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

############### TWOWAY SIMPLE SLOPES MACRO ############################

###### Version 1.1 // 9.1.07 // by J. Ullrich #######################

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

TWOWAY y= /x= /m= /z= /rs= /alpha= /n= /high= /low= .

EXE.

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

# This is how the command works:

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

# TWOWAY y= your dependent variable here

# /x= your focal predictor here

# /m= your moderator here

# /z= standardize predictors, yes (1) or no (0)?

# /rs= calculate region of significance, yes (1) or no (0)?

# /alpha= your desired alpha level for rs (e.g., .05)

# /n= your sample size (after listwise deletion of missings)

# /high= your high conditional value of m

# /low= your low conditional value of m.

# EXE.

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

# The simplest way to use the command is to specify only the DV,

# the focal predictor variable, and the moderator variable,

# and set z to 1.

# The command will then standardize the predictor variables before

# computing the product term and use 1 SD below/above the mean as

# the conditional values of the moderator variable. There is no need to

# include more arguments (although SPSS will produce error message #4024,

# talking about omitted operands, which you may ignore).

# Note that the macro temporarily removes all missing values. However,

# the macro treats any user-defined missing values (i.e., anything other

# than the sysmis value of ',') as real data! So please check if your

# dataset involves any user-defined missing values and change them to

# ','.

# Optionally, you can set rs to 1 in order to get the limits of the

# region of significance for the focal predictor variable. In this case,

# you must provide the effective sample size (after listwise deletion of

# missing values) and the desired alpha level.

# If you wish to use other conditional values than 1 SD below/above the

# mean, set z to 0 and provide appropriate high and low conditional

# values.

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

# See notes for details. Updates may appear on this website:

# http://web.uni-frankfurt.de/fb05/psychologie/Abteil/sozial/

# Comments and suggestions to ullrich at psych.uni-frankfurt.de

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

# Run the rest of the syntax to define the macro for the current session.

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

DEFINE TWOWAY

(y = !charend('/')

/x = !charend('/')

/m = !charend('/')

/z = !charend('/')

/rs = !charend('/') !default(0)

/alpha = !charend('/')

/n = !charend('/')

/high = !charend('/') !default(1)

/low = !charend('/') !default(1)).

DO IF (!rs=1).

COMPUTE tcrit = idf.t(1-!alpha/2,!n-4).

ELSE IF(!rs=0).

COMPUTE tcrit = 0.

END IF.

MATRIX.

get dd/variables = !y !x !m tcrit /names mmnms /missing = omit.

compute n = nrow(dd).

compute df = n - 4.

compute const = make(n,1,1).

compute data=dd.

compute yy=data(:,1).

compute xx=data(:,2).

compute mm=data(:,3).

compute tcrit=data(1,4).

compute meanx = msum(xx)/nrow(xx).

compute devx = xx - meanx.

compute sqx= mssq(devx).

compute sdx = sqrt(sqx/(n-1)).

compute xz = devx/sdx.

compute meanm = msum(mm)/nrow(mm).

compute devm = mm - meanm.

compute sqm= mssq(devm).

compute sdm = sqrt(sqm/(n-1)).

compute mz = devm/sdm.

compute xname = mmnms(1,2).

compute mname = mmnms(1,3).

compute he = t({"Constant", xname, mname, "Product"}).

do if (!z = 1).

compute xm = xz&\*mz.

compute pred = {const,xz,mz,xm}.

compute b = inv(t(pred)\*pred)\*t(pred)\*yy.

compute hat = pred \* b.

compute err = cssq(hat - yy)/df.

compute vcov = err \* inv(t(pred)\*pred).

compute se = {sqrt(vcov(1,1)); sqrt(vcov(2,2)); sqrt(vcov(3,3));sqrt(vcov(4,4))}.

compute t1 = abs(b(1)/sqrt(vcov(1,1))).

compute t2 = abs(b(2)/sqrt(vcov(2,2))).

compute t3 = abs(b(3)/sqrt(vcov(3,3))).

compute t4 = abs(b(4)/sqrt(vcov(4,4))).

compute p1 = 2\*(1-TCDF(t1,df)).

compute p2 = 2\*(1-TCDF(t2,df)).

compute p3 = 2\*(1-TCDF(t3,df)).

compute p4 = 2\*(1-TCDF(t4,df)).

compute bt = {t1;t2;t3;t4}.

compute bp = {p1;p2;p3;p4}.

compute out = {b, se, bt, bp}.

print /title="+++ TWOWAY Moderated Multiple Regression Results +++".

print out /title=" "

/clabels = "b" "SE" "t" "p(>|t|)"/rnames = he /format f10.4.

print df /title="degrees of freedom (error)".

