Doi: https://doi.org/10.1027/1614-2241/a000177

Electronic Supplementary Material

Appendix A

Appendix A1. R2OpenBUGS and coda syntax with diffuse generic priors

Line	Syntax
1	library(R2OpenBUGS)
2	library(readr)
3	DANS_std <- read_csv("E:/DANS_std.csv") # specify location and name of the data set
4	modelstring <- as.character("
5	model{
6	# Prior distributions
7	# Measurement model
8	# Intercepts are not used in the example analysis because x1-y6 are z-scores
9	# If your items are not z-scores, remove comments from lines 10-18
10	# nu.x11 ~ dnorm(0, 0.001); # intercept of x1
11	# nu.x21 ~ dnorm(0, 0.001); # intercept of x2
12	# nu.x31 ~ dnorm(0, 0.001); # intercept of x3
13	# nu.y11 ~ dnorm(0, 0.001); # intercept of y1
14	# nu.y21 ~ dnorm(0, 0.001); # intercept of y2
15	# nu.y31 ~ dnorm(0, 0.001); # intercept of y3
16	# nu.y42 ~ dnorm(0, 0.001); # intercept of y4
17	# nu.y52 ~ dnorm(0, 0.001); # intercept of y5
18	# nu.y62 ~ dnorm(0, 0.001); # intercept of y6
19	lam.x21 ~ dnorm(1, 0.001); # loading of x2
20	lam.x31 ~ dnorm(1, 0.001); # loading of x3
21	lam.y21 ~ dnorm(1, 0.001); # loading of y2
22	lam.y31 ~ dnorm(1, 0.001); # loading of y3
23	lam.y52 ~ dnorm(1, 0.001); # loading of y5
24	lam.y62 ~ dnorm(1, 0.001); # loading of y6
25	tau.x1 ~ dgamma(.5, .5); # error precision of x1
26	tau.x2 ~ dgamma(.5, .5); # error precision of x2
27	tau.x3 ~ dgamma(.5, .5); # error precision of x3
28	tau.y1 ~ dgamma(.5, .5); # error precision of y1
29	tau.y2 ~ dgamma(.5, .5); # error precision of y2
30	tau.y3 ~ dgamma(.5, .5); # error precision of y3

```
tau.y4 ~ dgamma(.5, .5); # error precision of y4
tau.y5 ~ dgamma(.5, .5); # error precision of y5
tau.y6 ~ dgamma(.5, .5); # error precision of y6
# Structural model
phi11 ~ dgamma(.5, .5); # precision of the independent variable
psi11 ~ dgamma(.5, .5); # residual precision of the mediator
psi22 ~ dgamma(.5, .5); # residual precision of the dependent variable
gamma11 ~ dnorm(0, .001); # path between the independent variable and mediator
beta21 ~ dnorm(0, .001); # path between the mediator and dependent variable
gamma21 ~ dnorm(0, .001); # path between the independent and dependent variables
 indirect.effect <-gamma11*beta21; # computation of the indirect effect
# Conditional probability of the data
for(i in 1:n){
x1.mean[i] <- 1*Ksi1[i];
x2.mean[i] <- lam.x21*Ksi1[i];
x3.mean[i] <- lam.x31*Ksi1[i];
y1.mean[i] <- 1*Eta1[i];
y2.mean[i] <- lam.y21*Eta1[i];
y3.mean[i] <- lam.y31*Eta1[i];
y4.mean[i] <- 1*Eta2[i];
y5.mean[i] <- lam.y52*Eta2[i];
y6.mean[i] <- lam.y62*Eta2[i];
Ksi1[i] ~ dnorm(0,phi11)
x1[i] ~ dnorm(x1.mean[i], tau.x1);
x2[i] ~ dnorm(x2.mean[i], tau.x2);
x3[i] ~ dnorm(x3.mean[i], tau.x3);
y1[i] ~ dnorm(y1.mean[i], tau.y1);
y2[i] ~ dnorm(y2.mean[i], tau.y2);
