三軍總醫院北投分院統計及實驗設計課程之四--再談中介分析

2021/6/26

ytai1123@gmail.com

使用方法:

使用gmail帳號登入 按"執行階段" -->"全部執行" 以執行全部內容,

若要個別執行可點選每格程式左方箭頭或按 Control + Enter 鍵執行。

```
##0-1
!git  clone  https://github.com/YuehMintTai/RPython.git
```

```
    Cloning into 'RPython'...
    remote: Enumerating objects: 141, done.
    remote: Counting objects: 100% (141/141), done.
    remote: Compressing objects: 100% (139/139), done.
    remote: Total 141 (delta 71), reused 0 (delta 0), pack-reused 0
    Receiving objects: 100% (141/141), 882.40 KiB | 7.81 MiB/s, done.
    Resolving deltas: 100% (71/71), done.
```

```
##0-2
!pip  install  rpy2
```

```
    Requirement already satisfied: rpy2 in /usr/local/lib/python3.7/dist-packages (3.4.5)
    Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from rpy2) (20
    Requirement already satisfied: cffi>=1.10.0 in /usr/local/lib/python3.7/dist-packages (from r
    Requirement already satisfied: tzlocal in /usr/local/lib/python3.7/dist-packages (from rpy2)
    Requirement already satisfied: jinja2 in /usr/local/lib/python3.7/dist-packages (from rpy2) (
    Requirement already satisfied: pycparser in /usr/local/lib/python3.7/dist-packages (from cffi
    Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.7/dist-packages (fr
```

```
##0-3
%load_ext  rpy2.ipython
```

```
##0-4
%%R
myData<-read.csv('RPython/samples.csv')
tail(myData,1)
```

```
       SID 性別 年齡 入伍前職業 教育程度 婚姻狀況 皆無過去病史01 早產兒01
    188   4    1   25         商        4        1              1        0
       頭部曾受傷01 發展遲緩01 注意力不足過動症01 癲癇01 癲癇服藥治療 癲癇服藥期間
    188            0          0                  0      0            0            0
       軍種 軍階 役別 入伍至今_年 聽過自殺課程_次 求助心輔_次 求助精神科_次
```

```
188      1    1    2         0.5              1           0             2
    使用1995_次 使用24h專線_次 特殊狀況 父母婚姻狀態 自殺意念_bsrs6 B型肝炎01
188           0              0        4            4              4           0
    C型肝炎01 氣喘史01 過敏史01 心臟病史01 高血壓01 醣尿病01 甲狀腺01 類風濕01
188         0        1        1          0        0        0        1        0
    重大意外01 自殺意念01 透露父母 透露手足 透露好友 透露同儕 透露長官 透露心輔
188          1          1        0        0        0        0        0        0
    透露醫師 拒告父母 拒告手足 拒告好友 拒告同儕 拒告長官 拒告心輔 拒告醫師
188        0        1        1        1        1        1        1        1
    BSRS總分 BSRSR總分 過動症總分 Inattention Impulsivity opposition depression
188       20         5         18            9           9          8         57
    anxiety burdensome belonging 家庭滿意度apgar 網路成癮症01 網路成癮分數YDQ
188 29.0294          42        12                0             0               0
    existeness meaning control seeking death suicidea 睡眠困擾_bsrs1
188         28       10       22      16    15          7               4
    睡眠困擾_bsrsr1 睡眠困擾_bdi16 易怒_bsrs3 易怒_bsrsr3 depress impuls
188              1              3          4            1      57      9
    Internet ADHD
188        0   18
```

```
##4-1
%%R
formula='網路成癮分數YDQ~過動症總分'
model1<-glm(formula,  myData,  family='gaussian')
summary(model1)
```

```
Call:
glm(formula = formula, family = "gaussian", data = myData)

Deviance Residuals:
    Min       1Q   Median       3Q      Max  
-4.0598  -1.6493  -0.6493   1.5229   6.5062  

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.26052    0.29073   4.336 2.38e-05 ***
過動症總分   0.07776    0.01734   4.483 1.28e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 5.51421)

    Null deviance: 1136.5  on 187  degrees of freedom
Residual deviance: 1025.6  on 186  degrees of freedom
AIC: 858.49

Number of Fisher Scoring iterations: 2
```

```
##4-2
%%R
formula='網路成癮分數YDQ~depression'
model1<-glm(formula,  myData,  family='gaussian')
summary(model1)
```

```
Call:
    glm(formula = formula, family = "gaussian", data = myData)

    Deviance Residuals: 
        Min       1Q   Median       3Q      Max  
    -5.4663  -1.5947  -0.6534   1.7375   6.5584  

    Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
    (Intercept)  1.44161    0.25296   5.699 4.65e-08 ***
    depression   0.07061    0.01512   4.669 5.78e-06 ***
    ---
    Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

    (Dispersion parameter for gaussian family taken to be 5.469108)

        Null deviance: 1136.5  on 187  degrees of freedom
    Residual deviance: 1017.3  on 186  degrees of freedom
    AIC: 856.94

    Number of Fisher Scoring iterations: 2
```

```
##4-3
%%R
formula='網路成癮分數YDQ~過動症總分+depression'
model1<-glm(formula, myData, family='gaussian')
summary(model1)
```

```
    Call:
    glm(formula = formula, family = "gaussian", data = myData)

    Deviance Residuals: 
        Min       1Q   Median       3Q      Max  
    -4.9045  -1.5275  -0.7244   1.4617   6.7524  

    Coefficients:
                Estimate Std. Error t value Pr(>|t|)   
    (Intercept)  0.91543    0.30212   3.030  0.00280 **
    過動症總分   0.05535    0.01822   3.038  0.00273 **
    depression   0.05250    0.01595   3.291  0.00120 **
    ---
    Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

    (Dispersion parameter for gaussian family taken to be 5.237408)

        Null deviance: 1136.48  on 187  degrees of freedom
    Residual deviance:  968.92  on 185  degrees of freedom
    AIC: 849.79

    Number of Fisher Scoring iterations: 2
```

```
##4-4
%%R
formula='過動症總分~depression'
model1<-glm(formula, myData, family='gaussian')
summary(model1)
```

```
Call:
glm(formula = formula, family = "gaussian", data = myData)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-19.645   -7.195   -1.412    6.717   32.223

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.50573    0.99610   9.543  < 2e-16 ***
depression   0.32709    0.05955   5.493 1.29e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 84.8061)

    Null deviance: 18332  on 187  degrees of freedom
Residual deviance: 15774  on 186  degrees of freedom
AIC: 1372.3

Number of Fisher Scoring iterations: 2
```

```
##4-5
%%R
formula='depression~過動症總分'
model1<-glm(formula, myData, family='gaussian')
summary(model1)
```

```
Call:
glm(formula = formula, family = "gaussian", data = myData)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-20.653   -7.426   -3.406    6.300   42.747

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.57281    1.30222   5.047 1.06e-06 ***
過動症總分   0.42668    0.07768   5.493 1.29e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 110.6284)

    Null deviance: 23914  on 187  degrees of freedom
Residual deviance: 20577  on 186  degrees of freedom
AIC: 1422.3

Number of Fisher Scoring iterations: 2
```

```
##4-6
%%R
formula='網路成癮分數YDQ~過動症總分+年齡'
```

```
formula='網路成癮分數IDQ~過動症總分+年齡'
model1<-glm(formula,  myData,  family='gaussian')
summary(model1)
```

```
Call:
glm(formula = formula, family = "gaussian", data = myData)

Deviance Residuals:
   Min      1Q  Median      3Q     Max  
-4.087  -1.617  -0.731   1.366   6.476  

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.93655    0.87418   2.215    0.028 *  
過動症總分   0.07896    0.01742   4.533 1.04e-05 ***
年齡        -0.03245    0.03957  -0.820    0.413    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 5.523934)

    Null deviance: 1136.5  on 187  degrees of freedom
Residual deviance: 1021.9  on 185  degrees of freedom
AIC: 859.81

Number of Fisher Scoring iterations: 2
```

```
##4-7
%%R
formula='年齡~過動症總分'
model1<-glm(formula,  myData,  family='gaussian')
summary(model1)
```

```
Call:
glm(formula = formula, family = "gaussian", data = myData)

Deviance Residuals:
    Min       1Q   Median       3Q      Max  
-9.0171  -2.9885   0.0202   2.0949  19.3165  

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 20.83095    0.53917  38.635   <2e-16 ***
過動症總分   0.03707    0.03216   1.152    0.251    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 18.96507)

    Null deviance: 3552.7  on 187  degrees of freedom
Residual deviance: 3527.5  on 186  degrees of freedom
AIC: 1090.7

Number of Fisher Scoring iterations: 2
```

```
##這個package會安裝很久，約15分鐘以上
```

```
##這個package會安裝很久...約15分鐘以上...
%%R
install.packages('mediation')
```

```
R[write to console]: Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

R[write to console]: also installing the dependencies ‘png’, ‘jpeg’, ‘checkmate’,


R[write to console]: trying URL 'https://cran.rstudio.com/src/contrib/png_0.1-7.tar.gz'

R[write to console]: Content type 'application/x-gzip'
R[write to console]:  length 24990 bytes (24 KB)

R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
R[write to console]: =
##4-8
%%R
library(mediation)
set.seed(2014)
med.fit<-  glm('depression~過動症總分',myData,  family='gaussian')
out.fit<-  glm('網路成癮分數YDQ~過動症總分+depression',myData,  family='gaussian')
med.out<-mediate(med.fit,  out.fit,  treat='過動症總分',mediator='depression')
summary(med.out)
```

```
R[write to console]: Loading required package: MASS

R[write to console]: Loading required package: Matrix

R[write to console]: Loading required package: mvtnorm

R[write to console]: Loading required package: sandwich

R[write to console]: mediation: Causal Mediation Analysis
Version: 4.5.0



Causal Mediation Analysis

Quasi-Bayesian Confidence Intervals

               Estimate 95% CI Lower 95% CI Upper p-value
ACME            0.02253      0.00745         0.04   0.004 **
ADE             0.05560      0.01989         0.09   0.004 **
Total Effect    0.07814      0.04427         0.11  <2e-16 ***
Prop. Mediated  0.29020      0.09451         0.60   0.004 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Sample Size Used: 188


Simulations: 1000
```

以下為"收合區段"內容為中介分析PROCESS程式碼

請按左方三角型按鍵可展開或收合 (由於程式5,300行,不建議展開,建議直接左下方三角型按鈕執行)

```
#由於程式上千行,建議直接按左方三角行按鍵
##以下程式會執行約10分鐘....作者說:Patience  is  a  virtue.
%%R
#  PROCESS  for  R  version  3.5.3  beta0.6
#  Written  by  Andrew  F.  Hayes
#  www.afhayes.com
#  www.processmacro.org {}
#  Copyright  2020  by  Andrew  F.  Hayes  ALL  RIGHTS  RESERVED
# Copyright 2020 by Andrew F. Hayes ALL RIGHTS RESERVED
# PROCESS workshop schedule at http://www.processmacro.org/workshops.html
#
# Distribution of this code in any form except through processmacro.org
# is prohibited without the permission of the copyright holder, as is
# distribution after modification.
#
# THIS SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
# MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT
# IN NO EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM
# DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT
# OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE
# USE OF THIS SOFTWARE IMPLIES AGREEMENT WITH THESE TERMS
#
# To activate PROCESS, run this script. It will take a few minutes.
# Patience is a virtue.

process.bcboot3<-function(databcbt,estmte,xp2,badend,priorlo,priorhi)
{
  databcbt<-as.matrix(sort(databcbt))
  badlo<-0;badhi<-0
  pv<-matrix(as.numeric(databcbt < estmte));
  pv<-sum(pv)/nrow(databcbt);ppv<-pv;
  if (pv > .5){ppv<-(1-pv)}
  y5<-sqrt(-2*log(ppv))
  p0<-(-.322232431088);p1<-(-1);p2<-(-.342242088547);p3<-(-.0204231210245)
  p4<-(-.0000453642210148);q0<-(.0993484626060);q1<-(.588581570495)
  q2<-(.531103462366);q3<-(.103537752850);q4<-(.0038560700634)
  xp<-y5+((((y5*p4+p3)*y5+p2)*y5+p1)*y5+p0)/((((y5*q4+q3)*y5+q2)*y5+q1)*y5+q0)
  if (pv <= .5){xp<-(-xp)}
  cilow<-round(nrow(databcbt)*pnorm(2*xp-xp2))
  cihigh<-trunc(nrow(databcbt)*pnorm(2*xp+xp2))+1
  if (cilow < 1){cilow<-1;booterr<-1;badlo<-1}
  if (cihigh > nrow(databcbt)){cihigh<-nrow(databcbt);booterr<-1;badhi<-1}
  llcit<-databcbt[cilow,1]
  ulcit<-databcbt[cihigh,1]
  if ((badlo==1) & (llcit != priorlo)){priorlo<-llcit;badend<-c(badend,llcit)}
  if ((badhi==1) & (ulcit != priorhi)){priorhi<-ulcit;badend<-c(badend,ulcit)}
  bootse<-sd(databcbt)
  cires<-as.matrix(c(bootse,llcit,ulcit))
  cires<-list(cires,badend,priorlo,priorhi)
  return(cires)
}

process.pboot3<-function(databcbt,lcval,hcval)
{
  databcbt<-as.matrix(sort(databcbt))
  llcit<-databcbt[lcval,1]
  ulcit<-databcbt[hcval,1]
  bootse<-sd(databcbt)
  cires<-as.matrix(c(bootse,llcit,ulcit))
  return(cires)
}
```

```
process.llrtest3<-function(lm,y,x,b,basemod,iterate,converge)
{
  lm<-as.matrix(lm)
  btemphld<-b
  llrdat<-matrix(-999,nrow(x),(nrow(lm)-sum(lm)))
  llrdf<-ncol(x)-ncol(llrdat)
  llrcnt<-0
  for (llri in (1:nrow(lm)))
  {
    if (lm[llri,1]==0){llrcnt<-llrcnt+1;llrdat[,llrcnt]<-x[,llri]}
  }
  LL2<-process.modelest(y,llrdat,2,0,xp2,5,iterate,converge)
  b<-btemphld
  pvchi<-(1-pchisq((LL2-basemod),df=llrdf))
  fresult<-cbind((LL2-basemod),llrdf,pvchi)
  return(fresult)
}

process.describ3<-function(descdatf,type=0,quantle=1)
{
  desctmp<-matrix(-999,(8-(4*type)),ncol(descdatf))
  # mean, sd, min, max, 16th, 50th, 84th, dich toggle
  for (jd in c(1:ncol(descdatf)))
  {
    descdat<-descdatf[,jd]
    #get the mean, sd, minimum, and maximum */
    desctmp[1,jd]<-mean(descdat)
    desctmp[2,jd]<-sd(descdat)
    desctmp[3,jd]<-min(descdat)
    desctmp[4,jd]<-max(descdat)
    if (type==0)
    {
      minwarn<-0;maxwarn<-0
      tmp=as.numeric(descdat==desctmp[3,jd])+as.numeric(descdat==desctmp[4,jd])
      desctmp[8,jd]<-as.numeric(sum(tmp)==length(tmp))
      if (desctmp[3,jd]==desctmp[4,jd]){desctmp[8,jd]<-2}
      descdat<-matrix(sort(descdat))
      decval<-c(.16,.5,.84)
      for (kd in c(1:3))
      {
        low<-trunc(decval[kd]*(length(descdat)+1))
        lowdec<-decval[kd]*(length(descdat)+1)-low
        value<-descdat[low,1]+(descdat[(low+1),1]-descdat[low,1])*lowdec
        desctmp[(4+kd),jd]<-value
      }
      mnotev<-(1)
      modvals<-matrix(desctmp[5:7,],ncol=ncol(descdatf))
      if (quantle != 1)
      {
        desctmp[5,jd]<-desctmp[1,jd]-desctmp[2,jd]
        desctmp[6,jd]<-desctmp[1,jd]
        desctmp[7,jd]<-desctmp[1,jd]+desctmp[2,jd]
        modvals<-matrix(desctmp[5:7,],ncol=ncol(descdatf))
        mnotev<-(2)
        if (modvals[1,1] < desctmp[3,1]){modvals[1,1]<-desctmp[3,1];minwarn<-1}
        if (modvals[1,1] < desctmp[3,1]){modvals[1,1]<-desctmp[3,1];minwarn<-1}
        if (modvals[3,1] > desctmp[4,1]){modvals[3,1]<-desctmp[4,1];maxwarn<-1}
      }
      if (desctmp[8,jd]==1)
      {modvals<-matrix(c(desctmp[3,1],desctmp[4,1]))
      mnotev<-0;minwarn<-0;maxwarn<-0
      }
    }
  }

  descrtrn<-list(desctmp,modvals,minwarn,maxwarn,mnotev)
  return(descrtrn)
}

process.ftest3<-function(lm,bcoef,cv=0,chr=0,brsq=0,skip=0,y,x)
{
  lmat2<-as.matrix(lm)
  y<-as.matrix(y)
  x<-as.matrix(x)
  n<-nrow(y)
  if (skip==0)
  {
    lmat2<-as.matrix(diag(as.numeric(lm)))
    lmat3<-matrix(0,nrow(lmat2),1)
    for (flp in c(1:ncol(lmat2)))
    {
      if (sum(lmat2[,flp])==1)
      {lmat3<-cbind(as.matrix(lmat3),as.matrix(lmat2[,flp]))}
    }
    lmat2<-as.matrix(lmat3[,2:ncol(lmat3)])
  }
  fratio<-(t(t(lmat2)%*%bcoef)%*%solve(t(lmat2)%*%cv%*%lmat2)%*%((t(lmat2)%*%bcoef)))/ncol(lmat2)
  pfr<-(1-pf(fratio,ncol(lmat2),(n-nrow(bcoef))))
  fresult<-matrix(c(fratio,ncol(lmat2),(n-nrow(bcoef)),pfr),ncol=4)
  if (chr==1)
  {
    lmat3<-as.matrix(1-rowSums(lmat2))
    xfm<-matrix(0,n,sum(lmat3))
    flpc<-1
    for (flp in (1:nrow(lmat3)))
    {
      if (lmat3[flp,1]==1){xfm[,flpc]=x[,flp];flpc<-flpc+1}
    }
    bfm<-solve(t(xfm)%*%xfm)%*%t(xfm)%*%y
    resid<-y-(xfm%*%bfm)
    sstotal<-t(y-(sum(y)/n))%*%(y-(sum(y)/n))
    ssresid<-t(resid)%*%resid
    rsqch<-as.numeric(brsq)-((sstotal-ssresid)/sstotal)
    fresult<-matrix(c(rsqch,fresult),ncol=5)
  }
  ftestout<-as.matrix(fresult)
  return(ftestout)
}

#type1 for ols, type2 for logistic LLR, type3 for logistic bootstrapping
process.modelest<-function(y,x,type,full,xp2,bc=5,iterate=100,converge=.00001)
process.modelest<-function(y,x,type,full,xp2,hc=3,iterate=100,converge=.00001)
{
  if (type==1)
  {
    invxtx<-solve(t(x)%*%x)
    b<-invxtx%*%t(x)%*%y
    modres<-b
    if (full==1)
    {
      n1<-nrow(x)
      dfres<-(n1-(ncol(x)))
      sstotal<-t(y-(sum(y)/n1))%*%(y-(sum(y)/n1))
      resid=y-x%*%b
      ssresid<-sum(t(resid)%*%resid)
      r2<-(sstotal-ssresid)/sstotal
      adjr2<-(1-((1-r2)*(n1-1)/(dfres)))
      mse<-ssresid/(n1-ncol(x))
      #HC covariance matrix
      varb<-mse*invxtx
      k3<-ncol(x)
      xhc<-0
      if (hc != 5)
      {
        xhc<-x
        hat<-matrix(xhc[,1])
        for (i3 in c(1:nrow(xhc)))
        {
          xhcm<-matrix(xhc[i3,])
          hat[i3,1]<-t(xhcm)%*%invxtx%*%xhcm
        }
        if ((hc==0) | (hc==1))
        {
          for (i3 in c(1:k3)){xhc[,i3]<-xhc[,i3]*resid}
        }
        if ((hc==3) | (hc==2))
        {
          for (i3 in c(1:k3))
          {xhc[,i3]<-(resid/(1-hat)^(1/(4-hc)))*xhc[,i3]}
        }
        if (hc==4)
        {
          hcmn<-matrix(4,n1,2);hcmn[,2]<-(n1*hat)/k3
          minr<-apply(hcmn,1,FUN=min)
          for (i3 in c(1:k3))
          {
            xhc[,i3]<-(resid/(1-hat)^(minr/2))*xhc[,i3]
          }
        }
        varb<-(invxtx%*%t(xhc)%*%xhc%*%invxtx)
        if (hc==1){varb<-(n1/(n1-ncol(x)))*varb}
      }
      seb<-sqrt(diag(varb))
      trat<-b/seb
      p<-2*pt(-abs(trat),df=dfres)
      tval<-sqrt(dfres* (exp((dfres-(5/6))*((xp2/(dfres-(2/3)+(.11/dfres)))*(xp2/(dfres-(2/3)+(.11
      modres=matrix(c(modres,seb, trat,p,(b-tval*seb),(b+tval*seb)),ncol=6)
      modres<-matrix(c(modres,ses, tval,p,(b-tval*ses),(b+tval*ses)),ncol=6)
      modresl<-t(matrix(c("coeff","hclab","t","p","LLCI","ULCI")))
      lmat<-diag(ncol(x));lmat<-lmat[,2:ncol(lmat)]
      fratio<-(t(t(lmat)%*%b)%*%solve(t(lmat)%*%varb%*%lmat)%*%((t(lmat)%*%b)))/(ncol(x)-1)
      pfr<-1-pf(fratio,(ncol(x)-1),dfres)
      modsum=matrix(c(sqrt(r2),r2,mse,fratio,(ncol(x)-1),dfres,pfr))
      modsuml=matrix(c("R","R-sq","MSE","hcflab","df1","df2",  "p"))
      modretrn<-list(modres,modresl,modsum,modsuml,b,varb,tval,resid)
      return(modretrn)
    }
    if  (full==0){return(modres)}
  }
  #for  logistic  Y  model
  if  ((type==2)  |  (type==3))
  {
    xlp<-x;ylp<-as.matrix(y)
    pt2<-matrix((sum(ylp)/nrow(ylp)),nrow(ylp),1)
    if  ((type==2)|(type==3))  {LL3<-(ylp*log(pt2)+(1-ylp)*log(1-pt2))}
    LL3<-(-2*sum(LL3))
    bt1<-matrix(0,ncol(xlp),1);LL1<-0
    pt1<-matrix(0.5,nrow(ylp),1);pt1lp<-pt1
    for  (jjj  in  (1:iterate))
    {
      pt1lpc<-(pt1lp*(1-pt1lp))
      vt1<-diag(c(pt1lpc))
      b<-bt1+solve(t(xlp)%*%vt1%*%xlp)%*%t(xlp)%*%(ylp-pt1lp)
      if  ((type==2)  |  (type==3))
      {
        xlpb<-xlp%*%b
        xlpbt<-as.numeric(xlpb  >  -709.7)
        xlpb709<-((1-xlpbt)*(-709.7))
        xlpb<-((xlpb*xlpbt)+xlpb709)
        pt1lp<-1/(1+exp(-(xlpb)))
      }
      itprob<-sum((pt1lp  <  .00000001)  |  (pt1lp  >  .9999999))
      if  (itprob  >  0)
      {
        for  (kkk  in  (1:nrow(pt1lp)))
        {
          if  (pt1lp[kkk,1]  >  .9999999){pt1lp[kkk,1]<-.9999999}
          if  (pt1lp[kkk,1]  <  .00000001){pt1lp[kkk,1]<-.00000001}
        }
        itprob<-0
      }
      if  (itprob==0)
      {
        if  ((type==2)|(type==3)){LL<-(ylp*log(pt1lp)+(1-ylp)*log(1-pt1lp))}
        LL2<-(-2*sum(LL))
      }
      if  (abs(LL1-LL2)  <  converge)
      {
        if  (full==1)
        {
          pt1lpc<-(pt1lp*(1-pt1lp))
          vt1<-diag(c(pt1lpc))
          varb<-solve(t(xlp)%*%vt1%*%xlp)
```

```
          seb<-matrix(sqrt(diag(varb)))
        }
        break
      }
      bt1<-b;LL1<-LL2
    }
    modres<-b
    if (jjj > iterate)
    {
      itprob<-2
      if (booting==0){iterrmod<-1}
      if (booting==1){bootiter==1}
      if (itprobtg==0)
      {
        itprobtg<-1;errcode[errs,1]<-47;errs<-errs+1
        if ((booting==0) & (full==1))
        {
          pt1lpc<-(pt1lp*(1-pt1lp))
          vt1<-diag(pt1lpc)
          varb<-solve(t(xlp)%*%vt1%*%xlp)
          seb<-matrix(sqrt(diag(varb)))
        }
      }
    }
    if (full==1)
    {
      trat<-b/seb
      dfres<-nrow(xlp)
      p<-2*(1-pnorm(abs(trat)))
      modres<-cbind(modres,seb,trat,p)
      modres<-matrix(c(modres,(b-xp2*seb),(b+xp2*seb)),ncol=6)
      pvchi<-(1-pchisq((LL3-LL2),df=(nrow(modres)-1)))
      mcF<-(LL3-LL2)/LL3
      cox<-1-exp(-(LL3-LL2)/nrow(xlp))
      nagel<-cox/(1-exp(-(LL3)/nrow(xlp)))
      modsum<-matrix(c(LL2,(LL3-LL2),(nrow(modres)-1),pvchi, mcF,cox,nagel))
      modsuml<-matrix(c("-2LL","ModelLL", "df", "p", "McFadden", "CoxSnell", "Nagelkrk"))
      modresl<-t(matrix(c("coeff","se","Z","p","LLCI","ULCI")))
      modretrn<-list(modres,modresl,modsum,modsuml,b,varb,xp2)
      return(modretrn)
    }
    if ((full==0) & (type==2)){return(LL2)}
    if ((full==0) & (type==3)){return(modres)}
  }
}


process.dummy3<-function(dd)
{
  uq<-unique(dd)
  uq<-matrix(unlist(uq[1]))
  dummy<-matrix(0,nrow(dd),nrow(uq))
  uq<-matrix(sort(uq))
  uq2<-matrix(seq(1:nrow(uq)))
  for (i in c(1:nrow(dd)))
  {for  (j  in  c(1:nrow(uq)))
    {if  (dd[i,1]==uq[j,1])
      {dummy[i,uq2[j,1]]<-1}
    }
  }
return(dummy)
}

process.makdummy<-function(dd,method,custcov=0,custcode=999)
{
  dd<-dd[order(dd[,2]),]
  newrow<-dd[,1]
  dd<-dd[,2]
  dd<-as.data.frame(dd)
  dummy<-process.dummy3(dd)
  uq<-unique(dd);uq<-matrix(unlist(uq[1]))
  criterrd<-0;errcode5<-0;errcode4<-0;errcode6<-0
  nvls<-ncol(dummy)
  nnvls<-colSums(dummy)
  mnvls<-min(nnvls)
  nnvls<-matrix(sort(uq))
  conmat1<-1
  if  (mnvls  <  2){errcode5<-1;criterrd<-1}
  if  (nvls  >  9){errcode4<-1;criterrd<-1}
  x<-0;dummat<-0
  if  (criterrd==0)
  {
    dumok<-1
    if  (method  >  0)
    {
      x<-dummy[,2:ncol(dummy)]
      nx<-ncol(x)
      minus1<-matrix(-1,1,ncol(x))
      if  (method==4)
      {
        for  (k  in  (1:nrow(dd)))
        {shole<-sum(x[k,])
        if  (shole==0)
        {x[k,]<-minus1}}
      }
      if  ((method==2)  |  (method==3)  |  (method==5))
      {
        for  (k  in  (1:nrow(dd)))
        {shole<-sum(x[k,])
          if  (shole  >  0)
          {
            for  (i  in  (1:ncol(x)))
            {if  (x[k,i]==0){x[k,i]<-1}  else{break}
            }
          }
        }
        if  (method==3)
        {
          conmat1=c(-8,1,1,1,1,1,1,1,1,0,-7,1,1,1,1,1,1,1,0,0,-6,1,1,1,1,1,1,0,0,0,-5,1,1,1,1,1,0,0
          conmat1<-t(matrix(conmat1,9,8))
          for (i in (1:8)){conmat1[i,]=conmat1[i,]/(10-i)}
          conmat1<-t(conmat1[(10-nvls):8,(10-nvls):9])
          for (k in (1:nrow(dd)))
          {shole<-sum(x[k,]);x[k,]=conmat1[(shole+1),]}
        }
      }
      if (method==5)
      {
        if (ncol(custcode) != (nvls*(nvls-1)))
        {errcode6<-1}
        if (ncol(custcode)==(nvls*(nvls-1)))
        {
          conmat1<-matrix(0,nvls,(nvls-1));cnt<-1
          for (i in (1:nvls))
          {
            for (k in (1:(nvls-1)))
            {conmat1[i,k]<-custcode[1,cnt];cnt<-cnt+1}
          }
          for (k in (1:nrow(dd))){x[k,]<-conmat1[(sum(x[k,])+1),]}      #check this
        }
      }
      xskip<-1
      dummat<-matrix(0,(nx+1),nx)
      dummat[(2:nrow(dummat)),]<-diag(nx)
      if (method==4){dummat[1,]<-minus1}
      if (method==2)
      {for (i in 2:nrow(dummat))
        {for (j in (1:(i-1))){dummat[i,j]<-1}}
      }
      if (method==3){dummat<-conmat1}
      if ((method==5) & (errcode6==0)){dummat<-conmat1}
      dummat<-cbind(nnvls,dummat)
    }
    x<-cbind(newrow,x)
    x<-x[order(x[,1]),]
  }
  dummrn<-list(x,dummat,nvls,nnvls,errcode4,errcode5,errcode6)
  return(dummrn)
}




process<-function(data,y="xxxxx",x="xxxxx",m="xxxxx",w="xxxxx",z="xxxxx",cov="xxxxx",
            model=999,converge=.00001,iterate=100,hc=5,jn=0,effsize=0,stand=0,xmtest=0,normal=0,
            mdichok=0,contrast=999,modelbt=0,matrices=0,covmy=0,covcoeff=0,boot=5000,mc=0,
            intprobe=0.1,plot=0,total=0,save=0,mcx=0,mcw=0,mcz=0,moments=0,progress=1,
            bmatrix=-999,wmatrix=-999,zmatrix=-999,wzmatrix=-999,cmatrix=-999,xcatcode=999,
            wcatcode=999,zcatcode=999,wmodval=999,zmodval=999,center=0,conf=95,seed=-999,
            decimals=9.4,maxboot=0,modelres=0,bc=0,outscreen=1,activate=0)
{

  #all this is initiation of variables and matrices
  ranseed<-0;
  if (seed != -999)
  {seed<-trunc(abs(seed));set.seed(seed)}
```

```
wnames<-w;znames<-z;mcerpt<-0;wiscov<-0;ziscov<-0;itprobtg<-0;
v2tag<-0;maxwwarn<-0;minwwarn<-0;maxzwarn<-0;minzwarn<-0
toomany<-0;wdich<-0;zdich<-0;wnotev<-0;znotev<-0;singlr<-0
nxpval<-1;nwpval<-1;nzpval<-1;errs<-1;notes<-1;criterr<-0
novar<-0;adjust<-0;ncs<-0;serial<-0;sobelok<-0;hasw<-0;ydich<-0;
hasz<-0;printw<-0;printz<-0;counterf<-0;wmodcust<-0;zmodcust<-0
booting<-0;bootiter<-0;iterrmod<-0;model<-trunc(model)
errcode<-matrix(0,100,1);notecode<-matrix(0,100,1)
iterate<-abs(trunc(iterate));converge=abs(converge);badend<-0;booterr<-0;
itprobtg<-0;v2tag<-0;maxwwarn<-0;minwwarn<-0;maxzwarn<-0
minzwarn<-0;toomany<-0;wdich<-0;zdich<-0;wnotev<-0;znotev<-0
nws<-0;nzs<-0;nms<-0;nys<-0;nxs<-0;maxresm<-9;bc<-as.numeric(bc==1);progress<-as.numeric(progress
