The 'Structural After Measurement' (SAM) approach to SEM

Yves Rosseel
Department of Data Analysis
Ghent University – Belgium

Pre-Conference Workshop – Meeting of the SEM Working Group
TU Chemnitz, 19 March, 2025

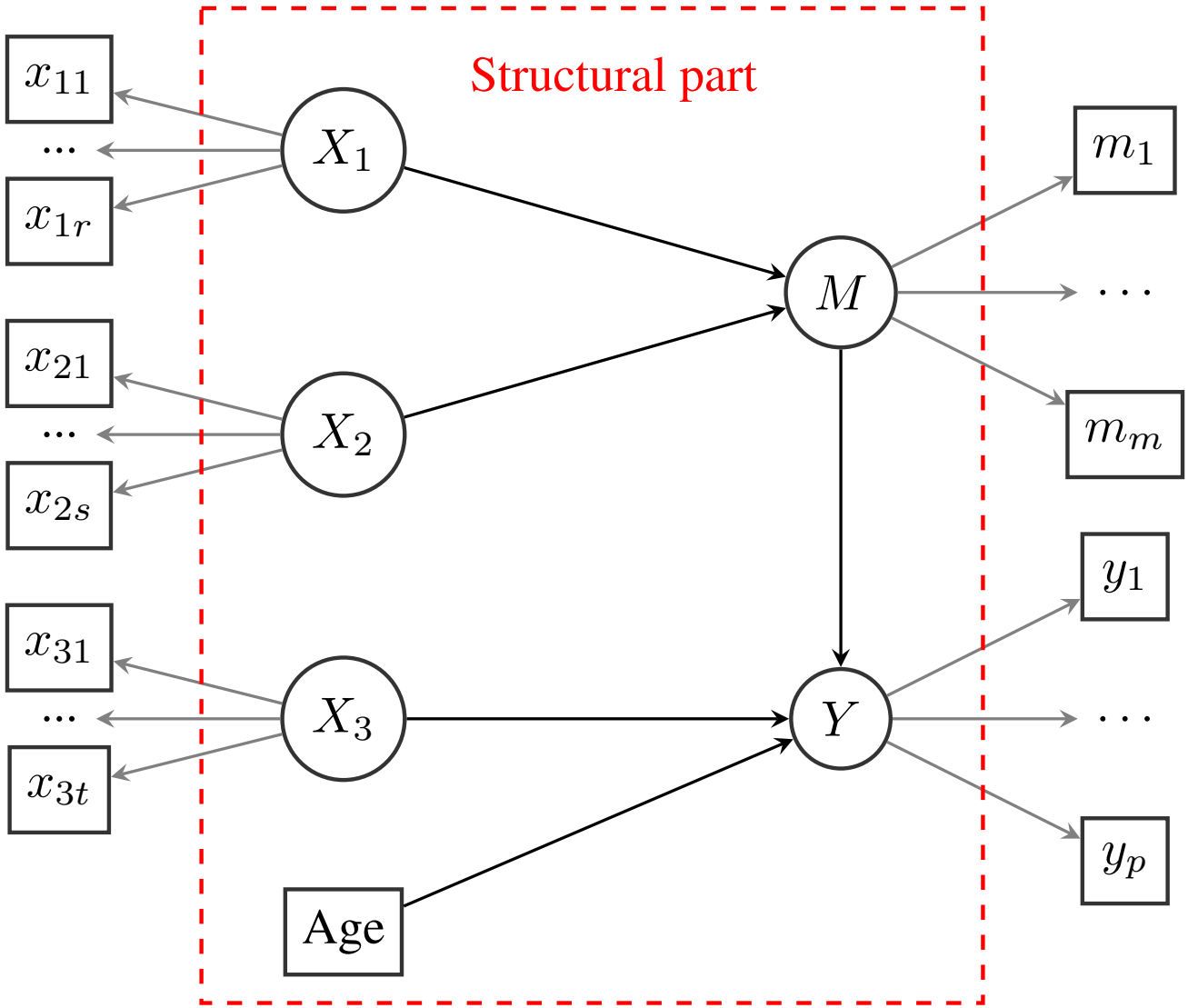
Contents

1 Introduction

1.1 Overview of the workshop

- provide an overview of SAM approaches:
  - replacing the latent variables by ('factor') scores
  - computing summary statistics—the mean vector and (co)variance matrix—for the latent variables
  - the single-indicator (SI) approach
  - global SAM
- why/when should we (not) consider SAM approaches?
- special attention: local SAM
- which one to choose? Consistency versus mean squared error (MSE)
- if time permits, practical session

1.2 The setting



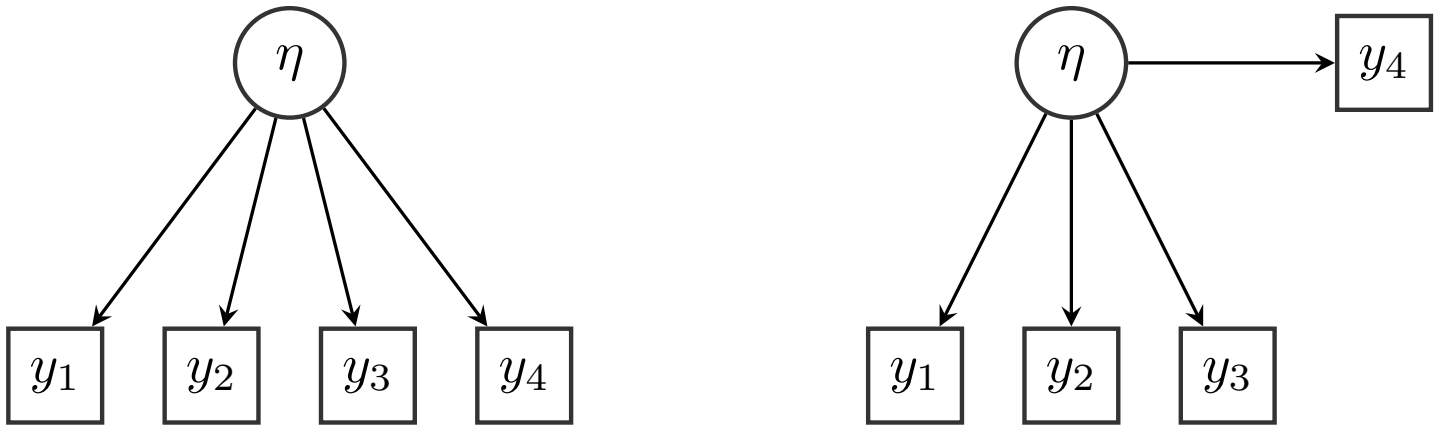
- we focus on 'large' models with many (say, > 100) parameters:
  - many constructs (motivation, ability, personality traits, ...)
  - each construct is measured by a set of (observed) indicators
  - many 'background' variables (age, gender, ...)
  - multilevel data, missing data, ...
- we are mostly interested in the structural part of the model:
  - if not saturated: how well does the structural part fit?
  - size of direct/indirect effects, hypothesis testing
- **assumption: the measurement instruments for the latent variables are well established, and fit (reasonably) well**
  - SAM approaches should not be used to 'test' the quality of your measurement instruments
- in addition: the sample size may not be very large (say, $N = 150$)

1.3 The standard estimation approach in SEM

- all parameters (measurement and structural) are estimated jointly
- we call this 'joint' or 'system-wide' estimation (Bollen, 1996)
- frequentist: typically using an iterative optimization approach
  - maximum likelihood (normal or Wishart) (ML)
  - generalized least squares (GLS)
  - (diagonally) weighted least squares, (D)WLS
- Bayesian: typically using MCMC
- advantages:
  - one-step, and therefore efficient (in terms of sampling variability)
  - inference is straightforward (standard errors, hypothesis testing)
  - (relatively) easy to handle constraints, missing data, ...
  - well-known, well-studied
  - widely available in software

- works very well if the following conditions are met:
  - correctly specified model
  - large sample size
  - (normally distributed data)
- but under less ideal circumstances, system-wide estimation does not (always) work well:
  - bias
  - instability (small changes in the data lead to very different results)
  - nonconvergence
  - improper solutions
- in addition: combining 'structural' and 'measurement' is not a good idea, due to a phenomenon called 'interpretational confounding'

example interpretational confounding (stolen from Roy Levy, 2023)



- left panel: only measurement
- right panel: measurement + structural
- conceptually very different
- mathematically identical (in system-wide SEM)

generate some data

```
> library(lavaan)
> Sigma <- matrix(c(2.0, 1.0, 1.0, 1.0, 1.5,
                    1.0, 2.0, 1.0, 1.0, 0.1,
                    1.0, 1.0, 2.0, 1.0, 0.1,
                    1.0, 1.0, 1.0, 2.0, 0.1,
                    1.5, 0.1, 0.1, 0.1, 2.0), nrow = 5, ncol = 5)
> rownames(Sigma) <- colnames(Sigma) <- c("y1", "y2", "y3", "y4", "z")
> Sigma
```

```
    y1  y2  y3  y4   z
y1 2.0 1.0 1.0 1.0 1.5
y2 1.0 2.0 1.0 1.0 0.1
y3 1.0 1.0 2.0 1.0 0.1
y4 1.0 1.0 1.0 2.0 0.1
z  1.5 0.1 0.1 0.1 2.0
```

```
> set.seed(3)
> Data <- MASS::mvrnorm(n = 200L, mu = rep(0, 5), Sigma = Sigma)
```

R code left panel (model1)

```
> model1 <- '
      f =~ y1 + y2 + y3 + y4
  '
> fit1 <- sem(model1, data = Data)
> summary(fit1)

lavaan 0.6-20.2277 ended normally after 25 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                         8

  Number of observations                           200

Model Test User Model:

  Test statistic                                 0.140
  Degrees of freedom                                 2
  P-value (Chi-square)                           0.932

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured
```

```
Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  f =~
    y1                1.000
    y2                0.996    0.123    8.122    0.000
    y3                0.894    0.116    7.689    0.000
    y4                0.911    0.120    7.608    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .y1                1.096    0.148    7.388    0.000
   .y2                0.837    0.127    6.601    0.000
   .y3                0.987    0.128    7.694    0.000
   .y4                1.081    0.138    7.820    0.000
    f                 1.031    0.208    4.959    0.000
```

R code right panel (model2)

```
> model2 <- '
     f =~ y1 + y2 + y3
     y4 ~ f
  '
> fit2 <- sem(model2, data = Data)
> summary(fit2)

lavaan 0.6-20.2277 ended normally after 26 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                         8

  Number of observations                           200

Model Test User Model:

  Test statistic                                 0.140
  Degrees of freedom                                 2
  P-value (Chi-square)                           0.932

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured
```

```
Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  f =~
    y1                1.000
    y2                0.996    0.123    8.122    0.000
    y3                0.894    0.116    7.689    0.000

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  y4 ~
    f                 0.911    0.120    7.608    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .y1                1.096    0.148    7.388    0.000
   .y2                0.837    0.127    6.601    0.000
   .y3                0.987    0.128    7.694    0.000
   .y4                1.081    0.138    7.820    0.000
    f                 1.031    0.208    4.959    0.000
```

change outcome variable (y4 becomes z) (model3)

```
> model3 <- '
      f =~ y1 + y2 + y3
      z ~ f
  '
> fit3 <- sem(model3, data = Data)
> summary(fit3)

lavaan 0.6-20.2277 ended normally after 58 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                         8

  Number of observations                           200

Model Test User Model:

  Test statistic                                50.659
  Degrees of freedom                                 2
  P-value (Chi-square)                           0.000

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured
```

```
Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  f =~
    y1                1.000
    y2                0.138    0.053    2.601    0.009
    y3                0.133    0.051    2.591    0.010

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  z ~
    f                 0.276    0.087    3.156    0.002

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .y1               -4.169    1.801   -2.315    0.021
   .y2                1.740    0.173   10.071    0.000
   .y3                1.699    0.168   10.086    0.000
   .z                 1.677    0.211    7.953    0.000
    f                 6.296    1.715    3.671    0.000
```

interpretational confounding

- replacing y_4 by z (= changing the structural part) also changes the parameters of the measurement model
- if the resulting parameters of the measurement model imply a different 'meaning' of the latent variable than was intended by the researcher, we have a problem
- this problem was coined "interpretational confounding" by Burt (1976)

  > Burt, R.S. (1976). Interpretational confounding of unobserved variables in structural equation models. *Sociological Methods & Research, 5*(1), 3–52.

- Burt (1976) already suggested the solution: first fit the measurement part of the model, and then fit the structural part of the model

solution: replace sem() by sam()

```
> fit3.sam <- sam(model3, data = Data)
> parameterEstimates(fit3.sam, remove.step1 = FALSE, ci = FALSE,
                     output = "text")
```

