# Statistical Analysis Using Structural Equation Models

## EPsy 8266

Christopher David Desjardins

Research Methodology Consulting Center

3/11/19

# Factor fallacies

- **Naming fallacy** - just cause you call it "intelligence" doesn't make it intelligence.
- **Reification** - belief that the factor must be a real thing.
- **Jingle-jangle fallacy** - Two things with the same name don't necessarily mean the same thing (jingle) and having two separate names doesn't make them distinct (jangle)

# Problems in CFA

Many problems arise within CFAs (and SEMs) such as Heywood cases (standardized loading $> 1$ & negative error variance) and nonconvergence.

Can also have nonpositive definite factor covariance and error covariance matrices.

**Especially likely when the number of observation is small.**

**Some causes/fixes**

- Model overparameterized/fix parameters
- Non-normal distributions & outliers/initial data analysis & transformations
- Empirical underidentification/bring in additional indicators
- Misspecified measurement model/look at residuals & modification indices

# Assessing empirical underidentification with `lavaan`

```r
# Checking empirical underidentification
library(lavaan)
HS.model <- ' visual  =~ x1 + x2 + x3
              textual =~ x4 + x5 + x6
              speed   =~ x7 + x8 + x9 '

# default lavaainstall.packages("BiocManager")n starting values
fit.raw <- cfa(HS.model, data = HolzingerSwineford1939)

# change the starting values
fit.altstart <- cfa(HS.model, data = HolzingerSwineford1939, start = "simple")

# verify these diff
inspect(fit.raw, "start")
inspect(fit.altstart, "start")

# extract model-implied covariance-matrix
covMat <- inspect(fit, "implied")$cov[,]
fit.cov <- cfa(HS.model, sample.cov = covMat, sample.nobs = nrow(HolzingerSwineford1939))

# obtain parameter estimates
raw.params <- parameterEstimates(fit.raw)[,"est"]
altstart.params <- parameterEstimates(fit.altstart)[,"est"]
cov.params <- parameterEstimates(fit.cov)[,"est"]

data.frame(params = do.call(paste, parameterEstimates(fit.cov)[,1:3]),
           raw = raw.params,
           alt = altstart.params,
           cov = cov.params)
```

# Types of indicators

Indicators may be of certain types ...

- **Congeneric** - Measure the same construct but not equally.
- **Tau-equivalent** - Congeneric and have equal true score variance (fix pattern coefficients to 1.0 for the two indicators).
- **Parallel** - Add equal error variance constraint (constrain error variances to be equal).

Can test with chi-square test of difference.

