HISTOGRAM NORMAL

/ORDER=ANALYSIS.

**ANALYSES**

\*\*examines each index by race, immigration status, and gender

GLM MD REI LS PAI PSI EOI STDEV GUTTN LZ BY RACE3

/METHOD=SSTYPE(3)

/INTERCEPT=INCLUDE

/PRINT=ETASQ

/CRITERIA=ALPHA(.05)

/DESIGN= RACE3.

GLM MD REI LS PAI PSI EOI STDEV GUTTN LZ BY GENDER

/METHOD=SSTYPE(3)

/INTERCEPT=INCLUDE

/PRINT=ETASQ

/CRITERIA=ALPHA(.05)

/DESIGN= GENDER.

GLM MD REI LS PAI PSI EOI STDEV GUTTN LZ BY IMMIG

/METHOD=SSTYPE(3)

/INTERCEPT=INCLUDE

/PRINT=ETASQ

/CRITERIA=ALPHA(.05)

/DESIGN= IMMIG.

ONEWAY MD REI LS PAI PSI EOI STDEV GUTTN LZ BY RACE3

/STATISTICS DESCRIPTIVES

/MISSING ANALYSIS

/POSTHOC=TUKEY ALPHA(0.05).

ONEWAY MD REI LS PAI PSI EOI STDEV GUTTN LZ BY GENDER

/STATISTICS DESCRIPTIVES

/MISSING ANALYSIS

/POSTHOC=TUKEY ALPHA(0.05).

ONEWAY MD REI LS PAI PSI EOI STDEV GUTTN LZ BY IMMIG

/STATISTICS DESCRIPTIVES

/MISSING ANALYSIS

/POSTHOC=TUKEY ALPHA(0.05).

\*\*examines intercorrelations between metrics and relationship to age

CORRELATIONS

/VARIABLES=AGE MD REI LS PAI PSI EOI STDEV GUTTN LZ

/PRINT=TWOTAIL NOSIG

/MISSING=PAIRWISE.

\*\*logistic regression with standardized variables

DESCRIPTIVES VARIABLES= MD REI LS PAI PSI EOI STDEV GUTTN LZ

/SAVE

/STATISTICS=MEAN STDDEV MIN MAX.

**LOGISTIC REGRESSIONS**

[use dataset 5: compute z scores in SPSS and logistic regression in R studio]

LOGISTIC REGRESSION VARIABLES BS

/METHOD=ENTER ZMD ZPSI ZPAI ZEOI ZLS ZREI ZSTDEV ZGUTTN ZLZ

/PRINT=CI(95)

/CRITERIA=PIN(0.05) POUT(0.10) ITERATE(20) CUT(0.5).

\*\*multinomial regression: load (nnet and foreign packages in r)

\*\*step 0 (each index with no other variables in the equation)

REI\_MULTINOM$BSCAT <- relevel(MULTINOM$BSCAT, ref = "careful")

test <- multinom(BSCAT ~ ZMD , data = RE\_DATA5\_OSF)

summary(test)

z <- summary(test)$coefficients/summary(test)$standard.errors

z

exp(coef(test))

REI\_MULTINOM$BSCAT <- relevel(MULTINOM$BSCAT, ref = "careful")

test <- multinom(BSCAT ~ ZPSI , data = RE\_DATA5\_OSF)

summary(test)

z <- summary(test)$coefficients/summary(test)$standard.errors

z

exp(coef(test))

REI\_MULTINOM$BSCAT <- relevel(MULTINOM$BSCAT, ref = "careful")

test <- multinom(BSCAT ~ ZPAI , data = RE\_DATA5\_OSF)

summary(test)

z <- summary(test)$coefficients/summary(test)$standard.errors

z

exp(coef(test))

REI\_MULTINOM$BSCAT <- relevel(MULTINOM$BSCAT, ref = "careful")

test <- multinom(BSCAT ~ ZEOI , data = RE\_DATA5\_OSF)

summary(test)

z <- summary(test)$coefficients/summary(test)$standard.errors

z

exp(coef(test))

REI\_MULTINOM$BSCAT <- relevel(MULTINOM$BSCAT, ref = "careful")

test <- multinom(BSCAT ~ ZLS , data = RE\_DATA5\_OSF)

summary(test)

z <- summary(test)$coefficients/summary(test)$standard.errors

z

exp(coef(test))

REI\_MULTINOM$BSCAT <- relevel(MULTINOM$BSCAT, ref = "careful")

test <- multinom(BSCAT ~ ZREI , data = RE\_DATA5\_OSF)

summary(test)

z <- summary(test)$coefficients/summary(test)$standard.errors

z

exp(coef(test))

REI\_MULTINOM$BSCAT <- relevel(MULTINOM$BSCAT, ref = "careful")

test <- multinom(BSCAT ~ ZSTDEV , data = RE\_DATA5\_OSF)

summary(test)

z <- summary(test)$coefficients/summary(test)$standard.errors

z

exp(coef(test))

REI\_MULTINOM$BSCAT <- relevel(MULTINOM$BSCAT, ref = "careful")

test <- multinom(BSCAT ~ ZGUTTN , data = RE\_DATA5\_OSF)

summary(test)

z <- summary(test)$coefficients/summary(test)$standard.errors

z

exp(coef(test))

