Statistical Analysis Using Structural Equation Models

EPsy 8266

Christopher David Desjardins

Research Methodology Consulting Center

4/25/19

Measurement Invariance Topics

- Partial MI
- MI with ordinal data

Partial MI

- MI assumes that the unstandardized coefficients (patterns, intercepts, errors) are the same across the groups.
- However, not all the indicators may have the same coefficients across the groups.
  - For example, it might be possible that 4 of the 5 unstandardized pattern coefficients are invariant, while the 5th is not.
  - If this is the case, then this parameter can be left to be freely estimated across the groups.
- Having partial invariance is okay, but how much is unclear.
  - Probably a few is okay, but once the number of non-invariant indicators increases, there is less confidence that the constructs are operationalized the same way.

HS data

The Holzinger and Swineford (1939) data set consists of mental ability test scores of seventh- and eighth-grade children from two different schools (Pasteur and Grant-White) ($n = 301$). A subset of 9 indicators measures

- visual,
- verbal/textual, and
- mental speed abilities.

We are interested in comparing the two schools on these factors.

```r
library(lavaan)
config.mod <- "
  visual  =~ x1 + x2 + x3
  textual =~ x4 + x5 + x6
  speed   =~ x7 + x8 + x9
"
config.fit <- cfa(config.mod, HolzingerSwineford1939, group = "school")
fitmeasures(config.fit, c("chisq", "df", "pvalue", "cfi", "rmsea", "srmr"))
```

```
##   chisq      df  pvalue     cfi   rmsea    srmr
## 115.851  48.000   0.000   0.923   0.097   0.068
```

Correlation residuals

```
lavResiduals(config.fit, type = "cor")$Pasteur$cov
```

```
##    x1     x2     x3     x4     x5     x6     x7     x8     x9
## x1  0.000
## x2 -0.021  0.000
## x3 -0.004  0.162  0.000
## x4  0.065 -0.071 -0.081  0.000
## x5 -0.062 -0.050 -0.169  0.014  0.000
## x6  0.067  0.024 -0.015 -0.024  0.006  0.000
## x7 -0.109 -0.209 -0.104  0.112 -0.031  0.056  0.000
## x8 -0.037 -0.036  0.029 -0.051 -0.071  0.028  0.031  0.000
## x9  0.155  0.125  0.205 -0.011  0.008  0.040 -0.033 -0.011  0.000
```

```
lavResiduals(config.fit, type = "cor")$`Grant-White`$cov
```

```
##    x1     x2     x3     x4     x5     x6     x7     x8     x9
## x1  0.000
## x2 -0.024  0.000
## x3 -0.021  0.059  0.000
## x4  0.025 -0.014  0.003  0.000
## x5  0.006 -0.072 -0.024  0.001  0.000
## x6  0.015 -0.036  0.037 -0.001  0.000  0.000
## x7 -0.129 -0.112 -0.165  0.017  0.070 -0.004  0.000
## x8  0.026 -0.047 -0.050 -0.128 -0.024 -0.100  0.062  0.000
## x9  0.239  0.058  0.118  0.111  0.160  0.077 -0.044 -0.029  0.000
```

Standardized residuals (z-scores)

```r
lavResiduals(config.fit, type = "cor")$Pasteur$cov.z
```

```
##      x1     x2     x3     x4     x5     x6     x7     x8     x9
## x1  0.000
## x2 -2.140  0.000
## x3 -0.840  2.719  0.000
## x4  1.891 -1.132 -1.579  0.000
## x5 -2.308 -0.782 -3.434  2.172  0.000
## x6  2.042  0.400 -0.311 -3.620  1.105  0.000
## x7 -2.130 -2.697 -1.483  1.888 -0.540  0.975  0.000
## x8 -1.013 -0.501  0.447 -1.078 -1.609  0.602  1.898  0.000
## x9  2.613  1.689  2.888 -0.189  0.135  0.706 -1.306 -0.731  0.000
```

```r
lavResiduals(config.fit, type = "cor")$`Grant-White`$cov.z
```

```
##      x1     x2     x3     x4     x5     x6     x7     x8     x9
## x1  0.000
## x2 -0.716  0.000
## x3 -1.266  1.692  0.000
## x4  0.577 -0.254  0.082  0.000
## x5  0.124 -1.268 -0.551  0.138  0.000
## x6  0.313 -0.635  0.827 -0.135 -0.003  0.000
## x7 -2.328 -1.679 -3.050  0.310  1.230 -0.076  0.000
## x8  0.541 -0.807 -1.142 -3.052 -0.518 -2.174  3.995  0.000
## x9  4.162  0.966  2.127  2.012  2.816  1.343 -2.188 -3.173  0.000
```

```
head(modificationindices(config.fit, sort. = TRUE))
```

```
##         lhs op rhs block group level     mi    epc sepc.lv sepc.all sepc.nox
## 178      x7 ~~  x8     2     2     1 24.819  0.612   0.612    1.247    1.247
## 132  visual =~  x9     2     2     1 24.539  0.748   0.581    0.567    0.567
## 113      x4 ~~  x6     1     1     1 11.280 -0.326  -0.326   -0.928   -0.928
## 130  visual =~  x7     2     2     1 11.267 -0.504  -0.391   -0.380   -0.380
## 78   visual =~  x9     1     1     1 11.073  0.304   0.318    0.322    0.322
## 79  textual =~  x1     1     1     1 10.185  0.944   0.893    0.756    0.756
```