print vcov /title="Variances and covariances of predictors"/format=E12.0.

compute high=!high.

compute low=!low\*-1.

compute sslowm = b(2)+(low \* b(4)).

print /title="simple slope at low moderator value:".

compute sslowmse = sqrt(vcov(2,2) + (2\*low\*vcov(2,4)) + ((low\*\*2)\*vcov(4,4))).

compute tlow = abs(sslowm/sslowmse).

compute plow = 2\*(1-TCDF(tlow,df)).

compute outm1 = {sslowm, sslowmse,tlow, plow}.

print outm1 /title = " "/clabels= "b" "SE" "t" "p(>|t|)"/format f10.4.

compute sshim = b(2)+(high \* b(4)).

print /title="simple slope at high moderator value:".

compute sshise = sqrt(vcov(2,2) + (2\*high\*vcov(2,4)) + ((high\*\*2)\*vcov(4,4))).

compute thigh = abs(sshim/sshise).

compute phigh = 2\*(1-TCDF(thigh,df)).

compute outm2 = {sshim, sshise, thigh, phigh}.

print outm2 /title = " "/clabels= "b" "SE" "t" "p(>|t|)"/format f10.4.

do if (!rs=1).

compute rsa = (((tcrit\*\*2)\*vcov(4,4))-b(4)\*\*2).

compute rsb = (2\*(((tcrit\*\*2)\*vcov(2,4))-(b(2)\*b(4)))).

compute rsc = (((tcrit\*\*2)\*vcov(2,2))-b(2)\*\*2).

compute rs1 = (-rsb + sqrt((rsb\*\*2)-(4\*rsa\*rsc)))/(2\*rsa).

compute rs2 = (-rsb - sqrt((rsb\*\*2)-(4\*rsa\*rsc)))/(2\*rsa).

compute rs12 = {rs1;rs2}.

print /title="+++ Region of Significance Results +++".

print rs12/title="focal predictor is significant between these moderator values:".

end if.

else if (!z = 0).

compute xm = xx&\*mm.

compute pred = {const,xx,mm,xm}.

compute b = inv(t(pred)\*pred)\*t(pred)\*yy.

compute hat = pred \* b.

compute err = cssq(hat - yy)/df.

compute vcov = err \* inv(t(pred)\*pred).

compute se = {sqrt(vcov(1,1)); sqrt(vcov(2,2)); sqrt(vcov(3,3));sqrt(vcov(4,4))}.

compute out = {b, se}.

print /title="+++ TWOWAY Moderated Multiple Regression Results +++".

print out /title=" "

/clabels = "b" "SE" /rnames = he /format f10.4.

print df /title="degrees of freedom (error)".

print vcov /title="Variances and covariances of predictors"/format=E12.0.

compute high=!high.

compute low=!low.

compute sslowm = b(2)+(low \* b(4)).

print /title="simple slope at low moderator value:".

compute sslowmse = sqrt(vcov(2,2) + (2\*low\*vcov(2,4)) + ((low\*\*2)\*vcov(4,4))).

compute tlow = abs(sslowm/sslowmse).

compute plow = 2\*(1-TCDF(tlow,df)).

compute outm1 = {sslowm, sslowmse,tlow, plow}.

print outm1 /title = " "/clabels= "b" "SE" "t" "p(>|t|)"/format f10.4.

compute sshim = b(2)+(high \* b(4)).

print /title="simple slope at high moderator value:".

compute sshise = sqrt(vcov(2,2) + (2\*high\*vcov(2,4)) + ((high\*\*2)\*vcov(4,4))).

compute thigh = abs(sshim/sshise).

compute phigh = 2\*(1-TCDF(thigh,df)).

compute outm2 = {sshim, sshise, thigh, phigh}.

print outm2 /title = " "/clabels= "b" "SE" "t" "p(>|t|)"/format f10.4.

do if (!rs=1).

compute rsa = (((tcrit\*\*2)\*vcov(4,4))-b(4)\*\*2).

compute rsb = (2\*(((tcrit\*\*2)\*vcov(2,4))-(b(2)\*b(4)))).

compute rsc = (((tcrit\*\*2)\*vcov(2,2))-b(2)\*\*2).

compute rs1 = (-rsb + sqrt((rsb\*\*2)-(4\*rsa\*rsc)))/(2\*rsa).

compute rs2 = (-rsb - sqrt((rsb\*\*2)-(4\*rsa\*rsc)))/(2\*rsa).

compute rs12 = {rs1;rs2}.

print /title="+++ Region of Significance Results +++".

print rs12/title="focal predictor is significant between these moderator values:".

end if.

else.

print /title =" Wrong value specified for z. Use 1 or 0." .

end if.

END MATRIX.

delete variables tcrit.

!END DEFINE.