y3[i] ~ dnorm(y3.mean[i], tau.y3);
y4[i] ~ dnorm(y4.mean[i], tau.y4);
y5[i] ~ dnorm(y5.mean[i], tau.y5);
y6[i] ~ dnorm(y6.mean[i], tau.y6);
Eta1.mean[i] <- gamma11*Ksi1[i];
Eta2.mean[i] <- gamma21*Ksi1[i] + beta21*Eta1[i];
Eta1[i] ~ dnorm(Eta1.mean[i],psi11)
Eta2[i] ~ dnorm(Eta2.mean[i],psi22)
}}) # closes the model as string
```

```
 # Write out the BUGS code to a file
BUGS.code.file.name <- "Diffuse.txt"
write(modelstring, BUGS.code.file.name)
write(modelstring, "temp.bug")
# Define data to give to BUGS
# Change gen_dat to the name of your data set
# The code does not accommodate missing values
raw.data = DANS_std # specify the name of the data set
n <- nrow(raw.data); # calculates the number of cases in the data
x1 = raw.data$z_prettig
x2 = raw.data$z_sympathiek
x3 = raw.data$z_prikkelbaar_r
y1 = raw.data$z_aggressive
y2 = raw.data$z_ridicules
y3 = raw.data$z_impatient
y4 = raw.data$z_critizied
y5 = raw.data$z_refuse_request
y6 = raw.data$z_accept_r
data <- list("n", "x1", "x2", "x3", "y1", "y2", "y3", "y4", "y5", "y6")
# Define parameters to monitor in BUGS
parameters <- c("lam.x21", "lam.x31", "tau.x1", "tau.x2", "tau.x3",
"lam.y21", "lam.y31", "tau.y1", "tau.y2", "tau.y3",
"lam.y52", "lam.y62", "tau.y4", "tau.y5", "tau.y6",
# Remove comments from lines 92 and 93 if your measurement equation has intercepts
# "nu.x11", "nu.x21", "nu.x31", "nu.y11", "nu.y21",
# "nu.y31", "nu.y42", "nu.y52", "nu.y62",
"phi11", "psi11", "psi22",
"gamma11", "beta21", "gamma21", "indirect.effect")
# Define initial values to give to OpenBUGS for each of 3 chains
lam.x2.inits.1 = 0
lam.x3.inits.1 = 0.3
tau.x1.inits.1 = 2
tau.x2.inits.1 = 1
tau.x3.inits.1 = .5
lam.y2.inits.1 = 0
lam.y3.inits.1 = 0.3
```

```
tau.y1.inits.1 = 2
tau.y2.inits.1 = 1
tau.y3.inits.1 = .5
lam.y5.inits.1 = 0
lam.y6.inits.1 = 0.3
tau.y4.inits.1 = 2
tau.y5.inits.1 = 1
tau.y6.inits.1 = .5
# Remove comments from lines 113-121 if your measurement equation has intercepts
#nu.x11.inits.1=12
#nu.x21.inits.1=13
#nu.x31.inits.1=14
#nu.y11.inits.1=15
#nu.y21.inits.1=16
#nu.y31.inits.1=17
#nu.y42.inits.1=18
#nu.y52.inits.1=19
#nu.y62.inits.1=20
phi11.inits.1= .5
psi11.inits.1= .5
psi22.inits.1= .5
gamma11.inits.1= .3
beta21.inits.1= .4
gamma21.inits.1= .7
inits1 <- list(lam.x21=lam.x2.inits.1, lam.x31=lam.x3.inits.1,
tau.x1=tau.x1.inits.1, tau.x2=tau.x2.inits.1, tau.x3=tau.x3.inits.1,
lam.y21=lam.y2.inits.1, lam.y31=lam.y3.inits.1,
tau.y1=tau.y1.inits.1, tau.y2=tau.y2.inits.1, tau.y3=tau.y3.inits.1,
lam.y52=lam.y5.inits.1, lam.y62=lam.y6.inits.1,
tau.y4=tau.y4.inits.1, tau.y5=tau.y5.inits.1, tau.y6=tau.y6.inits.1,
# Remove comments from lines 135-137 if your measurement model has intercepts
#nu.x11=nu.x11.inits.1, nu.x21=nu.x21.inits.1, nu.x31=nu.x31.inits.1,