REI\_MULTINOM$BSCAT <- relevel(MULTINOM$BSCAT, ref = "careful")

test <- multinom(BSCAT ~ ZLZ , data = RE\_DATA5\_OSF)

summary(test)

z <- summary(test)$coefficients/summary(test)$standard.errors

z

exp(coef(test))

\*\*step 1 (all variables together in the model)

REI\_MULTINOM$BSCAT <- relevel(MULTINOM$BSCAT, ref = "careful")

test <- multinom(BSCAT ~ ZMD + ZPSI + ZPAI + ZEOI + ZLS + ZREI + ZSTDEV + ZGUTTN + ZLZ , data = RE\_DATA5\_OSF)

summary(test)

p <- (1 - pnorm(abs(z), 0, 1)) \* 2

p

z <- summary(test)$coefficients/summary(test)$standard.errors

z

exp(coef(test))

**FACTOR ANALYSES**

[use dataset 1]

\*\*EFA (does not converge)

FACTOR

/VARIABLES GUTTN\_log LZ REI\_2 PAI\_recip LS\_recip STDEV EOI PSI MD\_log

/MISSING LISTWISE

/ANALYSIS GUTTN\_log LZ REI\_2 PAI\_recip LS\_recip STDEV EOI PSI MD\_log

/PRINT INITIAL KMO EXTRACTION ROTATION

/CRITERIA MINEIGEN(1) ITERATE(25)

/EXTRACTION PAF

/CRITERIA ITERATE(25)

/ROTATION PROMAX(4)

/METHOD=CORRELATION.

[use dataset 1: conduct in R Studio]

\*\*CFA based on theory

threefactor <- '

A =~ REI\_2 + LS\_recip + STDEV

B =~ PSI + PAI\_recip + EOI + MD\_log

C =~ GUTTN\_log + LZ

'

fit <- cfa (threefactor, data = RE\_DATA1\_OSF, std.lv = TRUE, orthogonal = FALSE)

summary(fit, standardized = TRUE, fit.measures=TRUE, rsq = TRUE)

modindices(fit, sort. = TRUE)

semPaths(fit, whatLabels = "std", layout = "tree")

\*\*doesn’t converge, examined correlation matrix…

threefactor <- '

Variability =~ REI\_2 + LS\_recip + STDEV

Regression =~ PSI + PAI\_recip + EOI

PersonFit =~ GUTT\_log + LZ + MD\_log

'

fit <- cfa (threefactor, data = RE\_DATA1\_OSF, std.lv = TRUE, orthogonal = FALSE)

summary(fit, standardized = TRUE, fit.measures=TRUE, rsq = TRUE)

modindices(fit, sort. = TRUE)

semPaths(fit, whatLabels = "std", layout = "tree")

\*\*modified

threefactor <- '

Variability =~ REI\_2 + LS\_recip + STDEV

Regression =~ PSI + PAI\_recip + EOI

PersonFit =~ GUTT\_log + LZ + MD\_log

STDEV ~~ LZ

STDEV ~~ MD\_log

STDEV ~~ GUTT\_log

'

fit <- cfa (threefactor, data = RE\_DATA1\_OSF, std.lv = TRUE, orthogonal = FALSE)

summary(fit, standardized = TRUE, fit.measures=TRUE, rsq = TRUE)

modindices(fit, sort. = TRUE)

semPaths(fit, whatLabels = "std", layout = "tree")

\*\*one factor

onefactor <- '

All =~ REI\_2 + LS\_recip + STDEV + PSI + PAI\_recip + EOI + MD\_log + GUTTN\_log + LZ

'

fit <- cfa (onefactor, data = RE\_DATA1\_OSF, std.lv = TRUE, orthogonal = FALSE)

summary(fit, standardized = TRUE, fit.measures=TRUE, rsq = TRUE)

modindices(fit, sort. = TRUE)

semPaths(fit, whatLabels = "std", layout = "tree")

**STRUCTURAL REGRESSION**

threefactor <- '

Variability =~ REI\_2 + LS\_recip + STDEV

Regression =~ PSI + PAI\_recip + EOI

PersonFit =~ GUTT\_log + LZ + MD\_log

STDEV ~~ LZ

STDEV ~~ MD\_log

STDEV ~~ GUTTN\_log

BS ~ Variability

BS ~ Regression

BS ~ PersonFit

'

fit <- cfa (threefactor, data = RE\_DATA2\_OSF, std.lv = TRUE, orthogonal = FALSE)

summary(fit, standardized = TRUE, fit.measures=TRUE, rsq = TRUE)

modindices(fit, sort. = TRUE)

semPaths(fit, whatLabels = "std", layout = "tree")

**threefactor <- '**

**Variability =~ REI\_2 + LS\_recip + STDEV**

**Regression =~ PSI + PAI\_recip + EOI**

**PersonFit =~ GUTT\_log + LZ + MD\_log**

**STDEV ~~ GUTT\_log**

**STDEV ~~ MD\_log**

**STDEV ~~ LZ**

**BS ~ Variability**

**BS ~ Regression**

**BS ~ PersonFit**

**'**

**fit <- cfa (threefactor, data = RE\_DATA2\_OSF\_rev, std.lv = TRUE, orthogonal = FALSE)**

**summary(fit, standardized = TRUE, fit.measures=TRUE, rsq = TRUE)**

**modindices(fit, sort. = TRUE)**

**semPaths(fit, whatLabels = "std", layout = "tree")**

**STUDY 2**

[use dataset 6]

**CONSTRUCT AND NORMALIZE VARIABLES**

IF (ASIAN = 1 and BLACK = 0 and LATINO = 0 and NATIVE = 0 and WHITE = 0)

RACE = 1.