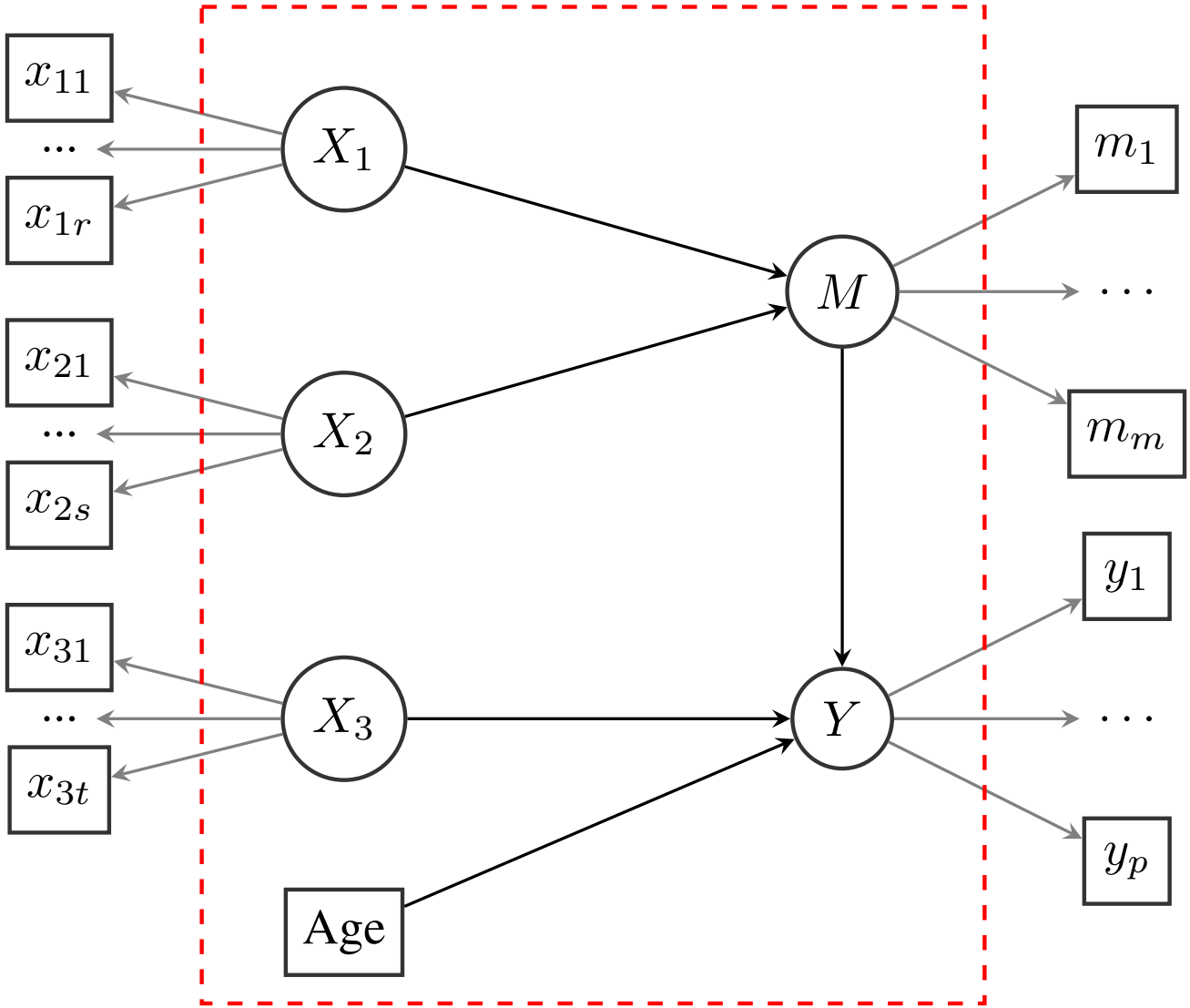
```
Latent Variables:
                   Step Estimate  Std.Err  z-value  P(>|z|)
  f =~
    y1                1    1.000
    y2                1    0.969    0.139    6.977    0.000
    y3                1    0.884    0.127    6.979    0.000

Regressions:
                   Step Estimate  Std.Err  z-value  P(>|z|)
  z ~
    f                 2    0.540    0.121    4.471    0.000

Variances:
                   Step Estimate  Std.Err  z-value  P(>|z|)
   .y1                1    1.067    0.170    6.287    0.000
   .y2                1    0.865    0.151    5.723    0.000
   .y3                1    0.983    0.142    6.900    0.000
   .z                 2    1.846    0.195    9.462    0.000
    f                 2    1.059    0.226    4.678    0.000
```

1.4 Running example

- throughout the workshop, we will make use of a simple 'latent mediation' example to illustrate the various estimation approaches
- we generate 'perfect' (population-level) data

```
> library(lavaan)
> pop.model <- '
      # factor loadings
      Y  =~ 1*y1 + 1.2*y2  + 0.8*y3  + 0.5*y4
      M  =~ 1*m1 + 0.5*m2  + 0.5*m3  + 0.7*m4
      X1 =~ 1*x1 + 0.7*x2  + 0.6*x3  + 1.1*x4
      X2 =~ 1*x5 + 0.7*x6  + 0.6*x7  + 0.9*x8
      X3 =~ 1*x9 + 0.7*x10 + 0.6*x11 + 1.1*x12

      # covariances among exogenous X1-X3 and Age
      X1 ~~ 0.4*X2; X1 ~~ -0.2*X3; X2 ~~ 0.4*X3
      Age ~~ 0.6*X1 + 0.7*X2 + 0.2*X3

      # regression part
      Y ~  0.25*X3 + 0.4*M + (-0.1)*Age
      M ~ -0.30*X1 + 1.1*X2
  '
> set.seed(1234)
> Data <- simulateData(pop.model, sample.nobs = 200L, empirical = TRUE)
```

diagram



fitting the model using traditional SEM

```
> model <- '
      # measurement part
      Y  =~ y1 + y2  + y3  + y4
      M  =~ m1 + m2  + m3  + m4
      X1 =~ x1 + x2  + x3  + x4
      X2 =~ x5 + x6  + x7  + x8
      X3 =~ x9 + x10 + x11 + x12

      # structural part
      Y ~ X3 + M + Age
      M ~ X1 + X2
      Age ~~ X1 + X2 + X3
  '
> fit.sem <- sem(model, data = Data, estimator = "ML")
> parameterEstimates(fit.sem, ci = FALSE, output = "text")[21:25,]

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  Y ~
    X3                0.250    0.108    2.307    0.021
    M                 0.400    0.086    4.676    0.000
    Age              -0.100    0.094   -1.060    0.289
  M ~
    X1               -0.300    0.133   -2.262    0.024
    X2                1.100    0.160    6.885    0.000
```

upgrading Age to a latent variable (for convenience only)

```
> model2 <- '
      # measurement part
      Y  =~ y1 + y2  + y3  + y4
      M  =~ m1 + m2  + m3  + m4
      X1 =~ x1 + x2  + x3  + x4
      X2 =~ x5 + x6  + x7  + x8
      X3 =~ x9 + x10 + x11 + x12
      fAge =~ 1*Age; Age ~~ 0*Age

      # structural part
      Y ~ X3 + M + fAge
      M ~ X1 + X2
  '
> fit.sem2 <- sem(model2, data = Data, estimator = "ML")
> parameterEstimates(fit.sem2, ci = FALSE, output = "text")[23:27,]

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  Y ~
    X3                0.250    0.108    2.307    0.021
    M                 0.400    0.086    4.676    0.000
    fAge             -0.100    0.094   -1.060    0.289
  M ~
    X1               -0.300    0.133   -2.262    0.024
    X2                1.100    0.160    6.885    0.000
```

model-implied variance-covariance matrix latent variables

- the standard way (in lavaan) to extract the (co)variance matrix of the latent variables:

```
> lavInspect(fit.sem, "cov.lv")
```

```
        Y      M     X1     X2     X3
Y   1.441
M   0.880  2.036
X1 -0.054  0.140  1.000
X2  0.422  0.980  0.400  1.000
X3  0.430  0.500 -0.200  0.400  1.000
```

- unfortunately, this matrix only contains the latent variables, not the observed covariate 'Age'

- that is why (in fit.sem2) we 'upgraded' Age to a latent variable (fAge); lavaan now thinks it is a latent variable, and includes it in the (co)variance matrix of the latent variables

```
> Var.eta <- lavInspect(fit.sem2, "cov.lv")
> Var.eta
```

```
          Y      M     X1     X2     X3   fAge
Y     1.441
M     0.880  2.036
X1   -0.054  0.140  1.000
X2    0.422  0.980  0.400  1.000
X3    0.430  0.500 -0.200  0.400  1.000
fAge  0.186  0.590  0.600  0.700  0.200  1.000
```

- fact: if we could somehow compute this matrix, then we can obtain all the regression coefficients using two regressions (one for 'Y', and one for 'M')
- recall the well known formula to estimate the regression coefficients in a linear model:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

- an equivalent formula based on the sample covariance matrix ($\mathbf{S}$):

$$\hat{\boldsymbol{\beta}} = (\mathbf{S}_{xx})^{-1}\mathbf{S}_{xy}$$

where $\mathbf{S}_{xx}$ is the part of $\mathbf{S}$ where the rows and columns correspond to the predictors ('x'), while $\mathbf{S}_{xy}$ is the part of $\mathbf{S}$ where the rows correspond to the predictors and the (single) column corresponds to the outcome ('y')

- regression for ‘Y’:

```
> # compute regression coefficients for Y
> beta.Y <- ( solve(Var.eta[c("X3", "M", "fAge"), c("X3", "M", "fAge")]) %*%
                    Var.eta[c("X3", "M", "fAge"), "Y", drop = FALSE] )
> round(beta.Y, 3)
```

```
          Y
X3     0.25
M      0.40
fAge -0.10
```

- regression for ‘M’:

```
> # compute regression coefficients for M
> beta.M <- ( solve(Var.eta[c("X1", "X2"), c("X1", "X2")]) %*%
                    Var.eta[c("X1", "X2"), "M", drop = FALSE] )
> round(beta.M, 3)
```

```
      M
X1 -0.3
X2  1.1
```

2 Structural-after-measurement (SAM) approaches

2.1 Introduction

- SAM is an umbrella term to describe many different (estimation) approaches that have the following in common:
  1. first step: we estimate the parameters related to the measurement part
  2. second step: we estimate the parameters related to the structural part
- SAM is an estimation approach, not a model-building approach; the main goal is to find (good quality) point estimates for the structural (i.e., regression) parameters in the model
- ideally, we also wish to obtain (good quality) standard errors and test statistics for these structural parameters (for statistical inference)
- the term SAM was used by Rosseel & Loh (2024), to avoid the overloaded terms 'two-step', 'two-stage', ... (and was first used during a presentation I gave at WWU Münster, Jan 2020)

2.2 Early SAM approaches

- some early references:
  - Burt (1976)
  - Hunter & Gerbing (1982)
  - Lance, Cornwell & Mulaik (1988)
- SAM approaches never received much attention in the SEM literature
- but SAM is the default approach in many other fields
  - 'measurement error models' in statistics
  - item response theory (IRT) (e.g., plausible values)
  - partial least squares (PLS)
- many applied researchers in psychology, educational sciences, ... replace latent variables by sum scores (or factor scores); this is a SAM approach
- imagine an applied researcher that is in doubt whether to use SEM or not and visits a statistical consultant:

applied researcher talking to a statistical consultant

(researcher:) I was thinking of computing factor scores, and then do a regression in SPSS.

applied researcher talking to a statistical consultant

(researcher:) I was thinking of computing factor scores, and then do a regression in SPSS.

(consultant:) Oh come on! You should use SEM. SEM is the gold standard.

applied researcher talking to a statistical consultant

(researcher:) I was thinking of computing factor scores, and then do a regression in SPSS.

(consultant:) Oh come on! You should use SEM. SEM is the gold standard.

(researcher:) Yes, I know. You have told me many times. But this is a really big model. And I am only interested in the regression part.

applied researcher talking to a statistical consultant

(researcher:) I was thinking of computing factor scores, and then do a regression in SPSS.

(consultant:) Oh come on! You should use SEM. SEM is the gold standard.

(researcher:) Yes, I know. You have told me many times. But this is a really big model. And I am only interested in the regression part.

(consultant:) If you ignore the measurement error, your results will be biased. This is really bad. You should use SEM.

applied researcher talking to a statistical consultant

(researcher:) I was thinking of computing factor scores, and then do a regression in SPSS.

(consultant:) Oh come on! You should use SEM. SEM is the gold standard.

(researcher:) Yes, I know. You have told me many times. But this is a really big model. And I am only interested in the regression part.

(consultant:) If you ignore the measurement error, your results will be biased. This is really bad. You should use SEM.

(researcher:) So using factor scores followed by regression is really a silly idea? I have seen it in many journals.

applied researcher talking to a statistical consultant

(researcher:) I was thinking of computing factor scores, and then do a regression in SPSS.

(consultant:) Oh come on! You should use SEM. SEM is the gold standard.

(researcher:) Yes, I know. You have told me many times. But this is a really big model. And I am only interested in the regression part.

(consultant:) If you ignore the measurement error, your results will be biased. This is really bad. You should use SEM.

(researcher:) So using factor scores followed by regression is really a silly idea? I have seen it in many journals.

(consultant:) Yeah, idiots are everywhere. You may get away with it in some journals, but not in a good journal. You must use SEM.

2.3 Overview of SAM approaches (for SEM)

- replacing the latent variables by ('factor') scores
  - (uncorrected) factor score regression (UFSR)
  - correlation-preserving factor scores
  - sum scores
  - instrumental variables
- computing ('corrected') summary statistics for the latent variables
  - Wall & Amemiya (2000), Croon (2001), Devlieger et al. (2016, 2017)
  - local SAM (Rosseel & Loh, 2024)
- the single-indicator (SI) approach
- global SAM

2.4 Potential advantages of a SAM approach

- reduced model complexity: the 'structural' model is often just a regression model, or a path analysis
- if the model is correctly specified, many SAM approaches (but not all) give identical results as SEM (for the structural part)
- (almost) no convergence issues, increased stability (in particular in small samples)
- less variability (lower MSE) for structural parameters
- in certain cases (but not always): more robust against local (structural) model misspecifications
- no interpretational confounding
- very flexible: the decoupling of 'measurement' and 'structural' allows for tackling much more complicated models (even with modest sample sizes)
- ...

2.5 Critique on the SAM approach

- the (naive) standard errors in the second step are wrong (because they ignore the uncertainty that stems from the first step)
- in general: inference (e.g., standard errors, confidence intervals, test statistics) is (more) complicated
- multiple step methods are less efficient (more sampling variability)
- Fornell and Yi (1992) gave an example where a misspecified (but well-fitting) measurement model was embedded in a correctly specified structural model; but the (global) model fit of the full model suggested that the model did not fit well, thus incorrectly implying a misspecified structural model
- do (latent) constructs 'exist' on their own? or are they only meaningful when embedded in a context (i.e., a structural model)?
- software packages only allow for joint estimation

2.6 Related techniques

- latent class analysis:

  Bakk, Z., & Kuha, J. (2021). Relating latent class membership to external variables: An overview. *British Journal of Mathematical and Statistical Psychology*, 74(2), 340–362.
  https://doi.org/10.1111/bmsp.12227 (Open Access)

- item response theory (IRT):

  Kuha, J., & Bakk, Z. (arxiv.org). Two-step estimation of latent trait models.
  https://arxiv.org/pdf/2303.16101.pdf

- Bayesian SAM:

  Levy, R. (2023). Precluding interpretational confounding in factor analysis with a covariate or outcome via measurement and uncertainty preserving parametric modeling. *Structural Equation Modeling: A Multidisciplinary Journal.*
  https://doi.org/10.1080/10705511.2022.2154214

3 Replacing the latent variables by ('factor') scores

- general idea: we try to compute individual scores for each latent variable that is involved in the structural part of the model
- once we have these scores, we treat them as if they were observed
- we then perform an observed-only procedure (e.g., regression or path-analysis) using these scores (and perhaps other observed covariates)
- there are various ways to compute these scores:
  - we compute so-called 'factor scores' (e.g., Bartlett or regression factor scores) for all the latent variables
  - we compute the sum or the mean of the latent variable indicators
  - we just take the observed scores of the first ('marker') indicator of the latent variable

3.1 Factor scores

- the standard factor model:

$$\boldsymbol{y} = \boldsymbol{\Lambda}\boldsymbol{\eta} + \boldsymbol{\epsilon}$$

  - $\boldsymbol{y}$ and $\boldsymbol{\epsilon}$ are P-dimensional continuous random vectors
  - $\boldsymbol{\Lambda}$ is a $P \times M$ matrix of factor loadings
  - $\boldsymbol{\eta}$ is a M-dimensional random vector of latent variables
  - we denote $\text{Var}(\boldsymbol{\epsilon}) = \boldsymbol{\Theta}$

