SEM 1: Confirmatory Factor Analysis

Week 4 - Advanced CFA topics

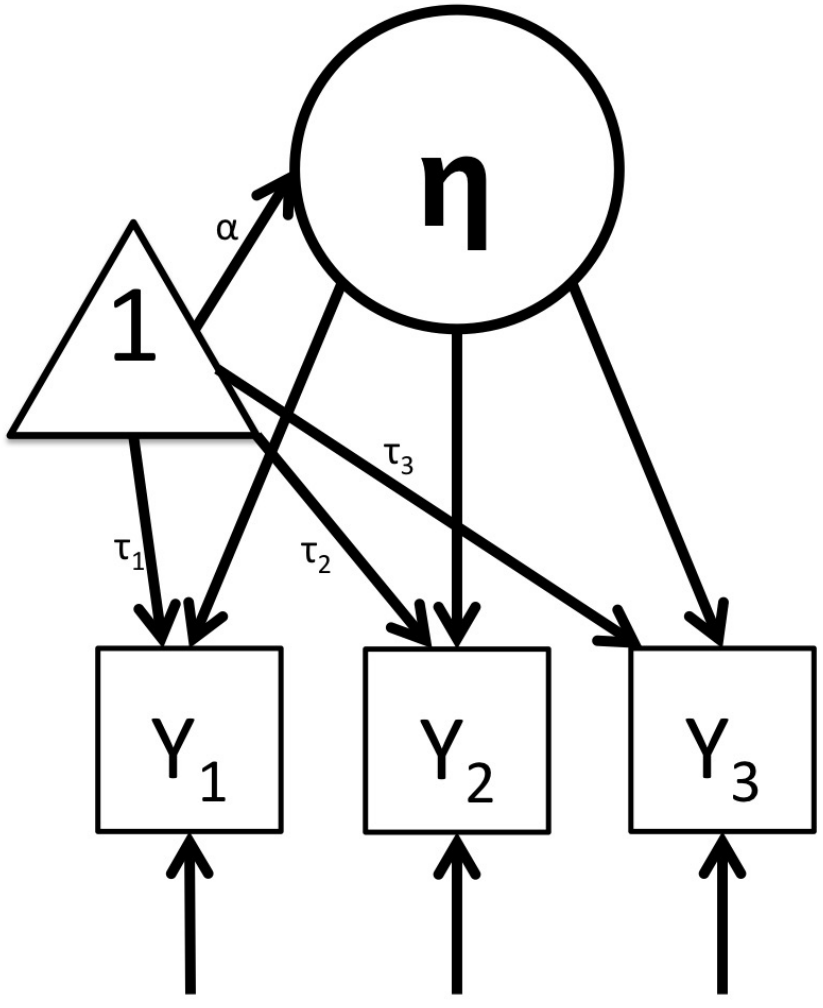
Sacha Epskamp

24-04-2018

Mean structure

$$\boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\Psi}\boldsymbol{\Lambda}^\top + \boldsymbol{\Theta}$$
$$\boldsymbol{\mu} = \boldsymbol{\tau} + \boldsymbol{\Lambda}\boldsymbol{\alpha}$$

- $\boldsymbol{\tau}$ can cancel $\boldsymbol{\alpha}$ out, hence we need to identify $\boldsymbol{\alpha} = \mathbf{0}$
- Number of parameters: $p(p+1)/2$ variances and covariances and p means!
- Number of parameters: p intercepts in $\boldsymbol{\tau}$
- p more observations, and p more parameters. This is why we normally ignore means!


η
α
1
τ3
τ1
τ2
Y1
Y2
Y3

Steps to assess measurement invariance:

- Configural invariance: Is the configuration of the model the same?
- Weak Invariance: Are factor loadings the same?
- Strong Invariance: Are the intercepts the same?
- Strict Invariance: Are the residual variances the same?

Sample Size

How big is 'big enough'?

- $n : q$ ratio should be high
  - Theory: to efficiently estimate lots of parameters, a larger sample is needed (5-10 per parameter)
  - There's very little evidence that it matters (Jackson, 2003)
  - This ratio is less important than absolute sample size
- $n \approx 200$ people
  - This is median SEM sample size (Shah & Goldstein, 2006)
  - Appropriate for an average model with ML estimation
  - Other recommendations: 100-200 people minimum
- Use larger n if:
  - Assumptions are violated (e.g., data are nonnormal)
  - Model is complex (e.g., latent interactions, multilevel structure)
  - Indicators have low reliability (factor loadings are low)

Power for Test of (Not-)Close Fit

- RMSEA estimates a population value
  - Its sampling distribution has been worked out
  - So we can put a confidence interval around it
  - This confidence interval allows us to ask whether RMSEA is significantly different from a specified value
- If the population model fit is NOT CLOSE, what is power to reject H_0 by the test of close fit?
- If the population model fit is CLOSE, what is power to reject H_0 by the test of not-close fit?
- Method described in MacCallum et al. (1996) is implemented in online calculators:
  - Power and minimum sample size for RMSEA: `http://quantpsy.org/rmsea/rmsea.htm`
  - Power curves for RMSEA: `http://quantpsy.org/rmsea/rmseaplot.htm`
  - See also `findRMSEAsamplesize` in semTools

Table 1
Relationship Between Confidence Intervals and Hypothesis Tests

Nature of confidence interval[a]	Reject close fit?	Reject not-close fit?
Entire confidence interval below 0.05	No	Yes
Confidence interval straddles 0.05	No	No
Entire confidence interval above 0.05	Yes	No

[a] This table assumes that close fit is defined as $\varepsilon \leq 0.05$. If hypotheses are constructed on the basis of some other value, ε_0, then that value becomes the reference point for relating confidence intervals to hypothesis tests.

- Sample size required to *reject* a null hypothesis with probability $\beta = 0.8$ can be computed
- Sample size required to reject $RMSEA < 0.05$ if the true $RMSEA = 0.8$ and $DF = 20$
  - Test for close fit, which we wish to reject if true RMSEA is high

```
library("semTools")
findRMSEAsamplesize(rmsea0=.05, rmseaA=.08, df=20, power=0.80)

## [1] 434
```

- Sample size required to reject $RMSEA > 0.05$ if the true $RMSEA = 0.1$ and $DF = 20$
  - Test for not-close fit, which we wish to reject if true RMSEA is low

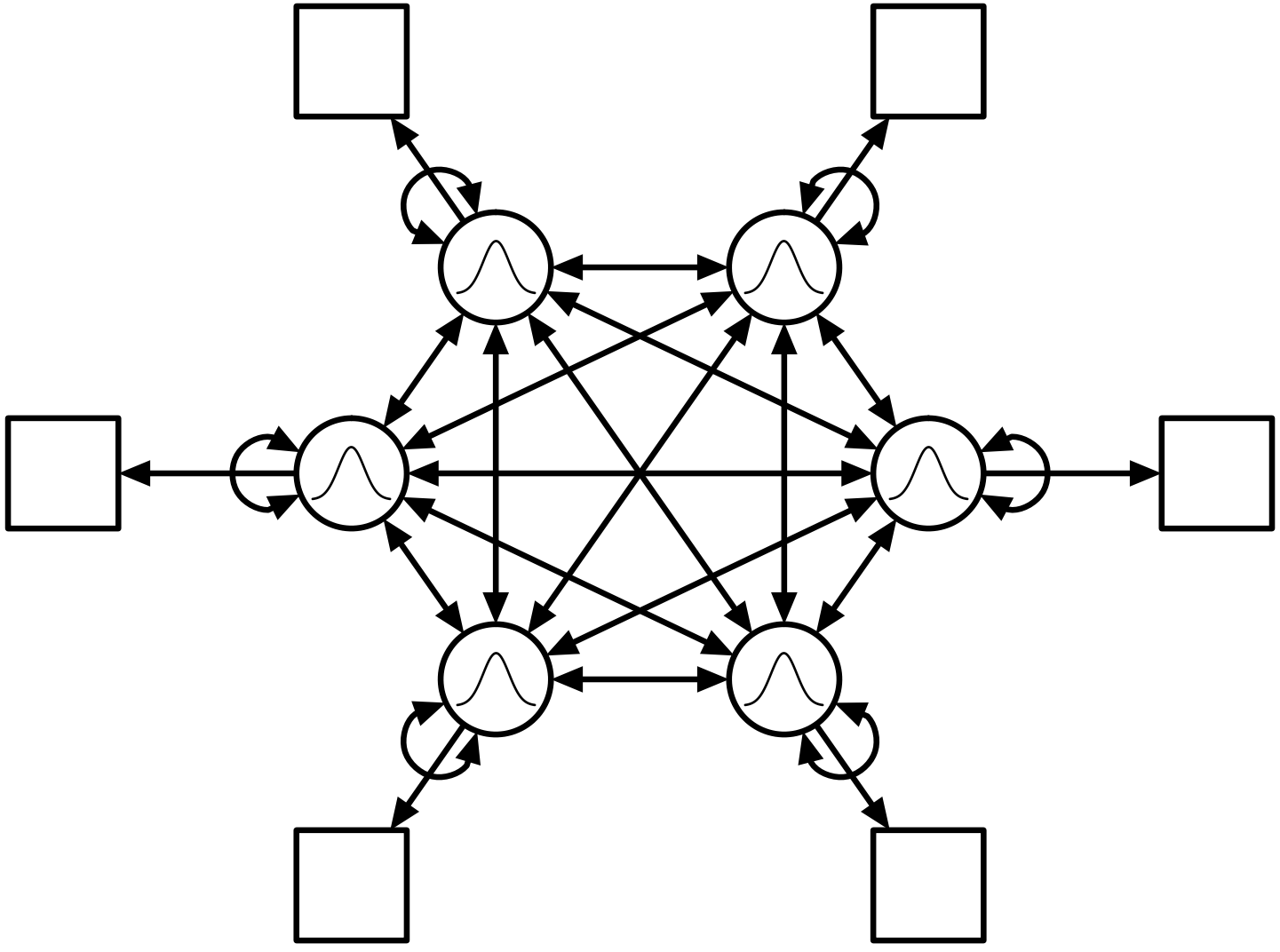
```
findRMSEAsamplesize(rmsea0=.05, rmseaA=.01, df=20, power=0.80)

## [1] 474
```

