The structural-after-measurement (SAM) approach to structural equation modeling

Yves Rosseel
Department of Data Analysis
Ghent University – Belgium

Seminar in Psychometrics
Prague, 3 May 2022

structure of this talk

- what is SEM?
- software for SEM
- standard (ML) estimation approach in SEM
- the structural-after-estimation (SAM) approach
  - joint work with Wen Wei Loh (postdoc UGent)
  - reference:

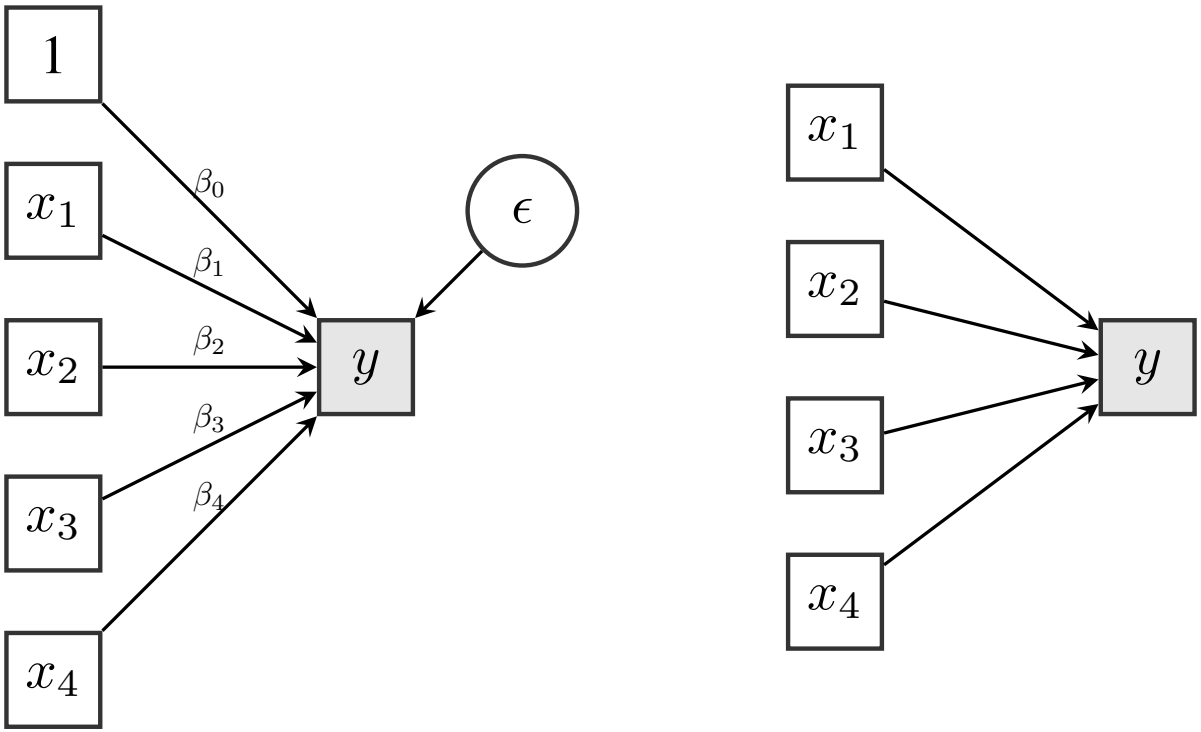
    Rosseel, Y., & Loh, W.W. (in press). A structural-after-measurement (SAM) approach to structural equation modeling. *Psychological Methods*. (preprint: `https://osf.io/pekbm/`)
- last slide

slides available at `https://lavaan.org`

SEM = structural equation modeling

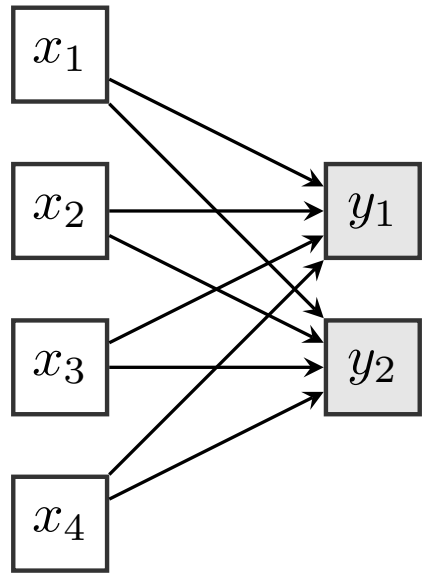
- SEM is a multivariate statistical modeling technique
- SEM allows us to test a hypothesis/model about the data
  - we postulate a data-generating model
  - this model may or may not fit the data
- what is so special about SEM?
  1. the model may contain latent variables
     - latent variables can be hypothetical 'constructs' (eg., depression) measured by a set of indicators
     - latent variables can be random effects (eg., random intercepts)
     - error terms, missing data, ...
  2. SEM allows for indirect effects (mediation), reciprocal effects, ...
  3. the model is depicted as a diagram

univariate linear regression



$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + \epsilon_i \quad (i = 1, 2, \ldots, n)$$

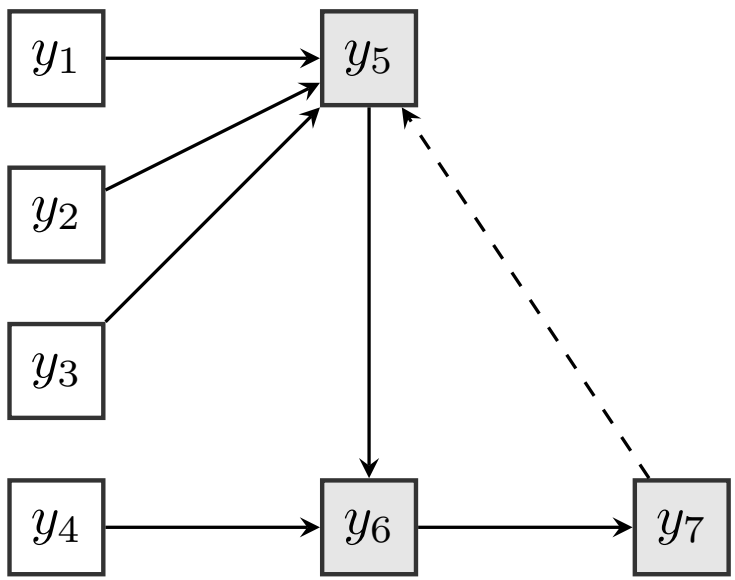
multivariate regression



- strict distinction between ‘dependent’ variables and ‘independent’ variables

SEM example: path analysis

- all variables are observed (manifest)
- we allow for indirect effects (eg., of y_5, via y_6 on y_7)
- we allow for cycles (eg. y_7 could influence y_5)



