Software for mediation analysis

Yves Rosseel
Department of Data Analysis
Ghent University – Belgium

Symposium on Causal Mediation Analysis
January 28–29, 2013 – Ghent University

Software for mediation analysis – two traditions

- traditional software for mediation analysis
  - Baron and Kenny (1986) tradition
  - many applied researchers still follow these steps
  - using SPSS/SAS, often in combination with macros/scripts
  - modern approach: using SEM software
  - psychologists are very familiar with this approach
- modern software for mediation analysis:
  - based on the causal inference literature
  - custom macros/code available for SPSS, SAS, Stata, R, ...
  - psychologists are very unfamiliar with this approach
    (and the accompanying software)

Some links and resources

- traditional approaches (mostly non-SEM)
  - Andrew F. Hayes website ('My Macros and Code for SPSS and SAS'): http://www.afhayes.com/
  - Kristopher J. Preacher
    http://quantpsy.org/medn.htm
  - David A. Kenny
    http://davidakenny.net
- approaches based on the counterfactual framework
  - Valeri, L. and VanderWeele, T.J. (in press). Mediation analysis allowing for exposure-mediator interactions and causal interpretation: theoretical assumptions and implementation with SAS and SPSS macros. *Psychological Methods*
  - Stata: command paramed (Richard Emsley and Hanhua Liu), ...
  - R: package 'mediation'

Software for SEM

Commercial – closed-source

- LISREL, EQS, AMOS, MPLUS
- SAS/Stat: proc CALIS, proc TCALIS
- SEPATH (Statistica), RAMONA (Systat), Stata 12
- Mx (free, closed-source)

Non-commercial – open-source

- outside the R ecosystem: gllamm (Stata), . . .
- R packages:
  - sem
  - OpenMx
  - lavaan

What is lavaan?

- **lavaan** is an R package for latent variable analysis:
  - confirmatory factor analysis: function `cfa()`
  - structural equation modeling: function `sem()`
  - latent curve analysis / growth modeling: function `growth()`
  - general mean/covariance structure modeling: function `lavaan()`
  - support for continuous, binary and ordinal data
- under development, future plans:
  - multilevel SEM, mixture/latent-class SEM, Bayesian SEM
- the long-term goal of **lavaan** is
  1. to implement all the state-of-the-art capabilities that are currently available in commercial packages
  2. to provide a modular and extensible platform that allows for easy implementation and testing of new statistical and modeling ideas

Installing lavaan, finding documentation

- **lavaan** depends on the R project for statistical computing:

  `http://www.r-project.org`

- to install **lavaan**, simply start up an R session and type:

```
> install.packages("lavaan")
```

- more information about **lavaan**:

  `http://lavaan.org`

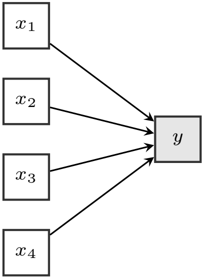
- the lavaan paper:

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

- **lavaan** development:

  `https://github.com/yrosseel/lavaan`

the lavaan model syntax – a simple regression




```
library(lavaan)
myData <- read.csv("c:/temp/myData.csv")

myModel <- ' y ~ x1 + x2 + x3 + x4 '

# fit model
fit <- sem(model = myModel,
           data  = myData)

# show results
summary(fit)
```

The standard linear model:

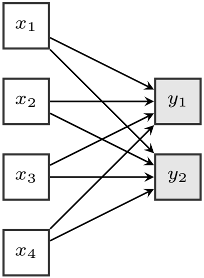
- $y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + \epsilon_i \quad (i = 1, 2, \ldots, n)$
- to ‘see’ the intercept, use

```
fit <- sem(model=myModel, data=myData, meanstructure=TRUE)
```

output (artificial data, N=100)

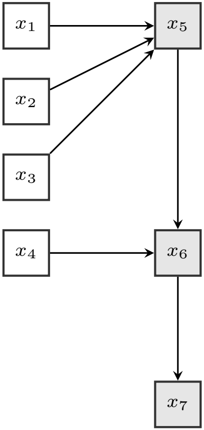
```
lavaan (0.5-12) converged normally after   1 iterations

  Number of observations                           100

  Estimator                                         ML
  Minimum Function Test Statistic                0.000
  Degrees of freedom                                 0
  P-value (Chi-square)                           1.000

Parameter estimates:

  Information                                 Expected
  Standard Errors                             Standard

                   Estimate  Std.err  Z-value  P(>|z|)
Regressions:
  y ~
    x1                5.047    0.048  105.425    0.000
    x2               -2.070    0.048  -43.198    0.000
    x3               10.029    0.049  204.811    0.000
    x4                0.119    0.052    2.308    0.021

Variances:
    y                 0.240    0.034
```