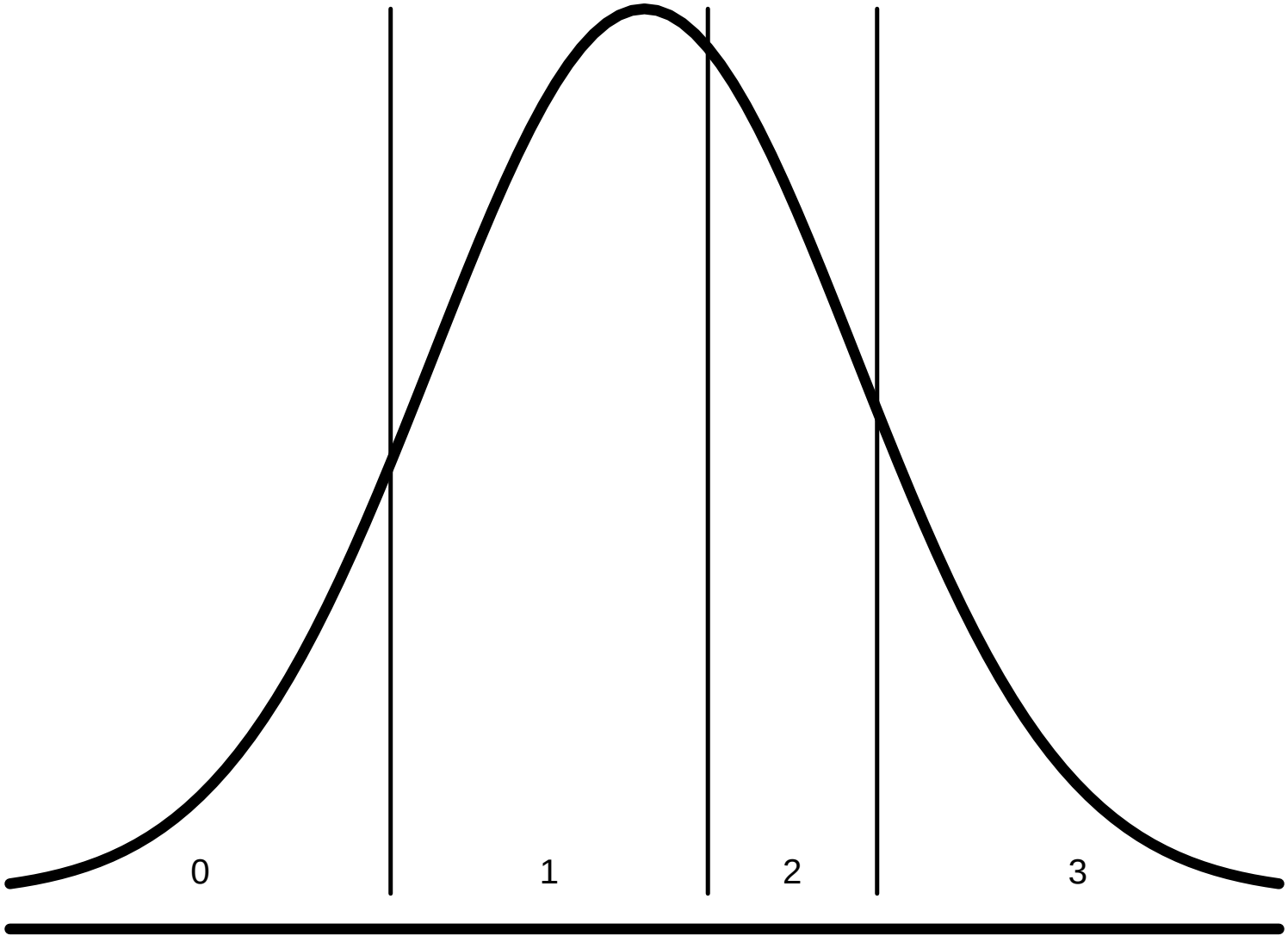
Ordinal data

- If data is ordinal and consists of only a few levels of measurement data cannot be assumed normal
  - Roughly less than five categories. Rhemtulla, M., Brosseau-Liard, P. É., & Savalei, V. (2012). When can categorical variables be treated as continuous? A comparison of robust continuous and categorical SEM estimation methods under suboptimal conditions. *Psychological methods, 17(3)*, 354–373.
- In this case threshold models should be used
- Then, it is assumed that underlying the response is a latent item that is normally distributed
- The covariance between this latent items and other such latent items or other continuous items can be estimated
  - Polychoric correlation if both variables are ordinal
  - Polyserial correlation if one item is ordinal and the other is continuous

I see myself as someone who is talkative

Disagree strongly	Disagree a little	Neither agree nor disagree	Agree a little	Agree Strongly
1	2	3	4	5




0
1
2
3

```r
set.seed(1)
# Setup:
sampleSize <- 1000
cor <- 0.5
thresh1 <- c(-2,0,2)
thresh2 <- c(-1,0.5,1.6)

# Generate data:
library("mvtnorm")
corMat <- matrix(c(1,0.5,0.5,1),2,2)
Data <- as.data.frame(rmvnorm(sampleSize, sigma = corMat))

# Make catagorical:
Data[,1] <- as.numeric(cut(Data[,1],breaks = c(-Inf,thresh1,Inf)))
Data[,2] <- as.numeric(cut(Data[,2],breaks = c(-Inf,thresh2,Inf)))
```

```r
# Pearson correlation:
cor(Data[,1], Data[,2])

## [1] 0.4076942

# Polychoric correlation:
library("lavaan")
DataOrdered <- Data
DataOrdered[,1] <- ordered(Data[,1])
DataOrdered[,2] <- ordered(Data[,2])
lavCor(DataOrdered)

##    V1    V2
## V1 1.000
## V2 0.499 1.000
```

Polychoric correlations

- Lavaan will automatically treat variables that are made ordered factors via `ordered()` as ordinal variables and will include thresholds
- Alternatively, the | operator can be used to define thresholds
- Polychoric and polyserial correlations relax the assumption of normality. However, they can sometimes go wrong!
- The crosstable should not have zero elements!
- When testing measurement invariance, now the thresholds need to be equated instead of intercepts

No thresholds:

```
table(Data)

##     V2
## V1     1   2   3   4
##    1  17  16   2   0
##    2 123 266  77   8
##    3  20 233 168  42
##    4   2   6  13   7
```

Zeroes.. So a bit dangerous!

No thresholds:

```
Model <- '
Fa =~ a
fb =~ b
Fa ~~ fb
'
names(Data) <- c("a","b")
fit <- cfa(Model, Data, std.lv = TRUE)
parameterEstimates(fit)

##   lhs op rhs   est    se      z pvalue ci.lower ci.upper
## 1  Fa =~   a 0.613 0.014 44.721      0    0.586    0.640
## 2  fb =~   b 0.778 0.017 44.721      0    0.744    0.812
## 3  Fa ~~  fb 0.408 0.026 15.463      0    0.356    0.459
## 4   a ~~   a 0.000 0.000     NA     NA    0.000    0.000
## 5   b ~~   b 0.000 0.000     NA     NA    0.000    0.000
## 6  Fa ~~  Fa 1.000 0.000     NA     NA    1.000    1.000
## 7  fb ~~  fb 1.000 0.000     NA     NA    1.000    1.000
```

Thresholds:

```
Model <- '
a ~~ b
a | t1 + t2 + t3
b | t1 + t2 + t3
'
names(Data) <- c("a","b")
fit <- cfa(Model, Data)
parameterEstimates(fit)
```

```
##    lhs  op rhs    est    se       z pvalue ci.lower ci.upper
## 1    a  ~~   b  0.499 0.029  17.143  0.000    0.442    0.556
## 2    a   |  t1 -1.812 0.075 -24.079  0.000   -1.959   -1.664
## 3    a   |  t2  0.023 0.040   0.569  0.569   -0.055    0.100
## 4    a   |  t3  1.911 0.081  23.524  0.000    1.752    2.070
## 5    b   |  t1 -0.986 0.048 -20.753  0.000   -1.079   -0.893
## 6    b   |  t2  0.476 0.041  11.519  0.000    0.395    0.557
## 7    b   |  t3  1.580 0.064  24.653  0.000    1.455    1.706
## 8    a  ~~   a  1.000 0.000      NA     NA    1.000    1.000
## 9    b  ~~   b  1.000 0.000      NA     NA    1.000    1.000
## 10   a ~*~   a  1.000 0.000      NA     NA    1.000    1.000
## 11   b ~*~   b  1.000 0.000      NA     NA    1.000    1.000
## 12   a  ~1      0.000 0.000      NA     NA    0.000    0.000
```