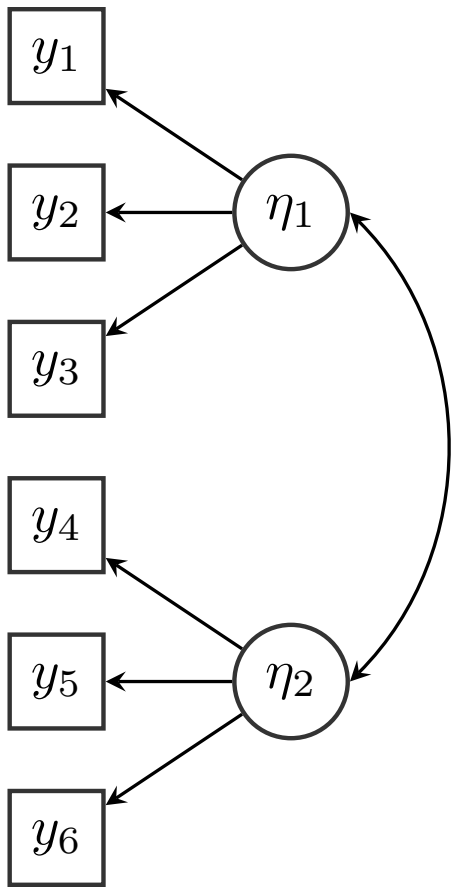
y_5 = reading motivation

y_6 = reading frequency

y_7 = reading ability

SEM example: confirmatory factor analysis (CFA)

- measurement model: representing the relationship between one or more latent variables and their (observed) indicators