mcxok<-0;mcwok<-0;mczok<-0;xprod<-0;zprod<-0;wprod<-0;modcok<-0;bc<-as.numeric(bc==1)
jn<-as.numeric(jn==1);effsize<-as.numeric(effsize==1);maxboots=abs(trunc(maxboot))
normal<-as.numeric(normal==1);xmtest<-as.numeric(xmtest==1);modelres<-as.numeric(modelres==1)
stand<-as.numeric(stand==1);outscreen<-as.numeric(outscreen==1);activate=as.numeric(activate==1)
if  (stand==1)  {effsize<-1}
pstog<-0;sobelok<-0;mdichok=as.numeric(mdichok==1)
resultm<-matrix(99999,1,maxresm)

#contrast  matrix
contrast<-matrix(contrast);
contrast<-t(contrast)
ncontr<-ncol(contrast)
if  (contrast[1,1]==999)  {ncontr<-1;contrast[1,1]<-0}
if  (ncontr==1)
{
  contrast<-trunc(contrast)
  if  ((contrast[1,1]  >  2)  |  (contrast[1,1]  <  0))
  {ncontr<-1;contrast[1,1]<-0}
}
if  (ncontr  >  1)
{
  contvec<-contrast;contrast[1,1]<-3
  if  (((model==2)  |  (model==3))  &  (nms==0))
  {
    if  (ncontr==4)
    {
    contvec<-t(matrix(contvec,2,2))
    contrast[1,1]<-0;modcok<-1;wcontval<-matrix(contvec[,1]);zcontval<-matrix(contvec[,2])
    }
    if  (ncontr  !=4)
    {
      notecode[notes,1]<-19;notes<-notes+1
      contrast[1,1]<-0
    }
  }
}
contrast<-contrast[1,1]
modelbt<-as.numeric(modelbt==1);matrices<-as.numeric(matrices==1)
covcoeff<-as.numeric(covcoeff==1);covmy<-trunc(covmy)
if  ((covmy  <  0)  |  (covmy  >  2))  {covmy<-0}
boot<-abs(trunc(boot));mc=abs(trunc(mc));hc=trunc(hc)
if  ((intprobe  <  0)  |  (intprobe  >  1))
{intprobe<-0.10}
```

```
plot<-trunc(plot)
if ((plot < 0) | (plot > 2)) {plot<-0}
total<-as.numeric(total==1)
dototal<-0
saveboot<-as.numeric((save==1)|(save==3))
if ((save==2) | (save==3)){intprobe=1}
saveest<-as.numeric(save > 1)
if ((hc >=0) & (hc < 5)) {notecode[notes,1]=4;notes=notes+1}
if ((hc > 5) | (hc < 0)) {hc=5}
####dont use these when y is dichotomous
hclab<-matrix(c("se(HC0)","se(HC1)","se(HC2)","se(HC3)","se(HC4)","se"))
hclab<-hclab[(hc+1),1]
hcflab<-matrix(c("F(HC0)","F(HC1)","F(HC2)","F(HC3)","F(HC4)","F"))
hcflab<-hcflab[(hc+1),1]
mcw=trunc(mcw);mcz=trunc(mcz);mcx=trunc(mcx);
if ((mcx > 0) & (model==74)) {mcw=mcx}
if ((mcw > 0) & (model==74)) {mcx=mcw}
if ((mcx > 0) & (contrast > 0))
{notecode[notes,1]=28;notes=notes+1;contrast=0};
nxvls<-1;nmvls<-1;nwvls<-1;nzvls<-1
paths<-matrix(999);pathsw<-matrix(999);pathsz<-matrix(999);
pathswz<-matrix(999);pathsmod<-matrix(999);pathtype<-matrix(999);obscoeff<-999;
pathsdv=matrix(" ");quantile<-1;moments=as.numeric(moments==1)
if (moments==1){quantile<-0}
bmatrix=t(matrix(bmatrix));wmatrix=t(matrix(wmatrix));zmatrix=t(matrix(zmatrix));
wzmatrix=t(matrix(wzmatrix));cmatrix=t(matrix(cmatrix));xcatcode=t(matrix(xcatcode));
wcatcode=t(matrix(wcatcode));zcatcode=t(matrix(zcatcode));
needed<-0
decimals<-paste("%",decimals,"f",sep='')
if ((trunc(conf) >= 100) | (trunc(conf <= 50)))
  {conf=95;notecode[notes,1]=2;notes=notes+1}
if ((model > 0)  & (model < 4) & (modelbt==0)) {boot<-0;mc<-0;bc<-0;saveboot<-0}
if ((boot > 0) & (mc > 0)) {boot<-0;bc<-0}
p0<- -.322232431088;p1<- -1;p2<- -.342242088547;p3<- -.0204231210245;
p4<- -.0000453642210148;q0<- .0993484626060;q1<- .588581570495;
q2<- .531103462366;q3<- .103537752850;q4<- .0038560700634;
priorlo<-(-9999999);priorhi<-9999999
alpha2<-(1-(conf/100))/2;cilm<-alpha2*2;y5<-sqrt(-2*log(alpha2));
xp2=(y5+((((y5*p4+p3)*y5+p2)*y5+p1)*y5+p0)/((((y5*q4+q3)*y5+q2)*y5+q1)*y5+q0))
medlb=matrix(c("     M1  :","     M2  :","     M3  :","     M4  :","     M5  :","     M6  :
medlb2=matrix(c("(M1)","(M2)","(M3)","(M4)","(M5)","(M6)","(M7)","(M8)","(M9)","(M10)"))
xlb=matrix(c("X1  :","X2  :","X3  :","X4  :","X5  :","X6  :","X7  :","X8  :","X9  :","X10:"))
highlbw=matrix(c("M1*W","M2*W","M3*W","M4*W","M5*W","M6*W","M7*W","M8*W","M9*W","M10*W"))
highlbz=matrix(c("M1*Z","M2*Z","M3*Z","M4*Z","M5*Z","M6*Z","M7*Z","M8*Z","M9*Z","M10*Z"))
highlbwz=matrix(c("M1*W*Z","M2*W*Z","M3*W*Z","M4*W*Z","M5*W*Z","M6*W*Z","M7*W*Z","M8*W*Z","M9*W*Z
highlbx=matrix(c("M1*X","M2*X","M3*X","M4*X","M5*X","M6*X","M7*X","M8*X","M9*X","M10*X"))
highlbbt=matrix(c("BOTH(M1)","BOTH(M2)","BOTH(M3)","BOTH(M4)","BOTH(M5)","BOTH(M6)","BOTH(M7)","B
skipwz=0
xnck=matrix(x);ynck=matrix(y);wnck=matrix(w);znck=matrix(z);mnck=matrix(m);cnck=matrix(cov)
if ((nrow(xnck)>1) | (nrow(ynck)>1) | (nrow(wnck)>1) | (nrow(znck)>1))
  {errcode[errs,1]<-3;errs<-errs+1;criterr<-1}


validm=matrix(c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,1,1,0,0,0,0,
        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,0,0,0
```

```
dim(validm)<-c(1,92)

# check for major errors in syntax
if (activate==1)
{errcode[errs,1]<-60;errs<-errs+1;criterr<-1}
if (criterr==0)
{
  if ((model > 0) & (model < 93))
    {
      if (validm[1,model]==0)
      {
        if (model != 74) {errcode[errs,1]<-6;errs<-errs+1;criterr<-1}
        if (model==74) {errcode[errs,1]<-46;errs<-errs+1;criterr<-1}
      }
    }
  if (((model > 92) | (model < 0)) & (model != 999))
    {errcode[errs,1]<-7;errs<-errs+1;criterr<-1}
  if ((model==999) & (bmatrix[1,1]==-999))
    {errcode[errs,1]<-24;errs<-errs+1;criterr<-1}
  if ((model != 999) & (bmatrix[1,1] != -999))
    {errcode[errs,1]<-25;errs<-errs+1;criterr<-1}
  if (((model==74) | ((model > 0) & (model < 4))) & ((wmatrix[1,1] !=-999) | (zmatr
      {errcode[errs,1]<-41;errs<-errs+1;criterr<-1}
  if ((y=="xxxxx") | (x=="xxxxx"))
      {errcode[errs,1]<-1;errs<-errs+1;criterr<-1}
  if ((mnck[1,1]=="xxxxx") & (model > 3))
      {errcode[errs,1]<-8;errs<-errs+1;criterr<-1}
}
#end of check for major errors in syntax


#A loop reads data and does some other things

if (criterr==0)
{
  #read y data
  ytmp<-data[y];nys<-ncol(ytmp);needed<-nys
  ynames<-matrix(y);n<-nrow(ytmp);varnames<-matrix(ynames)
  dat<-ytmp
  modelvar<-matrix(c(model,ynames))
  if (model==999){modelvar[1,1]="CUSTOM"}

  #read x data
  xtmp<-data[x];
  nxs<-ncol(xtmp);needed<-needed+nxs;
  dat<-cbind(dat,xtmp)
  xnames<-matrix(x);n<-nrow(xtmp);xcatlab<-xnames;
  varnames<-matrix(c(varnames,xnames))
  modelvar<-matrix(c(modelvar,xnames))
  if (nxs==1){modelvlb<-matrix(c("Model :","    Y :","    X :"))}

  #read m data
  if (mnck[1,1] != "xxxxx")
  {
    mtmp<-data[m]
    nms<-ncol(mtmp)
    mnames<-matrix(m)
    mprod<-matrix(0,1,nms)
    n<-nrow(mtmp)
    needed<-needed+nms
    varnames<-matrix(c(varnames,mnames))
    dat<-cbind(dat,mtmp)
    modelvar<-matrix(c(modelvar,mnames))
    x2m<-matrix(0,99,nms)
    x2m<-matrix(0,99,nms)
    x2y<-matrix(0,99,nms)
    onem<-matrix(1,nms,1)
    if ((nms > 1) & (nms < 11))
    {modelvlb<-matrix(c(modelvlb,medlb[1:nms,1]))}
    if (nms==1)
    {modelvlb<-matrix(c(modelvlb,"        M  :"))}
    if ((nms > 0) & (model < 4)){errcode[errs,1]<-9;errs<-errs+1;criterr<-1}
  }

  #read w data
  wlocatet<-0;wlocate<-0
  if (w != "xxxxx")
  {
    wtmp<-data[w];nws=ncol(wtmp);n<-nrow(wtmp)
    dat<-cbind(dat,wtmp)
    wnames<-matrix(w);wcatlab<-wnames;
    varnames<-matrix(c(varnames,wnames))
    modelvar<-matrix(c(modelvar,wnames))
    modelvlb<-matrix(c(modelvlb,"        W  :"))
    wlocate<-nrow(varnames)
    if (model==74)
    {
      wlocatet<-1;
      if (xnames != wnames)
      {errcode[errs,1]<-45;errs<-errs+1;criterr<-1}
    }
  }

  #read z data
  if (z != "xxxxx")
  {
    ztmp<-data[z];nzs<-ncol(ztmp);n<-nrow(ztmp);
    dat<-cbind(dat,ztmp)
    znames<-matrix(z);zcatlab<-znames;
    varnames<-matrix(c(varnames,znames))
    modelvar<-matrix(c(modelvar,znames))
    modelvlb<-matrix(c(modelvlb,"        Z  :"))
  }

  #read cov data
  if (cnck[1,1] != "xxxxx")
  {
    ctmp<-data[cov];ncs<-ncol(ctmp);n<-nrow(ctmp);
    dat<-cbind(dat,ctmp)
    covnames<-matrix(cov);varnames<-matrix(c(varnames,covnames));
```

```
}

if (((model==80) | (model==81)) & ((nms < 3) | (nms > 6)))
{errcode[errs,1]<-32;errs<-errs+1;criterr<-1}
if ((model==82) & (nms != 4))
{errcode[errs,1]<-33;errs<-errs+1;criterr<-1}
if (nms > 10)
{errcode[errs,1]<-37;errs<-errs+1;criterr<-1}
if (((model==6) | ((model > 82) & (model < 999))) & ((nms < 2) | (nms > 6)))
{errcode[errs,1]<-34;errs<-errs+1;criterr<-1}

#check for redundant variable names
match<-0;match2<-0;mcwzcov<-0
for (i in c(1:(nrow(varnames)-1)))
{
  for (j in c((i+1):nrow(varnames)))
  {
    if (varnames[i]==varnames[j])
    {
      if (i < (nxs+nms+nys+1)){match2<-match2+1}
      if ((wlocate==1) & (i==2) & (j==wlocate)){match2<-match2-1}
      if ((wnames==znames) & ((nws > 0) | (nzs > 0))){match<-match2+1}
      if ((i < (nrow(varnames)-ncs+1)) & (j > (ncol(varnames)-ncs)))
      {
        if ((varnames[j]==wnames) & (mcw==0))
        {match<-0;wiscov<-(j-(nrow(varnames)-ncs))}
        if ((varnames[j]==wnames) & (mcw !=0)){mcwzcov<=0}
        if ((varnames[j]==znames) & (mcz==0))
        {match<-0;ziscov<-(j-nrow(varnames)-ncs)}
        if ((varnames[j]==znames) & (mcz !=0)){mcwzcov<-0}
      }
    }
  }
}
if ((match2 > 0) | (match==1))
{errcode[errs,1]<-2;errs<-errs+1;criterr<-1}
if (mcwzcov==1)
{errcode[errs,1]<-50;errs<-errs+1;criterr<-1}

#listwise deletion
ninit<-nrow(dat);rownum<-seq(1:nrow(dat));dat<-cbind(rownum,dat)
dat<-na.omit(dat);n<-nrow(dat);nmiss=ninit-n;
rownum<-dat[,1];dat<-dat[,2:ncol(dat)]

#extract the data back into vectors or matrices
ytmp=as.data.frame(dat[,1:nys])
desctmp2<-process.describ3(ytmp,0,quantile)
desctmp<-matrix(unlist(desctmp2[1]))
ysd<-desctmp[2,1];ovsd<-ysd;ydich<-0
if (desctmp[8,1]==2){errcode[errs,1]<-15;errs<-errs+1;criterr<-1;novar<-1}
if (desctmp[8,1]==1)
{
  ydich<-1;
  if (total==1){total<-0;notecode[notes,1]<-24;notes<-notes+1}
  if (effsize==1){effsize<-0;notecode[notes,1]<-25;notes<-notes+1}
```

```
      omx<-max(ytmp);omn<-min(ytmp)
      ytmp<-matrix(as.numeric(ytmp==omx),ncol=nys)
      dat[,1:nys]<-matrix(as.numeric(dat[,1:nys]==omx),ncol=nys)
      rcd<-c(omn,omx,0,1);dim(rcd)<-c(2,2)
    }
    xtmp<-as.data.frame(dat[,(nys+1):(nys+nxs)])
    desctmp2<-process.describ3(xtmp,0,quantile)
    desctmp<-matrix(unlist(desctmp2[1]))
    xsd<-desctmp[2,1]
    if ((desctmp[8,1]==2) & (novar==0))
      {errcode[errs,1]<-15;errs<-errs+1;criterr<-1;novar<-1}
    xmodvals<-matrix(unlist(desctmp2[2]))
    nxpval<-nrow(xmodvals)
    xprobval<-as.matrix(xmodvals)
    xdich<-desctmp[8,1]
    if ((xdich==1) & (mcx > 0))
      {mcx<-0;errcode[errs,1]<-52;errs<-errs+1;criterr<-1}
    if ((model==74) & (xdich==1)){counterf==1}

    #need to do m
    if (nms > 0)
    {
      mtmp<-as.data.frame(dat[,(nys+nxs+1):(nys+nxs+nms)])
      desctmp2<-process.describ3(mtmp,0,quantile)
      desctmp<-matrix(unlist(desctmp2[1]),ncol=ncol(mtmp))
      ovsd<-matrix(c(desctmp[2,],ysd))
      zzzz<-rowSums(desctmp)
      if ((zzzz[8]>0) & (mdichok != 1))
      {errcode[errs,1]<-43;errs<-errs+1;criterr<-1}
      mmodvals<-matrix(unlist(desctmp2[2]),ncol=ncol(mtmp))
      mprobval<-mmodvals

    }


    if (nws > 0)
    {
      wtmp<-as.data.frame(dat[,(nys+nxs+nms+1):(nys+nxs+nms+nws)])
      desctmp2<-process.describ3(wtmp,0,quantile)
      desctmp<-matrix(unlist(desctmp2[1]))
      if ((desctmp[8,1]==2) & (novar==0))
      {errcode[errs,1]<-15;errs<-errs+1;criterr<-1;novar<-1}
      wmodvals<-matrix(unlist(desctmp2[2]))
      wdich<-desctmp[8,1]
      if ((wdich==1) & (mcw > 0))
      {mcw<-0;errcode[errs,1]<-52;errs<-errs+1;criterr<-1}
      wmin<-desctmp[3,1]
      wmax<-desctmp[4,1]
      minwwarn<-matrix(unlist(desctmp2[3]));maxwwarn<-matrix(unlist(desctmp2[4]))
      wnotev<-matrix(unlist(desctmp2[5]))
      wmodval<-t(matrix(wmodval))
      nwcontr<-ncol(wmodval)
      if (wmodval[1,1] != 999)
      {
        wmodvals<-matrix(wmodval[1,1]);wmodcust<-1
        if (nwcontr > 1){wmodvals<-t(wmodval)}
        minwwarn<-0;maxwwarn<-0;wnotev<-0
      }
      wprobval<-as.matrix(wmodvals)
      nwpval<-nrow(wmodvals)
    }

    if (nzs > 0)
    {
      ztmp<-as.data.frame(dat[,(nys+nxs+nms+nws+1):(nys+nxs+nms+nws+nzs)])
      desctmp2<-process.describ3(ztmp,0,quantile)
      desctmp<-matrix(unlist(desctmp2[1]))
      if ((desctmp[8,1]==2) & (novar==0))
      {errcode[errs,1]<-15;errs<-errs+1;criterr<-1;novar<-1}
      zmodvals<-matrix(unlist(desctmp2[2]))
      zdich<-desctmp[8,1]
      if ((zdich==1) & (mcz > 0))
      {mcz<-0;errcode[errs,1]<-52;errs<-errs+1;criterr<-1}
      zmin<-desctmp[3,1]
      zmax<-desctmp[4,1]
      minzwarn<-matrix(unlist(desctmp2[3]));maxzwarn<-matrix(unlist(desctmp2[4]));
      znotev<-matrix(unlist(desctmp2[5]))
      zmodval<-t(matrix(zmodval))
      nzcontr<-ncol(zmodval)
      if (zmodval[1,1] != 999)
      {
        zmodvals<-matrix(zmodval[1,1]);zmodcust<-1
        if (nzcontr > 1){zmodvals<-t(zmodval)}
        minzwarn<-0;maxzwarn<-0;znotev<-0
      }
      zprobval<-as.matrix(zmodvals)
      nzpval<-nrow(zmodvals)
    }
    if (ncs > 0)
      {ctmp<-as.data.frame(dat[,(nys+nxs+nms+nws+nzs+1):(nys+nxs+nms+nws+nzs+ncs)])}
    n<-nrow(ytmp)
    ones<-matrix(1,n,1)
    modresid<-matrix(9999,n,1)
    #creat codes for categorical variables
    if ((nws > 0) & (mcw > 0))
    {
      tmp<-cbind(rownum,wtmp[,1])
      dumtmp<-process.makdummy(tmp,mcw,2,wcatcode)
      wmodvals<-matrix(unlist(dumtmp[4]))
      nwpval<-nrow(wmodvals)
      errcode4<-unlist(dumtmp[5])
      errcode5<-unlist(dumtmp[6])
      errcode6<-unlist(dumtmp[7])
      if (errcode4==1){errcode[errs,1]<-4;errs<-errs+1;criterr<-1}
      if (errcode5==1){errcode[errs,1]<-5;errs<-errs+1;criterr<-1}
      if (errcode6==1){errcode[errs,1]<-39;errs<-errs+1;criterr<-1}
      if (criterr==0)
      {
        nvls<-unlist(dumtmp[3])
        nwvls<-nvls-1
        minwwarn<-0;maxwwarn<-0;wnotev<-0
        wtmp<-matrix(unlist(dumtmp[1]),ncol=(nwvls+1))
        wtmp<-wtmp[,2:ncol(wtmp)];wtmp<-as.data.frame(wtmp)
        wcatlab<-c("W1","W2","W3","W4","W5","W6","W7","W8","W9");wcatlab<-matrix(wcatlab)
        mcwok<-1
        dummatw<-matrix(unlist(dumtmp[2]),ncol=(nwvls+1),nrow=(nwvls+1))
        wprobval<-dummatw[,2:ncol(dummatw)]
        if (modcok==1)
        {
          wcontval<-matrix(-999,2,ncol(wprobval))
          temp<-0
          for (i in (1:2))
          {
            for (j in (1:nrow(dummatw)))
            {
              if (contvec[i,1]==dummatw[j,1])
              {wcontval[i,]<-wprobval[j,];temp<-temp+1}
            }
          }
          if (temp < 2)
          {notecode[notes,1]<-20;notes<-notes+1;modcok<-0}
        }
        if (wmodval[1,1] != 999)
        {notecode[notes,1]<-9;notes<-notes+1}
      }
    }
    if ((nzs > 0) & (mcz > 0))
    {
      tmp<-cbind(rownum,ztmp[,1])
      dumtmp<-process.makdummy(tmp,mcz,3,zcatcode)
      zmodvals<-matrix(unlist(dumtmp[4]))
      nzpval<-nrow(zmodvals)
      errcode4<-unlist(dumtmp[5])
      errcode5<-unlist(dumtmp[6])
      errcode6<-unlist(dumtmp[7])
      if (errcode4==1){errcode[errs,1]<-4;errs<-errs+1;criterr<-1}
      if (errcode5==1){errcode[errs,1]<-5;errs<-errs+1;criterr<-1}
      if (errcode6==1){errcode[errs,1]<-40;errs<-errs+1;criterr<-1}
      if (criterr==0)
      {
        nvls<-unlist(dumtmp[3])
        nzvls<-nvls-1
        minzwarn<-0;maxzwarn<-0;znotev<-0
        ztmp<-matrix(unlist(dumtmp[1]),ncol=(nzvls+1))
        ztmp<-ztmp[,2:ncol(ztmp)];ztmp<-as.data.frame(ztmp)
        zcatlab<-c("Z1","Z2","Z3","Z4","Z5","Z6","Z7","Z8","Z9");zcatlab=matrix(zcatlab)
        mczok<-1
        dummatz<-matrix(unlist(dumtmp[2]),ncol=(nzvls+1),nrow=(nzvls+1))
        zprobval<-dummatz[,2:ncol(dummatz)]
        if (modcok==1)
        {
          zcontval<-matrix(-999,2,ncol(zprobval))
          temp<-0
          for (i in (1:2))
          {
            for (j in (1:nrow(dummatz)))
```

```
          for (j in (1:nrow(dummatz)))
          {
            if (contvec[i,2]==dummatz[j,1])
            {zcontval[i,]<-zprobval[j,];temp<-temp+1}
          }
        }
        if (temp < 2)
        {notecode[notes,1]<-20;notes<-notes+1;modcok<-0}
      }
      if (zmodval[1,1] != 999)
      {notecode[notes,1]<-10;notes<-notes+1}
    }
  }
  if ((nxs > 0) & (mcx > 0))
  {
    tmp<-cbind(rownum,xtmp[,1])
    dumtmp<-process.makdummy(tmp,mcx,1,xcatcode)
    errcode4<-unlist(dumtmp[5])
    errcode5<-unlist(dumtmp[6])
    errcode6<-unlist(dumtmp[7])
    if (errcode4==1){errcode[errs,1]<-4;errs<-errs+1;criterr<-1}
    if (errcode5==1){errcode[errs,1]<-5;errs<-errs+1;criterr<-1}
    if (errcode6==1){errcode[errs,1]<-38;errs<-errs+1;criterr<-1}
    if (criterr==0)
    {
      nvls<-unlist(dumtmp[3])
      nxvls<-nvls-1
      xtmp<-matrix(unlist(dumtmp[1]),ncol=(nxvls+1))
      xtmp<-xtmp[,2:ncol(xtmp)];xtmp<-as.data.frame(xtmp)
      xcatlab<-c("X1","X2","X3","X4","X5","X6","X7","X8","X9");xcatlab=matrix(xcatlab)
      xdich<-as.numeric(nvls==2)
      mcxok<-1
      dummatx<-matrix(unlist(dumtmp[2]),ncol=(nxvls+1),nrow=(nxvls+1))
      xmodvals<-as.matrix(dummatx[,1])
      nxpval<-nrow(xmodvals)
    }
  }
  intlab<-matrix(" ",100,1)
  for (i in c(1:100)){intlab[i,1]<-paste("Int_",i,sep='')}
  bcmat<-matrix(0,needed,needed)
  wcmat<-matrix(0,needed,needed)
  zcmat<-matrix(0,needed,needed)
  wzcmat<-matrix(0,needed,needed)
  wsum<-0;zsum<-0;wzsum<-0
}
# End A loop reads data and does some other things

# Define model matrices for canned models
if ((criterr==0) & (model != 999))
{
  # X->MW, X->MZ, X->MWZ, M->YW, M->YZ, M->YWZ, X->YW, X->YZ, X->YWZ
  modelmat<-
  matrix(c(1,0,0,0,0,0,0,1,0,0,2,0,0,0,0,0,0,1,1,0,3,0,0,0,0,0,0,1,1,1,4,0,0,0,0,0,0,0,0,0,
  5,0,0,0,0,0,0,1,0,0,6,0,0,0,0,0,0,0,0,0,7,1,0,0,0,0,0,0,0,0,8,1,0,0,0,0,0,1,0,0,
  9,1,1,0,0,0,0,0,0,0,10,1,1,0,0,0,0,1,1,0,11,1,1,1,0,0,0,0,0,0,12,1,1,1,0,0,0,1,1,1,
  13,1,1,1,0,0,0,1,0,0,14,0,0,0,1,0,0,0,0,0,15,0,0,0,1,0,0,1,0,0,16,0,0,0,1,1,0,0,0,0
```

```
13,1,1,1,0,0,0,1,0,0,14,0,0,0,1,0,0,0,0,0,15,0,0,0,1,0,0,1,0,0,16,0,0,0,1,1,0,0,0,0,
17,0,0,0,1,1,0,1,1,0,18,0,0,0,1,1,1,0,0,0,19,0,0,0,1,1,1,1,1,1,20,0,0,0,1,1,1,1,0,0,
21,1,0,0,0,1,0,0,0,0,22,1,0,0,0,1,0,1,0,0,23,0,0,0,0,0,0,0,0,0,24,0,0,0,0,0,0,0,0,0,
25,0,0,0,0,0,0,0,0,0,26,0,0,0,0,0,0,0,0,0,27,0,0,0,0,0,0,0,0,0,28,1,0,0,0,1,0,0,1,0,
29,1,0,0,0,1,0,1,1,0,30,0,0,0,0,0,0,0,0,0,31,0,0,0,0,0,0,0,0,0,32,0,0,0,0,0,0,0,0,0,
33,0,0,0,0,0,0,0,0,0,34,0,0,0,0,0,0,0,0,0,35,0,0,0,0,0,0,0,0,0,36,0,0,0,0,0,0,0,0,0,
37,0,0,0,0,0,0,0,0,0,38,0,0,0,0,0,0,0,0,0,39,0,0,0,0,0,0,0,0,0,40,0,0,0,0,0,0,0,0,0,
41,0,0,0,0,0,0,0,0,0,42,0,0,0,0,0,0,0,0,0,43,0,0,0,0,0,0,0,0,0,44,0,0,0,0,0,0,0,0,0,
45,0,0,0,0,0,0,0,0,0,46,0,0,0,0,0,0,0,0,0,47,0,0,0,0,0,0,0,0,0,48,0,0,0,0,0,0,0,0,0,
49,0,0,0,0,0,0,0,0,0,50,0,0,0,0,0,0,0,0,0,51,0,0,0,0,0,0,0,0,0,52,0,0,0,0,0,0,0,0,0,
53,0,0,0,0,0,0,0,0,0,54,0,0,0,0,0,0,0,0,0,55,0,0,0,0,0,0,0,0,0,56,0,0,0,0,0,0,0,0,0,
57,0,0,0,0,0,0,0,0,0,58,1,0,0,1,0,0,0,0,0,59,1,0,0,1,0,0,1,0,0,60,1,1,0,1,0,0,0,0,0,
61,1,1,0,1,0,0,1,0,0,62,1,1,0,1,0,0,0,1,0,63,1,1,0,1,0,0,1,1,0,64,1,0,0,1,1,0,0,0,0,
65,1,0,0,1,1,0,1,0,0,66,1,0,0,1,1,0,0,1,0,67,1,0,0,1,1,0,1,1,0,68,1,1,1,1,0,0,0,0,0,
69,1,1,1,1,0,0,1,1,1,70,1,0,0,1,1,1,0,0,0,71,1,0,0,1,1,1,1,1,1,72,1,1,1,1,1,1,0,0,0,
73,1,1,1,1,1,1,1,1,1,74,0,0,0,1,0,0,0,0,0,75,1,1,0,1,1,0,0,0,0,76,1,1,0,1,1,0,1,1,0,
77,0,0,0,0,0,0,0,0,0,78,0,0,0,0,0,0,0,0,0,79,0,0,0,0,0,0,0,0,0,80,0,0,0,0,0,0,0,0,0,
81,0,0,0,0,0,0,0,0,0,82,0,0,0,0,0,0,0,0,0,83,1,0,0,0,0,0,0,0,0,84,1,0,0,0,0,0,0,0,0,
85,1,0,0,0,0,0,1,0,0,86,1,0,0,0,0,0,1,0,0,87,0,0,0,1,0,0,0,0,0,88,0,0,0,1,0,0,0,0,0,
89,0,0,0,1,0,0,1,0,0,90,0,0,0,1,0,0,1,0,0,91,0,0,0,0,0,0,0,0,0,92,1,0,0,1,0,0,1,0,0))
dim(modelmat)<-c(10,92);modelmat=t(modelmat);
tmp<-modelmat[model,2:ncol(modelmat)];tmp=t(matrix(tmp))
if (model < 4) {bcmat[(nxs+1),1]<-1}
if ((model > 3) & (model != 6))
  {bcmat[(nxs+1):(nxs+nms),1]<-onem
    bcmat[nrow(bcmat),(nxs+1):(nxs+nms)]<-t(onem)
    bcmat[nrow(bcmat),1]<-1}
if ((model==6) | ((model > 82) & (model < 93)))
{
  for (j in c(2:nrow(bcmat)))
  {for (i in c(1:(j-1)))
      {bcmat[j,i]<-1}
  }
}
if (model==80)
{for (i in c(1:nms)){bcmat[(nrow(bcmat)-1),i]<-1}}
if (model==81)
{for (j in c(3:nrow(bcmat))){bcmat[j,2]<-1}}
if (model==82)
{bcmat[3,2]<-1;bcmat[5,4]<-1}
if (tmp[1,1]==1)
{
  wcmat[(nxs+1):(nxs+nms),1]=onem;wprod<-1;xprod<-1
  if ((model==83) | (model==86))
  {onemsx<-onem
    for (i in c(1:(nms-1))){onemsx[(i+1),1]<-0}
    wcmat[(nxs+1):(nxs+nms),1]<-onemsx
  }
}
if (tmp[1,4]==1)
{
  wcmat[nrow(wcmat),(nxs+1):(nxs+nms)]<-t(onem);wprod<-1;
  if ((model==87) | (model==90))
  {onemsx<-onem
    for (i in c(1:(nms-1))){onemsx[i,1]=0}
    wcmat[nrow(wcmat),(nxs+1):(nxs+nms)]=t(onemsx)
```

```
   wcmat[nrow(wcmat),(nxs+1):(nxs+nms)]<-t(onemsx)
  }
 }
 if (tmp[1,7]==1){wcmat[nrow(wcmat),1]<-1;wprod<-1;xprod<-1}
 if (tmp[1,2]==1){zcmat[(nxs+1):(nxs+nms),1]<-onem;zprod<-1;xprod<-1}
 if (tmp[1,5]==1){zcmat[nrow(zcmat),(nxs+1):(nxs+nms)]=t(onem);zprod<-1}
 if (tmp[1,8]==1){zcmat[nrow(zcmat),1]<-1;zprod<-1;xprod<-1}
 if (tmp[1,3]==1){wzcmat[(nxs+1):(nxs+nms),1]<-onem;xprod<-1;wprod<-1;zprod<-1}
 if (tmp[1,6]==1){wzcmat[nrow(wzcmat),(nxs+1):(nxs+nms)]<-t(onem)}
 if (tmp[1,9]==1){wzcmat[nrow(wzcmat),1]<-1;xprod<-1;wprod<-1;zprod<-1}
 if ((model==91) | (model==92))
 {
  for (j in c(1:(nms-1)))
  {for (i in c(1:j)){wcmat[(nxs+1+j),(nxs+i)]<-1}
 }
}
}
#define and check covariates matrix
if (ncs > 0)
 {
  ccmat<-matrix(1,(nms+nys),ncs);ccmatoff<-ccmat
  if (covmy==1){ccmat[nrow(ccmat),]<-matrix(0,1,ncs)}
  if (covmy==2){ccmat[1:nms,]<-matrix(0,nms,ncs)}
  if (cmatrix[1,1] != -999)
   {if (ncol(cmatrix) != ((nms+nys)*ncs)){errcode[errs,1]<-29;errs<-errs+1;criterr<-1}
   if (criterr==0)
   {
     tmp<-1;for (i in c(1:(nms+nys)))
      {
      for (j in c(1:ncs))
      {ccmat[i,j]<-(1-as.numeric(cmatrix[1,tmp]==0));tmp<-tmp+1}
      }
     tmpcov<-colSums(ccmat);tmpcov<-as.numeric(tmpcov==0);tmpcov<-sum(tmpcov)
   if (tmpcov !=0)
   {errcode[errs,1]<-30;errs<-errs+1;criterr<-1}
   }
  if (covmy != 0){notecode[notes,1]<-1;notes<-notes+1}
 }
}

#Define matrices for custom models and do some error checking
if (criterr==0)
{
 needed<-needed*(needed-1)/2;nopath<-0
 if (bmatrix[1,1] != -999)
 {tmp<-1
  if ((ncol(bmatrix) != needed) | (sum(bmatrix)==0))
   {errcode[errs,1]<-16;errs<-errs+1;criterr<-1} else {
   for (i in c(2:nrow(bcmat)))
   {for (j in c(1:(i-1)))
    {bcmat[i,j]<-(1-as.numeric(bmatrix[1,tmp]==0));tmp<-tmp+1}
   }
  }
  #check to make sure X affects something */
  tmpcov<-sum(bcmat[,1])
  if ((tmpcov==0) & (criterr==0))
```

```
    if ((tmpcov==0) & (criterr==0))
    {errcode[errs,1]<-22;errs<-errs+1;criterr<-1}
    #check to make sure Y is affected by something */
    tmpcov<-sum(bcmat[nrow(bcmat),])
    if ((tmpcov==0) & (criterr==0))
    {errcode[errs,1]<-23;errs<-errs+1;criterr<-1}
    #check for dangling mediators
    dm<-0
    if (nms > 0)
    {
      for (i in c(1:nms))
      tmpcov<-sum(bcmat[(nxs+i),]);tmpcov2<-sum(bcmat[,(nxs+i)])
      {if (((tmpcov==0) | (tmpcov2==0)) & (dm==0) & (criterr==0))
        {errcode[errs,1]<-26;errs<-errs+1;criterr<-1;dm<-1}
      }
    }
  }
}

#start b
if (criterr==0)
{
  if (wmatrix[1,1] != -999)
  {
    tmp<-1
    if (ncol(wmatrix) != needed)
      {errcode[errs,1]<-17;errs<-errs+1;criterr<-1} else {
        modelvar[1,1]="CUSTOM"
        for (i in (2:nrow(wcmat)))
        {
          for (j in (1:(i-1)))
          {wcmat[i,j]<-(1-as.numeric(wmatrix[1,tmp]==0))
            #dont allow to specify moderation of a path that doesnt exist
            if ((wcmat[i,j]==1) & (bcmat[i,j]==0) & (nopath==0))
            {errcode[errs,1]<-20;errs<-errs+1;criterr<-1;nopath<-1}
            tmp<-tmp+1
          }
        }
      }
  }

  if (zmatrix[1,1] != -999)
  {
    tmp<-1
    if (ncol(zmatrix) != needed)
      {errcode[errs,1]<-18;errs<-errs+1;criterr<-1} else {
        modelvar[1,1]="CUSTOM"
        if ((sum(wcmat)==0) & (model==999))
        {errcode[errs,1]<-21;errs<-errs+1;criterr<-1}
        for (i in (2:nrow(zcmat)))
        {
          for (j in (1:(i-1)))
          {zcmat[i,j]<-(1-as.numeric(zmatrix[1,tmp]==0))
            #dont allow to specify moderation of a path that doesnt exist
            if ((zcmat[i,j]==1) & (bcmat[i,j]==0) & (nopath==0))
            {errcode[errs,1]<-20;errs<-errs+1;criterr<-1;nopath<-1}
            {errcode[errs,1]<-20;errs<-errs+1;criterr<-1;nopath<-1}
            tmp<-tmp+1
          }
        }
      }
  }
  tmp<-1
  if (wzmatrix[1,1] != -999)
  {
    if (ncol(wzmatrix) != needed)
    {errcode[errs,1]<-19;errs<-errs+1;criterr<-1}
    modelvar[1,1]="CUSTOM"
  }
  if (criterr==0)
  {
    for (i in (2:nrow(wzcmat)))
    {
      for (j in (1:(i-1)))
      {
        #set corresponding elements in W and Z for three way interaction
        if (wzmatrix[1,1] != -999){wzcmat[i,j]<-(1-as.numeric(wzmatrix[1,tmp]==0))}
        if (wzcmat[i,j]==1){wcmat[i,j]<-1;zcmat[i,j]<-1}
        #dont allow to specify moderation of a path that doesnt exist
        if ((wzcmat[i,j]==1) & (bcmat[i,j]==0) & (nopath==0))