Or use data with ordered columns:

```
Model <- '
a ~~ b
'
names(DataOrdered) <- c("a","b")
fit <- cfa(Model, DataOrdered)
parameterEstimates(fit)

##    lhs  op rhs    est    se       z pvalue ci.lower ci.upper
## 1    a  ~~   b  0.499 0.029  17.143  0.000    0.442    0.556
## 2    a   |  t1 -1.812 0.075 -24.079  0.000   -1.959   -1.664
## 3    a   |  t2  0.023 0.040   0.569  0.569   -0.055    0.100
## 4    a   |  t3  1.911 0.081  23.524  0.000    1.752    2.070
## 5    b   |  t1 -0.986 0.048 -20.753  0.000   -1.079   -0.893
## 6    b   |  t2  0.476 0.041  11.519  0.000    0.395    0.557
## 7    b   |  t3  1.580 0.064  24.653  0.000    1.455    1.706
## 8    a  ~~   a  1.000 0.000      NA     NA    1.000    1.000
## 9    b  ~~   b  1.000 0.000      NA     NA    1.000    1.000
## 10   a ~*~   a  1.000 0.000      NA     NA    1.000    1.000
## 11   b ~*~   b  1.000 0.000      NA     NA    1.000    1.000
## 12   a  ~1      0.000 0.000      NA     NA    0.000    0.000
## 13   b  ~1      0.000 0.000      NA     NA    0.000    0.000
```

Why are data missing? In a general X predicts Y case:

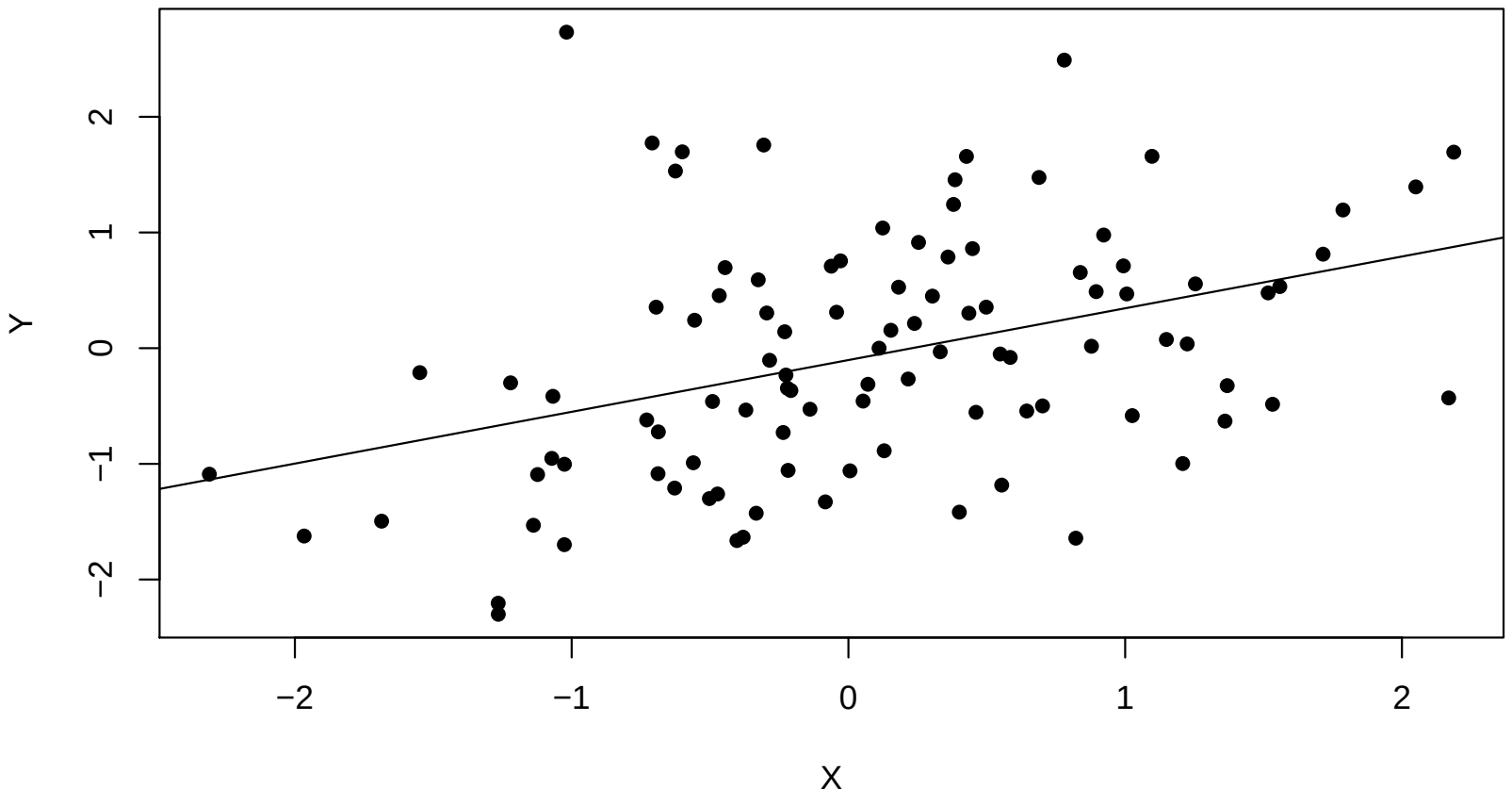
- **Missing completely at random** (MCAR)
  - Missingness is independent of Y or X
  - Everything is fine!
- **Missing at random** (MAR)
  - Missingness is independent of Y, but not of X
  - Example: Men less willing to respond to mental health questionnaire
  - Not a big problem
- **Missing not at random** (MNAR)
  - Missingness depends on Y
  - Example: People with severe mental health problems fill in less questionnaires
  - This is bad :(

Unfortunatly, there is no way to know how your data is missing.

A dataset:

X	Y
5	5
6	5
5	6
8	5
6	7
7	7
6	9
9	8
9	9
12	9

A larger dataset:

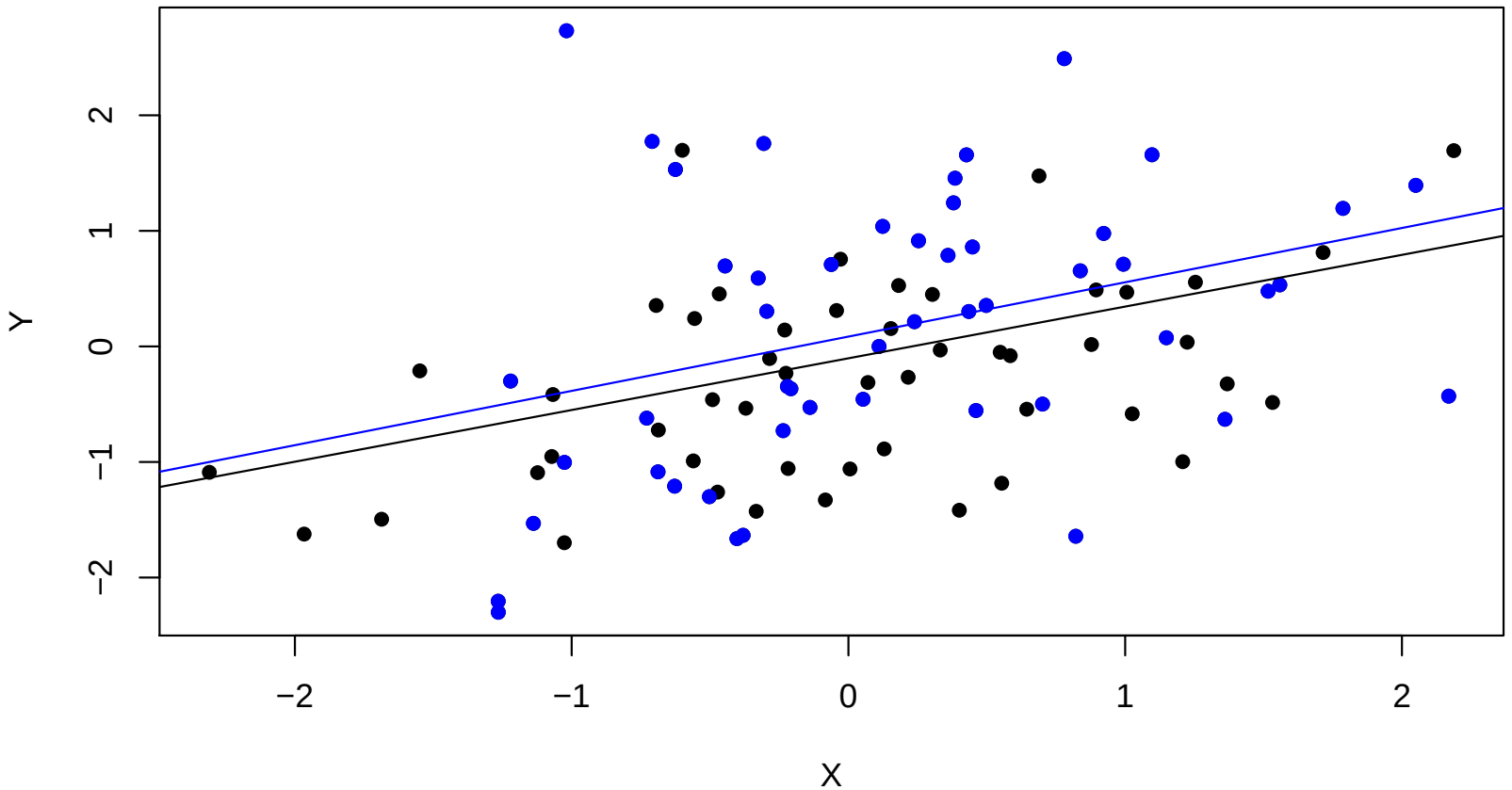


Missing completely at random (MCAR):