the lavaan model syntax – multivariate regression



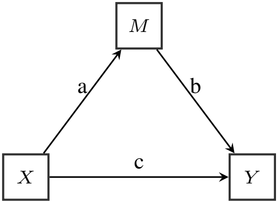

```
myModel <- ' y1 ~ x1 + x2 + x3 + x4
             y2 ~ x1 + x2 + x3 + x4 '
```

the lavaan model syntax – path analysis




```
myModel <- ' x5 ~ x1 + x2 + x3
             x6 ~ x4 + x5
             x7 ~ x6          '
```

Simple mediation example

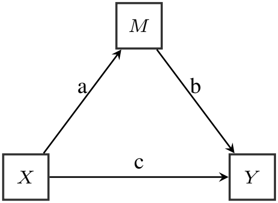



```
model <- '
           Y ~ M + X
           M ~ X
         '

fit <- sem(model, data=myData)
summary(fit)
```

output

```
...

Parameter estimates:

  Information                                    Expected
  Standard Errors                                Standard

                   Estimate  Std.err  Z-value  P(>|z|)
Regressions:
  Y ~
    M                 0.597    0.102    5.880    0.000
    X                 2.594    1.080    2.403    0.016
  M ~
    X                 2.739    1.027    2.666    0.008

Variances:
    Y               108.700   15.372
    M               105.408   14.907
```

Simple mediation example – add labels

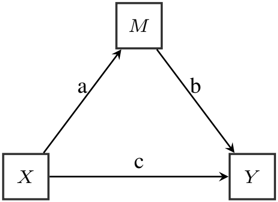



```
model <- '
          Y ~ b*M + c*X
          M ~ a*X
        '

fit <- sem(model, data=myData)
summary(fit)
```

output

```
...

Parameter estimates:

  Information                                    Expected
  Standard Errors                                Standard

                   Estimate  Std.err  Z-value  P(>|z|)
Regressions:
  Y ~
    M          (b)    0.597    0.102    5.880    0.000
    X          (c)    2.594    1.080    2.403    0.016
  M ~
    X          (a)    2.739    1.027    2.666    0.008

Variances:
    Y               108.700   15.372
    M               105.408   14.907
```

Simple mediation example – define indirect/total effect

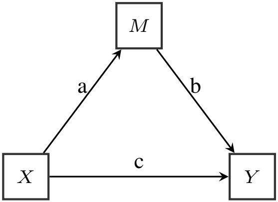



```
model <- '
          Y ~ b*M + c*X
          M ~ a*X

          indirect := a*b
          total    := c + (a*b)
        '

fit <- sem(model, data=myData)
summary(fit)
```

output

```
...

Parameter estimates:

  Information                                    Expected
  Standard Errors                                Standard

                    Estimate  Std.err  Z-value  P(>|z|)
Regressions:
  Y ~
    M         (b)      0.597    0.102    5.880    0.000
    X         (c)      2.594    1.080    2.403    0.016
  M ~
    X         (a)      2.739    1.027    2.666    0.008

Variances:
    Y                108.700   15.372
    M                105.408   14.907

Defined parameters:
    indirect           1.636    0.674    2.428    0.015
    total              4.230    1.210    3.495    0.000
```

Simple mediation example – bootstrapping

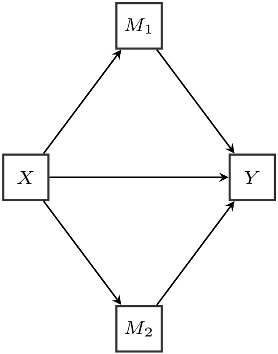



```
model <- '
          Y ~ b*M + c*X
          M ~ a*X

          indirect := a*b
          total    := c + (a*b)
        '

fit <- sem(model, data=myData,
          se="bootstrap")
summary(fit)
```

output

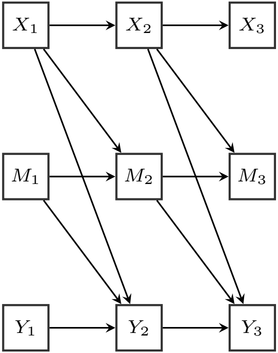
```
...

Parameter estimates:

  Information                                    Observed
  Standard Errors                               Bootstrap
  Number of requested bootstrap draws                1000
  Number of successful bootstrap draws               1000

                   Estimate  Std.err  Z-value  P(>|z|)
Regressions:
  Y ~
    M          (b)    0.597    0.098    6.068    0.000
    X          (c)    2.594    1.210    2.145    0.032
  M ~
    X          (a)    2.739    0.999    2.741    0.006

Variances:
    Y               108.700   17.747
    M               105.408   16.556

Defined parameters:
    indirect          1.636    0.645    2.535    0.011
    total             4.230    1.383    3.059    0.002
```

