Structural Equation Modeling

P.02 - Path Analysis

November 15, 2022 (12:09:40)

Lab description

The exercises for this lab are meant to help you understand how to conduct *path analysis* using the `lavaan` package in `R`. For this practical you will need two packages: `lavaan` and `semPlot`. You can install and load these packages using the following code:

```
# Install packages.
install.packages(c("lavaan", "semPlot"))

# Load the packages.
library(lavaan)
library(semPlot)
```

Lab exercises

Exercise 1

MacKinnon (2008, p. 113) provides a dataset from a hypothetical study of teacher expectancies and student achievement (sample size: $N = 40$). His path model is shown in Figure 1 and the covariances for the model are given in Figure 2. Your first task is to solve the exercise proposed by Beaujean (2014). More specifically you are asked to:

a. Input the covariances into `R`.
    - Hint: consider using the `lavaan` function `lav_matrix_lower2full` to do this.
b. Write the syntax for the model.
    - Hint: use the `:=` operator to define both indirect effects from *teacher expectancies* to *student achievement* ($a_1 \times b_1$ and $a_2 \times b_2$).
c. What are the indirect effects?

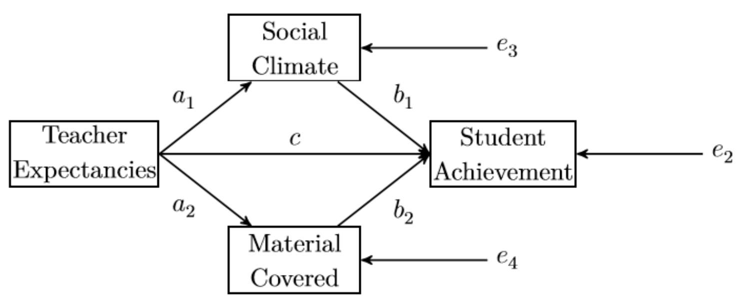


Figure 1: Path model

	Teacher Expectancies	Social Climate	Material Covered	Student Achievement
Teacher Expectancies	84.85	71.28	18.83	60.05
Social Climate	71.28	140.34	−6.25	84.54
Material Covered	18.83	−6.25	72.92	37.18
Student Achievement	60.05	84.54	37.18	139.48

Figure 2: Covariances between observed variables ($N = 40$)

Answers

We start by inputing the covariances in `R`.

```r
# Create the covariance matrix.
covariance_matrix <- lav_matrix_lower2full(
    c(84.85, 71.28, 140.34, 18.83, -6.25, 72.92, 60.05, 84.54, 37.18, 139.48)
)

# Write down the names of the variables.
#  - `TeachExp` stands for "teaching expectancies".
#  - `SocClim` stands for "social climate".
#  - `MatCov` stands for "material covered".
#  - `StAch` stands for "student achievement".
names <- c("TeachExp", "SocClim", "MatCov", "StAch")

# Add names to the rows and columns of the covariance matrix.
rownames(covariance_matrix) <- colnames(covariance_matrix) <- names

# Print the covariance matrix for inspection.
print(covariance_matrix)
```

```
##          TeachExp SocClim MatCov  StAch
## TeachExp    84.85   71.28  18.83  60.05
## SocClim     71.28  140.34  -6.25  84.54
## MatCov      18.83   -6.25  72.92  37.18
## StAch       60.05   84.54  37.18 139.48
```

Now that we have the covariance matrix, we can write down the model syntax in `lavaan`. Note that we need to use the `:=` operator, which as presented in the `lavaan` documentation:

> 'defines' new parameters which take on values that are an arbitrary function of the original model

