Lavaan tutorial

Zoë Hawks

10/20/2017

```
library(lavaan)
library(tidyverse)
library(XML)
library(OpenMx)
library(semPlot)
```

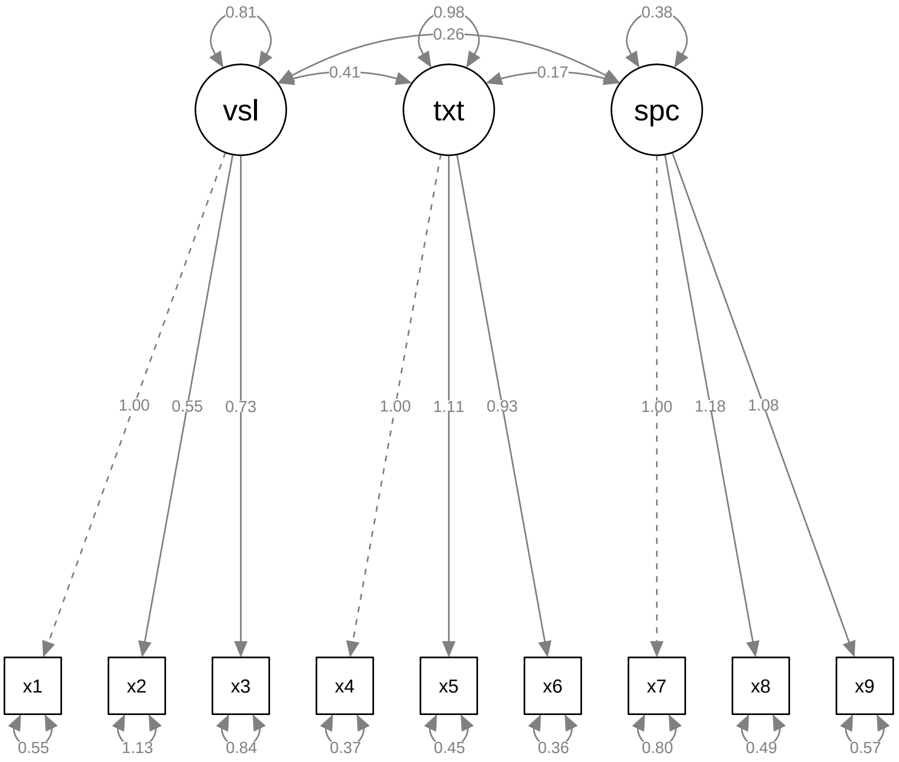
Model 1: CFA

```
CFA.mod <- '
            visual =~ x1 + x2 + x3
            textual =~ x4 + x5 + x6
            speech =~ x7 + x8 + x9
'

fit <- cfa(CFA.mod, data=HolzingerSwineford1939)
summary(fit, fit.measures = TRUE)
```

```
## lavaan (0.5-23.1097) converged normally after  35 iterations
##
##   Number of observations                           301
##
##   Estimator                                         ML
##   Minimum Function Test Statistic               85.306
##   Degrees of freedom                                24
##   P-value (Chi-square)                           0.000
##
## Model test baseline model:
##
##   Minimum Function Test Statistic              918.852
##   Degrees of freedom                                36
##   P-value                                        0.000
##
## User model versus baseline model:
##
##   Comparative Fit Index (CFI)                    0.931
##   Tucker-Lewis Index (TLI)                       0.896
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -3737.745
##   Loglikelihood unrestricted model (H1)      -3695.092
##
##   Number of free parameters                         21
##   Akaike (AIC)                                7517.490
##   Bayesian (BIC)                              7595.339
##   Sample-size adjusted Bayesian (BIC)         7528.739
```

```
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.092
##   90 Percent Confidence Interval          0.071  0.114
##   P-value RMSEA <= 0.05                          0.001
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.065
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual =~
##     x1                1.000
##     x2                0.554    0.100    5.554    0.000
##     x3                0.729    0.109    6.685    0.000
##   textual =~
##     x4                1.000
##     x5                1.113    0.065   17.014    0.000
##     x6                0.926    0.055   16.703    0.000
##   speech =~
##     x7                1.000
##     x8                1.180    0.165    7.152    0.000
##     x9                1.082    0.151    7.155    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual ~~
##     textual           0.408    0.074    5.552    0.000
##     speech            0.262    0.056    4.660    0.000
##   textual ~~
##     speech            0.173    0.049    3.518    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1                0.549    0.114    4.833    0.000
##    .x2                1.134    0.102   11.146    0.000
##    .x3                0.844    0.091    9.317    0.000
##    .x4                0.371    0.048    7.779    0.000
##    .x5                0.446    0.058    7.642    0.000
##    .x6                0.356    0.043    8.277    0.000
##    .x7                0.799    0.081    9.823    0.000
##    .x8                0.488    0.074    6.573    0.000
##    .x9                0.566    0.071    8.003    0.000
##     visual            0.809    0.145    5.564    0.000
##     textual           0.979    0.112    8.737    0.000
##     speech            0.384    0.086    4.451    0.000
```

```r
semPaths(fit, whatLabels = "est")
```

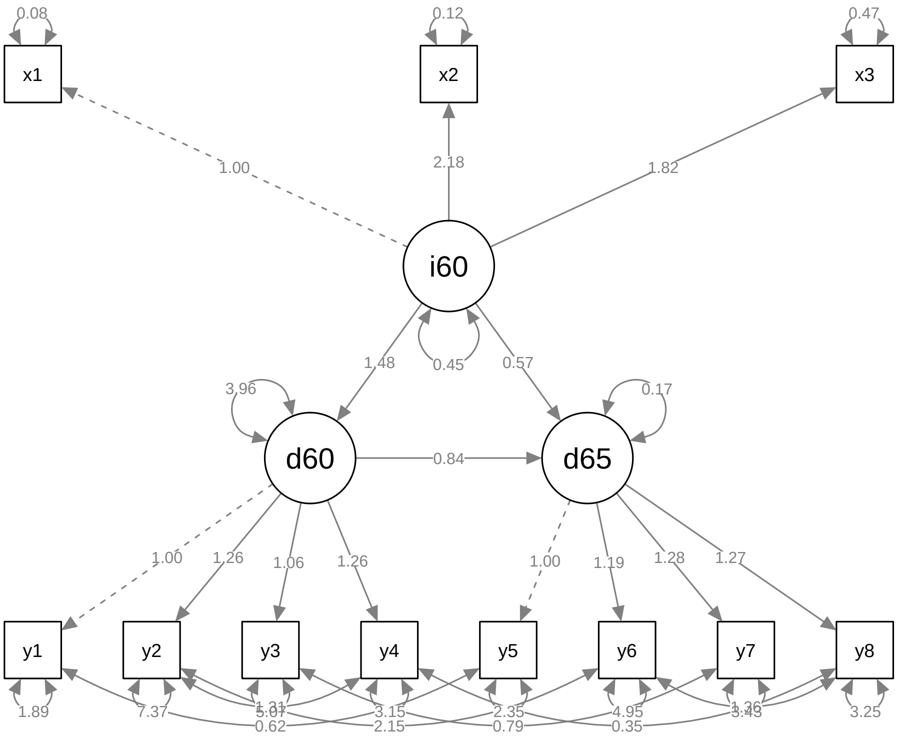


Model 2: SEM

```r
SEM.mod <- '
    ind60 =~ x1 + x2 + x3         #indicators onto latent variables
    dem60 =~ y1 + y2 + y3 + y4
    dem65 =~ y5 + y6 + y7 + y8

    dem60 ~ ind60                 #regression equations
    dem65 ~ ind60 + dem60

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

fit <- sem(SEM.mod, data=PoliticalDemocracy)
summary(fit, standardized=TRUE)
```

```
## lavaan (0.5-23.1097) converged normally after  68 iterations
##
##   Number of observations                            75
##
##   Estimator                                         ML
##   Minimum Function Test Statistic               38.125
##   Degrees of freedom                                35
##   P-value (Chi-square)                           0.329
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   ind60 =~
##     x1                1.000                               0.670    0.920
##     x2                2.180    0.139   15.742    0.000    1.460    0.973
##     x3                1.819    0.152   11.967    0.000    1.218    0.872
##   dem60 =~
##     y1                1.000                               2.223    0.850
##     y2                1.257    0.182    6.889    0.000    2.794    0.717
##     y3                1.058    0.151    6.987    0.000    2.351    0.722
##     y4                1.265    0.145    8.722    0.000    2.812    0.846
##   dem65 =~
##     y5                1.000                               2.103    0.808
##     y6                1.186    0.169    7.024    0.000    2.493    0.746
##     y7                1.280    0.160    8.002    0.000    2.691    0.824
##     y8                1.266    0.158    8.007    0.000    2.662    0.828
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   dem60 ~
##     ind60             1.483    0.399    3.715    0.000    0.447    0.447
##   dem65 ~
##     ind60             0.572    0.221    2.586    0.010    0.182    0.182
##     dem60             0.837    0.098    8.514    0.000    0.885    0.885
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##  .y1 ~~
##    .y5                0.624    0.358    1.741    0.082    0.624    0.296
##  .y2 ~~
##    .y4                1.313    0.702    1.871    0.061    1.313    0.273
##    .y6                2.153    0.734    2.934    0.003    2.153    0.356
##  .y3 ~~
##    .y7                0.795    0.608    1.308    0.191    0.795    0.191
##  .y4 ~~
##    .y8                0.348    0.442    0.787    0.431    0.348    0.109
##  .y6 ~~
##    .y8                1.356    0.568    2.386    0.017    1.356    0.338
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x1                0.082    0.019    4.184    0.000    0.082    0.154
##    .x2                0.120    0.070    1.718    0.086    0.120    0.053
##    .x3                0.467    0.090    5.177    0.000    0.467    0.239
##    .y1                1.891    0.444    4.256    0.000    1.891    0.277
##    .y2                7.373    1.374    5.366    0.000    7.373    0.486
##    .y3                5.067    0.952    5.324    0.000    5.067    0.478
##    .y4                3.148    0.739    4.261    0.000    3.148    0.285
##    .y5                2.351    0.480    4.895    0.000    2.351    0.347
##    .y6                4.954    0.914    5.419    0.000    4.954    0.443
##    .y7                3.431    0.713    4.814    0.000    3.431    0.322
##    .y8                3.254    0.695    4.685    0.000    3.254    0.315
##     ind60             0.448    0.087    5.173    0.000    1.000    1.000
##    .dem60             3.956    0.921    4.295    0.000    0.800    0.800
##    .dem65             0.172    0.215    0.803    0.422    0.039    0.039
```

```
semPaths(fit, whatLabels = "est")
```