# Reliability

**Composite reliability (factor rho coefficient)** ratio of explained variance over total variance.

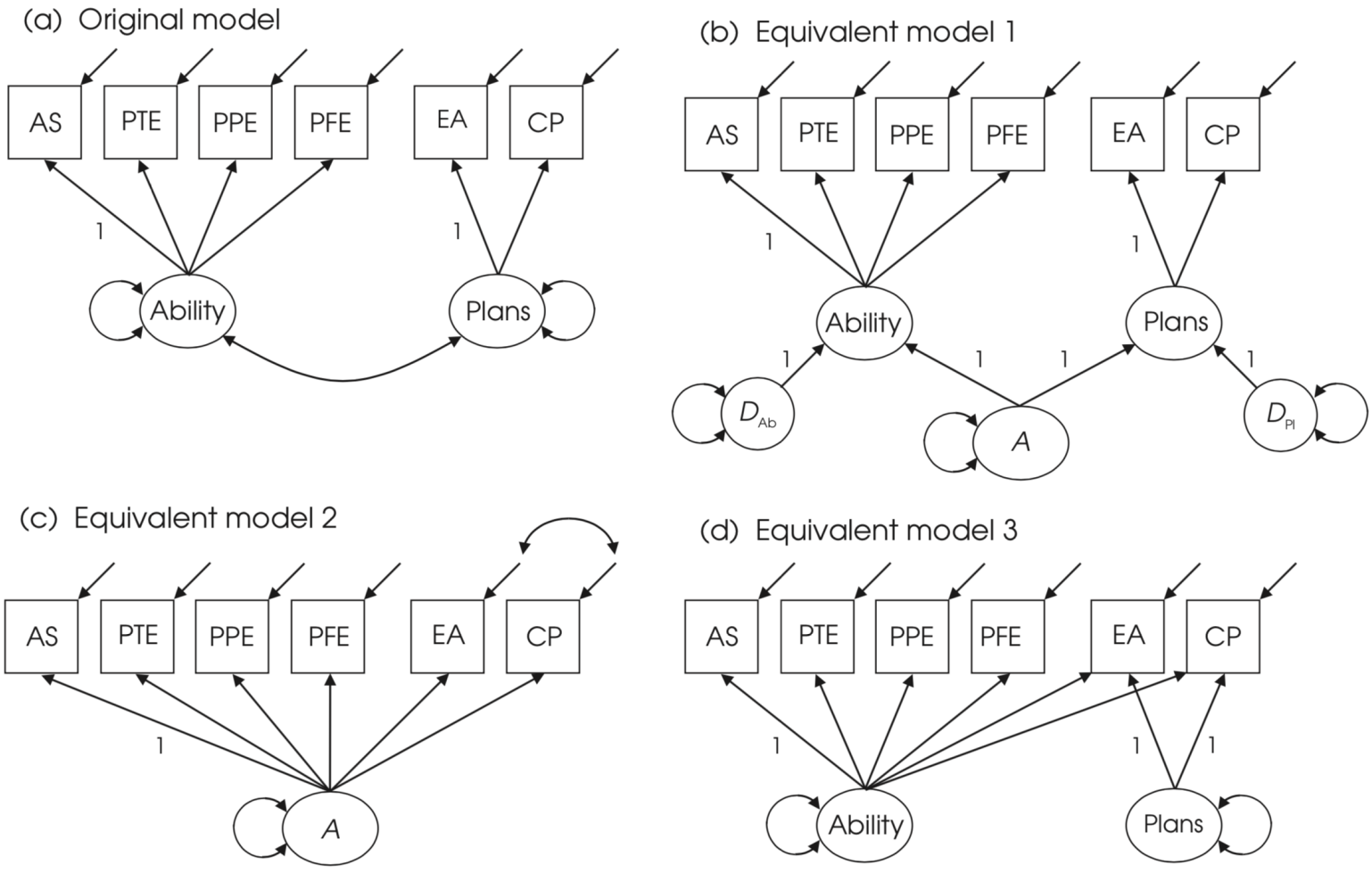
$$CR = \frac{(\sum \hat{\lambda}_i)^2 \hat{\phi}}{(\sum \hat{\lambda}_i)^2 \hat{\phi} + \sum \hat{\theta}_{ii} + 2 \sum \hat{\theta}_{ij}}$$

where

- $\hat{\lambda}_i$ is the unstandardized pattern coefficients among indicators for the same factor
- $\hat{\phi}$ is the factor variance
- $\hat{\theta}_{ii}$ is the unstandardized error variances
- $\hat{\theta}_{ij}$ are the nonzero unstandardized error covariances (often zero)

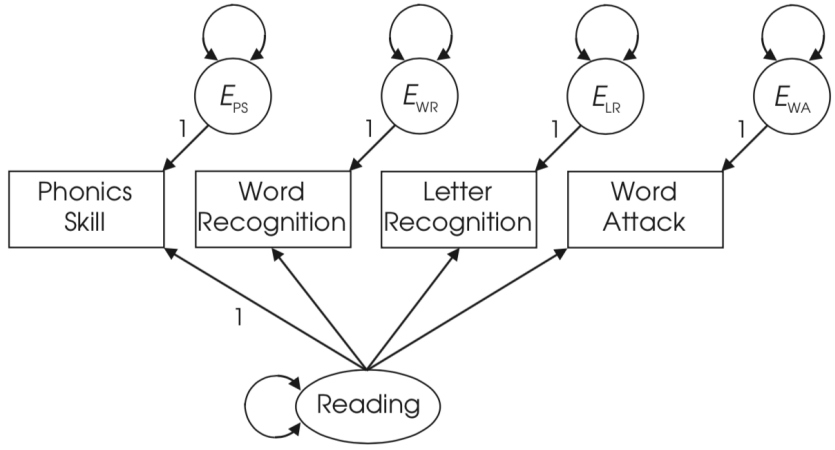
Alternatively, can take average of the squared standardized pattern coefficients (**average variance extracted**)