        {errcode[errs,1]<-20;errs<-errs+1;criterr<-1;nopath<-1}
        tmp<-tmp+1
      }
    }
  }
}
# end B

if (criterr==0)
{
  xprod<-(sum(wcmat[,1])+sum(zcmat[,1])+sum(wzcmat[,1]))
  xprod<-as.numeric(xprod > 0)
  wsum<-sum(wcmat)
  wprod<-as.numeric(wsum > 0)
  if (nms > 0)
  {
    for (i in c(1:nms))
    {
      tmp<-(sum(wcmat[,(1+i)])+sum(zcmat[,(1+i)])+sum(wzcmat[,(1+i)]))
      mprod[1,i]<-as.numeric(tmp>0)
    }
  }
  if ((wsum > 0) & (w=="xxxxx")){errcode[errs,1]<-11;errs<-errs+1;criterr<-1}
  if ((wsum==0) & (w != "xxxxx")){errcode[errs,1]<-10;errs<-errs+1;criterr<-1}
  zsum<-sum(zcmat)
  zprod<-as.numeric(zsum > 0)
  if ((zsum > 0) & (z=="xxxxx")){errcode[errs,1]<-13;errs<-errs+1;criterr<-1}
  if ((zsum==0) & (z != "xxxxx")){errcode[errs,1]<-12;errs<-errs+1;criterr<-1}
  if ((zsum > 0) & (wsum==0)){errcode[errs,1]<-35;errs<-errs+1;criterr<-1}
}
if ((criterr==0) & (nms > 1))
{
```

```
{
 serchk<-bcmat[2:(nrow(bcmat)-1),2:ncol(bcmat)]
 if (sum(serchk) > 0)
 {serial<-1
  if (nms > 6){errcode[errs,1]<-36;errs<-errs+1;criterr<-1}
 }
}
#mean center if needed
if ((center > 0) & (criterr==0))
{
 centvar<-matrix(c(" "))
 if (criterr==0)
 {
   if ((center==1) | ((center==2) & (wdich==0)))
   {
    if ((wprod==1) & (mcwok==0) & (nwpval > 0))
    {
     for (i in c(1:nws))
     {
      wtmp[,i]<-wtmp[,i]-(sum(wtmp[,i])/n)
      centvar<-cbind(centvar,wnames[1,i])
     }
     desctmp2<-process.describ3(wtmp,wmodcust,quantile)
     desctmp<-matrix(unlist(desctmp2[1]))
     wmin<-desctmp[3,1];wmax<-desctmp[4,1]
     modvals<-matrix(unlist(desctmp2[2]))
     if (wmodcust==0){wmodvals<-modvals;wprobval<-wmodvals}
    }
   }
   if ((center==1) | ((center==2) & (zdich==0)))
   {
    if ((zprod==1) & (mczok==0) & (nzpval > 0))
    {
     for (i in c(1:nzs))
     {
      ztmp[,i]<-ztmp[,i]-(sum(ztmp[,i])/n)
      centvar<-cbind(centvar,znames[1,i])
     }
     desctmp2<-process.describ3(ztmp,zmodcust,quantile)
     desctmp<-matrix(unlist(desctmp2[1]))
     zmin<-desctmp[3,1];zmax<-desctmp[4,1]
     modvals<-matrix(unlist(desctmp2[2]))
     if (zmodcust==0){zmodvals<-modvals;zprobval<-zmodvals}
    }
   }
   if ((center==1) | ((center==2) & (xdich==0)))
   {
    if ((xprod==1) & (mcxok==0))
    {
     for (i in c(1:nxs))
     {
      xtmp[,i]<-xtmp[,i]-(sum(xtmp[,i])/n)
      centvar<-cbind(centvar,xnames[1,i])
     }
     desctmp2<-process.describ3(xtmp,0,quantile)
     modvals<-matrix(unlist(desctmp2[2]))
     modvals<-matrix(unlist(desctmp2[2]))
     xmodvals<-modvals;xprobval<-as.matrix(xmodvals)
    }
   }
   if (nms > 0)
   {
    for (i in c(1:nms))
    {
      if (mprod[1,i]==1)
      {
       mtmp[,i]<-mtmp[,i]-(sum(mtmp[,i])/n)
       centvar<-cbind(centvar,mnames[i,1])
      }
    }
    desctmp2<-process.describ3(mtmp,0,quantile)
    modvals<-matrix(unlist(desctmp2[2]),ncol=ncol(mtmp))
    mmodvals<-modvals;mprobval<-mmodvals;
   }
 }
 if (ncol(centvar) > 1)
 {notecode[notes,1]<-3;notes<-notes+1}
}
# Start D
#CONSTRUCT THE DATA MATRICES FOR EACH OF THE MODELS
if (criterr==0)
{
 # The i loop is the dependent variable in the model matrices
 # The j loop is the predictor variables in the model matrices
 wsum<-sum(wcmat);zsum<-sum(zcmat);wzsum<-sum(wzcmat)
 nump<-matrix(-999,1,(nys+nms));numint<-matrix(0,1,(nys+nms))
 #DV
 datcount<-1;
 xtmpuse<-0;wtmpuse<-0;ztmpuse<-0;xwtmpus<-0;xztmpus<-0;wztmpus<-0;xwztmpu<-0;
 xtmploc<- matrix(-999);wtmploc<- -999;xwtmplo<- -999;ztmploc<- -999
 xztmplo<- -999;wztmplo<- -999;xwztmplo<- -999
 vlabs<-″ ″
 if (ncs > 0){ctmpuse<-matrix(0,1,ncs)}
 if (nms > 0)
 {
  mtmpuse<-matrix(0,1,nms)
  mwtmpus<-matrix(0,1,nms)
  mztmpus<-matrix(0,1,nms)
  mwztmpu<-matrix(0,1,nms)
  mtmploc<-matrix(0,1,nms)
  mwtmplo<-matrix(-999,nwvls,nms)
  mztmplo<-matrix(-999,nzvls,nms)
  mwztmplo<-matrix(-999,(nwvls*nzvls),nms)
 }
 if (ncs > 0){ctmploc<-matrix(0,1,ncs)}
 fulldat<-matrix(1,n,1)
 datindx<-matrix(-999,1000,(nms+nys))
 wherew<-matrix(-999,2,(nms+nys));wherex<-matrix(-999,2,(nms+nys))
 wherez<-matrix(-999,2,(nms+nys));wherexw<-matrix(-999,2,(nms+nys))
 wherexz<-matrix(-999,2,(nms+nys));wherewz<-matrix(-999,2,(nms+nys))
 wherexwz<-matrix(-999,2,(nms+nys))
 if (nms > 0)
```

```
  if (nms > 0)
  {
      wherem<-matrix(-999,nms,(nms+nys))
      wheremw<-matrix(-999,(nms*2),(nms+nys))
      wheremz<-matrix(-999,(nms*2),(nms+nys))
      wheremwz<-matrix(-999,(nms*2),(nms+nys))
  }
  wzhigh<-matrix(0,1000,(((nms+1)*(nms+2))/2))
  whigh<-matrix(0,1000,(((nms+1)*(nms+2))/2))
  zhigh<-matrix(0,1000,(((nms+1)*(nms+2))/2))
  fochigh<-matrix(0,1000,(((nms+1)*(nms+2))/2))
  xcoefloc<-matrix(c(1,2,3,4,5,6,7,8,9))
  intkey<-t(matrix(c(" ", " ", " ", " ", " ", " ", " ")))
  wzhighct<-0;whighct<-0;zhighct<-0;foccnt<-0
  if (nms > 0){mnames<-t(mnames)}
  if (ncs > 0){covnames<-t(covnames)}
  xtmp<-data.matrix(xtmp)

#START DV LOOP
  for (i in (2:nrow(bcmat)))
  {
    wdid<-0;zdid<-0;wzdid<-0;cntmp<-1;start<-1;
    if (i < nrow(bcmat))
    {outv<-mtmp[,(i-1)]
    modlabel<-matrix(c(mnames[1,(i-1)],"constant"))}
    if (i==nrow(bcmat))
    {outv<-ytmp;modlabel<-matrix(c(ynames,"constant"))}
    # The j loop is the mediator
    # START MED

    for (j in (1:(i-1)))
    {
      foccnt<-foccnt+1
      if ((j==1) & (bcmat[i,j]==1))
      {
        outv<-cbind(outv,xtmp)
        modlabel<-matrix(c(modlabel,xcatlab[1:nxvls,1]))
        if (xtmpuse==0)
        {
          fulldat<-matrix(c(fulldat,xtmp),nrow=n)
          xtmpuse<-1
          for (k4 in (datcount:(datcount+(nxvls-1))))
          {xtmploc<-matrix(c(xtmploc,k4))}
          xtmploc<-matrix(xtmploc[2:nrow(xtmploc),1])
          datcount<-datcount+nxvls
        }
        datindx[start:(start+nrow(xtmploc)-1),(i-1)]<-xtmploc
        wherex[1,(i-1)]<-start+1
        wherex[2,(i-1)]<-start+nrow(xtmploc)-1+1
      #do if (model = 74)
      #end if
        onebl<-matrix(1,nrow(xtmploc),1)
        fochigh[(start+1):(start+nrow(xtmploc)),foccnt]<-onebl
        start<-start+nrow(xtmploc)
      }
      if ((j > 1) & (bcmat[i,j]==1))
        {
          outv<-cbind(outv,mtmp[,(j-1)])
          modlabel<-matrix(c(modlabel,mnames[1,(j-1)]))
          if (mtmpuse[1,(j-1)]==0)
          {
            fulldat<-matrix(c(fulldat,mtmp[,(j-1)]),nrow=n)
            mtmpuse[1,(j-1)]<-1
            mtmploc[1,(j-1)]<-datcount
            datcount<-datcount+1
          }
          datindx[start:(start+nrow(mtmploc)-1),(i-1)]<-mtmploc[1,(j-1)]
          wherem[(j-1),(i-1)]<-start+1
          #onebl<-matrix(1,(nrow(mtmploc[1,(j-1)])),1)
          #onebl<-matrix(1,mtmploc[1,(j-1)],1)
          onebl<-1
          #ttt<-nrow(mtmploc[1,(j-1)])+start-1
          #fochigh[(start+1):(start+nrow(mtmploc[1,(j-1)])),foccnt]<-onebl
          #fochigh[(start+1):(start+mtmploc[1,(j-1)]),foccnt]<-onebl
          fochigh[(start+1):(start+1),foccnt]<-onebl
          #start<-start+mtmploc[1,(j-1)]
          start<-start+1
        }
    }
    #END MED

    #START W
    if (wsum > 0)
    wtmp<-data.matrix(wtmp)
    {
      for (j in c(1:(i-1)))
      {
        whighct<-whighct+1
        if ((j==1) & (wcmat[i,j]==1))
        {
          if (wdid==0)
          {
            outv<-cbind(outv,wtmp)
            if ((ncs > 0) & (wiscov > 0)){ccmatoff[(i-1),wiscov]<-0}
            modlabel<-matrix(c(modlabel,wcatlab[1:nwvls,1]))
            wdid<-1
            if (wtmpuse==0)
            {
                fulldat<-matrix(c(fulldat,wtmp),nrow=n)
                if ((ncs > 0) & (wiscov > 0)){ccmatoff[(i-1),wiscov]<-0}
                wtmpuse<-1
                for (k4 in (datcount:(datcount+(nwvls-1))))
                {wtmploc<-matrix(c(wtmploc,k4))}
                wtmploc<-matrix(wtmploc[2:nrow(wtmploc),1])
                datcount<-datcount+nwvls
            }
          }
          datindx[start:(start+nrow(wtmploc)-1),(i-1)]<-wtmploc
          wherew[1,(i-1)]<-start+1
          wherew[2,(i-1)]<-start+nrow(wtmploc)-1+1
          start<-start+nrow(wtmploc)
```

```
         for (k1 in c(1:nxvls))
         {
          for (k2 in c(1:nwvls))
          {
           outv<-as.matrix(cbind(outv,(xtmp[,k1]*wtmp[,k2])))
           if ((ncs > 0) & (wiscov > 0)){ccmatoff[(i-1),wiscov]<-0}
           modlabel<-matrix(c(modlabel,intlab[cntmp,1]))
           intkeyt<-matrix(c(intlab[cntmp,1],":",xcatlab[k1,1],"x",wcatlab[k2,1]," "," "),nc
           intkey<-rbind(intkey,intkeyt)
           cntmp<-cntmp+1
          }
         }
         if (xwtmpus==0)
         {
          fulldat<-matrix(c(fulldat,outv[,(ncol(outv)-(nxvls*nwvls)+1):ncol(outv)]),nrow=n)
          xwtmpus<-1
          if ((ncs > 0) & (wiscov > 0)){ccmatoff[(i-1),wiscov]<-0}
          for (k4 in (datcount:(datcount+((nwvls*nxvls)-1))))
          {xwtmplo<-matrix(c(xwtmplo,k4))}
          xwtmplo<-matrix(xwtmplo[2:nrow(xwtmplo),1])
          datcount<-datcount+(nxvls*nwvls)
         }
         datindx[start:(start+nrow(xwtmplo)-1),(i-1)]<-xwtmplo
         wherexw[1,(i-1)]<-start+1
         wherexw[2,(i-1)]<-start+nrow(xwtmplo)-1+1
         onebl<-matrix(1,nrow(xwtmplo),1)
         whigh[(start+1):(start+nrow(xwtmplo)),whighct]<-onebl
         start<-start+nrow(xwtmplo)
        }
        if ((j > 1) & (wcmat[i,j]==1))
        {
         if ((wdid==0) & (model != 74))
         {
          outv<-cbind(outv,wtmp)
          if ((ncs > 0) & (wiscov > 0)){ccmatoff[(i-1),wiscov]<-0}
          modlabel<-matrix(c(modlabel,wcatlab[1:nwvls,1]))
          wdid<-1
          if (wtmpuse==0)
          {
           fulldat<-matrix(c(fulldat,wtmp),nrow=n)
           if ((ncs > 0) & (wiscov > 0)){ccmatoff[(i-1),wiscov]<-0}
           wtmpuse<-1
           for (k4 in (datcount:(datcount+(nwvls-1))))
           {wtmploc<-matrix(c(wtmploc,k4))}
           wtmploc<-matrix(wtmploc[2:nrow(wtmploc),1])
           datcount<-datcount+nwvls
          }
          datindx[start:(start+nrow(wtmploc)-1),(i-1)]<-wtmploc
          wherew[1,(i-1)]<-start+1
          wherew[2,(i-1)]<-start+nrow(wtmploc)-1+1
          start<-start+nrow(wtmploc)
         }
         for (k2 in c(1:nwvls))
         {
          outv<-as.matrix(cbind(outv,(mtmp[,(j-1)]*wtmp[,k2])))
          if ((ncs > 0) & (wiscov > 0)){ccmatoff[(i-1),wiscov]<-0}
          modlabel<-matrix(c(modlabel,intlab[cntmp,1]))
          intkeyt<-matrix(c(intlab[cntmp,1],":",mnames[1,(j-1)],"x",wcatlab[k2,1]," "," "),nc
          intkey<-rbind(intkey,intkeyt)
          cntmp<-cntmp+1
         }
         if (mwtmpus[1,(j-1)]==0)
         {
          fulldat<-matrix(c(fulldat,outv[,(ncol(outv)-nwvls+1):ncol(outv)]),nrow=n)
          if ((ncs > 0) & (wiscov > 0)){ccmatoff[(i-1),wiscov]<-0}
          mwtmpus[1,(j-1)]<-1
          mw22<- -999
          for (k4 in (datcount:(datcount+(nwvls-1))))
          {mw22<-matrix(c(mw22,k4))}
          mwtmplo[,(j-1)]=matrix(mw22[2:nrow(mw22),1])
          datcount<-datcount+nwvls
         }
         datindx[start:(start+nrow(mwtmplo)-1),(i-1)]<-mwtmplo[,(j-1)]
         wheremw[((2*j)-3),(i-1)]<-start+1
         wheremw[((2*j)-2),(i-1)]<-start+nrow(mwtmplo)-1+1
         onebl<-matrix(1,nrow(mwtmplo),1)
         whigh[(start+1):(start+nrow(mwtmplo)),whighct]<-onebl
         start<-start+nrow(mwtmplo)
        }
      }
    }
    #END W
    #START Z
    if (zsum > 0)
    ztmp<-data.matrix(ztmp)
    {
      for (j in c(1:(i-1)))
      {
        zhighct<-zhighct+1
        if ((j==1) & (zcmat[i,j]==1))
        {
         if (zdid==0)
         {
          outv<-cbind(outv,ztmp)
          if ((ncs > 0) & (ziscov > 0)){ccmatoff[(i-1),ziscov]<-0}
          modlabel<-matrix(c(modlabel,zcatlab[1:nzvls,1]))
          zdid<-1
          if (ztmpuse==0)
          {
           fulldat<-matrix(c(fulldat,ztmp),nrow=n)
           if ((ncs > 0) & (ziscov > 0)){ccmatoff[(i-1),ziscov]<-0}
           ztmpuse<-1
           for (k4 in (datcount:(datcount+(nzvls-1))))
           {ztmploc<-matrix(c(ztmploc,k4))}
           ztmploc<-matrix(ztmploc[2:nrow(ztmploc),1])
           datcount<-datcount+nzvls
          }
         }
         datindx[start:(start+nrow(ztmploc)-1),(i-1)]<-ztmploc
         wherez[1,(i-1)]<-start+1
      wherez[2,(i-1)]<-start+nrow(ztmploc)-1+1
      start<-start+nrow(ztmploc)
      for (k1 in c(1:nxvls))
      {for (k2 in c(1:nzvls))
        {
         outv=as.matrix(cbind(outv,(xtmp[,k1]*ztmp[,k2])))
         if ((ncs > 0) & (ziscov > 0)){ccmatoff[(i-1),ziscov]<-0}
         modlabel<-matrix(c(modlabel,intlab[cntmp,1]))
         intkeyt<-matrix(c(intlab[cntmp,1],":",xcatlab[k1,1],"x",zcatlab[k2,1],"  ","  "),nc
         intkey<-rbind(intkey,intkeyt)
         cntmp<-cntmp+1
        }
      }
      if (xztmpus==0)
      {
        fulldat<-matrix(c(fulldat,outv[,(ncol(outv)-(nxvls*nzvls)+1):ncol(outv)]),nrow=n)
        if ((ncs > 0) & (ziscov > 0)){ccmatoff[(i-1),ziscov]<-0}
        xztmpus<-1
        for (k4 in (datcount:(datcount+((nzvls*nxvls)-1)))){xztmplo<-matrix(c(xztmplo,k4))
        xztmplo<-matrix(xztmplo[2:nrow(xztmplo),1])
        datcount<-datcount+(nxvls*nzvls)
      }
      datindx[start:(start+nrow(xztmplo)-1),(i-1)]<-xztmplo
      wherexz[1,(i-1)]<-start+1
      wherexz[2,(i-1)]<-start+nrow(xztmplo)-1+1
      onebl<-matrix(1,nrow(xztmplo),1)
      zhigh[(start+1):(start+nrow(xztmplo)),zhighct]<-onebl
      start<-start+nrow(xztmplo)
    }
    if ((j > 1) & (zcmat[i,j]==1))
    {
      if (zdid==0)
      {
        outv<-cbind(outv,ztmp)
        if ((ncs > 0) & (ziscov > 0)){ccmatoff[(i-1),ziscov]<-0}
        modlabel<-matrix(c(modlabel,zcatlab[1:nzvls,1]))
        zdid<-1
        if (ztmpuse==0)
        {
         fulldat<-matrix(c(fulldat,ztmp),nrow=n)
         if ((ncs > 0) & (ziscov > 0)){ccmatoff[(i-1),ziscov]<-0}
         ztmpuse<-1
         for (k4 in (datcount:(datcount+(nzvls-1))))
         {ztmploc<-matrix(c(ztmploc,k4))}
         ztmploc<-matrix(ztmploc[2:nrow(ztmploc),1])
         datcount<-datcount+nzvls
        }
        datindx[start:(start+nrow(ztmploc)-1),(i-1)]<-ztmploc
        wherez[1,(i-1)]<-start+1
        wherez[2,(i-1)]<-start+nrow(ztmploc)-1+1
        start<-start+nrow(ztmploc)
      }
      for (k2 in (1:nzvls))
      {
        outv=as.matrix(cbind(outv,(mtmp[,(j-1)]*ztmp[,k2])))
```

```
        if ((ncs > 0) & (ziscov > 0)){ccmatoff[(i-1),ziscov]<-0}
        modlabel<-matrix(c(modlabel,intlab[cntmp,1]))
        intkeyt<-matrix(c(intlab[cntmp,1],":", mnames[1,(j-1)],"x",zcatlab[k2,1]," "," "),
        intkey<-rbind(intkey,intkeyt)
        cntmp<-cntmp+1
      }
      if (mztmpus[1,(j-1)]==0)
      {
        fulldat<-matrix(c(fulldat,outv[,(ncol(outv)-nzvls+1):ncol(outv)]),nrow=n)
        if ((ncs > 0) & (ziscov > 0)){ccmatoff[(i-1),ziscov]<-0}
        mztmpus[1,(j-1)]<-1
        mz22<- -999
        for (k4 in (datcount:(datcount+(nzvls-1)))){mz22<-matrix(c(mz22,k4))}
        mztmplo[,(j-1)]<-matrix(mz22[2:nrow(mz22),1])
        datcount<-datcount+nzvls
      }
      datindx[start:(start+nrow(mztmplo)-1),(i-1)]<-mztmplo[,(j-1)]
      wheremz[((2*j)-3),(i-1)]<-start+1
      wheremz[((2*j)-2),(i-1)]<-start+nrow(mztmplo)-1+1
      onebl<-matrix(1,nrow(mztmplo),1)
      zhigh[(start+1):(start+nrow(mztmplo)),zhighct]<-onebl
      start<-start+nrow(mztmplo)
    }
  }
}
#END Z

#START WZ
if (wzsum > 0)
{
  for (j in (1:(i-1)))
  {
    wzhighct<-wzhighct+1
    if ((j==1) & (wzcmat[i,j]==1))
    {
      if (wzdid==0)
      {
        for (k1 in (1:nwvls))
        {
          for (k2 in (1:nzvls))
          {
            outv<-as.matrix(cbind(outv,(wtmp[,k1]*ztmp[,k2])))
            if ((ncs > 0) & (ziscov > 0)){ccmatoff[(i-1),ziscov]<-0}
            if ((ncs > 0) & (wiscov > 0)){ccmatoff[(i-1),wiscov]<-0}
            modlabel<-matrix(c(modlabel,intlab[cntmp,1]))
            intkeyt<-matrix(c(intlab[cntmp,1],":",wcatlab[k1,1],"x",zcatlab[k2,1]," "," "),
            intkey<-rbind(intkey,intkeyt)
            cntmp<-cntmp+1
          }
        }
        if (wztmpus==0)
        {
          fulldat<-matrix(c(fulldat,outv[,(ncol(outv)-(nwvls*nzvls)+1):ncol(outv)]),nrow=n)
          if ((ncs > 0) & (ziscov > 0)){ccmatoff[(i-1),ziscov]<-0}
          if ((ncs > 0) & (wiscov > 0)){ccmatoff[(i-1),wiscov]<-0}
```

```
                    wztmpus<-1
                    for (k4 in (datcount:(datcount+((nwvls*nzvls)-1))))
                    {wztmplo<-matrix(c(wztmplo,k4))}
                    wztmplo<-matrix(wztmplo[2:nrow(wztmplo),1])
                    datcount<-datcount+(nzvls*nwvls)
                  }
                  wzdid<-1
                }
                datindx[start:(start+nrow(wztmplo)-1),(i-1)]<-wztmplo
                wherewz[1,(i-1)]<-start+1
                wherewz[2,(i-1)]<-start+nrow(wztmplo)-1+1
                start<-start+nrow(wztmplo)
                for (k1 in (1:nxvls))
                {
                  for (k2 in (1:nwvls))
                  {
                    for (k3 in (1:nzvls))
                    {
                      outv<-as.matrix(cbind(outv,(xtmp[,k1]*wtmp[,k2]*ztmp[,k3])))
                      if ((ncs > 0) & (ziscov > 0)){ccmatoff[(i-1),ziscov]<-0}
                      if ((ncs > 0) & (wiscov > 0)){ccmatoff((i-1),wiscov)<-0}
                      modlabel<-matrix(c(modlabel,intlab[cntmp,1]))
                      intkeyt<-matrix(c(intlab[cntmp,1],":",xcatlab[k1,1],"x",wcatlab[k2,1],"x",zcatlab
                      intkey<-rbind(intkey,intkeyt)
                      cntmp<-cntmp+1
                    }
                  }
                }
                if (xwztmpu==0)
                {
                  fulldat<-matrix(c(fulldat,outv[,(ncol(outv)-(nxvls*nwvls*nzvls)+1):ncol(outv)]),nrow=
                  if ((ncs > 0) & (ziscov > 0)){ccmatoff[(i-1),ziscov]<-0}
                  if ((ncs > 0) & (wiscov > 0)){ccmatoff((i-1),wiscov)<-0}
                  xwztmpu<-1
                  for (k4 in (datcount:(datcount+((nzvls*nxvls*nwvls)-1))))
                  {xwztmplo<-matrix(c(xwztmplo,k4))}
                  xwztmplo<-matrix(xwztmplo[2:nrow(xwztmplo),1])
                  datcount<-datcount+(nxvls*nzvls*nwvls)
                }
                datindx[start:(start+nrow(xwztmplo)-1),(i-1)]<-xwztmplo
                wherexwz[1,(i-1)]<-start+1
                wherexwz[2,(i-1)]<-start+nrow(xwztmplo)-1+1
                onebl<-matrix(1,nrow(xwztmplo),1)
                wzhigh[(start+1):(start+nrow(xwztmplo)),wzhighct]<-onebl
                start<-start+nrow(xwztmplo)
              }
              if ((j > 1) & (wzcmat[i,j]==1))
              {
                if (wzdid==0)
                {
                  for (k1 in (1:nwvls))
                  {
                    for (k2 in (1:nzvls))
                    {
                      outv<-as.matrix(cbind(outv,(wtmp[,k1]*ztmp[,k2])))
```

```
      if ((ncs > 0) & (ziscov > 0)){ccmatoff[(i-1),ziscov]<-0}
      if ((ncs > 0) & (wiscov > 0)){ccmatoff[(i-1),wiscov]<-0}
      modlabel<-matrix(c(modlabel,intlab[cntmp,1]))
      intkeyt<-matrix(c(intlab[cntmp,1],":",wcatlab[k1,1],"x",zcatlab[k2,1]," "," "),
      intkey<-rbind(intkey,intkeyt)
      cntmp<-cntmp+1
    }
  }
  if (wztmpus==0)
  {
    fulldat<-matrix(c(fulldat,outv[,(ncol(outv)-(nwvls*nzvls)+1):ncol(outv)]),nrow=n)
    if ((ncs > 0) & (ziscov > 0)){ccmatoff[(i-1),ziscov]<-0}
    if ((ncs > 0) & (wiscov > 0)){ccmatoff[(i-1),wiscov]<-0}
    wztmpus<-1
    for (k4 in (datcount:(datcount+((nwvls*nzvls)-1))))
    {wztmplo<-matrix(c(wztmplo,k4))}
    wztmplo<-matrix(wztmplo[2:nrow(wztmplo),1])
    datcount<-datcount+(nzvls*nwvls)
  }
  wzdid<-1
  datindx[start:(start+nrow(wztmplo)-1),(i-1)]<-wztmplo
  wherewz[1,(i-1)]<-start+1
  wherewz[2,(i-1)]<-start+nrow(wztmplo)-1+1
  start<-start+nrow(wztmplo)
}
for (k1 in (1:nwvls))
{
  for (k2 in (1:nzvls))
  {
    outv<-as.matrix(cbind(outv,(mtmp[,(j-1)]*wtmp[,k1]*ztmp[,k2])))
    if ((ncs > 0) & (ziscov > 0)){ccmatoff[(i-1),ziscov]<-0}
    if ((ncs > 0) & (wiscov > 0)){ccmatoff[(i-1),wiscov]<-0}
    modlabel<-matrix(c(modlabel,intlab[cntmp,1]))
    intkeyt<-matrix(c(intlab[cntmp,1],":",mnames[1,(j-1)],"x",wcatlab[k1,1],"x",zcatlab
    intkey<-rbind(intkey,intkeyt)
    cntmp<-cntmp+1
  }
}
if (mwztmpu[1,(j-1)]==0)
{
  fulldat<-matrix(c(fulldat,outv[,(ncol(outv)-(nwvls*nzvls)+1):ncol(outv)]),nrow=n)
  if ((ncs > 0) & (ziscov > 0)){ccmatoff[(i-1),ziscov]<-0}
  if ((ncs > 0) & (wiscov > 0)){ccmatoff[(i-1),wiscov]<-0}
  mwztmpu[1,(j-1)]<-1
  mz22<- -999
  for (k4 in (datcount:(datcount+(nwvls*nzvls)-1)))
  {mz22<-matrix(c(mz22,k4))}
  mwztmplo[,(j-1)]<-matrix(mz22[2:nrow(mz22),1])
  datcount<-datcount+(nwvls*nzvls)
}
datindx[start:(start+nrow(mwztmplo)-1),(i-1)]<-mwztmplo[,(j-1)]
wheremwz[((2*j)-3),(i-1)]<-start+1
wheremwz[((2*j)-2),(i-1)]<-start+nrow(mwztmplo)-1+1
onebl<-matrix(1,nrow(mwztmplo),1)
wzhigh[(start+1):(start+nrow(mwztmplo)),wzhighct]<-onebl
```

```
        start<-start+nrow(mwztmplo)
      }
    }
  }
  #END WZ

  #START COV
  if (ncs > 0)
  {
    ccmat<-ccmat*ccmatoff
    for (j in c(1:ncs))
    {
      if (ccmat[(i-1),j]==1)
      {
        if (j==wiscov){ctmp[,j]<-wtmp}
        if (j==ziscov){ctmp[,j]<-ztmp}
        outv<-cbind(outv,ctmp[,j])
        modlabel<-matrix(c(modlabel,covnames[1,j]))
        if (ctmpuse[1,j]==0)
        {
          fulldat<-matrix(c(fulldat,ctmp[,j]),nrow=n)
          ctmpuse[1,j]<-1
          ctmploc[1,j]<-datcount
          datcount<-datcount+1
        }
        datindx[start:(start+nrow(ctmploc)-1),(i-1)]<-ctmploc[1,j]
        start<-start+1
      }
    }
  }
  #END COV

  wdid<-0;zdid<-0;wzdid<-0
  vlabs<-matrix(c(vlabs,modlabel[2:nrow(modlabel),1]))
  numint[1,(i-1)]<-cntmp-1
  nump[1,(i-1)]<-nrow(modlabel)-1

}
#END DV LOOP
if ((modcok==1) & ((nms > 0) | (zcmat[2,1] != 1) | (mcx != 0)))
{notecode[notes,1]<-19;notes<-notes + 1;modcok<-0}
if (((serial==1) | (sum(numint)>0) | (nms==0)) & (mc > 0))
{notecode[notes,1]<-15;notes<-notes+1;boot<-mc;mc<-0}


if ((boot != 0) | (mc != 0))
{
  bootsz<-boot
  if (mc > 0){bootsz<-mc;saveboot<-0}
  cilow<-0;cihigh<-bootsz+1
  while ((cilow <=0) | (cihigh > bootsz))
  {
    cilow<-round(bootsz*(1-(conf/100))/2)
    cihigh<-trunc((bootsz*(conf/100)+(bootsz*(1-(conf/100))/2)))+1
    if ((cilow < 1) | (cihigh > bootsz))
    {bootsz<-trunc((bootsz+1000)/1000)*1000;adjust<-1}
  }
  if (boot > 0){boot<-bootsz}
  if (mc > 0){mc<-bootsz}
  if ((adjust==1) & (boot > 0)){notecode[notes,1]<-8;notes<-notes+1}
  if ((adjust==1) & (mc > 0)){notecode[notes,1]<-16;notes<-notes+1}
}
maxboot<-trunc(2*boot)
if (maxboots > maxboot){maxboot<-trunc(maxboots)}
#insert seed here
if ((seed==-999) & (boot > 0) | (mc > 0))
{seed<-trunc(runif(1,1,1000000));set.seed(seed);ranseed<-1}
if (sum(numint) > 0){intkey<-matrix(intkey[2:nrow(intkey),],ncol=7)}
vlabs<-matrix(vlabs[2:nrow(vlabs),1])
fulldat<-as.data.frame(fulldat[,2:ncol(fulldat)])
fulldat<-data.matrix(fulldat)
fochigh<-fochigh[1:max(nump),]
whigh<-as.matrix(whigh[1:max(nump),])
zhigh<-as.matrix(zhigh[1:max(nump),])
wzhigh<-as.matrix(wzhigh[1:max(nump),])
coeffs<-fochigh+whigh+zhigh+wzhigh
bootloc<-matrix(0,max(nump),ncol(nump))

#Here is am deriving the locations in bootfile needed for indirect effects
if (nms > 0)
{
  cntmp<-1
  for (i in (1:ncol(nump)))
  {for (j in (1:nump[1,i]))
    {bootloc[j,i]<-cntmp;cntmp<-cntmp+1}
  }
  fochighb<-matrix(0,nrow(fochigh),ncol(fochigh))
  whighb<-fochighb
  zhighb<-fochighb
  wzhighb<-fochighb
  thetaxmb<-matrix(0,nrow(fochighb),nms)
  thetaxyb<-matrix(0,nrow(fochighb),1)
  pathsfoc<-matrix(0,nxvls,1)
  cntmp<-1
  for (i in (1:(nms+nys)))
  {
    for (j in (1:i))
    {
      fochighb[,cntmp]<-(fochigh[,cntmp]*bootloc[,i])
      whighb[,cntmp]<-(whigh[,cntmp]*bootloc[,i])
      zhighb[,cntmp]<-(zhigh[,cntmp]*bootloc[,i])
      wzhighb[,cntmp]<-(wzhigh[,cntmp]*bootloc[,i])
      coeffsb<-(fochighb+whighb+zhighb+wzhighb)
      if ((i < (nms+nys)) & (j==1)){thetaxmb[,i]<-coeffsb[,cntmp]}
      if ((i==(nms+nys)) & (j==1)){thetaxyb[,1]<-coeffsb[,cntmp]}
      cntmp<-cntmp+1
    }
  }
  thetamyb<-as.matrix(coeffsb[,(ncol(coeffsb)-nms+1):ncol(coeffsb)])
  if (serial==1)
```

```
    {thetammb<-matrix(0,nrow(coeffsb),((nms*(nms-1))/2))}
    cntmp<-1
    if ((nms > 1) & (serial==1))
    {
      for (i in (1:(nms-1)))
      {
        start<-((i+2)*(i+1))/2
        for (j in (2:(nms-i+1)))
        {
          thetammb[,cntmp]<-coeffsb[,start]
          start<-start+j+i-1
          cntmp<-cntmp+1
        }
      }
    }
  }

  if ((total==1) & (sum(numint)==0))
  {
    dototal<-1
    if ((sum(bcmat[,1]) != (nms+nys)) | (sum(bcmat[nrow(bcmat),]) != (nms+nys)))
    {dototal<-0;notecode[notes,1]<-12;notes<-notes+1}
    if (ncs > 0)
    {
      if ((sum(ccmat)) < (nrow(ccmat)*ncol(ccmat)))
      {dototal<-0;notecode[notes,1]<-11;notes<-notes+1}
    }
  }
}
# End D
if ((criterr==0) & (ncs > 0))