Configural model

- Configural model has poor fit.
- Correlation residuals indicate many correlations are over/underestimated.
- **If this was my model, I would fix this model before continuing**

Weak (Metric) Invariance

```r
weak.fit <- cfa(config.mod, HolzingerSwineford1939,
                group = "school",
                group.equal = "loadings")
fit.stat <- rbind(fitmeasures(config.fit, c("chisq", "df", "pvalue", "cfi", "rmsea", "srmr")),
                  fitmeasures(weak.fit, c("chisq", "df", "pvalue", "cfi", "rmsea", "srmr")))
rownames(fit.stat) <- c("Configural", "Weak")
fit.stat
```

```
##               chisq df       pvalue       cfi      rmsea       srmr
## Configural 115.8513 48 1.545283e-07 0.9233984 0.09691486 0.06786401
## Weak       124.0435 54 1.962798e-07 0.9209235 0.09283654 0.07165158
```

```r
# chi-square test of difference
pchisq(124.0435 - 115.8513, df = 54 - 48, lower.tail = FALSE)
```

```
## [1] 0.2243578
```

```r
# compare to:
anova(config.fit, weak.fit)
```

```
## Chi Square Difference Test
##
##            Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## config.fit 48 7484.4 7706.8 115.85
## weak.fit   54 7480.6 7680.8 124.04     8.1922       6     0.2244
```

Poor fit continues ...

```
lavResiduals(weak.fit, type = "cor")$Pasteur$cov

##     x1     x2     x3     x4     x5     x6     x7     x8     x9
## x1  0.000
## x2 -0.055  0.000
## x3 -0.010  0.076  0.000
## x4  0.111 -0.109 -0.116  0.000
## x5 -0.002 -0.083 -0.196  0.041  0.000
## x6  0.106 -0.020 -0.056 -0.042  0.015  0.000
## x7 -0.099 -0.229 -0.125  0.118 -0.020  0.058  0.000
## x8 -0.035 -0.068 -0.005 -0.055 -0.068  0.019  0.055  0.000
## x9  0.149  0.096  0.172 -0.022  0.003  0.025 -0.028 -0.026  0.000

lavResiduals(weak.fit, type = "cor")$`Grant-White`$cov

##     x1     x2     x3     x4     x5     x6     x7     x8     x9
## x1  0.000
## x2 -0.013  0.000
## x3 -0.024  0.114  0.000
## x4  0.005  0.012  0.027  0.000
## x5 -0.027 -0.056 -0.014 -0.016  0.000
## x6  0.005 -0.006  0.068  0.033  0.004  0.000
## x7 -0.155 -0.100 -0.157  0.013  0.057 -0.004  0.000
## x8 -0.001 -0.030 -0.038 -0.130 -0.036 -0.096  0.060  0.000
## x9  0.214  0.073  0.129  0.108  0.149  0.080 -0.047 -0.027  0.000
```

Weak invariance summary

- Model fit is not meaningful (or statistically) worse than configural model.
- **Model fit is still not good.**
- Conclude(?) weak invariance.

Strong (Scalar) invariance

```
strong.fit <- cfa(config.mod, HolzingerSwineford1939,
                  group = "school",
                  group.equal = c("loadings", "intercepts"))
fit.stat <- rbind(fit.stat,
                  fitmeasures(strong.fit, c("chisq", "df", "pvalue", "cfi", "rmsea", "srmr")))
rownames(fit.stat)[3] <- "Strong"
fit.stat
```

```
##               chisq df       pvalue       cfi      rmsea       srmr
## Configural 115.8513 48 1.545283e-07 0.9233984 0.09691486 0.06786401
## Weak       124.0435 54 1.962798e-07 0.9209235 0.09283654 0.07165158
## Strong     164.1028 60 1.296141e-11 0.8824718 0.10737110 0.08244706
```

```
# chi-square test of difference
anova(weak.fit, strong.fit)
```

```
## Chi Square Difference Test
##
##            Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## weak.fit   54 7480.6 7680.8 124.04
## strong.fit 60 7508.6 7686.6 164.10     40.059       6  4.435e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Strong invariance summary

- Model fit is worse than weak invariance model.
- Conclude no evidence of strong invariance (at least complete)
- Let's examine partial invariance.

Examining partial invariance

To examine which paths should be freed across groups, we will use the `lavTestScore` function.

We are looking for statistically significant paths.