#nu.y11=nu.y11.inits.1, nu.y21=nu.y21.inits.1, nu.y31=nu.y31.inits.1,
# nu.y42=nu.y42.inits.1, nu.y52=nu.y52.inits.1, nu.y62=nu.y62.inits.1,
phi11=phi11.inits.1, psi11=psi11.inits.1,
psi22=psi22.inits.1, gamma11=gamma11.inits.1, beta21=beta21.inits.1,
gamma21=gamma21.inits.1)
```

```
lam.x2.inits.2 = 0.3
lam.x3.inits.2 = 0.1
tau.x1.inits.2 = .5
tau.x2.inits.2 = 2
tau.x3.inits.2 = 1
lam.y2.inits.2 = 0.3
lam.y3.inits.2 = 0
tau.y1.inits.2 = 1
tau.y2.inits.2 = .5
tau.y3.inits.2 = 2
lam.y5.inits.2 = 0.3
lam.y6.inits.2 = 0
tau.y4.inits.2 = 1
tau.y5.inits.2 = .5
tau.y6.inits.2 = 2
# Remove comments from lines 157-165 if your measurement equation has intercepts
#nu.x11.inits.2=-12
#nu.x21.inits.2=-13
#nu.x31.inits.2=-14
#nu.y11.inits.2=-15
#nu.y21.inits.2=-16
#nu.y31.inits.2=-17
#nu.y42.inits.2=-18
#nu.y52.inits.2=-19
#nu.y62.inits.2=-20
psi11.inits.2= 1
psi22.inits.2= 1
gamma11.inits.2= .7
beta21.inits.2= .3
gamma21.inits.2= .4
inits2 <- list(lam.x21=lam.x2.inits.2, lam.x31=lam.x3.inits.2,
tau.x1=tau.x1.inits.2, tau.x2=tau.x2.inits.2, tau.x3=tau.x3.inits.2,
lam.y21=lam.y2.inits.2, lam.y31=lam.y3.inits.2,
tau.y1=tau.y1.inits.2, tau.y2=tau.y2.inits.2, tau.y3=tau.y3.inits.2,
lam.y52=lam.y5.inits.2, lam.y62=lam.y6.inits.2,
tau.y4=tau.y4.inits.2, tau.y5=tau.y5.inits.2, tau.y6=tau.y6.inits.2,
# Remove comments from lines 178-180 if your measurement model has intercepts
```

```
#nu.x11=nu.x11.inits.2, nu.x21=nu.x21.inits.2, nu.x31=nu.x31.inits.2,
#nu.y11=nu.y11.inits.2, nu.y21=nu.y21.inits.2, nu.y31=nu.y31.inits.2,
#nu.y42=nu.y42.inits.2, nu.y52=nu.y52.inits.2, nu.y62=nu.y62.inits.2,
phi11=phi11.inits.2, psi11=psi11.inits.2,
psi22=psi22.inits.2, gamma11=gamma11.inits.2, beta21=beta21.inits.2,
gamma21=gamma21.inits.2)
lam.x2.inits.3 = 0.01
lam.x3.inits.3 = 0
tau.x1.inits.3 = 1
tau.x2.inits.3 = .5
tau.x3.inits.3 = 2
lam.y2.inits.3 = 0.3
lam.y3.inits.3 = 0.7
tau.y1.inits.3 = 2.4
tau.y2.inits.3 = 1.7
tau.y3.inits.3 = .2
lam.y5.inits.3 = 0.3
lam.y6.inits.3 = 3
tau.y4.inits.3 = 2.5
tau.y5.inits.3 = 1.7
tau.y6.inits.3 = .85
# Remove comments from lines 200-208 if your measurement equation has intercepts
#nu.x11.inits.3=0
#nu.x21.inits.3=0
#nu.x31.inits.3=0
#nu.y11.inits.3=0
#nu.y21.inits.3=0
#nu.y31.inits.3=0
#nu.y42.inits.3=0
#nu.y52.inits.3=0
#nu.y62.inits.3=0
psi11.inits.3= .75
psi22.inits.3= 3.5
gamma11.inits.3= 2.3
beta21.inits.3= 1.4
gamma21.inits.3= .2
inits3 <- list(lam.x21=lam.x2.inits.3, lam.x31=lam.x3.inits.3,
```

```
tau.x1=tau.x1.inits.3, tau.x2=tau.x2.inits.3, tau.x3=tau.x3.inits.3,
lam.y21=lam.y2.inits.3, lam.y31=lam.y3.inits.3,
tau.y1=tau.y1.inits.3, tau.y2=tau.y2.inits.3, tau.y3=tau.y3.inits.3,
lam.y52=lam.y5.inits.3, lam.y62=lam.y6.inits.3,
tau.y4=tau.y4.inits.3, tau.y5=tau.y5.inits.3, tau.y6=tau.y6.inits.3,