Syntax, cont.

```
# if you want to constrain all COVARIANCES of latent variables to zero, 'orthogonal = TRUE'
fit <- cfa(CFA.mod,
                    data = HolzingerSwineford1939,
                    orthogonal = TRUE)
summary(fit, fit.measures = TRUE)
```

```
## lavaan (0.5-23.1097) converged normally after  32 iterations
##
##   Number of observations                           301
##
##   Estimator                                         ML
##   Minimum Function Test Statistic              153.527
##   Degrees of freedom                                27
##   P-value (Chi-square)                           0.000
##
## Model test baseline model:
```

```
##
##   Minimum Function Test Statistic              918.852
##   Degrees of freedom                                36
##   P-value                                        0.000
##
## User model versus baseline model:
##
##   Comparative Fit Index (CFI)                    0.857
##   Tucker-Lewis Index (TLI)                       0.809
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -3771.856
##   Loglikelihood unrestricted model (H1)      -3695.092
##
##   Number of free parameters                         18
##   Akaike (AIC)                                7579.711
##   Bayesian (BIC)                              7646.439
##   Sample-size adjusted Bayesian (BIC)         7589.354
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.125
##   90 Percent Confidence Interval          0.106  0.144
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.161
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual =~
##     x1                1.000
##     x2                0.778    0.141    5.532    0.000
##     x3                1.107    0.214    5.173    0.000
##   textual =~
##     x4                1.000
##     x5                1.133    0.067   16.906    0.000
##     x6                0.924    0.056   16.391    0.000
##   speech =~
##     x7                1.000
##     x8                1.225    0.190    6.460    0.000
##     x9                0.854    0.121    7.046    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual ~~
##     textual           0.000
##     speech            0.000
##   textual ~~
##     speech            0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1                0.835    0.118    7.064    0.000
##    .x2                1.065    0.105   10.177    0.000
##    .x3                0.633    0.129    4.899    0.000
##    .x4                0.382    0.049    7.805    0.000
##    .x5                0.416    0.059    7.038    0.000
##    .x6                0.369    0.044    8.367    0.000
##    .x7                0.746    0.086    8.650    0.000
##    .x8                0.366    0.097    3.794    0.000
##    .x9                0.696    0.072    9.640    0.000
##     visual            0.524    0.130    4.021    0.000
##     textual           0.969    0.112    8.640    0.000
##     speech            0.437    0.097    4.520    0.000
```

```
# if you want to constrain all VARIANCES of latent variables to zero, 'std.lv = TRUE'
fit <- cfa(CFA.mod,
                    data = HolzingerSwineford1939,
                    std.lv = TRUE)
summary(fit, fit.measures = TRUE)
```

```
## lavaan (0.5-23.1097) converged normally after  22 iterations
##
##   Number of observations                           301
##
##   Estimator                                         ML
##   Minimum Function Test Statistic               85.306
##   Degrees of freedom                                24
##   P-value (Chi-square)                           0.000
##
## Model test baseline model:
##
##   Minimum Function Test Statistic              918.852
##   Degrees of freedom                                36
##   P-value                                        0.000
##
## User model versus baseline model:
##
##   Comparative Fit Index (CFI)                    0.931
##   Tucker-Lewis Index (TLI)                       0.896
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -3737.745
##   Loglikelihood unrestricted model (H1)      -3695.092
##
##   Number of free parameters                         21
##   Akaike (AIC)                                7517.490
##   Bayesian (BIC)                              7595.339
##   Sample-size adjusted Bayesian (BIC)         7528.739
##
```

```
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.092
##   90 Percent Confidence Interval          0.071  0.114
##   P-value RMSEA <= 0.05                          0.001
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.065
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual =~
##     x1                0.900    0.081   11.127    0.000
##     x2                0.498    0.077    6.429    0.000
##     x3                0.656    0.074    8.817    0.000
##   textual =~
##     x4                0.990    0.057   17.474    0.000
##     x5                1.102    0.063   17.576    0.000
##     x6                0.917    0.054   17.082    0.000
##   speech =~
##     x7                0.619    0.070    8.903    0.000
##     x8                0.731    0.066   11.090    0.000
##     x9                0.670    0.065   10.305    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual ~~
##     textual           0.459    0.064    7.189    0.000
##     speech            0.471    0.073    6.461    0.000
##   textual ~~
##     speech            0.283    0.069    4.117    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1                0.549    0.114    4.833    0.000
##    .x2                1.134    0.102   11.146    0.000
##    .x3                0.844    0.091    9.317    0.000
##    .x4                0.371    0.048    7.778    0.000
##    .x5                0.446    0.058    7.642    0.000
##    .x6                0.356    0.043    8.277    0.000
##    .x7                0.799    0.081    9.823    0.000
##    .x8                0.488    0.074    6.573    0.000
##    .x9                0.566    0.071    8.003    0.000
##     visual            1.000
##     textual           1.000
##     speech            1.000
```

```
coef1 <- broom::tidy(coef(fit))
coef1
```

```
## # A tibble: 21 x 2
##          names         x
##          <chr>     <dbl>
##  1  visual=~x1 0.8996190
##  2  visual=~x2 0.4979392
##  3  visual=~x3 0.6561556
##  4 textual=~x4 0.9896926
##  5 textual=~x5 1.1016031
##  6 textual=~x6 0.9165999
##  7  speech=~x7 0.6194736
##  8  speech=~x8 0.7309475
##  9  speech=~x9 0.6699801
## 10      x1~~x1 0.5490548
## # ... with 11 more rows
```

```
# Playing around with paramter lavels
# Note: providing starting value != fixing factor loading
CFA.mod <- '
            visual =~ x1 + x2 + HEY*x3
            textual =~ x4 + x5 + x6
            speech =~ x7 + YO*x8 + start(0.9)*x9
'
fit <- cfa(CFA.mod, data = HolzingerSwineford1939)
summary(fit, fit.measures = TRUE)
```

```
## lavaan (0.5-23.1097) converged normally after  35 iterations
##
##   Number of observations                           301
##
##   Estimator                                         ML
##   Minimum Function Test Statistic               85.306
##   Degrees of freedom                                24
##   P-value (Chi-square)                           0.000
##
## Model test baseline model:
##
##   Minimum Function Test Statistic              918.852
##   Degrees of freedom                                36
##   P-value                                        0.000
##
## User model versus baseline model:
##
##   Comparative Fit Index (CFI)                    0.931
##   Tucker-Lewis Index (TLI)                       0.896
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -3737.745
##   Loglikelihood unrestricted model (H1)      -3695.092
##
##   Number of free parameters                         21
##   Akaike (AIC)                                7517.490
##   Bayesian (BIC)                              7595.339
##   Sample-size adjusted Bayesian (BIC)         7528.739
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.092
##   90 Percent Confidence Interval          0.071  0.114
##   P-value RMSEA <= 0.05                          0.001
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.065
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual =~
##     x1                1.000
##     x2                0.553    0.100    5.554    0.000
##     x3       (HEY)    0.729    0.109    6.685    0.000
##   textual =~
##     x4                1.000
##     x5                1.113    0.065   17.014    0.000
##     x6                0.926    0.055   16.703    0.000
##   speech =~
##     x7                1.000
##     x8        (YO)    1.180    0.165    7.152    0.000
##     x9                1.082    0.151    7.155    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual ~~
##     textual           0.408    0.074    5.552    0.000
##     speech            0.262    0.056    4.660    0.000
##   textual ~~
##     speech            0.173    0.049    3.518    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1                0.549    0.114    4.833    0.000
##    .x2                1.134    0.102   11.146    0.000
##    .x3                0.844    0.091    9.317    0.000
##    .x4                0.371    0.048    7.778    0.000
##    .x5                0.446    0.058    7.642    0.000
##    .x6                0.356    0.043    8.277    0.000
##    .x7                0.799    0.081    9.823    0.000
##    .x8                0.488    0.074    6.573    0.000
##    .x9                0.566    0.071    8.003    0.000
##     visual            0.809    0.145    5.564    0.000
##     textual           0.979    0.112    8.737    0.000
##     speech            0.384    0.086    4.451    0.000
```

```
coef1 <- broom::tidy(coef(fit))
coef1
```

```
## # A tibble: 21 x 2
##          names         x
##          <chr>     <dbl>
##  1  visual=~x2 0.5534995
##  2         HEY 0.7293703
##  3 textual=~x5 1.1130758
##  4 textual=~x6 0.9261459
##  5          YO 1.1799507
##  6  speech=~x9 1.0815305
##  7      x1~~x1 0.5490533
##  8      x2~~x2 1.1338392
##  9      x3~~x3 0.8443228
## 10      x4~~x4 0.3711726
## # ... with 11 more rows
```

```
# Specifying equality -- x3, x8, and x9 all the same
CFA.mod <- '
            visual =~ x1 + x2 + a*x3
            textual =~ x4 + x5 + x6
            speech =~ x7 + a*x8 + a*x9
'
fit <- cfa(CFA.mod, data = HolzingerSwineford1939)
summary(fit, fit.measures = TRUE)
```

```
## lavaan (0.5-23.1097) converged normally after  28 iterations
##
##   Number of observations                           301
##
##   Estimator                                         ML
##   Minimum Function Test Statistic               90.945
##   Degrees of freedom                                26
##   P-value (Chi-square)                           0.000
##
## Model test baseline model:
##
##   Minimum Function Test Statistic              918.852
##   Degrees of freedom                                36
##   P-value                                        0.000
##
## User model versus baseline model:
##
##   Comparative Fit Index (CFI)                    0.926
##   Tucker-Lewis Index (TLI)                       0.898
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -3740.565
##   Loglikelihood unrestricted model (H1)      -3695.092
##
##   Number of free parameters                         19
##   Akaike (AIC)                                7519.129
##   Bayesian (BIC)                              7589.564
##   Sample-size adjusted Bayesian (BIC)         7529.307
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.091
##   90 Percent Confidence Interval          0.071  0.112
##   P-value RMSEA <= 0.05                          0.001
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.073
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual =~
##     x1                1.000
##     x2                0.682    0.112    6.111    0.000
##     x3         (a)    0.986    0.091   10.897    0.000
##   textual =~
##     x4                1.000
##     x5                1.115    0.066   16.988    0.000
##     x6                0.927    0.056   16.674    0.000
##   speech =~
##     x7                1.000
##     x8         (a)    0.986    0.091   10.897    0.000
##     x9         (a)    0.986    0.091   10.897    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual ~~
##     textual           0.331    0.064    5.141    0.000
##     speech            0.253    0.056    4.537    0.000
##   textual ~~
##     speech            0.196    0.053    3.689    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1                0.697    0.089    7.870    0.000
##    .x2                1.100    0.101   10.838    0.000
##    .x3                0.743    0.090    8.262    0.000
##    .x4                0.373    0.048    7.787    0.000
##    .x5                0.444    0.059    7.596    0.000
##    .x6                0.356    0.043    8.255    0.000
##    .x7                0.775    0.082    9.444    0.000
##    .x8                0.527    0.059    8.875    0.000
##    .x9                0.557    0.061    9.084    0.000
##     visual            0.607    0.100    6.072    0.000
##     textual           0.978    0.112    8.722    0.000
##     speech            0.469    0.084    5.597    0.000
```

```r
# Setting constraints
set.seed(1234)
mydata <- data.frame(y = rnorm(100),
                   x1 = rnorm(100),
                   x2 = rnorm(100),
                   x3 = rnorm(100))

new.mod <- ' y ~ b1*x1 + b2*x2 + b3*x3    # model

            b1 == (b2 + b3)^2             # constraints
            b1 > exp(b2 + b3) '
fit <- sem(new.mod, data=mydata)
summary(fit)
```

```
## lavaan (0.5-23.1097) converged normally after  51 iterations
##
##   Number of observations                           100
##
##   Estimator                                         ML
##   Minimum Function Test Statistic               50.660
##   Degrees of freedom                                 1
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   y ~
##     x1        (b1)    0.495       NA
##     x2        (b2)   -0.405    0.092   -4.411    0.000
##     x3        (b3)   -0.299    0.092   -3.256    0.001
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .y                 1.610    0.228    7.071    0.000
##
## Constraints:
##                                                |Slack|
##     b1 - ((b2+b3)^2)                             0.000
##     b1 - (exp(b2+b3))                            0.000
```

```r
# Bringing in the means

  # can fix intercepts, e.g. variable ~ 1
  # can specify 'meanstructure = TRUE' when fitting model
  # by default, sem() and cfa() fix latent variable intercepts to zero
  # In practice, the only reason why a user would add intercept-formulas in the model syntax, is becaus
```