- we can use CFA to estimate all the model parameters

- here, we only consider linear factor scores of the form

$$\mathbf{f} = \hat{\boldsymbol{\eta}} = \mathbf{A}\,\boldsymbol{y}$$

- different types of factor scores can be constructed by choosing a different way to compute the $M \times P$ 'factor score matrix' $\mathbf{A}$

- three important types: 1) 'regression' factor scores, 2) Bartlett factor scores, and 3) correlation-preserving factor scores

regression factor scores (Thomson, 1934)

- Thomson regarded the quest for factor scores as a (multivariate) regression problem: we wish to 'predict' $\boldsymbol{\eta}$ as a (linear) function of the observed variables $\boldsymbol{y}$:

$$\boldsymbol{\eta} = \mathbf{A}\boldsymbol{y} + \mathbf{r}$$

- it is well known that the OLS solution is given by

$$\mathbf{A} = \text{Cov}(\boldsymbol{\eta}, \boldsymbol{y})\text{Var}(\boldsymbol{y})^{-1}$$

- because $\text{Var}(\boldsymbol{y}) = \boldsymbol{\Sigma}$ and $\text{Cov}(\boldsymbol{\eta}, \boldsymbol{y}) = \text{Var}(\boldsymbol{\eta})\boldsymbol{\Lambda}'$, this results in the following well known expression for $\mathbf{A}$:

$$\mathbf{A}_{reg} = \text{Var}(\boldsymbol{\eta})\boldsymbol{\Lambda}'\boldsymbol{\Sigma}^{-1}$$

- an equivalent expression is

$$\mathbf{A}_{reg} = (\text{Var}(\boldsymbol{\eta})^{-1} + \boldsymbol{\Lambda}'\boldsymbol{\Theta}^{-1}\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}'\boldsymbol{\Theta}^{-1}$$

- properties:
  - regression factor scores are 'best' in the sense that they minimize the MSE
  - but they are not (conditionally) unbiased: $\mathrm{E}(\mathbf{f}|\boldsymbol{\eta}) \neq \boldsymbol{\eta}$
  - $\mathrm{Var}(\mathbf{f})$ is *not* equal to $\mathrm{Var}(\boldsymbol{\eta})$ (i.e., not correlation-preserving)
- computing regression factor scores in lavaan:

```
> fs.reg <- lavPredict(fit.sem)
> head(fs.reg)
```

```
                Y          M          X1          X2          X3
[1,] -0.028224721 -0.8846010 -0.17290284 -0.9824259  0.08455516
[2,] -1.240496204 -1.3514714 -0.74127499 -0.7529585 -0.96667911
[3,] -0.971195982 -0.5307034  0.60926672 -0.2105924 -0.49018971
[4,]  1.595272994  1.3306954 -0.65291235  0.7840847  2.05366756
[5,] -0.700829697 -1.3379027  0.03224246 -0.9103079 -1.03610866
[6,] -0.006359753  0.1922130  1.11774632  0.5307825 -0.69236171
```

- very popular; default in many software packages

Bartlett (1937, 1938) factor scores

- Bartlett used the principle of maximum likelihood to find 'good' factor scores
- given $\boldsymbol{y}$, $\boldsymbol{\Lambda}$ and $\boldsymbol{\Theta}$, we need to find $\boldsymbol{\eta}$ that maximizes the following function:

$$f(\boldsymbol{\eta}) = \text{const} - \frac{1}{2}(\boldsymbol{y} - \boldsymbol{\Lambda\eta})^T \boldsymbol{\Theta}^{-1}(\boldsymbol{y} - \boldsymbol{\Lambda\eta})$$

- equivalently, we can minimize the weighted least squares (WLS) criterion

$$f(\boldsymbol{\eta}) = (\boldsymbol{y} - \boldsymbol{\Lambda\eta})^T \boldsymbol{\Theta}^{-1}(\boldsymbol{y} - \boldsymbol{\Lambda\eta})$$

- it is well known that the WLS solution for $\boldsymbol{\eta}$ is given by

$$\hat{\boldsymbol{\eta}} = \mathbf{f} = (\boldsymbol{\Lambda}^T \boldsymbol{\Theta}^{-1} \boldsymbol{\Lambda})^{-1} \boldsymbol{\Lambda}^T \boldsymbol{\Theta}^{-1} \boldsymbol{y}$$

- therefore, the factor score matrix used for Bartlett factor scores is given by

$$\mathbf{A}_{bar} = (\boldsymbol{\Lambda}^T \boldsymbol{\Theta}^{-1} \boldsymbol{\Lambda})^{-1} \boldsymbol{\Lambda}^T \boldsymbol{\Theta}^{-1}$$

- properties:
  - the MSE is (slightly) larger than the MSE for regression factor scores
  - Bartlett factor scores are conditionally unbiased, that is $\mathrm{E}(\mathbf{f}|\boldsymbol{\eta}) = \boldsymbol{\eta}$
  - $\mathrm{Var}(\mathbf{f})$ is *not* equal to $\mathrm{Var}(\boldsymbol{\eta})$ (i.e., not correlation-preserving)
- computing Bartlett factor scores in lavaan:

```
> fs.bar <- lavPredict(fit.sem, method = "Bartlett")
> head(fs.bar)
```

```
                Y          M         X1         X2         X3
[1,]  0.15144396 -0.5282003  0.3864508 -1.2292747  0.4506464
[2,] -1.26183316 -1.1792206 -0.7200066  0.1304861 -1.1825560
[3,] -1.18282373 -0.5415109  0.7580129 -0.4356033 -0.5191142
[4,]  1.71834988  1.1929370 -0.8850225  0.5464266  2.5861810
[5,] -0.59019076 -1.1978196  0.3701585 -0.6595145 -1.1726745
[6,] -0.08347654 -0.2613428  0.9606375  0.2891287 -1.0844284
```

correlation-preserving factor scores

- correlation-preserving factor scores have the property that $\text{Var}(\mathbf{f}) = \text{Var}(\boldsymbol{\eta})$
- a better name would have been 'covariance-preserving' or 'structure-preserving' factor scores
- first introduced by Anderson & Rubin (1956) for orthogonal factors
- general case: Green (1969), McDonald (1981), Krijnen, Wansbeek and ten Berge (1996)
- the 'regression' version (Green 1969):

$$\mathbf{A}_{green} = \text{Var}(\boldsymbol{\eta})^{1/2} \left(\text{Var}(\boldsymbol{\eta})^{3/2} \boldsymbol{\Lambda}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Lambda} \text{Var}(\boldsymbol{\eta})^{3/2}\right)^{-1/2} \text{Var}(\boldsymbol{\eta})^{3/2} \boldsymbol{\Lambda}' \boldsymbol{\Sigma}^{-1/2} \boldsymbol{\Sigma}^{-1/2}$$

- the 'Bartlett' version (Krijnen's determinant best version):

$$\mathbf{A}_{det} = \text{Var}(\boldsymbol{\eta})^{1/2} \left(\text{Var}(\boldsymbol{\eta})^{1/2} \boldsymbol{\Lambda}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Lambda} \text{Var}(\boldsymbol{\eta})^{1/2}\right)^{-1/2} \text{Var}(\boldsymbol{\eta})^{1/2} \boldsymbol{\Lambda}' \boldsymbol{\Sigma}^{-1/2} \boldsymbol{\Sigma}^{-1/2}$$

- it turns out that we can also compute them by 'transforming' the regression or Bartlett factor scores

- properties:
  - MSE is the largest of the three types
  - not conditionally unbiased; $\mathrm{E}(\mathbf{f}|\boldsymbol{\eta}) \neq \boldsymbol{\eta}$
  - but of course correlation-preserving: $\mathrm{Var}(\mathbf{f}) = \mathrm{Var}(\boldsymbol{\eta})$
- computing correlation-preserving (Bartlett) factor scores in lavaan:

```
> fs.cpbar <- lavPredict(fit.sem, method = "Bartlett", transform = TRUE)
> head(fs.cpbar)
```

```
              Y          M          X1         X2         X3
[1,]  0.08678644 -0.6752758  0.06857263 -1.0310056  0.2480118
[2,] -1.22439381 -1.1956105 -0.71204975 -0.3687312 -1.0373245
[3,] -1.08359990 -0.5730704  0.64651277 -0.3319827 -0.5211020
[4,]  1.65398192  1.2619023 -0.74443267  0.7049411  2.2981337
[5,] -0.63420336 -1.2348187  0.18017539 -0.7944636 -1.0937451
[6,] -0.08924933 -0.1018241  1.02362855  0.3531652 -0.8931581
```

- (unfortunately) almost never used in practice

factor indeterminacy

- there is not a single way to compute factor scores; there are many ways, some of them have 'good' properties
- this is called 'factor indeterminacy'
- in particular when the interest is in individual factor scores, this is problematic (different factor scores may result in a different ordering of the individuals)
- but we will only be concerned with the 'relation' between factor scores
- therefore, all we care about is the variance-covariance matrix of the factor scores $\text{Var}(\mathbf{f})$
- the fact that $\text{Var}(\mathbf{f}) \neq \text{Var}(\boldsymbol{\eta})$ for regression and Bartlett factor scores has important consequences
- reference: Steiger (1996)

3.2 (Uncorrected) factor score regression (UFSR)

- the idea of factor score regression is simple: we replace all latent variables by factor scores
- then, in a follow up analysis, we treat these factor scores as if they were observed
- traditionally, only regression (or Bartlett) factor scores have been used in FSR
- but this is problematic, because for these factor scores, we have that $\text{Var}(\mathbf{f}) \neq \text{Var}(\boldsymbol{\eta})$
- as a result, uncorrected FSR is not 'consistent': the regression coefficients that we obtain in the follow-up analysis will not match the population values, even if the model is correct and the sample size goes to infinity
- this has been well documented in the (SEM) literature, and has contributed to the widely spread opinion that FSR is not a good practice

example uncorrected FSR with regression factor scores

- first step: compute factor scores (and add observed covariates)

```
> # compute regression factor scores
> FS.reg <- as.data.frame(lavPredict(fit.sem2, method = "regression"))
```

- regression for 'Y':

```
> fit.y <- lm(Y ~ X3 + M + fAge, data = FS.reg)
> round(coef(fit.y), 3)

(Intercept)          X3           M        fAge
      0.000       0.275       0.473      -0.148
```

- regression for 'M':

```
> fit.m <- lm(M ~ X1 + X2, data = FS.reg)
> round(coef(fit.m), 3)

(Intercept)          X1          X2
      0.000      -0.450       1.327
```

- not the same as the population values

example uncorrected FSR with Bartlett factor scores

- first step: compute factor scores (and add observed covariates)

```
> # compute regression factor scores
> FS.bar <- as.data.frame(lavPredict(fit.sem2, method = "Bartlett"))
```

- regression for 'Y':

```
> fit.y <- lm(Y ~ X3 + M + fAge, data = FS.bar)
> round(coef(fit.y), 3)

(Intercept)          X3           M        fAge
      0.000       0.212       0.315      -0.042
```

- regression for 'M':

```
> fit.m <- lm(M ~ X1 + X2, data = FS.bar)
> round(coef(fit.m), 3)

(Intercept)          X1          X2
      0.000      -0.120       0.747
```

- not the same as the population values

the fundamental problem with FSR

- for regression and Bartlett regression scores, we have $\text{Var}(\mathbf{f}) \neq \text{Var}(\boldsymbol{\eta})$; as a result, all regressions based on $\text{Var}(\mathbf{f})$ will result in bias
- this is true even if the sample size goes to infinity
- recall that the 'correct' $\text{Var}(\boldsymbol{\eta})$ equals

```
> round(Var.eta, 3)

          Y      M     X1     X2     X3   fAge
Y     1.441
M     0.880  2.036
X1   -0.054  0.140  1.000
X2    0.422  0.980  0.400  1.000
X3    0.430  0.500 -0.200  0.400  1.000
fAge  0.186  0.590  0.600  0.700  0.200  1.000
```

- $\text{Var}(\mathbf{f})$ for regression factor scores:

```
> N <- nrow(FS.reg)
> FS.reg.cov <- cov(FS.reg) * (N - 1) / N
> round(FS.reg.cov, 3)
```

```
          Y     M     X1    X2     X3  fAge
Y     1.206 0.848 -0.051 0.415  0.416 0.186
M     0.848 1.692  0.154 0.920  0.494 0.590
X1   -0.051 0.154  0.796 0.385 -0.171 0.600
X2    0.415 0.920  0.385 0.823  0.375 0.700
X3    0.416 0.494 -0.171 0.375  0.773 0.200
fAge  0.186 0.590  0.600 0.700  0.200 1.000
```

- Var(**f**) for Bartlett factor scores:

```
> FS.bar.cov <- cov(FS.bar) * (N - 1) / N
> round(FS.bar.cov, 3)
```

```
          Y     M     X1    X2     X3  fAge
Y     1.741 0.880 -0.054 0.422  0.430 0.186
M     0.880 2.539  0.140 0.980  0.500 0.590
X1   -0.054 0.140  1.327 0.400 -0.200 0.600
X2    0.422 0.980  0.400 1.376  0.400 0.700
X3    0.430 0.500 -0.200 0.400  1.327 0.200
fAge  0.186 0.590  0.600 0.700  0.200 1.000
```

- note: the Bartlett based variance-covariance matrix is very close, except for the variances of the latent variables

3.3 Skrondal & Laake (2001)

- Skrondal & Laake (2001) noticed a remarkable result: if we use (blockwise) 'Bartlett' factor scores for the 'dependent' variable(s), and (blockwise) regression factor scores for the 'independent' variable(s), the resulting regression coefficients are consistent again

- the regression for 'Y':

```
> model.y.indep <- '
    X3 =~ x9 + x10 + x11 + x12
    M =~ m1 + m2 + m3 + m4
    fAge =~ Age; Age ~~ 0*Age
  '
> fit.y.indep <- cfa(model.y.indep, data = Data)
> SL.y.indep <- lavPredict(fit.y.indep, method = "regression")
> model.y.dep <- ' Y =~ y1 + y2 + y3 + y4 '
> fit.y.dep <- cfa(model.y.dep, data = Data)
> SL.y.dep <- lavPredict(fit.y.dep, method = "Bartlett")
> FS.y <- as.data.frame(cbind(SL.y.dep, SL.y.indep))
> fit.y <- lm(Y ~ X3 + M + fAge, data = FS.y)
> round(coef(fit.y), 3)

(Intercept)          X3           M        fAge
      0.00        0.25        0.40       -0.10
```

