# R Code First Example

```
library(lavaan)
library(xtable)

Sigma <- matrix(c(25,7.2,3.2,2,7.2,9,2,1.25,3.2,2,4,1.2,2,1.25,1.2,4),nrow=4)
rownames(Sigma) <- colnames(Sigma) <- paste0("X",1:4)
N <- 200
Sigma.rescale <- FALSE

# First Step: estimate and print for unrestricted model

# Fixed Marker Method
fixed.marker <- '
A =~ 1*X1+X2
B =~ 1*X3+X4'
res.marker <- cfa(fixed.marker,sample.cov=Sigma,sample.nobs=N,
                  sample.cov.rescale=Sigma.rescale)

# Fixed Factor Method
fixed.factor <- '
A =~ NA*X1+X2
B =~ NA*X3+X4
A ~~ 1*A
B ~~ 1*B'
res.factor <- cfa(fixed.factor,sample.cov=Sigma,sample.nobs=N,
                  sample.cov.rescale=Sigma.rescale)

# Effects Coding Method
effects.coding <- '
A =~ NA*X1 + a1*X1+a2*X2
a1 == 2-a2
B =~ NA*X3+b1*X3+b2*X4
b1 == 2-b2
A ~~ A
B ~~ B
A ~~ B
X1 ~~ X1
X2 ~~ X2
X3 ~~ X3
X4 ~~ X4'
```

```
res.effect <- cfa(effects.coding,sample.cov=Sigma,sample.nobs=N,
                  sample.cov.rescale=Sigma.rescale)

# Second Step: estimate and print for restricted model

# Fixed Marker Method
fixed.marker.restricted <- '
A =~ 1*X1+lambda*X2
B =~ 1*X3+lambda*X4'
res.marker.restricted <- cfa(fixed.marker.restricted,sample.cov=Sigma,
                             sample.nobs=N,sample.cov.rescale=Sigma.rescale)

# Fixed Factor Method
fixed.factor.restricted <- '
A =~ NA*X1+lambda*X2
B =~ NA*X3+lambda*X4
A ~~ 1*A
B ~~ 1*B'
res.factor.restricted <- cfa(fixed.factor.restricted,sample.cov=Sigma,
                             sample.nobs=N,sample.cov.rescale=Sigma.rescale)

# Effects Coding Method
effects.coding.restricted <- '
A =~ NA*X1+a1*X1+a2*X2
a1 == 2-a2
B =~ NA*X3+b1*X3+b2*X4
b1 == 2-b2
a2 == b2
A ~~ A
B ~~ B
A ~~ B
X1 ~~ X1
X2 ~~ X2
X3 ~~ X3
X4 ~~ X4'
res.effect.restricted <- cfa(effects.coding.restricted,sample.cov=Sigma,
                             sample.nobs=N,sample.cov.rescale=Sigma.rescale)
```

```
# Function for collecting several fit measures of unrestricted and restricted
# model, and for combining these (differences)
eval.restr <- function(resUnrestricted,resRestricted)
{
  measures <- c("chisq","df","pvalue","cfi","rmsea","srmr","logl")
  measures0 <- fitMeasures(resUnrestricted)[measures]
  measures1 <- fitMeasures(resRestricted)[measures]
  deltas <- measures1-measures0
  deltas["pvalue"] <- pchisq(deltas["chisq"],deltas["df"],lower.tail=FALSE)
  c(Unrestricted=measures0,Restricted=measures1,Difference=deltas)
}

mat <- cbind(# collecting fit measures of fixed marker, fixed factor,
             # and effects coding
              Marker=eval.restr(res.marker,res.marker.restricted),