{
  tmperr<-sum(as.numeric(colSums(ccmat)==0))
  if (sum(tmperr)!=0)
  {errcode[errs,1]<-51;errs<-errs+1;criterr<-1}
}
if (outscreen==1)
{
  cat("\n")
  cat("***************** PROCESS for R Version 3.5.3 beta0.6 ***************** \n \n")
  cat("                    Written by Andrew F. Hayes, Ph.D.    www.afhayes.com
  cat("     Documentation available in Hayes (2018). www.guilford.com/p/hayes3     \n \n"
}
if (criterr==0)
{
  if ((stand==1) & (ydich==1)){stand<-0}
  anymod2<-sum(wcmat+zcmat+wzcmat)
  if ((anymod2 > 0) & (stand==1))
  {notecode[notes,1]<-27;notes<-notes+1;stand<-0}
  rownames(modelvar)<-modelvlb
  colnames(modelvar)<-"  "
  funny<-1
  if (outscreen==1)
  {
    cat("************************************************************************ \n")
    print(noquote(modelvar))
    cat("\n")
    if (ncs > 0)
    {
    cat("Covariates: \n")
    covname2<-c("            ", covnames)
    covname2<-t(noquote(covname2))
    write.table(covname2,quote=FALSE,row.names=FALSE,col.names=FALSE)
    cat("\n")
    }
    cat("Sample size: ")
    write.table(n,quote=FALSE,row.names=FALSE,col.names=FALSE)
    cat("\n")
    if (seed != -999)
    {if (ranseed==0)
      {cat("Custom seed: ")}
      if (ranseed==1)
      {cat("Random seed: ")}
      write.table(seed,quote=FALSE,row.names=FALSE,col.names=FALSE)
      cat("\n")
    }
    if (mcxok==1)
    {cat("Coding of categorical X variable for analysis: \n")
    dummatx2<-noquote(matrix(sprintf(decimals,dummatx),nrow=nrow(dummatx)))
    colnames(dummatx2)<-c(xnames,xcatlab[1:nxvls,1])
    rownames(dummatx2)<-t(matrix(replicate((nxvls+1),"  ")))
    print(dummatx2,right=T)}
    if (mcwok==1)
    {cat("Coding of categorical W variable for analysis: \n")
    dummatw2<-noquote(matrix(sprintf(decimals,dummatw),nrow=nrow(dummatw)))
    colnames(dummatw2)<-c(wnames,t(wcatlab[1:nwvls,1]))
    rownames(dummatw2)<-t(matrix(replicate((nwvls+1),"  ")))
    print(dummatw2,right=T)}
    if (mczok==1)
    {cat("Coding of categorical Z variable for analysis: \n")
    dummatz2<-noquote(matrix(sprintf(decimals,dummatz),nrow=nrow(dummatz)))
    colnames(dummatz2)<-c(znames,t(zcatlab[1:nzvls,1]))
    rownames(dummatz2)<-t(matrix(replicate((nzvls+1),"  ")))
    print(dummatz2,right=T)}
  }
}
#START cycle through the models
if (criterr==0)
{
  outnames<-ynames
  outvars<-ytmp
  if (nms > 0)
  {
    outnames<-matrix(c(mnames,ynames))
    outvars<-cbind(mtmp,ytmp)
    indcov<-matrix(0,((nms*2)+(nms*(nxvls-1))),((nms*2)+(nms*(nxvls-1))))
    mcsopath<-matrix(0,((nms*2)+(nms*(nxvls-1))),1)
  }
  labstart<-1
  intstart<-1
```

```
start<-1
coeffmat<-matrix(0,1,6)
conseq<-"                    "
dfmat<-0;coeffcol<-0;pathscnt<-1;pathscn2<-1

# START G LOOP
for (i in c(1:(nms+nys)))
{
  if (outscreen==1)
  {cat("\n************************************************************************  \n")}
  highf<-matrix(0,1,5);highf2<-highf
  if ((i==(nms+nys)) & (ydich==1)){highf=matrix(0,1,3);highf2<-highf}
  flabel<-" "
  y<-outvars[,i]
  xindx<-datindx[1:(nump[1,i]-1),i]
  x<-fulldat[,xindx]
  x<-cbind(ones,x)
  xsq<-t(x)%*%x
  exsq<-eigen(xsq)
  exsq<-matrix(unlist(exsq[1]))
  zeroeig<-sum(as.numeric(exsq <= 0.000000000002))
  if (outscreen==1)
  {
    cat("Outcome Variable: ")
    write.table(outnames[i,1],quote=FALSE,row.names=FALSE,col.names=FALSE)
    cat("\n")
    if ((ydich==1) & (i==(nms+nys)))
    {
      cat("Coding of binary Y for logistic regression analysis:\n")
      rcd<-noquote(matrix(sprintf(decimals,rcd),nrow=nrow(rcd)))
      colnames(rcd)<-c(outnames[i,1], "Analysis")
      rownames(rcd)<-t(matrix(replicate(nrow(rcd)," ")))
      print(rcd,right=T)
      cat("\n")
    }
  }
  if (zeroeig > 0)
  {
    cat("\nSINGULAR OR NEAR SINGULAR DATA MATRIX.\n")
    criterr<-1;errcode[errs,1]<-31;errs<-errs+1;
  }
  means<-colSums(x)/n
  vlabsm<-matrix(vlabs[labstart:(labstart+(nump[1,i]-1)),1])
  #START E
  if (criterr==0)
  {
    if ((ydich==0) | (i < (nms+nys)))
    {
      modoutz<-process.modelest(y,x,1,1,xp2,hc)
      modres<-matrix(unlist(modoutz[1]),nrow=nump[1,i])
      direff<-matrix(modres,nrow=ncol(x))
      #modresl<-matrix(unlist(modoutz[2]))
      modsum<-matrix(unlist(modoutz[3]))
      tval<-matrix(unlist(modoutz[7]))
      resid<-matrix(unlist(modoutz[8]))
      modresid<-cbind(modresid,resid)
      modresid<-cbind(modresid,resid)
      brsq2<-modsum[2,1]
      dfres<-modsum[6,1]
      #for recording results
      resultm2<-matrix(99999,1,maxresm)
      resultm2[1,1:nrow(modsum)]<-t(modsum)
      resultm<-rbind(resultm,resultm2)
      resultm2<-matrix(99999,nrow(modres),maxresm)
      resultm2[1:nrow(modres),1:ncol(modres)]<-modres
      resultm<-rbind(resultm,resultm2)
      #end recording results
      modrest9<-modres
      modres<-noquote(matrix(sprintf(decimals,modres),nrow=nump[1,i]))
      modsum<-noquote(matrix(sprintf(decimals,modsum),nrow=1))
      b<-matrix(unlist(modoutz[5]))
      varb<-matrix(unlist(modoutz[6]),nrow=nump[1,i])
      dfmatt<-matrix(modsum[1,6],nrow(modres),1)
      modsuml<-matrix(c("R","R-sq","MSE",hcflab,"df1","df2", "p"))
      modresl<-t(matrix(c("coeff",hclab,"t","p","LLCI","ULCI")))
    }
    if ((ydich==1) & (i==(nms+nys)))
    {
      modoutz<-process.modelest(y,x,2,1,xp2,5,iterate,converge)
      modres<-matrix(unlist(modoutz[1]),nrow=nump[1,i])
      direff<-matrix(modres,nrow=ncol(x))
      modsum<-matrix(unlist(modoutz[3]))
      basemod<-modsum[1,1]
      basemodx<-basemod
      tval<-matrix(unlist(modoutz[7]))
      resultm2<-matrix(99999,1,maxresm)
      resultm2[1,1:nrow(modsum)]<-t(modsum)
      resultm<-rbind(resultm,resultm2)
      resultm2<-matrix(99999,nrow(modres),maxresm)
      resultm2[1:nrow(modres),1:ncol(modres)]<-modres
      resultm<-rbind(resultm,resultm2)
      modrest9<-modres
      modres<-noquote(matrix(sprintf(decimals,modres),nrow=nump[1,i]))
      modsum<-noquote(matrix(sprintf(decimals,modsum),nrow=1))
      b<-matrix(unlist(modoutz[5]))
      varb<-matrix(unlist(modoutz[6]),nrow=nump[1,i])
      modsuml<-matrix(c("-2LL","ModelLL", "df", "p", "McFadden", "CoxSnell", "Nagelkrk"))
      modresl<-t(matrix(c("coeff","se","Z","p","LLCI","ULCI")))
      #modretrn<-list(modres,modresl,modsum,modsuml,b,varb,xp2)
      dfmatt<-matrix(-999,nrow(modres),1)
    }
    obscoeff<-cbind(obscoeff,t(b))
    if (outscreen==1)
    {
      cat("Model Summary: \n")
      colnames(modsum)<-modsuml;rownames(modsum)<-" "
      print(modsum,right=T)
      cat("\n")
      cat("Model: \n")
      rownames(modres)<-vlabsm;colnames(modres)<-modresl
      print(modres,right=T)
      if ((ydich==1) & (i==(nms+nys)))
```

```
  if ((ydich==1) & (i==(nms+nys)))
  {
    cat("\nThese results are expressed in a log-odds metric.\n")
    notecode[notes,1]<-26;notes<-notes+1
  }
}
coeffmat<-rbind(coeffmat,modres)
conseqt<-matrix(outnames[i,1],nrow(modres),1)
conseq<-matrix(c(conseq,conseqt))
dfmat<-rbind(dfmat,dfmatt)
labstart<-labstart+nump[1,i]

if (stand==1)
{
  predsd<-matrix(0,nrow(modres),1)
  stdmod<-as.numeric(modres[,1])/ovsd[i,1]
  for (jd in c(1:ncol(x)))
  {
  predsd[jd,1]<-sd(x[,jd])
  }
  if ((wherex[1,i] != -999) & ((nxvls > 1) | (xdich==1)))
  {
    sdmsone<-matrix(1,nxvls,1)
    predsd[wherex[1,i]:wherex[2,i],1]<-sdmsone
    pstog<-1
  }
  predsd[1,1]<-1
  stdmod<-stdmod*predsd
  stdmod<-matrix(stdmod[2:nrow(stdmod),1])
  sdvlabs<-vlabsm[2:nrow(vlabsm),1]
  if (outscreen==1){cat("\nStandardized coefficients:\n")}
  resultm2<-matrix(99999,nrow(stdmod),maxresm)
  resultm2[1:nrow(stdmod),1:ncol(stdmod)]<-stdmod
  resultm<-rbind(resultm,resultm2)
  stdmod<-noquote(matrix(sprintf(decimals,stdmod),nrow=(nump[1,i]-1)))
  colnames(stdmod)<-"coeff"
  rownames(stdmod)<-sdvlabs
  if (outscreen==1)
  {print(stdmod,right=T)}
}
if ((nms > 0) & (serial==0) & (sum(numint)==0) & ((normal==1) | (mc > 0)))
{
  if (i < (nms+nys))
  {
    indcov[(((i-1)*nxvls)+1):(i*nxvls),(((i-1)*nxvls)+1):(i*nxvls)]<-varb[2:(1+nxvls),2:(1+
    mcsopath[(((i-1)*nxvls)+1):(i*nxvls),1]<-matrix(modrest9[2:(1+nxvls),1])
  }
  if (i==(nms+nys))
  {
    atm<-ncol(wherem)
    indcov[((nms*nxvls)+1):nrow(mcsopath),((nms*nxvls)+1):nrow(mcsopath)]<-varb[wherem[1,at
    mcsopath[((nms*nxvls)+1):nrow(mcsopath),1]<-matrix(modrest9[wherem[1,atm]:(wherem[1,atm
    sobelok<-1
  }
}
if ((i ==(nms+nys)) & (bcmat[nrow(bcmat),1]==1))
if ((i ==(nms+nys)) & (bcmat[nrow(bcmat),1]==1))
{
  direff<-matrix(direff[2:(1+nxvls),],nrow=nxvls)
  direfflb<-modresl
  direffl2<-vlabsm[2:(1+nxvls),]
  lmat<-matrix(0,nrow(b),1)
  lmat2<-matrix(1,nxvls,1)
  lmat[2:(1+nxvls),1]<-lmat2
  if (ydich != 1)
  {
    diromni<-process.ftest3(lmat,b,varb,1,brsq2,0,y,x)
  }
  if (ydich==1){diromni<-process.llrtest3(lmat,y,x,b,basemod,iterate,converge)}
}

if (numint[1,i] > 0)
{
  if (outscreen==1)
  {
    cat("\nProduct terms key:\n")
    intkeym<-matrix(intkey[intstart:(intstart+numint[1,i]-1),],ncol=7)
    write.table(intkeym,quote=FALSE,row.names=FALSE,col.names=FALSE,sep = "    ")
  }
}
if (covcoeff==1)
{
  if (outscreen==1){cat("\nCovariance matrix of regression parameter estimates:\n")}
  resultm2<-matrix(99999,nrow(varb),maxresm)
  if (ncol(varb) <= maxresm)
  {
    resultm2[1:nrow(varb),1:ncol(varb)]<-varb
    resultm<-rbind(resultm,resultm2)
  }
  if (ncol(varb) > maxresm)
  {
    resultmt<-matrix(99999,nrow(resultm),ncol(varb))
    resultmt[1:nrow(resultm),1:ncol(resultm)]<-resultm
    resultm<-resultmt
    resultm2<-matrix(99999,nrow(varb),ncol(resultm))
    resultm2[1:nrow(varb),1:ncol(varb)]<-varb
    resultm<-rbind(resultm,resultm2)
    maxresm<-ncol(resultm)
  }
  varbpr<-noquote(matrix(sprintf(decimals,varb),nrow=nump[1,i]))
  rownames(varbpr)<-vlabsm;colnames(varbpr)<-vlabsm
  if (outscreen==1)
  {print(varbpr,right=T)}
}
#start X by M interaction */
if ((model != 74) & (xmtest==1) & (nms > 0))
{
  r2tmp<-brsq2;btmp<-b;varbtmp<-varb
  dfrestmp<-dfres
  tvaltmp<-tval
  xmtst<-matrix(0,nms,4)
  xmtstlbs<-matrix(c(bcflab,"df1","df2","p"))
```

```
xmtstlbc<-matrix(c(ncrlab,″df1″,″df2″,″p″))
if ((i==(nms+nys)) & (ydich==1))
{xmtst<-matrix(0,nms,3);xmtstlbc<-c(″Chi-sq″,″df″,″p″)
}
xmtstlb<-″ ″;xmtmat<-x;numxint<-0
if (i > 1)
{
  for (xmint in (2:i))
  {
    x<-xmtmat
    if ((bcmat[(i+1),xmint]==1) & (wzcmat[(i+1),xmint] != 1))
    {
      if (bcmat[(i+1),1]==0)
      {
        x<-cbind(xmtmat,xtmp)
        if ((ydich==1) & (i==(nms+nys)))
        {
          LL2<-process.modelest(y,x,2,0,xp2,5,iterate,converge)
          basemodx<-LL2
        }
      }
      for (xmtlp1 in (1:nxvls))
      {
        mtmpmns<-sum(mtmp[,(xmint-1)])/nrow(mtmp)
        x<-as.matrix(cbind(x,(xtmp[,xmtlp1]*((mtmp[,(xmint-1)])-mtmpmns))))
      }
      if ((i < (nms+nys)) | (ydich==0))
      {
        modoutz<-process.modelest(y,x,1,1,xp2,hc)
        b<-matrix(unlist(modoutz[5]))
        varb<-matrix(unlist(modoutz[6]),nrow=nrow(b))
        lmat<-matrix(0,nrow(b),nxvls)
        lmattmp<-diag(nxvls)
        lmat[(nrow(lmat)-nxvls+1):nrow(lmat),]<-lmattmp
        fresult2<-process.ftest3(lmat,b,varb,0,0,1,y,x)
        numxint<-numxint+1
        xmtst[numxint,]<-fresult2
      }
      #right here
      if ((i==(nms+nys)) & (ydich==1))
      {
        LL2<-process.modelest(y,x,2,0,xp2,5,iterate,converge)
        chidfxm<-(basemodx-LL2)
        numxint<-numxint+1
        xmtst[numxint,1]<-chidfxm
        xmtst[numxint,3]<-(1-pchisq(chidfxm,df=nxvls))
      }
      xmtstlbt<-matrix(c(highlbx[(xmint-1),1]))
      xmtstlb<-rbind(xmtstlb,xmtstlbt)
    }
  }
}
x<-xmtmat
if (numxint > 0)
{
  xmtstlb<-matrix(xmtstlb[(2:(numxint+1)),])
```

```
    xmtstlb<-matrix(xmtstlb[(2:(numxint+1)),])
    xmtst<-matrix(xmtst[1:numxint,],nrow=numxint)
    if (nms==1){xmtstlb<-" "}
    resultm2<-matrix(99999,nrow(xmtst),maxresm)
    resultm2[1:nrow(xmtst),1:ncol(xmtst)]<-xmtst
    resultm<-rbind(resultm,resultm2)
    if ((i < (nms+nys)) | (ydich==0))
    {
      xmtst<-noquote(matrix(sprintf(decimals,xmtst),ncol=4))
      colnames(xmtst)<-xmtstlbc
      rownames(xmtst)<-xmtstlb
      if (outscreen==1)
      {cat("\nTest(s) of X by M interaction:\n")
      print(xmtst,right=T)}
    }
    if ((i==(nms+nys)) & (ydich==1))
    {
      xmtst<-noquote(matrix(sprintf(decimals,xmtst),ncol=3))
      colnames(xmtst)<-xmtstlbc
      rownames(xmtst)<-xmtstlb
      if (outscreen==1)
      {cat("\nLikelihood ratio test(s) of X by M interaction:\n")
      print(xmtst,right=T)}
    }
  }
  b<-btmp;varb<-varbtmp;dfres<-dfrestmp;tval<-tvaltmp;brsq2<-r2tmp
}
#end X by M interaction

#here is where we do F tests
#START F
if (criterr==0)
{
  jj<-0
  for (j in (start:((start+i)-1)))
  {
    dbint<-0
    lmat<-whigh[1:nump[1,i],j]
    lmat2<-wzhigh[1:nump[1,i],j]
    if ((sum(lmat) > 0) & (sum(lmat2)==0))
    {
      if ((i < (nms+nys)) | (ydich != 1))
      {
        fresult2<-process.ftest3(lmat,b,varb,chr=1,brsq2,0,y,x)
        lmatdb<-lmat
        dbint=dbint<-1
      }
      if ((ydich==1) & (i==(nms+nys)))
      {
        fresult2<-process.llrtest3(lmat,y,x,b,basemod,iterate,converge)
        lmatdb<-lmat
        dbint<-dbint+1
      }
      highf<-rbind(highf,fresult2)
      highf2<-rbind(highf2,fresult2)
      if (i==start){flabel<-matrix(c(flabel,"X*W"))}
      if (j==start){flabel<-matrix(c(flabel,"X*W"))}
      if (j > start)
      {
        if (nms > 1){flabel<-matrix(c(flabel,highlbw[jj,1]))}
        if (nms==1){flabel<-matrix(c(flabel,"M*W"))}
      }
    }
    lmat<-zhigh[1:nump[1,i],j]
    lmat2<-wzhigh[1:nump[1,i],j]
    if ((sum(lmat) > 0) & (sum(lmat2)==0))
    {
      if ((i < (nms+nys)) | (ydich != 1))
      {
        fresult2<-process.ftest3(lmat,b,varb,chr=1,brsq2,0,y,x)
        dbint<-dbint+1
      }
      if ((ydich==1) & (i==(nms+nys)))
      {
        fresult2<-process.llrtest3(lmat,y,x,b,basemod,iterate,converge)
        dbint<-dbint+1
      }
      highf<-rbind(highf,fresult2)
      highf2<-rbind(highf2,fresult2)
      if (j==start){flabel<-matrix(c(flabel,"X*Z"))}
      if (j > start)
      {
        if (nms > 1){flabel<-matrix(c(flabel,highlbz[jj,1]))}
        if (nms==1){flabel<-matrix(c(flabel,"M*Z"))}
      }
    }
    if (dbint==2)
    {
      lmatdb<-(lmatdb+lmat)
      if ((ydich==1) & (i==(nms+nys)))
      {fresult2<-process.llrtest3(lmatdb,y,x,b,basemod,iterate,converge)}
      if ((ydich != 1) | (i < (nms+nys)))
      {
        fresult2<-process.ftest3(lmatdb,b,varb,chr=1,brsq2,0,y,x)
      }
      dbint<-0
      highf<-rbind(highf,fresult2)
      if ((jj==0) & (nms > 0)){flabel<-matrix(c(flabel,"BOTH(X)"))}
      if ((jj==0) & (nms==0)){flabel<-matrix(c(flabel,"BOTH"))}
      if ((jj > 0) & (nms==1)){flabel<-matrix(c(flabel,"BOTH(M)"))}
      if ((nms > 1) & (jj > 0)){flabel<-matrix(c(flabel,highlbbt[jj,1]))}
    }
    lmat2<-wzhigh[1:nump[1,i],j]
    if (sum(lmat2) > 0)
    {
      if ((i < (nms+nys)) | (ydich != 1))
      {
        fresult2<-process.ftest3(lmat2,b,varb,chr=1,brsq2,0,y,x)
      }
      if ((ydich==1) & (i==(nms+nys)))
      {fresult2<-process.llrtest3(lmat2,y,x,b,basemod,iterate,converge)}
      highf<-rbind(highf,fresult2)
```

```
        highf<-rbind(highf,fresult2)
        highf2<-rbind(highf2,fresult2)
        if  (j==start){flabel<-matrix(c(flabel,"X*W*Z"))}
        if  (j  >  start)
        {
          if  (nms  >  1){flabel<-matrix(c(flabel,highlbwz[jj,1]))}
          if  (nms==1){flabel<-matrix(c(flabel,"M*W*Z"))}
        }
      }
      jj<-jj+1
    }
    start<-start+i
    highfsz<-ncol(highf)
  }
  #END  F

  if  (nrow(highf)  >  1)
  {
    highf<-matrix(highf[2:nrow(highf),],ncol=highfsz)
    highf2<-matrix(highf2[2:nrow(highf2),],ncol=highfsz)
    flabel<-as.matrix(flabel[2:nrow(flabel),])
    resultm2<-matrix(99999,nrow(highf),maxresm)
    resultm2[1:nrow(highf),1:ncol(highf)]<-highf
    resultm<-rbind(resultm,resultm2)
    highf<-noquote(matrix(sprintf(decimals,highf),ncol=highfsz))
    if  (outscreen==1)
    {
      if  ((i  <  (nms+nys))  |  (ydich==0))
      {
        cat("\nTest(s)  of  highest  order  unconditional  interaction(s):\n")
        colnames(highf)<-matrix(c("R2-chng",hcflab,"df1","df2","p"))
        rownames(highf)<-flabel
        print(highf,right=T)
      }
      if  ((ydich==1)  &  (i==(nms+nys)))
      {
        cat("\nLikelihood  ratio  test  of  highest  order\n")
        cat("unconditional  interaction(s):\n")
        colnames(highf)<-matrix(c("Chi-sq",  "df","p"))
        rownames(highf)<-flabel
        print(highf,right=T)
      }
    }
    intpb<-as.matrix(highf2[,ncol(highf2)])
  }
  intstart<-intstart+numint[1,i]
}
#END  E

#Start  PROBEandPLOT
if  (criterr==0)
{
  threeway<-0;didprint<-0;didsome<-0;sigintct<-0
  #Start  R
  for  (jmed  in  (1:(nms+1)))
  {
```

```
hasw<-0;hasz<-0;jnok<-0;nm1vls<-0;nm2vls<-0;panelgrp<-0
focpred4<-matrix(" ")
intprint<-0;modcat<-0
#Start A
if (jmed <= i)
{
  if ((jmed==1) & ((i+1)==nrow(bcmat)))
  {pathscnt<-pathscnt+1} else {
    paths<-cbind(paths,bcmat[(i+1),jmed])
    pathsw<-cbind(pathsw,wcmat[(i+1),jmed])
    pathsz<-cbind(pathsz,zcmat[(i+1),jmed])
    pathswz<-cbind(pathswz,wzcmat[(i+1),jmed])
    temp<-matrix(fochigh[,pathscnt]*bootloc[,i])
    pathsfoc<-as.matrix(cbind(pathsfoc,pathsfoc[,1]))
    if (jmed==1){pathtype<-cbind(pathtype,1)}
    if ((i+1)==nrow(bcmat)){pathtype<-cbind(pathtype,3)}
    if ((jmed > 1) & ((i+1) < nrow(bcmat))){pathtype<-cbind(pathtype,2)}
    if ((jmed==1) & (nxvls > 1) & (bcmat[(i+1),jmed]==1))
    {pathsfoc[,(pathscn2+1)]<-temp[2:(nxvls+1),1]}
    if ((jmed > 1) | ((jmed==1) & (nxvls==1)))
    {
      temp<-matrix(apply(temp,2,max))    #column max
      pathsfoc[1,(pathscn2+1)]<-temp
    }
    pathscnt<-pathscnt+1;pathscn2<-pathscn2+1
    if (i <= nms){pathsdv<-matrix(c(pathsdv,mnames[1,i]))}
    if (i > nms){pathsdv<-matrix(c(pathsdv,ynames))}
  }
  coeffcol<-coeffcol+1
  probettt<-matrix(coeffs[1:nrow(b),coeffcol])
  if ((jmed==1) & (bcmat[(i+1),jmed]==1))
  {
    omni<-matrix(0,nrow(probettt),nxvls)
    omnitmp<-diag(nxvls)
    omni[2:(1+nxvls),]<-omnitmp
  }
  if (sum(probettt) > 0)
  {
    probvarb<-matrix(999,sum(probettt),sum(probettt))
    probcoef<-matrix(999,sum(probettt),1)
    coefflp2<-1
    for (coefflp in (1:nrow(probettt)))
    {
      if (probettt[coefflp,1]==1)
      {
        probcoef[coefflp2,1]<-b[coefflp,1]
        coefflp2<-coefflp2+1
      }
    }
    coefflp<-0;coefflp2<-0
    for (iclp in (1:nrow(probettt)))
    {
      if (probettt[iclp,1]==1)
      {
        coefflp<-coefflp+1
        coefflp2<-coefflp
        probvarb[coefflp,coefflp]<-varb[iclp,iclp]
        if (iclp < nrow(probettt))
        {
          for (jclp in ((iclp+1):nrow(probettt)))
          {
            if (probettt[jclp,1]==1)
            {
              coefflp2<-coefflp2+1
              probvarb[coefflp,coefflp2]<-varb[iclp, jclp]
              probvarb[coefflp2,coefflp]<-varb[iclp, jclp]
            }
          }
        }
      }
    }
  }
}
#End A
xprobval<-as.matrix(xmodvals)
if ((nxvls > 1) | (mcx > 0)){xprobval<-as.matrix(dummatx[,2:ncol(dummatx)])}
#Start B
if ((wcmat[(i+1),jmed]==1) & (zcmat[(i+1),jmed]==0))
{
  numplps<-1;modvals<-wmodvals;probeval<-wmodvals;wheremv1<-wherexw
  nm1vls<-nwvls;lpstsp<-t(matrix(c(1,1)));modcat<-0;jnmod<-wtmp;jnmodlab<-wnames
  jnok<-1;jnmin<-wmin;jnmax<-wmax;wherejn1<-2
  if (jmed==1)
  {
    wherejn3<-wherexw[1,i]
    if (nxvls > 1){jnok<-0}
  }
  if (jmed > 1)
  {
    wherejn1<-wherem[(jmed-1),i]
    wherejn3<-wheremw[((2*jmed)-3),i]
  }
  if (nwvls > 1)
  {
    probeval<-wprobval
    lpstsp[1,2]<-ncol(probeval)
    modcat<-1;jnok<-0
  }
  if (wdich==1){modcat<-1;jnok<-0}
  problabs<-wnames
  focpred3<-t(matrix(c(wnames,"(W)")))
  hasw<-1
  modgrph<-wnames;intprint<-1;sigintct<-sigintct+1;
  printpbe<-intpb[sigintct,1]
}
#end B
#start C
if ((wcmat[(i+1),jmed]==0) & (zcmat[(i+1),jmed]==1))
{
  numplps<-1;modvals<-zmodvals;probeval<-zmodvals;wheremv1<-wherexz
  nm1vls<-nzvls;lpstsp<-t(matrix(c(1,1)));jnok<-1;jnmod<-ztmp;jnmin<-zmin;jnmax<-zmax
  jnmodlab<-znames;wherejn1<-2
  if (jmed==1)
  {
  wherejn3<-wherexz[1,i]
  if (nxvls > 1){jnok<-0}
  }
  if (jmed > 1){wherejn1<-wherem[(jmed-1),i];wherejn3<-wheremz[((2*jmed)-3),i]}
  if (nzvls > 1)
  {
    probeval<-zprobval;lpstsp[1,2]<-ncol(probeval);modcat<-1;jnok<-0
  }
  if (zdich == 1){modcat<-1;jnok<-0}
  problabs<-znames
  focpred3<-t(matrix(c(znames,"(Z)")))
  modgrph<-znames;hasz<-1;intprint<-1;sigintct<-sigintct+1;
  printpbe<-intpb[sigintct,1]
}
# end C
# start D
if ((wzcmat[(i+1),jmed]==1) |  ((wcmat[(i+1),jmed]==1) & (zcmat[(i+1),jmed]==1)))
{
  numplps<-2;probecnt<-1;intprint<-1
  if (wzcmat[(i+1),jmed]==1)
  {
    sigintct<-sigintct+1;printpbe<-intpb[sigintct,1]
  } else {
    sigintct<-sigintct+2;printpbe<-min(intpb[(sigintct-1):sigintct,1])
  }
  panelgrp<-1;hasw<-1;hasz<-1
  modgrph<-wnames
  lpstsp<-matrix(1,2,2)
  wheremv1<-wherexw;nm1vls<-nwvls;wheremv2<-wherexz;nm2vls<-nzvls;jnok<-0
  if (wzcmat[(i+1),jmed]==1){jnok<-1}
  if (jmed > 1){mprobval<-mmodvals}
  if (jmed==1){if (nxvls > 1){jnok<-0}}
  if (nwvls > 1){lpstsp[1,2]<-ncol(wprobval);modcat<-1;jnok<-0}
  if (zdich==1){modcat<-1;jnok<-0}
  lpstsp[2,1]<-lpstsp[1,2]+1;lpstsp[2,2]<-lpstsp[1,2]+1
  if (nzvls > 1)
  {
    lpstsp[2,1]<-lpstsp[1,2]+1
    lpstsp[2,2]<-lpstsp[1,2]+ncol(zprobval)
    jnok<-0
  }
  if (zdich==1){jnok<-0}
  omni3<-matrix(0,nrow(b),(nxvls*nwvls),0)
  if (jmed > 1){omni3<-matrix(0,nrow(b),nwvls)}
  focpred3<-t(matrix(c(wnames,"(W)")))
  focpred4<-t(matrix(c("         Moderator:", znames, "(Z)")))
  modvals<-matrix(0,(nrow(wmodvals)*nrow(zmodvals)),2)
  probeval<-matrix(0,(nrow(wmodvals)*nrow(zmodvals)),(ncol(wprobval)+ncol(zprobval)))
  for (probei in (1:nrow(wmodvals)))
  {
    for (probej in (1:nrow(zmodvals)))
    {
      modvals[probecnt,1]<-wmodvals[probei,1]
      probeval[probecnt,1:nwvls]<-wprobval[probei,]
      modvals[probecnt,2]<-zmodvals[probej,1]
      probeval[probecnt,(nwvls+1):(nwvls+nzvls)]<-zprobval[probej,]
      probecnt<-probecnt+1
    }
  }
  if (wzcmat[(i+1),jmed]==1)
  {
    numplps<-numplps+1
    probprod<-matrix(0,1,(ncol(wprobval)*ncol(zprobval)))
    lpstsp2<-t(matrix(c(1,1)))
    lpstsp=rbind(lpstsp,lpstsp2)
    lpstsp[3,1]<-lpstsp[2,2]+1
    lpstsp[3,2]<-lpstsp[2,2]+ncol(probprod)
    jnmod<-ztmp;jnmin<-zmin;jnmax<-zmax;jnmodlab<-znames
    if (jmed == 1){wherejn1<-wherexw[1,i];wherejn3<-wherexwz[1,i]}
    if (jmed > 1)
    {
      wherejn1<-wheremw[((2*jmed)-3),i]
      wherejn3<-wheremwz[((2*jmed)-3),i]
    }
    for (probei in (1:nrow(wmodvals)))
    {
      for (probej in (1:nrow(zmodvals)))
      {
        probtemp<-1
        for (probek in (1:ncol(wprobval)))
        {
          #probtemp<-cbind(probtemp,(wprobval[probei,probek]*zprobval[probej,]))
          probtemp<-cbind(probtemp,matrix((wprobval[probei,probek]*zprobval[probej,]),nco
        }
          probprod<-rbind(probprod,probtemp[1,2:ncol(probtemp)])
      }
    }
    probprod<-probprod[2:nrow(probprod),]
    probeval<-cbind(probeval,probprod)
  }
  problabs<-cbind(wnames,znames)
}
# end D
# start E
if (intprint==1)
{
  focpred<-"Focal predictor:"
  if (jmed==1)
  {focpred<-cbind(focpred,xnames,"(X)");focplotv<-as.matrix(xmodvals)}
  if (jmed >1)
  {
    if (nms > 1){focpred<-cbind(focpred,mnames[1,(jmed-1)], medlb2[(jmed-1),1])}
    if (nms==1){focpred<-cbind(focpred,mnames[1,(jmed-1)], "(M)")}
    focplotv<-as.matrix(mmodvals[,(jmed-1)],nrow=nrow(mmodvals))
  }
  focpred2<-cbind("             Moderator:",focpred3)
```

```
focpred<-rbind(focpred,focpred2)
if (ncol(focpred4) > 1){focpred<-rbind(focpred,focpred4);focpred4<-matrix(c(" "))}
if (outscreen==1)
{
  if (((plot==1) | (plot==2)) | (printpbe <= intprobe))
  {cat("----------\n");write.table(focpred,quote=FALSE,row.names=FALSE,col.names=FALSE)
}
foctmp<-matrix(1,nrow(modvals),1)
probexpl<-1
probeva2<-cbind(foctmp,probeval)
if ((jmed==1) & (nxs > 0) & (mcx > 0)){probexpl<-nxvls}
foctmp<-matrix(1,nrow(modvals),1)
modvals3<-matrix(0,1,(6+ncol(problabs)))
probrown<-matrix(0,nrow(probeval),1)
jtmp<-1
for (probei in (1:nrow(probeval))){probrown[probei,1]<-jtmp;jtmp<-jtmp+nxvls}
probrow<-999;modvarl<-problabs

# start F */
if ((plot==1) | (plot==2) | (nxvls > 1))
{
  plotvals<-matrix(999,(nrow(modvals)*nrow(focplotv)),(ncol(modvals)+1))
  for (ploti in (1:nrow(modvals)))
  {
    for (plotj in (1:nrow(focplotv)))
    {
      plotvals[(((ploti-1)*nrow(focplotv))+plotj),2:ncol(plotvals)]<-modvals[ploti,]
      plotvals[(((ploti-1)*nrow(focplotv))+plotj),1]<-focplotv[plotj,1]
    }
  }
  focpredn<-3
  if (jmed==1)
  {
    if (nxvls > 1){focpredn<-(nxvls+1)}
    if ((nxvls==1) & (xdich==1)){focpredn<-2}
  }
  meanmat<-diag(c(means))
  onesmat<-matrix(1,nrow(meanmat),(nrow(probeval)*focpredn))