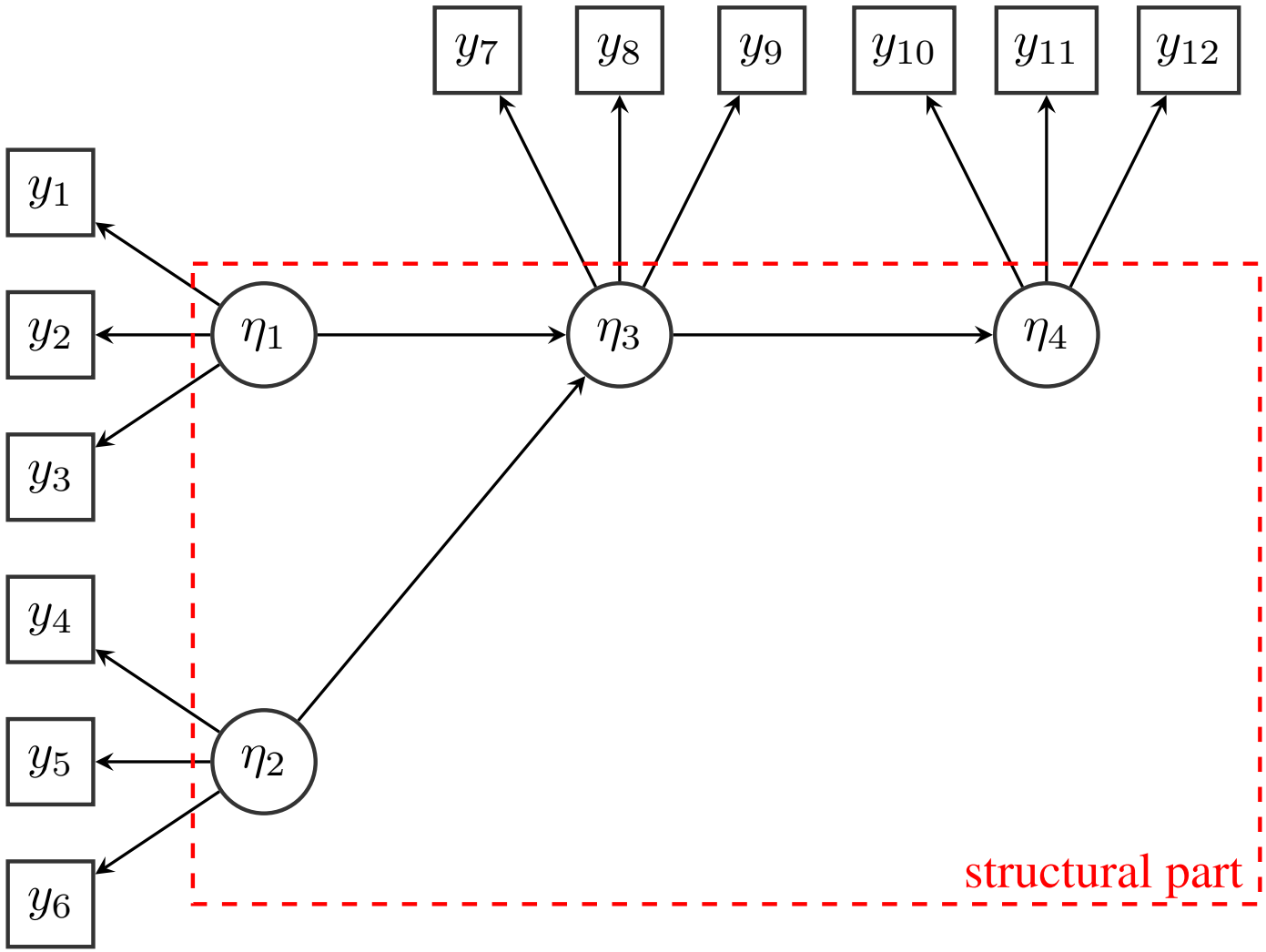


η_1 = depression

η_2 = neuroticism

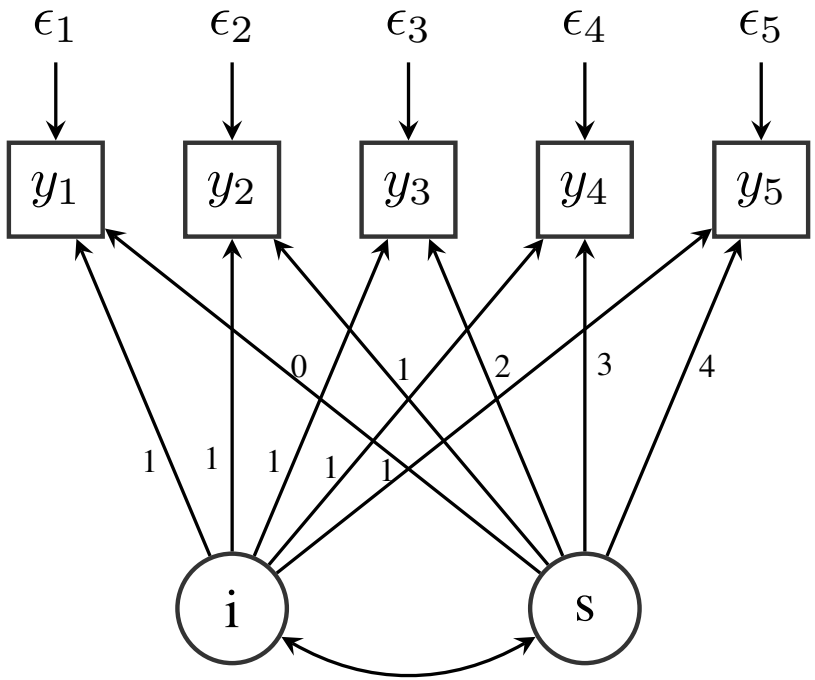
SEM example: measurement models + structural part

- path analysis with latent variables



SEM example: growth curve model

- random intercept and random slope



- y_t = intercept + slope*time + error

who is using SEM?

- it is widely used in the social sciences
- it is increasingly 'discovered' by other fields:
  - medical sciences
  - neuroimaging
  - biology, ecology (climate change!)
  - ...
- SEM software is also used to perform standard analyses (eg., regression), but where there is need for:
  - dealing with missing data, clustered data, categorical data
  - robust standard errors, goodness-of-fit measures
  - (in)equality constraints
  - ...

software for SEM: commercial – closed-source

- the big four (and the main developer):
  - LISREL (’70s, Karl Jöreskog)
  - EQS (’80s, Peter Bentler)
  - AMOS (’90s, James Arbuckle)
  - Mplus (Bengt Muthén, 1998-now)
- SAS/Stat: proc CALIS, proc TCALIS
- Statistica (SEPATH), Systat (RAMONA), Stata 12
- Mx (Michael Neale, free, closed-source, ’90s)
- (not in SPSS!)

software for SEM: non-commercial – open-source

- outside the R ecosystem:
  - 'gllamm' in stata (Rabe-Hesketh, Skrondal & Pickles, since 2002)
  - 'semopy' in python (`https://pypi.org/project/semopy/`) (since 2018)
- R packages:
  - sem (John Fox, since 2001)
  - OpenMx (Steven Boker, Michael Neale, . . . since 2009)
  - lavaan (Yves Rosseel, since 2010)
  - lava (Klaus Holst, since 2012)
  - psychonetrics (Sacha Epskamp, since 2019)
- interfaces between R and commercial packages:
  - REQS (Patrick Mair, Eric Wu, since 2008)
  - MplusAutomation (Michael Hallquist, since 2010)

the lavaan package in one page

- development started in 2009
- first CRAN version: 0.3-1 (May 2010)
- more information about lavaan:

  https://lavaan.org

- the lavaan paper:

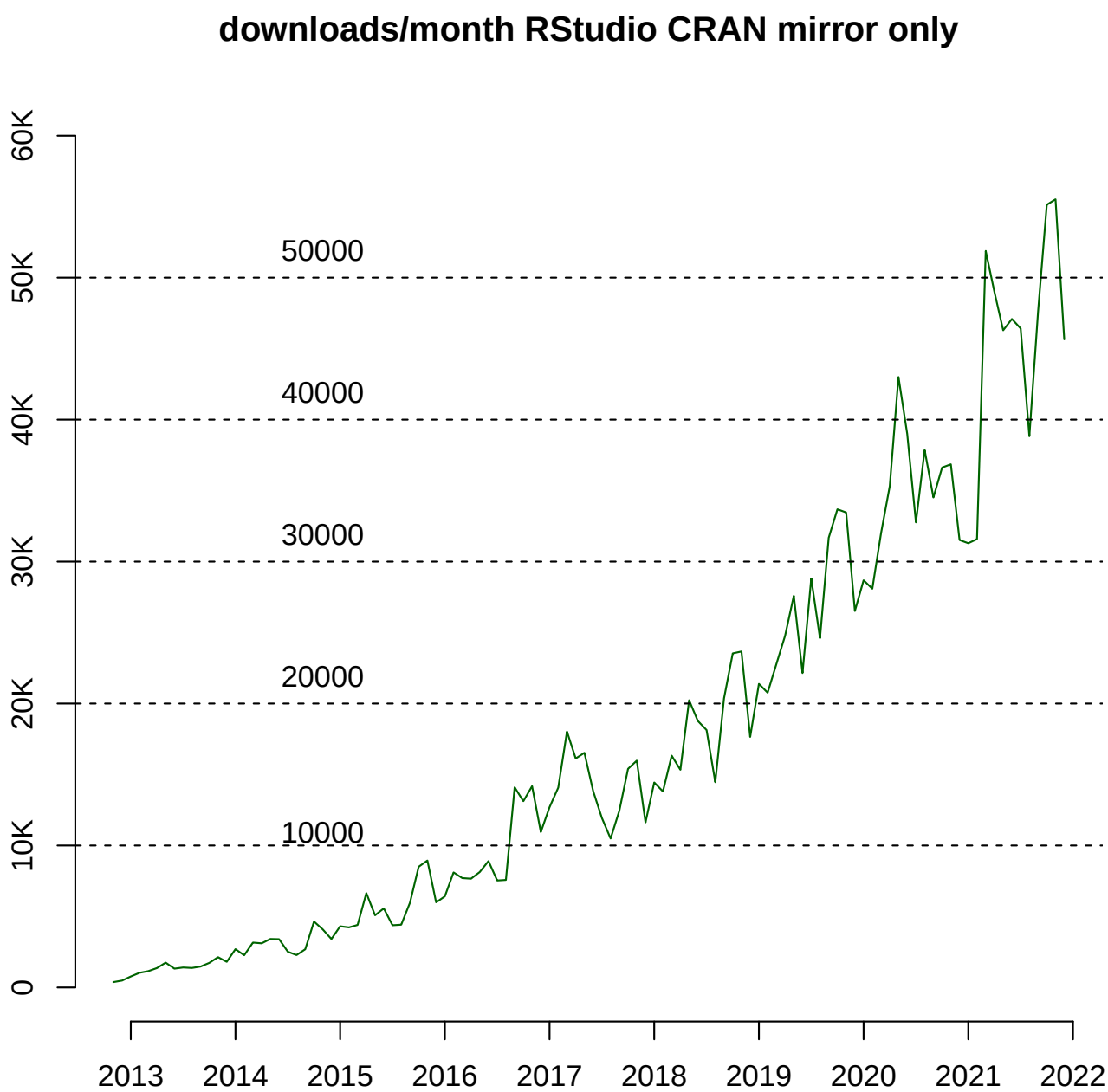
  Rosseel (2012). lavaan: an R package for structural equation modeling. *Journal of Statistical Software*, 48(2), 1–36.