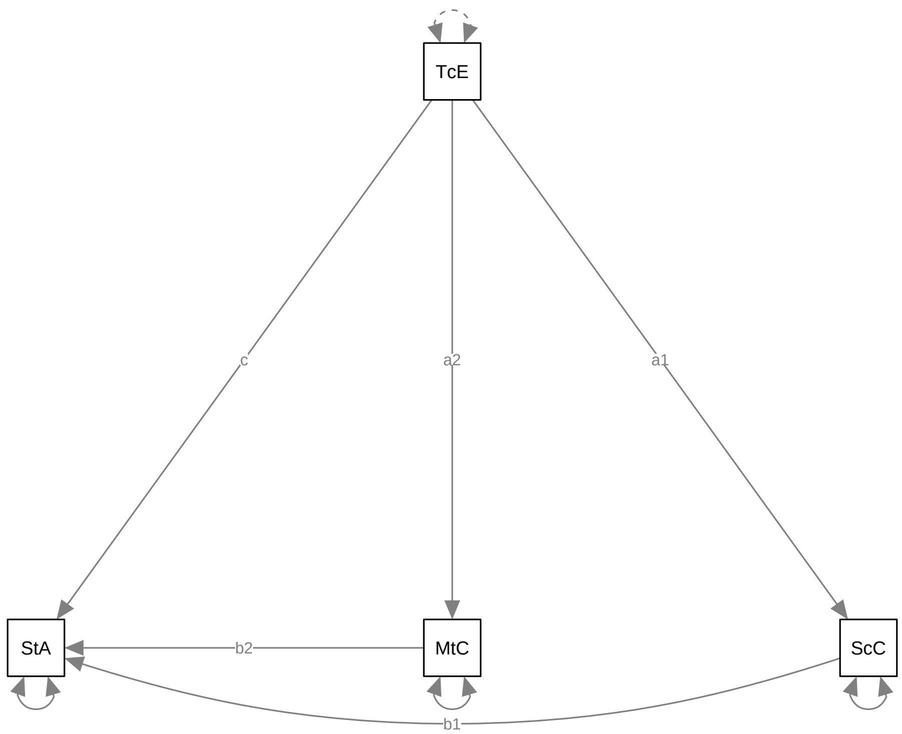
parameters. The function, however, must be specified in terms of the parameter labels that are explicitly mentioned in the model syntax.

```r
# The model syntax.
model_syntax <- '
    # Specify the regression equations and add labels.
    StAch ~ b1 * SocClim + b2 * MatCov + c * TeachExp
    MatCov ~ a2 * TeachExp
    SocClim ~ a1 * TeachExp

    # Specify the indirect effects.
    indirect1 := a1 * b1
    indirect2 := a2 * b2
'

# You can print the syntax to convince yourself that it's just text.
print(model_syntax)
```

```
## [1] "\n    # Specify the regression equations and add labels.\n    StAch ~ b1 * SocClim + b2 * MatCov + c * TeachExp\n    MatCov ~ a2 * TeachEx
```

And finally we can fit the model in order to obtain the parameter estimates.

```r
# Fit the model.
fit <- sem(model_syntax, sample.cov = covariance_matrix, sample.nobs = 40)

# Visualize the model.
semPaths(fit, what = "path")
```




```r
# Get fit summary.
summary(fit, standardized = TRUE)
```

```
## lavaan 0.6-12 ended normally after 1 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                         8
```

```
##
##   Number of observations                            40
##
## Model Test User Model:
##
##   Test statistic                                 3.687
##   Degrees of freedom                                 1
##   P-value (Chi-square)                           0.055
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   StAch ~
##     SocClim   (b1)    0.569    0.142    4.006    0.000    0.569    0.545
##     MatCov    (b2)    0.530    0.154    3.446    0.001    0.530    0.366
##     TeachExp   (c)    0.112    0.186    0.603    0.546    0.112    0.084
##   MatCov ~
##     TeachExp  (a2)    0.222    0.142    1.559    0.119    0.222    0.239
##   SocClim ~
##     TeachExp  (a1)    0.840    0.154    5.456    0.000    0.840    0.653
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .StAch            63.323   14.159    4.472    0.000   63.323    0.425
##    .MatCov           67.023   14.987    4.472    0.000   67.023    0.943
##    .SocClim          78.448   17.542    4.472    0.000   78.448    0.573
##
## Defined Parameters:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     indirect1         0.478    0.148    3.229    0.001    0.478    0.356
##     indirect2         0.118    0.083    1.421    0.155    0.118    0.088
```

The indirect effect through social climate is 0.478. The indirect effect through material covered is 0.118.