Multiple groups

```r
# add 'group' argument within fit formula --> able to estimate different fits for different groups
fit <- cfa(CFA.mod,
           data = HolzingerSwineford1939,
           group = "school")
summary(fit)
```

```
## lavaan (0.5-23.1097) converged normally after  49 iterations
##
##   Number of observations per group
##   Pasteur                                          156
##   Grant-White                                      145
##
##   Estimator                                         ML
##   Minimum Function Test Statistic              119.924
##   Degrees of freedom                                50
##   P-value (Chi-square)                           0.000
##
## Chi-square for each group:
##
##   Pasteur                                       68.381
##   Grant-White                                   51.542
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
##
## Group 1 [Pasteur]:
##
## Latent Variables:
##                     Estimate  Std.Err  z-value  P(>|z|)
##   visual =~
##     x1                 1.000
##     x2                 0.544    0.140    3.897    0.000
##     x3          (a)    0.829    0.125    6.627    0.000
##   textual =~
##     x4                 1.000
##     x5                 1.188    0.103   11.577    0.000
##     x6                 0.877    0.077   11.381    0.000
##   speech =~
##     x7                 1.000
##     x8          (a)    0.829    0.125    6.627    0.000
##     x9          (a)    0.829    0.125    6.627    0.000
##
## Covariances:
##                     Estimate  Std.Err  z-value  P(>|z|)
##   visual ~~
##     textual            0.392    0.097    4.060    0.000
##     speech             0.193    0.081    2.393    0.017
##   textual ~~
##     speech             0.218    0.076    2.882    0.004
```

```
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1                4.941    0.093   53.054    0.000
##    .x2                5.984    0.098   60.949    0.000
##    .x3                2.487    0.095   26.154    0.000
##    .x4                2.823    0.092   30.689    0.000
##    .x5                3.995    0.105   38.183    0.000
##    .x6                1.922    0.079   24.321    0.000
##    .x7                4.432    0.089   49.961    0.000
##    .x8                5.563    0.076   73.043    0.000
##    .x9                5.418    0.079   68.333    0.000
##     visual            0.000
##     textual           0.000
##     speech            0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1                0.560    0.144    3.876    0.000
##    .x2                1.269    0.156    8.140    0.000
##    .x3                0.866    0.134    6.462    0.000
##    .x4                0.429    0.070    6.156    0.000
##    .x5                0.451    0.086    5.215    0.000
##    .x6                0.290    0.050    5.750    0.000
##    .x7                0.770    0.126    6.095    0.000
##    .x8                0.591    0.087    6.795    0.000
##    .x9                0.667    0.094    7.067    0.000
##     visual            0.793    0.179    4.443    0.000
##     textual           0.890    0.150    5.941    0.000
##     speech            0.457    0.124    3.680    0.000
##
##
## Group 2 [Grant-White]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual =~
##     x1                1.000
##     x2                0.736    0.155    4.760    0.000
##     x3                0.925    0.166    5.584    0.000
##   textual =~
##     x4                1.000
##     x5                0.990    0.087   11.418    0.000
##     x6                0.963    0.085   11.377    0.000
##   speech =~
##     x7                1.000
##     x8                1.226    0.187    6.569    0.000
##     x9                1.058    0.165    6.429    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual ~~
##     textual           0.408    0.098    4.153    0.000
##     speech            0.276    0.076    3.639    0.000
##   textual ~~
##     speech            0.222    0.073    3.022    0.003
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1                4.930    0.095   51.696    0.000
##    .x2                6.200    0.092   67.416    0.000
##    .x3                1.996    0.086   23.195    0.000
##    .x4                3.317    0.093   35.625    0.000
##    .x5                4.712    0.096   48.986    0.000
##    .x6                2.469    0.094   26.277    0.000
##    .x7                3.921    0.086   45.819    0.000
##    .x8                5.488    0.087   63.174    0.000
##    .x9                5.327    0.085   62.571    0.000
##     visual            0.000
##     textual           0.000
##     speech            0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1                0.715    0.126    5.676    0.000
##    .x2                0.899    0.123    7.339    0.000
##    .x3                0.557    0.103    5.409    0.000
##    .x4                0.315    0.065    4.870    0.000
##    .x5                0.419    0.072    5.812    0.000
##    .x6                0.406    0.069    5.880    0.000
##    .x7                0.600    0.091    6.584    0.000
##    .x8                0.401    0.094    4.248    0.000
##    .x9                0.535    0.089    6.010    0.000
##     visual            0.604    0.160    3.762    0.000
##     textual           0.942    0.152    6.177    0.000
##     speech            0.461    0.118    3.910    0.000
```

```
# If you want to fix parameters, or provide starting values, you can use the same pre-multiplication te
Group.mod <- '
              visual =~ .5*x1 + c(a,a)*x2 + c(a,a)*x3
              textual =~ x4 + x5 + c(.5, .7)*x6
              speed   =~ c(1, NA)*x7 + x8 + x9
'
fit <- cfa(Group.mod,
           data = HolzingerSwineford1939,
           group = "school")
summary(fit)
```

```
## lavaan (0.5-23.1097) converged normally after  68 iterations
##
##   Number of observations per group
##   Pasteur                                          156
##   Grant-White                                      145
##
##   Estimator                                         ML
##   Minimum Function Test Statistic              162.771
##   Degrees of freedom                                52
##   P-value (Chi-square)                           0.000
##
```

```
## Chi-square for each group:
##
##   Pasteur                                       97.625
##   Grant-White                                   65.147
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
##
## Group 1 [Pasteur]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual =~
##     x1                0.500
##     x2         (a)    0.347    0.044    7.934    0.000
##     x3         (a)    0.347    0.044    7.934    0.000
##   textual =~
##     x4                1.000
##     x5                1.007    0.090   11.146    0.000
##     x6                0.500
##   speed =~
##     x7                1.000
##     x8                1.124    0.278    4.043    0.000
##     x9                0.938    0.229    4.095    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual ~~
##     textual           0.912    0.219    4.158    0.000
##     speed             0.340    0.143    2.371    0.018
##   textual ~~
##     speed             0.199    0.078    2.555    0.011
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1                4.941    0.093   52.999    0.000
##    .x2                5.984    0.103   58.328    0.000
##    .x3                2.487    0.092   27.133    0.000
##    .x4                2.823    0.097   29.043    0.000
##    .x5                3.995    0.105   38.183    0.000
##    .x6                1.922    0.069   27.834    0.000
##    .x7                4.432    0.087   51.181    0.000
##    .x8                5.563    0.078   71.214    0.000
##    .x9                5.418    0.079   68.440    0.000
##     visual            0.000
##     textual           0.000
##     speed             0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1                0.541    0.141    3.850    0.000
##    .x2                1.250    0.158    7.895    0.000
##    .x3                0.919    0.123    7.477    0.000
##    .x4                0.301    0.088    3.411    0.001
##    .x5                0.518    0.105    4.912    0.000
##    .x6                0.451    0.056    8.092    0.000
##    .x7                0.824    0.125    6.611    0.000
##    .x8                0.516    0.115    4.470    0.000
##    .x9                0.673    0.105    6.412    0.000
##     visual            3.260    0.696    4.685    0.000
##     textual           1.173    0.174    6.754    0.000
##     speed             0.346    0.125    2.761    0.006
##
##
## Group 2 [Grant-White]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual =~
##     x1                0.500
##     x2         (a)    0.347    0.044    7.934    0.000
##     x3         (a)    0.347    0.044    7.934    0.000
##   textual =~
##     x4                1.000
##     x5                0.892    0.076   11.784    0.000
##     x6                0.700
##   speed =~
##     x7                1.457    0.145   10.062    0.000
##     x8                1.788    0.135   13.287    0.000
##     x9                1.569    0.140   11.172    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual ~~
##     textual           0.973    0.219    4.449    0.000
##     speed             0.431    0.098    4.401    0.000
##   textual ~~
##     speed             0.163    0.050    3.231    0.001
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1                4.930    0.097   50.754    0.000