# Equivalent CFA models

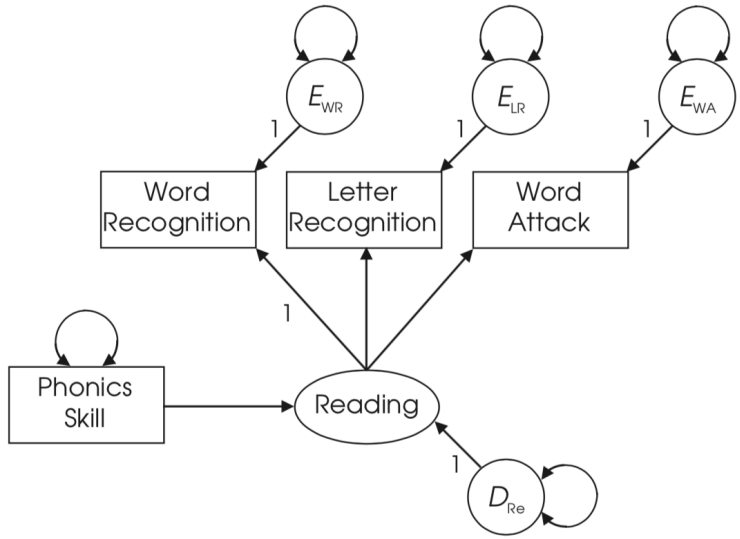


# Equivalent CFA models - 2

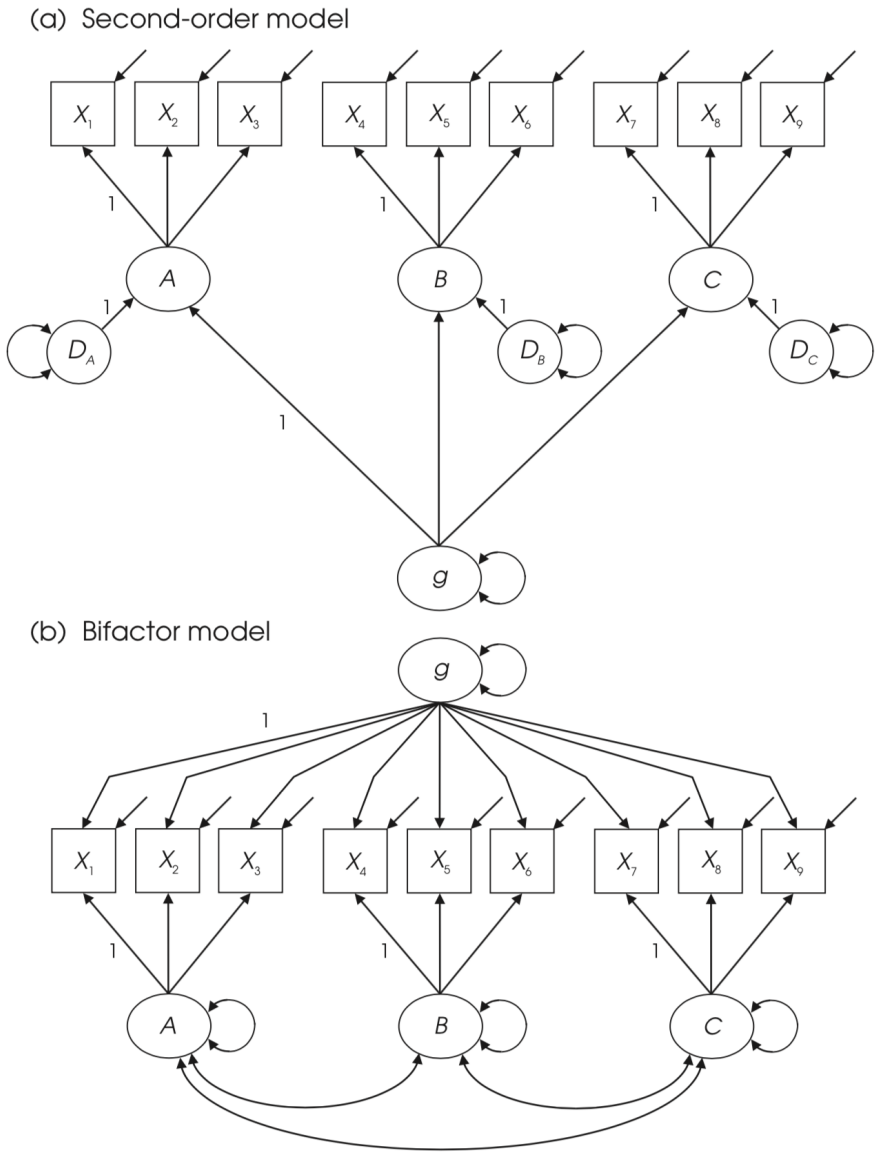
(a) Original model with effect indicators




(b) Equivalent model with a causal indicator



# Hierarchical & bifactor models



# Hierarchical & bifactor models

- Hierarchical
  - A second order factor causes the relationship between the first order factors.
  - Measured indirectly through the first order factors (i.e., no direct indicators).
- Bifactor
  - Indicators directly load onto the general factor and orthogonal to specific factors
  - General factor unrelated to specific factors.
  - Predictive validity of specific factors, partialling out a general factor, can be examined.