X	Y
5	5
6	5
5	6
8	5
6	7
7	7
6	9
9	8
9	9
12	9

MCAR

A larger dataset:



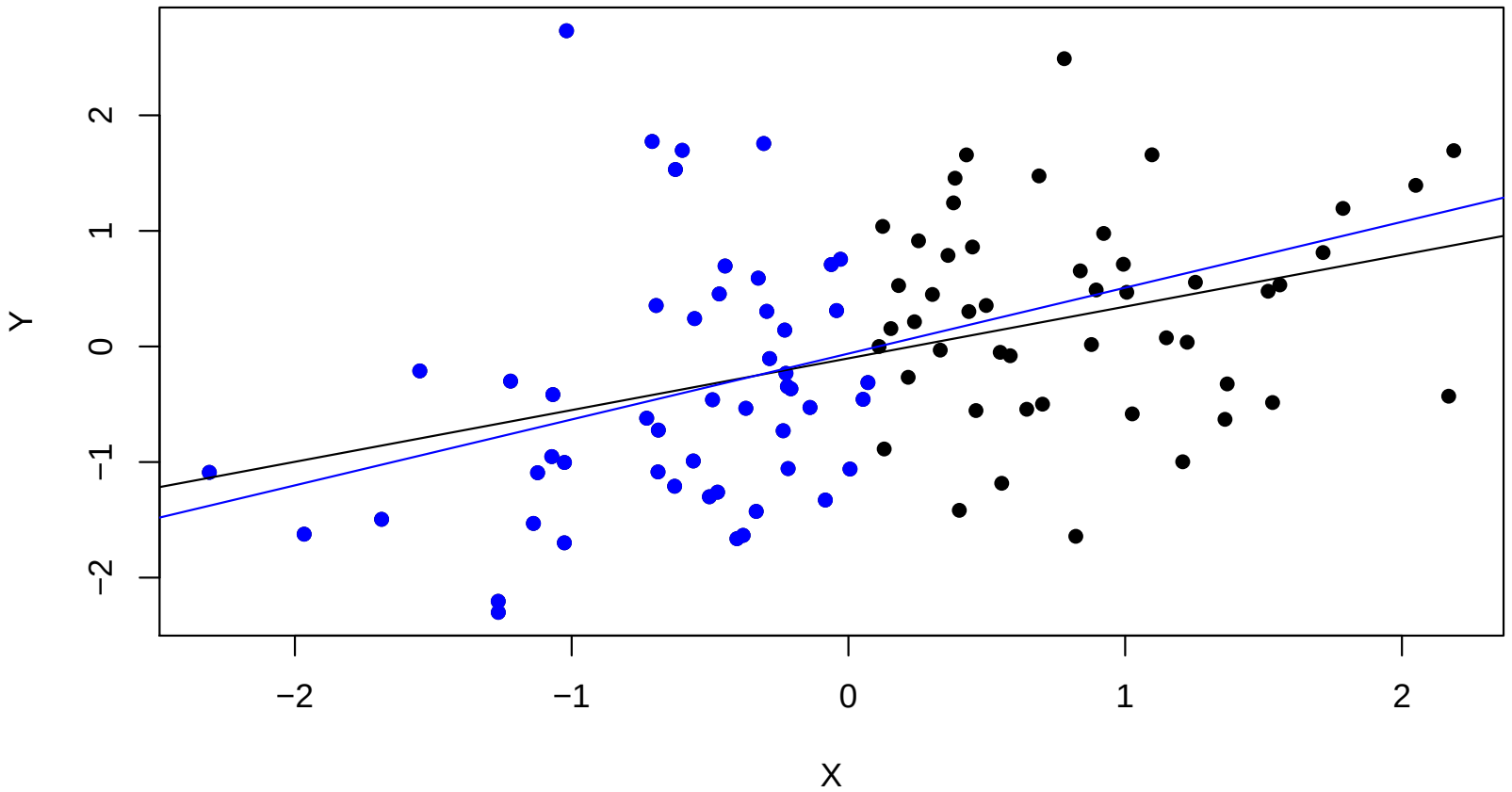
Blue = observed

Missing at random (MAR):

X	Y
5	5
6	5
5	6
8	5
6	7
7	7
6	9
9	8
9	9
12	9

MAR

A larger dataset:

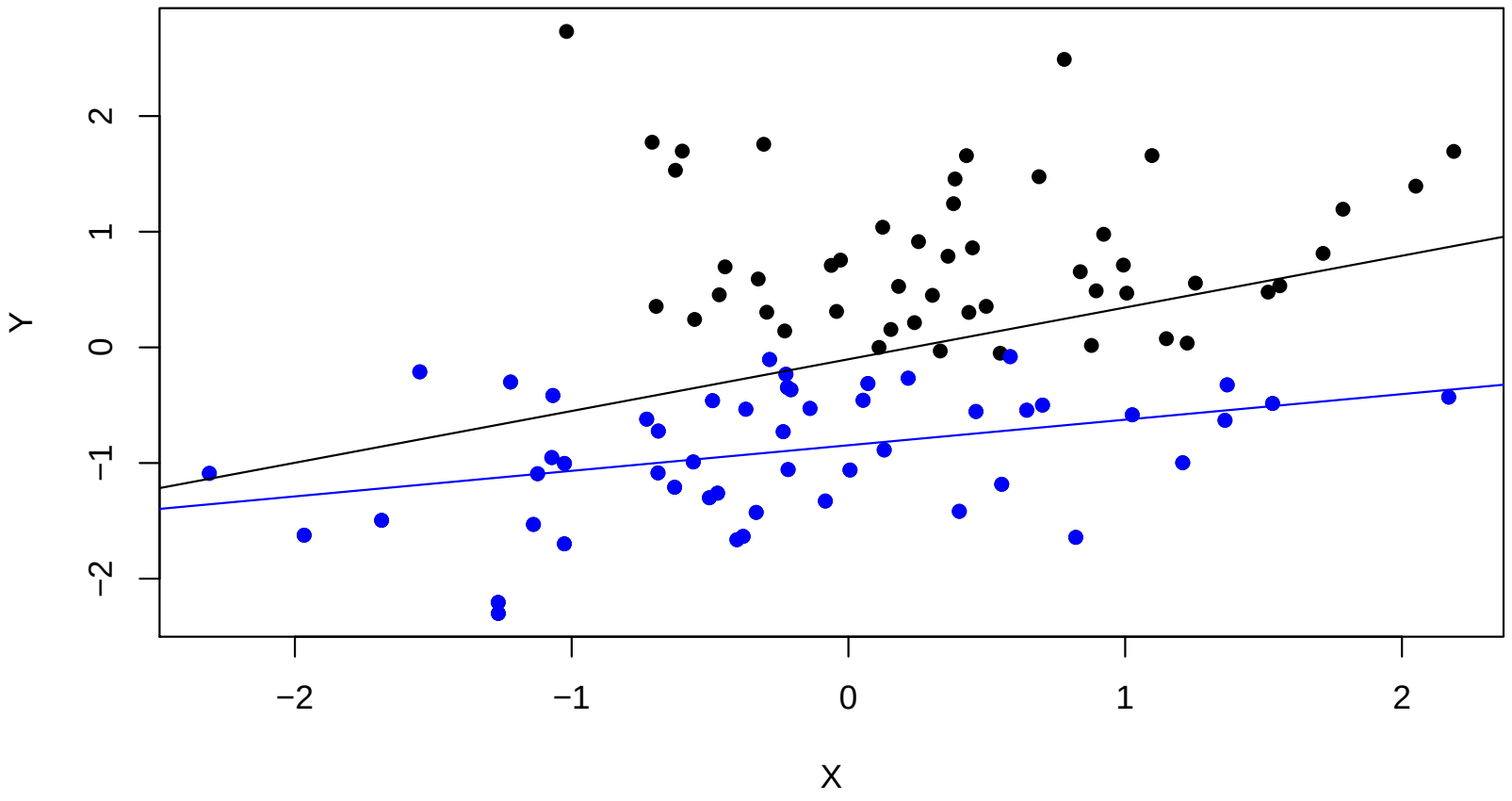


Blue = observed

Missing not at random (MNAR):

X	Y
5	5
6	5
5	6
8	5
6	7
7	7
6	9
9	8
9	9
12	9

MNAR



Blue = observed

Missing data

- Best case: no missings
- MCAR or MAR: this is ok
- MNAR: This is not ok
- Unfortunatly, no real statistical way to checking if missings are MNAR
- Thus, MAR needs to be assumed to continue

Old ways of handling missing data

- Compute $\boldsymbol{S}$ using list-wise deletion
  - Delete all rows with a missing value
  - Downside: deletes observed data
- Compute $\boldsymbol{S}$ using pair-wise estimation
  - Estimate each element of $\boldsymbol{S}$ using all available data
  - Downside: Each covariance is based on different n
- (multiple) inputations
  - Inpute missingness using mean scores or regression models
  - Downside: complicated, can increase bias if MNAR

Modern way: full-information maximum likelihood (FIML)