- lavaan source code:

  https://github.com/yrosseel/lavaan

- lavaan discussion group (mailing list)

  https://groups.google.com/d/forum/lavaan

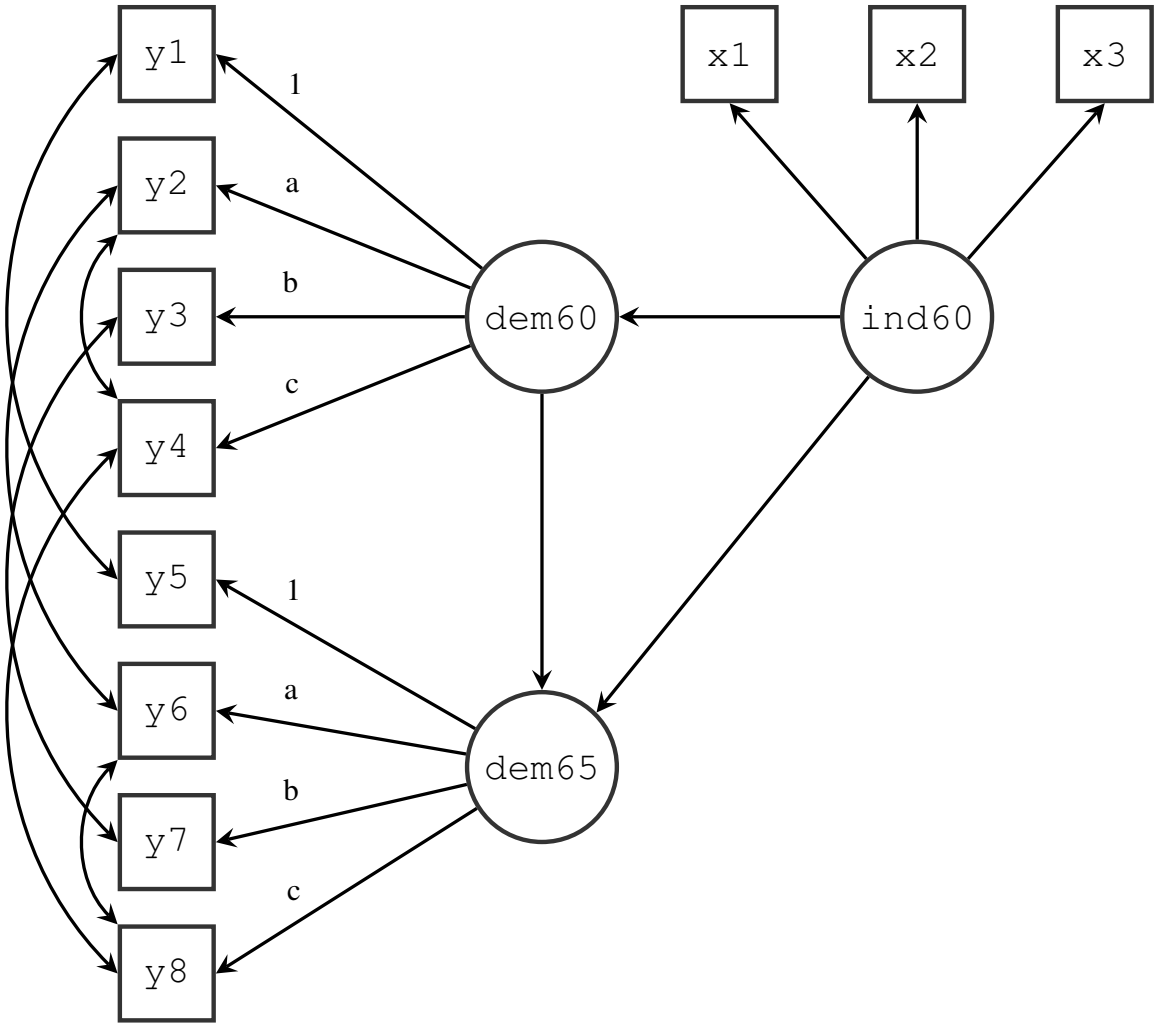

downloads/month RStudio CRAN mirror only
0
10K
20K
30K
40K
50K
60K
10000
20000
30000
40000
50000
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022

lavaan today

- current version: 0.6-11 (about 70,000 lines of R code)
- lavaan is a mainstream package for SEM
  - in the last years, 3 books about 'lavaan' have been published
- lavaan is widely used for both teaching and research
- the lavaan 'ecosystem' contains about 90 packages that depend on, or extend lavaan:

bain Bayesrel BifactorIndicesCalculator blavaan bmem bnpa bruceR coefficientalpha conmet CoTiMA covsim cSEM detectR discnorm dmacs eatRep EFAtools EffectLiteR EGAnet eqs2lavaan equaltestMI ezCutoffs faoutlier fSRM gimme gorica IIVpredictor influence.SEM IPV jmv JWileymisc kfa lavaan.shiny lavaan.survey lavaanPlot lcsm lsl lslx lvnet matrixpls MBESS medmod MedSurvey merDeriv metaSEM MIIVsem misty MonteCarloSEM multid multilevelTools nlsem nonnest2 pathmodelfit pompom processR profileR PROsetta pscore psychonetrics psycModel pwr2ppl qgraph RAMpath regmed regsem Replication restriktor RMediation rosetta RSA rsem semdrw SEMgraph seminr semnova semPlot semptools SEMsens semTable semTools semtree sesem ShortForm simsem simstandard thurstonianIRT tidySEM umx unusualprofile vampyr WebPower

lavaan example: political democracy dataset



fitting the model with lavaan

```
# 1. specifying the model
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 + dem60

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

# 2. fitting the model using the sem() function
fit <- sem(model, data = PoliticalDemocracy)

# 3. display the results
summary(fit, standardized = TRUE)
```

output

```
lavaan 0.6-11 ended normally after 66 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        31
  Number of equality constraints                     3

  Number of observations                            75

Model Test User Model:

  Test statistic                                40.179
  Degrees of freedom                                38
  P-value (Chi-square)                           0.374

Parameter Estimates:

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

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  ind60 =~
    x1                1.000                               0.670    0.920
    x2                2.180    0.138   15.751    0.000    1.460    0.973
```

```
    x3                1.818    0.152   11.971    0.000    1.218    0.872
  dem60 =~
    y1                1.000                               2.201    0.850
    y2         (a)    1.191    0.139    8.551    0.000    2.621    0.690
    y3         (b)    1.175    0.120    9.755    0.000    2.586    0.758
    y4         (c)    1.251    0.117   10.712    0.000    2.754    0.838
  dem65 =~
    y5                1.000                               2.154    0.817
    y6         (a)    1.191    0.139    8.551    0.000    2.565    0.755
    y7         (b)    1.175    0.120    9.755    0.000    2.530    0.802
    y8         (c)    1.251    0.117   10.712    0.000    2.694    0.829

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  dem60 ~
    ind60             1.471    0.392    3.750    0.000    0.448    0.448
  dem65 ~
    ind60             0.600    0.226    2.661    0.008    0.187    0.187
    dem60             0.865    0.075   11.554    0.000    0.884    0.884

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
 .y1 ~~
   .y5                0.583    0.356    1.637    0.102    0.583    0.281
 .y2 ~~
   .y4                1.440    0.689    2.092    0.036    1.440    0.291
   .y6                2.183    0.737    2.960    0.003    2.183    0.356
 .y3 ~~
```

```
   .y7                0.712    0.611    1.165    0.244    0.712    0.169
 .y4 ~~
   .y8                0.363    0.444    0.817    0.414    0.363    0.111
 .y6 ~~
   .y8                1.372    0.577    2.378    0.017    1.372    0.338

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .x1                0.081    0.019    4.182    0.000    0.081    0.154
   .x2                0.120    0.070    1.729    0.084    0.120    0.053
   .x3                0.467    0.090    5.177    0.000    0.467    0.239
   .y1                1.855    0.433    4.279    0.000    1.855    0.277
   .y2                7.581    1.366    5.549    0.000    7.581    0.525
   .y3                4.956    0.956    5.182    0.000    4.956    0.426
   .y4                3.225    0.723    4.458    0.000    3.225    0.298
   .y5                2.313    0.479    4.831    0.000    2.313    0.333
   .y6                4.968    0.921    5.393    0.000    4.968    0.430
   .y7                3.560    0.710    5.018    0.000    3.560    0.357
   .y8                3.308    0.704    4.701    0.000    3.308    0.313
    ind60             0.449    0.087    5.175    0.000    1.000    1.000
   .dem60             3.875    0.866    4.477    0.000    0.800    0.800
   .dem65             0.164    0.227    0.725    0.469    0.035    0.035
```

the statistical model used by SEM

- the structural part of the model is defined as:

$$\boldsymbol{\eta} = \boldsymbol{\alpha} + \mathbf{B}\,\boldsymbol{\eta} + \boldsymbol{\zeta}$$

  - $\boldsymbol{\eta}$ is an $M \times 1$ random vector of latent variables
  - $\boldsymbol{\alpha}$ is an $M \times 1$ vector of intercepts
  - $\mathbf{B}$ is an $M \times M$ matrix of coefficients for the regressions of the latent variables on each other
  - $\boldsymbol{\zeta}$ is an $M \times 1$ random vector of disturbance terms
  - in most settings, the diagonal elements of $\mathbf{B}$ are set to zero, and $(\mathbf{I} - \mathbf{B})$ is invertible by design

- to simplify the notation, we will use the convention that an observed variable involved in the structural part of the model is upgraded to a latent variable, with the observed variable as its only indicator with no measurement error

the statistical model used by SEM (2)

- to obtain the model–implied mean vector $\mathrm{E}(\boldsymbol{\eta})$ and variance–covariance matrix $\mathrm{Var}(\boldsymbol{\eta})$, we first rewrite the structural model in its so–called reduced form, where $\boldsymbol{\eta}$ only appears on the left–hand side of the equation:

$$\begin{aligned}
\boldsymbol{\eta} &= \boldsymbol{\alpha} + \mathbf{B}\,\boldsymbol{\eta} + \boldsymbol{\zeta} \\
\boldsymbol{\eta} - \mathbf{B}\,\boldsymbol{\eta} &= \boldsymbol{\alpha} + \boldsymbol{\zeta} \\
(\mathbf{I} - \mathbf{B})\,\boldsymbol{\eta} &= \boldsymbol{\alpha} + \boldsymbol{\zeta} \\
\boldsymbol{\eta} &= (\mathbf{I} - \mathbf{B})^{-1}\,(\boldsymbol{\alpha} + \boldsymbol{\zeta}) \\
&= (\mathbf{I} - \mathbf{B})^{-1}\boldsymbol{\alpha} + (\mathbf{I} - \mathbf{B})^{-1}\boldsymbol{\zeta}
\end{aligned}$$