Exercise 2

In-class discussion of the code below and answer the following questions:

a. Is the model `multiple_mediation` just-identified, over-identified or under identified? Show calculations that proof your position.
b. How many degrees of freedom does the model `constrained_mediation` have? Motivate your answer.

The code is adapted from https://paolotoffanin.wordpress.com/2017/05/06/multiple-mediator-analysis-with-lavaan.

Answers

To answer **Exercise 2, point (a)** we need to look at the degrees of freedom (DF). We can, of course, take them directly from the `lavaan` summary output, but that wouldn't be any fun, so we calculate them ourselves based on the following straightforward the formula:

$$DF = \# \text{ parameters} - \# \text{ free parameters},$$

where the symbol # stands for 'the number of'.

We proceed by first determining how many parameters are in the model (i.e., how many elements are in the covariance matrix). We can calculate the number of elements in the covariance matrix as follows:

$$\# \text{ parameters} = \frac{p \times (p+1)}{2},$$

where p represents the number of variables in the model. In our case, this translates to:

$$\# \text{ parameters} = \frac{4 \times (4+1)}{2} = 10$$

Next, we determine how many free parameters (i.e., parameters that need to be estimated) are in the model:

- 1 variance, 3 error variances
- 1 covariance (i.e., between the mediators)
- 5 structural path coefficients

In total that gives us # free parameters = 10. Therefore, we have $DF = 10 - 10$, i.e., $DF = 0$. In this case, we say that the model `multiple_mediation` is just-identified as it contains 0 degrees of freedom.

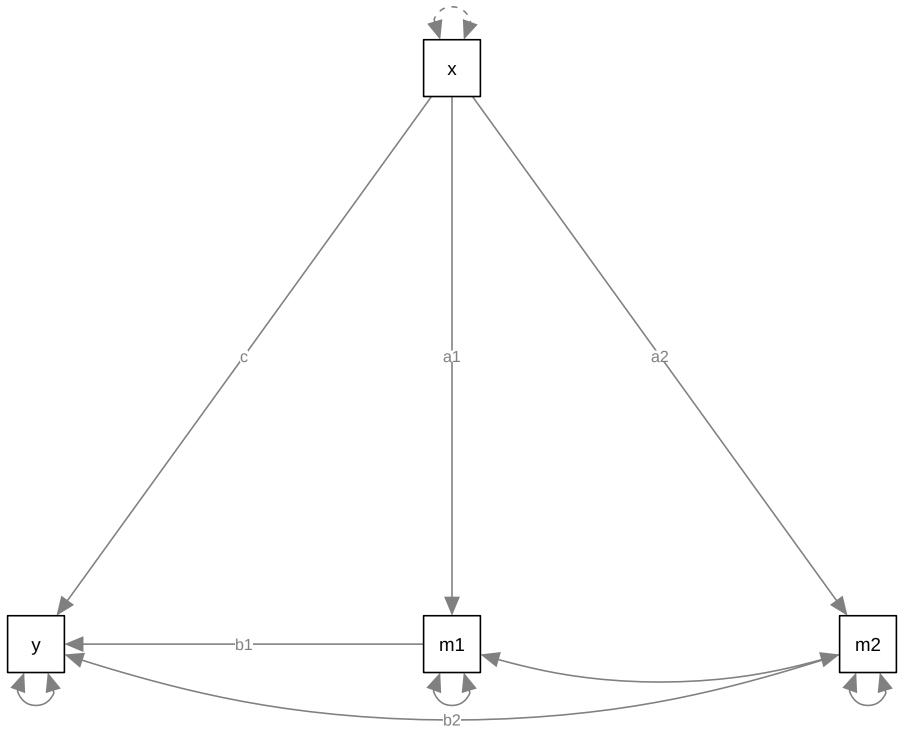
To answer **Exercise 2, point (b)** we need to understand that when adding a constraint we are asking `lavaan` to estimate one fewer parameter. Therefore, compared to the `multiple_mediation` model, the `constrained_mediation` model has 1 degrees of freedom (i.e., given by the constrain we set on the indirect effects).