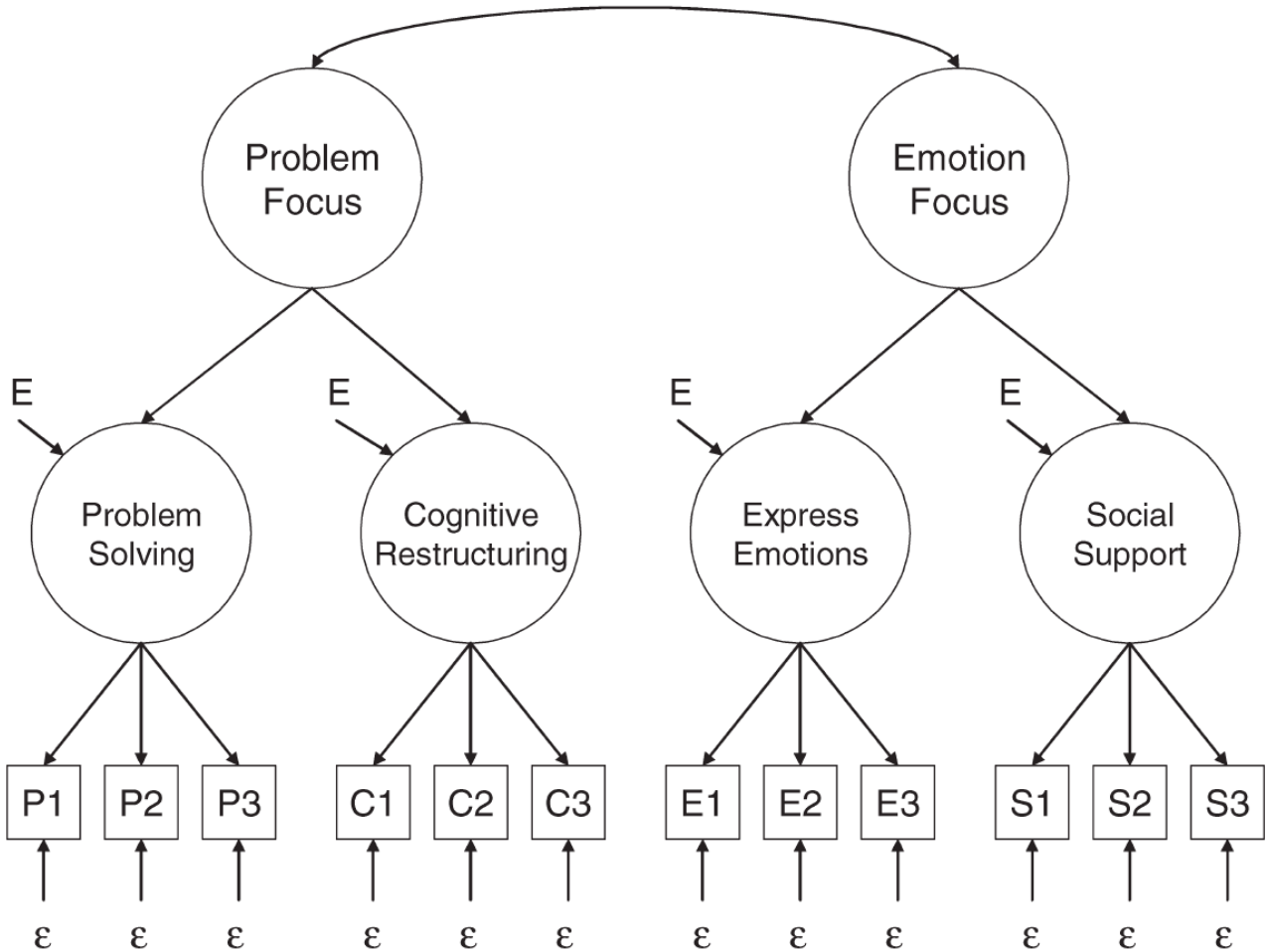
- Uses the full data set and all observations
- Downside: full data needed (analysis can not be done using covariance matrix)
- Assumes MAR

```
fit <- cfa(model, data, missing = "FIML")
```

Assumptions of ML

1. Independence: Observations are a simple random sample from some population
   - Consequence: underestimated standard errors, inflated Type-I error rates
   - Solution 1: use SE correction for dependence structure
   - Solution 2: multilevel SEM
2. Multivariate Normality: Variables are univariate normally distributed at levels of all other variables, residuals are normal and homoscedastic, latent variables are normal, bivariate relations are linear
   - Consequence: standard errors are incorrect (probably too low), Type-I error rate is not accurate (probably too high)
   - Solution 1: use robust standard errors (`estimator = 'MLM'`, with complete data; `estimator = 'MLR'` with incomplete data
   - Solution 2: use bootstrapped standard errors & test statistic

Higher order models



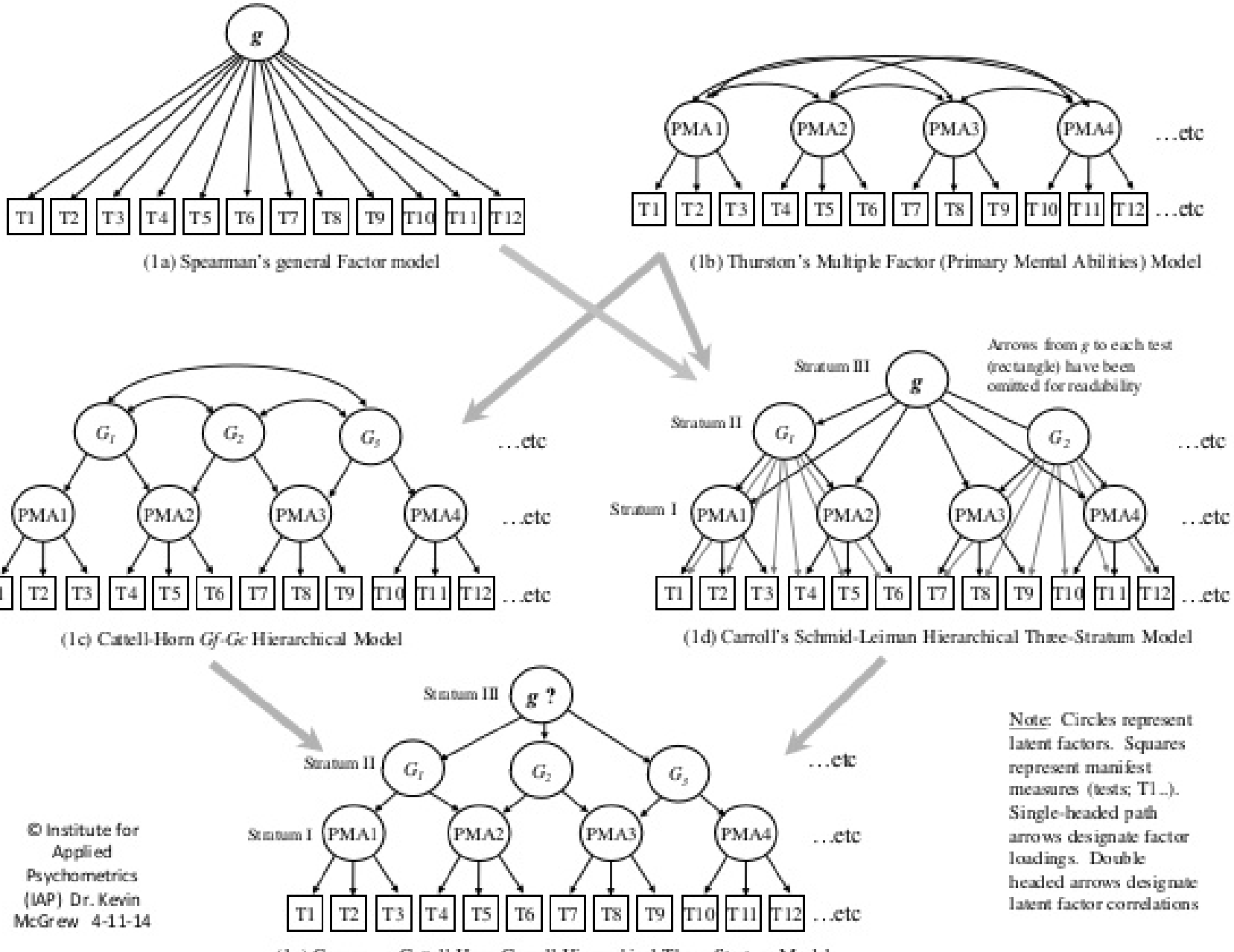
Higher order models

Mathematically, simply a second factor model on the latent variable variance–covariance matrix:

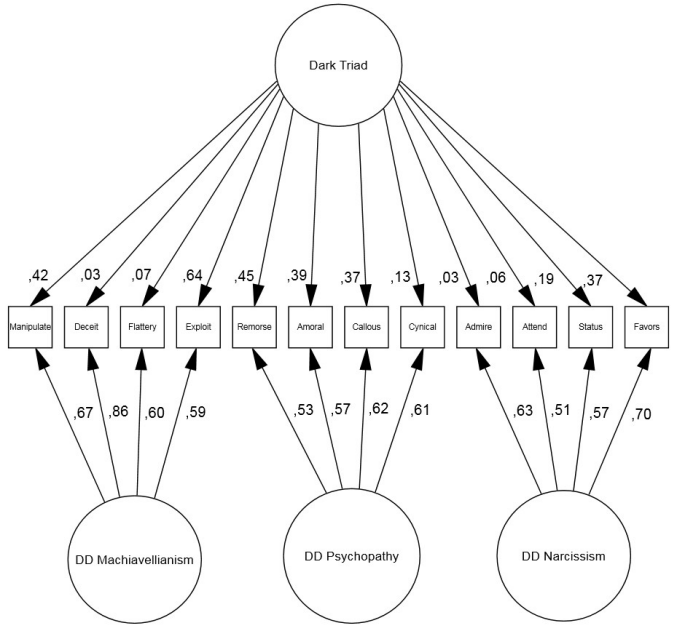
$$\boldsymbol{\Psi} = \boldsymbol{\Lambda}^{*}\boldsymbol{\Psi}^{*}\boldsymbol{\Lambda}^{*\top} + \boldsymbol{\Theta}^{*}$$