# Remove comments from lines 221-223 if your measurement model has intercepts
#nu.x11=nu.x11.inits.3, nu.x21=nu.x21.inits.3, nu.x31=nu.x31.inits.3,
#nu.y11=nu.y11.inits.3, nu.y21=nu.y21.inits.3, nu.y31=nu.y31.inits.3,
#nu.y42=nu.y42.inits.3, nu.y52=nu.y52.inits.3, nu.y62=nu.y62.inits.3,
phi11=phi11.inits.3, psi11=psi11.inits.3,
psi22=psi22.inits.3, gamma11=gamma11.inits.3, beta21=beta21.inits.3,
gamma21=gamma21.inits.3)
inits <- list(inits1, inits2, inits3)
#   Choose
#   the number of chains
#   the number of iterations to burn-in,
#   the number of iterations to thin by,
#   the total number of iterations
n.chains = 3
n.burnin = 25000
n.thin = 1
n.iters.total.per.chain = 50000
# Call OpenBUGS to run the model
model.in.openbugs <- bugs(
data=data,
inits=inits,
parameters.to.save=parameters,
model.file=BUGS.code.file.name,
n.chains=n.chains,
n.iter=n.iters.total.per.chain,
n.burnin=n.burnin,
n.thin=n.thin,
debug=FALSE,
codaPkg=TRUE,
OpenBUGS.pgm=NULL,
working.directory=getwd(),
```

```
clearWD=FALSE, useWINE=FALSE, WINE=NULL,
newWINE=TRUE, WINEPATH=NULL, bugs.seed=1,
DIC=TRUE )
# This code reads in the draws from BUGS
library(coda)
coda.file.names <- model.in.openbugs
draws.from.bugs <- read.bugs(coda.file.names)
# Obtain trace plots and density plots and the PSRF
plot(draws.from.bugs, trace=TRUE, density = TRUE)
gelman.diag(draws.from.bugs)
# Combine all chains
coda.options(combine.stats=TRUE, combine.plots=TRUE)
# Summarize the posteriors
# Mean, median, and equal-tail credibility intervals
summary(draws.from.bugs)
# HPD intervals
str(draws.from.bugs)
draws.to.analyze.as.one.list <- as.mcmc(do.call(rbind,draws.from.bugs))
str(draws.to.analyze.as.one.list)
HPDinterval(draws.to.analyze.as.one.list, prob=.95)
# DIC
log.file<-paste(getwd(), "/log.txt", sep="")
bugs.log(log.file)
# Probability that the indirect effect is greater than a meaningful value ("criterion")
criterion<- 0
draws.to.analyze.as.one.list <- as.mcmc(do.call(rbind,draws.from.bugs))
all.draws<-as.data.frame(draws.to.analyze.as.one.list)
ind.draws<-all.draws$indirect.effect
prob.greater.than.criterion<-sum(ind.draws >= criterion )/length(ind.draws)
```

Appendix A2. Changes in R2OpenBUGS code to specify diffuse conjugate priors

```
6  # Prior distributions
7  # Measurement model
8  # Intercepts are not used in the example analysis because x1-y6 are z-scores
9  # If your items are not z-scores, remove comments from lines 10-18
10 # nu.x11 ~ dnorm(0, 0.001); # intercept of x1
11 # nu.x21 ~ dnorm(0, 0.001); # intercept of x2
12 # nu.x31 ~ dnorm(0, 0.001); # intercept of x3
13 # nu.y11 ~ dnorm(0, 0.001); # intercept of y1
14 # nu.y21 ~ dnorm(0, 0.001); # intercept of y2
15 # nu.y31 ~ dnorm(0, 0.001); # intercept of y3
16 # nu.y42 ~ dnorm(0, 0.001); # intercept of y4