Overview for the code below.

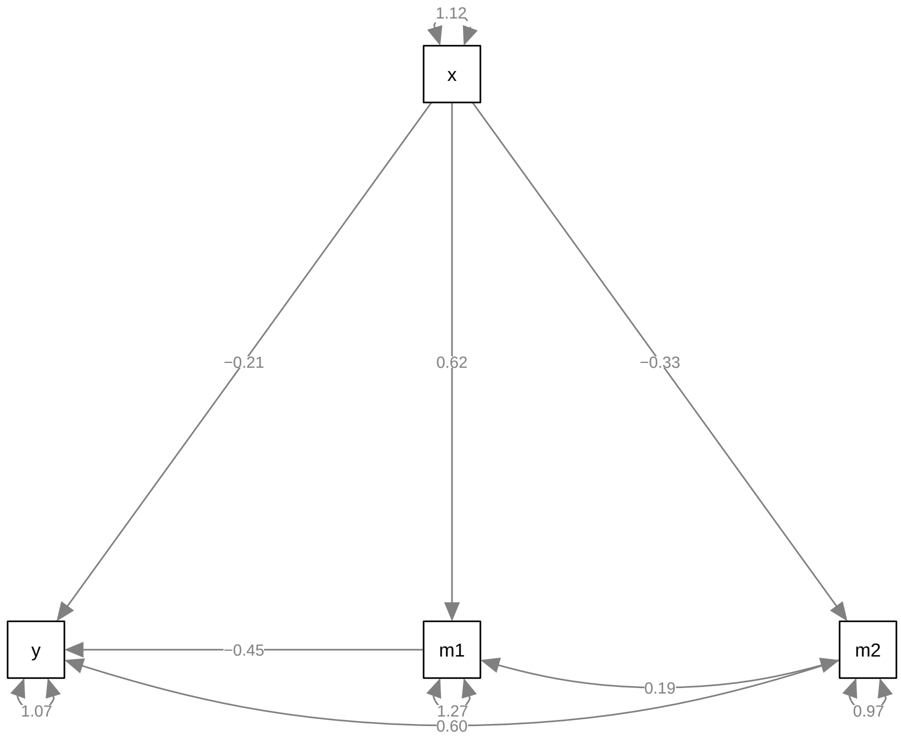
The code below is used to fit and visualize a multiple mediation model. Then, we pretend that we are interested in determining whether the two indirect effects are significantly different. We test this scenario in three different ways:

1. First, in the `contrast_mediation` model we estimate a `contrast` parameter that is defined as the difference between the two indirect effects and we check the p-value provided by `lavaan` for that parameter.
2. Then, we estimate a new model `constrained_mediation` where we constraint the indirect effects to be equal (i.e., our null hypothesis). To test this hypothesis, we perform a Likelihood Ration Test (LRT) by comparing the `constrained_mediation` with the `multiple_mediation` model for which the constraint is not applied.
3. Finally, we can also investigate the difference between these two parameters by using the same approach as in the `contrast_mediation` scenario, but this time instructing `lavaan` to construct the standard errors based on bootstrapping, instead of relying on an assumption of normality. Then, we can consult the confidence intervals provided. In order to obtain valid results, make sure you use a sufficiently large number of bootstraps (e.g., 2000 or more).

```r
# Set the seed to be able to replicate the results.
set.seed(03101972)

# Simulate data with two mediators.
x <- rnorm(100)
m1 <- 0.65 * x + rnorm(100)
m2 <- -0.40 * x + rnorm(100)
y <- 0.77 * m2 + 0.45 * -m1 + rnorm(100)

# Put the variables together in a data frame.
data <- data.frame(x = x, y = y, m1 = m1, m2 = m2)

# Model syntax for the multiple mediation model.
multiple_mediation <- '
    y ~ b1 * m1 + b2 * m2 + c * x
    m1 ~ a1 * x
    m2 ~ a2 * x

    # Allow for covariance between the mediators (i.e., as in Preacher and Hayes, 2008).
    m1 ~~ m2

    # Indirect effects.
    indirect1 := a1 * b1
    indirect2 := a2 * b2

    # Total effect.
    total := c + (a1 * b1) + (a2 * b2)
'

# Fit the model.
fit_mediation <- sem(model = multiple_mediation, data = data)

# Visualize the model.
semPaths(fit_mediation, what = "path", whatLabels = "label")
```



```r
# We can also see the values of the estimated parameters instead of the labels.
semPaths(fit_mediation, what = "path", whatLabels = "par")
```




```r
# Extract fit statistics.
summary(fit_mediation)
```

```
## lavaan 0.6-12 ended normally after 10 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                         9
##
##   Number of observations                           100
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 0
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   y ~
##     m1        (b1)   -0.450    0.093   -4.829    0.000
##     m2        (b2)    0.597    0.107    5.584    0.000
##     x          (c)   -0.209    0.122   -1.714    0.086
##   m1 ~
##     x         (a1)    0.618    0.107    5.803    0.000
##   m2 ~
##     x         (a2)   -0.326    0.093   -3.505    0.000
```

```
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##  .m1 ~~
##    .m2                0.189    0.112    1.685    0.092
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .y                 1.072    0.152    7.071    0.000
##    .m1                1.270    0.180    7.071    0.000
##    .m2                0.966    0.137    7.071    0.000
##
## Defined Parameters:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     indirect1        -0.278    0.075   -3.712    0.000
##     indirect2        -0.194    0.066   -2.968    0.003
##     total            -0.681    0.119   -5.748    0.000
```

Now include a contrast in the model to test the null hypothesis that the indirect effects are equal to each other.

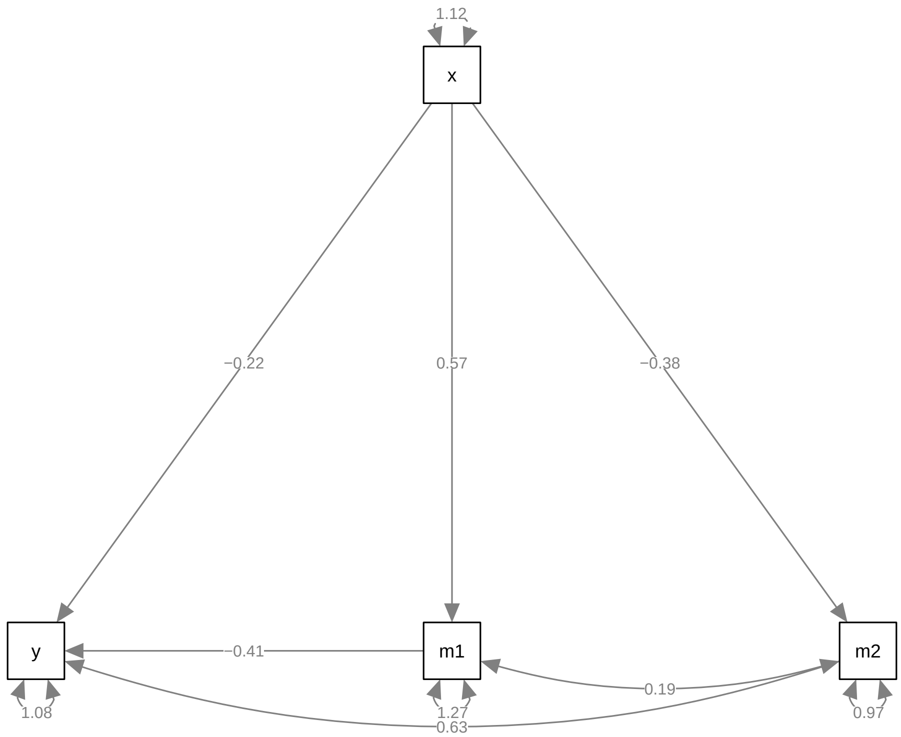
```r
# Model syntax for multiple mediation model with contrast.
contrast_mediation <- '
    y  ~ b1 * m1 + b2 * m2 + c * x
    m1 ~ a1 * x
    m2 ~ a2 * x

    # Allow for covariance between the mediators.
    m1 ~~ m2

    # Indirect effects.
    indirect1 := a1 * b1
    indirect2 := a2 * b2

    # Total effect.
    total := c + (a1 * b1) + (a2 * b2)

    # Contrast.
    contrast := indirect1 - indirect2
'

# Fit the model.
fit_contrast_mediation <- sem(model = contrast_mediation, data = data)

# Extract fit statistics.
summary(fit_contrast_mediation)
```

```
## lavaan 0.6-12 ended normally after 10 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                         9
##
##   Number of observations                           100
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 0
##
## Parameter Estimates:
```

```
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   y ~
##     m1        (b1)   -0.450    0.093   -4.829    0.000
##     m2        (b2)    0.597    0.107    5.584    0.000
##     x          (c)   -0.209    0.122   -1.714    0.086
##   m1 ~
##     x         (a1)    0.618    0.107    5.803    0.000
##   m2 ~
##     x         (a2)   -0.326    0.093   -3.505    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##  .m1 ~~
##    .m2                0.189    0.112    1.685    0.092
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .y                 1.072    0.152    7.071    0.000
##    .m1                1.270    0.180    7.071    0.000
##    .m2                0.966    0.137    7.071    0.000
##
## Defined Parameters:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     indirect1        -0.278    0.075   -3.712    0.000
##     indirect2        -0.194    0.066   -2.968    0.003
##     total            -0.681    0.119   -5.748    0.000
##     contrast         -0.084    0.101   -0.834    0.404
```

