Full Structural Equation Modeling

Theory Construction and Statistical Modeling

Kyle M. Lang

Department of Methodology & Statistics
Utrecht University

Outline

Structural Equation Modeling

- Measurement Model
- Structural Model

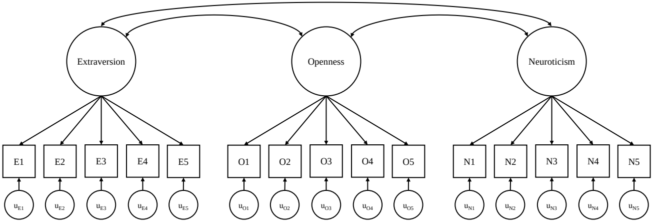
Mediation with SEM

Full SEM

A full structural equation model (SEM) simply combines path analysis and CFA.

- SEM allows us to model complicated structural relations among latent variables.

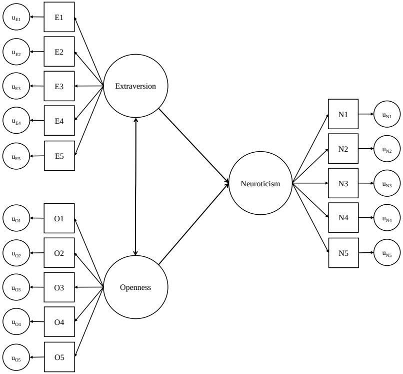
Let's consider a simple, three-factor CFA model.



CFA → SEM

We first evaluate the validity of the measurement model via CFA.

- We then convert the CFA to an SEM by converting some covariances to latent regression paths.



CFA Example

```r
## Load the lavaan package and some data:
library(lavaan)
data(bfi, package = "psych")

## Specify the CFA model:
cfaMod <- '
extra =~ E1 + E2 + E3 + E4 + E5
open  =~ O1 + O2 + O3 + O4 + O5
neuro =~ N1 + N2 + N3 + N4 + N5
'

## Estimate the model:
cfaOut <- cfa(cfaMod, data = bfi, missing = "fiml", std.lv = TRUE)

## Check the fit:
fitMeasures(cfaOut,
            c("chisq", "df", "pvalue", "cfi", "tli", "rmsea", "srmr")
            )

   chisq       df   pvalue      cfi      tli    rmsea     srmr
2251.679   87.000    0.000    0.809    0.769    0.094    0.081
```

CFA Example

```
partSummary(cfaOut, 7)

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  extra =~
    E1                0.973    0.032   30.607    0.000
    E2                1.163    0.030   38.171    0.000
    E3               -0.815    0.027  -30.358    0.000
    E4               -0.979    0.028  -35.254    0.000
    E5               -0.714    0.027  -26.638    0.000
  open =~
    O1                0.630    0.025   24.886    0.000
    O2               -0.605    0.036  -16.781    0.000
    O3                0.897    0.029   30.765    0.000
    O4                0.290    0.028   10.402    0.000
    O5               -0.602    0.031  -19.734    0.000
  neuro =~
    N1                1.272    0.027   47.254    0.000
    N2                1.218    0.026   46.491    0.000
    N3                1.157    0.029   40.195    0.000
    N4                0.892    0.030   29.356    0.000
    N5                0.823    0.031   26.163    0.000
```

CFA Example

```
partSummary(cfaOut, 8)

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  extra ~~
    open             -0.444    0.024  -18.472    0.000
    neuro             0.240    0.023   10.551    0.000
  open ~~
    neuro            -0.117    0.025   -4.667    0.000
```

CFA Example

```
partSummary(cfaOut, 9)

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .E1                2.974    0.031   96.223    0.000
   .E2                3.143    0.030  103.424    0.000
   .E3                4.002    0.026  156.117    0.000
   .E4                4.421    0.028  160.350    0.000
   .E5                4.417    0.025  174.595    0.000
   .O1                4.816    0.021  224.964    0.000
   .O2                2.713    0.030   91.745    0.000
   .O3                4.436    0.023  191.555    0.000
   .O4                4.892    0.023  211.519    0.000
   .O5                2.490    0.025   98.932    0.000
   .N1                2.932    0.030   98.589    0.000
   .N2                3.508    0.029  121.459    0.000
   .N3                3.217    0.030  106.147    0.000
   .N4                3.185    0.030  106.894    0.000
   .N5                2.969    0.031   96.663    0.000
    extra             0.000
    open              0.000
    neuro             0.000
```