              Factor=eval.restr(res.factor,res.factor.restricted),
              Effects=eval.restr(res.effect,res.effect.restricted))
colnames(mat) <- c("Marker","Factor","Effects")
print(round(mat,5))
```

# R Code Second Example

```
library(lavaan)
library(xtable)

true_model <- '
A =~ 2*X1+10*X2+8*X3+5*X4
B =~ 5*Y1+5*Y2+4*Y3+2*Y4
A ~~ 0.16*A
B ~~ 1*B # 2.56
A ~~ 0.096*B
X1 ~~ 3*X1
X2 ~~ 1*X2
X3 ~~ 4*X3
X4 ~~ 2*X4
Y1 ~~ 2*Y1
Y2 ~~ 7*Y2
Y3 ~~ 1*Y3
Y4 ~~ 8*Y4
X1 ~~ 0.2*Y1
X2 ~~ 0.5*Y2
X3 ~~ 0.25*Y3
X4 ~~ 0.5*Y4'
dat <- simulateData(true_model, model.type = "cfa",sample.nobs = 500,
                    seed = 123, std.lv = F, return.type = "data.frame",
                    return.fit = T)

Sigma <- matrix(NA,nrow=8,ncol=8)
rownames(Sigma) <- colnames(Sigma) <- c(paste0("X",1:4),paste0("Y",1:4))
Sigma[,] <- attr(dat,"fit")@implied$cov[[1]]
Sigma <- round(Sigma,3)
# print(Sigma)
# print(xtable(Sigma,digits=c(0,3,2,3,2,3,3,2,2)),include.rownames=FALSE)
N <- 150
Sigma.rescale <- FALSE
```

```
# First Step: estimate and print for unrestricted model

# Fixed Marker Method
fixed.marker1 <- '
A =~ 1*X1+X2+X3+X4
B =~ 1*Y1+Y2+Y3+Y4
X1 ~~ Y1
X2 ~~ Y2
X3 ~~ Y3
X4 ~~ Y4'
res.marker1 <- cfa(fixed.marker1,sample.cov=Sigma,sample.nobs=N,
                   sample.cov.rescale=Sigma.rescale)

fixed.marker3 <- '
A =~ NA*X1+NA*X2+1*X3+X4
B =~ NA*Y1+NA*Y2+1*Y3+Y4
X1 ~~ Y1
X2 ~~ Y2
X3 ~~ Y3
X4 ~~ Y4'
res.marker3 <- cfa(fixed.marker3,sample.cov=Sigma,sample.nobs=N,
                   sample.cov.rescale=Sigma.rescale)

fixed.marker4 <- '
A =~ NA*X1+NA*X2+NA*X3+1*X4
B =~ NA*Y1+NA*Y2+NA*Y3+1*Y4
X1 ~~ Y1
X2 ~~ Y2
X3 ~~ Y3
X4 ~~ Y4'
res.marker4 <- cfa(fixed.marker4,sample.cov=Sigma,sample.nobs=N,
                   sample.cov.rescale=Sigma.rescale)
```

```
# Fixed Factor Method
fixed.factor <- '
A =~ NA*X1+X2+X3+X4
B =~ NA*Y1+Y2+Y3+Y4
A ~~ 1*A
B ~~ 1*B
X1 ~~ Y1
X2 ~~ Y2
X3 ~~ Y3
X4 ~~ Y4'
res.factor <- cfa(fixed.factor,sample.cov=Sigma,sample.nobs=N,
                  sample.cov.rescale=Sigma.rescale)

# Effects Coding Method
effects.coding <- '
A =~ NA*X1+a1*X1+a2*X2+a3*X3+a4*X4
a1 == 4-a2-a3-a4
B =~ NA*Y1+b1*Y1+b2*Y2+b3*Y3+b4*Y4
b1 == 4-b2-b3-b4
A ~~ A
B ~~ B
A ~~ B
X1 ~~ X1
X2 ~~ X2
X3 ~~ X3
X4 ~~ X4
Y1 ~~ Y1
Y2 ~~ Y2
Y3 ~~ Y3
Y4 ~~ Y4
X1 ~~ Y1
X2 ~~ Y2
X3 ~~ Y3
X4 ~~ Y4'
res.effect <- cfa(effects.coding,sample.cov=Sigma,sample.nobs=N,
                  sample.cov.rescale=Sigma.rescale)
```

```
# Second Step: estimate and print for restricted model

# Fixed Marker Method
fixed.marker1.restricted <- '
A =~ 1*X1+lambda*X2+X3+X4
B =~ 1*Y1+lambda*Y2+Y3+Y4
X1 ~~ Y1
X2 ~~ Y2
X3 ~~ Y3