- the regression for 'M':

```
> model.m.indep <- '
    X1 =~ x1 + x2 + x3 + x4
    X2 =~ x5 + x6 + x7 + x8
  '
> fit.m.indep <- cfa(model.m.indep, data = Data)
> SL.m.indep <- lavPredict(fit.m.indep, method = "regression")
> model.m.dep <- ' M =~ m1 + m2 + m3 + m4 '
> fit.m.dep <- cfa(model.m.dep, data = Data)
> SL.m.dep <- lavPredict(fit.m.dep, method = "Bartlett")
> FS.m <- as.data.frame(cbind(SL.m.dep, SL.m.indep))
> fit.m <- lm(M ~ X1 + X2, data = FS.m)
> round(coef(fit.m), 3)

(Intercept)          X1          X2
        0.0        -0.3         1.1
```

- this works for all recursive models, but a separate analysis is needed for each endogenous latent variable

- but only for the unstandardized solution; the 'standardized' solution is still biased, because we have no (consistent) estimate for the total variance of the outcome variable (recall: $\beta^z = \frac{s_x}{s_y}\beta$)

3.4 Correlation-preserving factor scores

- what if we use correlation-preserving factor scores (instead of regression or Bartlett factor scores)?
- the regression for 'Y':

```
> fs.cpreg <- lavPredict(fit.sem2, method = "regression", transform = TRUE)
> FS.cpreg <- as.data.frame(fs.cpreg)
> fit.y <- lm(Y ~ X3 + M + fAge, data = FS.cpreg)
> round(coef(fit.y), 3)

(Intercept)          X3           M        fAge
      0.00        0.25        0.40       -0.10
```

- the regression for 'M':

```
> fit.m <- lm(M ~ X1 + X2, data = FS.cpreg)
> round(coef(fit.m), 3)

(Intercept)          X1          X2
        0.0        -0.3         1.1
```

- it works equally well with the 'Bartlett' version

- that we obtain the correct results is of course due to the fact that for correlation-preserving factor scores, we have that $\text{Var}(\mathbf{f}) = \text{Var}(\boldsymbol{\eta})$
- it is remarkable that these correlation-preserving factor scores are almost never used in the applied literature
- in fact, it turns out that correlation-preserving factor scores work really well, as shown in an extensive simulation study reported in Chapter 4 of the PhD thesis of Jasper Bogaert (a paper is under preparation)
- they are a good alternative for other (consistent) approaches that we will discuss in the next section
- but (as with all FSR methods), it is not easy to obtain 'corrected' standard errors (for the regression coefficients) for valid inference

3.5 Sum scores

- sum scores are a special type of factor scores
- they (implicitly) assume that:
  1. all factor loadings are the same (say, equal to 1.0)
  2. all residual variances of the indicators are equal to each other
- if this is approximately true, they can easily replace factor scores
- as an additional benefit, no CFA model needs to be fitted in order to obtain the sum scores
- there is a renewed interested (and respect) in the recent SEM literature
- in the example below, we will use mean scores (instead of sum scores), so that the scale of the regression coefficients is more in line with the population values

```
> Ymean <- rowSums(Data[, c("y1", "y2", "y3" ,"y4")])/4
> Mmean <- rowSums(Data[, c("m1", "m2", "m3" ,"m4")])/4
> X1mean <- rowSums(Data[, c("x1", "x2", "x3" ,"x4")])/4
> X2mean <- rowSums(Data[, c("x5", "x6", "x7" ,"x8")])/4
> X3mean <- rowSums(Data[, c("x9", "x10", "x11" ,"x12")])/4
> Age <- Data$Age
```

- regression for 'Y':

```
> fit.y <- lm(Ymean ~ X3mean + Mmean + Age)
> round(coef(fit.y), 3)
```

```
(Intercept)      X3mean       Mmean         Age
      0.000       0.217       0.400      -0.033
```

- regression for 'M':

```
> fit.m <- lm(Mmean ~ X1mean + X2mean)
> round(coef(fit.m), 3)
```

```
(Intercept)      X1mean      X2mean
      0.000      -0.092       0.623
```

sum scores (for Y) via CFA

```
> model.ysum <- '
      Y  =~ 1*y1 + 1*y2  + 1*y3  + 1*y4
      # equal residual variances per factor
      y1 ~~ vy*y1; y2 ~~ vy*y2; y3 ~~ vy*y3; y4 ~~ vy*y4
  '
> fit.ysum <- sem(model.ysum, data = Data, estimator = "ML")
> ysum.reg <- lavPredict(fit.ysum)[,1]
> cor(ysum.reg, Ymean)

[1] 1


> ysum.bar <- lavPredict(fit.ysum, method = "Bartlett")[,1]
> cor(ysum.bar, Ymean)

[1] 1


> head(cbind(ysum.reg, Ymean)) # regression: different scale

       ysum.reg      Ymean
[1,]  0.2289396  0.2892364
[2,] -0.9078576 -1.1469642
[3,] -0.8036490 -1.0153097
[4,]  1.0639416  1.3441567
[5,] -0.2207324 -0.2788677
[6,]  0.1059182  0.1338144
```

```
> head(cbind(ysum.bar, Ymean)) # Bartlett: same scale

       ysum.bar      Ymean
[1,]  0.2892364  0.2892364
[2,] -1.1469642 -1.1469642
[3,] -1.0153097 -1.0153097
[4,]  1.3441567  1.3441567
[5,] -0.2788677 -0.2788677
[6,]  0.1338144  0.1338144
```

Reference:

Sijtsma K., Ellis J.L., Borsboom D. (2024). Recognize the Value of the Sum Score, Psychometrics' Greatest Accomplishment. *Psychometrika, 89*(1), 84–117
https://doi.org/10.1007/s11336-024-09964-7

3.6 Instrumental variables

- Bollen (1996) suggested to use the method of 'instrumental variables' (IVs) in combination with two-stage least-squares (2SLS) to estimate the 'regression' coefficients in all the 'equations' in a SEM, one at a time
- this includes all the structural relations, but (optionally) also the factor loadings in the measurement blocks
- IV-2SLS is a commonly used estimation technique in econometrics
- Mandansky (1964) proposed IV-2SLS estimators for factor analysis models
- Hägglund (1982) and Jöreskog (1983) developed this further
- lavaan includes the FABIN2 and FABIN3 estimators for CFAs
- here, we will discuss only the estimation of the regression coefficients in the structural part, as described in Bollen (1996)

the IV-2SLS procedure

- key idea: replace each latent variable by its marker indicator
- then, everything is observed, and we can apply regression methods
- but because the error term is no longer independent from the predictors, we cannot use OLS
- but we can use 2SLS: first we regress the predictor ('x') indicators on their instruments; then we regress the outcome ('y') indicator on the predicted values of the predictor indicators
- the problem is to find 'good' instruments: they must be correlated with the predictor indicators, but uncorrelated with the error term of the equation
- Bollen showed that we can use the other indicators of the predictors (except those that are involved in the equation) as instruments; as this is determined by the model, this is referred to as 'model implied instrumental variables' or MIIVs
- this is implemented in the R package 'MIIVsem'

R code for the IV-2SLS procedure

```
> # regress 'x' variables on IVs (for Y)
> fit.IV <- lm(cbind(x9, m1, Age) ~ x1 + x2 + x3 + x4 +  # X1
                                    x5 + x6 + x7 + x8 +  # X2
                                    x10 + x11 + x12 +    # X3
                                    m2 + m3 + m4 + Age, data = Data) # M and Age
> # regress 'y' on predicted values 'x'
> fit.y <- lm(Data$y1 ~ predict(fit.IV))
> round(coef(fit.y), 3)

      (Intercept)  predict(fit.IV)x9  predict(fit.IV)m1 predict(fit.IV)Age
             0.00               0.25               0.40              -0.10



> # regress 'x' variables on IVs (for M)
> fit.IV <- lm(cbind(x1, x5) ~ x2 + x3 + x4 + # X1
                               x6 + x7 + x8 + # X2
                               x10 + x11 + x12 + Age, data = Data) # X3 and Age
> # regress 'y' on predicted values 'x'
> fit.m <- lm(Data$m1 ~ predict(fit.IV))
> round(coef(fit.m), 3)

     (Intercept) predict(fit.IV)x1 predict(fit.IV)x5
             0.0              -0.3               1.1
```

3.7 Two-step standard errors

- so far, we only discussed point estimation
- some procedures are consistent, some are not
- but what about standard errors?
- the 'naive' standard errors that we obtain in the regression step in FSR are (usually) not correct
- they fail to take the uncertainty of the first step into account
- analytic standard errors are (currently) only available for the IV-2SLS approach (see Bollen 1996)
- we can always use the bootstrap to obtain 'correct' standard errors for the final regression coefficients (but this can be time consuming)
- analytic standard errors for FSR may also be possible with the 'local' method as used in local SAM (work in progress)

4 Computing summary statistics for latent variables

4.1 Introduction

- if our goal is (only) to compute the regression coefficients of the structural relations, then we do not need the individual factor scores
- we only need the variance–covariance matrix $\text{Var}(\mathbf{f})$ (and perhaps the mean vector)
- let $\mathbf{S} = \text{Var}(\boldsymbol{y})$ be the sample covariance matrix of the observed data
- let $\mathbf{A}$ be the factor score matrix
- it is easy to show that:

$$\text{Var}(\mathbf{f}) = \text{Var}(\mathbf{A}\,\mathbf{y}) = \mathbf{A}\,\text{Var}(\mathbf{y})\,\mathbf{A}^T = \mathbf{A}\,\mathbf{S}\,\mathbf{A}^T$$

- with the exception of correlation-preserving factor scores, the main reason why uncorrected FSR is not consistent is because $\text{Var}(\mathbf{f}) \neq \text{Var}(\boldsymbol{\eta})$

from Var(f) to Var(η)

- several authors have suggested to 'correct' Var($\mathbf{f}$) so that it coincides with Var($\boldsymbol{\eta}$)
  - Dijkstra (1981), Dijkstra & Henseler (2015): consistent PLS
  - Fuller (1987): measurement error models
  - Wall & Amemiya (2000, 2003): 2 stage method of moments (2SMM) for polynomial SEM
  - Croon (2002): '(Croon's) corrections' for latent variable models
  - Rosseel & Loh (2024): local SAM
- instead of focusing on individual factor scores, these methods try to compute a consistent estimate for Var($\boldsymbol{\eta}$) (and perhaps E($\boldsymbol{\eta}$) if needed)
- once we have these 'summary statistics', we can proceed in a second step with a regression (or path analysis) and obtain consistent results for the structural relations (just like SEM)
- in addition, many of them provide 'two-step corrected' standard errors

4.2 The method of Croon (2001)

- Croon (2001) considered a very general framework, including discrete latent variables, and discrete observed indicators
- here, we only consider his approach for the continuous case (continuous latent variables, continuous observed indicators)
- Croon's corrections were originally expressed in scalar form; here, we will present them in a more general form (although for simplicity, we will ignore the meanstructure)
- we assume that we fitted all the measurement blocks, either one by one, in subsets, or all together in one joint CFA
- therefore, we have estimates for $\mathbf{\Lambda}$ and $\mathbf{\Theta}$
- we also compute the factor score matrix $\mathbf{A}$ (either the regression or the Bartlett version)
- furthermore, we define the matrices $\mathbf{D} = \mathbf{A}\mathbf{\Lambda}$ and $\mathbf{E} = \mathbf{A}\,\mathbf{\Theta}\,\mathbf{A}^T$

- Croon's corrections can then be expressed as follows:

$$\begin{aligned}
\text{Var}(\boldsymbol{\eta}) &= \mathbf{D}^{-1}\left[\text{Var}(\mathbf{f}) - \mathbf{E}\right]\mathbf{D}^{-T} \\
&= \mathbf{D}^{-1}\left[\mathbf{A}\,\mathbf{S}\,\mathbf{A}^{T} - \mathbf{E}\right]\mathbf{D}^{-T} \\
&= \mathbf{D}^{-1}\left[\mathbf{A}\,\mathbf{S}\,\mathbf{A}^{T} - \mathbf{A}\,\boldsymbol{\Theta}\,\mathbf{A}^{T}\right]\mathbf{D}^{-T} \\
&= \mathbf{D}^{-1}\,\mathbf{A}\,\left[\mathbf{S} - \boldsymbol{\Theta}\right]\,\mathbf{A}^{T}\,\mathbf{D}^{-T} \\
&= (\mathbf{A}\boldsymbol{\Lambda})^{-1}\mathbf{A}\,\left[\mathbf{S} - \boldsymbol{\Theta}\right]\,\mathbf{A}^{T}\,(\mathbf{A}\boldsymbol{\Lambda})^{-T}
\end{aligned}$$

- if we use Bartlett factor scores, we have that $\mathbf{D} = \mathbf{A}\boldsymbol{\Lambda} = \mathbf{I}$ and the last expression simplifies to

$$\text{Var}(\boldsymbol{\eta}) = \mathbf{A}_{bar}\,\left[\mathbf{S} - \boldsymbol{\Theta}\right]\,\mathbf{A}_{bar}^{T}$$

- several papers showed that the 'Method of Croon' worked remarkably well, and sometimes even better than SEM
- we will not provide an example here, because the 'Method of Croon' turns out to be a special case of local SAM

some references on the method of Croon

Devlieger, I., Mayer, A., & Rosseel, Y. (2016). Hypothesis testing using factor score regression: A comparison of four methods. *Educational and Psychological Measurement, 76*, 741–770.

Devlieger, I., & Rosseel, Y. (2017). Factor Score Path Analysis. *Methodology, 13*, 31–38.

Takane, Y., & Hwang, H. (2017). Comparisons among several consistent estimators of structural equation models. *Behaviormetrika, 45*, 157–188.

Kelcey, B. (2019). A robust alternative estimator for small to moderate sample SEM: Bias-corrected factor score path analysis. *Addictive Behaviors, 94*, 83–98.

Bogaert, J., Loh, W.W., & Rosseel, Y. (2023). A small sample correction for factor score regression. *Educational and Psychological Measurement, 83*(3), 495–519.