Multiple mediator model



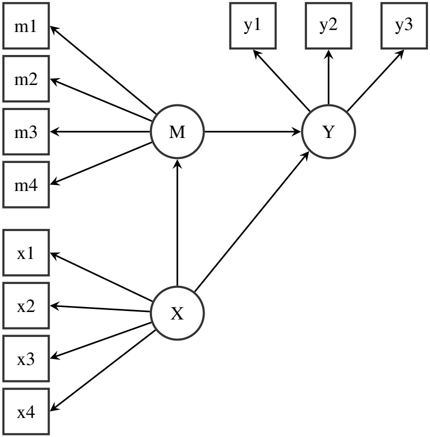

```
model <- '
            Y  ~ M1 + M2 + X
            M1 ~ X
            M2 ~ X
         '

fit <- sem(model, data=myData,
           se="bootstrap")
summary(fit)
```

Longitudinal mediation models



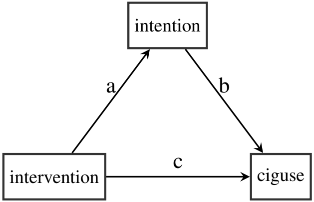

```
model <- '
           X2 ~ X1
           X3 ~ X2
           M2 ~ X1 + M1
           M3 ~ X2 + M2
           Y2 ~ X1 + M1 + Y1
           Y3 ~ X2 + M2 + Y2
         '

fit <- sem(model, data=myData,
           se="bootstrap")
summary(fit)
```

Mediation with latent variables




```
model <- '
  # latent variable definitions
    X =~ x1 + x2 + x3 + x4
    M =~ m1 + m2 + m3 + m4
    Y =~ y1 + y2 + y3

  # regressions
    M ~ X
    Y ~ X + M
'

fit <- sem(model, data=...)
```

Mediation with an ordinal mediation, and a binary outcome




```
model <- '
          ciguse    ~ c*intervention +
                      b*intention
          intention ~ a*intervention

          naive.indirect := a*b
          naive.direct   := c
         '

fit <- sem(model, data=myData,
           ordered=c("ciguse","intention"))
summary(fit)
```

Muthén, B. Applications of Causally Defined Direct and Indirect Effects in Mediation Analysis using SEM in Mplus. (retrieved from `www.statmodel.com`)

MacKinnon, D. P., Lockwood, C. M., Brown, C.H., and Hoffman, J. M. (2007). The intermediate endpoint effect in logistic and probit regression. *Clinical Trials*, 4, 499 - 513.

output

```
Parameter estimates:

  Information                                     Expected
  Standard Errors                               Robust.sem

                   Estimate  Std.err  Z-value  P(>|z|)
Regressions:
  ciguse ~
    tx          (c)    -0.130    0.093   -1.402    0.161
    intent      (b)     0.631    0.042   15.105    0.000
  intent ~
    tx          (a)    -0.246    0.089   -2.758    0.006

Thresholds:
    cigus|t1            0.760    0.072   10.491    0.000
    intnt|t1            0.525    0.067    7.844    0.000
    intnt|t2            0.970    0.071   13.572    0.000
    intnt|t3            1.378    0.082   16.710    0.000

Variances:
    ciguse              1.000
    intent              1.000

Defined parameters:
    naive.indirct      -0.155    0.057   -2.713    0.007
    naive.direct       -0.130    0.093   -1.402    0.161
```

Computing 'causal' direct and indirect effects

- Imai et al. (2010) showed in a simulation study that the naive approach (taking the product of coefficients) results in biased estimates of the direct and indirect effects
- But we can compute the correct estimates using the parameter values as computed in a SEM
- References:

Muthén, B. Applications of Causally Defined Direct and Indirect Effects in Mediation Analysis using SEM in Mplus.