IF (ASIAN = 0 and BLACK = 1 and LATINO = 0 and NATIVE = 0 and WHITE = 0)

RACE = 2.

IF (ASIAN = 0 and BLACK = 0 and LATINO = 1 and NATIVE = 0 and WHITE = 0)

RACE = 3.

IF (ASIAN = 0 and BLACK = 0 and LATINO = 0 and NATIVE = 0 and WHITE = 1)

RACE = 5.

EXECUTE.

RECODE GEN2 BEH4 (6=1) (5=2) (4=3) (3=4) (2=5) (1=6) INTO GEN2r BEH4r.

EXECUTE.

COMPUTE BARS=SPEC1 + SPEC2 + SPEC3 + SPEC4 + GEN1 + GEN2r + GEN3 + GEN4 + PHEN1 + PHEN2 + PHEN3 + PHEN4 + BEH1 + BEH2 + BEH3 + BEH4r.

EXECUTE.

**CALCULATE VARIABLES**

\*\*calculates MD

REGRESSION

/MISSING LISTWISE

/STATISTICS COEFF OUTS R ANOVA

/DEPENDENT X

/METHOD=ENTER SPEC1 SPEC2 SPEC3 SPEC4 GEN1 GEN2r GEN3 GEN4 PHEN1 PHEN2 PHEN3 PHEN4 BEH1 BEH2 BEH3 BEH4r

/CASEWISE PLOT(ZRESID) OUTLIERS(3)

/SAVE MAHAL (MD).

exe.

\*\*\*\*calculates REI

COUNT VALUE1tally= SPEC1, SPEC2, SPEC3, SPEC4, GEN1, GEN2, GEN3, GEN4, PHEN1, PHEN2, PHEN3, PHEN4, BEH1, BEH2, BEH3, BEH4 (1).

COUNT VALUE2tally= SPEC1, SPEC2, SPEC3, SPEC4, GEN1, GEN2, GEN3, GEN4, PHEN1, PHEN2, PHEN3, PHEN4, BEH1, BEH2, BEH3, BEH4 (2).

COUNT VALUE3tally= SPEC1, SPEC2, SPEC3, SPEC4, GEN1, GEN2, GEN3, GEN4, PHEN1, PHEN2, PHEN3, PHEN4, BEH1, BEH2, BEH3, BEH4 (3).

COUNT VALUE4tally= SPEC1, SPEC2, SPEC3, SPEC4, GEN1, GEN2, GEN3, GEN4, PHEN1, PHEN2, PHEN3, PHEN4, BEH1, BEH2, BEH3, BEH4 (4).

COUNT VALUE5tally= SPEC1, SPEC2, SPEC3, SPEC4, GEN1, GEN2, GEN3, GEN4, PHEN1, PHEN2, PHEN3, PHEN4, BEH1, BEH2, BEH3, BEH4 (5).

COUNT VALUE6tally= SPEC1, SPEC2, SPEC3, SPEC4, GEN1, GEN2, GEN3, GEN4, PHEN1, PHEN2, PHEN3, PHEN4, BEH1, BEH2, BEH3, BEH4 (6).

EXECUTE.

COMPUTE VALUE1prop=VALUE1tally / (VALUE1tally + VALUE2tally + VALUE3tally + VALUE4tally +

VALUE5tally + VALUE6tally).

COMPUTE VALUE2prop=VALUE2tally / (VALUE1tally + VALUE2tally + VALUE3tally + VALUE4tally +

VALUE5tally + VALUE6tally).

COMPUTE VALUE3prop=VALUE3tally / (VALUE1tally + VALUE2tally + VALUE3tally + VALUE4tally +

VALUE5tally + VALUE6tally).

COMPUTE VALUE4prop=VALUE4tally / (VALUE1tally + VALUE2tally + VALUE3tally + VALUE4tally +

VALUE5tally + VALUE6tally).

COMPUTE VALUE5prop=VALUE5tally / (VALUE1tally + VALUE2tally + VALUE3tally + VALUE4tally +

VALUE5tally + VALUE6tally).

COMPUTE VALUE6prop=VALUE6tally / (VALUE1tally + VALUE2tally + VALUE3tally + VALUE4tally +

VALUE5tally + VALUE6tally).

EXECUTE.