CFA Example

```
partSummary(cfaOut, 10)

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .E1                1.713    0.054   31.442    0.000
   .E2                1.224    0.049   24.952    0.000
   .E3                1.163    0.038   30.388    0.000
   .E4                1.166    0.041   28.522    0.000
   .E5                1.272    0.039   32.789    0.000
   .O1                0.878    0.031   28.320    0.000
   .O2                2.083    0.062   33.705    0.000
   .O3                0.686    0.043   16.130    0.000
   .O4                1.407    0.039   36.236    0.000
   .O5                1.401    0.044   31.837    0.000
   .N1                0.848    0.037   23.029    0.000
   .N2                0.842    0.035   24.184    0.000
   .N3                1.228    0.043   28.308    0.000
   .N4                1.666    0.051   32.808    0.000
   .N5                1.942    0.056   34.465    0.000
    extra             1.000
    open              1.000
    neuro             1.000
```

SEM Example

```r
## Add structural paths:
semMod <- '
extra =~ E1 + E2 + E3 + E4 + E5
open  =~ O1 + O2 + O3 + O4 + O5
neuro =~ N1 + N2 + N3 + N4 + N5

neuro ~ extra + open
'

## Estimate the model:
semOut <- sem(semMod, data = bfi, missing = "fiml", std.lv = TRUE)

## Check the fit:
fitMeasures(semOut,
            c("chisq", "df", "pvalue", "cfi", "tli", "rmsea", "srmr")
            )

   chisq       df   pvalue      cfi      tli    rmsea     srmr 
2251.679   87.000    0.000    0.809    0.769    0.094    0.081 
```

SEM Example

```
partSummary(semOut, 7)

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  extra =~
    E1                0.973    0.032   30.607    0.000
    E2                1.163    0.030   38.172    0.000
    E3               -0.815    0.027  -30.358    0.000
    E4               -0.979    0.028  -35.254    0.000
    E5               -0.714    0.027  -26.638    0.000
  open =~
    O1                0.630    0.025   24.886    0.000
    O2               -0.605    0.036  -16.781    0.000
    O3                0.897    0.029   30.765    0.000
    O4                0.290    0.028   10.402    0.000
    O5               -0.602    0.031  -19.734    0.000
  neuro =~
    N1                1.235    0.027   45.916    0.000
    N2                1.183    0.026   45.360    0.000
    N3                1.123    0.028   39.976    0.000
    N4                0.866    0.029   29.605    0.000
    N5                0.799    0.031   26.204    0.000
```

SEM Example

```
partSummary(semOut, 8:9)

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  neuro ~
    extra             0.241    0.030    8.169    0.000
    open             -0.014    0.031   -0.448    0.654

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  extra ~~
    open             -0.444    0.024  -18.472    0.000
```

SEM Example

```
partSummary(semOut, 10)

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .E1                2.974    0.031   96.223    0.000
   .E2                3.143    0.030  103.424    0.000
   .E3                4.002    0.026  156.117    0.000
   .E4                4.421    0.028  160.350    0.000
   .E5                4.417    0.025  174.595    0.000
   .O1                4.816    0.021  224.964    0.000
   .O2                2.713    0.030   91.745    0.000
   .O3                4.436    0.023  191.555    0.000
   .O4                4.892    0.023  211.520    0.000
   .O5                2.490    0.025   98.932    0.000
   .N1                2.932    0.030   98.589    0.000
   .N2                3.508    0.029  121.459    0.000
   .N3                3.217    0.030  106.146    0.000
   .N4                3.185    0.030  106.894    0.000
   .N5                2.969    0.031   96.663    0.000
    extra             0.000
    open              0.000
   .neuro             0.000
```

SEM Example

```
partSummary(semOut, 11)

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .E1                1.713    0.054   31.442    0.000
   .E2                1.224    0.049   24.952    0.000
   .E3                1.163    0.038   30.388    0.000
   .E4                1.166    0.041   28.522    0.000
   .E5                1.272    0.039   32.789    0.000
   .O1                0.878    0.031   28.320    0.000
   .O2                2.083    0.062   33.705    0.000
   .O3                0.686    0.043   16.130    0.000
   .O4                1.407    0.039   36.236    0.000
   .O5                1.401    0.044   31.837    0.000
   .N1                0.848    0.037   23.029    0.000
   .N2                0.842    0.035   24.184    0.000
   .N3                1.228    0.043   28.308    0.000
   .N4                1.666    0.051   32.808    0.000
   .N5                1.942    0.056   34.465    0.000
    extra             1.000
    open              1.000
   .neuro             1.000
```

Why SEM?

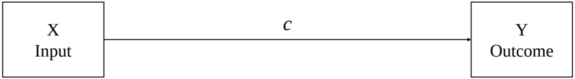
The beauty of SEM is that we get to model the types of complex relations we can specify via path models while leveraging all the strengths of latent variables.

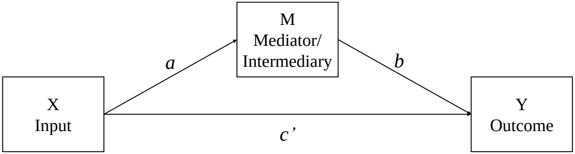
- Multiple-group SEM models moderation by group.
  - The latent variables give us the ability to evaluate measurement invariance across groups.
  - We'll see more of these ideas in the next lecture.
- Path analysis and SEM lend themselves especially well to mediation analysis and conditional process analysis.