- assuming $\mathrm{E}(\boldsymbol{\zeta}) = \mathbf{0}$ and writing $\mathrm{Var}(\boldsymbol{\zeta}) = \boldsymbol{\Psi}$, it follows that:

$$\begin{aligned}
\mathrm{E}(\boldsymbol{\eta}) &= (\mathbf{I} - \mathbf{B})^{-1}\boldsymbol{\alpha} \\
\mathrm{Var}(\boldsymbol{\eta}) &= (\mathbf{I} - \mathbf{B})^{-1}\boldsymbol{\Psi}\,(\mathbf{I} - \mathbf{B})^{-1\prime}
\end{aligned}$$

the statistical model used by SEM (3)

- the measurement part of the model is defined as:

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

  - $\boldsymbol{y}$ is a $P \times 1$ random vector of observed variables
  - $\boldsymbol{\nu}$ is a $P \times 1$ vector of intercepts
  - $\boldsymbol{\Lambda}$ is a $P \times M$ matrix of factor loadings relating the M latent variables to the P observed variables
  - $\boldsymbol{\epsilon}$ is a $P \times 1$ random vector of residual errors
  - we assume: $\mathrm{E}(\boldsymbol{\epsilon}) = \mathbf{0}$, $\mathrm{Cov}(\boldsymbol{\eta}, \boldsymbol{\epsilon}) = \mathbf{0}$, $\mathrm{Cov}(\boldsymbol{\epsilon}, \boldsymbol{\zeta}) = \mathbf{0}$, and we denote the variance–covariance matrix of $\boldsymbol{\epsilon}$ by $\boldsymbol{\Theta}$

- it follows that the model–implied mean vector $\boldsymbol{\mu} = \mathrm{E}(\boldsymbol{y})$ and variance–covariance matrix $\boldsymbol{\Sigma} = \mathrm{Var}(\boldsymbol{y})$ are given by:

$$\boldsymbol{\mu} = \boldsymbol{\nu} + \boldsymbol{\Lambda}\,\mathrm{E}(\boldsymbol{\eta})$$
$$\boldsymbol{\Sigma} = \boldsymbol{\Lambda}\,\mathrm{Var}(\boldsymbol{\eta})\,\boldsymbol{\Lambda}' + \boldsymbol{\Theta}$$

model parameters

- the model matrices are then $\boldsymbol{\alpha}$, $\mathbf{B}$, and $\boldsymbol{\Psi}$ for the structural part, and $\boldsymbol{\nu}$, $\boldsymbol{\Lambda}$, and $\boldsymbol{\Theta}$ for the measurement part
- a **structural model** can be defined by setting certain elements of these model matrices to a fixed constant (often zero or one), while allowing other elements to be free
- the T free elements are collected in a T-dimensional parameter vector $\boldsymbol{\theta}$
- two 'types' of free parameters:
  - parameters related to the measurement part (factor loadings, residual variances of the indicators)
  - parameters related to the structural part (regression coefficients, residual variances, ...)

the standard estimation approach in SEM

- for simplicity, assume all variables are continuous and data is complete
- by far the most used estimation method in SEM is maximum likelihood (ML) estimation, assuming multivariate normality
- alternative estimation methods: normal-theory generalized least-squares (GLS), asymptotically distribution free (ADF, WLS), unweighted least-squares (ULS)
- three aspects that all these estimators have in common:
  1. they rely on iterative optimization procedures
  2. although all these estimators are consistent, optimal statistical properties are attained only if all the assumptions hold, and the sample size is 'sufficiently' large
  3. all parameters (in both the measurement and structural parts) are estimated simultaneously ('system-wide')

the ML discrepancy function (optional)

- given a set of i.i.d. data vectors $\mathcal{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N\}$, ML estimation seeks those values of $\boldsymbol{\theta}$ that maximize the multivariate normal log-likelihood given by:

$$\begin{aligned}\text{logl}(\boldsymbol{\theta}|\mathcal{Y}) = &-\frac{NP}{2}\ln(2\pi) - \frac{N}{2}\ln|\boldsymbol{\Sigma}(\boldsymbol{\theta})| - \frac{N}{2}\text{tr}\left[\mathbf{S}\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}\right] \\ &- \frac{N}{2}[\bar{\boldsymbol{y}} - \boldsymbol{\mu}(\boldsymbol{\theta})]'\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}[\bar{\boldsymbol{y}} - \boldsymbol{\mu}(\boldsymbol{\theta})]\end{aligned}$$

  where $\bar{\boldsymbol{y}}$ is the sample mean vector, $\mathbf{S}$ is the (biased) sample variance–covariance matrix, and tr$[\cdot]$ is the trace operator that computes the sum of the diagonal elements of its matrix argument

- it is more convenient (but equivalent) to minimize:

$$\begin{aligned}F_{NML}(\boldsymbol{\theta}) = &\text{tr}\left[\mathbf{S}\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}\right] - \ln|\mathbf{S}\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}| - P \\ &+ [\bar{\boldsymbol{y}} - \boldsymbol{\mu}(\boldsymbol{\theta})]^T \boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1} [\bar{\boldsymbol{y}} - \boldsymbol{\mu}(\boldsymbol{\theta})]\end{aligned}$$

  which is called the normal theory-based ML discrepancy function

problems with the standard estimation approach in SEM

- because the optimization procedure is iterative, it is possible that no (local or global) solution is found at all: 'non-convergence' (often in small samples)
- distributional assumptions:
  - ML and GLS rely on normality and should be accompanied with 'robust' standard errors and test statistics
  - ADF/WLS needs a huge sample size (> 5000)
  - ULS is consistent, but less efficient and neither scale-free (=rescaling implies simple scaling factor change) nor scale-invariant (=same fit function after rescaling)
- they simultaneously estimate all the parameters (in both the measurement and structural parts): 'system-wide'
  - this only works well if the model is correctly specified
  - if one part of the model is misspecified, all parameters are affected, leading to biased estimates in other correctly specified parts of the model

solutions suggested in the literature

- the model-implied instrumental variable (MIIV) approach (Bollen, 1996): noniterative, equation-by-equation, and much more robust against local model misspecifications
- a similar procedure has been proposed by Burghgraeve et al. (2021)
- structural-after-measurement (SAM) approaches:
  - first estimate the measurement part, then (keeping the measurement part parameters fixed) estimate the structural part
  - early references: Burt (1973, 1976), Hunter & Gerbing (1982), Lance, Cornwell & Mulaik (1988)
  - standard approach in the 'measurement error models' literature (Fuller, 1987) (See also Wall & Amemiya, 2000, 2003)
  - standard approach in the analysis of large-scale assessment programs such as TIMSS and PISA
  - recent work: bias-corrected factor score regression (Devlieger et al., 2016) using Croon's correction