##    .x2                6.200    0.093   66.491    0.000
##    .x3                1.996    0.083   24.048    0.000
##    .x4                3.317    0.097   34.270    0.000
##    .x5                4.712    0.096   48.986    0.000
##    .x6                2.469    0.086   28.852    0.000
##    .x7                3.921    0.086   45.819    0.000
##    .x8                5.488    0.087   63.174    0.000
##    .x9                5.327    0.085   62.571    0.000
##     visual            0.000
##     textual           0.000
##     speed             0.000
##
## Variances:
```

```
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1                0.633    0.128    4.944    0.000
##    .x2                0.907    0.121    7.504    0.000
##    .x3                0.645    0.092    7.015    0.000
##    .x4                0.238    0.071    3.353    0.001
##    .x5                0.451    0.078    5.781    0.000
##    .x6                0.513    0.069    7.421    0.000
##    .x7                0.607    0.091    6.645    0.000
##    .x8                0.410    0.094    4.366    0.000
##    .x9                0.524    0.089    5.902    0.000
##     visual            2.939    0.655    4.488    0.000
##     textual           1.121    0.161    6.961    0.000
##     speed             0.214    0.031    6.867    0.000
```

```r
# What if we want constraints for a large number of parameters to be equal across groups?
# Use 'group.equal' argument within fit function
# For 'group.equal,' might also specify: intercepts, means, residuals, residual.covariances, lv.varianc
fit <- cfa(CFA.mod,
           data = HolzingerSwineford1939,
           group = "school",
           group.equal = c("loadings"))
summary(fit)
```

```
## lavaan (0.5-23.1097) converged normally after  33 iterations
##
##   Number of observations per group
##   Pasteur                                           156
##   Grant-White                                       145
##
##   Estimator                                          ML
##   Minimum Function Test Statistic               129.051
##   Degrees of freedom                                 56
##   P-value (Chi-square)                            0.000
##
## Chi-square for each group:
##
##   Pasteur                                        72.468
##   Grant-White                                    56.583
##
## Parameter Estimates:
##
##   Information                                  Expected
##   Standard Errors                              Standard
##
##
## Group 1 [Pasteur]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual =~
##     x1                1.000
##     x2      (.p2.)    0.719    0.110    6.516    0.000
##     x3         (a)    1.011    0.088   11.513    0.000
##   textual =~
##     x4                1.000
##     x5      (.p5.)    1.086    0.068   16.052    0.000
##     x6      (.p6.)    0.912    0.058   15.750    0.000
##   speech =~
##     x7                1.000
##     x8         (a)    1.011    0.088   11.513    0.000
##     x9         (a)    1.011    0.088   11.513    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual ~~
##     textual           0.330    0.086    3.860    0.000
##     speech            0.165    0.062    2.638    0.008
##   textual ~~
##     speech            0.195    0.065    2.983    0.003
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1                4.941    0.092   53.786    0.000
##    .x2                5.984    0.099   60.268    0.000
##    .x3                2.487    0.095   26.145    0.000
##    .x4                2.823    0.093   30.362    0.000
##    .x5                3.995    0.101   39.723    0.000
##    .x6                1.922    0.081   23.701    0.000
##    .x7                4.432    0.087   50.833    0.000
##    .x8                5.563    0.077   72.132    0.000
##    .x9                5.418    0.080   67.935    0.000
##     visual            0.000
##     textual           0.000
##     speech            0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1                0.700    0.119    5.859    0.000
##    .x2                1.219    0.154    7.930    0.000
##    .x3                0.782    0.128    6.129    0.000
##    .x4                0.436    0.070    6.236    0.000
##    .x5                0.503    0.081    6.174    0.000
##    .x6                0.268    0.051    5.289    0.000
##    .x7                0.833    0.115    7.267    0.000
##    .x8                0.567    0.086    6.614    0.000
##    .x9                0.632    0.092    6.863    0.000
##     visual            0.617    0.125    4.927    0.000
##     textual           0.912    0.137    6.646    0.000
##     speech            0.353    0.080    4.398    0.000
##
##
## Group 2 [Grant-White]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual =~
##     x1                1.000
##     x2      (.p2.)    0.719    0.110    6.516    0.000
##     x3         (a)    1.011    0.088   11.513    0.000
##   textual =~
##     x4                1.000
##     x5      (.p5.)    1.086    0.068   16.052    0.000
##     x6      (.p6.)    0.912    0.058   15.750    0.000
##   speech =~
##     x7                1.000
##     x8         (a)    1.011    0.088   11.513    0.000
##     x9         (a)    1.011    0.088   11.513    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual ~~
##     textual           0.382    0.086    4.425    0.000
##     speech            0.292    0.076    3.850    0.000
##   textual ~~
##     speech            0.260    0.077    3.387    0.001
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1                4.930    0.095   52.037    0.000
##    .x2                6.200    0.091   68.181    0.000
##    .x3                1.996    0.087   22.897    0.000
##    .x4                3.317    0.092   35.960    0.000
##    .x5                4.712    0.100   47.154    0.000
##    .x6                2.469    0.091   27.261    0.000
##    .x7                3.921    0.088   44.326    0.000
##    .x8                5.488    0.084   65.180    0.000
##    .x9                5.327    0.085   62.312    0.000
##     visual            0.000
##     textual           0.000
##     speech            0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1                0.739    0.115    6.444    0.000
##    .x2                0.908    0.119    7.605    0.000
##    .x3                0.527    0.097    5.463    0.000
##    .x4                0.330    0.062    5.302    0.000
##    .x5                0.383    0.073    5.252    0.000
##    .x6                0.438    0.067    6.582    0.000
##    .x7                0.589    0.092    6.420    0.000
##    .x8                0.471    0.077    6.091    0.000
##    .x9                0.503    0.080    6.256    0.000
##     visual            0.562    0.117    4.817    0.000
##     textual           0.904    0.136    6.640    0.000
##     speech            0.545    0.108    5.041    0.000
```

```
# What if we want all but a few constraints to be equal across groups?
# Use 'group.partial' argument within fit function to specify parameters that should remain free
fit <- cfa(CFA.mod,
           data = HolzingerSwineford1939,
           group = "school",
           group.equal = c("loadings", "intercepts"),
           group.partial = c("visual=~x2", "x7~1"))
```

```
# Measurement invariance
# To test the measurement invariance of a CFA model across several groups; each model is compared to th
library(semTools)
measurementInvariance(CFA.mod,
                      data = HolzingerSwineford1939,
                      group = "school")
```

```
##
## Measurement invariance models:
##
## Model 1 : fit.configural
## Model 2 : fit.loadings
## Model 3 : fit.intercepts
## Model 4 : fit.means
##
## Chi Square Difference Test
##
##                Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## fit.configural 50 7484.5 7699.5 119.92
## fit.loadings   56 7481.6 7674.4 129.05      9.127       6     0.1665
## fit.intercepts 62 7507.7 7678.3 167.18     38.132       6  1.058e-06 ***
## fit.means      65 7545.5 7704.9 210.94     43.761       3  1.696e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Fit measures:
##
##                  cfi rmsea cfi.delta rmsea.delta
## fit.configural 0.921 0.096        NA          NA
## fit.loadings   0.918 0.093     0.004       0.003
## fit.intercepts 0.881 0.106     0.036       0.013
## fit.means      0.835 0.122     0.046       0.016
```