  probeplt<-t(diag(means)%*%onesmat)
  # start G
  if (jmed==1)
  {
    if ((wcmat[(i+1),1]==1) | (zcmat[(i+1),1]==1))
    {
      plotcnt<-1;iloops<-nwpval*nzpval;plotmx<-nxpval*nzpval
      if ((wcmat[(i+1),1]==1) & (zcmat[(i+1),1]==0))
      {iloops<-nwpval;plotmx<-nxpval}
      if ((wcmat[(i+1),1]==0) & (zcmat[(i+1),1]==1))
      {iloops<-nzpval;plotmx<-nxpval}
      xestvals<-matrix(-999,(nxpval*iloops),ncol(xprobval))
      if (wcmat[(i+1),1]==1){westvals<-matrix(-999,nrow(xestvals),ncol(wprobval))}
      if (zcmat[(i+1),1]==1){zestvals<-matrix(-999,nrow(xestvals),ncol(zprobval))}
      for (ploti in (1:iloops))
      {
        for (plotj in (1:nxpval))
        {xestvals[plotcnt,]<-xprobval[plotj,];plotcnt<-plotcnt+1}
      }
      plotcnt<-1;plotcnt1<-1;plotcnt2<-1;plotcntz<-1
      for (ploti in (1:(iloops*nxpval)))
      {
        if (wcmat[(i+1),1]==1){westvals[ploti,]=wprobval[plotcnt1,]}
        if ((wcmat[(i+1),1]==0) & (zcmat[(i+1),1]==1))
        {zestvals[ploti,]<-zprobval[plotcnt1,]}
        if ((wcmat[(i+1),1]==1) & (zcmat[(i+1),1]==1))
        {zestvals[ploti,]<-zprobval[plotcnt2,];plotcntz=plotcntz+1}
        plotcnt<-plotcnt+1
        if (plotcnt > plotmx){plotcnt<-1;plotcnt1<-plotcnt1+1}
        if (plotcntz > nxpval)
        {
          plotcnt2<-plotcnt2+1;plotcntz<-1
          if (plotcnt2 > nzpval){plotcnt2<-1}
        }
      }
      probeplt[,2:(1+(ncol(xestvals)))]<-xestvals
      if (wcmat[(i+1),1]==1)
      {probeplt[,wherew[1,i]:wherew[2,i]]<-westvals}
      if (zcmat[(i+1),1]==1)
      {probeplt[,wherez[1,i]:wherez[2,i]]<-zestvals}
    }
  }
  # end G
  # start H
  if (jmed > 1)
  {
    if ((wcmat[(i+1),jmed]==1) | (zcmat[(i+1),jmed]==1))
    {
      plotcnt<-1;iloops<-(nwpval*nzpval);plotmx<-(3*nzpval)
      if ((wcmat[(i+1),jmed]==1) & (zcmat[(i+1),jmed]==0))
      {iloops<-nwpval;plotmx<-3}
      if ((wcmat[(i+1),jmed]==0) & (zcmat[(i+1),jmed]==1))
      {iloops<-nzpval;plotmx<-3}
      mestvals<-matrix(-999,(3*iloops),1)
      if (wcmat[(i+1),jmed]==1)
      {westvals<-matrix(-999,nrow(mestvals),ncol(wprobval))}
      if (zcmat[(i+1),jmed]==1)
      {zestvals<-matrix(-999,nrow(mestvals),ncol(zprobval))}
      for (ploti in (1:iloops))
      {
        for (plotj in (1:3))
        {
          mestvals[plotcnt,]<-mprobval[plotj,(jmed-1)]
          plotcnt<-plotcnt+1
        }
      }
      plotcnt<-1;plotcnt1<-1;plotcnt2<-1;plotcntz<-1
      for (ploti in (1:(iloops*3)))
      {
        if (wcmat[(i+1),jmed]==1)
        {westvals[ploti,]<-wprobval[plotcnt1,]}
        if ((wcmat[(i+1),jmed]==0) & (zcmat[(i+1),jmed]==1))
        {zestvals[ploti,]<-zprobval[plotcnt1,]}
        if ((wcmat[(i+1),jmed]==1) & (zcmat[(i+1),jmed]==1))
        {zestvals[ploti,]<-zprobval[plotcnt2,];plotcntz<-plotcntz+1}
        plotcnt<-plotcnt+1
        if (plotcnt > plotmx){plotcnt<-1;plotcnt1<-plotcnt1+1}
        if (plotcntz > 3)
        {
          plotcnt2<-plotcnt2+1;plotcntz<-1
          if (plotcnt2 > nzpval){plotcnt2<-1}
        }
      }
      probeplt[,wherem[(jmed-1),i]]<-mestvals
      if (wcmat[(i+1),jmed]==1)
      {
        if (model != 74)
        {probeplt[,(wherew[1,i]):(wherew[2,i])]<-westvals}
        if (model==74)
        {probeplt[,(wherex[1,i]):(wherex[2,i])]<-westvals}
      }
      if (zcmat[(i+1),jmed]==1)
      {probeplt[,(wherez[1,i]):(wherez[2,i])]<-zestvals}
    }
  }
  # end H
  # Here I am doing the multiplications to produce data for the plot
  prodloop <-1
  if (jmed==1){prodloop<-ncol(xestvals)}
  if ((wcmat[(i+1),jmed])==1)
  {
    plotcnt<-0
    for (ploti in (1:prodloop))
    {
      for (plotj in (1:ncol(westvals)))
      {
        if (jmed==1)
        {probeplt[,(wherexw[1,i]+plotcnt)]<-(xestvals[,ploti]*westvals[,plotj])}
        if (jmed > 1)
        {probeplt[,(wheremw[((jmed*2)-3) ,i]+plotcnt)]<-(mestvals[,ploti]*westvals[,pl
        plotcnt<-plotcnt+1
      }
    }
  }
  if ((zcmat[(i+1),jmed])==1)
  {
    plotcnt<-0
    for (ploti in (1:prodloop))
    {
      for (plotj in (1:ncol(zestvals)))
      {
        if (jmed==1)
        {probeplt[,(wherexz[1,i]+plotcnt)]<-(xestvals[,ploti]*zestvals[,plotj])}
        if (jmed > 1)
        {probeplt[,(wheremz[((jmed*2)-3),i]+plotcnt)]<-(mestvals[,ploti]*zestvals[,plot
        plotcnt<-plotcnt+1
      }
    }
  }
  if ((wzcmat[(i+1),jmed])==1)
  {
    plotcnt<-0;threeway<-1
    for (ploti in (1:ncol(westvals)))
    {
      for (plotj in (1:ncol(zestvals)))
      {
        probeplt[,(wherewz[1,i]+plotcnt)]<-(westvals[,ploti]*zestvals[,plotj])
        plotcnt<-plotcnt+1
      }
    }
    plotcnt<-0
    for (plotk in (1:prodloop))
    {
      for (ploti in (1:ncol(westvals)))
      {
        for (plotj in (1:ncol(zestvals)))
        {
          if (jmed==1)
          {probeplt[,(wherexwz[1,i]+plotcnt)]<-(xestvals[,plotk]*westvals[,ploti]*zestv
          if (jmed > 1)
          {probeplt[,(wheremwz[((jmed*2)-3),i]+plotcnt)]<-(mestvals[,plotk]*westvals[,p
          plotcnt<-plotcnt+1
        }
      }
    }
  }
  # here is where we add holding constant products as needed
  # start I
  for (newplp in (1:i))
  {
    if (newplp != jmed)
    {
      if (wcmat[(i+1),newplp]==1)
      {
        prodloop<-1
        if (newplp==1){prodloop<-nxvls}
        plotcnt<-0
        for (ploti in (1:prodloop))
        {
          for (plotj in (1:nwvls))
          {
            if (newplp==1)
            {probeplt[,(wherexw[1,i]+plotcnt)]<-probeplt[,(1+ploti)]*probeplt[,(wherew[
            if (newplp > 1)
            {
              if (model != 74)
              {probeplt[,(wheremw[((newplp*2)-3) ,i]+plotcnt)]<-probeplt[,wherem[(newp
              if (model==74)
              {probeplt[,(wheremw[((newplp*2)-3) ,i]+plotcnt)]<-probeplt[,wherem[(newp
            }
            plotcnt<-plotcnt+1
          }
        }
      }
      if (zcmat[(i+1),newplp]==1)
      {
        prodloop<-1
        if (newplp==1){prodloop<-nxvls}
        plotcnt<-0
        for (ploti in (1:prodloop))
        {
          for (plotj in (1:nzvls))
          {
            if (newplp==1)
            {probeplt[,(wherexz[1,i]+plotcnt)]<-probeplt[,(1+ploti)]*probeplt[,(wherez[
            if (newplp > 1)
            {probeplt[,(wheremz[((newplp*2)-3),i]+plotcnt)]<-probeplt[,wherem[(newplp-1
            plotcnt<-plotcnt+1
          }
        }
      }
      if (wzcmat[(i+1),newplp]==1)
      {
        plotcnt<-0
        if (threeway==0)
        {
          for (ploti in (1:nwvls))
          {
            for (plotj in (1:nzvls))
            {
              probeplt[,(wherewz[1,i]+plotcnt)]<-probeplt[,(wherew[1,i]+ploti-1)]*probe
              plotcnt<-plotcnt+1
            }
          }
        }
        prodloop<-1
        if (newplp==1){prodloop<-nxvls}
        plotcnt<-0
        for (plotk in (1:prodloop))
        {
          for (ploti in (1:nwvls))
          {
            for (plotj in (1:nzvls))
            {
              if (newplp==1)
              {probeplt[,(wherexwz[1,i]+plotcnt)]<-probeplt[,(1+plotk)]*probeplt[,(wher
              if (newplp > 1)
              {probeplt[,(wheremwz[((newplp*2)-3),i]+plotcnt)]<-probeplt[,wherem[(newpl
              plotcnt<-plotcnt+1
            }
          }
        }
      }
    }
  }
  #END I
  predvals<-probeplt%*%b
```

```
    if ((i==(nms+nys)) & (ydich==1))
    {
      predvalt<-as.numeric(predvals < 709.7)
      prevalt7<-(1-predvalt)*(709.7)
      predvals<-(predvals*predvalt)+prevalt7
      expyhat<-exp(predvals)/(1+exp(predvals))
    }
    sepred<-matrix(999,nrow(plotvals),3)
    for (sei in (1:nrow(plotvals)))
    {
      ask<-as.matrix(probeplt[sei,],ncol=ncol(probeplt))
      sepred[sei,1]<-sqrt(t(ask)%*%varb%*%ask)
      if ((i < (nms+nys)) | (ydich==0))
      {
        sepred[sei,2]<-predvals[sei,1]-tval*sepred[sei,1]
        sepred[sei,3]<-predvals[sei,1]+tval*sepred[sei,1]
      }
      if ((i==(nms+nys)) & (ydich==1))
      {
        sepred[sei,2]<-predvals[sei,1]-xp2*sepred[sei,1]
        sepred[sei,3]<-predvals[sei,1]+xp2*sepred[sei,1]
      }
    }
    prevloc<-ncol(plotvals)+1
    probeplt<-cbind(plotvals,predvals)
    if (plot==2){probeplt<-cbind(probeplt,sepred)}
    if ((i==(nms+nys)) & (ydich==1))
    {probeplt<-cbind(probeplt,expyhat)}
    didsome<-0
}
# END F

# here is the loop that is printing the conditional effects
# this does conditional two way interactions
# START J
if ((wzcmat[(i+1),jmed]==1) & (printpbe <= intprobe))
{
  if (jmed==1)
  {omnilp2<-nxvls*nwvls;omnitmp<-diag(omnilp2);omni3[wherexw[1,i]:wherexw[2,i],]<-omnitmp
  if (jmed>1)
  {omnilp2<-nwvls;omnitmp<-diag(omnilp2);omni3[wheremw[((jmed*2)-3),i]:wheremw[((jmed*2)-
  omnif<-matrix(0,1,4)
  if ((i==(nms+nys)) & (ydich==1))
  {omnif<-matrix(0,1,3)}
  condeff3<-0
  for (omnilp1 in (1:nrow(zprobval)))
  {
    for (omnilp in (1:omnilp2))
    {
      if (jmed==1)
      {
        omni3[(wherexwz[1,i]+((omnilp-1)*nzvls)):(wherexwz[1,i]+((omnilp-1)*nzvls)+(nzvls
      }
      if (jmed > 1)
      {
```

```
      {
        omni3[(wheremwz[((jmed*2)-3),i]+((omnilp-1)*nzvls)):(wheremwz[((jmed*2)-3),i]+((o
      }
    }
    condeff<-t(omni3)%*%b
    condeff3<-rbind(condeff3,condeff)
    fresult2<-process.ftest3(omni3,b,varb,chr=0,brsq2,1,y,x)

    if ((i==(nms+nys)) & (ydich==1))
    {
      fratio<-fresult2[1,1]*ncol(omni3)
      pfr<-(1-pchisq(fratio,df=ncol(omni3)))
      fresult2<-cbind(fratio,ncol(omni3),pfr)
    }
    omnif<-rbind(omnif,fresult2)
  }
  omnif<-matrix(omnif[2:nrow(omnif),],ncol=ncol(omnif))
  clabtmp<-znames
  condeff3<-matrix(condeff3[2:nrow(condeff3),],ncol=ncol(condeff3))
  if ((nxvls*nwvls)==1)
  {
    omnif<-cbind(condeff3,omnif)
    clabtmp<-cbind(clabtmp,"effect")
  }
  omnif<-cbind(zmodvals,omnif)
  if ((i < nms+nys) | (ydich==0))
  {clabtmp<-c(clabtmp,hcflab,"df1","df2","p")}
  if ((i==(nms+nys)) & (ydich==1))
  {clabtmp<-c(clabtmp,"Chi-sq","df","p")}
  resultm2<-matrix(99999,nrow(omnif),maxresm)
  resultm2[1:nrow(omnif),1:ncol(omnif)]<-omnif
  resultm<-rbind(resultm,resultm2)
  omnif2<-noquote(matrix(sprintf(decimals,omnif),nrow=nrow(omnif)))
  colnames(omnif2)<-clabtmp
  rownames(omnif2)<-t(matrix(replicate(nrow(omnif)," ")))
  if (outscreen==1)
  {
    if (jmed==1)
    {cat("\nTest of conditional X*W interaction at value(s) of Z:\n")
    print(omnif2,right=T)}
    if (jmed > 1)
    {cat("\nTest of conditional M*W interaction at value(s) of Z:\n")
    print(omnif2,right=T)}
  }
}
# END J

  # start 0
  for (probei in (1:probexpl))
  {
    if (probexpl > 1)
    {
      foctmp<-matrix(0,nrow(modvals),probexpl)
      foctmp[,probei]<-foctmp[,probei]+1
      probtemp<-matrix(0,nrow(modvals),1)
      for (probem in (1:numlps))
      {for (probek in (1:nxvls))
        {for (probej in (lpstsp[probem,1]:lpstsp[probem,2]))
          {probtemp<-cbind(probtemp,foctmp[,probek]*probeval[,probej])}
        }
      }
      probeva2<-probtemp[,2:ncol(probtemp)]
      probeva2<-cbind(foctmp,probeva2)
    }
    probres<-probeva2%*%probcoef
    probrese<-as.matrix(sqrt(diag(probeva2%*%probvarb%*%t(probeva2))))
    tratio<-probres/probrese
    if ((ydich==1) & (i==(nms+nys)))
    {
      p<-2*(1-pnorm(abs(tratio)))
    } else {
      p<-2*pt(-abs(tratio),df=dfres)
    }
    modvals2<-cbind(modvals,probres,probrese,tratio, p)
    if ((i < nms+nys) | (ydich==0))
    {
      modvals2<-cbind(modvals2,(probres-as.numeric(tval)*probrese),(probres+as.numeric(tv
      problabs<-cbind(problabs,"effect",hclab,"t", "p", "LLCI", "ULCI")
    }
    if ((ydich==1) & (i==(nms+nys)))
    {
      modvals2<-cbind(modvals2,(probres-xp2*probrese),(probres+xp2*probrese))
      problabs<-cbind(problabs,"effect","se","Z", "p", "LLCI", "ULCI")
    }
    # start L
    if ((probexpl > 1) & (printpbe <= intprobe))
    {
      if (hasz==1){printz<-1}
      if (hasw==1){printw<-1}
      probrlab<-matrix(xcatlab[probei,1],nrow(modvals),1)
      modvals3<-rbind(modvals3,modvals2)
      probrow<-rbind(probrow,probrown)
      probrown<-probrown+1
      if (probei==probexpl)
      {
        xproblab<-xcatlab[1:nxvls,1]
        probrow<-matrix(probrow[2:nrow(probrow),1])
        modvals3<-modvals3[2:nrow(modvals3),]
        #modvals3<-modvals3[order(modvals3[,1]),]
        modvals3<-modvals3[order(probrow[,1]),]
        start2<-1
        problabs<-problabs[1,(1+(ncol(modvar1))):ncol(problabs)]
        pstart<-1
        # start K
        for (probek in (1:nrow(probeval)))
        {
          endstart<-start2+(nxvls-1)
          temp<-modvals3[start2:endstart,(1+ncol(modvar1)):ncol(modvals3)]
          temp2<-t(modvals3[start2:start2,1:ncol(modvar1)])
          trnames<-t(modvar1)
          if (outscreen==1)
          {
            if (probek > 1){cat("----------\n")}
            if (probek==1)
            {
              cat("\nConditional effects of the focal predictor at values of the
              if ((jmed==1) & (i ==(nms+nys)) & (nms > 0))
              {
                if (nxvls==1)
                {cat("\n(These are also the conditional direct effects of X on
                if (nxvls != 1)
                {cat("\n(These are also the relative conditional direct effects o
              }
              cat("\n")
            }
            cat("Moderator value(s):\n")
          }
          resultm2<-matrix(99999,nrow(temp2),maxresm)
          resultm2[1:nrow(temp2),1:ncol(temp2)]<-temp2
          resultm<-rbind(resultm,resultm2)
          temp2<-noquote(matrix(sprintf(decimals,temp2),nrow=nrow(temp2)))
          temp3<-matrix(temp2[1,1:ncol(temp2)])
  rownames(temp3)<-trnames
          colnames(temp3)<-" "
          if (outscreen==1)
          {print(temp3,right=TRUE,quote=FALSE)
            cat("\n")}
          #for (mdpntr in (1:ncol(temp2)))
          #{
          #write.table(temp2[1,mdpntr],quote=FALSE,row.names=trnames[mdpntr,1],col.names=
          #}
          resultm2<-matrix(99999,nrow(temp),maxresm)
          resultm2[1:nrow(temp),1:ncol(temp)]<-temp
          resultm<-rbind(resultm,resultm2)
          temp<-noquote(matrix(sprintf(decimals,temp),nrow=nrow(temp)))
          colnames(temp)<-problabs[1:ncol(temp)]
          rownames(temp)<-xproblab
          if (outscreen==1)
          {print(temp,right=T)}
          start2<-start2+nxvls
          didsome<-1
          if (jmed==1)
          {
            mod1val<-probeval[probek,1:nm1vls]
            for (omnilp in (1:nxvls))
            {
              omni[(wheremv1[1,i]+((omnilp-1)*nm1vls)):(wheremv1[1,i]+((omnilp-1)*nm1vls)
              if (nm1vls < ncol(probeval))
              {
                mod2val<-probeval[probek,(nm1vls+1):(nm1vls+nm2vls)]
                omni[(wheremv2[1,i]+((omnilp-1)*nm2vls)):(wheremv2[1,i]+((omnilp-1)*nm2vl
                if ((nm1vls+nm2vls) < ncol(probeval))
                {
                  intlen<-(nm1vls*nm2vls)
                  modintvl<-probeval[probek,(nm1vls+nm2vls+1):ncol(probeval)]
                  omni[(wheremwz[1,i]+((omnilp-1)*intlen)):(wheremwz[1,i]+((omnilp-1)*int
                  omnii[(wherexwz[1,1]+((omniip-1)*intlen)):(wherexwz[1,1]+((omniip-1)*int
                }
              }
            }
            fresult2<-process.ftest3(omni,b,varb,0,brsq2,1,y,x)
            if  ((i  ==  (nms+nys))  &  (ydich==1))
            {
              fratio<-fresult2[1,1]*nxvls
              pfr<-(1-pchisq(fratio,df=nxvls))
              fresult2<-cbind(fratio,nxvls,pfr)
            }
            resultm2<-matrix(99999,nrow(fresult2),maxresm)
            resultm2[1:nrow(fresult2),1:ncol(fresult2)]<-fresult2
            resultm<-rbind(resultm,resultm2)
            fresult2<-noquote(matrix(sprintf(decimals,fresult2),nrow=nrow(fresult2)))
            if  (outscreen==1)
            {
              if  (i  <  (nms  +  nys)  |  (ydich  ==  0))
              {
                cat("\nTest  of  equality  of  conditional  means\n")
                colnames(fresult2)<-c(hcflab,"df1","df2","p")
                rownames(fresult2)<-"  "
                print(fresult2,right=T)
              }
              if  ((i==(nms  +  nys))  &  (ydich==1))
              {
                cat("\nTest  of  equality  of  conditional  logits  or  probabilities\n")
                rownames(fresult2)<-"  "
                colnames(fresult2)<-c("Chi-sq","df","p")
                print(fresult2,right=T)
              }
            }
            probetmp<-probeplt[pstart:(pstart+nxvls),1]
            probetmp<-cbind(probetmp,probeplt[pstart:(pstart+nxvls),prevloc:ncol(probeplt
            pstart<-pstart+(nxvls+1)
            resultm2<-matrix(99999,nrow(probetmp),maxresm)
            resultm2[1:nrow(probetmp),1:ncol(probetmp)]<-probetmp
            resultm<-rbind(resultm,resultm2)
            probetmp<-noquote(matrix(sprintf(decimals,probetmp),nrow=nrow(probetmp)))
            if  (outscreen==1)
            {
              if  (i  <  ((nms  +  nys))  |  (ydich==0))
              {
                clabtmp<-cbind(xnames,  outnames[i,1],  hclab,  "LLCI",  "ULCI")
                cat("\nEstimated  conditional  means  being  compared:\n")
                colnames(probetmp)<-clabtmp[1:ncol(probetmp)]
                rownames(probetmp)<-t(matrix(replicate(nrow(probetmp),"  ")))
                print(probetmp,right=T)
              }
              if  ((i==(nms+nys))  &  (ydich==1))
              {
                cat("\nEstimated  conditional  logits  and  probabilities:\n")
                clabtmp<-c(xnames,  outnames[i,1],  "prob")
                probetm2<-matrix(probetmp[,1:2],ncol=2)
                probetm2<-cbind(probetm2,probetmp[,ncol(probetmp)])
                probetm2<-noquote(probetm2)
```

```
            probetm2<-noquote(probetm2)
            colnames(probetm2)<-clabtmp[1:ncol(probetm2)]
            rownames(probetm2)<-t(matrix(replicate(nrow(probetm2)," ")))
            print(probetm2,right=T)
          }
        }
      }
    }
    # end K
  }
}
# end L
# start N
if ((probexpl==1) & (printpbe <= intprobe))
{
  resultm2<-matrix(99999,nrow(modvals2),maxresm)
  resultm2[1:nrow(modvals2),1:ncol(modvals2)]<-modvals2
  resultm<-rbind(resultm,resultm2)
  modva2<-noquote(matrix(sprintf(decimals,modvals2),nrow=nrow(modvals2)))
  colnames(modva2)<-problabs[1:ncol(modva2)]
  rownames(modva2)<-t(matrix(replicate(nrow(modva2)," ")))
  if (outscreen==1)
  {cat("\nConditional effects of the focal predictor at values of the moder
  print(modva2,right=T)}
  didsome<-1
  if (hasz==1){printz<-1}
  if (hasw==1){printw<-1}
  # start M (JN method)
  if ((jn==1) & (jnok==1))
  {
    if (criterr==0)
    {
      dfres<-(n-nrow(b));
      jncrit<-(dfres* (exp((dfres-(5/6))*((xp2/(dfres-(2/3)+(.11/dfres)))*(xp2/(dfre
      if ((i ==(nms+nys)) & (ydich==1)){jncrit<-xp2*xp2}
      jnb1<-b[wherejn1,1]
      jnb3<-b[wherejn3,1]
      jnsb1<-varb[wherejn1,wherejn1]
      jnsb3<-varb[wherejn3,wherejn3]
      jnsb1b3<-varb[wherejn1,wherejn3]
      ajn<-(jncrit*jnsb3)-(jnb3*jnb3)
      bjn<-2*((jncrit*jnsb1b3)-(jnb1*jnb3))
      cjn<-((jncrit*jnsb1)-(jnb1*jnb1))
      radarg<-(bjn*bjn)-(4*ajn*cjn)
      den<-2*ajn
      nrts<-0
      if ((radarg >= 0) & (den != 0))
      {
        x21<-(-bjn+sqrt(radarg))/den
        x22<-(-bjn-sqrt(radarg))/den
        roots<-as.matrix(0)
        if ((x21 >= jnmin) & (x21 <= jnmax))
        {nrts<-1;roots<-rbind(roots,x21)}
        if ((x22 >= jnmin) & (x22 <= jnmax))
        {nrts<-nrts+1;roots<-rbind(roots,x22)}
        roots<-cbind(roots,matrix(0,nrow(roots),2))
        roots<-cbind(roots,matrix(0,nrow(roots),2))
      }
      if (nrts > 0)
  {
        roots<-matrix(roots[2:nrow(roots),1:3],ncol=3)
        roots[1,2]<-sum((as.numeric(jnmod < roots[1,1]))/n)*100
        roots[1,3]<-sum((as.numeric(jnmod > roots[1,1]))/n)*100
        if (nrow(roots)==2)
        {
          roots[2,2]<-sum((as.numeric(jnmod < roots[2,1]))/n)*100
          roots[2,3]<-sum((as.numeric(jnmod > roots[2,1]))/n)*100
        }
        resultm2<-matrix(99999,nrow(roots),maxresm)
        resultm2[1:nrow(roots),1:ncol(roots)]<-roots
        resultm<-rbind(resultm,resultm2)
        roots2<-noquote(matrix(sprintf(decimals,roots),nrow=nrow(roots)))
        colnames(roots2)<-c("Value", "% below", "% above")
        rownames(roots2)<-t(matrix(replicate(nrow(roots)," ")))
        if (outscreen==1)
        {cat("\nModerator value(s) defining Johnson-Neyman significance region(s
        print(roots2,right=T)}
      }
      if (nrts==0)
      {
        if (outscreen==1)
        {cat("\nThere are no statistical significance transition points within
        cat("range of the moderator found using the Johnson-Neyman method.\n"
      }
      jnvals<-matrix(0,(21+nrts),7)
      for (jni in (0:20))
      {jnvals[(jni+1),1]<-jnmin+(jni*((jnmax-jnmin)/20))}
      if (nrts > 0)
      {
        for (jni in (1:nrts))
        {
          for (jnj in (2:nrow(jnvals)))
          {
            if ((roots[jni,1] > jnvals[(jnj-1),1]) & (roots[jni,1] < jnvals[jn
            {
              jnvals[(jnj+1):(21+jni),1]<-jnvals[jnj:(20+jni),1]
              jnvals[jnj,1]<-roots[jni,1]
            }
          }
        }
      }
      for (jni in (1:nrow(jnvals)))
      {
        jnvals[jni,2]<-jnb1+jnb3*jnvals[jni,1]
        jnvals[jni,3]<-sqrt(jnsb1+2*jnvals[jni,1]*jnsb1b3+(jnvals[jni,1]*jnvals[jni,1
        jnvals[jni,4]<-jnvals[jni,2]/jnvals[jni,3]
        jnvals[jni,5]<-2*pt(-abs(jnvals[jni,4]), df=dfres)
        jnvals[jni,6]<-jnvals[jni,2]-sqrt(jncrit)*jnvals[jni,3]
        jnvals[jni,7]<-jnvals[jni,2]+sqrt(jncrit)*jnvals[jni,3]
        if ((i==(nms + nys)) & (ydich==1))
        {
          jnvals[jni,5]<-2*(1-pnorm(abs(jnvals[jni,4])))
```

```
                jnvals[jni,5]<-2*(1-pnorm(abs(jnvals[jni,4])))
                jnvals[jni,6]<-jnvals[jni,2]-xp2*jnvals[jni,3]
                jnvals[jni,7]<-jnvals[jni,2]+xp2*jnvals[jni,3]
              }
            }
            resultm2<-matrix(99999,nrow(jnvals),maxresm)
            resultm2[1:nrow(jnvals),1:ncol(jnvals)]<-jnvals
            resultm<-rbind(resultm,resultm2)
            jnvals2<-noquote(matrix(sprintf(decimals,jnvals),nrow=nrow(jnvals)))
            rownames(jnvals2)<-t(matrix(replicate(nrow(jnvals)," ")))
            if ((i < nms+nys) | (ydich==0))
            {colnames(jnvals2)<-c(jnmodlab,"effect", hclab, "t", "p", "LLCI", "ULCI")}
            if ((i==(nms+nys)) & (ydich==1))
            {colnames(jnvals2)<-c(jnmodlab,"effect","se","Z", "p", "LLCI", "ULCI")}
            if (outscreen==1)
            {
              if (((wcmat[(i+1),jmed]==1) | (zcmat[(i+1),jmed]==1)) & (wzcmat[(i+1),jm
              {cat("\nConditional effect of focal predictor at values of the moder
              if ((jmed==1) & (wzcmat[(i+1),jmed]==1))
              {cat("\nConditional X*W interaction at values of the moderator Z:\n")
              if ((jmed > 1) & (wzcmat[(i+1),jmed]==1))
              {cat("\nConditional M*W interaction at values of the moderator Z:\n")
            }
          }
        }
        # END M (JN method)
      }
      if ((i==(nms+nys)) & (jmed==1) & (bcmat[nrow(bcmat),1]==1))
      {
        if (probei==1)
        {direfflb<-problabs;direff<-modvals2}
        if (probei>1)
        {direff<-rbind(direff,modvals2)}
      }
      intprint<-0
#    This does the contrast for conditional effects */
      if ((jmed==1) & (i==1) & (nms==0) & (modcok==1))
      {
        contvec2=matrix(1,2,1)
        contvec2<-cbind(contvec2,wcontval,zcontval)
        if (wzcmat[(i+1),jmed]==1)
        {
          for (conti in (1:ncol(wcontval)))
          {
            for (contj in (1:ncol(zcontval)))
            {contvec2<-cbind(contvec2,wcontval[,conti]*zcontval[,contj])}
          }
        }
        conteff<-contvec2%*%probcoef
        contdiff<-matrix(contvec2[1,]-contvec2[2,])
        contse<-sqrt(t(contdiff)%*%probvarb%*%contdiff)
        conteffd<-conteff[1,1]-conteff[2,1]
        contvec=cbind(contvec,conteff)
        contvecm<-contvec
        resultm2<-matrix(99999,nrow(contvecm),maxresm)
        resultm2[1:nrow(contvecm),1:ncol(contvecm)]<-contvecm
```

```
            resultm2[1:nrow(contvecm),1:ncol(contvecm)]<-contvecm
            resultm<-rbind(resultm,resultm2)
            contvecm<-noquote(matrix(sprintf(decimals,contvecm),nrow=nrow(contvecm)))
            rownames(contvecm)<-c("Effect1:","Effect2:")
            colnames(contvecm)<-problabs[1:3]
            if (outscreen==1)
            {cat("\nContrast between conditional effects of X:\n")
            print(contvecm,right=T)}
            if (ydich==0)
            {
              p<-2*pt(-abs(conteffd/contse),df=dfres)
              contvec<-cbind(conteffd,contse,conteffd/contse, p)
              contvec<-cbind(contvec,(conteffd-(tval*contse)))