COMPUTE REI= - ((VALUE1prop \* LG10(VALUE1prop)) + (VALUE2prop \* LG10(VALUE2prop)) + (VALUE3prop \* LG10(VALUE3prop)) + (VALUE4prop \* LG10(VALUE4prop)) + (VALUE5prop \* LG10(VALUE5prop))+ VALUE6prop \* LG10(VALUE6prop)).

EXECUTE.

ADD FILES FILE=\* /DROP=VALUE1tally,VALUE2tally,VALUE3tally,VALUE4tally,VALUE5tally, VALUE6tally, VALUE6prop, VALUE1prop,VALUE2prop,VALUE3prop,VALUE4prop,VALUE5prop.

EXECUTE.

\*\*calculates LS

VECTOR V = SPEC1 TO BEH4.

COMPUTE #RUN = 1.

COMPUTE MAXRUN = 1.

LOOP #i = 2 TO 21.

DO IF V(#i) EQ V(#i-1) and V(#i) EQ 1.

COMPUTE #RUN = #RUN + 1.

COMPUTE MAXRUN1 = MAX(MAXRUN, #RUN).

ELSE.

COMPUTE #RUN = 1.

END IF.

END LOOP.

EXECUTE.

VECTOR V = SPEC1 TO BEH4.

COMPUTE #RUN = 1.

COMPUTE MAXRUN = 1.

LOOP #i = 2 TO 21.

DO IF V(#i) EQ V(#i-1) and V(#i) EQ 1 + 1.

COMPUTE #RUN = #RUN + 1.

COMPUTE MAXRUN2 = MAX(MAXRUN, #RUN).

ELSE.

COMPUTE #RUN = 1.

END IF.

END LOOP.

EXECUTE.

VECTOR V = SPEC1 TO BEH4.

COMPUTE #RUN = 1.

COMPUTE MAXRUN = 1.

LOOP #i = 2 TO 21.

DO IF V(#i) EQ V(#i-1) and V(#i) EQ 1 + 2.

COMPUTE #RUN = #RUN + 1.

COMPUTE MAXRUN3 = MAX(MAXRUN, #RUN).

ELSE.

COMPUTE #RUN = 1.

END IF.

END LOOP.

EXECUTE.

VECTOR V = SPEC1 TO BEH4.

COMPUTE #RUN = 1.

COMPUTE MAXRUN = 1.

LOOP #i = 2 TO 21.

DO IF V(#i) EQ V(#i-1) and V(#i) EQ 1 + 3.

COMPUTE #RUN = #RUN + 1.

COMPUTE MAXRUN4 = MAX(MAXRUN, #RUN).

ELSE.

COMPUTE #RUN = 1.

END IF.

END LOOP.

EXECUTE.

VECTOR V = SPEC1 TO BEH4.

COMPUTE #RUN = 1.

COMPUTE MAXRUN = 1.

LOOP #i = 2 TO 21.

DO IF V(#i) EQ V(#i-1) and V(#i) EQ 1 + 4.

COMPUTE #RUN = #RUN + 1.

COMPUTE MAXRUN5 = MAX(MAXRUN, #RUN).

ELSE.

COMPUTE #RUN = 1.

END IF.

END LOOP.

EXECUTE.

VECTOR V = SPEC1 TO BEH4.

COMPUTE #RUN = 1.

COMPUTE MAXRUN = 1.

LOOP #i = 2 TO 21.

DO IF V(#i) EQ V(#i-1) and V(#i) EQ 1 + 5.

COMPUTE #RUN = #RUN + 1.

COMPUTE MAXRUN6 = MAX(MAXRUN, #RUN).

ELSE.

COMPUTE #RUN = 1.

END IF.

END LOOP.

EXECUTE.

COMPUTE LS=MAX(MAXRUN1,MAXRUN2,MAXRUN3,MAXRUN4, MAXRUN5, MAXRUN6).

EXECUTE.

ADD FILES FILE=\* /DROP=MAXRUN, MAXRUN1, MAXRUN2, MAXRUN3, MAXRUN4, MAXRUN5, MAXRUN6.

EXECUTE.

RECODE LS (SYSMIS=1) (2=2) (3=3) (4=4) (5=5) (6=6) (7=7) (8=8) (9=9) (10=10) (11=11)

(12=12) (13=13) (14=14) (15=15) (16=16) (17=17) (18=18) (19=19) (20=20).

EXECUTE.

\*\*\*\*calculates PAI without reverse scored items

CORRELATIONS

/VARIABLES= SPEC1 SPEC2 SPEC3 SPEC4 GEN1 GEN2 GEN3 GEN4 PHEN1 PHEN2 PHEN3 PHEN4 BEH1 BEH2 BEH3 BEH4 /PRINT=TWOTAIL NOSIG

/MISSING=PAIRWISE.

OUTPUT MODIFY

/REPORT PRINTREPORT=NO

/SELECT TABLES

/IF COMMANDS=[LAST] SUBTYPES="Correlations"

/TABLECELLS SELECT=[CORRELATION] SELECTDIMENSION=ROWS SELECTCONDITION="Abs(x)<=-.500"

BACKGROUNDCOLOR=RGB(255, 255, 0) APPLYTO=CELL.