4.3 Local SAM

- the measurement model:

$$\boldsymbol{y} = \boldsymbol{\nu} + \boldsymbol{\Lambda}\boldsymbol{\eta} + \boldsymbol{\epsilon}$$

- to solve this for $\boldsymbol{\eta}$, we proceed as follows:

$$\begin{aligned} \boldsymbol{\nu} + \boldsymbol{\Lambda}\,\boldsymbol{\eta} + \boldsymbol{\epsilon} &= \boldsymbol{y} \\ \boldsymbol{\Lambda}\,\boldsymbol{\eta} &= \boldsymbol{y} - \boldsymbol{\nu} - \boldsymbol{\epsilon} \\ \mathbf{M}\boldsymbol{\Lambda}\,\boldsymbol{\eta} &= \mathbf{M}\left[\boldsymbol{y} - \boldsymbol{\nu} - \boldsymbol{\epsilon}\right] \\ \boldsymbol{\eta} &= \mathbf{M}\left[\boldsymbol{y} - \boldsymbol{\nu} - \boldsymbol{\epsilon}\right] \end{aligned}$$

  where $\mathbf{M}$ is $M \times P$ mapping matrix such that $\mathbf{M}\boldsymbol{\Lambda} = \mathbf{I}_M$

- we assume $\mathrm{E}(\boldsymbol{\epsilon}) = \mathbf{0}$ and write $\mathrm{Var}(\boldsymbol{\epsilon}) = \boldsymbol{\Theta}$; it follows that

$$\begin{aligned} \mathrm{E}(\boldsymbol{\eta}) &= \mathbf{M}\left[\mathrm{E}(\boldsymbol{y}) - \boldsymbol{\nu}\right] \\ \mathrm{Var}(\boldsymbol{\eta}) &= \mathbf{M}\left[\mathrm{Var}(\boldsymbol{y}) - \boldsymbol{\Theta}\right]\mathbf{M}^T \end{aligned}$$

local SAM: first stage

- first stage: estimation of the measurement part of the model (only)
- M is the number of latent variables; B is the number of measurement 'blocks'
- three options:
  1. $B = 1$: single CFA
  2. $B = M$: as many 'blocks' as we have latent variables
  3. $B < M$: if some blocks are 'linked' together
- we recommend $B = M$ whenever possible (except if the reliabilites are very low, and the number of indicators is small)
- measurement models that are 'linked' (due to cross-loadings, correlated residuals, or equality constraints) should be fitted together, leading to $B < M$
- for each block, we can use ML, GLS, ..., or we can we use noniterative estimators

local SAM: creating the mapping matrix M

- recall, the mapping matrix must chosen such that $\mathbf{M}\boldsymbol{\Lambda} = \mathbf{I}_M$
- three possible solutions for the mapping matrix $\mathbf{M}$:

$$\mathbf{M} = (\boldsymbol{\Lambda}^T \boldsymbol{\Theta}^{-1} \boldsymbol{\Lambda})^{-1} \boldsymbol{\Lambda}^T \boldsymbol{\Theta}^{-1} \qquad (ML)$$

$$\mathbf{M} = (\boldsymbol{\Lambda}^T \mathbf{S}^{-1} \boldsymbol{\Lambda})^{-1} \boldsymbol{\Lambda}^T \mathbf{S}^{-1} \qquad (GLS)$$

$$\mathbf{M} = (\boldsymbol{\Lambda}^T \boldsymbol{\Lambda})^{-1} \boldsymbol{\Lambda}^T \qquad (ULS)$$

- we then estimate $\mathrm{E}(\boldsymbol{\eta})$ and $\mathrm{Var}(\boldsymbol{\eta})$ as follows:

$$\widehat{\mathrm{E}(\boldsymbol{\eta})} = \hat{\mathbf{M}} \left[\bar{\boldsymbol{y}} - \hat{\boldsymbol{\nu}}\right]$$

$$\widehat{\mathrm{Var}(\boldsymbol{\eta})} = \hat{\mathbf{M}} \left[\mathbf{S} - \hat{\boldsymbol{\Theta}}\right] \hat{\mathbf{M}}^T$$

- $\mathbf{M}$ is called a mapping matrix because it 'maps' the (centered and error-corrected) observed data to the latent space

local SAM: second stage

- second stage: $\widehat{\mathrm{E}(\boldsymbol{\eta})}$ and $\widehat{\mathrm{Var}(\boldsymbol{\eta})}$ are used to estimate the parameters in the structural part of the model
- this can be done using 'path analysis', where we treat everything as observed, and the data is presented via summary statistics
- we can use ML, GLS, ...
- or we can use noniterative estimators: OLS (if the model is recursive) or TSLS (if the model is not recursive)

local SAM: further comments

- two-step corrected standard errors are available (see Appendix C in the SAM paper); recently, we added 'local' standard errors
- local fit measures only (for each measurement block, for the structural part)
- the (co)variance matrix of the latent variables is always positive definite
- we can handle missing data (fiml or two-stage), categorical indicators, two-level data (random intercepts only), latent interactions, higher-order measurement models, ...
- but still some limitations (we plan to address these limitations in future work):
  - the factor loading matrix ($\boldsymbol{\Lambda}$) must have full column rank
  - no support for (e.g., variance components) models where zeroes in the variance-covariance matrix of the latent variables are needed in order to identify the model

example: fit measurement blocks, using $B = M$

```
> fit.Y  <- sem('Y  =~ y1 + y2  + y3  + y4',  data = Data)
> fit.M  <- sem('M  =~ m1 + m2  + m3  + m4',  data = Data)
> fit.X1 <- sem('X1 =~ x1 + x2  + x3  + x4',  data = Data)
> fit.X2 <- sem('X2 =~ x5 + x6  + x7  + x8',  data = Data)
> fit.X3 <- sem('X3 =~ x9 + x10 + x11 + x12', data = Data)

> # assemble Lambda and Theta
> Lambda <- matrix(0, 20, 5)
> Lambda[ 1:4,  1] <- lavInspect(fit.Y,  "est")$lambda
> Lambda[ 5:8,  2] <- lavInspect(fit.M,  "est")$lambda
> Lambda[ 9:12, 3] <- lavInspect(fit.X1, "est")$lambda
> Lambda[13:16, 4] <- lavInspect(fit.X2, "est")$lambda
> Lambda[17:20, 5] <- lavInspect(fit.X3, "est")$lambda

> Theta <- lav_matrix_bdiag(lavInspect(fit.Y,  "est")$theta,
                            lavInspect(fit.M,  "est")$theta,
                            lavInspect(fit.X1, "est")$theta,
                            lavInspect(fit.X2, "est")$theta,
                            lavInspect(fit.X3, "est")$theta)
```

example: compute ML version of the mapping matrix M

```
> Theta.inv <- solve(Theta)
> M <- solve(t(Lambda) %*% Theta.inv %*% Lambda) %*% t(Lambda) %*% Theta.inv

> # add age
> M     <- lav_matrix_bdiag(M,     matrix(1, nrow = 1L, ncol = 1L))
> Theta <- lav_matrix_bdiag(Theta, matrix(0, nrow = 1L, ncol = 1L))
> rownames(M) <- c("Y", "M", "X1", "X2", "X3", "Age")

> # compute (biased) sample covariance matrix 'S'
> N <- nrow(Data)
> S <- cov(Data) * (N - 1L)/N

> # compute Var(Eta)
> Var.eta <- M %*% (S - Theta) %*% t(M)
> round(Var.eta, 3)

          Y     M     X1    X2    X3   Age
Y     1.441 0.880 -0.054 0.422  0.43 0.186
M     0.880 2.036  0.140 0.980  0.50 0.590
X1   -0.054 0.140  1.000 0.400 -0.20 0.600
X2    0.422 0.980  0.400 1.000  0.40 0.700
X3    0.430 0.500 -0.200 0.400  1.00 0.200
Age   0.186 0.590  0.600 0.700  0.20 1.000
```

example: second stage – using OLS

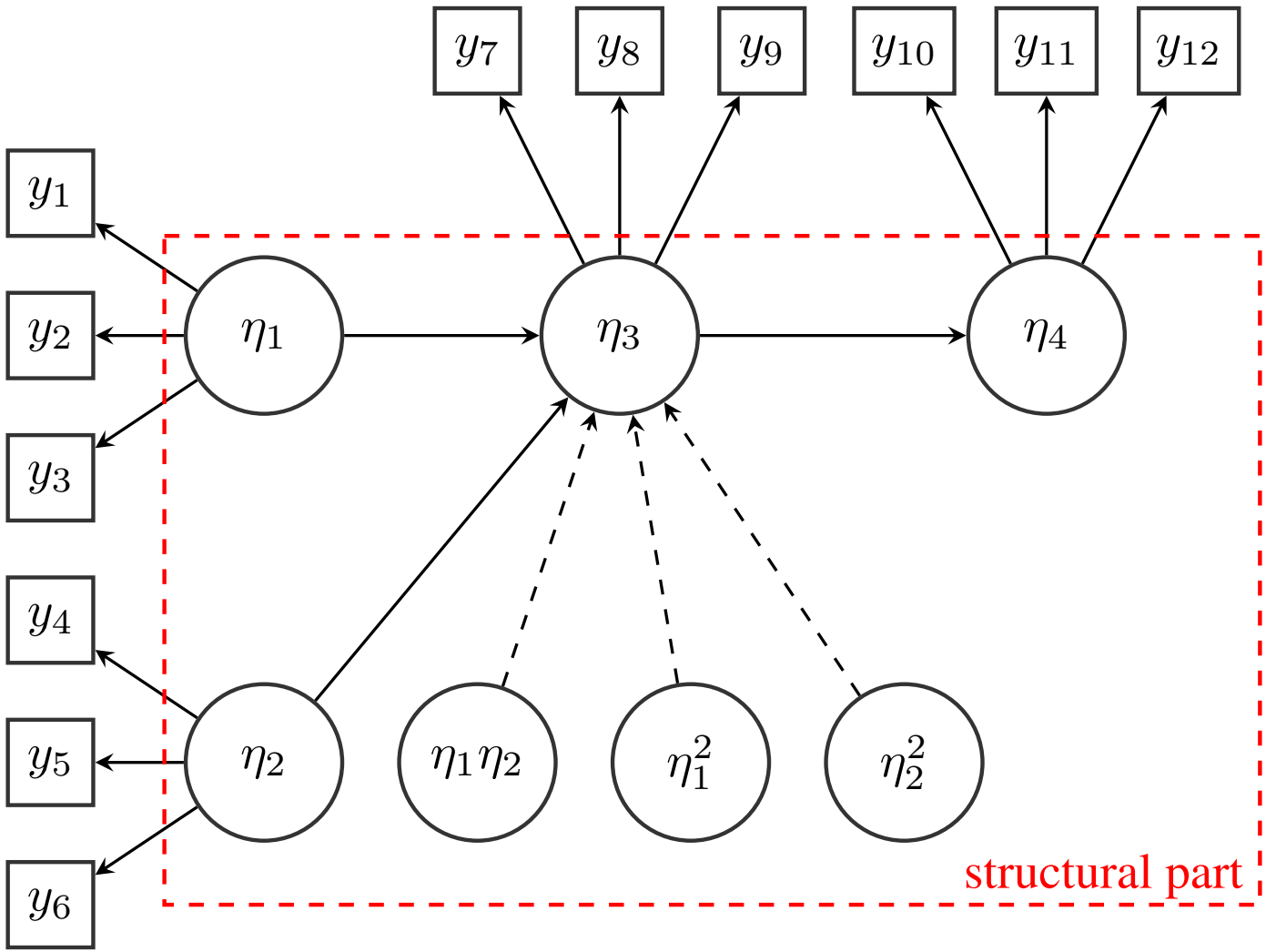
```
> # compute regression coefficients for M
> beta.M <- ( solve(Var.eta[c("X1", "X2"), c("X1", "X2")]) %*%
              Var.eta[c("X1", "X2"), "M", drop = FALSE] )
> round(beta.M, 3)

      M
X1 -0.3
X2  1.1



> # compute regression coefficients for Y
> beta.Y <- ( solve(Var.eta[c("X3", "M", "Age"), c("X3", "M", "Age")]) %*%
              Var.eta[c("X3", "M", "Age"), "Y", drop = FALSE] )
> round(beta.Y, 3)

        Y
X3   0.25
M    0.40
Age -0.10
```

example: using the sam() function

```
> fit.lsam <- sam(model = model, data = Data)
> parameterEstimates(fit.lsam, ci = FALSE, output = "text")[1:5,]
```

```
Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  Y ~
    X3                0.250    0.109    2.299    0.021
    M                 0.400    0.087    4.581    0.000
    Age              -0.100    0.094   -1.060    0.289
  M ~
    X1               -0.300    0.133   -2.249    0.025
    X2                1.100    0.173    6.347    0.000
```

4.4 Application 1: adding latent quadratic and interaction terms



- in the joint setting, adding latent quadratic/interaction terms is not trivial
- two popular methods are the product-indicator (PI) approach, and the so-called 'Latent Moderated Structural Equations' (LMS) approach
- none of these scale well: they cannot handle many quadratic and latent interaction terms simultaneously
- but if you can decouple the measurement and structural part, this becomes feasible
- a very general SAM solution (allowing for polynomial relations between latent variables) was already described in Wall & Amemiya (2000)
- the local SAM approach: find an explicit expression for

$$\mathrm{E}(\boldsymbol{\eta} \otimes \boldsymbol{\eta}) \qquad \text{and} \qquad \mathrm{Var}(\boldsymbol{\eta} \otimes \boldsymbol{\eta})$$

  where $\otimes$ denotes the tensor (or Kronecker) product

local SAM and interaction/quadratic terms

- creating the 'augmented' vector $\boldsymbol{\eta} \otimes \boldsymbol{\eta}$:
- several elements are duplicated
- based on the model, we select what we need

$$\boldsymbol{\eta} = \begin{pmatrix} 1 \\ \eta_1 \\ \eta_2 \\ \eta_3 \end{pmatrix}$$

$$\boldsymbol{\eta} \otimes \boldsymbol{\eta} = \begin{pmatrix} 1 \\ \eta_1 \\ \eta_2 \\ \eta_3 \\ \eta_1 1 \\ \eta_1\eta_1 \\ \eta_1\eta_2 \\ \eta_1\eta_3 \\ \eta_2 1 \\ \eta_2\eta_1 \\ \eta_2\eta_2 \\ \eta_2\eta_3 \\ \eta_3 1 \\ \eta_3\eta_1 \\ \eta_3\eta_2 \\ \eta_3\eta_3 \end{pmatrix} = \begin{pmatrix} 1 \\ \eta_1 \\ \eta_2 \\ \eta_3 \\ \eta_1 \\ \eta_1^2 \\ \eta_1\eta_2 \\ \eta_1\eta_3 \\ \eta_2\eta_1 \\ \eta_2 \\ \eta_2^2 \\ \eta_2\eta_3 \\ \eta_3 \\ \eta_3\eta_1 \\ \eta_3\eta_2 \\ \eta_3^2 \end{pmatrix}$$

the augmented (latent) sample statistics

- augmented mean vector:

$$\mathrm{E}(\boldsymbol{\eta} \otimes \boldsymbol{\eta}) = \mathrm{vec}[\mathrm{Var}(\boldsymbol{\eta})] + \mathrm{E}(\boldsymbol{\eta}) \otimes \mathrm{E}(\boldsymbol{\eta})$$

- augmented variance-covariance matrix (simple version, assuming normality for the measurement error):

$$\mathrm{Var}(\boldsymbol{\eta} \otimes \boldsymbol{\eta}) \approx \mathrm{Var}(\mathbf{f} \otimes \mathbf{f}) - \left[\mathbf{Q} + \mathbf{K}_m\mathbf{Q} + \mathbf{Q}\mathbf{K}_m^T + \mathbf{K}_m\mathbf{Q}\mathbf{K}_m^T + \boldsymbol{\Gamma}_{22}^{\star(NT)}(\mathbf{r})\right]$$

where

$$\mathbf{Q} = \mathrm{Var}(\boldsymbol{\eta}) \otimes \mathrm{Var}(\mathbf{r}) + \mathrm{E}(\boldsymbol{\eta})\mathrm{E}(\boldsymbol{\eta})^T \otimes \mathrm{Var}(\mathbf{r})$$

and

$$\boldsymbol{\Gamma}_{22}^{\star(NT)}(\mathbf{r}) = \left(\mathbf{I}_{m^2} + \mathbf{K}_m\right)\left(\mathrm{Var}(\mathbf{r}) \otimes \mathrm{Var}(\mathbf{r})\right)$$

- $\mathbf{K}_m$ is the commutation matrix
- $\boldsymbol{\Gamma}_{22}^{\star(NT)}(\mathbf{r})$ is the 'Gamma' matrix of the measurement error $(\mathbf{r})$

implementation in lavaan

```
> model <- '
      # measurement part
      f1 =~ y1 + y2 + y3
      f2 =~ y4 + y5 + y6
      f3 =~ y7 + y8 + y9

      # structural part
      f3 ~ f1 + f2 + f1:f1 + f2:f2 + f1:f2
  '
> fit <- sam(model, data = Data, se = "none") # or se = "bootstrap"
```

- two-step analytic standard errors can be obtained using the 'local' approach (work in progress)
- recent paper:

  Rosseel, Y., Burghgraeve, E., Loh, W.W., Schermelleh-Engel, K. (2025). Structural after Measurement (SAM) approaches for accommodating latent quadratic and interaction effects. *Behavior Research Methods*.