X4 ~~ Y4'
res.marker1.restricted <- cfa(fixed.marker1.restricted,sample.cov=Sigma,
                              sample.nobs=N,sample.cov.rescale=Sigma.rescale)

fixed.marker3.restricted <- '
A =~ NA*X1+lambda*X2+1*X3+X4
B =~ NA*Y1+lambda*Y2+1*Y3+Y4
X1 ~~ Y1
X2 ~~ Y2
X3 ~~ Y3
X4 ~~ Y4'
res.marker3.restricted <- cfa(fixed.marker3.restricted,sample.cov=Sigma,
                              sample.nobs=N,sample.cov.rescale=Sigma.rescale)

fixed.marker4.restricted <- '
A =~ NA*X1+lambda*X2+NA*X3+1*X4
B =~ NA*Y1+lambda*Y2+NA*Y3+1*Y4
X1 ~~ Y1
X2 ~~ Y2
X3 ~~ Y3
X4 ~~ Y4'
res.marker4.restricted <- cfa(fixed.marker4.restricted,sample.cov=Sigma,
                              sample.nobs=N,sample.cov.rescale=Sigma.rescale)
```

```
# Fixed Factor Method
fixed.factor.restricted <- '
A =~ NA*X1+lambda*X2+X3+X4
B =~ NA*Y1+lambda*Y2+Y3+Y4
A ~~ 1*A
B ~~ 1*B
X1 ~~ Y1
X2 ~~ Y2
X3 ~~ Y3
X4 ~~ Y4'
res.factor.restricted <- cfa(fixed.factor.restricted,sample.cov=Sigma,
                             sample.nobs=N,sample.cov.rescale=Sigma.rescale)

# Effects Coding Method
effects.coding.restricted <- '
A =~ NA*X1+a1*X1+a2*X2+a3*X3+a4*X4
a1 == 4-a2-a3-a4
B =~ NA*Y1+b1*Y1+b2*Y2+b3*Y3+b4*Y4
b1 == 4-b2-b3-b4
a2 == b2
A ~~ A
B ~~ B
A ~~ B
X1 ~~ X1
X2 ~~ X2
X3 ~~ X3
X4 ~~ X4
Y1 ~~ Y1
Y2 ~~ Y2
Y3 ~~ Y3
Y4 ~~ Y4
X1 ~~ Y1
X2 ~~ Y2
X3 ~~ Y3
X4 ~~ Y4'
res.effect.restricted <- lavaan(effects.coding.restricted,sample.cov=Sigma,
                                sample.nobs=N,sample.cov.rescale=Sigma.rescale)
```

```
# Function for collecting several fit measures of unrestricted and
# restricted model, and for combining these (differences)
eval.restr <- function(resUnrestricted,resRestricted)
{
  measures <- c("chisq","df","pvalue","cfi","rmsea","srmr","logl")
  measures0 <- fitMeasures(resUnrestricted)[measures]
  measures1 <- fitMeasures(resRestricted)[measures]
  deltas <- measures1-measures0
  deltas["pvalue"] <- pchisq(deltas["chisq"],deltas["df"],lower.tail=FALSE)
  c(Unrestricted=measures0,Restricted=measures1,Difference=deltas)
}

mat <- cbind(# collecting fit measures of fixed marker variants,
             # fixed factor, and effects coding
              Marker1=eval.restr(res.marker1,res.marker1.restricted),
              Marker3=eval.restr(res.marker3,res.marker3.restricted),
              Marker4=eval.restr(res.marker4,res.marker4.restricted),
              Factor=eval.restr(res.factor,res.factor.restricted),
              Effects=eval.restr(res.effect,res.effect.restricted))
colnames(mat) <- c("Marker 1","Marker 3","Marker 4","Factor","Effects")
print(round(mat,5))
```