```
lavTestScore(strong.fit)$uni
```

```
##
## univariate score tests:
##
##      lhs op   rhs     X2 df p.value
## 1   .p2. == .p38.  0.306  1   0.580
## 2   .p3. == .p39.  1.636  1   0.201
## 3   .p5. == .p41.  2.744  1   0.098
## 4   .p6. == .p42.  2.627  1   0.105
## 5   .p8. == .p44.  0.027  1   0.871
## 6   .p9. == .p45.  0.004  1   0.952
## 7  .p25. == .p61.  5.847  1   0.016
## 8  .p26. == .p62.  6.863  1   0.009
## 9  .p27. == .p63. 19.193  1   0.000
## 10 .p28. == .p64.  2.139  1   0.144
## 11 .p29. == .p65.  1.563  1   0.211
## 12 .p30. == .p66.  0.032  1   0.857
## 13 .p31. == .p67. 15.021  1   0.000
## 14 .p32. == .p68.  4.710  1   0.030
## 15 .p33. == .p69.  1.498  1   0.221
```

Identifying .p27.

```
params <- parameterEstimates(strong.fit)
subset(params, label %in% c(".p27."))
```

```
##    lhs op rhs block group label   est    se      z pvalue ci.lower ci.upper
## 27  x3 ~1         1     1 .p27. 2.271 0.083 27.387      0    2.109    2.434
## 63  x3 ~1         2     2 .p27. 2.271 0.083 27.387      0    2.109    2.434
```

Strong (partial) invariance

```r
strong.fit.p1 <- cfa(config.mod, HolzingerSwineford1939,
                     group = "school",
                     group.equal = c("loadings", "intercepts"),
                     group.partial = c("x3 ~ 1"))

# chi-square test of difference
anova(weak.fit, strong.fit.p1)
```

```
## Chi Square Difference Test
##
##               Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## weak.fit      54 7480.6 7680.8 124.04
## strong.fit.p1 59 7491.1 7672.8 144.58     20.535       5  0.0009912 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Looking again

```r
lavTestScore(strong.fit.p1)$uni
```

```
##
## univariate score tests:
##
##      lhs op   rhs     X2 df p.value
## 1   .p2. == .p38.  0.734  1   0.392
## 2   .p3. == .p39.  0.485  1   0.486
## 3   .p5. == .p41.  2.760  1   0.097
## 4   .p6. == .p42.  2.630  1   0.105
## 5   .p8. == .p44.  0.026  1   0.872
## 6   .p9. == .p45.  0.002  1   0.960
## 7  .p25. == .p61.  2.833  1   0.092
## 8  .p26. == .p62.  2.833  1   0.092
## 9  .p28. == .p64.  2.136  1   0.144
## 10 .p29. == .p65.  1.560  1   0.212
## 11 .p30. == .p66.  0.032  1   0.857
## 12 .p31. == .p67. 15.023  1   0.000
## 13 .p32. == .p68.  4.727  1   0.030
## 14 .p33. == .p69.  1.492  1   0.222
```

Identifying .p31.

```
params <- parameterEstimates(strong.fit.p1)
subset(params, label %in% c(".p31."))
```

```
##    lhs op rhs block group label   est    se      z pvalue ci.lower ci.upper
## 31  x7 ~1         1     1 .p31. 4.242 0.073 57.966      0    4.099    4.386
## 67  x7 ~1         2     2 .p31. 4.242 0.073 57.966      0    4.099    4.386
```

Strong (partial) invariance again

```
strong.fit.p2 <- cfa(config.mod, HolzingerSwineford1939,
                     group = "school",
                     group.equal = c("loadings", "intercepts"),
                     group.partial = c("x3 ~ 1",
                                       "x7 ~ 1"))

# chi-square test of difference
anova(weak.fit, strong.fit.p2)
```

```
## Chi Square Difference Test
##
##               Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## weak.fit      54 7480.6 7680.8 124.04
## strong.fit.p2 58 7478.0 7663.3 129.42     5.3789       4     0.2506
```

```
# fit statistics
fit.stat[3,] <- fitmeasures(strong.fit.p2, c("chisq", "df", "pvalue", "cfi", "rmsea", "srmr"))
rownames(fit.stat)[3] <- "Partial strong"
fit.stat
```

```
##                   chisq df       pvalue       cfi      rmsea       srmr
## Configural     115.8513 48 1.545283e-07 0.9233984 0.09691486 0.06786401
## Weak           124.0435 54 1.962798e-07 0.9209235 0.09283654 0.07165158
## Partial strong 129.4225 58 2.277881e-07 0.9193667 0.09045555 0.07298884
```

Strong invariance summary

- Evidence for partial strong invariance.
- Non-invariant intercepts associated with the verbal and speed factors.
- Again, fit is still poor.