COMPUTE PScrossProd=(GEN2 \* GEN1) + (GEN2 \* GEN3) + (BEH4 \* BEH1) + (BEH4 \* BEH2) + (BEH4 \* BEH3).

EXECUTE.

COMPUTE PSavgA=(GEN2 + GEN2 + BEH4 + BEH4 + BEH4) / 5.

EXECUTE.

COMPUTE PSsumAsq=(GEN2 \*\* 2 + GEN2 \*\* 2 + BEH4 \*\* 2 + BEH4 \*\* 2 + BEH4 \*\* 2).

EXECUTE.

COMPUTE PSavgB=(GEN1 + GEN3 + BEH1 + BEH2 + BEH3) / 5.

EXECUTE.

COMPUTE PSsumBsq=(GEN1 \*\* 2 + GEN3 \*\* 2 + BEH1 \*\* 2 + BEH2 \*\* 2 + BEH3 \*\* 2).

EXECUTE.

COMPUTE PSNum=PScrossProd - (5 \* PSavgA \* PSavgB).

EXECUTE.

COMPUTE PSD1=PSsumAsq - (5 \* (PSavgA \*\* 2)).

EXECUTE.

COMPUTE PSD2=PSsumBsq - (5 \* (PSavgB \*\* 2)).

EXECUTE.

COMPUTE PSDen=SQRT(PSD1 \* PSD2).

EXECUTE.

COMPUTE PAI=PSNum / PSDen.

EXECUTE.

ADD FILES FILE=\* /DROP= PScrossProd, PSavgA, PSsumAsq, PSavgB, PSsumBsq, PSNum, PSD1, PSD2, PSDen.

EXECUTE.

\*\*calculates PSI based on reverse scored items

CORRELATIONS

/VARIABLES= SPEC1 SPEC2 SPEC3 SPEC4 GEN1 GEN2r GEN3 GEN4 PHEN1 PHEN2 PHEN3 PHEN4 BEH1 BEH2 BEH3 BEH4r /PRINT=TWOTAIL NOSIG

/MISSING=PAIRWISE.

OUTPUT MODIFY

/REPORT PRINTREPORT=NO

/SELECT TABLES

/IF COMMANDS=[LAST] SUBTYPES="Correlations"

/TABLECELLS SELECT=[CORRELATION] SELECTDIMENSION=ROWS SELECTCONDITION="Abs(x)>=.500"

BACKGROUNDCOLOR=RGB(255, 255, 0) APPLYTO=CELL.

COMPUTE PScrossProd=(SPEC2 \* SPEC3) + (SPEC2 \* SPEC4) + (GEN1 \* GEN3) + (GEN1 \* GEN4) + (PHEN1 \* PHEN2) + (PHEN1 \* PHEN3) + (PHEN2 \* PHEN3) + (BEH1 \* BEH2) + (BEH1 \* BEH3) + (BEH2 \* BEH3).

EXECUTE.

COMPUTE PSavgA=(SPEC2 + SPEC2 + GEN1 + GEN1 + PHEN1 + PHEN1 + PHEN2 + BEH1 + BEH1 + BEH2) / 10.

EXECUTE.

COMPUTE PSsumAsq=( SPEC2\*\*2 + SPEC2\*\*2 + GEN1\*\*2 + GEN1\*\*2 + PHEN1\*\*2 + PHEN1\*\*2 + PHEN2\*\*2 + BEH1\*\*2 + BEH1\*\*2 + BEH2\*\*2).

EXECUTE.

COMPUTE PSavgB=(SPEC3 + SPEC4 + GEN3 + GEN4 + PHEN2 + PHEN3 + PHEN3 + BEH2 + BEH3 + BEH3) / 10.

EXECUTE.

COMPUTE PSsumBsq=(SPEC3\*\*2 + SPEC4\*\*2 + GEN3\*\*2 + GEN4\*\*2 + PHEN2\*\*2 + PHEN3\*\*2 + PHEN3\*\*2 + BEH2\*\*2 + BEH3\*\*2 + BEH3\*\*2).

EXECUTE.

COMPUTE PSNum=PScrossProd - (10 \* PSavgA \* PSavgB).

EXECUTE.

COMPUTE PSD1=PSsumAsq - (10 \* (PSavgA \*\* 2)).

EXECUTE.

COMPUTE PSD2=PSsumBsq - (10 \* (PSavgB \*\* 2)).

EXECUTE.

COMPUTE PSDen=SQRT(PSD1 \* PSD2).

EXECUTE.

COMPUTE PSI=PSNum / PSDen.

EXECUTE.

ADD FILES FILE=\* /DROP= PScrossProd, PSavgA, PSsumAsq, PSavgB, PSsumBsq, PSNum, PSD1, PSD2, PSDen.

EXECUTE.

\*\*calculates EOI based on reverse scored items

COMPUTE PScrossProd=(SPEC1 \* SPEC2) + (SPEC3 \* SPEC4) + (GEN1 \* GEN2r) + (GEN3 \* GEN4) + (PHEN1 \* PHEN2) + (PHEN3 \* PHEN4) + (BEH1 \* BEH2) + (BEH3 \* BEH4r).

EXECUTE.