              contvec<-cbind(contvec,(conteffd+(tval*contse)))
              contlabs<-c("Contrast", hclab, "t", "p", "LLCI", "ULCI")
            }
            if (ydich==1)
            {
              p<-2*(1-pnorm(abs(conteffd/contse)))
              contvec<-cbind(conteffd,contse,conteffd/contse, p)
              contvec<-cbind(contvec,(conteffd-(xp2*contse)))
              contvec<-cbind(contvec,(conteffd+(xp2*contse)))
              contlabs<-c("Contrast", "se", "Z", "p", "LLCI", "ULCI")
            }
            resultm2<-matrix(99999,nrow(contvec),maxresm)
            resultm2[1:nrow(contvec),1:ncol(contvec)]<-contvec
            resultm<-rbind(resultm,resultm2)
            contvec<-noquote(matrix(sprintf(decimals,contvec),nrow=nrow(contvec)))
            colnames(contvec)<-contlabs
            rownames(contvec)<-" "
            if (outscreen==1)
            {cat("\nTest of Effect1 minus Effect2\n")
            print(contvec,right=T)}
          }
        }
        # end 0

if ((plot==1) | (plot==2))
        {
          datalabs<-cbind(t(focpred[,2]),outnames[i,1])
          if (plot==2)
          {datalabs<-c(datalabs,"se", "LLCI", "ULCI")}
          if ((i==(nms+nys)) & (ydich==1))
          {datalabs<-c(datalabs,"prob")}
          resultm2<-matrix(99999,nrow(probeplt),maxresm)
          resultm2[1:nrow(probeplt),1:ncol(probeplt)]<-probeplt
          resultm<-rbind(resultm,resultm2)
          probepnt<-noquote(matrix(sprintf(decimals,probeplt),nrow=nrow(probeplt)))
          colnames(probepnt)<-datalabs
          rownames(probepnt)<-t(matrix(replicate(nrow(probepnt)," ")))
          if (outscreen==1)
          {cat("\nData for visualizing the conditional effect of the focal predictor:\
          print(probepnt,right=T)}
        }
      }
      #End E
      #End L
    }
    #End R
  }
  #End PROBEandPLOT
}
#END G LOOP
if ((criterr==0) & (dototal==1))
{
  if (outscreen==1)
  {
    cat("\n************************ TOTAL EFFECT MODEL *************************** \n")
    cat("Outcome Variable: ")
    write.table(outnames[nrow(outnames),1],quote=FALSE,row.names=FALSE,col.names=FALSE)
    cat("\n")
  }
  x<-xtmp
  vlabsm<-matrix(c("constant",xcatlab[1:nxvls,1]))
  if (ncs > 0)
  {
    x<-cbind(as.matrix(x),as.matrix(ctmp))
    vlabsm<-matrix(c(vlabsm,t(covnames)))
  }
  x<-cbind(ones,x)
  modoutz<-process.modelest(y,x,1,1,xp2,hc)
  modsum<-matrix(unlist(modoutz[3]))
  resultm2<-matrix(99999,1,maxresm)
  resultm2[1,1:nrow(modsum)]<-t(modsum)
  resultm<-rbind(resultm,resultm2)
  modsum<-noquote(matrix(sprintf(decimals,modsum),nrow=1))
  modres<-matrix(unlist(modoutz[1]),nrow=ncol(x))
  toteff<-matrix(modres,nrow=ncol(x))
  toteff<-matrix(toteff[2:(1+nxvls),],nrow=nxvls)
  resultm2<-matrix(99999,nrow(modres),maxresm)
  resultm2[1:nrow(modres),1:ncol(modres)]<-modres
  resultm<-rbind(resultm,resultm2)
  modres<-noquote(matrix(sprintf(decimals,modres),nrow=ncol(x)))
  varb<-matrix(unlist(modoutz[6]),nrow=ncol(x))
  brsq2<-modsum[1,2]
  b<-matrix(unlist(modoutz[5]))
  colnames(modsum)<-modsuml;rownames(modsum)<-" "
  rownames(modres)<-vlabsm;colnames(modres)<-modresl
  if (outscreen==1)
  {
    cat("Model Summary: \n")
    print(modsum,right=T)
    cat("\n")
    cat("Model: \n")
    print(modres,right=T)
  }
  totefflb<-modresl
  toteffl2<-vlabsm[2:(1+nxvls),]
  lmat<-matrix(0,nrow(b),1)
  lmat2<-matrix(1,nxvls,1)
  lmat[2:(1+nxvls),1]<-lmat2
  if (vdich != 1)
```

```
      {totomni<-process.ftest3(lmat,b,varb,1,brsq2,0,y,x)}
      if (stand==1)
      {
       predsd<-matrix(0,nrow(modres),1)
       stdmod<-as.numeric(modres[,1])/ovsd[nrow(ovsd),1]
       for (jd in c(1:ncol(x)))
       {predsd[jd,1]<-sd(x[,jd])}
       if ((wherex[1,ncol(wherex)] != -999) & ((nxvls > 1) | (xdich==1)))
       {
        sdmsone<-matrix(1,nxvls,1)
        predsd[wherex[1,ncol(wherex)]:wherex[2,ncol(wherex)],1]<-sdmsone
        pstog<-1
       }
       predsd[1,1]<-1
       stdmod<-stdmod*predsd
       stdmod<-matrix(stdmod[2:nrow(stdmod),1])
       sdvlabs<-vlabsm[2:nrow(vlabsm),1]
       resultm2<-matrix(99999,nrow(stdmod),maxresm)
       resultm2[1:nrow(stdmod),1:ncol(stdmod)]<-stdmod
       resultm<-rbind(resultm,resultm2)
       stdmod<-noquote(matrix(sprintf(decimals,stdmod,nrow(b)-1)))
       colnames(stdmod)<-"coeff"
       rownames(stdmod)<-sdvlabs
       if (outscreen==1)
       {cat("\nStandardized coefficients:\n")
        print(stdmod,right=T)}
      }
      if (covcoeff==1)
      {
       varbpr<-noquote(matrix(sprintf(decimals,varb),nrow=nrow(b)))
       rownames(varbpr)<-vlabsm;colnames(varbpr)<-vlabsm
       if (outscreen==1)
       {cat("\nCovariance matrix of regression parameter estimates:\n")
       print(varbpr,right=T)}
      }
     }
    }
    #END cycle through the models

    if ((criterr==0) & (nms > 0) & (ydich==0) & (modelres==1))
    {
     modresid<-matrix(modresid[,2:ncol(modresid)],ncol=(ncol(modresid)-1))
     corall<-cor(modresid)
     resultm2<-matrix(99999,nrow(corall),maxresm)
     resultm2[1:nrow(corall),1:ncol(corall)]<-corall
     resultm<-rbind(resultm,resultm2)
     corall2<-noquote(matrix(sprintf(decimals,corall),nrow=nrow(corall)))
     colnames(corall2)<-outnames
     rownames(corall2)<-outnames
     if (outscreen==1)
     {cat("\n*************** CORRELATIONS BETWEEN MODEL RESIDUALS *************** \n\n")
     print(corall2,right=T)}
    }

    #DO BOOTSTRAPPING
```

```
  if ((criterr==0) & (boot > 0))
  {
    bootres<-matrix(-999,1,sum(nump))
    if (effsize==1){bootysd<-matrix(-999,1,1);bootxsd<-matrix(-999,1,1)}
    badboot<-0;goodboot<-0;smallest<-1;booting<-1
    j<-1
    if (outscreen==1)
    {
      cat("\n*************************************************************************** \n")
      if (progress==1)
      {cat("Bootstrapping progress:\n")
      bootprog<-txtProgressBar(min=0,max=boot,char=">",width=62,style=3)}
      if (progress !=1){cat("Bootstrapping in progress. Please wait.\n");flush.console()}
    }
    while ((goodboot < boot) & (j <= maxboot))
    {
      if ((outscreen==1) & (progress==1)){setTxtProgressBar(bootprog,j)}
      nobootx<-1;modres2<-999;bad<-0;
      v<-as.matrix(trunc(runif(n)*n)+1)
      for (i in (1:(nms+nys)))
      {
        y<-as.matrix(outvars[v,i])
        ynovar<-sum((y-(sum(y)/nrow(y)))*(y-(sum(y)/nrow(y))))
        if (ynovar==0){bad<-1}
        xindx<-datindx[1:(nump[1,i]-1),i]
        hello<-0
        x<-as.matrix(fulldat[v,xindx])
        x<-cbind(ones,x)
        xsq<-t(x)%*%x
        exsq<-eigen(xsq)
        exsq<-matrix(unlist(exsq[1]))
        holymoly<-min(exsq)
        zeroeig<-sum(as.numeric(exsq <= 0.000000000002))
        bad<-bad+as.numeric(zeroeig > 0)
        bad=bad+as.numeric(sd(y)==0)
        if (bad==0)
        {
          if (holymoly < smallest){smallest<-holymoly}
          if ((ydich==0) | (i < (nms+nys)))
          {modrest<-process.modelest(y,x,type=1,full=0,xp2,hc)}
          if ((ydich==1) & (i==(nms+nys)))
          {modrest<-process.modelest(y,x,3,0,xp2,hc,iterate,converge)}
          modres2<-as.matrix(c(modres2,modrest))
          if ((bcmat[(i+1),1]==1) & (nobootx==1) & (effsize==1))
          {
            nobootx<-0
            xsdtemp<-sum((x[,2]-(sum(x[,2])/nrow(x)))*(x[,2]-(sum(x[,2])/nrow(x))))
            xsdtemp<-sqrt(xsdtemp/(nrow(x)-1))
          }
        }
      }
      if (bad==0)
      {
        modres2<-t(modres2[2:nrow(modres2),])
        #modres2<-matrix(modres2[,2:nrow(modres2),ncol=ncol(modres)-1)
        bootres<-rbind(bootres,modres2)
        if (effsize==1)
        {
          ysdtemp<-sd(y)
          bootysd<-rbind(bootysd,ysdtemp)
          bootxsd<-rbind(bootxsd,xsdtemp)
        }
        goodboot<-goodboot+1
      }
      if (bad != 0){badboot<-badboot+1}
      j<-j+1
    }
    bootres<-as.matrix(bootres[2:nrow(bootres),])
    if (effsize==1)
    {
      bootysd<-matrix(bootysd[2:nrow(bootysd),])
      if (nrow(bootxsd) > 1){bootxsd<-matrix(bootxsd[2:nrow(bootxsd)])}
    }
    if (goodboot < (boot))
    {boot<-0;modelbt<-0;notecode[notes,1]<-7;notes<-notes+1}
    if (boot > 0)
    {
      if (effsize==1){bootysd<-rbind(ysd,bootysd);bootxsd<-rbind(xsd,bootxsd)}
      if (saveboot==1)
      {
        savlabs<-matrix(" ",ncol(bootres))
        for (i in c(1:ncol(bootres))){savlabs[i,1]<-paste("col",i,sep='')}
        boots<-as.data.frame(bootres)
        colnames(boots)<-savlabs
        #assign("process.boots", boots, envir = .GlobalEnv)
      }
      if (modelbt==1)
      {
        bootcim<-matrix(-99999,ncol(bootres),5)
        bootcim[,2]<-t(colSums(bootres)/nrow(bootres))
        bootcim[,1]<-as.numeric(coeffmat[2:nrow(coeffmat),1])
        for (i in (1:ncol(bootres)))
        {
          if (bc==0){bootcim[i,3:5]<-t(process.pboot3(bootres[,i],cilow,cihigh))}
          if (bc==1)
          {
            bcbout<-process.bcboot3(bootres[,i],bootcim[i,1],xp2,badend,priorlo,priorhi)
            bootcim[i,3:5]<-t(matrix(unlist(bcbout[1])))
            badend<-unlist(bcbout[2]);priorlo<-unlist(bcbout[3]);priorhi<-unlist(bcbout[4])
          }
        }
      }
    }

    if (badboot > 0){notecode[notes,1]<-6;notes<-notes+1}
    if ((outscreen==1) & (progress==1)){close(bootprog)}
  }
  #end of bootstrapping

  #Here is for the indirect effects
```

```
if ((criterr==0) & (nms > 0))
{
  paths<-matrix(paths[,2:ncol(paths)],nrow=nrow(paths))
  pathsw<-matrix(pathsw[,2:ncol(pathsw)],nrow=nrow(pathsw))
  pathsz<-matrix(pathsz[,2:ncol(pathsz)],nrow=nrow(pathsz))
  pathswz<-matrix(pathswz[,2:ncol(pathswz)],nrow=nrow(pathswz))
  pathsmod<-pathsw+pathsz+pathswz
  pathsdv<-t(pathsdv[2:nrow(pathsdv),])
  pathsfoc<-matrix(pathsfoc[,2:ncol(pathsfoc)],nrow=nrow(pathsfoc))
  pathtype<-matrix(pathtype[,2:ncol(pathtype)],nrow=nrow(pathtype))
  anymod<-as.numeric(sum(pathsmod) > 0)
  obscoeff<-t(as.matrix(obscoeff[1,2:ncol(obscoeff)]))
  if ((dototal==0) & (outscreen==1))
  {cat("\n**************** DIRECT AND INDIRECT EFFECTS OF X ON Y ****************\n")}
  if (dototal==1)
  {
    if (outscreen==1)
    {cat("\n************ TOTAL, DIRECT, AND INDIRECT EFFECTS OF X ON Y ************\n")
    totefflb[1,1]<-"effect"
    if (counterf != 1)
    {
      if (effsize==1)
      {
        toteffsz<-toteff[,1]/ysd
        totefflb<-cbind(totefflb,"c_ps")
        if ((xdich==0) & (mcx==0))
        {toteffsz<-cbind(toteffsz,(toteffsz*xsd));totefflb<-cbind(totefflb,"c_cs")}
        toteff<-cbind(toteff,toteffsz)
      }
      resultm2<-matrix(99999,nrow(toteff),maxresm)
      resultm2[1:nrow(toteff),1:ncol(toteff)]<-toteff
      resultm<-rbind(resultm,resultm2)
      toteff2<-noquote(matrix(sprintf(decimals,toteff),nrow=nrow(toteff)))
      colnames(toteff2)<-totefflb
      if (nxvls > 1)
      {
        rownames(toteff2)<-toteffl2
        if (outscreen==1)
        {cat("\nRelative total effects of X on Y:\n")
        print(toteff2,right=T)}
        clabtmp<-c("R2-chng", hcflab, "df1","df2","p")
        resultm2<-matrix(99999,nrow(totomni),maxresm)
        resultm2[1:nrow(totomni),1:ncol(totomni)]<-totomni
        resultm<-rbind(resultm,resultm2)
        totomni2<-noquote(matrix(sprintf(decimals,totomni),nrow=nrow(totomni)))
        colnames(totomni2)<-clabtmp
        rownames(totomni2)<-" "
        if (outscreen==1)
        {
          cat("\nOmnibus test of total effect of X on Y:\n")
          print(totomni2,right=T)
          cat("----------\n")
        }
      }
      if (nxvls < 2)
      {
        rownames(toteff2)<-"  "
        if  (outscreen==1)
        {cat("\nTotal  effect  of  X  on  Y:\n")
        print(toteff2,right=T)}
      }
    }
  }
  moddir<-wcmat[nrow(bcmat),1]+zcmat[nrow(bcmat),1]
  if  ((bcmat[nrow(bcmat),1]==1)  &  (counterf  !=  1))
  {
    if  (ydich==1)
    {
      direfflb[,(ncol(direfflb)-5):ncol(direfflb)]<-t(matrix(c("Effect","se","Z","p","LLCI","ULCI
    }
    if  (moddir==0){direfflb[1,1]="effect"}
    if  ((effsize==1)  &  (moddir==0)  &  (anymod  ==  0))
    {
      direffsz<-direff[,1]/ysd
      direfflb<-cbind(direfflb,"c'_ps")
      if  ((xdich==0)  &  (mcx==0))
      {direffsz<-cbind(direffsz,  (direffsz*xsd));direfflb<-cbind(direfflb,"c'_cs")}
      direff<-cbind(direff,direffsz)
    }
    resultm2<-matrix(99999,nrow(direff),maxresm)
    resultm2[1:nrow(direff),1:ncol(direff)]<-direff
    resultm<-rbind(resultm,resultm2)
    direff2<-noquote(matrix(sprintf(decimals,direff),nrow=nrow(direff)))
    colnames(direff2)<-direfflb
    if  ((moddir==0)  &  (nxvls==1)  &  (outscreen==1))
    {
      rownames(direff2)<-"  "
      cat("\nDirect  effect  of  X  on  Y:\n")
      print(direff2,right=T)
    }
    if  ((moddir==0)  &  (nxvls>1))
    {
      rownames(direff2)<-direffl2
      if  (outscreen==1)
      {cat("\nRelative  direct  effects  of  X  on  Y:\n")
      print(direff2,right=T)}
      resultm2<-matrix(99999,nrow(diromni),maxresm)
      resultm2[1:nrow(diromni),1:ncol(diromni)]<-diromni
      resultm<-rbind(resultm,resultm2)
      diromni2<-noquote(matrix(sprintf(decimals,diromni),nrow=nrow(diromni)))
      rownames(diromni2)<-"  "
      if  (ydich==0)
      {
        colnames(diromni2)<-c("R2-chng",  hcflab,  "df1","df2","p")
        if  (outscreen==1)
        {cat("\nOmnibus  test  of  direct  effect  of  X  on  Y:\n")
        print(diromni2,right=T)}
      }
      if  (ydich==1)
      {
```

```
   colnames(diromni2)<-c("Chi-sq", "df", "p")
   if (outscreen==1)
   {cat("\nOmnibus likelihood ratio test of direct effect of X on Y:\n")
   print(diromni2,right=T)}
  }
  if (outscreen==1)
  {cat("\n----------\n")}
 }
 if ((moddir > 0) & (nxvls==1) & (outscreen==1))
 {cat("\nConditional direct effect(s) of X on Y:\n")
  rownames(direff2)<-t(matrix(replicate(nrow(direff2)," ")))
  print(direff2,right=T)}
 direffl4<-direffl2
 if ((moddir > 0) & (nxvls>1))
 {
  direffl2=" "
  for (i in (1:nxvls))
  {
   for (j in (1:(nrow(direff)/nxvls)))
   {direffl2<-rbind(direffl2,xcatlab[i,1])}
  }
  direffl2<-direffl2[2:nrow(direffl2),1]
  rownames(direff2)<-direffl2
  if (outscreen==1)
  {cat("\nRelative conditional direct effect(s) of X on Y:\n")
  print(direff2,right=T)}
 }
 direffl2<-direffl4
}

 if ((bcmat[nrow(bcmat),1]==0) & (counterf != 1) & (outscreen==1))
 {cat("\nThe direct effect of X on Y is fixed to zero.\n")}

#Here is the start of the indirect effects
if (nms==1){indmark<-matrix(2);indsets<-t(matrix(c(1,2)))}
if (nms==2){indmark<-t(matrix(c(2,2,3)));indsets<-t(matrix(c(1,4,2,5,1,3,5)));thetam<-matrix(1
if (nms==3)
{indmark<-t(matrix(c(2,2,2,3,3,3,4)))
 indsets<-t(matrix(c(1,7,2,8,4,9,1,3,8,1,5,9,2,6,9,1,3,6,9)))
 thetam<-t(matrix(c(1,2,3)))}
 if (nms==4)
{indmark<-t(matrix(c(2,2,2,2,3,3,3,3,3,3,4,4,4,4,5)))
 indsets<-t(matrix(c(1,11,2,12,4,13,7,14,1,3,12,1,5,13,1,8,14,2,6)))
 indsets<-cbind(indsets,t(matrix(c(13,2,9,14,4,10,14,1,3,6,13,1,3,9,14,1,5,10,14,2,6,10,14,1,3
 thetam<-t(matrix(c(1,2,4,3,5,6)))}
if (nms==5)
{indmark<-t(matrix(c(2,2,2,2,2,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5,6)))
 indsets<-t(matrix(c(1,16,2,17,4,18,7,19,11,20,1,3,17,1,5,18,1,8,19,1,12,20,2,6,18,2,9,19,2)))
 indsets<-cbind(indsets,t(matrix(c(13,20,4,10,19,4,14,20,7,15,20,1,3,6,18,1,3,9,19,1,3,13,20,1
 indsets<-cbind(indsets,t(matrix(c(19,1,5,14,20,1,8,15,20,2,6,10,19,2,6,14,20,2,9,15,20,4,10,1
 indsets<-cbind(indsets,t(matrix(c(3,6,10,19,1,3,6,14,20,1,3,9,15,20,1,5,10,15,20,2,6,10,15,20
 indsets<-cbind(indsets,t(matrix(c(6,10,15,20))))
 thetam<-t(matrix(c(1,2,5,3,6,8,4,7,9,10)))}
if (nms==6)
{indmark<-t(matrix(c(2,2,2,2,2,2,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,
 indmark<-cbind(indmark,t(matrix(c(5,5,5,5,5,5,5,5,5,5,6,6,6,6,6,6,7))))
 indsets<-t(matrix(c(1,22,2,23,4,24,7,25,11,26,16,27,1,3,23,1,5,24,1,8,25,1,12,26,1,17,27,2,6,
 indsets<-cbind(indsets,t(matrix(c(27,4,10,25,4,14,26,4,19,27,7,15,26,7,20,27,11,21,27,1,3,6,2
 indsets<-cbind(indsets,t(matrix(c(18,27,1,5,10,25,1,5,14,26,1,5,19,27,1,8,15,26,1,8,20,27,1,1
 indsets<-cbind(indsets,t(matrix(c(2,6,19,27,2,9,15,26,2,9,20,27,2,13,21,27,4,10,15,26,4,10,20
 indsets<-cbind(indsets,t(matrix(c(1,3,6,10,25,1,3,6,14,26,1,3,6,19,27,1,3,9,15,26,1,3,9,20,27
 indsets<-cbind(indsets,t(matrix(c(1,5,10,20,27,1,5,14,21,27,1,8,15,21,27,2,6,10,15,26,2,6,10,
 indsets<-cbind(indsets,t(matrix(c(21,27,4,10,15,21,27,1,3,6,10,15,26,1,3,6,10,20,27,1,3,6,14,
 indsets<-cbind(indsets,t(matrix(c(15,21,27,2,6,10,15,21,27,1,3,6,10,15,21,27))))
 thetam<-t(matrix(c(1,2,6,3,7,10,4,8,11,13,5,9,12,14,15)))}
if  (nms==7)
{indmark<-t(matrix(c(2,2,2,2,2,2,2)));indsets<-t(matrix(c(1,29,2,30,4,31,7,32,11,33,16,34,22,35
if  (nms==8)
{indmark<-t(matrix(c(2,2,2,2,2,2,2,2)));indsets<-t(matrix(c(1,37,2,38,4,39,7,40,11,41,16,42,22,
if  (nms==9)
{indmark<-t(matrix(c(2,2,2,2,2,2,2,2,2)));indsets<-t(matrix(c(1,46,2,47,4,48,7,49,11,50,16,51,2
if  (nms==10)
{indmark<-t(matrix(c(2,2,2,2,2,2,2,2,2,2)))
indsets<-t(matrix(c(1,56,2,57,4,58,7,59,11,60,16,61,22,62,29,63,37,64,46,65)))}
indlbl<-"Ind1"
for  (indb  in  (2:63))
{indlbl<-cbind(indlbl,paste("Ind",indb,sep=''))}
indlbl<-matrix(indlbl)
cntname<-"(C1)"
for  (indb  in  (2:105))
{cntname<-cbind(cntname,paste("(C",indb,")",sep=''))}
cntname<-matrix(cntname)
indmake<-matrix(0,ncol(indmark),(nms+2))
indmod<-matrix(999,ncol(indmark),1)
indmmm<-matrix(0,ncol(indmark),1)
indmmmt<-matrix(0,ncol(indmark),1)
start<-1;end<-0;nindfx<-0
indlocs<-matrix(999,nrow(thetaxmb),ncol(paths))
indkey<-matrix("            ",ncol(indmark),1+((max(indmark)*2)+1))
c1<-1;c2<-1;c3<-1
for  (i  in  (1:ncol(paths)))
{
 if  (pathtype[1,i]==1){indlocs[,i]<-thetaxmb[,c1];c1<-(c1+1)}
 if  (pathtype[1,i]==3){indlocs[,i]<-thetamyb[,c2];c2<-(c2+1)}
 if  ((pathtype[1,i]==2)  &  (nms  <  7)  &  (serial==1)){indlocs[,i]<-thetammb[,thetam[1,c3]]
}
for  (i  in  (1:ncol(indlocs)))
{
 c1<-2
 for  (j  in  (2:nrow(indlocs)))
 {
  if  (indlocs[j,i]  !=  0){indlocs[c1,i]<-indlocs[j,i];c1<-(c1+1)}
 }
 indlocs[1,i]<-(c1-2)
}
indlocs<-indlocs[1:max((indlocs[1,])+1),]
for  (i  in  (1:ncol(indmark)))
{
 numget<-indmark[1,i];end<-end+numget;gotcha<-t(matrix(indsets[1,start:end]))
 start<-end+1;ok<-1;temp<-0;repoman<-matrix(0,4,1)
```

```
for (j in (1:ncol(gotcha)))
{
  if (paths[1,gotcha[1,j]]==0){ok<-0}
  if (pathsmod[1,gotcha[1,j]] > 0)
  {
    temp<-1
    temp2<-rbind(pathsw[1,gotcha[1,j]],pathsz[1,gotcha[1,j]],pathswz[1,gotcha[1,j]],0)
    temp2<-matrix(temp2)
    if ((temp2[1,1]==1) & (temp2[2,1]==1) & (temp2[3,1]==0)){temp2[4,1]<-1}
    repoman<-repoman+temp2
  }
}
temp<-0;tempmmm<-0;typemmm<-0
if ((repoman[1,1] > 0) & (repoman[2,1]==0))
{
  temp<-1
  if (repoman[1,1]==1){tempmmm<-1}
  if ((repoman[1,1] > 1) & ((wdich==1) | (mcw > 0)))
  {
    tempmmm<-12;typemmm<-mcw
    if (wdich==1){typemmm<-1}
  }
  if ((repoman[1,1] > 1) & ((wdich==0) & (mcw==0))){tempmmm<-101}
}
if ((repoman[1,1]==0) & (repoman[2,1] > 0))
{
  temp<-2
  if (repoman[2,1]==1){tempmmm<-2}
  if ((repoman[2,1] > 1) & ((zdich==1) | (mcz > 0)))
  {
    tempmmm<-22;typemmm<-mcz
    if (zdich==1){typemmm<-1}
  }
  if ((repoman[2,1] > 1) & ((zdich==0) & (mcw==0))){tempmmm<-102}
}
if ((repoman[1,1] > 0) & (repoman[2,1] > 0))
{
  temp<-3
  if ((repoman[1,1]==1) & (repoman[2,1]==1))
  {
    if (repoman[4,1]==1){tempmmm<-31}
    if (repoman[3,1]==1){tempmmm<-41}
  }
}
if ((repoman[1,1]==1) & (repoman[2,1]==1) & (repoman[3,1]==0) & (repoman[4,1]==0)){tem
if (ok==1)
{
  nindfx<-nindfx+1
  indmake[nindfx,1]<-numget;indmod[nindfx,1]<-temp;indmmm[nindfx,1]<-tempmmm
  indmmmt[nindfx,1]<-typemmm;indmake[nindfx,2:(1+numget)]<-gotcha;indkey[nindfx,1]<-xnames
  for (j in (1:numget))
  {
    indkey[nindfx,(j*2+1)]<-pathsdv[1,gotcha[1,j]]
    indkey[nindfx,(j*2)]<-"     ->     "
  }
  }
}
indkey<-matrix(indkey[1:nindfx,1:((max(indmake[,1])*2)+1)],nrow=nindfx)
indmake<-matrix(indmake[1:nindfx,1:(max(indmake[,1])+1)],ncol=(max(indmake[,1])+1))
indmod<-as.matrix(indmod[1:nrow(indmake),1])
indmmm<-as.matrix(indmmm[1:nrow(indmake),1])
indmmmt<-as.matrix(indmmmt[1:nrow(indmake),1])
ncpairs<-(((nindfx)*(nindfx-1))/2)
if (((contrast==1) | (contrast==2)) & (ncpairs > 105))
{contrast<-0;notecode[notes,1]<-13;notes<-notes+1}
if (contrast==3)
{
  if (ncol(contvec) != nindfx)
  {contrast<-0;notecode[notes,1]<-14;notes<-notes+1}
}

#This is for models with no moderator */
if (anymod==0)
{
  if ((nms==1) & (contrast > 0)){contrast<-0}
  efloop<-(((1-as.numeric(effsize==0))*2)+1)-((((mcx>0) | (xdich==1)))*(1-as.numeric(effsize=
  for (kk in (1:efloop))
  {
    if (boot==0)
    {bootres<-obscoeff;indtab<-matrix(999);inddiff<-matrix(999);bootysd<-matrix(ysd);bootxsd<-m
    if (boot > 0)
    {bootres<-rbind(obscoeff,bootres);indtab<-matrix(999,1,4);inddiff<-matrix(999,nrow(bootres)
    indtotal<-matrix(0,nrow(bootres),1)
    for (i in (1:nrow(indmake)))
    {
      for (j in (1:nxvls))
      {
        indtemp<-matrix(1,nrow(bootres),1)
        for (k in (1:indmake[i,1]))
        {
          jtemp<-1
          if ((j > 1) & (k==1)){jtemp<-j}
          indtemp<-indtemp*bootres[,pathsfoc[jtemp,indmake[i,(k+1)]]]
        }
        if (kk==2){indtemp<-indtemp/bootysd}
        if (kk==3){indtemp<-((bootxsd*indtemp)/bootysd)}
        if (contrast != 0){inddiff<-cbind(inddiff,indtemp)}
        if (nxvls==1){indtotal<-(indtotal+indtemp)}
        indeff<-indtemp[1,1]
        if (boot > 0)
        {
          if (bc==0){bcitmp<-process.pboot3(indtemp[2:nrow(indtemp),1],cilow,cihigh)}
          if (bc==1)
          {
            bcbout<-process.bcboot3(indtemp[2:nrow(indtemp),1],indtemp[1,1],xp2,badend,priorlo,
            bcitmp<-matrix(unlist(bcbout[1]))
            badend<-unlist(bcbout[2]);priorlo<-unlist(bcbout[3]);priorhi<-unlist(bcbout[4])
          }
          indeff<-cbind(indeff,t(bcitmp))
        }
        indtab<-rbind(indtab,indeff)
      }
    }
    indtab<-matrix(indtab[2:nrow(indtab),],nrow=(nrow(indtab)-1))
    rowlbs<-matrix(indlbl[1:nrow(indtab),1])
    rowbls3<-rowlbs
    if (mc > 0)
    {
      inddiff<-matrix(-999,mc,1)
      indtab2<-matrix(-999,nrow(indtab),4)
      indtab2[,1]<-indtab;indtab<-indtab2;mcct<-0
      indtotal<-matrix(0,mc,1)
      if (kk==1)
      {
        x1<-sqrt(-2*log(matrix(runif(mc*nrow(mcsopath)),mc,nrow(mcsopath))))*cos((2*3.141592653
        x1<-x1%*%chol(indcov)
        for (ii in (1:nrow(x1))){x1[ii,]<-(x1[ii,]+t(mcsopath))}
      }
      for (ii in (1:nms))
      {
        tmpb<-x1[,((nms*nxvls)+ii)];tmpb2<-tmpb
        if (nxvls > 1)
        {
          for (jj in (1:(nxvls-1)))
          {tmpb2<-cbind(tmpb2,tmpb)}
        }
        indtemp<-as.matrix(x1[,(((ii-1)*nxvls)+1):(ii*nxvls)]*tmpb2)
        for (jj in (1:ncol(indtemp)))
        {
          if (kk==2){indtemp[,jj]<-indtemp[,jj]/ysd}
          if (kk==3){indtemp[,jj]<-(xsd*indtemp[,jj])/ysd}
          mcicon<-process.pboot3(indtemp[,jj],cilow,cihigh)
          mcct<-mcct+1
          indtab[mcct,2:4]<-t(mcicon)
        }
        if (nxvls==1)
        {
          indtotal<-indtotal+indtemp
          if (contrast != 0){inddiff=cbind(inddiff,indtemp)}
        }
      }
    }
    if ((normal==1) & (sobelok==1))
    {
      sobelmat<-matrix(indtab[,1])
      sobelmat<-cbind(sobelmat,(sobelmat/2),sobelmat,sobelmat)
      for (ii in (1:nms))
      {
        se2b<-(indcov[((nms*nxvls)+ii),((nms*nxvls)+ii)])
        bpath2<-(mcsopath[((nms*nxvls)+ii),1])^2
        se2a<-matrix(diag(matrix(indcov[(((ii-1)*nxvls)+1):(ii*nxvls),(((ii-1)*nxvls)+1):(ii*nx
        apath2<-matrix(mcsopath[(((ii-1)*nxvls)+1):(ii*nxvls),1])^2
        sesobel<-sqrt(apath2*se2b+bpath2*se2a+se2a*se2b)
        sobelmat[(((ii-1)*nxvls)+1):(ii*nxvls),2]<-sesobel
      }
      sobelmat[,3]<-sobelmat[,1]/sobelmat[,2]
      sobelmat[,4]<-2*(1-pnorm(abs(sobelmat[,3])))
    }
    if (serial==0){rowlbs<-t(mnames)}
    if ((nxvls==1) & (nms > 1))
    {
      rowlbs<-rbind("TOTAL",rowlbs)
      indtemp<-indtotal[1,1]
      if ((boot > 0) & (nxvls==1))
      {
        if (bc==0){bcitmp<-process.pboot3(indtotal[2:nrow(indtotal),1],cilow,cihigh)}
        if (bc==1)
        {
          bcbout<-process.bcboot3(indtotal[2:nrow(indtotal),1],indtotal[1,1],xp2,badend,priorlo
          bcitmp<-matrix(unlist(bcbout[1]))
          badend<-unlist(bcbout[2]);priorlo<-unlist(bcbout[3]);priorhi<-unlist(bcbout[4])
        }
        indtemp<-cbind(indtemp,t(bcitmp))
      }
      if (mc > 0)
      {
        obtmc<-indtab[,1];indtemp<-sum(obtmc)
        mcicon2<-process.pboot3(indtotal[,1],cilow,cihigh)
        indtemp<-cbind(indtemp,t(mcicon2))
      }
      indtab<-rbind(indtemp,indtab)
    }
    bootlbs<-"Effect"
    if (boot > 0){bootlbs<-c("Effect","BootSE","BootLLCI","BootULCI")}
    if (mc > 0){bootlbs<-c("Effect","MC SE","MC LLCI","MC ULCI")}
    if (nxvls==1)
    {
      if (contrast != 0)