Strict invariance

```
strict.fit <- cfa(config.mod, HolzingerSwineford1939,
                  group = "school",
                  group.equal = c("loadings", "intercepts", "residuals"),
                  group.partial = c("x3 ~ 1",
                                    "x7 ~ 1"))

# chi-square test of difference
anova(strong.fit.p2, strict.fit)
```

```
## Chi Square Difference Test
##
##               Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## strong.fit.p2 58 7478.0 7663.3 129.42
## strict.fit    67 7477.8 7629.8 147.26     17.838       9     0.0371 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fit.stat <- rbind(fit.stat,
                  fitmeasures(strict.fit, c("chisq", "df", "pvalue", "cfi", "rmsea", "srmr")))
rownames(fit.stat)[4] <- "Strict"
fit.stat
```

```
##                   chisq df       pvalue       cfi      rmsea       srmr
## Configural     115.8513 48 1.545283e-07 0.9233984 0.09691486 0.06786401
## Weak           124.0435 54 1.962798e-07 0.9209235 0.09283654 0.07165158
## Partial strong 129.4225 58 2.277881e-07 0.9193667 0.09045555 0.07298884
## Strict         147.2605 67 5.882821e-08 0.9093890 0.08921649 0.07899220
```

What to conclude

1. Simulation studies (Cheung & Rensvol, 2002) have suggested ΔCFI $\leq .01$ indicate stricter invariance should not be rejected.

```
# difference in CFI between strong and strict invariance
fit.stat[3, 4] - fit.stat[4, 4]

## [1] 0.009977741
```

2. n < 300, ΔCFI $\leq .005$ and ΔRMSEA $\leq .010$ support invariance (Chen, 2007).

```
# difference in RMSEA between strong and strict invariance
fit.stat[4, 5] - fit.stat[3, 5]

## [1] -0.001239055
```

3. Evidence suggest that strict invariance is met, **but again fit is not good**!

Strict fit output (select), reference group

```
summary(strict.fit)

## Group 1 [Pasteur]:
##
## Intercepts:
##                   Estimate  Std.Err  z-value  P(>|z|)
##    .x3               2.487    0.090   27.772    0.000
##    .x7               4.432    0.082   53.865    0.000
##     visual           0.000
##     textual          0.000
##     speed            0.000
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)
##     visual           0.776    0.164    4.737    0.000
##     textual          0.893    0.131    6.826    0.000
##     speed            0.318    0.080    3.990    0.000
##
##
## Group 2 [Grant-White]:
##
## Intercepts:
##                   Estimate  Std.Err  z-value  P(>|z|)
##    .x3               1.951    0.114   17.044    0.000
##    .x7               3.992    0.099   40.135    0.000
##     visual           0.054    0.128    0.423    0.672
##     textual          0.575    0.118    4.888    0.000
##     speed           -0.071    0.089   -0.805    0.421
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)
##     visual           0.664    0.150    4.436    0.000
##     textual          0.876    0.132    6.620    0.000
##     speed            0.446    0.109    4.095    0.000
```

Latent response variables - dichotomous item

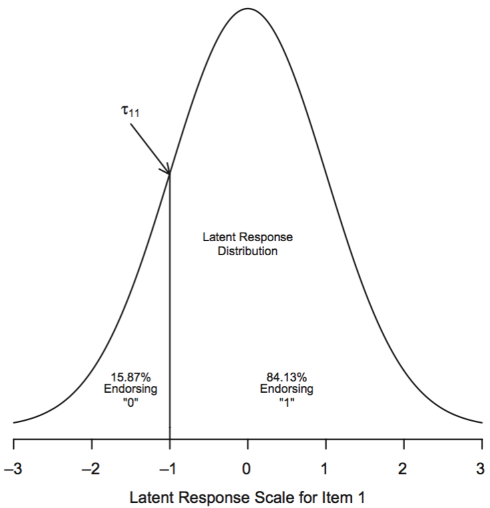


Figure 1. Latent response distribution for a single dichotomous item representing the latent distribution of interest. τ_{11} marks the latent cut-point between observed responses.

Latent response variables - dichotomous item

Let X^* be the latent response variable (LRV).

If we let $X^* \sim N(0,1)$ then the threshold (τ_1) correspond to z-scores and

$$X = \begin{cases} 0 & \text{if } X^* \leq \tau_{11} \\ 1 & \text{if } X^* > \tau_{11} \end{cases}$$

So, if a respondents score on the LRV is $\leq \tau_1$ they will not endorse the item.

LRVs have **nonlinear relationships with the indicators** BUT have **linear relationships with the factors**.

Fit an ordinal variable in lavaan

```
lsat6 <- data.frame(psych::lsat6)
library(lavaan)
lsat.mod <- '
  lsat =~ Q1 + Q2 + Q3 + Q4 + Q5
'
lsat.fit <- cfa(lsat.mod, lsat6, ordered = paste0("Q", 1:5))
```