17 # nu.y52 ~ dnorm(0, 0.001); # intercept of y5
18 # nu.y62 ~ dnorm(0, 0.001); # intercept of y6
19 lam.x21 ~ dnorm(1, tau.x2); # loading of x2
20 lam.x31 ~ dnorm(1, tau.x3); # loading of x3
21 lam.y21 ~ dnorm(1, tau.y2); # loading of y2
22 lam.y31 ~ dnorm(1, tau.y3); # loading of y3
23 lam.y52 ~ dnorm(1, tau.y5); # loading of y5
24 lam.y62 ~ dnorm(1, tau.y6); # loading of y6
25 tau.x1 ~ dgamma(.5, .5); # error precision of x1
26 tau.x2 ~ dgamma(.5, .5); # error precision of x2
27 tau.x3 ~ dgamma(.5, .5); # error precision of x3
28 tau.y1 ~ dgamma(.5, .5); # error precision of y1
29 tau.y2 ~ dgamma(.5, .5); # error precision of y2
30 tau.y3 ~ dgamma(.5, .5); # error precision of y3
31 tau.y4 ~ dgamma(.5, .5); # error precision of y4
32 tau.y5 ~ dgamma(.5, .5); # error precision of y5
33 tau.y6 ~ dgamma(.5, .5); # error precision of y6
34 # Structural model
35 phi11 ~ dgamma(.5, .5); # precision of the independent variable
36 psi11 ~ dgamma(.5, .5); # residual precision of the mediator
37 psi22 ~ dgamma(.5, .5); # residual precision of the dependent variable
38 gamma11 ~ dnorm(0, phi11); # path between the independent variable and mediator
39 beta21 ~ dnorm(0, psi11); # path between the mediator and dependent variable
40 gamma21 ~ dnorm(0, psi22); # path between the independent and dependent variables
```

Appendix A3. Changes in R2OpenBUGS code to specify informative priors

```
6  # Prior distributions
7  # Measurement model
8  # Intercepts are not used in the example analysis because x1-y6 are z-scores
9  # If your items are not z-scores, remove comments from lines 10-18
10 # nu.x11 ~ dnorm(0, 0.444); # intercept of x1
11 # nu.x21 ~ dnorm(0, 0.444); # intercept of x2
12 # nu.x31 ~ dnorm(0, 0.444); # intercept of x3
13 # nu.y11 ~ dnorm(0, 0.444); # intercept of y1
14 # nu.y21 ~ dnorm(0, 0.444); # intercept of y2
15 # nu.y31 ~ dnorm(0, 0.444); # intercept of y3
16 # nu.y42 ~ dnorm(0, 0.444); # intercept of y4
17 # nu.y52 ~ dnorm(0, 0.444); # intercept of y5
18 # nu.y62 ~ dnorm(0, 0.444); # intercept of y6
19 lam.x21 ~ dnorm(1, 0.444); # loading of x2
20 lam.x31 ~ dnorm(1, 0.444); # loading of x3
21 lam.y21 ~ dnorm(1, 0.444); # loading of y2
22 lam.y31 ~ dnorm(1, 0.444); # loading of y3
23 lam.y52 ~ dnorm(1, 0.444); # loading of y5
24 lam.y62 ~ dnorm(1, 0.444); # loading of y6
25 tau.x1 ~ dgamma(.5, .5); # error precision of x1
26 tau.x2 ~ dgamma(.5, .5); # error precision of x2
27 tau.x3 ~ dgamma(.5, .5); # error precision of x3
28 tau.y1 ~ dgamma(.5, .5); # error precision of y1
29 tau.y2 ~ dgamma(.5, .5); # error precision of y2
30 tau.y3 ~ dgamma(.5, .5); # error precision of y3
31 tau.y4 ~ dgamma(.5, .5); # error precision of y4
32 tau.y5 ~ dgamma(.5, .5); # error precision of y5