Same rules of identification apply:

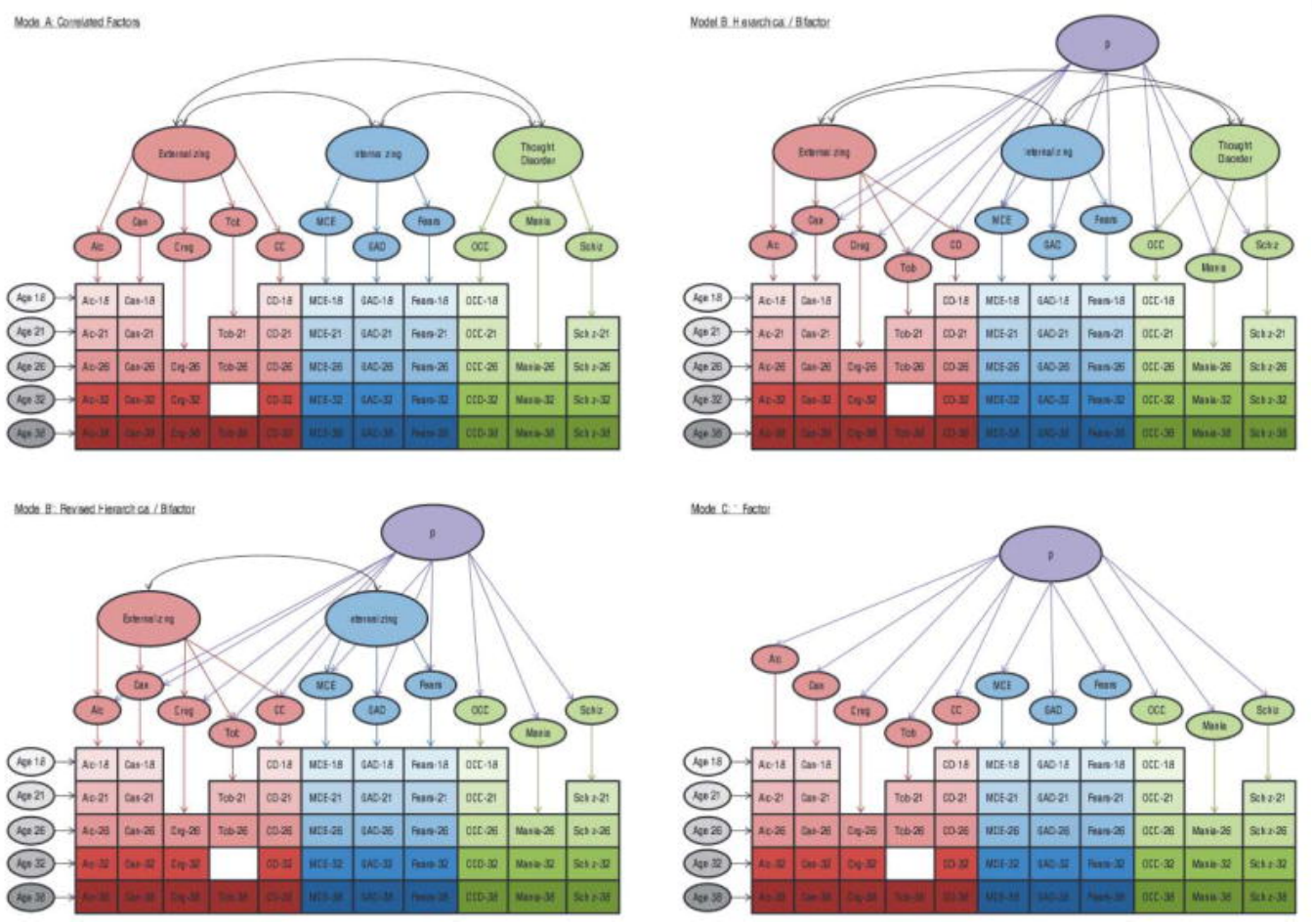
- The higher order factor must be scaled (one factor loading or the variance fixed to 1)
- The number of variances and covariances in $\boldsymbol{\Psi}$ must be at least as much as the number of parameters used to model $\boldsymbol{\Psi}$


g
T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12
(1a) Spearman's general Factor model
PMA1 PMA2 PMA3 PMA4 ...etc
T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 ...etc
(1b) Thurston's Multiple Factor (Primary Mental Abilities) Model
Gf Gc Gs ...etc
PMA1 PMA2 PMA3 PMA4 ...etc
T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 ...etc
(1c) Cattell-Horn Gf-Gc Hierarchical Model
Stratum III
g
Arrows from g to each test (rectangle) have been omitted for readability
Stratum II
Gf Gc ...etc
Stratum I
PMA1 PMA2 PMA3 PMA4 ...etc
T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 ...etc
(1d) Carroll's Schmid-Leiman Hierarchical Three-Stratum Model
Stratum III
g ?
Stratum II
Gf Gc Gs ...etc
Stratum I
PMA1 PMA2 PMA3 PMA4 ...etc
T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 ...etc
© Institute for Applied Psychometrics (IAP) Dr. Kevin McGrew 4-11-14
Note: Circles represent latent factors. Squares represent manifest measures (tests; T1...). Single-headed path arrows designate factor loadings. Double headed arrows designate latent factor correlations

Bi-factor models

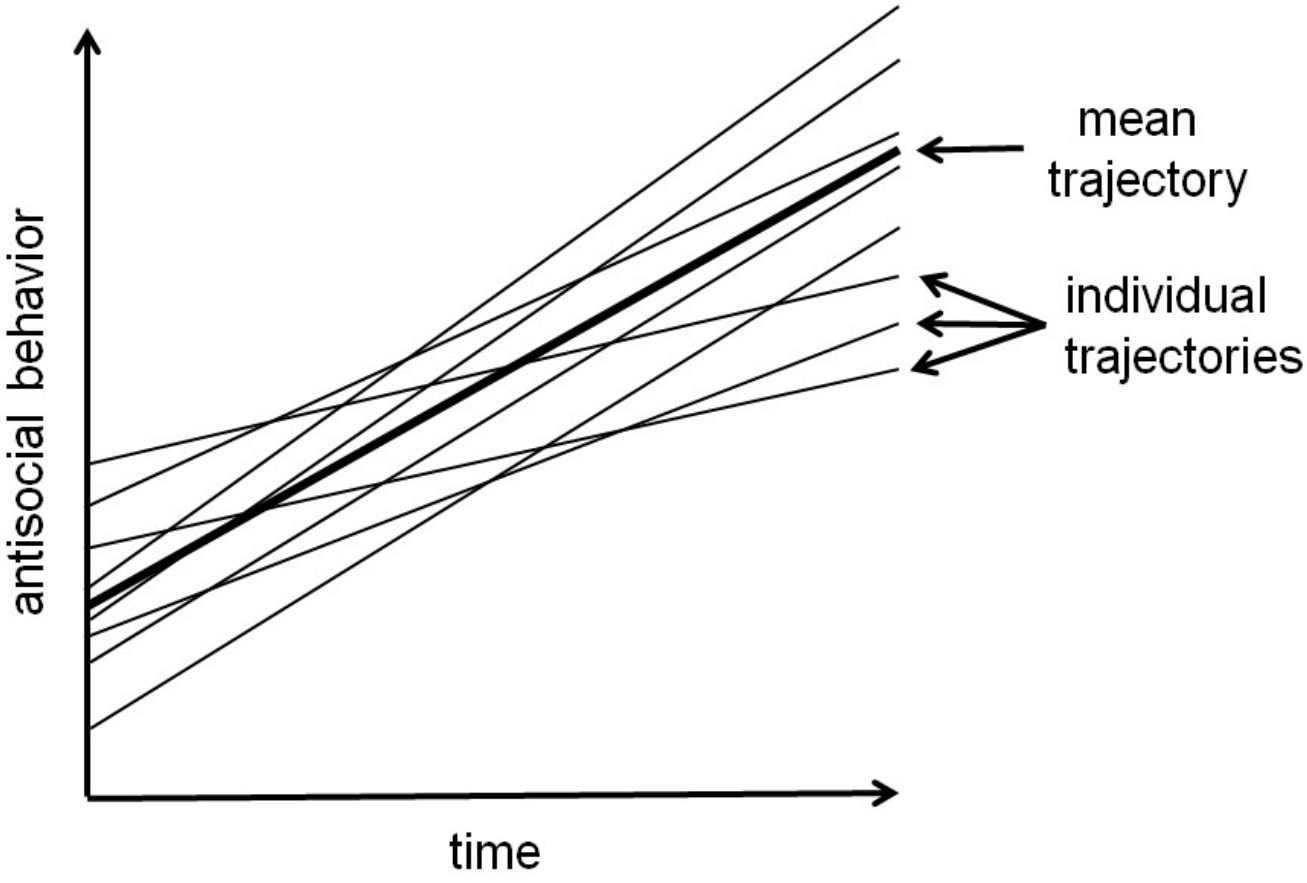


- Uncorrelated factors in combination with an uncorrelated bifactor
- Higher order model is nested in the bi-factor model
- Increasingly popular, but hard to interpret