      {
        inddiff<-matrix(inddiff[,2:ncol(inddiff)],nrow=nrow(inddiff))
        if (mc > 0){inddiff=rbind(t(obtmc),inddiff)}
        if (contrast==3)
        {
          inddifft<-matrix(inddiff%*%t(contvec))
          indtemp<-inddifft[1,1]
          if ((boot > 0) | (mc > 0))
          {
            if ((mc > 0) | ((boot > 0) & (bc==0)))
            {bcicon<-process.pboot3(inddifft[2:nrow(inddifft),1],cilow,cihigh)}
            if ((boot > 0) & (bc==1))
            {
              bcbout<-process.bcboot3(inddifft[2:nrow(inddifft),1],inddifft[1,1],xp2,badend,pri
              bcicon<-matrix(unlist(bcbout[1]))
              badend<-unlist(bcbout[2]);priorlo<-unlist(bcbout[3]);priorhi<-unlist(bcbout[4])
            }
            indtemp<-cbind(indtemp,t(bcicon))
          }
          indtab<-rbind(indtab,indtemp)
        }
        if ((contrast==1) | (contrast==2))
        {
```

```
  {
    conkey<-matrix(" ",1,4)
    for (i in (1:(ncol(inddiff)-1)))
    {
      for (j in ((i+1):ncol(inddiff)))
      {
        inddifft<-matrix(inddiff[,i]-inddiff[,j])
        if (contrast==2){inddifft<-matrix(abs(inddiff[,i])-abs(inddiff[,j]))}
        indtemp<-inddifft[1,1]
        conkeyt<-cbind(" ", rowlbs[(i+1),1]," minus ",rowlbs[(j+1),1])
        conkey<-rbind(conkey,conkeyt)
        if ((boot > 0) | (mc > 0))
        {
          if ((mc > 0) | ((boot > 0) & (bc==0)))
          {bcitmp2<-process.pboot3(inddifft[2:nrow(inddifft),1],cilow,cihigh)}
          if ((boot > 0) & (bc==1))
          {
            bcbout<-process.bcboot3(inddifft[2:nrow(inddifft),1],inddifft[1,1],xp2,badend
            bcitmp2<-matrix(unlist(bcbout[1]))
            badend<-unlist(bcbout[2]);priorlo<-unlist(bcbout[3]);priorhi<-unlist(bcbout[4
          }
          indtemp<-cbind(indtemp,t(bcitmp2))
        }
        indtab<-rbind(indtab,indtemp)
      }
    }
  }
  if (contrast != 3)
  {contlbs<-matrix(cntname[1:(((nindfx)*(nindfx-1))/2),1])}
  if (contrast==3)
  {contlbs<-matrix("(C1)")}
  rowlbs<-rbind(rowlbs,contlbs)
}
resultm2<-matrix(99999,nrow(indtab),maxresm)
resultm2[1:nrow(indtab),1:ncol(indtab)]<-indtab
resultm<-rbind(resultm,resultm2)
indtab5<-noquote(matrix(sprintf(decimals,indtab),nrow=nrow(indtab)))
colnames(indtab5)<-bootlbs
rownames(indtab5)<-rowlbs
if (outscreen==1)
{
  if (kk==1)
  {
    cat("\nIndirect effect(s) of X on Y:\n")
    print(indtab5,right=T)
  }
  if (kk==2)
  {
    cat("\nPartially standardized indirect effect(s) of X on Y:\n")
    print(indtab5,right=T)
  }
  if (kk==3)
  {
    cat("\nCompletely standardized indirect effect(s) of X on Y:\n")
    print(indtab5,right=T)
  }
```

```
    }
  }
  if ((normal==1) & (sobelok==1) & (kk==1))
  {
    sobellab<-c("Effect",hclab,"Z","p")
    sobelrlb<-rowlbs
    if (nms > 1){sobelrlb<-matrix(rowlbs[2:(1+nms),1])}
    resultm2<-matrix(99999,nrow(sobelmat),maxresm)
    resultm2[1:nrow(sobelmat),1:ncol(sobelmat)]<-sobelmat
    resultm<-rbind(resultm,resultm2)
    sobelmt2<-noquote(matrix(sprintf(decimals,sobelmat),nrow=nrow(sobelmat)))
    rownames(sobelmt2)<-sobelrlb
    colnames(sobelmt2)<-sobellab
    if (outscreen==1)
    {cat("\nNormal theory test for indirect effect(s):\n")
    print(sobelmt2,right=T)}
  }
  if (contrast != 0)
  {
    if (((contrast==1) | (contrast==2)) & (kk==efloop))
    {
      conkey<-matrix(conkey[2:nrow(conkey),],nrow=(nrow(conkey)-1))
      if (outscreen==1)
      {cat("\nSpecific indirect effect contrast definition(s):\n")
      write.table(conkey,quote=FALSE,row.names=contlbs,col.names=FALSE,sep = "    ")}
    }
    if ((contrast==3) & (kk==efloop))
    {
      crowlbs<-rowlbs[2:(nindfx+1),1]
      contvect<-noquote(matrix(sprintf(decimals,contvec),nrow=1))
      colnames(contvect)<-crowlbs
      rownames(contvect)<-"(C1)"
      if (outscreen==1)
      {cat("\nSpecific indirect effect contrast weights:\n")
      print(contvect,right=T)}
    }
    if ((contrast==2) & (kk==efloop) & (outscreen==1))
    {cat("\nContrasts are differences between absolute values of indirect effects\n
  }
  if ((serial==1) & (kk==efloop))
  {rowlbst<-matrix(rowlbs[2:nrow(rowlbs),1],nrow<-(nrow(rowlbs)-1))
  if (outscreen==1)
  {cat("\nIndirect effect key:\n")
  write.table(indkey,quote=FALSE,row.names=rowbls3,col.names=FALSE,sep="  ")}}
}
if (nxvls > 1)
{
  if (outscreen==1)
  {
    if (kk==1){cat("\nRelative indirect effects of X on Y:\n")}
    if (kk==2){cat("\nPartially standardized relative indirect effects of X on Y:\
    if (kk==3){cat("\nCompletely standardized relative indirect effects of X on Y:
  }
  for (i in (1:nrow(indmake)))
  {
    indtabsm<-indtab[(((i-1)*nxvls)+1):(nxvls*i),]
```

```
        indtabsm<-indtab[(((i-1)*nxvls)+1):(nxvls*i),]
        indkeyt<-indkey[i,]
        if (outscreen==1)
        {
          cat("\n")
          write.table(t(indkeyt),quote=FALSE,row.names=FALSE,col.names=FALSE,sep="  ")
          cat("\n")
        }
        if (bcmat[nrow(bcmat),1]==0)
        {direffl2<-xcatlab[1:nxvls,1]}
        resultm2<-matrix(99999,nrow(indtabsm),maxresm)
        resultm2[1:nrow(indtabsm),1:ncol(indtabsm)]<-indtabsm
        resultm<-rbind(resultm,resultm2)
        indtasm2<-noquote(matrix(sprintf(decimals,indtabsm),nrow=nrow(indtabsm)))
        colnames(indtasm2)<-bootlbs
        rownames(indtasm2)<-direffl2
        if (outscreen==1)
        {print(indtasm2,right=T)}
        if ((normal==1) & (sobelok==1) & (kk==1))
        {
          sobelsm=sobelmat[(((i-1)*nxvls)+1):(nxvls*i),]
          sobellab=c("Effect",hclab,"Z","p")
          resultm2<-matrix(99999,nrow(sobelsm),maxresm)
          resultm2[1:nrow(sobelsm),1:ncol(sobelsm)]<-sobelsm
          resultm<-rbind(resultm,resultm2)
          sobelsm2<-noquote(matrix(sprintf(decimals,sobelsm),nrow=nrow(sobelsm)))
          colnames(sobelsm2)<-sobellab
          rownames(sobelsm2)<-direffl2
          if (outscreen==1)
          {cat("\n     Normal theory test for relative indirect effects:\n")
          print(sobelsm2,right=T)}
        }
      }
    }
    if ((effsize==1) & (boot > 0)){bootres=bootres[2:nrow(bootres),]}
  }
}
#this is the end of the no moderators loop */


if (anymod > 0)
{
  if (boot==0){bootres<-obscoeff;indtab<-999}
  if (boot > 0){bootres<-rbind(obscoeff,bootres);indtab<-matrix(999,1,4)}
  if (sum(as.numeric(indmod > 0))==nrow(indmod))
  {if (outscreen==1)
    {
      if (nxvls > 1)
      {cat("\nRelative conditional indirect effects of X on Y:\n")}
      if (nxvls==1)
      {cat("\nConditional indirect effects of X on Y:\n")}
    }
  }
  if (sum(as.numeric(indmod > 0)) < nrow(indmod))
  {if (outscreen==1)
    {
```

```
  {
    if (nxvls > 1)
    {cat("\nRelative conditional and unconditional indirect effects of X on Y:\n")}
    if (nxvls==1)
    {cat("\nConditional and unconditional indirect effects of X on Y:\n")}
  }
}
for (i in (1:nrow(indmake)))
{
  indtab<-matrix(0,1,4)
  indkeyt<-indkey[i,]
  if (outscreen==1)
  {
    cat("\nINDIRECT EFFECT:\n\n")
    write.table(t(indkeyt),quote=FALSE,row.names=FALSE,col.names=FALSE,sep="  ")
    cat("\n")
  }
  if (indmod[i,1]==0)
  {
    for (j in (1:nxvls))
    {
      indtemp<-matrix(1,nrow(bootres),1)
      for (k in (1:indmake[i,1]))
      {
        jtemp<-1
        if ((j > 1) & (k==1)){jtemp<-j}
        indtemp<-indtemp*bootres[,pathsfoc[jtemp,indmake[i,(k+1)]]]
      }
      indeff<-indtemp[1,1]
      if (boot > 0)
      {
        if (bc==0){bcitmp3<-process.pboot3(indtemp[2:nrow(indtemp),1],cilow,cihigh)}
        if (bc==1)
        {
          bcbout<-process.bcboot3(indtemp[2:nrow(indtemp),1],indtemp[1,1],xp2,badend,priorlo,
          bcitmp3<-matrix(unlist(bcbout[1]))
          badend<-unlist(bcbout[2]);priorlo<-unlist(bcbout[3]);priorhi<-unlist(bcbout[4])
        }
        indeff<-cbind(indeff,t(bcitmp3))
      }
      indtab<-rbind(indtab,indeff)
    }
    indtab<-matrix(indtab[2:nrow(indtab),],ncol=ncol(indtab))
    resultm2<-matrix(99999,nrow(indtab),maxresm)
    resultm2[1:nrow(indtab),1:ncol(indtab)]<-indtab
    resultm<-rbind(resultm,resultm2)
    indtab10<-noquote(matrix(sprintf(decimals,indtab),nrow=nrow(indtab)))
    if (nxvls > 1)
    {rownames(indtab10)<-xcatlab[1:nxvls,1]}
    if (nxvls==1)
    {rownames(indtab10)<-"  "}
    colnames(indtab10)<-c("Effect","BootSE","BootLLCI","BootULCI")
    if (outscreen==1)
    {print(indtab10,right=T)}
  }
  # end of unmoderated
  # end of unmoderated
  # start of moderated
  if (indmod[i,1] > 0)
  {
    if (indmod[i,1]==1)
    {indmodva<-wmodvals;indprova<-wprobval;condlbs<-wnames;printw<-1}
    if (indmod[i,1]==2)
    {indmodva<-zmodvals;indprova<-zprobval;condlbs<-znames;printz<-1}
    if (indmod[i,1]==3)
    {
      cntmp<-1;printz<-1;printw<-1
      indmodva<-matrix(999,(nrow(wmodvals)*nrow(zmodvals)),2)
      for (k7 in (1:nrow(wmodvals)))
      {
        for (k8 in (1:nrow(zmodvals)))
        {indmodva[cntmp,]<-cbind(wmodvals[k7,1],zmodvals[k8,1]);cntmp<-cntmp+1}
      }
      condlbs<-cbind(wnames,znames)
    }
    condres<-matrix(999,nrow(indmodva),1)
    if (boot > 0)
    {condres=matrix(999,nrow(indmodva),4)}
    condres<-cbind(indmodva,condres)

    # Here is where the computations start
    for (k4 in (1:nxvls))
    {
      imm3<-matrix(1,nrow(bootres),1);imm4<-matrix(1,nrow(bootres),1);indcontr<-0
      if (indmod[i,1]==3){tihsw<-wprobval;tihsz<-zprobval}
      for (k1 in (1:nrow(indmodva)))
      {
        tucker2<-matrix(1,nrow(bootres),1)
        imm2<-matrix(1,nrow(bootres),1)
        wfirst<-0;zfirst<-0;immset<-0
        for (k2 in (1:indmake[i,1]))
        {
          colnumb<-indmake[i,(k2+1)]
          if (k2==1)
          {
            wbb<-matrix(0,nrow(bootres),(nwvls*nxvls))
            zbb<-matrix(0,nrow(bootres),(nzvls*nxvls))
            wzbb<-matrix(0,nrow(bootres),(nwvls*nzvls*nxvls))
          }
          if (k2 != 1)
          {
            wbb<-matrix(0,nrow(bootres),nwvls)
            zbb<-matrix(0,nrow(bootres),nzvls)
            wzbb<-matrix(0,nrow(bootres),(nwvls*nzvls))
          }
          cnt<-1;tihs<-matrix(indlocs[2:((indlocs[1,colnumb])+1),colnumb])
          if (k2==1)
          {
            focbb<-matrix(tihs[1:nxvls,1]);focbb<-matrix(bootres[,focbb],ncol=nxvls)
            if (indmmm[i,1] > 0)
            {imm<-matrix(focbb[,k4]);condbb<-matrix(imm,ncol=ncol(imm))}
            focaddon<-matrix(0,1,nxvls);focaddon[1,k4]<-1;cnt<-cnt+nxvls;placeh<-nxvls
```

```
focaddon<-matrix(0,1,nxvls);focaddon[1,k4]<-1;cnt<-cnt+nxvls;placen<-nxvls
if  (indmod[i,1]==1)
{
  tihsz<-matrix(0,nrow(wprobval),(nzvls*nxvls))
  tihswz<-matrix(0,nrow(wprobval),(nwvls*nzvls*nxvls))
  if  (pathsw[1,colnumb]==1)
  {
    temp<-matrix(0,nrow(wprobval),(nxvls*nwvls))
    for  (k5  in  (1:nrow(wprobval)))
      {
        for  (k6  in  (1:nwvls))
        {temp[k5,  (((k4-1)*nwvls)+k6)]<-wprobval[k5,k6]}
      }
      indprova<-cbind(temp,tihsz,tihswz)
  }  else  {
      indprova<-cbind(wprobval,tihsz,tihswz)}
}
if  (indmod[i,1]==2)
{
  tihsw<-matrix(0,nrow(zprobval),(nwvls*nxvls))
  tihswz<-matrix(0,nrow(zprobval),(nwvls*nzvls*nxvls))
  if  (pathsz[1,colnumb]==1)
  {
    temp<-matrix(0,nrow(zprobval),(nxvls*nzvls))
    for  (k5  in  (1:nrow(zprobval)))
    {
      for  (k6  in  (1:nzvls))
      {temp[k5,(((k4-1)*nzvls)+k6)]<-zprobval[k5,k6]}
    }
      indprova<-cbind(tihsw,temp,tihswz)
  }  else  {
      indprova<-cbind(tihsw,zprobval,tihswz)}
}
if  (indmod[i,1]==3)
{
  indprova<-matrix(0,(nrow(wprobval)*nrow(zprobval)),((ncol(wprobval)*nxvls)+(nco
  cntemp<-1
  for  (k7  in  (1:nrow(wprobval)))
  {
    for  (k8  in  (1:nrow(zprobval)))
    {
      temp<-(wprobval[k7,]*focaddon[1,k4])
      indprova[cntemp,(((k4-1)*nwvls)+1):(k4*(nwvls))]<-temp
      temp<-zprobval[k8,]*focaddon[1,k4]
      indprova[cntemp,((((k4-1)*nzvls)+1)+(nxvls*nwvls)):((((k4-1)*nzvls)+1)+(nxv
      cntemp<-cntemp+1
    }
  }
  if  (pathsz[1,colnumb]==0)
  {
    temp<-matrix(0,nrow(indprova),(ncol(zprobval)*nxvls))
    indprova[,((ncol(wprobval)*nxvls)+1):((ncol(wprobval)+ncol(zprobval))*nxvls)]
  }
  if  (pathsw[1,colnumb]==0)
  {
    temp<-matrix(0,nrow(indprova),(ncol(wprobval)*nxvls))
```

```
      temp<-matrix(0,nrow(indprova),(ncol(wprobval)*nxvls))
      indprova[,1:(ncol(wprobval)*nxvls)]<-temp
    }
    if (pathswz[1,colnumb]==1)
    {
      cntemp<-(ncol(wprobval)*nxvls)+(ncol(zprobval)*nxvls)+((k4-1)*ncol(wprobval)*
      for (k7 in (1:ncol(wprobval)))
      {
        for (k8 in (1:ncol(zprobval)))
        {
          indprova[,cntemp]<-matrix(indprova[,((ncol(wprobval)*(k4-1))+k7)])*matrix
          cntemp<-cntemp+1
        }
      }
    }
  }
}
if (k2 > 1)
{
  focbb<-tihs[1,1];focbb=matrix(bootres[,focbb])
  if (indmmm[i,1] > 0){imm<-matrix(focbb[,1]);condbb<-matrix(imm)}
  focaddon<-matrix(1);cnt<-cnt+1;placeh<-1
  if (indmod[i,1]==1)
  {
    tihsz<-matrix(0,nrow(wprobval),nzvls)
    tihswz<-matrix(0,nrow(wprobval),(nwvls*nzvls))
    indprova<-cbind(wprobval,tihsz,tihswz)
  }
  if (indmod[i,1]==2)
  {
    tihsw<-matrix(0,nrow(zprobval),nwvls)
    tihswz<-matrix(0,nrow(zprobval),(nwvls*nzvls))
    indprova<-cbind(tihsw,zprobval,tihswz)
  }
  if (indmod[i,1]==3)
  {
    indprova<-matrix(0,(nrow(wprobval)*nrow(zprobval)),((ncol(wprobval)+ncol(zprobv
    cntemp<-1
    for (k7 in (1:nrow(wprobval)))
    {
      for (k8 in (1:nrow(zprobval)))
      {
        indprova[cntemp,1:(ncol(wprobval)+ncol(zprobval))]<-cbind(t(matrix(wprobval
        cntemp<-cntemp+1
      }
    }
    if (pathsz[1,colnumb]==0)
    {
      temp<-matrix(0,nrow(indprova),ncol(zprobval))
      indprova[,(ncol(wprobval)+1):(ncol(wprobval)+ncol(zprobval))]<-temp
    }
    if (pathsw[1,colnumb]==0)
    {
      temp<-matrix(0,nrow(indprova),ncol(wprobval))
      indprova[,1:ncol(wprobval)]<-temp
    }
```

```
            ,
            if  (pathswz[1,colnumb]==1)
            {
              cntemp<-(ncol(wprobval)+ncol(zprobval)+1)
              for  (k7  in  (1:ncol(wprobval)))
              {
                for  (k8  in  (1:ncol(zprobval)))
                {
                  indprova[,cntemp]<-matrix(indprova[,k7])*matrix(indprova[,(ncol(wprobval)
                  cntemp<-cntemp+1
                }
              }
            }
          }
        }
        if  (pathsw[1,colnumb]==1)
        {
          wbb<-matrix(tihs[cnt:(cnt+(placeh*nwvls)-1),1])
          wbb<-matrix(bootres[,wbb],ncol=nrow(wbb))
          immlbs2<-matrix(wcatlab[1:nwvls,1])
          if  (zfirst==0){wfirst<-1}
          if  ((indmmm[i,1]==1)  |  (indmmm[i,1]==31)  |  (indmmm[i,1]==51))
          {
            imm<-matrix(wbb[,1])
            for  (k7  in  (1:nwvls))
            {imm<-cbind(imm,wbb[,(((k4-1)*nwvls*(as.numeric(k2==1)))+k7)])}
              imm<-matrix(imm[,2:ncol(imm)],ncol=(ncol(imm)-1))
          }
          if  ((indmmm[i,1]==41)  |  (indmmm[i,1]==51))
          {
            condbb<-matrix(0,nrow(bootres),1)
            for  (k7  in  (1:nwvls))
            {condbb<-cbind(condbb,wbb[,(((k4-1)*nwvls*(as.numeric(k2==1)))+k7)])}
            condbb<-matrix(condbb[,2:ncol(condbb)],ncol=(ncol(condbb)-1))
          }
          cnt<-cnt+(placeh*nwvls)
        }
        if  (pathsz[1,colnumb]==1)
{
          zbb<-matrix(tihs[cnt:(cnt+(placeh*nzvls)-1),1])
          zbb<-matrix(bootres[,zbb],ncol=nrow(zbb))
          if  (wfirst==0){zfirst<-1}
          if  (indmmm[i,1]  !=  31){immlbs2<-matrix(zcatlab[1:nzvls,1])}
          if  ((indmmm[i,1]==2)  |  (indmmm[i,1]==31)  |  (indmmm[i,1]==51))
          {
            if  (indmmm[i,1]==2){imm<-matrix(zbb[,1])}
            for  (k7  in  (1:nzvls))
            {imm<-cbind(imm,zbb[,(((k4-1)*nzvls*(as.numeric(k2==1)))+k7)])}
            if  ((indmmm[i,1]==2)  |  (indmmm[i,1]==51))
            {
              imm<-matrix(imm[,2:ncol(imm)],ncol=(ncol(imm)-1))
              if  (indmmm[i,1]==51){condbb<-cbind(condbb,imm)}
            }
          }
          cnt<-cnt+(placeh*nzvls)
        }
```

```
if (pathswz[1,colnumb]==1)
{
  wzbb<-matrix(tihs[cnt:(cnt+(placeh*nwvls*nzvls)-1),1])
  wzbb<-matrix(bootres[,wzbb],ncol=nrow(wzbb))
  if (indmmm[i,1]==41)
  {
    imm<-matrix(wzbb[,1])
    for (k7 in (1:(nwvls*nzvls)))
    {imm<-cbind(imm,wzbb[,(((k4-1)*nzvls*nwvls*(as.numeric(k2==1)))+k7)])}
  }
  if (indmmm[i,1]==41)
  {
    imm<-matrix(imm[,2:ncol(imm)],ncol=(ncol(imm)-1))
    condbb<-cbind(condbb,imm[,(ncol(imm)-(nwvls*nzvls)+1):ncol(imm)])
  }
    cnt<-cnt+(placeh*nzvls*nwvls)
}
indprobe<-cbind(focaddon,t(matrix(indprova[k1,])))
tucker<-cbind(focbb,wbb,zbb,wzbb)
for (k3 in (1:ncol(indprobe)))
{tucker[,k3]<-matrix(tucker[,k3]*indprobe[1,k3])}
tucker2<-tucker2*matrix(rowSums(tucker))
if ((indmmm[i,1]==1) | (indmmm[i,1]==2) | (indmmm[i,1]==31) | (indmmm[i,1]==
{
  if (immset==1)
  {
    if ((ncol(imm2)==1) & (ncol(imm)==1)){imm2<-(matrix(imm2)*matrix(imm))}
    if ((indmmm[i,1]==41) | (indmmm[i,1]==51))
    {
      if ((ncol(condbb2) > 1) & (ncol(condbb) > 1))
      {
        condbb2t<-matrix(-999999,nrow(condbb2),(ncol(condbb2)*ncol(condbb)))
        k9<-1
        if (wfirst==1)
        {
          for (k7 in (1:ncol(condbb2)))
          {
            for (k8 in (1:ncol(condbb)))
            {condbb2t[,k9]<-(matrix(condbb2[,k7])*matrix(condbb[,k8]));k9<-k9+1}
          }
        }
        if (zfirst==1)
        {
          for (k7 in (1:ncol(condbb)))
          {
            for (k8 in (1:ncol(condbb2)))
            {condbb2t[,k9]<-(matrix(condbb[,k7])*matrix(condbb2[,k8]));k9<-k9+1}
          }
        }
        condbb2<-matrix(condbb2t,ncol=ncol(condbb2t))
      }
      if ((ncol(condbb2) > 1) & (ncol(condbb)==1))
      {
        for (k7 in (1:ncol(condbb2)))
        {condbb2[,k7]<-(matrix(condbb2[,k7])*matrix(condbb))}
        }
        if ((ncol(condbb2)==1) & (ncol(condbb) > 1))
        {
          for (k7 in (1:ncol(condbb)))
          {condbb[,k7]<-(matrix(condbb2)*matrix(condbb[,k7]))}
          condbb2<-matrix(condbb,ncol=ncol(condbb))
        }
      }
      if ((ncol(imm2) != 1) & (ncol(imm) != 1))
      {
        imm2t<-matrix(-999999,nrow(imm2),(ncol(imm2)*ncol(imm)))
        k9<-1
        if (wfirst==1)
        {
          for (k7 in (1:ncol(imm2)))
          {
            for (k8 in (1:ncol(imm)))
            {imm2t[,k9]<-(matrix(imm2[,k7])*matrix(imm[,k8]));k9<-k9+1}
          }
        }
        if (zfirst==1)
        {
          for (k7 in (1:ncol(imm)))
          {
            for (k8 in (1:ncol(imm2)))
            {imm2t[,k9]<-(matrix(imm[,k7])*matrix(imm2[,k8]));k9<-k9+1}
          }
        }
        imm2<-matrix(imm2t,ncol=ncol(imm2t))
      }
      if ((ncol(imm2) > 1) & (ncol(imm)==1))
      {
        for (k7 in (1:ncol(imm2)))
        {imm2[,k7]<-(matrix(imm2[,k7])*matrix(imm))}
      }
      if ((ncol(imm2)==1) & (ncol(imm) > 1))
      {
        for (k7 in (1:ncol(imm)))
        {imm[,k7]<-(matrix(imm2)*matrix(imm[,k7]))}
        imm2<-matrix(imm,ncol=ncol(imm))
      }
    }
    if (immset==0)
    {
      imm2<-matrix(imm,ncol=ncol(imm))
      if ((indmmm[i,1]==41) | (indmmm[i,1]==51))
      {condbb2<-matrix(condbb,ncol=ncol(condbb))}
      immset<-1
    }
  }
  # that is it for mmm loop */
}
# end of looping through paths: k2
indtemp<-tucker2[1,1]
if ((indmmm[i,1]==12) | (indmmm[i,1]==22))
```

```
{
  imm3<-cbind(imm3,tucker2)
  if  (k1==nrow(indmodva))
  {
    imm3<-matrix(imm3[,2:ncol(imm3)],ncol=(ncol(imm3)-1))
    immstop<-ncol(imm3)
    for  (k8  in  (2:immstop))
    {
      if  (indmmmt[i,1]==1){imm3<-cbind(imm3,(imm3[,k8]-imm3[,1]))}
      if  (indmmmt[i,1]==2){imm3=cbind(imm3,(imm3[,k8]-imm3[,(k8-1)]))}
      if  (indmmmt[i,1]==3)
      {
        imm3<-cbind(imm3,((rowSums(imm3[,(k8:immstop)])/(immstop-k8+1))-imm3[,(k8-1)]
      }
      if  (indmmmt[i,1]==4)
      {imm3<-cbind(imm3,(imm3[,k8]-(rowSums(imm3[,1:immstop])/immstop)))}
    }
    if  (indmmmt[i,1]  <  5)
    {imm2<-matrix(imm3[,(immstop+1):ncol(imm3)],ncol=(ncol(imm3)-immstop))}
  }
}
if  ((indmmm[i,1]>-1)  &  ((contrast==1)  |  (contrast==2)))
{
  imm4<-cbind(imm4,tucker2)
  if  ((k1==nrow(indmodva))  &  (k1  >  1))
  {
    imm4<-matrix(imm4[,2:ncol(imm4)],ncol=(ncol(imm4)-1))
    immstop<-ncol(imm4)
    condcont<-matrix(-999,(immstop*(immstop-1)/2),6)
    for  (k8  in  (1:(immstop-1)))
    {
      for  (k9  in  ((k8+1):immstop))
      {
        if  (contrast==1){imm4<-cbind(imm4,matrix(imm4[,k9]-imm4[,k8]))}
        if  (contrast==2){imm4<-cbind(imm4,matrix(abs(imm4[,k9])-abs(imm4[,k8])))}
        condcont[(ncol(imm4)-immstop),1]<-imm4[1,k9]
        condcont[(ncol(imm4)-immstop),2]<-imm4[1,k8]
      }
    }
    imm4<-matrix(imm4[,(immstop+1):ncol(imm4)],ncol=(ncol(imm4)-immstop))
    for  (k8  in  (1:ncol(imm4)))
    {
      condcont[k8,3]<-imm4[1,k8]
      if  (boot  >  0)
      {
        if  (bc==0){condcon3<-process.pboot3(imm4[2:nrow(imm4),k8],cilow,cihigh)}
        if  (bc==1)
        {
          bcbout<-process.bcboot3(imm4[2:nrow(imm4),k8],imm4[1,k8],xp2,badend,priorlo
          condcon3<-matrix(unlist(bcbout[1]))
          badend<-unlist(bcbout[2]);priorlo<-unlist(bcbout[3]);priorhi<-unlist(bcbout
        }
        condcont[k8,4:6]<-t(condcon3)
      }
    }
```

```
      if (boot==0){condcont<-matrix(condcont[,1:3],ncol=3)}
      indcontr<-1
    }
  }
  if (boot > 0)
  {
    if (bc==0){bcitmp11<-process.pboot3(tucker2[2:nrow(tucker2),1],cilow,cihigh)}
    if (bc==1)
    {
      bcbout<-process.bcboot3(tucker2[2:nrow(tucker2),1],tucker2[1,1],xp2,badend,priorl
      bcitmp11<-matrix(unlist(bcbout[1]))
      badend<-unlist(bcbout[2]);priorlo<-unlist(bcbout[3]);priorhi<-unlist(bcbout[4])
    }
    indtemp<-cbind(indtemp,t(bcitmp11))
  }
  condres[k1,(ncol(indmodva)+1):ncol(condres)]<-indtemp
}
# end of looping through indirect effects: k1
# Here is where the computations end
resultm2<-matrix(99999,nrow(condres),maxresm)
resultm2[1:nrow(condres),1:ncol(condres)]<-condres
resultm<-rbind(resultm,resultm2)
condresp<-noquote(matrix(sprintf(decimals,condres),nrow=nrow(condres)))
if (k4==1)
{
  condlbs<-cbind(condlbs,"Effect")
  if (boot > 0)
  {condlbs<-cbind(condlbs,"BootSE","BootLLCI","BootULCI")}
  condrlb<-t(matrix(replicate(nrow(condres)," ")))
}
if (nxvls > 1)
{condrlb<-matrix(xcatlab[k4,1],nrow(condres),1)}
colnames(condresp)<-condlbs
rownames(condresp)<-condrlb
if (outscreen==1)
{print(condresp,right=T)}
if (indmmm[i,1]==0){cat("\n")}
dichadj<-0;immcat<-0
if (indmmm[i,1] > 0)
{
  if ((indmmm[i,1]==1) | (indmmm[i,1]==12) | (indmmm[i,1]==31))
  {
    if ((wdich==1) & (mcw==0))
    {
      if (indmmm[i,1] != 12){imm2[,1]<-imm2[,1]*(wmax-wmin)}
      if (indmmm[i,1] != 31){dichadj<-1}
    }
    if (((mcw==1) | (mcw==2)) & (indmmm[i,1] != 31)){immcat<-1}
  }
  if ((indmmm[i,1]==2) | (indmmm[i,1]==22) | (indmmm[i,1]==31))
  {
    if ((zdich==1) & (mcz==0))
    {
      if (indmmm[i,1]==31)
      {
```

```
   imm2[,(nwvls+1):ncol(imm2)]<-matrix(imm2[,(nwvls+1):ncol(imm2)],ncol=(ncol(imm2
  }
  if (indmmm[i,1]==2){imm2[,1]<-imm2[,1]*(zmax-zmin)}
  if (indmmm[i,1] != 31){dichadj<-1}
 }
 if (((mcz==1) | (mcz==2)) & (indmmm[i,1] != 31)){immcat<-1}
}
immtemp2<-t(matrix(imm2[1,],ncol=ncol(imm2)))
immtemp<-matrix(immtemp2,ncol=ncol(immtemp2))
immlbs<-"Index"
if (boot > 0)
{
 immtemp<-matrix(0,1,3)
 for (k7 in (1:ncol(imm2)))
 {
  if (bc==0){immbtci<-process.pboot3(imm2[2:nrow(imm2),k7],cilow,cihigh)}
  if (bc==1)
  {
   bcbout<-process.bcboot3(imm2[2:nrow(imm2),k7],imm2[1,k7],xp2,badend,priorlo,pri
   immbtci<-matrix(unlist(bcbout[1]))
   badend<-unlist(bcbout[2]);priorlo<-unlist(bcbout[3]);priorhi<-unlist(bcbout[4])
  }
  immtemp<-rbind(immtemp,t(immbtci))
 }
 immtemp<-matrix(immtemp[2:nrow(immtemp),],ncol=ncol(immtemp))
 immtemp<-cbind(immtemp2,immtemp)
 immlbs<-c(immlbs,"BootSE","BootLLCI","BootULCI")
}
resultm2<-matrix(99999,nrow(immtemp),maxresm)
resultm2[1:nrow(immtemp),1:ncol(immtemp)]<-immtemp
resultm<-rbind(resultm,resultm2)
immtempr<-noquote(matrix(sprintf(decimals,immtemp),nrow=nrow(immtemp)))
colnames(immtempr)<-immlbs
if ((dichadj==0) & (immcat==0) & (indmmmt[i,1] != 5) & (indmmm[i,1] < 100)
{
 if (indmmm[i,1] < 30)
 {
  rownames(immtempr)<-immlbs2
  if (outscreen==1)
  {cat("\n     Index of moderated mediation:\n")
  print(immtempr,right=T)}
  if ((nxvls > 1) & (k4 < nxvls)){cat("\n")}
 }
 if (indmmm[i,1]==31)
 {
  immlbs2<-rbind(immlbs2,matrix(zcatlab[1:nzvls,1]))
  rownames(immtempr)<-immlbs2
  if (outscreen==1)
  {cat("\n     Indices of partial moderated mediation:\n")
  print(immtempr,right=T)}
  if ((nxvls > 1) & (k4 < nxvls)){cat("\n")}
 }
 if ((nzvls==1) & (nwvls==1))
 {
  if ((indmmm[i,1]==41) | (indmmm[i,1]==51))
```

```
{
  for (k7 in (1:nwvls))
  {
    immlbs2<-zcatlab[1:nzvls,1]
    immtemp2<-matrix(immtemp[(((k7-1)*nzvls)+1):(((k7-1)*nzvls)+nzvls),],ncol=nco