Growth curve models

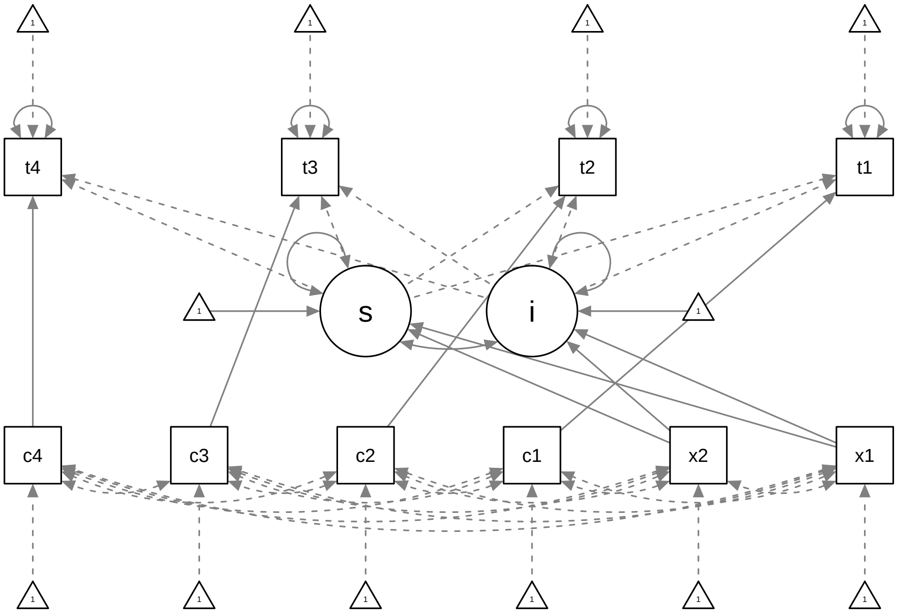
```
# Random effects are used to capture individual differences. The random effects are conveniently repres

# linear growth model with 4 timepoints
# to specify intercept: coefficients stable across timepoints
# to specify slope: coefficients increment across timepoints
Growth.mod <- ' i =~ 1*t1 + 1*t2 + 1*t3 + 1*t4
            s =~ 0*t1 + 1*t2 + 2*t3 + 3*t4 '
fit <- growth(Growth.mod, data=Demo.growth)
summary(fit)
```

```
## lavaan (0.5-23.1097) converged normally after  29 iterations
##
##   Number of observations                           400
##
##   Estimator                                         ML
##   Minimum Function Test Statistic                8.069
##   Degrees of freedom                                 5
##   P-value (Chi-square)                           0.152
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
## Latent Variables:
##                     Estimate  Std.Err  z-value  P(>|z|)
##   i =~
##     t1                1.000
##     t2                1.000
##     t3                1.000
##     t4                1.000
##   s =~
##     t1                0.000
##     t2                1.000
##     t3                2.000
##     t4                3.000
##
## Covariances:
##                     Estimate  Std.Err  z-value  P(>|z|)
##   i ~~
##     s                 0.618    0.071    8.686    0.000
##
## Intercepts:
##                     Estimate  Std.Err  z-value  P(>|z|)
##    .t1                0.000
##    .t2                0.000
##    .t3                0.000
##    .t4                0.000
##     i                 0.615    0.077    8.007    0.000
##     s                 1.006    0.042   24.076    0.000
##
## Variances:
##                     Estimate  Std.Err  z-value  P(>|z|)
##    .t1                0.595    0.086    6.944    0.000
##    .t2                0.676    0.061   11.061    0.000
##    .t3                0.635    0.072    8.761    0.000
##    .t4                0.508    0.124    4.090    0.000
##     i                 1.932    0.173   11.194    0.000
##     s                 0.587    0.052   11.336    0.000
```

```
# a linear growth model with a time-varying covariate (c) and two regressors (x1 & x2) that influence t
model <- '
  # intercept and slope with fixed coefficients
    i =~ 1*t1 + 1*t2 + 1*t3 + 1*t4
    s =~ 0*t1 + 1*t2 + 2*t3 + 3*t4
  # regressions
    i ~ x1 + x2
    s ~ x1 + x2
  # time-varying covariates
    t1 ~ c1
    t2 ~ c2
    t3 ~ c3
    t4 ~ c4
'
fit <- growth(model, data = Demo.growth)
summary(fit)
```

```
## lavaan (0.5-23.1097) converged normally after  31 iterations
##
##   Number of observations                           400
##
##   Estimator                                         ML
##   Minimum Function Test Statistic               26.059
##   Degrees of freedom                                21
##   P-value (Chi-square)                           0.204
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   i =~
##     t1                1.000
##     t2                1.000
##     t3                1.000
##     t4                1.000
##   s =~
##     t1                0.000
##     t2                1.000
##     t3                2.000
##     t4                3.000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   i ~
##     x1                0.608    0.060   10.134    0.000
##     x2                0.604    0.064    9.412    0.000
##   s ~
##     x1                0.262    0.029    9.198    0.000
##     x2                0.522    0.031   17.083    0.000
##   t1 ~
##     c1                0.143    0.050    2.883    0.004
##   t2 ~
##     c2                0.289    0.046    6.295    0.000
##   t3 ~
##     c3                0.328    0.044    7.361    0.000
##   t4 ~
##     c4                0.330    0.058    5.655    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##  .i ~~
##    .s                  0.075    0.040    1.855    0.064
##
## Intercepts:
##                     Estimate  Std.Err  z-value  P(>|z|)
##    .t1                 0.000
##    .t2                 0.000
##    .t3                 0.000
##    .t4                 0.000
##    .i                  0.580    0.062    9.368    0.000
##    .s                  0.958    0.029   32.552    0.000
##
## Variances:
##                     Estimate  Std.Err  z-value  P(>|z|)
##    .t1                 0.580    0.080    7.230    0.000
##    .t2                 0.596    0.054   10.969    0.000
##    .t3                 0.481    0.055    8.745    0.000
##    .t4                 0.535    0.098    5.466    0.000
##    .i                  1.079    0.112    9.609    0.000
##    .s                  0.224    0.027    8.429    0.000
```

```
semPaths(fit, rotation = 3, exoVar = FALSE)
```



Categorical variables

```
# exogenous ~ independent, endogenous ~ dependent
# if exogenous categorical variable, treat as normal regression (e.g., dummy code)
# if endogenous categorical variable, need to either (1) delcare variables "ordered" or (2) specifiy as

#where items 1-4 are your binary/ordered variables
# fit <- cfa(myModel, data = myData,
#             ordered=c("item1","item2",
#                       "item3","item3"))
```

Using covariance matrix as input

```
lower <- '
  11.834
  6.947   9.364
  6.819   5.091  12.532
  4.783   5.028   7.495   9.986
 -3.839  -3.889  -3.841  -3.625  9.610
 -21.899 -18.831 -21.748 -18.775 35.522 450.288 '

wheaton.cov <-
    getCov(lower, names = c("anomia67", "powerless67",
                            "anomia71", "powerless71",
                            "education", "sei"))

wheaton.cov
```