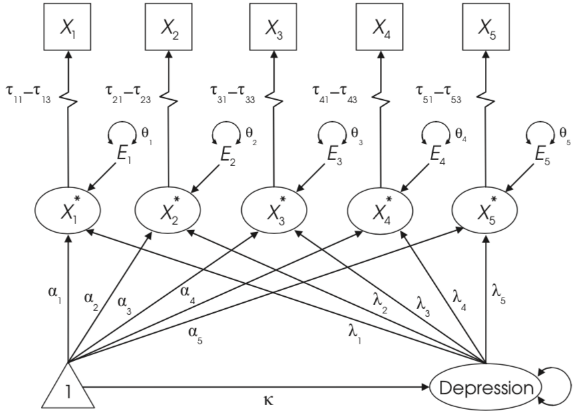
Parameterizations

- **Delta scaling**
  - **Total variance** of LRV fixed to 1.
  - For the standardized solution, pattern coefficients represent for a 1 SD increase in the factor, expect an XX SD change for the latent response variable.
  - For the standardized solution, threshold correspond to normal deviates corresponding to cumulative probabilities
- **Theta scaling**
  - **Residual variance** of each LRV fixed to 1 (like probit regression scaling).
  - For the unstandardized solution, pattern coefficients represent for a 1 unit increase in the factor, expect an XX probit (normal deviates) change for the latent response variable,
  - For the unstandardized solution, threshold correspond to normal deviates for the next lowest response category where the latent response variable is not standardized.
- Standardized solution identical between the two parameterizations

Invariance with ordinal indicators

- For a given indicator, the probability of endorsing/selecting an option (e.g., SA, A, N, D, SD) across the groups is the same given the same underlying score on the factor.
- Observed ordinal responses are only indirectly, through continuous LRVs, related to the common factor.
- The observed responses are related to the continuous LRVs through the set of thresholds.

Single-factor (Depression) CFA ordinal indicators (Kline)



Identification - Configural model

Millsap & Yun-Tein (2004) "Assessing Factorial Invariance in Ordered-Categorical Measures" is the definitive guide to dealing with invariance with ordinal data.

We won't cover binary data, See Millsap & Yun-Tein (2004) for the rules.

Need at **least 3 categories** and each LRV is a simple indicator with a single pattern coefficient (i.e., **simple structure**).

Use theta parameterization (fixing residual variance of each LRV 1 in the reference group).

1. **In just the reference group**, fix the mean of the factors to zero and standardize (1) the residual variance of every LRV.
2. **In every group**, fix the direct effect of the constant on every LRV to 0 and set the same LRV as a maker variable and fix it's unstandardized pattern coefficient to 1.
3. **For every LRV**, constrain one threshold parameter to equality across the groups and **for the marker variable** constrain a second threshold parameter to equality.

Every thing else is free

Invariance steps

- Fit the configural model (previous slide).
- Constrain the unstandardized pattern coefficients for each latent response variable (weak invariance). Compare to configural.
- Constrain the remaining free thresholds (strong invariance). Compare to weak invariance.
  - Equality of pattern coefficients and thresholds is required to claim ordinal indicators measure the same common factors but with differing degrees of precision.
- Constrain the error variances/covariances (weak invariance). Compare to strong invariance.
  - If strict invariance holds the indicators measure the same common factors in identical ways across the groups.
- As with continuous indicators, factor means/variances do not need to be the same across the groups to have invariance.

Example

- Example from Klein
- N = 2,252 (2,004 white men, 248 African American men)
- Responded to 5 Likert-type items corresponding to symptoms of depression from the CES-D scale.
- The items each have 4 response categories.

Reading in data

```
radloff <- read.csv("https://www.guilford.com/add/kline/radloff-lavaan.txt", sep = "\t", header = FALSE,
                    col.names = c("x1", "x2", "x3", "x4", "x5", "g"))
radloff[, 1:5] <- lapply(radloff[, 1:5], function(x) as.ordered(x))
radloff$g <- factor(radloff$g, levels = c(1, 2), labels = c("Wh", "AA"))
str(radloff)
```

```
## 'data.frame': 2252 obs. of  6 variables:
##  $ x1: Ord.factor w/ 4 levels "0"<"1"<"2"<"3": 1 1 2 1 2 1 1 1 1 1 ...
##  $ x2: Ord.factor w/ 4 levels "0"<"1"<"2"<"3": 1 1 2 1 1 1 1 1 4 1 ...
##  $ x3: Ord.factor w/ 4 levels "0"<"1"<"2"<"3": 1 1 1 1 2 2 1 1 3 1 ...
##  $ x4: Ord.factor w/ 4 levels "0"<"1"<"2"<"3": 1 1 1 1 1 1 1 3 4 2 ...
##  $ x5: Ord.factor w/ 4 levels "0"<"1"<"2"<"3": 1 1 2 1 2 1 1 2 2 1 ...
##  $ g : Factor w/ 2 levels "Wh","AA": 1 1 1 1 1 1 1 1 1 1 ...
```

```
summary(radloff)
```

```
##  x1       x2       x3       x4       x5        g
##  0:1749   0:1899   0:1561   0:1388   0:1600   Wh:2004
##  1: 330   1: 213   1: 383   1: 510   1: 414   AA: 248
##  2: 104   2:  66   2: 134   2: 199   2: 139
##  3:  69   3:  74   3: 174   3: 155   3:  99
```