COMPUTE PSavgA=(SPEC1 + SPEC3 + GEN1 + GEN3 + PHEN1 + PHEN3 + BEH1 + BEH3) / 8.

EXECUTE.

COMPUTE PSsumAsq=(SPEC1 \*\* 2 + SPEC3 \*\* 2 + GEN1 \*\* 2 + GEN3 \*\* 2 + PHEN1 \*\* 2 + PHEN3 \*\* 2 + BEH1 \*\* 2 + BEH3 \*\* 2).

EXECUTE.

COMPUTE PSavgB=(SPEC2 + SPEC4 + GEN2r+ GEN4 + PHEN2 + PHEN4 + BEH2 + BEH4r) / 8.

EXECUTE.

COMPUTE PSsumBsq=(SPEC2 \*\* 2 + SPEC4 \*\* 2 + GEN2r \*\* 2 + GEN4 \*\* 2 + PHEN2 \*\* 2 + PHEN4 \*\* 2 + BEH2 \*\* 2 + BEH4r \*\* 2).

EXECUTE.

COMPUTE PSNum=PScrossProd - (8 \* PSavgA \* PSavgB).

EXECUTE.

COMPUTE PSD1=PSsumAsq - (8 \* (PSavgA \*\* 2)).

EXECUTE.

COMPUTE PSD2=PSsumBsq - (8 \* (PSavgB \*\* 2)).

EXECUTE.

COMPUTE PSDen=SQRT(PSD1 \* PSD2).

EXECUTE.

COMPUTE EOI1=PSNum / PSDen.

EXECUTE.

ADD FILES FILE=\* /DROP= PScrossProd, PSavgA, PSsumAsq, PSavgB, PSsumBsq, PSNum, PSD1, PSD2, PSDen.

EXECUTE.

\*\*adjusts EOI score using Spearman-Brown Prediction Formula and recoded values less than -1 to -1

COMPUTE EOI2=2 \* EOI1 / (1 + (2 - 1) \* EOI1).

EXECUTE.

RECODE EOI2 (Lowest thru -1=-1) (ELSE=Copy) INTO EOI.

EXECUTE.

ADD FILES FILE=\* /DROP= EOI1, EOI2.

EXECUTE.

\*\*calculates STDEV

COMPUTE STDEV=VARIANCE(SPEC1, SPEC2, SPEC3, SPEC4, GEN1, GEN2r, GEN3, GEN4, PHEN1, PHEN2, PHEN3, PHEN4, BEH1, BEH2, BEH3, BEH4r).

EXECUTE.

\*\*calculates SD (social desirability)

RECODE SD1 SD2 SD3 SD4 SD5 SD6 SD7 SD8 SD9 SD10 SD11 SD12 SD13 SD14 SD15 SD16 SD17 SD18 SD19 SD20

SD21 SD22 SD23 SD24 SD25 SD26 SD27 SD28 SD29 SD30 SD31 SD32 SD33 (1=1) (2=0).

EXECUTE.

COMPUTE SD=SD1 + SD2 + SD3 + SD4 + SD5 + SD6 + SD7 + SD8 + SD9 + SD10 + SD11 + SD12 + SD13 + SD14 +

SD15 + SD16 + SD17 + SD18 + SD19 + SD20 + SD21 + SD22 + SD23 + SD24 + SD25 + SD26 + SD27 + SD28 +

SD29 + SD30 + SD31 + SD32 + SD33.

EXECUTE.

\*\*calculates MDPT

COUNT VALUE3tally= SPEC1, SPEC2, SPEC3, SPEC4, GEN1, GEN2, GEN3, GEN4, PHEN1, PHEN2, PHEN3, PHEN4, BEH1, BEH2, BEH3, BEH4 (3).

COUNT VALUE4tally= SPEC1, SPEC2, SPEC3, SPEC4, GEN1, GEN2, GEN3, GEN4, PHEN1, PHEN2, PHEN3, PHEN4, BEH1, BEH2, BEH3, BEH4 (4).

EXECUTE.

COMPUTE MIDPT= VALUE3tally + VALUE4tally.

EXECUTE.

ADD FILES FILE=\* /DROP= VALUE3tally,VALUE4tally.

EXECUTE.

\*\*calculates EXT

COUNT VALUE1tally= SPEC1, SPEC2, SPEC3, SPEC4, GEN1, GEN2, GEN3, GEN4, PHEN1, PHEN2, PHEN3, PHEN4, BEH1, BEH2, BEH3, BEH4 (1).

COUNT VALUE6tally= SPEC1, SPEC2, SPEC3, SPEC4, GEN1, GEN2, GEN3, GEN4, PHEN1, PHEN2, PHEN3, PHEN4, BEH1, BEH2, BEH3, BEH4 (6).

EXECUTE.

COMPUTE EXTREME= VALUE1tally + VALUE6tally.

EXECUTE.

ADD FILES FILE=\* /DROP= VALUE1tally,VALUE6tally.

EXECUTE.

\*\*calculates ACQU

COMPUTE ACQU=(SPEC1 + SPEC2 + SPEC3 + SPEC4 + GEN1 + GEN2 + GEN3 + GEN4 + PHEN1 + PHEN2 + PHEN3 +

PHEN4 + BEH1 + BEH2 + BEH3 + BEH4) / 16.