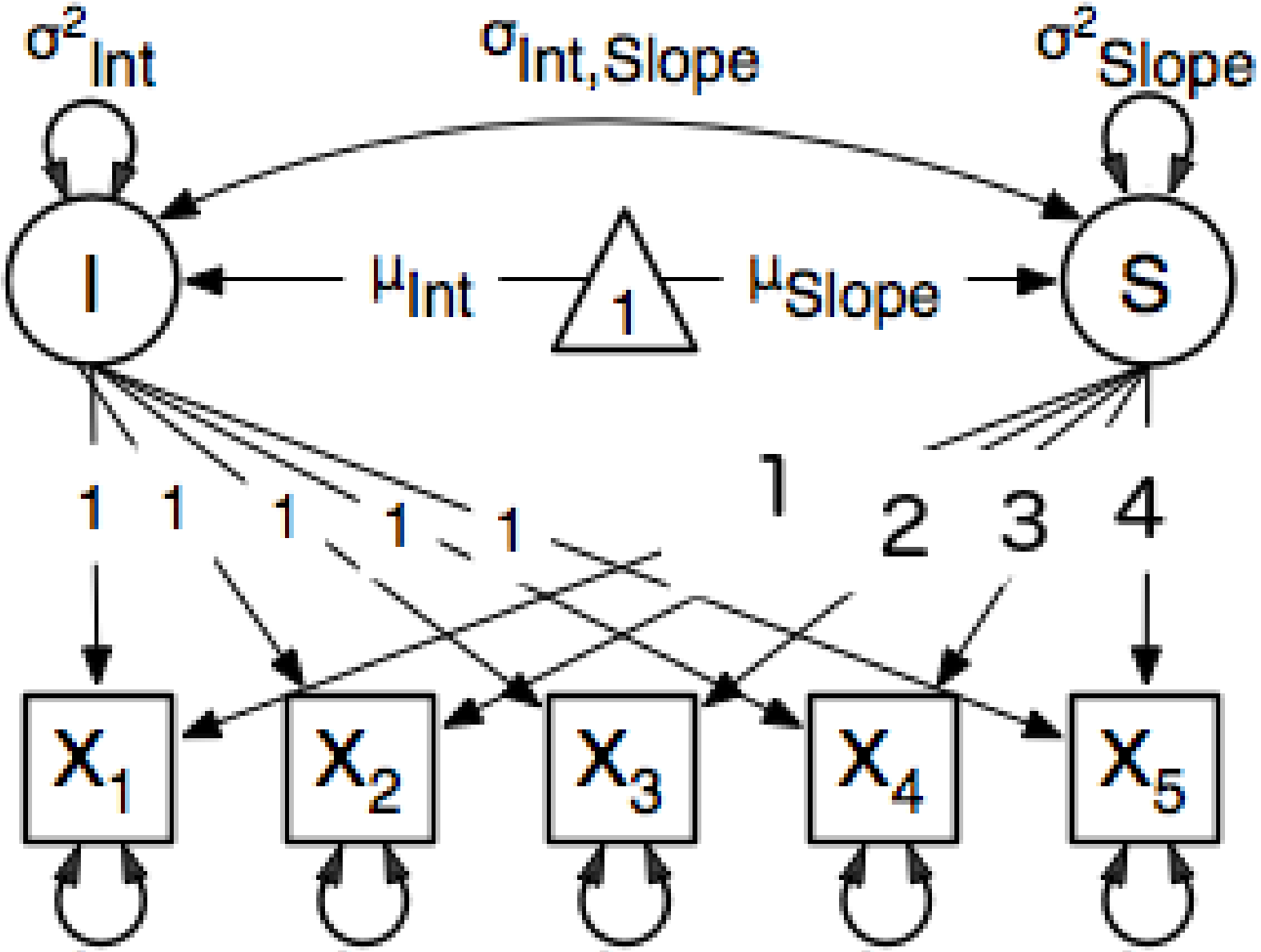


Caspi, A., Houts, R. M., Belsky, D. W., Goldman-Mellor, S. J., Harrington, H., Israel, S., ... & Moffitt, T. E. (2014). The p factor: one general psychopathology factor in the structure of psychiatric disorders?. Clinical Psychological Science, 2(2), 119-137.

Latent growth models



Latent growth models



Exploratory Factor Analysis (EFA)

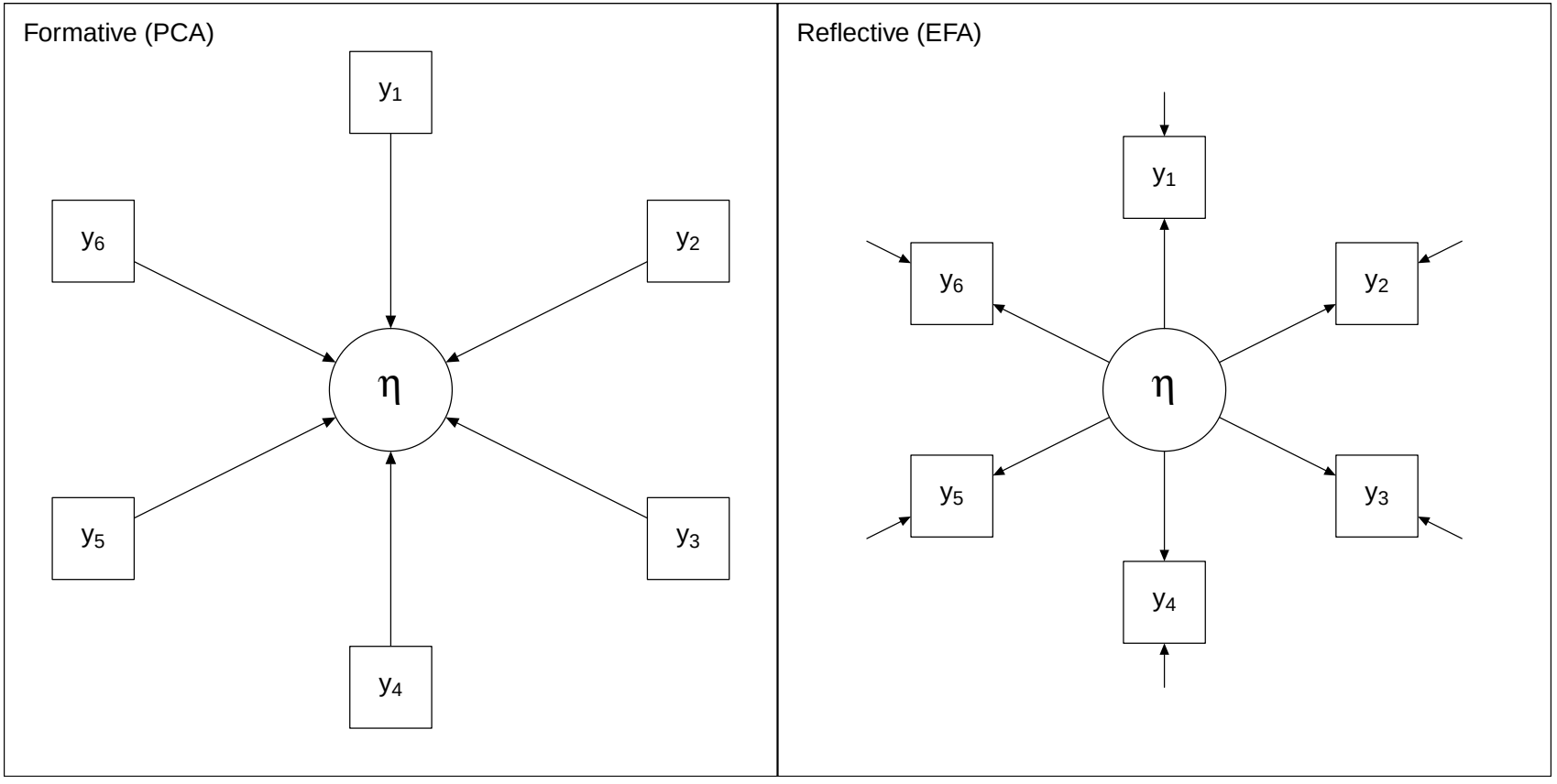
Exploratorily estimate $\mathbf{\Lambda}$ (no free elements in $\mathbf{\Lambda}$):

$$\mathbf{\Sigma} = \mathbf{\Lambda}\mathbf{\Psi}\mathbf{\Lambda}^\top + \mathbf{\Theta}$$

Very close, but not the same (!!) as principal component analysis (PCA):

$$\mathbf{\Sigma} = \mathbf{\Lambda}\mathbf{\Psi}\mathbf{\Lambda}^\top$$

Very different interpretation. EFA *measures* latents (there is measurement error), PCA only *summarizes* the data.


Formative (PCA)
y1
y6
y2
η
y5
y3
y4
Reflective (EFA)
y1
y6
y2
η
y5
y3
y4

Exploratory Factor Analysis (EFA)

If $\boldsymbol{\Lambda}$ is not somehow constrained, latent variable variance is not identified. We can arbitrarily add rotation matrices $\boldsymbol{T}$ and not change the decomposition:

$$\boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{T}\boldsymbol{T}^{-1}\boldsymbol{\Psi}\boldsymbol{T}^{-1\top}\boldsymbol{T}^{\top}\boldsymbol{\Lambda}^{\top} + \boldsymbol{\Theta}$$

Can be seen as a different factor model with $\boldsymbol{\Lambda}^* = \boldsymbol{\Lambda}\boldsymbol{T}$ and $\boldsymbol{\Psi}^* = \boldsymbol{T}^{-1}\boldsymbol{\Psi}\boldsymbol{T}^{-1\top}$. To this end, in estimation one can assume uncorrelated factors, $\boldsymbol{\Psi} = \boldsymbol{I}$. Afterwards, rotation methods can be used to obtain simple structure for $\boldsymbol{\Lambda}$ while possibly allowing factors to correlate:

- orthogonal (varimax): axes remain orthogonal, independent
- oblique (promax/oblimin): axes become correlated

I always use promax.