4.5 Application 2: noniterative SEM

- for CFA, many noniterative estimators are available; some (i.e., the multiple group method) perform better than ML in terms of mean squared error

  Dhaene, S. & Rosseel, Y. (2023). An Evaluation of Non-Iterative Estimators in Confirmatory Factor Analysis. *Structural Equation Modeling: A Multidisciplinary Journal*.

- we can use these noniterative estimators for the measurement part in SAM

  Dhaene, S., & Rosseel, Y. (2023). An Evaluation of Non-Iterative Estimators in the Structural after Measurement (SAM) Approach to Structural Equation Modeling (SEM). *Structural Equation Modeling: A Multidisciplinary Journal*, 30(6), 926–940

- "[the] local SAM approach outperforms traditional SEM in small to moderate samples (both in terms of convergence and MSE values), especially when reliability drops. "

4.6 Application 3: comparing structural relations across many groups

- reference (open access):

  > Perez Alonso, A.F., Rosseel, Y., Vermunt, J.K., & De Roover, K. (in press). Mixture Multigroup Structural Equation Modeling: A Novel Method for Comparing Structural Relations Across Many Groups. *Psychological Methods*. https://doi.org/10.1037/met0000667

- relationships between latent variables are often different across groups (e.g., countries); but some groups may be similar in the sense that they have similar values for the regression coefficients; we like to 'discover' these hidden clusters of similar groups

- in a first step, we estimated the measurement part across all groups (fixing the factor loadings to be the same across groups); this resulted in (model-implied) latent (co)variance matrices for all the groups

- in a second step, a mixture modeling approach is used to find homogeneous clusters that share similar regression coefficients

5 SAM: other approaches

5.1 The single-indicator (SI) approach

- a good reference:

  > Savalei, V. (2019). A comparison of several approaches for controlling measurement error in small samples. *Psychological methods, 24*(3), 352–370.

- if a first step, we estimate the measurement blocks, typically one by one, and we compute factor scores for each latent variable

- alternatively, we could also directly compute sum (or mean) scores, without fitting a measurement model

- in the second step, we fit the structural part; we keep the latent variables, but we replace their indicators by a single indicator (either the factor scores or the sum scores)

setting the reliability of the single indicator

- the (single) factor loading is set to one
- the residual variance of the single indicator is fixed to a constant; let Y_j be the single indicator of the j-th latent variable; we set the residual variance of Y_j as follows:

$$\text{Var}(\epsilon_j) = (1 - \text{REL}_j)\text{Var}(Y_j)$$

- REL_j reflects the 'reliability' of Y_j
- there are (at least) three ways to choose the reliability:
  1. we can estimate REL_j based on the measurement model ('model-based')
  2. we can provide a fixed guess (say, 0.7 or 0.8)
  3. we can estimate the reliability using, for example, coefficient alpha
- therefore, several variants of this single indicator approach are possible

SI + sum scores + fixed reliability (0.8)

```
> Ymean <- rowSums(Data[, c("y1", "y2", "y3" ,"y4")])/4
> Mmean <- rowSums(Data[, c("m1", "m2", "m3" ,"m4")])/4
> X1mean <- rowSums(Data[, c("x1", "x2", "x3" ,"x4")])/4
> X2mean <- rowSums(Data[, c("x5", "x6", "x7" ,"x8")])/4
> X3mean <- rowSums(Data[, c("x9", "x10", "x11" ,"x12")])/4
> CS <- data.frame(Y=Ymean, M=Mmean, X1=X1mean, X2=X2mean,
                   X3=X3mean, Age = Data$Age)

> res.Y <- (1 - 0.8)*var(Ymean)
> res.M <- (1 - 0.8)*var(Mmean)
> res.X1 <- (1 - 0.8)*var(X1mean)
> res.X2 <- (1 - 0.8)*var(X2mean)
> res.X3 <- (1 - 0.8)*var(X3mean)
```

```
> # create model syntax
> model.si <- c("fY =~ 1*Y", paste0("Y ~~ ", res.Y, "*Y"),
                "fM =~ 1*M", paste0("M ~~ ", res.M, "*M"),
                "fX1 =~ 1*X1", paste0("X1 ~~ ", res.X1, "*X1"),
                "fX2 =~ 1*X2", paste0("X2 ~~ ", res.X2, "*X2"),
                "fX3 =~ 1*X3", paste0("X3 ~~ ", res.X3, "*X3"),
                "fAge =~ 1*Age", "Age ~~ 0*Age")
> model.struc <- '
   fY ~ fX3 + fM + fAge
   fM ~ fX1 + fX2
  '
> fit.si <- sem(c(model.si, model.struc), data = CS)
> parameterEstimates(fit.si, ci = FALSE, output = "text")[13:17,]

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  fY ~
    fX3               0.246    0.100    2.460    0.014
    fM                0.513    0.098    5.216    0.000
    fAge             -0.080    0.082   -0.974    0.330
  fM ~
    fX1              -0.186    0.092   -2.024    0.043
    fX2               0.832    0.097    8.578    0.000
```

SI + factor scores + model-based reliability

```
> fit.Y <- sem('Y =~ y1 + y2 + y3 + y4', data = Data)
> fit.M <- sem('M =~ m1 + m2 + m3  + m4',  data = Data)
> fit.X1 <- sem('X1 =~ x1 + x2 + x3  + x4',  data = Data)
> fit.X2 <- sem('X2 =~ x5 + x6 + x7  + x8',  data = Data)
> fit.X3 <- sem('X3 =~ x9 + x10 + x11 + x12', data = Data)

> fs.Y <- lavPredict(fit.Y, method = "Bartlett", fsm = TRUE)
> fs.M <- lavPredict(fit.M, method = "Bartlett", fsm = TRUE)
> fs.X1 <- lavPredict(fit.X1, method = "Bartlett", fsm = TRUE)
> fs.X2 <- lavPredict(fit.X2, method = "Bartlett", fsm = TRUE)
> fs.X3 <- lavPredict(fit.X3, method = "Bartlett", fsm = TRUE)
> FS <- data.frame(Y=fs.Y, M=fs.M, X1=fs.X1, X2=fs.X2,
                   X3=fs.X3, Age = Data$Age)

> A <- attr(fs.Y, "fsm")[[1]]; THETA <- lavInspect(fit.Y, "est")$theta
> res.Y <- drop(A %*% THETA %*% t(A))
> A <- attr(fs.M, "fsm")[[1]]; THETA <- lavInspect(fit.M, "est")$theta
> res.M <- drop(A %*% THETA %*% t(A))
> A <- attr(fs.X1, "fsm")[[1]]; THETA <- lavInspect(fit.X1, "est")$theta
> res.X1 <- drop(A %*% THETA %*% t(A))
> A <- attr(fs.X2, "fsm")[[1]]; THETA <- lavInspect(fit.X2, "est")$theta
> res.X2 <- drop(A %*% THETA %*% t(A))
> A <- attr(fs.X3, "fsm")[[1]]; THETA <- lavInspect(fit.X3, "est")$theta
> res.X3 <- drop(A %*% THETA %*% t(A))
```

```
> # create model syntax
> model.si <- c("fY =~ 1*Y", paste0("Y ~~ ", res.Y, "*Y"),
                "fM =~ 1*M", paste0("M ~~ ", res.M, "*M"),
                "fX1 =~ 1*X1", paste0("X1 ~~ ", res.X1, "*X1"),
                "fX2 =~ 1*X2", paste0("X2 ~~ ", res.X2, "*X2"),
                "fX3 =~ 1*X3", paste0("X3 ~~ ", res.X3, "*X3"),
                "fAge =~ 1*Age", "Age ~~ 0*Age")
> model.struc <- '
    fY ~ fX3 + fM + fAge
    fM ~ fX1 + fX2
  '
> fit.si <- sem(c(model.si, model.struc), data = FS)
> parameterEstimates(fit.si, ci = FALSE, output = "text")[13:17,]

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  fY ~
    fX3               0.250    0.105    2.374    0.018
    fM                0.400    0.079    5.088    0.000
    fAge             -0.100    0.094   -1.064    0.287
  fM ~
    fX1              -0.300    0.130   -2.314    0.021
    fX2               1.100    0.132    8.355    0.000
```

- note: when using Bartlett factor scores, the diagonal elements of $\mathbf{E} = \mathbf{A}\,\mathbf{\Theta}\,\mathbf{A}^T$ contain the needed residual variances

5.2 Global SAM

- global SAM is very close to regular SEM
- in a first step, we estimate the parameters of the measurement blocks (either one by one, or in a single CFA)
- in a second step, we fit the full SEM, but we fix the values of the measurement parameters to the values we obtained in the first step; they are no longer free parameters
- this is the approach that was advocated in the early literature (e.g., Burt 1976), and many people associate 'SAM' (or 'two-step') with this approach
- because the second step 'sees' the full model (measurement and structural), Rosseel & Loh (2024) called this 'global' SAM
- two-step corrected standard errors are available
- when the model is correctly specified, global SAM usually provides identical results as local SAM

global SAM using the sam() function

```
> model <- '
    # measurement part
    Y =~ y1 + y2 + y3 + y4
    M =~ m1 + m2 + m3 + m4
    X1 =~ x1 + x2 + x3 + x4
    X2 =~ x5 + x6 + x7 + x8
    X3 =~ x9 + x10 + x11 + x12
    fAge =~ 1*Age; Age ~~ 0*Age

    # structural part
    Y ~ X3 + M + fAge
    M ~ X1 + X2
  '
> fit.gsam <- sam(model, data = Data, sam.method = "global")
> parameterEstimates(fit.gsam, ci = FALSE, output = "text")[1:5,]

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  Y ~
    X3                0.250    0.109    2.299    0.021
    M                 0.400    0.087    4.581    0.000
    fAge             -0.100    0.094   -1.060    0.289
  M ~
    X1               -0.300    0.133   -2.249    0.025
    X2                1.100    0.173    6.347    0.000
```

6 Consistency versus MSE

6.1 Consistency

- (informally:) an estimator is said to be consistent if the estimates (produced by the estimator) "converge" to the true value of the parameter being estimated, as the sample size increases (to infinity)
- in statistics, consistency is considered to be a fundamental property of estimators; if an estimator is not consistent, it is often disregarded immediately
- which SAM approaches are consistent:
  - FSR with Skrondal & Laake (2001) factor scores
  - FSR with correlation-preserving factor scores
  - instrumental variables with 2SLS
  - the method of Croon, local SAM
  - single-indicator + Bartlett factor scores + model-based reliability
  - global SAM

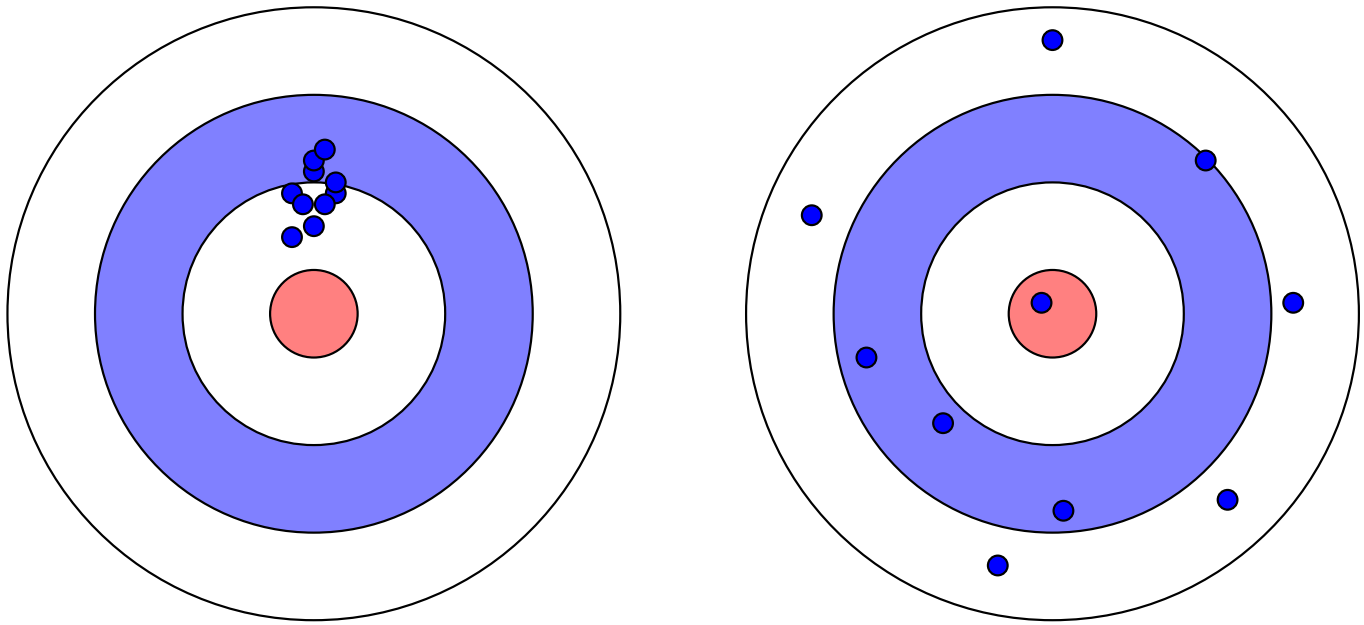
- which SAM approaches are NOT consistent:
  - uncorrected FSR (with Bartlett or regression factor scores)
  - sum (or mean) scores
  - single-indicator + fixed/alpha reliability
- (ML/GLS/WLS as used in SEM are also consistent estimators)
- consistency is not the same as unbiasedness
- (informally:) an estimator is unbiased if the *average* estimate across a very large (infinite) number of samples approaches the true value
- example: the sample variance $\sum_i (X_i - \bar{X})^2 / N$ divided by N (instead of $N - 1$) is biased but consistent
- example: if you always take the first observation (X_1) as the estimate of a mean, then this is unbiased, but not consistent
- in a simulation study, we usually study the bias of an estimator (given a specific sample size)

6.2 Bias, variability and mean squared error (MSE)

- unbiasedness is nice, but there is a price to be paid: unbiased estimators often exhibit increased variability
- on the other hand, a biased estimator may exhibit less variability (in comparison)
- this is known as the bias–variance trade–off
- one way to capture both bias and variance in a single quantity is to use the mean squared error (MSE)
- MSE = bias^2 + variance
- in statistics, when comparing estimators, we often prefer estimators that have the lowest MSE
- in the SEM literature, MSE has been mostly neglected (in favor of unbiasedness)

low bias versus low MSE

- suppose you are playing darts; which player would you like to be?