33 tau.y6 ~ dgamma(.5, .5); # error precision of y6
34 # Structural model
35 phi11 ~ dgamma(.5, .5); # precision of the independent variable
36 psi11 ~ dgamma(.5, .5); # residual precision of the mediator
37 psi22 ~ dgamma(.5, .5); # residual precision of the dependent variable
38 gamma11 ~ dnorm(0, 0.444); # path between the independent variable and mediator
```

```
39  beta21 ~ dnorm(0, 0.444); # path between the mediator and dependent variable
40  gamma21 ~ dnorm(0, 0.444); # path between the independent and dependent variables
```

Appendix B1. Mplus syntax with diffuse generic priors

Line	Syntax
1	TITLE: Bayesian analysis with diffuse priors
2	DATA: FILE IS DANS_std_nonames.csv;
3	VARIABLE: NAMES ARE z_prettig z_sympathiek z_prikkelbaar_r z_aggressive
4	z_ridicules z_impatient z_critizied z_refuse_request z_accept_r;
5	USEVARIABLES ARE ALL;
6	analysis:
7	estimator = bayes;
8	chains = 3;
9	proc = 2;
10	biter = 25000;
11	fbiter = 50000;
12	thin = 1;
13	model:
14	ksi1 BY x1@1;
15	ksi1 BY x2*3(lamx21);
16	ksi1 BY x3*5 (lamx31);
17	eta1 BY y1@1;
18	eta1 BY y2*2 (lamy21);
19	eta1 BY y3*9 (lamy31);
20	eta2 BY y4@1;
21	eta2 BY y5*4 (lamy52);
22	eta2 BY y6*7 (lamy62);
23	eta1 ON ksi1*4 (gamma11);
24	eta2 ON ksi1*3 (gamma21);
25	eta2 ON eta1*2 (beta21);
26	[x1@0]; [x2@0]; [x3@0];
27	[y1@0]; [y2@0]; [y3@0];
28	[y4@0]; [y5@0]; [y6@0];
29	[ksi1@0]; [eta1@0]; [eta2@0];
30	! Assigning starting values for each parameter

```
31      x1*7 (varx1);
32      x2*7 (varx2);
33      x3*7 (varx3);
34      y1*7 (vary1);
35      y2*7 (vary2);
36      y3*7 (vary3);
37      y4*7 (vary4);
38      y5*7 (vary5);
39      y6*7 (vary6);
40      ksi1*7 (phi11);
41      eta1*7 (psi11);
42      eta2*7 (psi22);
43    ! Computing the indirect effect
44    MODEL CONSTRAINT:
45    NEW(ind);
46    ind=gamma11*beta21;
47    !Assigning prior distributions
48    model priors:
49    lamx21 ~ N(1,1000); ! loading of x2
50    lamx31 ~ N(1,1000); ! loading of x3
51    lamy21 ~ N(1,1000); ! loading of y2
52    lamy31 ~ N(1,1000); ! loading of y3
53    lamy52 ~ N(1,1000); ! loading of y5
54    lamy62 ~ N(1,1000); ! loading of y6
55    varx1 ~ IG(.5,.5); ! error variance of x1
56    varx2 ~ IG(.5,.5); ! error variance of x2
57    varx3 ~ IG(.5,.5); ! error variance of x3
58    vary1 ~ IG(.5,.5); ! error variance of y1
59    vary2 ~ IG(.5,.5); ! error variance of y2
60    vary3 ~ IG(.5,.5); ! error variance of y3
61    vary4 ~ IG(.5,.5); ! error variance of y4
62    vary5 ~ IG(.5,.5); ! error variance of y5
63    vary6 ~ IG(.5,.5); ! error variance of y6
64    gamma11 ~ N(0,1000); ! path between the independent variable and mediator
65    gamma21 ~ N(0,1000); ! path between the independent and dependent variables
66    beta21 ~ N(0,1000); ! path between the mediator and dependent variable