`http://www.statmodel.com/examples/penn.shtml`

lavaan syntax - computing probits and probabilities

```
model <- '
          ciguse    ~ c*intervention +
                      b*intention
          intention ~ a*intervention

          # label threshold for ciguse
          ciguse | b0*t1

          probit11 := (-b0+c+b*a)/sqrt(b^2+1)
          probit10 := (-b0+c    )/sqrt(b^2+1)
          probit00 := (-b0      )/sqrt(b^2+1)

          indirect := pnorm(probit11) - pnorm(probit10)
          direct   := pnorm(probit10) - pnorm(probit00)

          OR.indirect := (pnorm(probit11)/(1-pnorm(probit11)))/
                         (pnorm(probit10)/(1-pnorm(probit10)))

          OR.direct   := (pnorm(probit10)/(1-pnorm(probit10)))/
                         (pnorm(probit00)/(1-pnorm(probit00)))
'
```

output

```
...

Defined parameters:

    probit11         -0.884    0.062  -14.182    0.000
    probit10         -0.752    0.070  -10.714    0.000
    probit00         -0.643    0.063  -10.184    0.000

    indirect         -0.037    0.014   -2.647    0.008
    direct           -0.034    0.024   -1.403    0.161

    OR.indirect       0.796    0.066   12.030    0.000
    OR.direct         0.829    0.111    7.450    0.000
```

The R package ‘mediation’

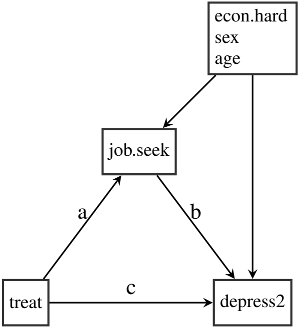
- References:

  Imai, K., Keele, L. and Tingley, D. (2010) A General Approach to Causal Mediation Analysis, *Psychological Methods*, 15(4), pp. 309-334.

  Tingley, D., Yamamoto, T., Keel, L. and Imai, K. (under review). *mediation: R Package for Causal Mediation Analysis*. `imai.princeton.edu/research/files/mediationR2.pdf`

- ‘General’ approach to causal mediation analysis, based on the counterfactual framework
- accomodates linear and nonlinear relationships, parametric and nonparametric models, continuous and discrete mediators, various types of outcomes
- allows for a sensitivity analysis

The jobs example - linear model for outcome and mediator



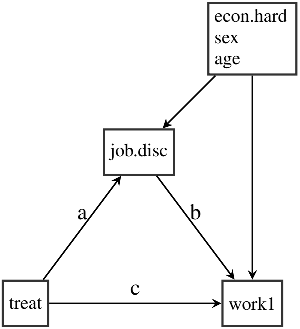
- `job_seek`: level of job-search self-efficacy
- `econ_hard`: level of economic hardship pre-treatment

```
# model for M
model.M <-
lm(job_seek ~ treat +
              econ_hard + sex + age,
              data=jobs)

# model for Y
model.Y <-
lm(depress2 ~ treat + job_seek +
              econ_hard + sex + age,
              data=jobs)

# Estimation via quasi-Bayesian approx.
out <- mediate(model.M,
               model.Y,
               treat    = "treat",
               mediator = "job_seek")
summary(out)
```

output

```
Causal Mediation Analysis

Quasi-Bayesian Confidence Intervals

                         Estimate 95% CI Lower 95% CI Upper p-value
Mediation Effect         -0.01600    -0.04191      0.00673    0.21
Direct Effect            -0.04091    -0.12296      0.04441    0.34
Total Effect             -0.05690    -0.14009      0.03313    0.22
Proportion via Mediation  0.23156    -1.70527      3.26928    0.44

Sample Size Used: 899


Simulations: 1000
```

The jobs example - binary outcome and ordered (K=4) mediator



- `job_disc`: The `job_seek` measure recoded into four categories from lowest to highest
- `work1`: Indicator variable for employment. 1 = employed

```
# model for M
model.M <-
polr(job_disc ~ treat +
                econ_hard + sex + age,
                data=jobs, method="probit",
                Hess=TRUE)

# model for Y
model.Y <-
glm(work1 ~ treat + job_disc +
            econ_hard + sex + age,
            data=jobs,
            family=binomial(link="probit"))

out <- mediate(model.M,
               model.Y,
               treat    = "treat",
               mediator = "job_disc")
summary(out)
```

Conclusions

- traditional software for mediation analysis: SEM
  - psychologists are very familiar with this approach
  - many results from the causal inference literature can be computed using SEM software (with care)
  - open-source software is available (R package 'lavaan')
- modern software for mediation analysis:
  - based on the causal inference literature
  - R package 'mediation'
  - sensitivity analysis
- towards an integration of both approaches in one single coherent software package?

Thank you!