# Ordinal indicators

- So far, we've assumed our indicators are ratio/interval scale (i.e., continuous)
- This means we can use full information maximum likelihood (more about this soon!)
- With ordinal data, requires new parameters, new intermediate, latent variables, and a new estimator.

# Latent response variables

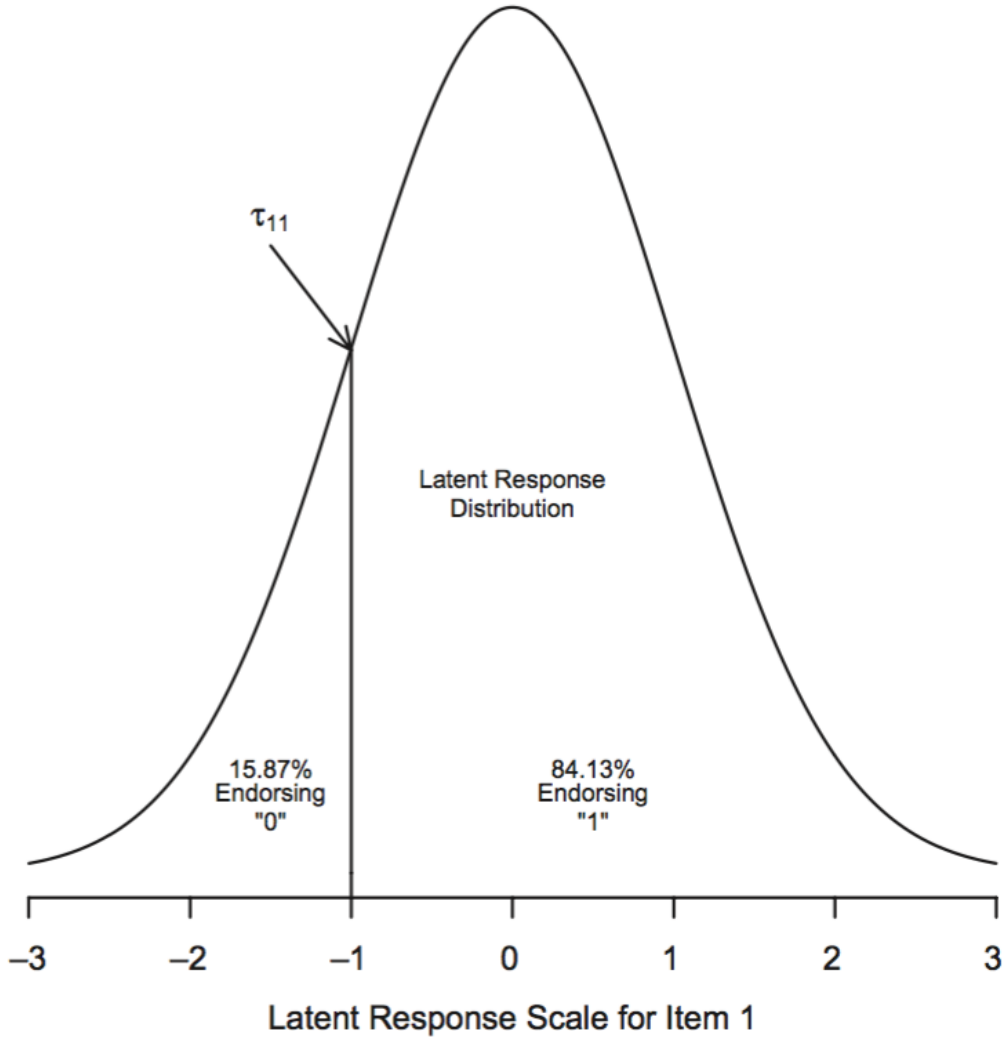



*Figure 1.* Latent response distribution for a single dichotomous item representing the latent distribution of interest. $\tau_{11}$ marks the latent cut-point between observed responses.

# Latent response variables

Let $X^*$ be the latent response variable.

If we let $X^* \sim N(0, 1)$ then the threshold $(\tau_1)$ 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 latent response variable is $leq\tau_1$ they will not endorse the item.

Latent response variables have **nonlinear relationships with the indicators** BUT have **linear relationships with the factors**.