EXECUTE.

\*\*dichotomizes RACE

RECODE RACE (1=1) (5=2) (2=SYSMIS) INTO AW.

EXECUTE.

RECODE RACE (1=SYSMIS) (5=2) (2=1) INTO BW.

EXECUTE.

**PERSON FIT**

[computation for person fit statistics conducted in R studio (guttman errors, guttman errors normed, and standardized log likelihood variables are already included in SPSS datasets 1- 6)]

RECODE SPEC1 SPEC2 SPEC3 SPEC4 GEN1 GEN2 GEN3 GEN4 PHEN1 PHEN2 PHEN3 PHEN4 BEH1 BEH2 BEH3 BEH4

(1=0) (2=1) (3=2) (4=3) (5=4) (6=5).

EXECUTE.

\*\*save as csv with no headings and import to R

\*\*human datafile

human\_matrix<- as.matrix(human)

sink("output.human.csv")

Gpoly(human\_matrix, Ncat=6,NA.method = "Pairwise", Save.MatImp = FALSE,IP = NULL, IRT.PModel = "GRM", Ability = NULL, Ability.PModel = "EAP")

lzpoly(human\_matrix, Ncat=6,NA.method = "Pairwise", Save.MatImp = FALSE,IP = NULL, IRT.PModel = "GRM", Ability = NULL, Ability.PModel = "EAP")

sink()

**NORMALIZE METRICS**

DESCRIPTIVES VARIABLES=MD REI LS PAI PSI STDEV EOI SD MIDPT EXTREME ACQU GUTTN LZ

/STATISTICS=MEAN STDDEV MIN MAX KURTOSIS SKEWNESS.

COMPUTE MD\_log=LG10(MD).

COMPUTE LS\_log=LG10(LS).

EXECUTE.

FREQUENCIES VARIABLES=MD\_log LS\_log

/FORMAT=NOTABLE

/STATISTICS=SKEWNESS SESKEW KURTOSIS SEKURT

/HISTOGRAM NORMAL

/ORDER=ANALYSIS.

**CORRELATION**

CORRELATIONS

/VARIABLES=MD PSI PAI EOI LS REI GUTTN LZ STDEV MIDPT ACQU EXTREME SD

/PRINT=TWOTAIL NOSIG

/MISSING=PAIRWISE.

**MULTIVARIATE ANALYSIS**

GLM MD REI LS PAI PSI STDEV EOI GUTTN LZ BY RACE

/METHOD=SSTYPE(3)

/INTERCEPT=INCLUDE

/PRINT = ETASQ

/POSTHOC=RACE(TUKEY BTUKEY)

/CRITERIA=ALPHA(.05)

/DESIGN=RACE.

GLM MIDPT EXTREME ACQU SD BY RACE

/METHOD=SSTYPE(3)

/INTERCEPT=INCLUDE

/PRINT = ETASQ

/POSTHOC=RACE(TUKEY BTUKEY)

/CRITERIA=ALPHA(.05)

/DESIGN=RACE.

ONEWAY MD REI LS PAI PSI STDEV EOI GUTTN LZ BY RACE

/STATISTICS DESCRIPTIVES

/MISSING ANALYSIS

/POSTHOC=BTUKEY ALPHA(0.05).

ONEWAY MIDPT EXTREME ACQU SD BY RACE

/STATISTICS DESCRIPTIVES

/MISSING ANALYSIS

/POSTHOC=BTUKEY ALPHA(0.05).

GLM MD REI LS PAI PSI STDEV EOI GUTTN LZ BY IMMIG

/METHOD=SSTYPE(3)

/INTERCEPT=INCLUDE

/PRINT = ETASQ

/POSTHOC=IMMIG(TUKEY BTUKEY)

/CRITERIA=ALPHA(.05)

/DESIGN=IMMIG.

GLM MIDPT EXTREME ACQU SD BY IMMIG

/METHOD=SSTYPE(3)

/INTERCEPT=INCLUDE

/PRINT = ETASQ

/POSTHOC=IMMIG(TUKEY BTUKEY)

/CRITERIA=ALPHA(.05)

/DESIGN=IMMIG.

GLM MD REI LS PAI PSI STDEV EOI GUTTN LZ BY GENDER\_BINARY

/METHOD=SSTYPE(3)

/INTERCEPT=INCLUDE

/PRINT = ETASQ

/POSTHOC= GENDER\_BINARY (TUKEY BTUKEY)

/CRITERIA=ALPHA(.05)

/DESIGN= GENDER\_BINARY.

GLM MIDPT EXTREME ACQU SD BY GENDER\_BINARY

/METHOD=SSTYPE(3)

/INTERCEPT=INCLUDE

/PRINT = ETASQ

/POSTHOC= GENDER\_BINARY (TUKEY BTUKEY)

/CRITERIA=ALPHA(.05)

/DESIGN= GENDER\_BINARY.