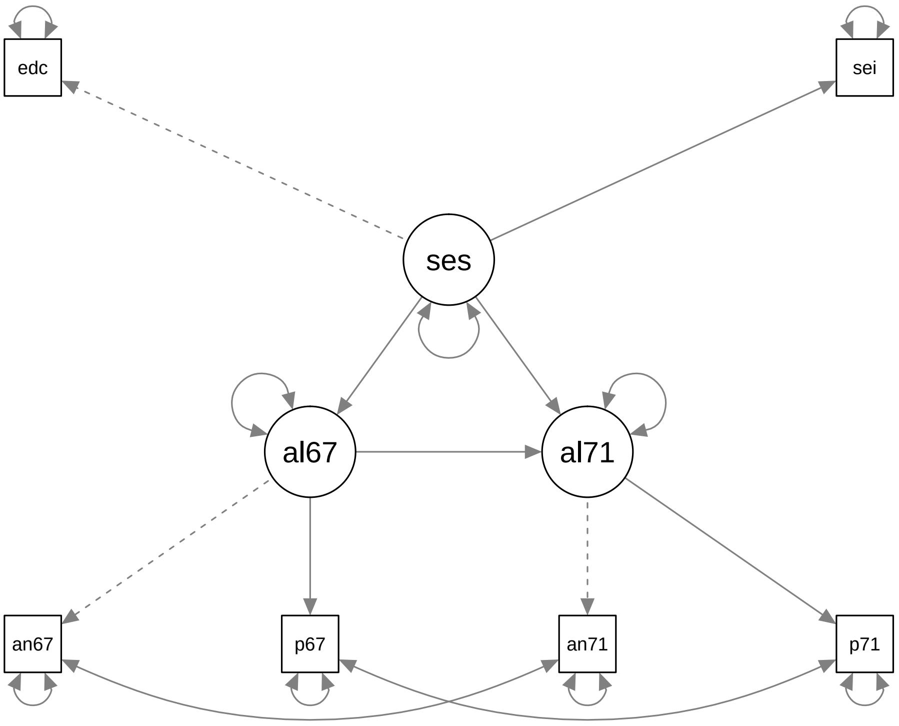
```
##             anomia67 powerless67 anomia71 powerless71 education     sei
## anomia67      11.834       6.947    6.819       4.783    -3.839 -21.899
## powerless67    6.947       9.364    5.091       5.028    -3.889 -18.831
## anomia71       6.819       5.091   12.532       7.495    -3.841 -21.748
## powerless71    4.783       5.028    7.495       9.986    -3.625 -18.775
## education     -3.839      -3.889   -3.841      -3.625     9.610  35.522
## sei          -21.899     -18.831  -21.748     -18.775    35.522 450.288
```

```
wheaton.model <- '
                                            #latent variables
    ses     =~ education + sei
    alien67 =~ anomia67 + powerless67
    alien71 =~ anomia71 + powerless71
                                            #regressions
    alien71 ~ alien67 + ses
    alien67 ~ ses
                                            #correlated residuals
    anomia67 ~~ anomia71
    powerless67 ~~ powerless71
'
fit <- sem(wheaton.model,
           sample.cov = wheaton.cov,
           sample.nobs = 932)               #need to specify number of observations
summary(fit, standardized = TRUE)
```

```
## lavaan (0.5-23.1097) converged normally after  73 iterations
##
##   Number of observations                           932
##
##   Estimator                                         ML
##   Minimum Function Test Statistic                4.735
##   Degrees of freedom                                 4
##   P-value (Chi-square)                           0.316
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
```

```
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   ses =~
##     education         1.000                               2.607    0.842
##     sei               5.219    0.422   12.364    0.000   13.609    0.642
##   alien67 =~
##     anomia67          1.000                               2.663    0.774
##     powerless67       0.979    0.062   15.895    0.000    2.606    0.852
##   alien71 =~
##     anomia71          1.000                               2.850    0.805
##     powerless71       0.922    0.059   15.498    0.000    2.628    0.832
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   alien71 ~
##     alien67           0.607    0.051   11.898    0.000    0.567    0.567
##     ses              -0.227    0.052   -4.334    0.000   -0.207   -0.207
##   alien67 ~
##     ses              -0.575    0.056  -10.195    0.000   -0.563   -0.563
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##  .anomia67 ~~
##    .anomia71          1.623    0.314    5.176    0.000    1.623    0.356
##  .powerless67 ~~
##    .powerless71       0.339    0.261    1.298    0.194    0.339    0.121
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .education         2.801    0.507    5.525    0.000    2.801    0.292
##    .sei             264.597   18.126   14.597    0.000  264.597    0.588
##    .anomia67          4.731    0.453   10.441    0.000    4.731    0.400
##    .powerless67       2.563    0.403    6.359    0.000    2.563    0.274
##    .anomia71          4.399    0.515    8.542    0.000    4.399    0.351
##    .powerless71       3.070    0.434    7.070    0.000    3.070    0.308
##     ses               6.798    0.649   10.475    0.000    1.000    1.000
##    .alien67           4.841    0.467   10.359    0.000    0.683    0.683
##    .alien71           4.083    0.404   10.104    0.000    0.503    0.503
```

```
semPaths(fit)
```



Estimators, standard errors, and missing values

```r
# Estimators: ML (default), GLS, WLS, DWLS, ULS
# ML estimation -- default is biased (n); if you want unbiased (n-1), use argument likelihood = "wishar

fit <- cfa(CFA.mod,
           data = HolzingerSwineford1939,
           likelihood = "wishart")
summary(fit)
```

```
## lavaan (0.5-23.1097) converged normally after  29 iterations
##
##   Number of observations                           301
##
##   Estimator                                         ML
##   Minimum Function Test Statistic               90.643
##   Degrees of freedom                                26
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
## Latent Variables:
##                     Estimate  Std.Err  z-value  P(>|z|)
##   visual =~
##     x1                 1.000
##     x2                 0.682    0.112    6.101    0.000
##     x3          (a)    0.986    0.091   10.879    0.000
##   textual =~
##     x4                 1.000
##     x5                 1.115    0.066   16.959    0.000
##     x6                 0.927    0.056   16.646    0.000
##   speech =~
##     x7                 1.000
##     x8          (a)    0.986    0.091   10.879    0.000
##     x9          (a)    0.986    0.091   10.879    0.000
##
## Covariances:
##                     Estimate  Std.Err  z-value  P(>|z|)
##   visual ~~
##     textual            0.332    0.065    5.132    0.000
##     speech             0.253    0.056    4.530    0.000
##   textual ~~
##     speech             0.196    0.053    3.683    0.000
##
## Variances:
##                     Estimate  Std.Err  z-value  P(>|z|)
##    .x1                 0.700    0.089    7.857    0.000
##    .x2                 1.104    0.102   10.820    0.000
##    .x3                 0.745    0.090    8.249    0.000
##    .x4                 0.374    0.048    7.775    0.000
##    .x5                 0.446    0.059    7.584    0.000
##    .x6                 0.357    0.043    8.241    0.000
##    .x7                 0.777    0.082    9.428    0.000
##    .x8                 0.529    0.060    8.861    0.000
##    .x9                 0.559    0.062    9.069    0.000
##     visual             0.609    0.100    6.062    0.000
##     textual            0.981    0.113    8.708    0.000
##     speech             0.470    0.084    5.588    0.000
```

```
# Page 32 of tutorial: information about missing values, standard errors, and bootstrapping in Lavaan
```

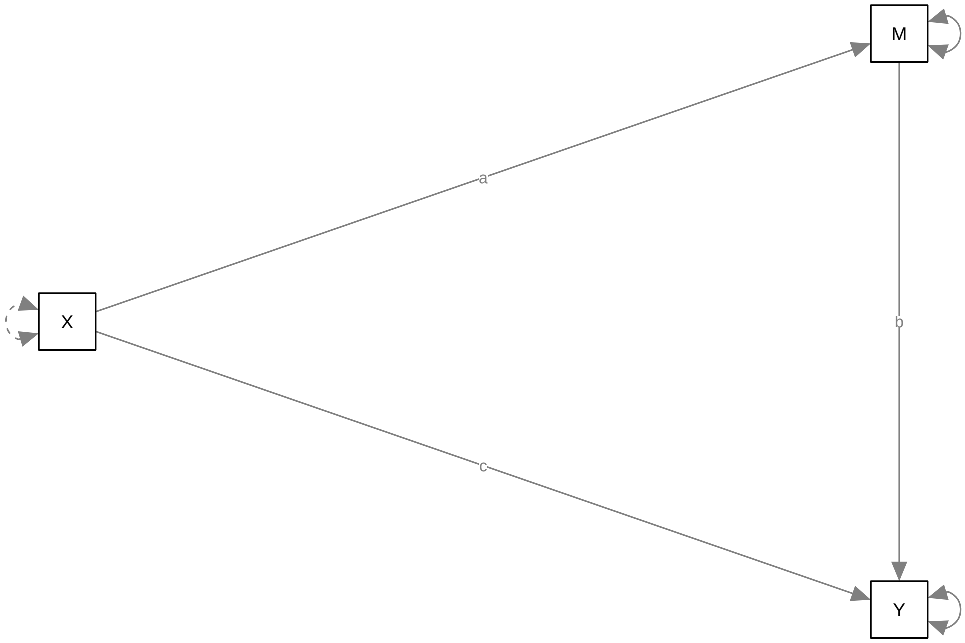
Indirect effects & mediation

```
#create fake dataset on which to conduct mediation analysis
set.seed(1234)
X <- rnorm(100)
M <- 0.5*X + rnorm(100)
Y <- 0.7*M + rnorm(100)
Data <- data.frame(X = X, Y = Y, M = M)

model <- ' # direct effect
             Y ~ c*X
           # mediator
             M ~ a*X
             Y ~ b*M
           # indirect effect (a*b)
             ab := a*b
           # total effect
             total := c + (a*b)
```