    #resultm2<-matrix(99999,nrow(immtemp2),maxresm)
    #resultm2[1:nrow(immtemp2),1:ncol(immtemp2)]<-immtemp2
    #print(resultm)
    #print(resultm2)
    #resultm<-rbind(resultm,resultm2)
    #print(resultm)
    immtem2r<-noquote(matrix(sprintf(decimals,immtemp2),nrow=nrow(immtemp2)))
    colnames(immtem2r)<-immlbs
    if (nwvls > 1)
    {
    primodv<-cbind("Primary moderator:", wcatlab[k7,1])
    if (outscreen==1)
    {write.table(primodv,quote=FALSE,row.names=FALSE,col.names=FALSE)}
    }
    if (nzvls==1)
    {
      rownames(immtem2r)<-" "
      if (outscreen==1)
      {cat("\n          Index of moderated moderated mediation:\n")
      print(immtem2r,right=T)}
    } else {
      rownames(immtem2r)<-immlbs2
      if (outscreen==1)
      {cat("\n          Indices of moderated moderated mediation:\n")
      print(immtem2r,right=T)}
    }
    cmmtemp<-matrix(0,nrow(zprobval),4)
    for (k8 in (1:nrow(zprobval)))
    {
      bbst<-((nwvls+1)+((k7-1)*nzvls));bben<-((nwvls+1)+((k7-1)*nzvls)+(nzvls-1))
      condbb3<-matrix(condbb2[,bbst:bben],ncol=(bben-bbst+1))
      if (ncol(zprobval) > 1)
      {
        condbb3<-condbb3*diag(c(zprobval[k8,]))
      } else {
        condbb3<-condbb3*zprobval[k8,]
      }
      condbb3<-cbind(condbb2[,k7],condbb3)
      icmm<-matrix(rowSums(condbb3))
      cmmtemp[k8,1]<-icmm[1,1]
      if (boot > 0)
      {
        if (bc==0){cmmt3<-process.pboot3(icmm[2:nrow(icmm),1],cilow,cihigh)}
        if (bc==1)
        {
          bcbout<-process.bcboot3(icmm[2:nrow(icmm),1],icmm[1,1],xp2,badend,prior
          cmmt3<-matrix(unlist(bcbout[1]))
          badend<-unlist(bcbout[2]);priorlo<-unlist(bcbout[3]);priorhi<-unlist(bc
        }
        cmmtemp[k8,2:4]<-t(cmmt3)
```

```
                    }
                  cmmlbs<-cbind(znames,t(matrix(immlbs)))
                  }
                  if (boot==0){cmmtemp<-matrix(cmmtemp[,1:1],ncol=1);cmmlbs<-cbind(znames,"Ind
                  cmmtemp<-cbind(zmodvals,cmmtemp)
                  resultm2<-matrix(99999,nrow(cmmtemp),maxresm)
                  resultm2[1:nrow(cmmtemp),1:ncol(cmmtemp)]<-cmmtemp
                  resultm<-rbind(resultm,resultm2)
                  cmmtempr<-noquote(matrix(sprintf(decimals,cmmtemp),nrow=nrow(cmmtemp)))
                  #if (boot==0){cmmtempr<-matrix(cmmtemp[,1:2],ncol=2)}
                  #cmmlbs<-cbind(znames,t(matrix(immlbs)))
                  rownames(cmmtempr)<-t(matrix(replicate(nrow(cmmtempr)," ")))
                  colnames(cmmtempr)<-cmmlbs
                  if (outscreen==1)
                  {cat("\n          Indices of conditional moderated mediation by W:\n")
                  print(cmmtempr,right=T)}
                }
              }
            }
          }
          if (((dichadj==1) | (immcat==1)) & (indmmm[i,1] < 30))
          {
            if (outscreen==1)
            {
              cat("\n          Index of moderated mediation\n")
              cat("          (differences beween conditional indirect effects):\n")
              rownames(immtempr)<-immlbs2
              print(immtempr,right=T)
            }
          }
        }
        if (indcontr==1)
        {
          resultm2<-matrix(99999,nrow(condcont),maxresm)
          resultm2[1:nrow(condcont),1:ncol(condcont)]<-condcont
          resultm<-rbind(resultm,resultm2)
          condcnt3<-noquote(matrix(sprintf(decimals,condcont),nrow=nrow(condcont)))
          condctlb<-c("Effect1","Effect2","Contrast","BootSE", "BootLLCI","BootULCI")
          colnames(condcnt3)<-condctlb
          rownames(condcnt3)<-t(matrix(replicate(nrow(condcnt3)," ")))
          if (outscreen==1)
          {
            cat("\n          Pairwise contrasts between conditional indirect effects\n")
            cat("          (Effect1 minus Effect2):\n")
            print(condcnt3,right=T)
          }
        }
        if (outscreen==1)
        {cat("\n---\n")}
      }
    }
    #end of moderated */
  }
}
# This is the end of the moderated loop */
```

```
}


if ((criterr==0) & (saveboot==1))
{
  if (boot > 0)
  {
    if (outscreen==1)
    {
      cat("\n*********************************************************************** \n")
      conseq<-conseq[2:nrow(conseq),]
      #cat("Bootstrap estimates were saved in a global dataframe named process.boots\n")
      colslab<-cbind(savlabs,conseq,vlabs)
      cat("\nMap of column names to model coefficients in bootstrap matrix:\n")
      rownames(colslab)<-t(matrix(replicate(nrow(colslab)," ")))
      colnames(colslab)<-c("name","Conseqnt","Antecdnt")
      print(noquote(colslab),right=T)
    }
  }
}

#PRINT BOOTSTRAP RESULTS FOR MODEL PARAMETERS
if ((criterr==0) & (boot > 0) & (modelbt==1))
{
  labstart<-1
  if (outscreen==1)
  {cat("\n********** BOOTSTRAP RESULTS FOR REGRESSION MODEL PARAMETERS **********\n\n")}
  for (iboot in (1:(nms+nys)))
  {
    vlabsm<-matrix(vlabs[labstart:(labstart+(nump[1,iboot]-1)),1])
    bootcimt<-matrix(bootcim[labstart:(labstart+(nump[1,iboot]-1)),],ncol=5)
    resultm2<-matrix(99999,nrow(bootcimt),maxresm)
    resultm2[1:nrow(bootcimt),1:ncol(bootcimt)]<-bootcimt
    resultm<-rbind(resultm,resultm2)
    bootcimt<-noquote(matrix(sprintf(decimals,bootcimt),ncol=5))
    colnames(bootcimt)<-matrix(c("Coeff","BootMean","BootSE","BootLLCI","BootULCI"))
    rownames(bootcimt)<-vlabsm
    if (outscreen==1)
    {
      cat("Outcome variable: ")
      write.table(outnames[iboot,1],quote=FALSE,row.names=FALSE,col.names=FALSE)
      cat("\n")
      print(bootcimt,right=T)
    }
    labstart<-labstart+nump[1,iboot]
    if ((iboot < (nms+nys)) & (outscreen==1)){cat("----------\n")}
  }
}

#print model matrices
if ((criterr==0) & (matrices==1) & (outscreen==1))
{
  cat("\n********************* MODEL DEFINITION MATRICES *********************** \n \n")
  cat("FROM variables are columns, TO variables are rows. \n \n")
  temp2<-matrix("0",nrow(bcmat),ncol(bcmat))
```

```
for (i in c(2:nrow(bcmat)))
{for (j in c(1:(ncol(bcmat)-1)))
  {if (bcmat[i,j]==1){temp2[i,j]<-″1″}
    if (j >= i){temp2[i,j]<-″ ″}}
}
temp2<-as.matrix(noquote(temp2[2:nrow(bcmat),(1:(ncol(bcmat)-1))]))
if (nms > 0){cmatlabs<-c(xnames,mnames);rmatlabs<-c(mnames,ynames)}
if (nms==0){cmatlabs<-xnames;rmatlabs<-ynames}
colnames(temp2)<-cmatlabs
rownames(temp2)<-rmatlabs
cat(″BMATRIX: Paths freely estimated (1) and fixed to zero (0): \n″)
print(temp2)
z<-0
if (sum(wcmat) !=0)
{
  temp2<-matrix(″0″,nrow(wcmat),ncol(wcmat))
  for (i in c(2:nrow(wcmat)))
  {for (j in c(1:(ncol(wcmat)-1)))
    {if (wcmat[i,j]==1){temp2[i,j]<-″1″}
      if (j >= i){temp2[i,j]<-″ ″}}
  }
  temp2<-as.matrix(noquote(temp2[2:nrow(wcmat),(1:(ncol(wcmat)-1))]))
  colnames(temp2)<-cmatlabs
  rownames(temp2)<-rmatlabs
  cat(″\nWMATRIX: Paths moderated (1) and not moderated (0) by W: \n″)
  print(temp2)
}
if (sum(zcmat) !=0)
{
  temp2<-matrix(″0″,nrow(zcmat),ncol(zcmat))
  for (i in c(2:nrow(zcmat)))
  {for (j in c(1:(ncol(zcmat)-1)))
    {if (zcmat[i,j]==1){temp2[i,j]<-″1″}
      if (j >= i){temp2[i,j]<-″ ″}}
  }
  temp2<-as.matrix(noquote(temp2[2:nrow(zcmat),(1:(ncol(zcmat)-1))]))
  colnames(temp2)<-cmatlabs
  rownames(temp2)<-rmatlabs
  cat(″\nZMATRIX: Paths moderated (1) and not moderated (0) by Z: \n″)
  print(temp2)
}
if (sum(wzcmat) !=0)
{
  temp2<-matrix(″0″,nrow(wzcmat),ncol(wzcmat))
  for (i in c(2:nrow(wzcmat)))
  {for (j in c(1:(ncol(wzcmat)-1)))
    {if (wzcmat[i,j]==1){temp2[i,j]<-″1″}
      if (j >= i){temp2[i,j]<-″ ″}}
  }
  temp2<-as.matrix(noquote(temp2[2:nrow(wzcmat),(1:(ncol(wzcmat)-1))]))
  colnames(temp2)<-cmatlabs
  rownames(temp2)<-rmatlabs
  cat(″\nWZMATRIX: W moderated paths moderated (1) and not moderated (0) by Z: \n″)
  print(temp2)
}
```

```
  if (ncs > 0)
  {colnames(ccmat)<-covnames
    rownames(ccmat)<-rmatlabs
    cat("\nCMATRIX: Covariates (columns) in (1) and not in (0) the models of M and
    print(ccmat)
  }
}
#end print model matrices

if (criterr==0)
{
  resultm<-resultm[2:nrow(resultm),]
  bocaj<-matrix(as.numeric(resultm==99999),nrow=nrow(resultm))
  bocaj<-(colSums(bocaj)==nrow(resultm))
  bocaj<-matrix(1-as.numeric(bocaj))
  j<-1
  for (i in (1:ncol(resultm)))
  {if (bocaj[i,1]==1){resultm[j,1]<-resultm[i,1];j=j+1}}
  resultm<-resultm[,1:(j-1)]
  resultm[resultm==99999]<-NA
  #if (saveest==1)
  #{assign("process.results", resultm, envir = .GlobalEnv)}
  #if ((outscreen==1) & (saveest==1))
  #{
  # cat("\n*********************************************************************** \n")
  # cat("\nprocess.results matrix saved to the global environment.\n")
  # cat("It looks like this but to more significant digits:\n\n")
  # print(noquote(matrix(sprintf(decimals,resultm),nrow=nrow(resultm))),right=T)
  #}
}
if ((outscreen==1) & (activate==0))
{cat("\n******************** ANALYSIS NOTES AND ERRORS ************************ \n")}
if (activate==1)
{cat("*********************************************************************** \n")}
if ((criterr==0) & (outscreen==1))
{
  cat("\nLevel of confidence for all confidence intervals in output: ")
  write.table(conf,quote=FALSE,row.names=FALSE,col.names=FALSE)
  if (boot>0)
  {
    if ((goodboot==boot) & (bc==0))
    {
      cat("\nNumber of bootstraps for percentile bootstrap confidence intervals: ")
      write.table(boot,quote=FALSE,row.names=FALSE,col.names=FALSE)
    }
    if ((goodboot==boot) & (bc==1))
    {
      cat("\nNumber of bootstraps for bias-corrected bootstrap confidence intervals: ")
      write.table(boot,quote=FALSE,row.names=FALSE,col.names=FALSE)
    }
    if (length(badend) > 1)
    {
      cat("\nWARNING: Bootstrap CI endpoints below are not trustworthy. Decrease confide
      cat("or increase the number of bootstrap samples.\n")
      badend<-matrix(badend);badend<-t(badend[2:nrow(badend),1])
      badend<-noquote(matrix(sprintf(decimals,badend),nrow=nrow(badend)))
      write.table(badend,quote=FALSE,row.names=FALSE,col.names=FALSE)
    }
  }
  if (mc > 0)
  {
    cat("\nNumber of samples for Monte Carlo confidence intervals: ")
    write.table(mc,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
  if ((wnotev > 0) & (printw==1))
  {
    if (wnotev==1)
    {cat("\nW values in conditional tables are the 16th, 50th, and 84th percentiles.\n
    if ((wnotev==2) & (minwwarn==0) & (maxwwarn==0))
    {cat("\nW values in conditional tables are the mean and +/- SD from the mean.\n"
    if (minwwarn==1)
    {cat("\nW values in conditional tables are the minimum, the mean, and 1 SD above
    if (maxwwarn==1)
    {cat("\nW values in conditional tables are 1 SD below the mean, the mean, and t
  }
  if ((znotev > 0) & (printz==1))
  {
    if (znotev==1)
    {cat("\nZ values in conditional tables are the 16th, 50th, and 84th percentiles.\n
    if ((znotev==2) & (minzwarn==0) & (maxzwarn==0))
    {cat("\nZ values in conditional tables are the mean and +/- SD from the mean.\n"
    if (minzwarn==1)
    {cat("\nZ values in conditional tables are the minimum, the mean, and 1 SD above
    if (maxzwarn==1)
    {cat("\nZ values in conditional tables are 1 SD below the mean, the mean, and t
  }
  if (minwwarn > 0)
  {cat("\nNOTE: One SD below the mean is below the minimum observed in the data for
    cat("      so the minimum measurement on W is used for conditioning instead.\
  if (maxwwarn > 0)
  {cat("\nNOTE: One SD above the mean is above the maximum observed in the data for
    cat("      so the maximum measurement on W is used for conditioning instead.\
  if (minzwarn > 0)
  {cat("\nNOTE: One SD below the mean is below the minimum observed in the data for
    cat("      so the minimum measurement on Z is used for conditioning instead.\
  if (maxzwarn > 0)
  {cat("\nNOTE: One SD above the mean is above the maximum observed in the data for
    cat("      so the maximum measurement on Z is used for conditioning instead.\
  if (pstog==1)
  {cat("\nNOTE: Standardized coefficients for dichotomous or multicategorical X are\n")
    cat("      in partially standardized form.\n")}
  for (i in c(1:100))
  {
  if (notecode[i,1]==1)
    {cat(" \n")
    cat("NOTE: COVMY is ignored when using CMATRIX option. \n")}
  if (notecode[i,1]==2)
    {cat(" \n")
    cat("NOTE: Confidence level restricted to between 50 and 99.9999%. 95% confidence
  if (notecode[i,1]==3)
    {cat(" \n")
```

```
          {cat("  \n")
          cat("NOTE: The following variables were mean centered prior to analysis: \n")
          centvar<-c("          ", centvar)
          centvar<-t(noquote(centvar))
          write.table(centvar,quote=FALSE,row.names=FALSE,col.names=FALSE)
          }
        if (notecode[i,1]==4)
          {cat("  \n")
          cat("NOTE: A heteroscedasticity consistent standard error and covariance matrix esti
        if (notecode[i,1]==6)
          {cat("  \n")
          cat("NOTE: Due to estimation problems, some bootstrap samples had to be replaced.
          cat("          The number of times this happened was: ")
          write.table(badboot,quote=FALSE,row.names=FALSE,col.names=FALSE)}
        if (notecode[i,1]==7)
          {cat("  \n")
          cat("NOTE: The bootstrapping was not completed due to problematic bootstrap samples
          cat("          Bootstrap confidence intervals are therefore suppressed.\n")}
        if (notecode[i,1]==8)
          {cat("  \n")
          cat("NOTE: The number of bootstrap samples was adjusted upward given your desired
        if (notecode[i,1]==9)
          {cat("  \n")
          cat("NOTE: WMODVAL is ignored when W is specified as multicategorical. \n")}
        if (notecode[i,1]==10)
          {cat("  \n")
          cat("NOTE: ZMODVAL is ignored when Z is specified as multicategorical. \n")}
        if (notecode[i,1]==11)
          {cat("  \n")
          cat("NOTE: Total effect model and estimate generated only when all covariates\n")
          cat("          are specified in all models of M and Y.\n")}
        if (notecode[i,1]==12)
          {cat("  \n")
          cat("NOTE: Total effect model and estimate generated only when X is freely estima
          cat("          to affect each M and both X and M are freely estimated to affec
        if (notecode[i,1]==13)
          {cat("  \n")
          cat("NOTE: There are too many pairwise contrasts to conduct with this model. \n")
        if (notecode[i,1]==14)
          {cat("  \n")
          cat("NOTE: The number of contrast weights must equal the number of indirect effec
        if (notecode[i,1]==15)
          {cat("  \n")
          cat("NOTE: Monte Carlo confidence intervals not available for this model. \n")
          cat("          Bootstrapping is used instead.\n")}
        if (notecode[i,1]==16)
          {cat("  \n")
          cat("NOTE: The number of Monte Carlo samples was adjusted upward given your desir
        if (notecode[i,1]==19)
          {cat("  \n")
          cat("NOTE: Your contrast matrix is invalid or not applicable to this model. \n")}
        if (notecode[i,1]==20)
          {cat("  \n")
          cat("NOTE: One of the groups specified by your contrast matrix does not exist in
        if (notecode[i,1]==24)
```

```
  {cat("  \n")
  cat("NOTE:  Total  effect  model  not  available  with  dichotomous  Y.  \n")}
 if  (notecode[i,1]==25)
  {cat("  \n")
  cat("NOTE:  Effect  size  option  not  available  with  dichotomous  Y.  \n")}
 if  (notecode[i,1]==27)
  {cat("  \n")
  cat("NOTE:  Standardized  coefficients  not  available  for  models  with  moderators.  \n")}
 if  (notecode[i,1]==28)
  {cat("  \n")
  cat("NOTE:  The  contrast  option  is  not  available  with  a  multicategorical  X.  \n")}
 }
 if  (nmiss  >  0)
 {cat("  \n")
  cat("NOTE:  Some  cases  with  missing  data  were  deleted.  The  number  of  deleted  cases
  write.table(nmiss,quote=FALSE,row.names=FALSE,col.names=FALSE)}
}

if  (outscreen==1)
{
#  print  errors
for  (i  in  c(1:100))
{
if  (errcode[i,1]==1)
 {cat("  \n")
 cat("ERROR:  You  must  specify  a  Y  and  an  X  variable.  \n")}
if  (errcode[i,1]==2)
 {cat("  \n")
 cat("ERROR:  X,  M,  or  Y  variable  used  more  than  once  or  W  and  Z  are  the  same  va
if  (errcode[i,1]==3)
 {cat("  \n")
 cat("ERROR:  You  have  specified  more  than  one  variable  for  W,  Y,  X,  or  Z.  \n")}
if  (errcode[i,1]==4)
 {cat("  \n")
 cat("ERROR:  A  variable  specified  as  multicategorical  has  more  than  nine  categories.
if  (errcode[i,1]==5)
 {cat("  \n")
 cat("ERROR:  One  of  the  categories  contains  only  a  single  case.  \n")}
if  (errcode[i,1]==6)
 {cat("  \n")
 cat("ERROR:  Invalid  model  number  in  this  version  of  PROCESS.  \n")}
if  (errcode[i,1]==7)
 {cat("  \n")
 cat("ERROR:  Invalid  model  number.  \n")}
if  (errcode[i,1]==8)
 {cat("  \n")
 cat("ERROR:  You  must  specify  an  M  variable  for  this  model.  \n")}
if  (errcode[i,1]==9)
 {cat("  \n")
 cat("ERROR:  You  have  specified  an  M  variable  in  a  model  that  does  not  use  it.  \n
if  (errcode[i,1]==10)
 {cat("  \n")
 cat("ERROR:  You  have  specified  a  W  variable  in  a  model  that  does  not  use  it.  \n"
if  (errcode[i,1]==11)
 {cat("  \n")
 cat("ERROR:  You  have  not  specified  a  W  variable  in  a  model  that  requires  it.  \n")
```

```
  cat("ERROR: You have not specified a W variable in a model that requires it. \n")
if (errcode[i,1]==12)
  {cat(" \n")
  cat("ERROR: You have specified a Z variable in a model that does not use it. \n"
if (errcode[i,1]==13)
  {cat(" \n")
  cat("ERROR: You have not specified a Z variable in a model that requires it. \n")
if (errcode[i,1]==15)
  {cat(" \n")
  cat("ERROR: One of your model variables exhibits no variation (it is a constant).
if (errcode[i,1]==16)
  {cat(" \n")
  cat("ERROR: BMATRIX is not the correct length or is otherwise invalid. \n")}
if (errcode[i,1]==17)
  {cat(" \n")
  cat("ERROR: WMATRIX is not the correct length or is otherwise invalid. \n")}
if (errcode[i,1]==18)
  {cat(" \n")
  cat("ERROR: ZMATRIX is not the correct length or is otherwise invalid. \n")}
if (errcode[i,1]==19)
  {cat(" \n")
  cat("ERROR: WZMATRIX is not the correct length or is otherwise invalid. \n")}
if (errcode[i,1]==20)
  {cat(" \n")
  cat("ERROR: A path fixed at zero cannot be moderated. \n")}
if (errcode[i,1]==60)
  {cat(" \n")
  cat("PROCESS is now ready for use.\n")
  cat("Copyright 2020 by Andrew F. Hayes ALL RIGHTS RESERVED\n \n")
  cat("Distribution of this beta release of PROCESS is prohibited\n")
  cat("without written authorization from the copyright holder.\n")}
if (errcode[i,1]==21)
  {cat(" \n")
  cat("ERROR: If only one moderator is specified, it must be specified as W. \n")}
if (errcode[i,1]==22)
  {cat(" \n")
  cat("ERROR: In BMATRIX, X must be specified to affect at least one variable. \n")}
if (errcode[i,1]==23)
  {cat(" \n")
  cat("ERROR: In BMATRIX, at least one variable must be specified to affect Y. \n")}
if (errcode[i,1]==24)
  {cat(" \n")
  cat("ERROR: You must specify a model number or a custom BMATRIX specification. \n")
if (errcode[i,1]==25)
  {cat(" \n")
  cat("ERROR: BMATRIX cannot be used in conjunction with a model number. \n")}
if (errcode[i,1]==26)
  {cat(" \n")
  cat("ERROR: Your model has a dangling mediator (all Ms must affect and be affected
if (errcode[i,1]==29)
  {cat(" \n")
  cat("ERROR: CMATRIX is not the correct length or is otherwise invalid. \n")}
if (errcode[i,1]==30)
  {cat(" \n")
  cat("ERROR: In CMATRIX, all covariates must be assigned to an M or a Y. \n")}
if ((errcode[i,1]==31) & (singlr==0))
```

```
    if ((errcode[i,1]==31) & (singlr==0))
     {cat(" \n")
     singlr<-1
     cat("ERROR: A linear or near linear dependence (singularity) exists in the data. \n
    if (errcode[i,1]==32)
     {cat(" \n")
     cat("ERROR: Models 80 and 81 require between 3 and 6 mediators. \n")}
    if (errcode[i,1]==33)
     {cat(" \n")
     cat("ERROR: Model 82 requires 4 mediators. \n")}
    if (errcode[i,1]==34)
     {cat(" \n")
     cat("ERROR: This model number requires between 2 and 6 mediators. \n")}
    if (errcode[i,1]==35)
     {cat(" \n")
     cat("ERROR: In a model with only one moderator, that moderator must be W. \n")}
    if (errcode[i,1]==36)
     {cat(" \n")
     cat("ERROR: A serial mediation model cannot have more than 6 mediators. \n")}
    if (errcode[i,1]==37)
     {cat(" \n")
     cat("ERROR: No more than 10 mediators are allowed in a PROCESS command. \n")}
    if (errcode[i,1]==38)
     {cat(" \n")
     cat("ERROR: XCATCODE is not provided, not the correct length, or is otherwise inval
    if (errcode[i,1]==39)
     {cat(" \n")
     cat("ERROR: WCATCODE is not provided, not the correct length, or is otherwise inval
    if (errcode[i,1]==40)
     {cat(" \n")
     cat("ERROR: ZCATCODE is not provided, not the correct length, or is otherwise inval
    if (errcode[i,1]==41)
     {cat(" \n")
     cat("ERROR: Models 1, 2, 3, and 74 cannot be customized. \n")}
    if (errcode[i,1]==43)
     {cat(" \n")
     cat("ERROR: PROCESS does not allow dichotomous mediators. \n")}
    if (errcode[i,1]==45)
     {cat(" \n")
     cat("ERROR: In model 74, X and W must be the same variable. \n")}
    if (errcode[i,1]==46)
     {cat(" \n")
     cat("ERROR: Model 74 is temporarily disabled in this release of PROCESS. \n")}
    if (errcode[i,1]==50)
     {cat(" \n")
     cat("ERROR: A multicategorical moderator cannot be specified as a covariate. \n")}
    if (errcode[i,1]==51)
     {cat(" \n")
     cat("ERROR: A variable you specified as a covariate is a moderator in all equation
    if ((errcode[i,1]==52) & (mcerpt==0))
     {mcerpt<-1
     cat(" \n")
     cat("ERROR: A variable specified as multicategorical must have at least three catego
    }
    }
    #if (saveboot==1)
```

```
  #if  (saveboot==1)
  #{resultm<-list(resultm,boots)}
  #invisible(resultm)
  resultms<-NULL
  if  ((saveboot==0)  &  (saveest==1)){resultms<-resultm}
  if  ((saveboot==1)  &  (saveest==0)){resultms<-boots}
  if  ((saveboot==1)  &  (saveest==1)){resultms<-(list(boots,resultm))}
  invisible(resultms)
}
process(activate=1)
```

```
***************** PROCESS for R Version 3.5.3 beta0.6 *****************

           Written by Andrew F. Hayes, Ph.D.  www.afhayes.com
   Documentation available in Hayes (2018). www.guilford.com/p/hayes3

************************************************************************

PROCESS is now ready for use.
Copyright 2020 by Andrew F. Hayes ALL RIGHTS RESERVED

Distribution of this beta release of PROCESS is prohibited
without written authorization from the copyright holder.
```

以上為"收合區段",按上方方格的左方三角型按鍵可展開或收合

```
%%R
model1<-process(data=myData,  y="網路成癮分數YDQ",  x='過動症總分',m=c('depression','anxiety','年齡
```

```
***************** PROCESS for R Version 3.5.3 beta0.6 *****************

           Written by Andrew F. Hayes, Ph.D.  www.afhayes.com
   Documentation available in Hayes (2018). www.guilford.com/p/hayes3

************************************************************************

Model : 4
    Y : 網路成癮分數YDQ
    X : 過動症總分
   M1 : depression
   M2 : anxiety
   M3 : 年齡

Sample size: 188

Random seed: 338771


************************************************************************
Outcome Variable: depression
```

```
Model Summary:
          R       R-sq        MSE          F        df1        df2          p
     0.3736     0.1396   110.6284    30.1690     1.0000   186.0000     0.0000

Model:
               coeff         se          t          p       LLCI       ULCI
constant      6.5728     1.3022     5.0474     0.0000     4.0038     9.1418
過動症總分    0.4267     0.0777     5.4926     0.0000     0.2734     0.5799

***********************************************************************
Outcome Variable: anxiety

Model Summary:
          R       R-sq        MSE          F        df1        df2          p
     0.3304     0.1091    60.9954    22.7878     1.0000   186.0000     0.0000

Model:
               coeff         se          t          p       LLCI       ULCI
constant     25.2994     0.9669    26.1644     0.0000    23.3918    27.2070
過動症總分    0.2754     0.0577     4.7737     0.0000     0.1616     0.3891

***********************************************************************
Outcome Variable: 年齡

Model Summary:
          R       R-sq        MSE          F        df1        df2          p
     0.0842     0.0071    18.9651     1.3282     1.0000   186.0000     0.2506

Model:
               coeff         se          t          p       LLCI       ULCI
constant     20.8310     0.5392    38.6350     0.0000    19.7673    21.8946
過動症總分    0.0371     0.0322     1.1525     0.2506    -0.0264     0.1005

***********************************************************************
Outcome Variable: 網路成癮分數YDQ
```