Finally, add a constraint in the multiple mediation model specifying the two indirect effect to be equal.

```r
constrained_mediation <- '
    y ~ b1 * m1 + b2 * m2 + c * x
    m1 ~ a1 * x
    m2 ~ a2 * x

    # Allow for covariance between the mediators.
    m1 ~~ m2

    # Indirect effects.
    indirect1 := a1 * b1
    indirect2 := a2 * b2

    # Total effect.
    total := c + (a1 * b1) + (a2 * b2)

    # Equality constraint.
    indirect1 == indirect2
'

# Fit the model.
fit_constrained_mediation <- sem(model = constrained_mediation, data = data)

# Visualize the model.
semPaths(fit_constrained_mediation, what = "path", whatLabels = "par")
```




```r
# Extract fit statistics and check that the constrain is satisfied.
summary(fit_constrained_mediation)
```

```
## lavaan 0.6-12 ended normally after 27 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                         9
##
##   Number of observations                           100
##
## Model Test User Model:
##
##   Test statistic                                 0.704
##   Degrees of freedom                                 1
##   P-value (Chi-square)                           0.401
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   y ~
##     m1        (b1)   -0.414    0.083   -5.006    0.000
##     m2        (b2)    0.625    0.102    6.155    0.000
##     x          (c)   -0.222    0.121   -1.833    0.067
##   m1 ~
##     x         (a1)    0.570    0.089    6.365    0.000
##   m2 ~
##     x         (a2)   -0.377    0.070   -5.397    0.000
##
## Covariances:
```

```
##                    Estimate  Std.Err  z-value  P(>|z|)
##  .m1 ~~
##    .m2               0.192    0.113    1.705    0.088
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .y                1.075    0.152    7.071    0.000
##    .m1               1.272    0.180    7.071    0.000
##    .m2               0.969    0.137    7.071    0.000
##
## Defined Parameters:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     indirect1        -0.236    0.049   -4.805    0.000
##     indirect2        -0.236    0.049   -4.805    0.000
##     total            -0.694    0.118   -5.894    0.000
##
## Constraints:
##                                               |Slack|
##     indirect1 - (indirect2)                    0.000
```