CORRELATIONS

/VARIABLES=MD REI LS PAI PSI STDEV EOI SD GUTTN LZ MIDPT EXTREME ACQU AGE

/PRINT=TWOTAIL NOSIG

/MISSING=PAIRWISE.

**FACTOR ANALYSIS**

FACTOR

/VARIABLES REI PAI PSI STDEV EOI SD MIDPT EXTREME ACQU MD\_log LS\_log LZ GUTTN

/MISSING LISTWISE

/ANALYSIS REI PAI PSI STDEV EOI SD MIDPT EXTREME ACQU MD\_log LS\_log LZ GUTTN

/PRINT INITIAL KMO EXTRACTION ROTATION

/FORMAT SORT

/CRITERIA MINEIGEN(1) ITERATE(25)

/EXTRACTION ML

/CRITERIA ITERATE(25) DELTA(0)

/ROTATION OBLIMIN.

**CFA**

[conducted in R studio]

one <- 'A =~ EXTREME + MIDPT + STDEV + MD\_log + REI + LS\_log + PSI + EOI + ACQU + SD + PAI + GUTTN + LZ

'

fit <- cfa (one, data = RE\_DATA6\_OSF\_rev, std.lv = TRUE, orthogonal = FALSE)

summary(fit, standardized = TRUE, fit.measures=TRUE, rsq = TRUE)

modindices(fit, sort. = TRUE)

semPaths(fit, whatLabels = "std", layout = "tree")

four <- 'A =~ EXTREME + MIDPT + STDEV

B =~ REI + LS

C =~ GUTTN + LZ + MD

D =~ ACQU + PSI + PAI + EOI + SD

'

fit <- cfa (four, data = RE\_DATA6\_OSF\_rev, std.lv = TRUE, orthogonal = FALSE)

summary(fit, standardized = TRUE, fit.measures=TRUE, rsq = TRUE)

modindices(fit, sort. = TRUE)

semPaths(fit, whatLabels = "std", layout = "tree")

**MEDIATION MODELS**

DESCRIPTIVES VARIABLES=MIDPT ACQU EXTREME SD MD REI LS PAI PSI STDEV EOI LZ GUTTN

/SAVE

/STATISTICS=MEAN STDDEV RANGE MIN MAX KURTOSIS SKEWNESS.

[requires PROCESS v3 macro]

**STDEV**

process vars =

/y = ZSTDEV

/m = ZMIDPT ZEXTREME ZACQU ZSD

/x = BW

/ model = 4

/mcw = 1

/plot = 1

process vars =

/y = ZSTDEV

/m = ZMIDPT ZEXTREME ZACQU ZSD

/x = AW

/ model = 4

/mcw = 1

/plot = 1

**MD**

process vars =

/y = ZMD

/m = ZMIDPT ZEXTREME ZACQU ZSD

/x = BW

/ model = 4

/mcw = 1

/plot = 1

process vars =

/y = ZMD

/m = ZMIDPT ZEXTREME ZACQU ZSD

/x = AW

/ model = 4

/mcw = 1

/plot = 1

**PS**

process vars =

/y = ZPSI

/m = ZMIDPT ZEXTREME ZACQU ZSD

/x = BW

/ model = 4

/mcw = 1

/plot = 1

process vars =

/y = ZPSI

/m = ZMIDPT ZEXTREME ZACQU ZSD

/x = AW

/ model = 4

/mcw = 1

/plot = 1

**PAI**

process vars =

/y = ZPAI

/m = ZMIDPT ZEXTREME ZACQU ZSD

/x = BW

/ model = 4

/mcw = 1

/plot = 1

process vars =

/y = ZPAI

/m = ZMIDPT ZEXTREME ZACQU ZSD

/x = AW

/ model = 4

/mcw = 1

/plot = 1

**EOI**

process vars =

/y = ZEOI

/m = ZMIDPT ZEXTREME ZACQU ZSD

/x = BW

/ model = 4

/mcw = 1

/plot = 1

process vars =

/y = ZEOI

/m = ZMIDPT ZEXTREME ZACQU ZSD

/x = AW

/ model = 4

/mcw = 1

/plot = 1

**REI**

process vars =

/y = ZREI

/m = ZMIDPT ZEXTREME ZACQU ZSD

/x = BW

/ model = 4

/mcw = 1

/plot = 1

process vars =

/y = ZREI

/m = ZMIDPT ZEXTREME ZACQU ZSD

/x = AW

/ model = 4

/mcw = 1

/plot = 1

**GUTTN**

process vars =

/y = ZGUTTN

/m = ZMIDPT ZEXTREME ZACQU ZSD

/x = BW

/ model = 4

/mcw = 1

/plot = 1

process vars =

/y = ZGUTTN

/m = ZMIDPT ZEXTREME ZACQU ZSD

/x = AW

/ model = 4

/mcw = 1

/plot = 1

**LZ**

process vars =

/y = ZLZ

/m = ZMIDPT ZEXTREME ZACQU ZSD

/x = BW

/ model = 4

/mcw = 1

/plot = 1

process vars =

/y = ZLZ

/m = ZMIDPT ZEXTREME ZACQU ZSD

/x = AW

/ model = 4

/mcw = 1

/plot = 1