```

```
67  phi11 ~ IG(.5,.5); ! variance of the independent variable
68  psi11 ~ IG(.5,.5); ! residual variance of the mediator
69  psi22 ~ IG(.5,.5); ! residual variance of the dependent variable
70  output: tech1 tech8 stdy svalues cint(hpd);
71  plot: type = plot2;
72  savedata: bparameters=mplus_results_diffuse.txt; !export draws
```

Appendix B2. Mplus syntax with informative priors

```
Line  Syntax
1     TITLE: Bayesian analysis with informative priors
2     DATA: FILE IS DANS_std_nonames.csv;
3     VARIABLE: NAMES ARE z_prettig z_sympathiek z_prikkelbaar_r z_aggressive
4     z_ridicules z_impatient z_critizied z_refuse_request z_accept_r;
5     USEVARIABLES ARE ALL;
6     analysis:
7     estimator = bayes;
8     chains = 3;
9     proc = 2;
10    biter = 25000;
11    fbiter = 50000;
12    thin = 1;
13    model:
14    ksi1 BY x1@1;
15    ksi1 BY x2*3(lamx21);
16    ksi1 BY x3*5 (lamx31);
17    eta1 BY y1@1;
18    eta1 BY y2*2 (lamy21);
19    eta1 BY y3*9 (lamy31);
20    eta2 BY y4@1;
21    eta2 BY y5*4 (lamy52);
22    eta2 BY y6*7 (lamy62);
23    eta1 ON ksi1*4 (gamma11);
24    eta2 ON ksi1*3 (gamma21);
25    eta2 ON eta1*2 (beta21);
```

```
[x1@0]; [x2@0]; [x3@0];
[y1@0]; [y2@0]; [y3@0];
[y4@0]; [y5@0]; [y6@0];
[ksi1@0]; [eta1@0]; [eta2@0];
! Assigning starting values for each parameter
x1*7 (varx1);
x2*7 (varx2);
x3*7 (varx3);
y1*7 (vary1);
y2*7 (vary2);
y3*7 (vary3);
y4*7 (vary4);
y5*7 (vary5);
y6*7 (vary6);
ksi1*7 (phi11);
eta1*7 (psi11);
eta2*7 (psi22);
! Computing the indirect effect
MODEL CONSTRAINT:
NEW(ind);
ind=gamma11*beta21;
!Assigning prior distributions
model priors:
lamx21 ~ N(1,2.25); ! loading of x2
lamx31 ~ N(1,2.25); ! loading of x3
lamy21 ~ N(1,2.25); ! loading of y2
lamy31 ~ N(1,2.25); ! loading of y3
lamy52 ~ N(1,2.25); ! loading of y5
lamy62 ~ N(1,2.25); ! loading of y6
varx1 ~ IG(.5,.5); ! error variance of x1
varx2 ~ IG(.5,.5); ! error variance of x2
varx3 ~ IG(.5,.5); ! error variance of x3
vary1 ~ IG(.5,.5); ! error variance of y1
vary2 ~ IG(.5,.5); ! error variance of y2
vary3 ~ IG(.5,.5); ! error variance of y3
vary4 ~ IG(.5,.5); ! error variance of y4
```

```
vary5 ~ IG(.5,.5); ! error variance of y5
vary6 ~ IG(.5,.5); ! error variance of y6
gamma11 ~ N(1,2.25); ! path between the independent variable and mediator
gamma21 ~ N(1,2.25); ! path between the independent and dependent variables
beta21 ~ N(1,2.25); ! path between the mediator and dependent variable
phi11 ~ IG(.5,.5); ! variance of the independent variable
psi11 ~ IG(.5,.5); ! residual variance of the mediator
psi22 ~ IG(.5,.5); ! residual variance of the dependent variable
output: tech1 tech8 stdy svalues cint(hpd);
plot: type = plot2;
savedata: bparameters=mplus_results_informative.txt; !export draws
```