# Fit an ordinal variable in lavaan

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

# How are thresholds calculated?

```
lsat.params <- parameterEstimates(lsat.fit)
calc_cumprob <- function(x){
  cumsum(prop.table(table(x)))
}
cum_probs <- apply(lsat6, 2, calc_cumprob); cum_probs
```

```
##      Q1    Q2    Q3    Q4   Q5
## 0 0.076 0.291 0.447 0.237 0.13
## 1 1.000 1.000 1.000 1.000 1.00
```

```
qnorm(cum_probs[1, ])
```

```
##         Q1         Q2         Q3         Q4
## -1.4325027 -0.5504657 -0.1332445 -0.7159860
##         Q5
## -1.1263911
```

```
subset(lsat.params, rhs == "t1", select = est, drop = TRUE)
```

```
## [1] -1.4325027 -0.5504657 -0.1332445 -0.7159860
## [5] -1.1263911
```

# Parameterizations

There are two ways to scale latent response variables.

- **Delta scaling**
  - Total variance of latent response variable 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 based corresponding to cumulative probabilities
- **Theta scaling**
  - Residual variance of each latent response variable 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 lowest response category.
- Standardized solution identical between the two

# Ordinal model in lavaan

```
summary(lsat.fit, standardized = TRUE)


## lavaan 0.6-3 ended normally after 29 iterations
##
##   Optimization method                           NLMINB
##   Number of free parameters                         10
##   Number of observations                          1000
##   Estimator                                       DWLS      Robust
##   Model Fit Test Statistic                       4.051       4.740
##   Degrees of freedom                                 5           5
##   P-value (Chi-square)                           0.542       0.448
##   Scaling correction factor                                  0.867
##   Shift parameter                                            0.070
##     for simple second-order correction (Mplus variant)
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model        Unstructured
##   Standard Errors                           Robust.sem
##
## Latent Variables:
##                    Estimate  Std.Err  z-value   P(>|z|)   Std.lv  Std.all
##   lsat =~
##     Q1                1.000                               0.389    0.389
##     Q2                1.020    0.358    2.846     0.004    0.397    0.397
##     Q3                1.210    0.447    2.709     0.007    0.471    0.471
##     Q4                0.968    0.352    2.751     0.006    0.377    0.377
##     Q5                0.879    0.352    2.499     0.012    0.342    0.342
##
## Intercepts:
##                    Estimate  Std.Err  z-value   P(>|z|)   Std.lv  Std.all
##    .Q1                0.000                               0.000    0.000
##    .Q2                0.000                               0.000    0.000
##    .Q3                0.000                               0.000    0.000
##    .Q4                0.000                               0.000    0.000
##    .Q5                0.000                               0.000    0.000
##     lsat              0.000                               0.000    0.000
```

# Ordinal model in lavaan

```
summary(lsat.fit, standardized = TRUE)


## Thresholds:
##                   Estimate  Std.Err  z-value   P(>|z|)   Std.lv  Std.all
##     Q1|t1           -1.433    0.059  -24.431     0.000   -1.433   -1.433
##     Q2|t1           -0.550    0.042  -13.133     0.000   -0.550   -0.550
##     Q3|t1           -0.133    0.040   -3.349     0.001   -0.133   -0.133
##     Q4|t1           -0.716    0.044  -16.430     0.000   -0.716   -0.716
##     Q5|t1           -1.126    0.050  -22.395     0.000   -1.126   -1.126
##
## Variances:
##                   Estimate  Std.Err  z-value   P(>|z|)   Std.lv  Std.all
##    .Q1               0.848                               0.848    0.848
##    .Q2               0.842                               0.842    0.842
##    .Q3               0.778                               0.778    0.778
##    .Q4               0.858                               0.858    0.858
##    .Q5               0.883                               0.883    0.883
##     lsat             0.152    0.087    1.743     0.081    1.000    1.000
##
## Scales y*:
##                   Estimate  Std.Err  z-value   P(>|z|)   Std.lv  Std.all
##     Q1               1.000                               1.000    1.000
##     Q2               1.000                               1.000    1.000
##     Q3               1.000                               1.000    1.000
##     Q4               1.000                               1.000    1.000
##     Q5               1.000                               1.000    1.000
```

# Estimating ordinal data

There are two types of robust estimator: mean-adjusted WLS (WLSM) and mean- and variance-adjusted WLS (WLSMV)

Makes different adjustments to the chi-square statistic to better approximate a chi-square distribution

WLSMV is the more favored approach (and is labeled Robust in lavaan)

Other estimators available and will talk more about this later.