MEDIATION WITH SEM

Boring Model

In Week 3, all of our models have looked something like:

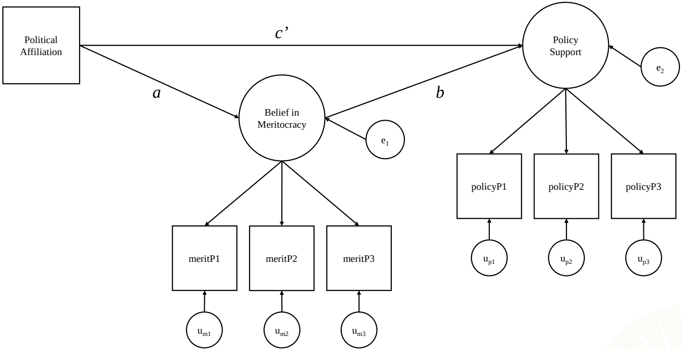




But there is no reason that we need to restrict ourselves to mucking around with observed variables.

Better Model

We can (and should) test for indirect effects using full SEMs such as:



Measurement error can be a big problem for mediation analysis, so latent variable modeling is highly recommended.

Example

```r
dat1 <- readRDS("../data/adamsKlpsData.rds") %>% select(-merit, -policy)

## Specify the CFA model:
mod5.1 <- '
merit  =~ meritP1 + meritP2 + meritP3
policy =~ policyP1 + policyP2 + policyP3
'

## Fit the CFA and check model:
out5.1 <- cfa(mod5.1, data = dat1, std.lv = TRUE)

## Check model fit:
fitMeasures(out5.1,
            c("chisq", "df", "pvalue", "cfi", "tli", "rmsea", "srmr")
            )

 chisq      df  pvalue    cfi    tli  rmsea   srmr
16.869   8.000   0.031  0.922  0.853  0.113  0.065
```

Example

```
partSummary(out5.1, 7)

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  merit =~
    meritP1           0.690    0.134    5.155    0.000
    meritP2           0.968    0.142    6.830    0.000
    meritP3           0.748    0.137    5.458    0.000
  policy =~
    policyP1          0.851    0.186    4.570    0.000
    policyP2          0.996    0.167    5.967    0.000
    policyP3          1.121    0.177    6.339    0.000
```

Example

```
partSummary(out5.1, 8:9)

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  merit ~~
    policy           -0.336    0.131   -2.563    0.010

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .meritP1           0.865    0.165    5.248    0.000
   .meritP2           0.445    0.201    2.211    0.027
   .meritP3           0.833    0.172    4.857    0.000
   .policyP1          1.836    0.324    5.671    0.000
   .policyP2          0.942    0.256    3.683    0.000
   .policyP3          0.857    0.297    2.882    0.004
    merit             1.000
    policy            1.000
```

Example

```r
## Specify the structural model:
mod5.2 <- '
merit  =~ meritP1 + meritP2 + meritP3
policy =~ policyP1 + policyP2 + policyP3

policy ~ b*merit + polAffil
merit  ~ a*polAffil

ab := a*b
'

## Fit the structural model and test the indirect effect:
out5.2 <-
    sem(mod5.2, data = dat1, std.lv = TRUE, se = "boot", bootstrap = 500)
```

Example

```
partSummary(out5.2, 7:8)

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  merit =~
    meritP1           0.545    0.132    4.136    0.000
    meritP2           0.858    0.133    6.452    0.000
    meritP3           0.609    0.128    4.776    0.000
  policy =~
    policyP1          0.799    0.186    4.303    0.000
    policyP2          0.924    0.161    5.749    0.000
    policyP3          1.001    0.277    3.614    0.000

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  policy ~
    merit      (b)   -0.195    0.203   -0.962    0.336
    polAffil          0.169    0.141    1.203    0.229
  merit ~
    polAffil   (a)   -0.411    0.100   -4.093    0.000
```

Example

```
partSummary(out5.2, 9:10)

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .meritP1           0.922    0.175    5.262    0.000
   .meritP2           0.341    0.213    1.599    0.110
   .meritP3           0.869    0.182    4.783    0.000
   .policyP1          1.801    0.325    5.548    0.000
   .policyP2          0.918    0.304    3.023    0.003
   .policyP3          0.922    0.598    1.542    0.123
   .merit             1.000
   .policy            1.000

Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)
    ab                0.080    0.092    0.872    0.383
```

Example

```r
parameterEstimates(out5.2, boot.ci.type = "bca.simple") %>%
    select(c("label", "est", "ci.lower", "ci.upper")) %>%
    filter(label != "")

  label    est ci.lower ci.upper
1     b -0.195   -0.667    0.170
2     a -0.411   -0.637   -0.241
3    ab  0.080   -0.074    0.293
```