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

```
## lavaan (0.5-23.1097) converged normally after  12 iterations
##
##   Number of observations                           100
##
##   Estimator                                         ML
##   Minimum Function Test Statistic                0.000
##   Degrees of freedom                                 0
##   Minimum Function Value               0.0000000000000
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
## Regressions:
##                     Estimate  Std.Err  z-value  P(>|z|)
##   Y ~
##     X          (c)    0.036    0.104    0.348    0.728
##   M ~
##     X          (a)    0.474    0.103    4.613    0.000
##   Y ~
##     M          (b)    0.788    0.092    8.539    0.000
##
## Variances:
##                     Estimate  Std.Err  z-value  P(>|z|)
##    .Y                 0.898    0.127    7.071    0.000
##    .M                 1.054    0.149    7.071    0.000
##
## Defined Parameters:
##                     Estimate  Std.Err  z-value  P(>|z|)
##     ab                0.374    0.092    4.059    0.000
##     total             0.410    0.125    3.287    0.001
```

```
semPaths(fit, layout = "tree2", rotation = 2)
```



```
# The example illustrates the use of the ":=" operator in the lavaan model syntax. This operator ‘defin
```

Modification indices

```
#Large modification index ~ most likely to improve (or simply change?) fit of model

#Method 1
fit <- cfa(CFA.mod,
           data = HolzingerSwineford1939)
summary(fit, modindices = TRUE)
```

```
## lavaan (0.5-23.1097) converged normally after  28 iterations
##
##   Number of observations                           301
##
##   Estimator                                         ML
##   Minimum Function Test Statistic               90.945
##   Degrees of freedom                                26
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual =~
##     x1                1.000
##     x2                0.682    0.112    6.111    0.000
##     x3         (a)    0.986    0.091   10.897    0.000
##   textual =~
##     x4                1.000
##     x5                1.115    0.066   16.988    0.000
##     x6                0.927    0.056   16.674    0.000
##   speech =~
##     x7                1.000
##     x8         (a)    0.986    0.091   10.897    0.000
##     x9         (a)    0.986    0.091   10.897    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   visual ~~
##     textual           0.331    0.064    5.141    0.000
##     speech            0.253    0.056    4.537    0.000
##   textual ~~
##     speech            0.196    0.053    3.689    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x1                0.697    0.089    7.870    0.000
##    .x2                1.100    0.101   10.838    0.000
##    .x3                0.743    0.090    8.262    0.000
##    .x4                0.373    0.048    7.787    0.000
##    .x5                0.444    0.059    7.596    0.000
##    .x6                0.356    0.043    8.255    0.000
##    .x7                0.775    0.082    9.444    0.000
##    .x8                0.527    0.059    8.875    0.000
##    .x9                0.557    0.061    9.084    0.000
##     visual            0.607    0.100    6.072    0.000
##     textual           0.978    0.112    8.722    0.000
##     speech            0.469    0.084    5.597    0.000
##
## Modification Indices:
##
##        lhs op rhs     mi    epc sepc.lv sepc.all sepc.nox
## 1   visual =~  x1  4.835  0.412   0.321    0.281    0.281
## 7   speech =~  x7  4.835 -0.412  -0.282   -0.253   -0.253
## 27  visual =~  x4  0.960  0.078   0.061    0.053    0.053
## 28  visual =~  x5  8.058 -0.251  -0.196   -0.152   -0.152
## 29  visual =~  x6  3.726  0.147   0.114    0.104    0.104
## 30  visual =~  x7 24.613 -0.526  -0.410   -0.367   -0.367
## 31  visual =~  x8  0.267 -0.044  -0.034   -0.034   -0.034
## 32  visual =~  x9 20.904  0.390   0.304    0.302    0.302
## 33 textual =~  x1 15.212  0.305   0.302    0.264    0.264
## 34 textual =~  x2  0.017 -0.011  -0.011   -0.009   -0.009
## 35 textual =~  x3 14.778 -0.299  -0.296   -0.256   -0.256
## 36 textual =~  x7  0.732 -0.057  -0.057   -0.051   -0.051
## 37 textual =~  x8  1.208 -0.063  -0.063   -0.063   -0.063
## 38 textual =~  x9  3.483  0.109   0.107    0.107    0.107
## 39  speech =~  x1  2.887  0.220   0.150    0.132    0.132
## 40  speech =~  x2  2.880 -0.245  -0.168   -0.143   -0.143
## 41  speech =~  x3  0.446 -0.086  -0.059   -0.051   -0.051
## 42  speech =~  x4  0.006  0.006   0.004    0.004    0.004
## 43  speech =~  x5  0.310 -0.050  -0.034   -0.026   -0.026
## 44  speech =~  x6  0.246  0.038   0.026    0.024    0.024
## 45      x1 ~~  x2  0.568 -0.059  -0.059   -0.044   -0.044
## 46      x1 ~~  x3  1.664 -0.181  -0.181   -0.137   -0.137
## 47      x1 ~~  x4  4.399  0.087   0.087    0.066    0.066
## 48      x1 ~~  x5  0.166 -0.019  -0.019   -0.013   -0.013
## 49      x1 ~~  x6  0.296  0.022   0.022    0.017    0.017
## 50      x1 ~~  x7  5.919 -0.137  -0.137   -0.107   -0.107
## 51      x1 ~~  x8  0.033 -0.009  -0.009   -0.008   -0.008
## 52      x1 ~~  x9  9.367  0.151   0.151    0.131    0.131
## 53      x2 ~~  x3  3.234  0.139   0.139    0.102    0.102
## 54      x2 ~~  x4  0.502 -0.033  -0.033   -0.024   -0.024
## 55      x2 ~~  x5  0.002 -0.002  -0.002   -0.001   -0.001
## 56      x2 ~~  x6  0.633  0.035   0.035    0.027    0.027
## 57      x2 ~~  x7  9.735 -0.192  -0.192   -0.146   -0.146
## 58      x2 ~~  x8  0.084 -0.015  -0.015   -0.013   -0.013
## 59      x2 ~~  x9  1.705  0.071   0.071    0.060    0.060
## 60      x3 ~~  x4  0.243 -0.021  -0.021   -0.015   -0.015
## 61      x3 ~~  x5  8.363 -0.134  -0.134   -0.090   -0.090
## 62      x3 ~~  x6  1.350  0.047   0.047    0.037    0.037
## 63      x3 ~~  x7  0.732 -0.048  -0.048   -0.037   -0.037
## 64      x3 ~~  x8  0.147 -0.019  -0.019   -0.017   -0.017
## 65      x3 ~~  x9  2.486  0.079   0.079    0.068    0.068
## 66      x4 ~~  x5  3.025  0.218   0.218    0.145    0.145
## 67      x4 ~~  x6  6.308 -0.253  -0.253   -0.199   -0.199
## 68      x4 ~~  x7  5.580  0.096   0.096    0.074    0.074
## 69      x4 ~~  x8  3.647 -0.067  -0.067   -0.059   -0.059
## 70      x4 ~~  x9  0.289 -0.019  -0.019   -0.017   -0.017
## 71      x5 ~~  x6  0.698  0.095   0.095    0.067    0.067
## 72      x5 ~~  x7  1.335 -0.052  -0.052   -0.036   -0.036
## 73      x5 ~~  x8  0.337  0.023   0.023    0.018    0.018
## 74      x5 ~~  x9  1.207  0.044   0.044    0.034    0.034
## 75      x6 ~~  x7  0.384 -0.024  -0.024   -0.020   -0.020
## 76      x6 ~~  x8  0.285  0.018   0.018    0.017    0.017
## 77      x6 ~~  x9  0.085 -0.010  -0.010   -0.009   -0.009
## 78      x7 ~~  x8 12.962  0.212   0.212    0.192    0.192
## 79      x7 ~~  x9  6.121 -0.146  -0.146   -0.130   -0.130
## 80      x8 ~~  x9  1.896 -0.097  -0.097   -0.097   -0.097
```

```r
#Method 2
fit <- cfa(CFA.mod,
           data = HolzingerSwineford1939)
mi <- modindices(fit)
as_tibble(mi[mi$op == "=~",]) #filtering such that we only pull factor loadings
```

```
## # A tibble: 20 x 8
##        lhs    op   rhs          mi          epc      sepc.lv     sepc.all
##   *   <chr> <chr> <chr>       <dbl>        <dbl>        <dbl>        <dbl>
##   1  visual    =~    x1  4.83547889  0.412266442  0.321102198  0.281190815
##   2  speech    =~    x7  4.83535943 -0.412261350 -0.282228266 -0.253088731
##   3  visual    =~    x4  0.95994213  0.078405703  0.061067894  0.052545941
##   4  visual    =~    x5  8.05796327 -0.251067969 -0.195549452 -0.151785601
##   5  visual    =~    x6  3.72565205  0.146741842  0.114292901  0.104493337
##   6  visual    =~    x7 24.61295810 -0.526002834 -0.409688126 -0.367388602
##   7  visual    =~    x8  0.26693702 -0.043682359 -0.034022904 -0.034319874
##   8  visual    =~    x9 20.90424072  0.390298735  0.303992197  0.302029133
##   9 textual    =~    x1 15.21218566  0.305353589  0.301941367  0.264411580
## 10 textual    =~    x2  0.01744675 -0.011080418 -0.010956598 -0.009320853
## 11 textual    =~    x3 14.77776816 -0.298973602 -0.295632675 -0.256081214
## 12 textual    =~    x7  0.73229619 -0.057234975 -0.056595393 -0.050752026
## 13 textual    =~    x8  1.20814314 -0.063294635 -0.062587340 -0.063133635
## 14 textual    =~    x9  3.48311660  0.108638228  0.107424233  0.106730529
## 15  speech    =~    x1  2.88743276  0.219674753  0.150386217  0.131693969
## 16  speech    =~    x2  2.87991675 -0.244869854 -0.167634425 -0.142607754
## 17  speech    =~    x3  0.44551090 -0.085702431 -0.058670667 -0.050821364
## 18  speech    =~    x4  0.00618625  0.006366542  0.004358444  0.003750228
## 19  speech    =~    x5  0.30974779 -0.049763511 -0.034067393 -0.026443131
## 20  speech    =~    x6  0.24573191  0.038175125  0.026134148  0.023893386
## # ... with 1 more variables: sepc.nox <dbl>
```

```
as_tibble(mi[mi$op != "=~",]) #if we wanted everything except factor loadings
```

```
## # A tibble: 36 x 8
##      lhs    op   rhs         mi         epc     sepc.lv     sepc.all
##   * <chr> <chr> <chr>      <dbl>       <dbl>       <dbl>        <dbl>
##   1    x1    ~~    x2 0.56789390 -0.05857004 -0.05857004 -0.043632825
##   2    x1    ~~    x3 1.66396228 -0.18124183 -0.18124183 -0.137480649
##   3    x1    ~~    x4 4.39931777  0.08694508  0.08694508  0.065513242
##   4    x1    ~~    x5 0.16597836 -0.01862100 -0.01862100 -0.012657116
##   5    x1    ~~    x6 0.29644844  0.02156241  0.02156241  0.017263328
##   6    x1    ~~    x7 5.91930700 -0.13662370 -0.13662370 -0.107289244
##   7    x1    ~~    x8 0.03274569 -0.00877712 -0.00877712 -0.007753258
##   8    x1    ~~    x9 9.36718269  0.15100613  0.15100613  0.131382899
##   9    x2    ~~    x3 3.23363774  0.13878553  0.13878553  0.102270268
##  10    x2    ~~    x4 0.50226766 -0.03260474 -0.03260474 -0.023866398
## # ... with 26 more rows, and 1 more variables: sepc.nox <dbl>
```