- left panel: biased, but low variability
- right panel: unbiased, but high variability

MSE for SAM approaches

- 'consistent' approaches typically have the lowest MSE . . . if the sample size is large enough
- however, for small to medium sample sizes, it may very well be that inconsistent approaches (for example: uncorrected FSR) have a lower MSE
- the next two figures are taken from:

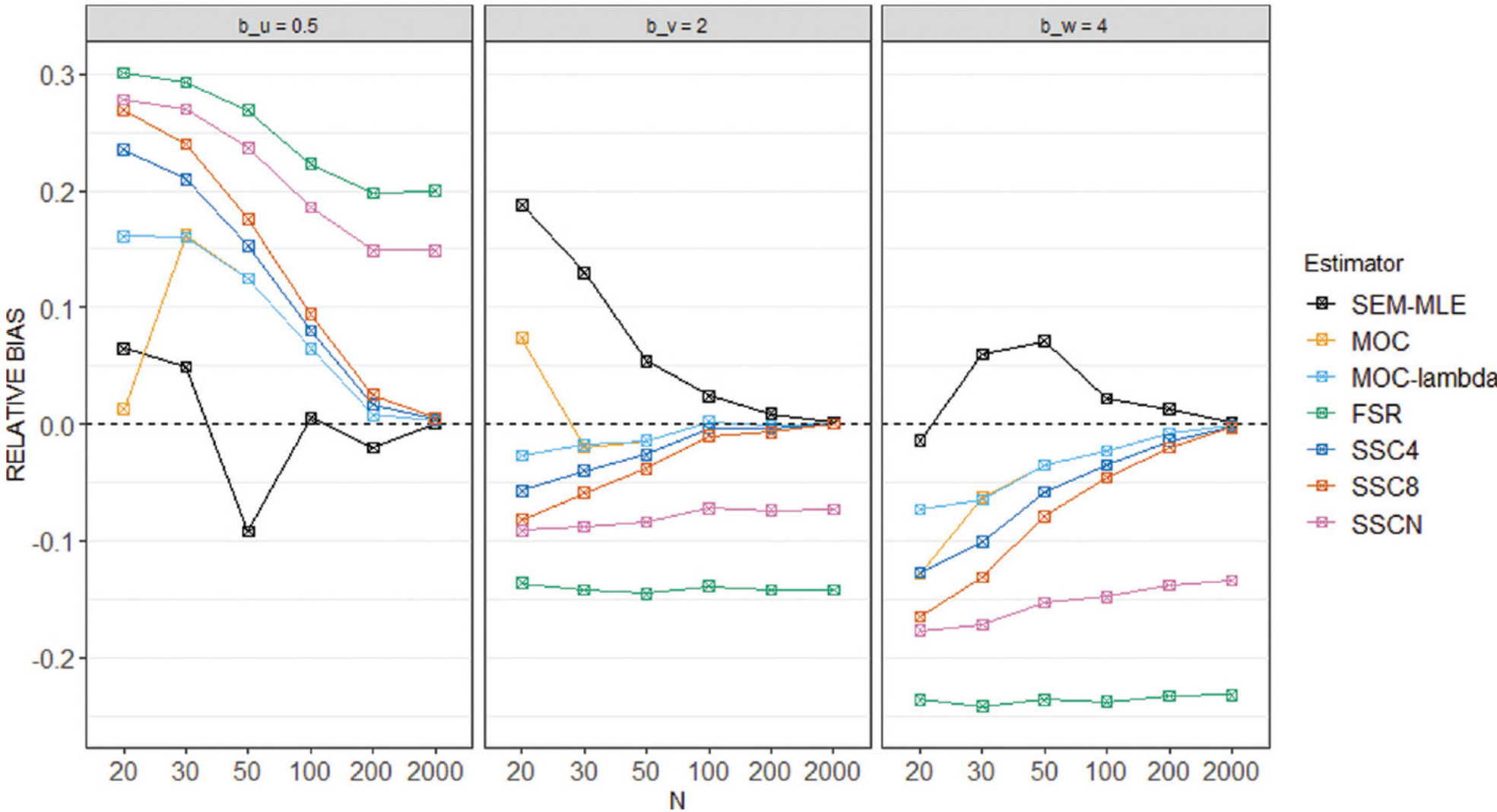
  Bogaert, J., Loh, W.W., & Rosseel, Y. (2023). A small sample correction for factor score regression. *Educational and Psychological Measurement, 83*(3), 495–519.

- similar results regarding single-indicator approaches:

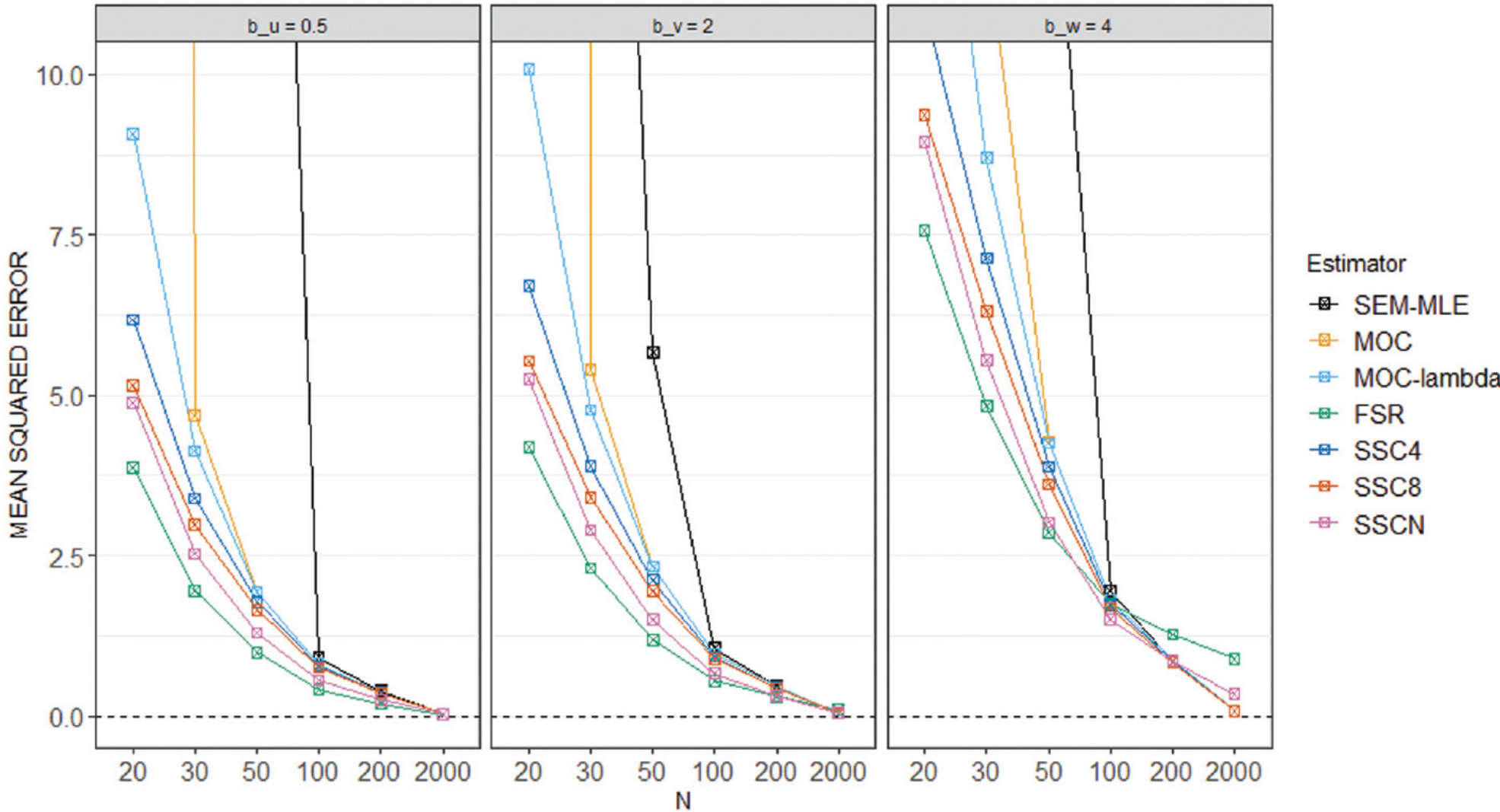
  Savalei, V. (2019). A comparison of several approaches for controlling measurement error in small samples. *Psychological methods, 24*(3), 352–370.

- in the Savalei paper, SI + sum scores + fixed reliability (0.8) had the lowest MSE when the sample size was small (N=30 to N=200)

reliative bias (figure 3)



mean squared error (figure 7)



the alpha correction

- local SAM is consistent, because it uses

$$\widehat{\text{Var}(\boldsymbol{\eta})} = \text{Var}(\mathbf{f}) - \widehat{\text{Var}(\mathbf{r})}$$

  where $\widehat{\text{Var}(\mathbf{r})}$ is the correction (for measurement error) term

- uncorrected FSR (with Bartlett factor scores) is not consistent, because it uses

$$\widehat{\text{Var}(\boldsymbol{\eta})} = \text{Var}(\mathbf{f})$$

- Bogaert, Loh, & Rosseel (2023) discuss the so-called 'alpha' correction

$$\widehat{\text{Var}(\boldsymbol{\eta})} = \text{Var}(\mathbf{f}) - \alpha^{\star}\widehat{\text{Var}(\mathbf{r})}$$

  where $\alpha^{\star}$ is (here) a value between 0 and 1

- if $\alpha^{\star} = 1$, we have local SAM; if $\alpha^{\star} = 0$, we have uncorrected FSR

- future research: for a given model, a given sample size: find the sweet spot: the value for $\alpha^{\star}$ that results (on average) in the lowest MSE

coverage probability

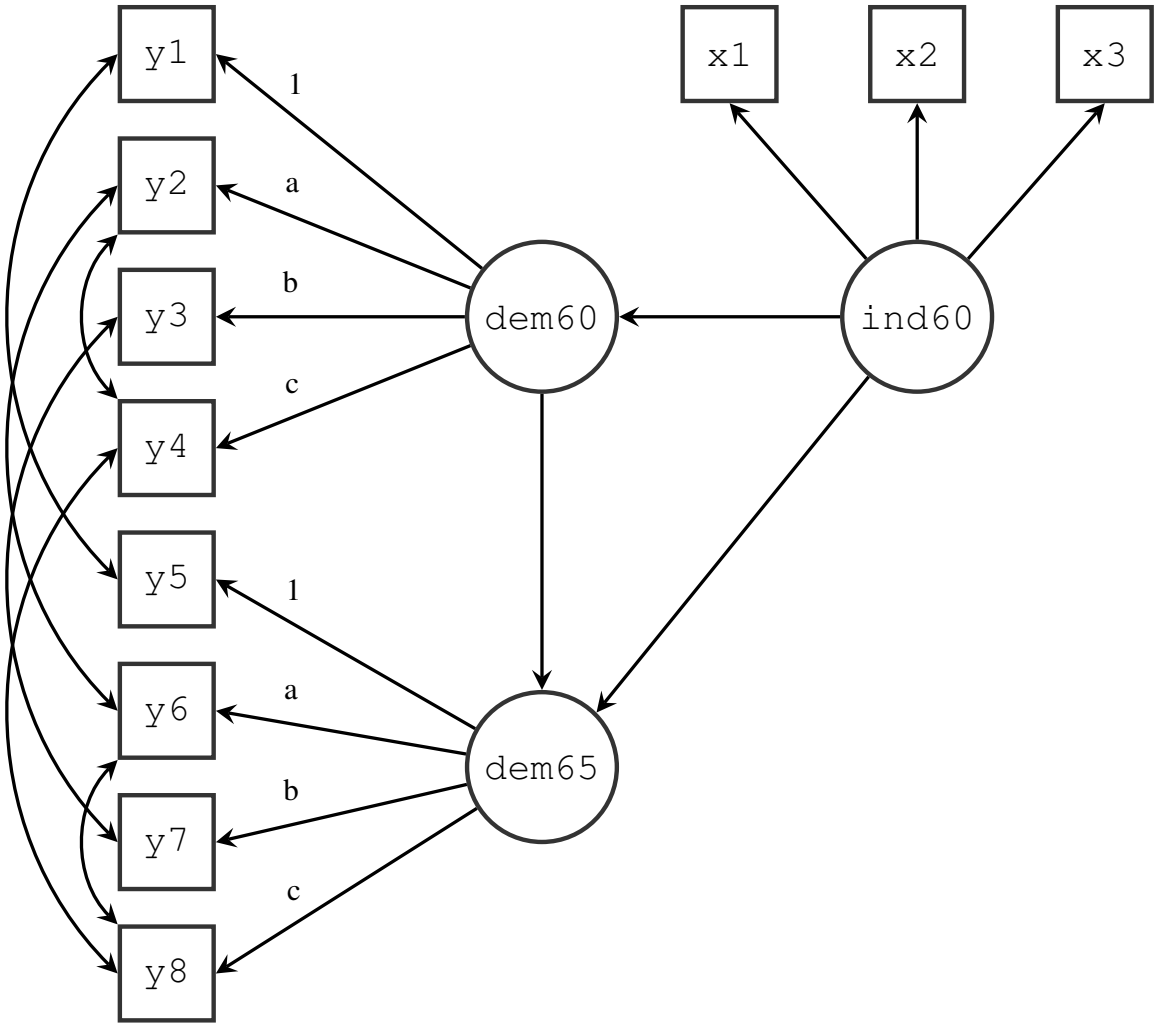
- low MSE is good, but ...
- for valid inference, we also need good 'coverage'
- the coverage is the probability that a confidence interval will include the true value of the parameter of interest
- for example, set the nominal coverage probability at 0.95; for a large number of replications, the proportion of replications that produce a confidence interval (for the parameter of interest) that includes the true value should approximate 0.95
- ideally, we have it all: almost no bias, not much variability, and good coverage
- so which SAM method is best? (it will depend on many factors, including sample size)
- as always, more research is needed