Obtaining the polychoric correlations for each group

```
mod <- "
dep =~ x1 + x2 + x3 + x4 + x5
"
fit <- cfa(mod, data = radloff, group = "g", parameterization = "theta")
inspect(fit, what = "sampstat")$Wh$cov
```

```
##    x1    x2    x3    x4    x5
## x1 1.000
## x2 0.437 1.000
## x3 0.471 0.480 1.000
## x4 0.401 0.418 0.454 1.000
## x5 0.423 0.489 0.627 0.465 1.000
```

```
inspect(fit, what = "sampstat")$AA$cov
```

```
##    x1    x2    x3    x4    x5
## x1 1.000
## x2 0.508 1.000
## x3 0.351 0.373 1.000
## x4 0.305 0.336 0.398 1.000
## x5 0.464 0.371 0.531 0.483 1.000
```

Configural Model

```
mod <- "
# x1 is marker variable
# define latent variable
dep =~ c(1, 1)*x1 + x2 + x3 + x4 + x5

# fix thresholds
x1 | c(t11, t11)*t1 + c(t12, t12)*t2 + t3
x2 | c(t21, t21)*t1 + t2 + t3
x3 | c(t31, t31)*t1 + t2 + t3
x4 | c(t41, t41)*t1 + t2 + t3
x5 | c(t51, t51)*t1 + t2 + t3

# fix factor mean to zero in reference group
# freely estimate it (NA) in the second group
dep ~ c(0, NA)*1

# freely estimate variance
dep ~~ NA*dep

# fix residual variance to 1 for reference group
x1 ~~ c(1, NA)*x1
x2 ~~ c(1, NA)*x2
x3 ~~ c(1, NA)*x3
x4 ~~ c(1, NA)*x4
x5 ~~ c(1, NA)*x5
"

fit <- cfa(mod, data = radloff, group = "g", parameterization = "theta", estimator = "wlsmv")
fitmeasures(fit, c("chisq.scaled", "df.scaled", "pvalue.scaled", "cfi.scaled", "rmsea.scaled", "srmr.scaled"))


##  chisq.scaled     df.scaled pvalue.scaled    cfi.scaled  rmsea.scaled
##        25.162        10.000         0.005         0.994         0.037
```

Correlation residuals

```
lavResiduals(fit, type = "cor")$Wh$cov
```

```
##     x1     x2     x3     x4     x5
## x1  0.000
## x2  0.041  0.000
## x3 -0.005 -0.029  0.000
## x4  0.030  0.020 -0.024  0.000
## x5 -0.046 -0.013  0.024 -0.005  0.000
```

```
lavResiduals(fit, type = "cor")$AA$cov
```

```
##     x1     x2     x3     x4     x5
## x1  0.000
## x2  0.128  0.000
## x3 -0.060 -0.021  0.000
## x4 -0.066 -0.018  0.014  0.000
## x5 -0.014 -0.087  0.034  0.037  0.000
```

Kline adds the error covariance between x1 and x2 for just AA group.
I would say it's debatable if it should be added and should be theoretical driven.
Overall, I would conclude this is a reasonably good model as is.

Weak Model

```
weak.mod <- "
dep =~ c(1, 1)*x1 + c(lam2, lam2)*x2 + c(lam3, lam3)*x3 + c(lam4, lam4)*x4 + c(lam5, lam5)*x5
dep ~ c(0, NA)*1
dep ~~ NA*dep

x1 | c(t11, t11)*t1 + c(t12, t12)*t2 + t3
x2 | c(t21, t21)*t1 + t2 + t3
x3 | c(t31, t31)*t1 + t2 + t3
x4 | c(t41, t41)*t1 + t2 + t3
x5 | c(t51, t51)*t1 + t2 + t3
x1 ~~ c(1, NA)*x1
x2 ~~ c(1, NA)*x2
x3 ~~ c(1, NA)*x3
x4 ~~ c(1, NA)*x4
x5 ~~ c(1, NA)*x5
"
weak.fit <- cfa(weak.mod, data = radloff, group = "g", parameterization = "theta", estimator = "WLSMV")
fit.stats <- rbind(
  fitmeasures(fit, c("chisq.scaled", "df.scaled", "pvalue.scaled", "cfi.scaled", "rmsea.scaled", "srmr.scaled")),
  fitmeasures(weak.fit, c("chisq.scaled", "df.scaled", "pvalue.scaled", "cfi.scaled", "rmsea.scaled", "srmr.scaled")))
row.names(fit.stats) <- c("configural", "weak")
fit.stats
```

```
##            chisq.scaled df.scaled pvalue.scaled cfi.scaled rmsea.scaled
## configural     25.16183        10   0.005047058  0.9943511   0.03671128
## weak           34.09612        14   0.001996754  0.9925128   0.03572036
```

```
anova(fit, weak.fit)
```

```
## Scaled Chi Square Difference Test (method = "satorra.2000")
##
##          Df AIC BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## fit      10         14.222
## weak.fit 14         21.271     5.8314       4     0.2121
```