Extracting information from a fitted model

```
#Summary function is for viewing, extractor function is for pulling & utilizing information
CFA.mod <- '
            visual  =~ x1 + x2 + x3
            textual =~ x4 + x5 + x6
            speed   =~ x7 + start(.5)*x8 + x9
'

fit <- cfa(CFA.mod, data=HolzingerSwineford1939)
as_tibble(parameterEstimates(fit))
```

```
## # A tibble: 24 x 9
##         lhs    op   rhs       est         se          z       pvalue
##       <chr> <chr> <chr>     <dbl>      <dbl>      <dbl>        <dbl>
##   1  visual    =~    x1 1.0000000 0.00000000        NA           NA
##   2  visual    =~    x2 0.5534993 0.09966514  5.553589 2.798624e-08
##   3  visual    =~    x3 0.7293698 0.10910982  6.684731 2.313483e-11
##   4 textual    =~    x4 1.0000000 0.00000000        NA           NA
##   5 textual    =~    x5 1.1130761 0.06542007 17.014291 0.000000e+00
##   6 textual    =~    x6 0.9261458 0.05544883 16.702714 0.000000e+00
##   7   speed    =~    x7 1.0000000 0.00000000        NA           NA
##   8   speed    =~    x8 1.1799505 0.16498665  7.151794 8.564260e-13
##   9   speed    =~    x9 1.0815302 0.15116754  7.154513 8.397727e-13
## 10      x1    ~~    x1 0.5490525 0.11360103  4.833165 1.343791e-06
## # ... with 14 more rows, and 2 more variables: ci.lower <dbl>,
## #   ci.upper <dbl>
```

```r
as_tibble(standardizedSolution(fit))
```

```
## # A tibble: 24 x 7
##        lhs    op   rhs   est.std         se         z       pvalue
##      <chr> <chr> <chr>     <dbl>      <dbl>     <dbl>        <dbl>
##  1  visual    =~    x1 0.7718808 0.05497282 14.041134 0.000000e+00
##  2  visual    =~    x2 0.4236002 0.05961916  7.105103 1.202372e-12
##  3  visual    =~    x3 0.5811319 0.05513936 10.539330 0.000000e+00
##  4 textual    =~    x4 0.8515824 0.02254309 37.775767 0.000000e+00
##  5 textual    =~    x5 0.8550654 0.02234125 38.272945 0.000000e+00
##  6 textual    =~    x6 0.8380100 0.02335518 35.881121 0.000000e+00
##  7   speed    =~    x7 0.5695145 0.05315489 10.714244 0.000000e+00
##  8   speed    =~    x8 0.7230442 0.05053062 14.309029 0.000000e+00
##  9   speed    =~    x9 0.6650092 0.05109438 13.015311 0.000000e+00
## 10      x1    ~~    x1 0.4042001 0.08486493  4.762864 1.908649e-06
## # ... with 14 more rows
```

```r
fitted.values(fit)
```

```
## $cov
##    x1    x2    x3    x4    x5    x6    x7    x8    x9   
## x1 1.358                                                
## x2 0.448 1.382                                          
## x3 0.590 0.327 1.275                                    
## x4 0.408 0.226 0.298 1.351                              
## x5 0.454 0.252 0.331 1.090 1.660                        
## x6 0.378 0.209 0.276 0.907 1.010 1.196                  
## x7 0.262 0.145 0.191 0.173 0.193 0.161 1.183            
## x8 0.309 0.171 0.226 0.205 0.228 0.190 0.453 1.022      
## x9 0.284 0.157 0.207 0.188 0.209 0.174 0.415 0.490 1.015
## 
## $mean
## x1 x2 x3 x4 x5 x6 x7 x8 x9 
##  0  0  0  0  0  0  0  0  0
```

```r
residuals(fit, type = "standardized")
```

```
## $type
## [1] "standardized"
## 
## $cov
##    x1     x2     x3     x4     x5     x6     x7     x8     x9    
## x1  0.000                                                        
## x2 -2.196  0.000                                                 
## x3 -1.199  2.692  0.000                                          
## x4  2.465 -0.283 -1.948  0.000                                   
## x5 -0.362 -0.610 -4.443  0.856  0.000                            
## x6  2.032  0.661 -0.701     NA  0.633  0.000                     
## x7 -3.787 -3.800 -1.881  0.839 -0.837 -0.321     NA              
## x8 -1.456 -1.137 -0.305 -2.049 -1.100 -0.635  3.804  0.000       
## x9  4.062  1.517  3.328  1.237  1.723  1.436 -2.771     NA  0.000
## 
```

```
## $mean
## x1 x2 x3 x4 x5 x6 x7 x8 x9
##  0  0  0  0  0  0  0  0  0
```

```
fitMeasures(fit, c("npar", "cfi"))
```

```
##   npar    cfi
## 21.000  0.931
```

```
inspect(fit, what = "start")
```

```
## $lambda
##    visual textul speed
## x1      1      0   0.0
## x2      1      0   0.0
## x3      1      0   0.0
## x4      0      1   0.0
## x5      0      1   0.0
## x6      0      1   0.0
## x7      0      0   1.0
## x8      0      0   0.5
## x9      0      0   1.0
##
## $theta
##    x1    x2    x3    x4    x5    x6    x7    x8    x9
## x1 0.679
## x2 0.000 0.691
## x3 0.000 0.000 0.637
## x4 0.000 0.000 0.000 0.675
## x5 0.000 0.000 0.000 0.000 0.830
## x6 0.000 0.000 0.000 0.000 0.000 0.598
## x7 0.000 0.000 0.000 0.000 0.000 0.000 0.592
## x8 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.511
## x9 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.508
##
## $psi
##         visual textul speed
## visual  0.05
## textual 0.00   0.05
## speed   0.00   0.00   0.05
```

```
head(inspect(fit, what = "list"))
```

```
##   id     lhs op rhs user block group free ustart exo label plabel start
## 1  1  visual =~  x1    1     1     1    0      1   0         .p1.     1
## 2  2  visual =~  x2    1     1     1    1     NA   0         .p2.     1
## 3  3  visual =~  x3    1     1     1    2     NA   0         .p3.     1
## 4  4 textual =~  x4    1     1     1    0      1   0         .p4.     1
## 5  5 textual =~  x5    1     1     1    3     NA   0         .p5.     1
## 6  6 textual =~  x6    1     1     1    4     NA   0         .p6.     1
##     est    se
## 1 1.000 0.000
## 2 0.553 0.100
## 3 0.729 0.109
## 4 1.000 0.000
## 5 1.113 0.065
## 6 0.926 0.055
```