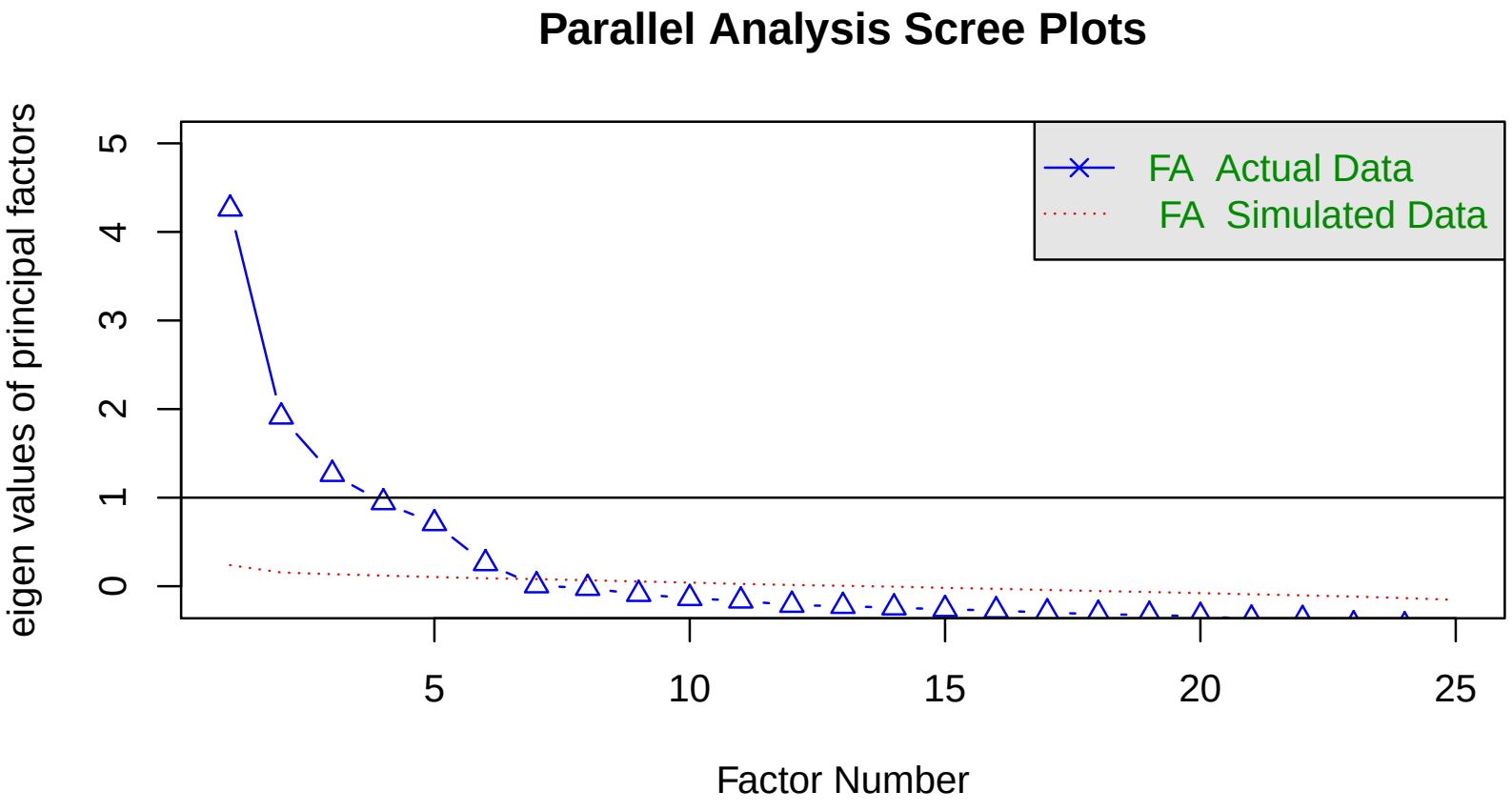
Choosing the number of Factors is a bit more involved than PCA

- One method involves checking how many eigenvalues in $\boldsymbol{S} - \hat{\boldsymbol{\Theta}}$ are above 0
  - $\hat{\boldsymbol{\Theta}}$ is then estimated using a 1-factor model
- Parallel analysis takes sampling variation into account, and checks how many eigenvalues are statistically above what can be expected given an independence model

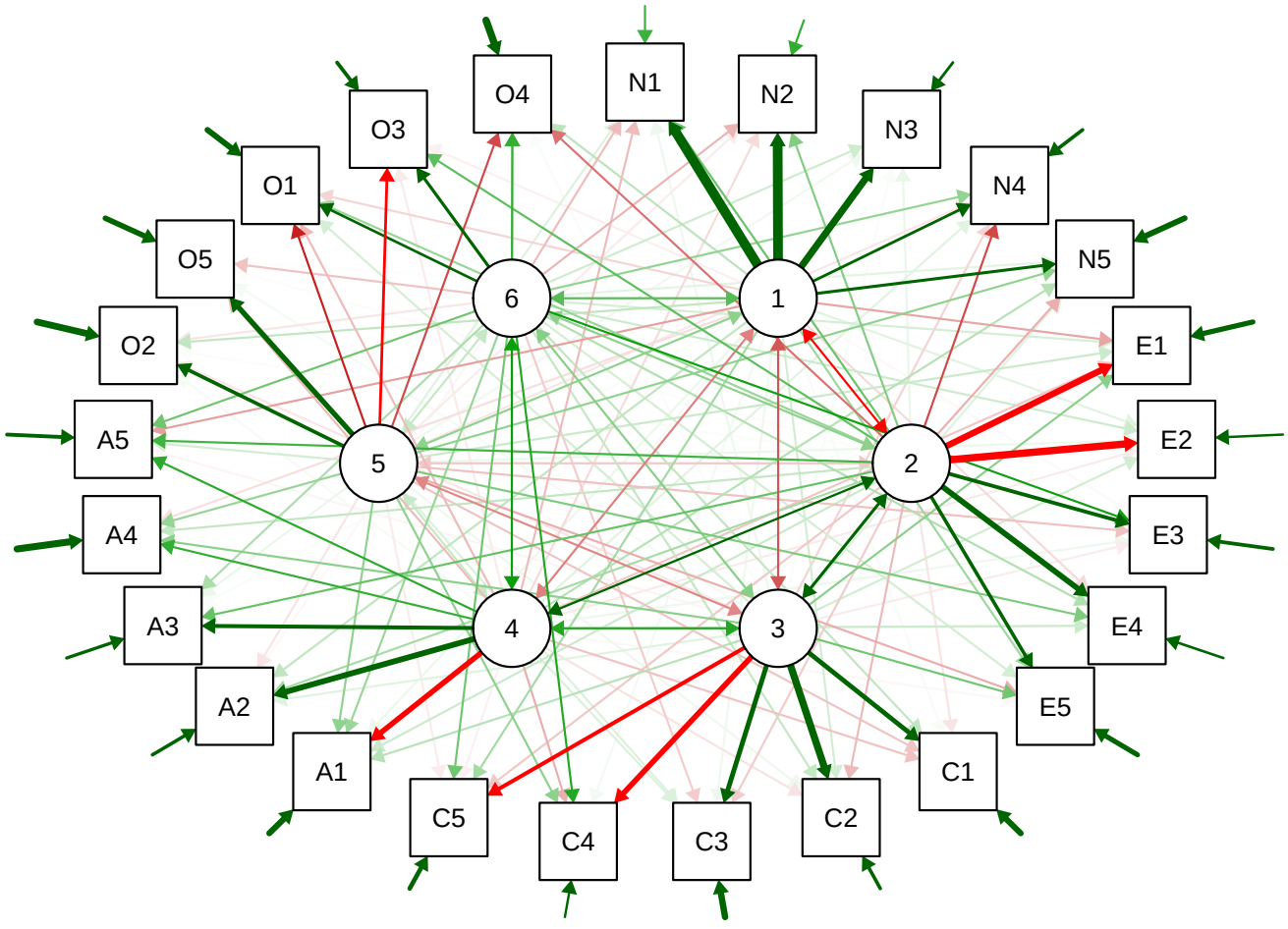
BFI example

```
library("psych")

##
## Attaching package:  'psych'
## The following object is masked from 'package:semTools':
##
##     skew
## The following object is masked from 'package:lavaan':
##
##     cor2cov

# Load data:
data(bfi)
bfiSub <- bfi[,1:25]

# Correlations:
corMat <- cor(bfiSub, use = "pairwise.complete.obs")
N <- nrow(bfiSub)
```

```
fa.parallel(corMat, N, fa = "fa")

## Parallel analysis suggests that the number of factors =  6  and the
```



```
## Loading required namespace: GPArotation
```



- When data are ordinal, polychoric and polyserial correlations can be computed
- Missing data needs assumption of missing at random (MAR)
- Advanced CFA models:
  - Higher-order models
  - Bi-factor models
  - Latent growth models
- Exploratory factor analysis can be used when no prior theory is available