Strong Model

```
strong.mod <- "
dep =~ c(1, 1)*x1 + c(lam2, lam2)*x2 + c(lam3, lam3)*x3 + c(lam4, lam4)*x4 + c(lam5, lam5)*x5
dep ~ c(0, NA)*1
dep ~~ NA*dep

x1 | c(t11, t11)*t1 + c(t12, t12)*t2 + c(t13, t13)*t3
x2 | c(t21, t21)*t1 + c(t22, t22)*t2 + c(t23, t23)*t3
x3 | c(t31, t31)*t1 + c(t32, t32)*t2 + c(t33, t33)*t3
x4 | c(t41, t41)*t1 + c(t42, t42)*t2 + c(t43, t43)*t3
x5 | c(t51, t51)*t1 + c(t52, t52)*t2 + c(t53, t53)*t3
x1 ~~ c(1, NA)*x1
x2 ~~ c(1, NA)*x2
x3 ~~ c(1, NA)*x3
x4 ~~ c(1, NA)*x4
x5 ~~ c(1, NA)*x5
"
strong.fit <- cfa(strong.mod, data = radloff, group = "g", parameterization = "theta", estimator = "WLSMV")
fit.stats <- rbind(fit.stats,
  fitmeasures(strong.fit, c("chisq.scaled", "df.scaled", "pvalue.scaled", "cfi.scaled", "rmsea.scaled", "srmr.scaled")))
row.names(fit.stats)[3] <- c("strong")
fit.stats
```

```
##            chisq.scaled df.scaled pvalue.scaled cfi.scaled rmsea.scaled
## configural     25.16183        10   0.005047058  0.9943511   0.03671128
## weak           34.09612        14   0.001996754  0.9925128   0.03572036
## strong         39.81064        23   0.016137246  0.9937369   0.02548895
```

```
anova(strong.fit, weak.fit)
```

```
## Scaled Chi Square Difference Test (method = "satorra.2000")
##
##            Df AIC BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## weak.fit   14         21.271
## strong.fit 23         26.268     8.6613       9     0.4691
```

Strict Model

```
strict.mod <- "
dep =~ c(1, 1)*x1 + c(lam2, lam2)*x2 + c(lam3, lam3)*x3 + c(lam4, lam4)*x4 + c(lam5, lam5)*x5
dep ~ c(0, NA)*1
dep ~~ NA*dep

x1 | c(t11, t11)*t1 + c(t12, t12)*t2 + c(t13, t13)*t3
x2 | c(t21, t21)*t1 + c(t22, t22)*t2 + c(t23, t23)*t3
x3 | c(t31, t31)*t1 + c(t32, t32)*t2 + c(t33, t33)*t3
x4 | c(t41, t41)*t1 + c(t42, t42)*t2 + c(t43, t43)*t3
x5 | c(t51, t51)*t1 + c(t52, t52)*t2 + c(t53, t53)*t3
x1 ~~ c(1, 1)*x1
x2 ~~ c(1, 1)*x2
x3 ~~ c(1, 1)*x3
x4 ~~ c(1, 1)*x4
x5 ~~ c(1, 1)*x5
"
strict.fit <- cfa(strict.mod, data = radloff, group = "g", parameterization = "theta", estimator = "WLSMV")
fit.stats <- rbind(fit.stats,
  fitmeasures(strict.fit, c("chisq.scaled", "df.scaled", "pvalue.scaled", "cfi.scaled", "rmsea.scaled", "srmr.scaled")))
row.names(fit.stats)[4] <- c("strict")
fit.stats
```

```
##            chisq.scaled df.scaled pvalue.scaled cfi.scaled rmsea.scaled
## configural     25.16183        10  5.047058e-03  0.9943511   0.03671128
## weak           34.09612        14  1.996754e-03  0.9925128   0.03572036
## strong         39.81064        23  1.613725e-02  0.9937369   0.02548895
## strict         79.40325        28  8.188092e-07  0.9808487   0.04039615
```

```
anova(strong.fit, strict.fit)
```

```
## Scaled Chi Square Difference Test (method = "satorra.2000")
##
##            Df AIC BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## strong.fit 23         26.268
## strict.fit 28         61.580     14.029       5    0.01543 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

What to conclude and do next?

- Could, and probably should, examine partial strict invariance.
  - n > 300, ΔCFI $\leq .010$ and ΔRMSEA $\leq .015$ support invariance (Chen, 2007).

```
# delta CFI
abs(diff(fit.stats[3:4, 4]))

##     strict
## 0.01288819

# delta RMSEA
abs(diff(fit.stats[3:4, 5]))

##     strict
## 0.01490721
```

- Once settle on a model, can compare mean differences calculate effects, etc.
- For brevity, we'll look at the strong.fit model.

Strong fit model

```
summary(strong.fit, fit = TRUE, standardized = TRUE, rsquare = TRUE)