7 SAM: software implementation

- some of the SAM approaches have been implemented in the sam() function in the R package lavaan:
  - `sam.method = "local"` (default)
  - `sam.method = "global"`
  - `sam.method = "fsr"` (using Bartlett factor scores)
- typical call:

```
> fit.sam <- sam(model, data = Data, sam.method = "local",
                 local.options = list(M.method = "ML",
                                      alpha.correction = 0L), # range: 0 -- N-1
                 # optionally link measurement blocks
                 mm.list = list(block1 = c("Y", "M"),
                                block2 = c("X1","X2","X3")),
                 # measurement options
                 mm.args = list(estimator = "ML"),
                 # structural options
                 struc.args = list(estimator = "GLS"),
                 # global options
                 meanstructure = FALSE)
```

example: the Political Democracy model



R code (with a misspecification in the structural part)

```
> model <- '
   # latent variable definitions
      ind60 =~ x1 + x2 + x3
      dem60 =~ y1 + a*y2 + b*y3 + c*y4
      dem65 =~ y5 + a*y6 + b*y7 + c*y8

   # regressions
     dem60 ~ ind60
     dem65 ~ ind60 + 0*dem60 # misspecified: fixed-to-zero

   # residual correlations
     y1 ~~ y5
     y2 ~~ y4 + y6
     y3 ~~ y7
     y4 ~~ y8
     y6 ~~ y8
 '


> fit.sam <- sam(model, data = PoliticalDemocracy,
                 # link measurement blocks
                 mm.list = list(ind = "ind60", dem = c("dem60", "dem65")),
                 struc.args = list(estimator = "GLS")) # just for fun
```

standard sam() output: only structural part

```
> # standard sam output
> summary(fit.sam)

This is lavaan 0.6-20.2277 -- using the SAM approach to SEM

  SAM method                                      LOCAL
  Mapping matrix M method                            ML
  Number of measurement blocks                        2
  Estimator measurement part                         ML
  Estimator  structural part                        GLS

  Number of observations                             75

Summary Information Measurement + Structural:

  Block      Latent Nind Chisq Df
      1       ind60    3  0.00  0
      2 dem60,dem65    8 15.32 16

  Model-based reliability latent variables:

  ind60 dem60 dem65
  0.966 0.868  0.87

  Summary Information Structural part:
```

```
   chisq df   cfi rmsea  srmr
  35.817  1 0.287 0.686 0.505

Parameter Estimates:

  Standard errors                             Twostep
  Information                                Expected
  Information saturated (h1) model          Structured

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  dem60 ~
    ind60             1.454    0.195    7.452    0.000
  dem65 ~
    ind60             1.824    0.214    8.506    0.000
    dem60             0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
    ind60             0.446    0.087    5.135    0.000
   .dem60             0.120    0.161    0.749    0.454
   .dem65             0.097    0.153    0.636    0.525
```

sam() output, including measurement part

```
> # just for illustration, we also show the estimated parameters
> # of the measurement blocks
> #
> summary(fit.sam, remove.step1 = FALSE)

This is lavaan 0.6-20.2277 -- using the SAM approach to SEM

  SAM method                                        LOCAL
  Mapping matrix M method                              ML
  Number of measurement blocks                          2
  Estimator measurement part                           ML
  Estimator  structural part                          GLS

  Number of observations                               75

Summary Information Measurement + Structural:

  Block       Latent Nind Chisq Df
      1        ind60    3  0.00  0
      2 dem60,dem65    8 15.32 16

  Model-based reliability latent variables:

  ind60 dem60 dem65
  0.966 0.868  0.87
```

```
  Summary Information Structural part:

   chisq df   cfi rmsea  srmr
  35.817  1 0.287 0.686 0.505

Parameter Estimates:

  Standard errors                             Twostep
  Information                                Expected
  Information saturated (h1) model         Structured

Latent Variables:
                     Step Estimate  Std.Err  z-value  P(>|z|)
  ind60 =~
    x1                  1    1.000
    x2                  1    2.193    0.142   15.403    0.000
    x3                  1    1.824    0.153   11.883    0.000
  dem60 =~
    y1                  1    1.000
    y2         (a)      1    1.213    0.143    8.483    0.000
    y3         (b)      1    1.210    0.125    9.690    0.000
    y4         (c)      1    1.273    0.122   10.453    0.000
  dem65 =~
    y5                  1    1.000
    y6         (a)      1    1.213    0.143    8.483    0.000
    y7         (b)      1    1.210    0.125    9.690    0.000
    y8         (c)      1    1.273    0.122   10.453    0.000
```

```
Regressions:
                   Step Estimate  Std.Err  z-value  P(>|z|)
  dem60 ~
    ind60             2    1.454    0.195    7.452    0.000
  dem65 ~
    ind60             2    1.824    0.214    8.506    0.000
    dem60             2    0.000

Covariances:
                   Step Estimate  Std.Err  z-value  P(>|z|)
 .y1 ~~
   .y5                1    0.577    0.364    1.585    0.113
 .y2 ~~
   .y4                1    1.390    0.685    2.030    0.042
   .y6                1    2.068    0.733    2.822    0.005
 .y3 ~~
   .y7                1    0.727    0.611    1.190    0.234
 .y4 ~~
   .y8                1    0.476    0.453    1.049    0.294
 .y6 ~~
   .y8                1    1.257    0.583    2.156    0.031

Variances:
                   Step Estimate  Std.Err  z-value  P(>|z|)
   .x1                1    0.084    0.020    4.140    0.000
   .x2                1    0.108    0.074    1.455    0.146
   .x3                1    0.468    0.091    5.124    0.000
   .y1                1    1.879    0.431    4.355    0.000
```

```
   .y2                1    7.530    1.363    5.523    0.000
   .y3                1    4.966    0.966    5.141    0.000
   .y4                1    3.214    0.722    4.449    0.000
   .y5                1    2.499    0.518    4.824    0.000
   .y6                1    4.809    0.924    5.202    0.000
   .y7                1    3.302    0.699    4.722    0.000
   .y8                1    3.227    0.720    4.482    0.000
    ind60             2    0.446    0.087    5.135    0.000
   .dem60             2    0.120    0.161    0.749    0.454
   .dem65             2    0.097    0.153    0.636    0.525
```

(optional) extract the various components

```
> out <- sam(model, data = PoliticalDemocracy,
             mm.list = list(ind = "ind60", dem = c("dem60", "dem65")),
             struc.args = list(estimator = "GLS"),
             output = "list")
> names(out)

 [1] "MM.FIT"          "Sigma.11"        "step1.free.idx" "block.mm.idx"
 [5] "block.ptm.idx"   "PT.free"         "mm.list"        "PT"
 [9] "COV"             "YBAR"            "LAMBDA"         "THETA"
[13] "VETA"            "REL"             "M"              "lambda"
[17] "alpha"           "MSM"             "MTM"            "FS.mean"
[21] "LV.NAMES"        "sam.method"      "local.options"  "FIT.PA"
[25] "PT"              "reg.idx"         "step2.free.idx" "extra.id"
[29] "pt.idx"          "pts.idx"         "V2"             "V1"
[33] "VCOV"



> out$MM.FIT[[1]]

lavaan 0.6-20.2277 ended normally after 21 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                         6
  Row rank of the constraints matrix                 6
```

```
  Number of observations                            75

Model Test User Model:

  Test statistic                                 0.000
  Degrees of freedom                                 0


> out$FIT.PA

lavaan 0.6-20.2277 ended normally after 28 iterations

  Estimator                                        GLS
  Optimization method                           NLMINB
  Number of model parameters                         5

  Number of observations                            75

Model Test User Model:

  Test statistic                                35.817
  Degrees of freedom                                 1
  P-value (Chi-square)                           0.000
```

Thank you for attending this workshop!

https://lavaan.org

https://lavaan.ugent.be/about/donate.html

references

Bakk, Z., & Kuha, J. (2021). Relating latent class membership to external variables: An overview. *British Journal of Mathematical and Statistical Psychology*, 74(2), 340–362.
https://doi.org/10.1111/bmsp.12227 (Open Access)

Bartlett, M. S. (1937). The statistical conception of mental factors. *British Journal of Psychology, 28*(1), 97–104

Bartlett, M. S. (1938). Methods of estimating mental factors. *Nature, 141*, 609–610.

Bogaert, J., Loh, W.W., & Rosseel, Y. (2023). A small sample correction for factor score regression. *Educational and Psychological Measurement, 83*(3), 495–519.

Bollen, K.A. (1996). An alternative two stage least squares (2SLS) estimator for latent variable equations. *Psychometrika, 61*(1), 109–121
https://doi.org/10.1007/BF02296961

Burt, R.S. (1976). Interpretational confounding of unobserved variables in structural equation models. *Sociological Methods & Research*, 5(1), 3–52.

Croon, M. (2002). *Using predicted latent scores in general latent structure models*. In Marcoulides, G., Moustaki, I. (Eds.), Latent variable and latent structure modeling (pp. 195–223). Mahwah, NJ: Lawrence Erlbaum.

Devlieger, I., Mayer, A., & Rosseel, Y. (2016). Hypothesis testing using factor score regression: A comparison of four methods. *Educational and Psychological Measurement, 76*, 741–770.

Devlieger, I., & Rosseel, Y. (2017). Factor Score Path Analysis. *Methodology, 13*, 31–38.

Dhaene, S., & Rosseel, Y. (2023). An Evaluation of Non-Iterative Estimators in the Structural after Measurement (SAM) Approach to Structural Equation Modeling (SEM). *Structural Equation Modeling: A Multidisciplinary Journal*, 30(6), 926–940
https://doi.org/10.1080/10705511.2023.2220135 (Open Access)

Dhaene, S. & Rosseel, Y. (2023). An Evaluation of Non-Iterative Estimators in Confirmatory Factor Analysis. *Structural Equation Modeling: A Multidisciplinary Journal*.
https://doi.org/10.1080/10705511.2023.2187285. (Open Access)

Dijkstra, T.K. (1981). *Latent variables in linear stochastic models*, PhD-thesis, Amsterdam: Sociometric Research Foundation.

Dijkstra, T.K., & Henseler, J. (2015). Consistent and asymptotically normal PLS estimators for linear structural equations. *Computational Statistics & Data Analysis, 81*, 10–23.
https://doi.org/10.1016/j.csda.2014.07.008

Fornell, C., & Yi, Y. (1992). Assumptions of the two-step approach to latent variable modeling. *Sociological Methods & Research*, 20(3), 291–320.

Fuller, W A. (1987). *Measurement error models*. John Wiley & Sons.

Green, B. F. (1969). Best linear composites with a specified structure. *Psychometrika, 34*(3), 301–318.

Hägglund, G. (1982). Factor analysis by instrumental variables methods. *Psychometrika, 47*(2), 209–222.

Hunter, J.E., & Gerbing, D. W. (1982). Unidimensional measurement, second-order factor analysis, and causal models. *Research in Organizational Behavior*, 4, 267–320.

Jöreskog, K.G. (1983). Factor analysis as an error-in-variables model. In H. Walner & S. Messick (Eds.) *Principles of modern psychological measurement* (pp. 185-196). Hillsdale, NJ: Lawrence Erlbaum Associates.

Kelcey, B. (2019). A robust alternative estimator for small to moderate sample SEM: Bias-corrected factor score path analysis. *Addictive Behaviors, 94*, 83–98.

Krijnen, W.P., Wansbeek, T., & ten Berge, J.M. (1996). Best linear predictors for factor scores. *Communications in Statistics–Theory and Methods, 25*(12), 3013–3025.

Lance, C.E., Cornwell, J.M., & Mulaik, S.A. (1988). Limited information parameter estimates for latent or mixed manifest and latent variable models. *Multivariate Behavioral Research*, 23 (2), 171–187.

Levy, R. (2023). Precluding interpretational confounding in factor analysis with a covariate or outcome via measurement and uncertainty preserving parametric modeling. *Structural Equation Modeling: A Multidisciplinary Journal.* https://doi.org/10.1080/10705511.2022.2154214

Madansky, A. (1964). Instrumental variables in factor analysis. *Psychometrika, 29*, 105–113.

McDonald, R.P. (1981). Constrained least squares estimators of oblique common factors. *Psychometrika, 46*(3), 337–341.

Perez Alonso, A.F., Rosseel, Y., Vermunt, J.K., & De Roover, K. (in press). Mixture Multigroup Structural Equation Modeling: A Novel Method for Comparing Structural Relations Across Many Groups. *Psychological Methods*.

Rosseel, Y., & Loh, W.W. (2024). A structural after measurement approach to structural equation modeling. *Psychological Methods, 29*(3), 561–588.
https://doi.org/10.1037/met0000503
https://osf.io/pekbm/ (includes original version)

Rosseel, Y., Burghgraeve, E., Loh, W.W., Schermelleh-Engel, K. (2025). Structural after Measurement (SAM) approaches for accommodating latent quadratic and interaction effects. *Behavior Research Methods*.
https://doi.org/10.3758/s13428-024-02532-y

Savalei, V. (2019). A comparison of several approaches for controlling measurement error in small samples. *Psychological methods, 24*(3), 352–370.

Sijtsma K., Ellis J.L., Borsboom D. (2024). Recognize the Value of the Sum Score, Psychometrics' Greatest Accomplishment. *Psychometrika, 89*(1), 84–117.
https://doi.org/10.1007/s11336-024-09964-7
Skrondal, A., & Laake, P. (2001). Regression among factor scores. *Psychometrika, 66*(4), 563–575.

Steiger J.H. (1996). Coming Full Circle in the History of Factor Indeterminancy. *Multivariate behavioral research, 31*(4), 617–630.
https://doi.org/10.1207/s15327906mbr3104_14

Takane, Y., & Hwang, H. (2017). Comparisons among several consistent estimators of structural equation models. *Behaviormetrika, 45*, 157–188.

Thomson, G. (1934). The meaning of i in the estimate of g. *British Journal of Psychology, 25*, 92–99

Wall, M.M., & Amemiya, Y. (2000). Estimation for polynomial structural equation models. *Journal of the American Statistical Association*, 95(451), 929–940.

Wall, M. M., & Amemiya, Y. (2003). A method of moments technique for fitting interaction effects in structural equation models. *British Journal of Mathematical and Statistical Psychology, 56*, 47–63.