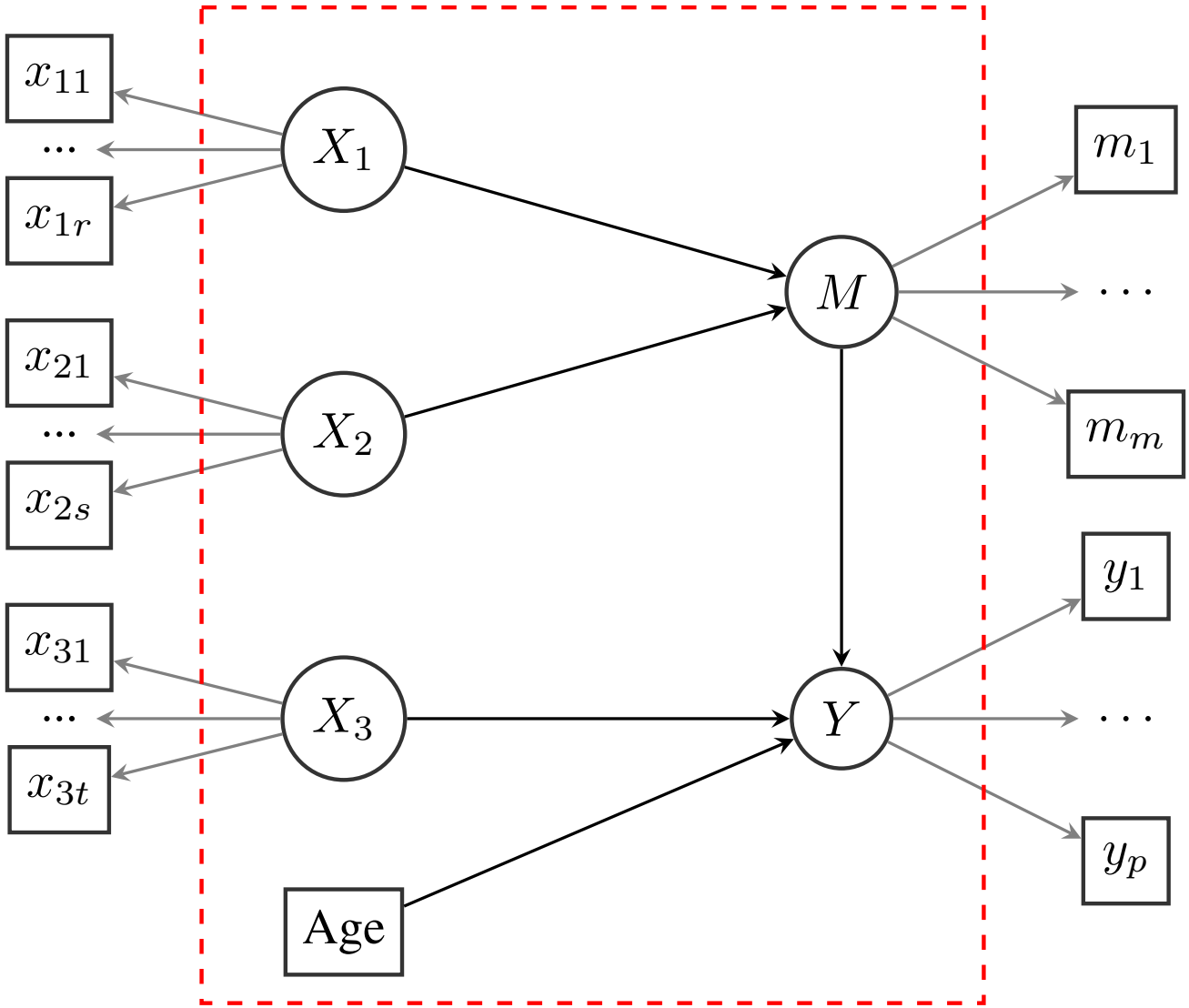
the SAM framework (Rosseel & Loh, 2022) (1)

- the SAM framework unifies and formalizes earlier ideas that we should decouple estimation for the measurement and the structural part of the model
- the 'SAM approach' implies a two-step approach:
  - in the first step: we estimate the parameters related to the measurement part (factor loadings, residual variance of the indicators)
  - in the second step: we estimate the parameters related to the structural part (regression coefficients, residual variances)
- two main approaches:
  1. local SAM: the first step creates the (sufficient) summary statistics (only) for the second step: an estimate of $\mathrm{E}(\boldsymbol{\eta})$ and $\mathrm{Var}(\boldsymbol{\eta})$
  2. global SAM: the point estimates of the first step are kept fixed while fitting the full model, estimating only the remaining parameters of the structural part (Burt, 1976)

the SAM framework (Rosseel & Loh, 2022) (2)

- the SAM framework includes two-step corrected standard errors, and permits computing both local and (pseudo) global fit measures
- note: the SAM approach is an **estimation strategy**, and should not be regarded as a model-building tool
  - SAM does not replace CFA: we 'assume' that the measurement models are well established and fit (reasonably) well
  - if there are any doubts about the measurement part, this should be addressed first
  - we assume that the focus of the analysis is on the structural part of the model
- the SAM framework is implemented in the `sam()` function of the lavaan package

structural versus measurement



local SAM: rationale

- 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)
- this results in estimates for $\boldsymbol{\nu}$, $\boldsymbol{\Lambda}$ and $\boldsymbol{\Theta}$, collected in $\boldsymbol{\theta}_1$
- M is the number of latent variables; B is the number of measurement 'blocks'
- three options:
  1. $B = 1$: single CFA (not recommended)
  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
- measurement models that are 'linked' (due to cross-loadings, correlated residuals, or equality constraints) should be treated 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:

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

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

- in local SAM, we proceed in the second stage with these 'sufficient statistics' only

local SAM: second stage

- second stage: $\widehat{\mathrm{E}(\boldsymbol{\eta})}$ and $\widehat{\mathrm{Var}(\boldsymbol{\eta})}$ are used to estimate $\boldsymbol{\alpha}$, $\mathbf{B}$ and $\boldsymbol{\Psi}$, collected in $\boldsymbol{\theta}_2$
- 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)

example: generate population data mediation model

```
> 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
+     X1 ~~ 0.4*X2; X1 ~~ -0.2*X3; X2 ~~ 0.4*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)
```

example: 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
+ '
> 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    2.308    0.021
    M                 0.400    0.079    5.078    0.000
    Age              -0.100    0.083   -1.203    0.229
  M ~
    X1               -0.300    0.133   -2.258    0.024
    X2                1.100    0.165    6.670    0.000
```

example: model-implied variance-covariance matrix latent variables

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

```
     Y      M      X1     X2     X3
Y   1.498
M   0.939  2.036
X1  0.006  0.140  1.000
X2  0.492  0.980  0.400  1.000
X3  0.450  0.500 -0.200  0.400  1.000
```

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     1.498 0.939  0.006 0.492  0.45 -0.1
M     0.939 2.036  0.140 0.980  0.50  0.0
X1    0.006 0.140  1.000 0.400 -0.20  0.0
X2    0.492 0.980  0.400 1.000  0.40  0.0
X3    0.450 0.500 -0.200 0.400  1.00  0.0
Age  -0.100 0.000  0.000 0.000  0.00  1.0
```

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    2.301    0.021
    M                 0.400    0.080    4.971    0.000
    Age              -0.100    0.083   -1.203    0.229
  M ~
    X1               -0.300    0.133   -2.251    0.024
    X2                1.100    0.176    6.235    0.000
```

example 2: misspecified structural part

- non-perfect data ($N = 200$)
- misspecified structural part: no direct effect of X3 on Y

```
> set.seed(1234)
> Data <- simulateData(pop.model, sample.nobs = 200L, empirical = FALSE)
> model <- '
+     # factor loadings
+     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 ~ M + Age # direct effect of X3 is missing
+     M ~ X1 + X2
+ '
> fit.lsam <- sam(model      = model,
+                 data       = Data,
+                 sam.method = "local",
+                 mm.args    = list(estimator = "ML"),
+                 struc.args = list(fixed.x = FALSE))
```

lavaan output

```
> summary(fit.lsam)

This is lavaan 0.6-12.1689 -- using the SAM approach to SEM

  SAM method                                        LOCAL
  Mapping matrix M method                              ML
  Number of measurement blocks                          5
  Estimator measurement part                           ML
  Estimator  structural part                           ML

  Number of observations                              200

Summary Information Measurement + Structural:

  Block Latent Nind Chisq Df
      1      Y    4 2.080  2
      2      M    4 3.762  2
      3     X1    4 0.283  2
      4     X2    4 3.284  2
      5     X3    4 0.188  2

  Model-based reliability latent variables:

      Y     M    X1    X2    X3 Age
  0.829 0.806 0.769 0.799 0.741   1
```

```
  Summary Information Structural part:

   chisq df pvalue   cfi rmsea  srmr
  10.134  6  0.119 0.989 0.059 0.043

Parameter Estimates:

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

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  Y ~
    M                 0.515    0.079    6.488    0.000
    Age              -0.156    0.080   -1.943    0.052
  M ~
    X1               -0.338    0.118   -2.864    0.004
    X2                1.051    0.134    7.822    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  X1 ~~
    X2                0.369    0.101    3.635    0.000
    X3               -0.261    0.092   -2.841    0.004
  X2 ~~
    X3                0.387    0.108    3.588    0.000
```

```
Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .Y                 0.870    0.161    5.413    0.000
   .M                 0.608    0.156    3.900    0.000
    X1                0.848    0.181    4.684    0.000
    X2                1.169    0.211    5.527    0.000
    X3                0.962    0.209    4.610    0.000
```

advantages of (local) SAM

- simulation studies show three key advantages:
  - estimates are more robust against local model misspecifications
  - estimation routines are less vulnerable to convergence issues in small samples
  - estimates exhibit smaller finite sample biases under correctly specified models (in particular when the sample size is small, and the items have low reliability)
- in addition:
  - decoupling measurement and structural part: different (noniterative) estimators can be used for each (sub)model
  - extends to the multigroup and multilevel setting
  - twostep-corrected standard errors, local and global fit measures are available

(current) limitations

- no support (yet) for higher-order factor models: all 'indicators' of latent variables must be observed
- the 'Lambda' matrix of factor loadings cannot be rank deficient
- Var($\boldsymbol{\eta}$) must be unrestricted (no apriori zeroes/constants)
- more research is needed to study:
  - compare the different mapping matrices (and explore alternatives)
  - best approach to handle missing data
  - performance when observed indicators are categorical
  - compare to equation-by-equation approaches (MIIV, James-Stein, ...)

special cases of local SAM

- sum scores (biased)
- sum scores + reliability (= single-indicator models)
- factor scores (biased)
- factor scores + Croon's correction
- (linear) measurement error models (Fuller, 1987)
- two-stage method-of-moments estimator (Wall & Amemiya, 2000)
- ...

last slide

- we advocate the 'structural-after-measurement' (SAM) approach as an estimation strategy in structural equation modeling
  - works equally well in 'ideal' settings (correct model, large sample)
  - works better in more 'realistic' settings (small local misspecifications, small-to-medium sample sizes)
  - for each submodel, we can use a different (consistent) possible noniterative estimator
  - many previous solutions (e.g., Croon's correction) turn out to be a special case
- future work:
  - extension to $\mathsf{Var}(\boldsymbol{\eta} \otimes \boldsymbol{\eta})$ (Elissa Burghgraeve)
  - small-sample corrections for SAM (Jasper Bogaert)
  - informative hypothesis testing in small samples (Caroline Keck)
  - noniterative estimators (Sara Dhaene)

Thank you!

(questions?)

`https://lavaan.org`