##   Optimization method                           NLMINB
##   Number of free parameters                         46
##   Number of equality constraints                    19
##
##   Number of observations per group
##   Wh                                              2004
##   AA                                               248
##
##   Estimator                                       DWLS      Robust
##   Model Fit Test Statistic                      26.269      39.811
##   Degrees of freedom                                23          23
##   P-value (Chi-square)                           0.288       0.016
##   Scaling correction factor                                  0.711
##   Shift parameter for each group:
##     Wh                                                       2.548
##     AA                                                       0.315
##     for simple second-order correction (Mplus variant)
##
## Chi-square for each group:
##
##   Wh                                             9.976      16.580
##   AA                                            16.292      23.231
##
## Model test baseline model:
##
##   Minimum Function Test Statistic             3408.088    2704.055
##   Degrees of freedom                                20          20
##   P-value                                        0.000       0.000
##
## User model versus baseline model:
##
##   Comparative Fit Index (CFI)                    0.999       0.994
##   Tucker-Lewis Index (TLI)                       0.999       0.995
##   RMSEA                                          0.011       0.025
##   SRMR                                           0.026       0.026
```

Strong fit model

```
summary(strong.fit, fit = TRUE, standardized = TRUE, rsquare = TRUE)
```

```
## Group 1 [Wh]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   dep =~
##     x1                1.000                               0.764    0.607
##     x2      (lam2)    1.123    0.108   10.384    0.000    0.859    0.651
##     x3      (lam3)    1.694    0.158   10.720    0.000    1.295    0.791
##     x4      (lam4)    0.996    0.086   11.646    0.000    0.761    0.606
##     x5      (lam5)    1.557    0.142   10.993    0.000    1.190    0.766
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     dep               0.000                               0.000    0.000
##    .x1                0.000                               0.000    0.000
##    .x2                0.000                               0.000    0.000
##    .x3                0.000                               0.000    0.000
##    .x4                0.000                               0.000    0.000
##    .x5                0.000                               0.000    0.000
##
## Thresholds:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     x1|t1    (t11)    0.962    0.046   21.017    0.000    0.962    0.765
##     x1|t2    (t12)    1.799    0.064   28.043    0.000    1.799    1.429
##     x1|t3    (t13)    2.358    0.084   28.184    0.000    2.358    1.873
##     x2|t1    (t21)    1.355    0.061   22.131    0.000    1.355    1.028
##     x2|t2    (t22)    2.056    0.080   25.733    0.000    2.056    1.560
##     x2|t3    (t23)    2.470    0.094   26.225    0.000    2.470    1.874
##     x3|t1    (t31)    0.870    0.062   14.000    0.000    0.870    0.532
##     x3|t2    (t32)    1.878    0.093   20.248    0.000    1.878    1.148
##     x3|t3    (t33)    2.453    0.113   21.764    0.000    2.453    1.499
##     x4|t1    (t41)    0.378    0.036   10.408    0.000    0.378    0.301
##     x4|t2    (t42)    1.270    0.047   27.141    0.000    1.270    1.011
##     x4|t3    (t43)    1.872    0.060   31.221    0.000    1.872    1.490
##     x5|t1    (t51)    0.872    0.058   14.933    0.000    0.872    0.561
##     x5|t2    (t52)    1.952    0.089   21.886    0.000    1.952    1.256
##     x5|t3    (t53)    2.661    0.115   23.131    0.000    2.661    1.712
```

Strong fit model

```
summary(strong.fit, fit = TRUE, standardized = TRUE, rsquare = TRUE)

## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     dep               0.584    0.082    7.122    0.000    1.000    1.000
##    .x1                1.000                               1.000    0.631
##    .x2                1.000                               1.000    0.576
##    .x3                1.000                               1.000    0.374
##    .x4                1.000                               1.000    0.633
##    .x5                1.000                               1.000    0.414
##
## Scales y*:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     x1                0.794                               0.794    1.000
##     x2                0.759                               0.759    1.000
##     x3                0.611                               0.611    1.000
##     x4                0.796                               0.796    1.000
##     x5                0.643                               0.643    1.000
##
## R-Square:
##                    Estimate
##     x1                0.369
##     x2                0.424
##     x3                0.626
##     x4                0.367
##     x5                0.586
```

Strong fit model

```
summary(strong.fit, fit = TRUE, standardized = TRUE, rsquare = TRUE)
```

```
## Group 2 [AA]:
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     dep               0.059    0.091    0.651    0.515    0.075    0.075
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     dep               0.622    0.142    4.385    0.000    1.000    1.000
##    .x1                0.907    0.188    4.819    0.000    0.907    0.593
##    .x2                1.408    0.302    4.655    0.000    1.408    0.642
##    .x3                3.719    0.947    3.926    0.000    3.719    0.676
##    .x4                0.960    0.199    4.828    0.000    0.960    0.609
##    .x5                0.876    0.243    3.602    0.000    0.876    0.368
##
## Scales y*:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     x1                0.809                               0.809    1.000
##     x2                0.675                               0.675    1.000
##     x3                0.426                               0.426    1.000
##     x4                0.796                               0.796    1.000
##     x5                0.648                               0.648    1.000
##
## R-Square:
##                    Estimate
##     x1                0.407
##     x2                0.358
##     x3                0.324
##     x4                0.391
##     x5                0.632
```