Test if the constrained model fits equally well as the model without the equality constraint using a Likelihood-Ratio Test (LRT). We can perform a LRT for two models fited with `lavaan` in `R` using the `anova` function.

```r
# Perform LRT.
anova(fit_mediation, fit_constrained_mediation)
```

```
## Chi-Squared Difference Test
##
##                           Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## fit_mediation              0 893.80 917.25 0.0000
## fit_constrained_mediation  1 892.51 913.35 0.7044    0.70441       1     0.4013
```

```r
# Note, you should use a sufficiently large number of bootstraps.
fit <- sem(
    model = contrast_mediation,
    data  = data,
    se = "bootstrap",
    bootstrap = 2000
)

# Extract information.
summary(
    fit, fit.measures = TRUE, standardize = TRUE,
    rsquare = TRUE, estimates = TRUE, ci = TRUE
)
```

```
## lavaan 0.6-12 ended normally after 10 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                         9
##
##   Number of observations                           100
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 0
##
## Model Test Baseline Model:
##
##   Test statistic                               110.408
```

```
##   Degrees of freedom                                 6
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    1.000
##   Tucker-Lewis Index (TLI)                       1.000
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)               -437.900
##   Loglikelihood unrestricted model (H1)       -437.900
##
##   Akaike (AIC)                                 893.801
##   Bayesian (BIC)                               917.247
##   Sample-size adjusted Bayesian (BIC)          888.823
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.000
##   90 Percent confidence interval - lower         0.000
##   90 Percent confidence interval - upper         0.000
##   P-value RMSEA <= 0.05                             NA
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.000
##
## Parameter Estimates:
##
##   Standard errors                            Bootstrap
##   Number of requested bootstrap draws             2000
##   Number of successful bootstrap draws            2000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##   y ~
##     m1        (b1)   -0.450    0.102   -4.404    0.000   -0.646   -0.248
##     m2        (b2)    0.597    0.118    5.043    0.000    0.379    0.841
##     x          (c)   -0.209    0.102   -2.054    0.040   -0.419   -0.007
##   m1 ~
##     x         (a1)    0.618    0.124    5.002    0.000    0.393    0.886
##   m2 ~
##     x         (a2)   -0.326    0.087   -3.747    0.000   -0.483   -0.144
##    Std.lv  Std.all
##
##    -0.450   -0.406
##     0.597    0.430
##    -0.209   -0.153
##
##     0.618    0.502
##
##    -0.326   -0.331
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##  .m1 ~~
##    .m2                0.189    0.103    1.833    0.067   -0.024    0.385
##    Std.lv  Std.all
##
##     0.189    0.171
```

```
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##    .y                 1.072    0.146    7.368    0.000    0.752    1.343
##    .m1                1.270    0.162    7.838    0.000    0.925    1.563
##    .m2                0.966    0.129    7.495    0.000    0.720    1.222
##    Std.lv  Std.all
##     1.072    0.513
##     1.270    0.748
##     0.966    0.891
##
## R-Square:
##                    Estimate
##     y                 0.487
##     m1                0.252
##     m2                0.109
##
## Defined Parameters:
##                    Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##     indirect1        -0.278    0.083   -3.357    0.001   -0.454   -0.131
##     indirect2        -0.194    0.063   -3.103    0.002   -0.327   -0.078
##     total            -0.681    0.113   -6.034    0.000   -0.910   -0.461
##     contrast         -0.084    0.108   -0.776    0.438   -0.312    0.117
##    Std.lv  Std.all
##    -0.278   -0.204
##    -0.194   -0.142
##    -0.681   -0.498
##    -0.084   -0.061
```

References

Beaujean, A. A. (2014). *Latent variable modeling using R: A step by step guide*. Routledge/Taylor & Francis Group.

MacKinnon, D. P. (2008). *Introduction to statistical mediation analysis*. Lawrence Erlbaum Associates.