SEM review

```r
# Load the lavaan library
library(lavaan)
```

```
## This is lavaan 0.6-6
```

```
## lavaan is BETA software! Please report any bugs.
```

```r
# =~  to define latent variables
# ~~ to define covariance and correlation
# ~ to define direct prediction
```

```r
# Look at the dataset
data(HolzingerSwineford1939)
head(HolzingerSwineford1939[ , 7:15])
```

```
##         x1   x2    x3       x4   x5        x6       x7   x8       x9
## 1 3.333333 7.75 0.375 2.333333 5.75 1.2857143 3.391304 5.75 6.361111
## 2 5.333333 5.25 2.125 1.666667 3.00 1.2857143 3.782609 6.25 7.916667
## 3 4.500000 5.25 1.875 1.000000 1.75 0.4285714 3.260870 3.90 4.416667
## 4 5.333333 7.75 3.000 2.666667 4.50 2.4285714 3.000000 5.30 4.861111
## 5 4.833333 4.75 0.875 2.666667 4.00 2.5714286 3.695652 6.30 5.916667
## 6 5.333333 5.00 2.250 1.000000 3.00 0.8571429 4.347826 6.65 7.500000
```

Define your model specification

```r
text.model <- 'textspeed =~ x4 + x5 + x6 + x7 + x8 + x9'

#model name: 'text.model',
#latent variable : 'textspeed' (1 latent var) ,
#observed variables: x4 through x9 (6 observed var)
```

Analyze the model with cfa()

```r
text.fit <- cfa(model = text.model, data = HolzingerSwineford1939)
```

```r
#Summarize the model
summary(text.fit)
```

```
## lavaan 0.6-6 ended normally after 20 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         12
##
##   Number of observations                           301
##
## Model Test User Model:
##
##   Test statistic                               149.786
##   Degrees of freedom                                 9
##   P-value (Chi-square)                           0.000
```

```
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   textspeed =~
##     x4                1.000
##     x5                1.130    0.067   16.946    0.000
##     x6                0.925    0.056   16.424    0.000
##     x7                0.196    0.067    2.918    0.004
##     x8                0.186    0.062    2.984    0.003
##     x9                0.279    0.062    4.539    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x4                0.383    0.048    7.903    0.000
##    .x5                0.424    0.059    7.251    0.000
##    .x6                0.368    0.044    8.419    0.000
##    .x7                1.146    0.094   12.217    0.000
##    .x8                0.988    0.081   12.215    0.000
##    .x9                0.940    0.077   12.142    0.000
##     textspeed         0.968    0.112    8.647    0.000
```

It is also important to examine model variances, which indicate the size of error in estimating manifest or latent variables.

You were able to view the coefficients for the model using the summary() function. However, the standardized coefficients in the Estimate column are often hard to interpret for how well they represent the latent variable.

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

```
## lavaan 0.6-6 ended normally after 20 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         12
##
##   Number of observations                           301
##
## Model Test User Model:
##
##   Test statistic                               149.786
##   Degrees of freedom                                 9
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
```

```
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   textspeed =~
##     x4                1.000                               0.984    0.846
##     x5                1.130    0.067   16.946    0.000    1.112    0.863
##     x6                0.925    0.056   16.424    0.000    0.910    0.832
##     x7                0.196    0.067    2.918    0.004    0.193    0.177
##     x8                0.186    0.062    2.984    0.003    0.183    0.181
##     x9                0.279    0.062    4.539    0.000    0.275    0.273
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x4                0.383    0.048    7.903    0.000    0.383    0.284
##    .x5                0.424    0.059    7.251    0.000    0.424    0.256
##    .x6                0.368    0.044    8.419    0.000    0.368    0.308
##    .x7                1.146    0.094   12.217    0.000    1.146    0.969
##    .x8                0.988    0.081   12.215    0.000    0.988    0.967
##    .x9                0.940    0.077   12.142    0.000    0.940    0.926
##     textspeed         0.968    0.112    8.647    0.000    1.000    1.000
```

Look at the Std.all column for the completely standardized solution to see which variables have a poor relationship to the text speed latent variable.

Looking at 'Latent Variables: Std.all', we can tell that variables 7, 8, and 9 do not measure text speed very well, as these loading coefficients are close to zero.(.177, .181, .273)

After reviewing the standardized loadings in the previous exercise, we found that several of the manifest variables may not represent our latent variable well.

As a second measure of our model, you can examine the fit indices to see if the model appropriately fits the data. You can look at both the goodness of fit and badness of fit statistics using the fit.measures argument within the summary() function.

```r
#goodness of fit and badness of fit statistics
summary(text.fit, fit.measures=TRUE )
```

```
## lavaan 0.6-6 ended normally after 20 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         12
##
##   Number of observations                           301
##
## Model Test User Model:
##
##   Test statistic                               149.786
##   Degrees of freedom                                 9
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                               681.336
##   Degrees of freedom                                15
##   P-value                                        0.000
##
## User Model versus Baseline Model:
```

```
##
##   Comparative Fit Index (CFI)                    0.789
##   Tucker-Lewis Index (TLI)                       0.648
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2476.130
##   Loglikelihood unrestricted model (H1)      -2401.237
##
##   Akaike (AIC)                                4976.261
##   Bayesian (BIC)                              5020.746
##   Sample-size adjusted Bayesian (BIC)         4982.689
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.228
##   90 Percent confidence interval - lower         0.197
##   90 Percent confidence interval - upper         0.261
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.148
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   textspeed =~
##     x4                1.000
##     x5                1.130    0.067   16.946    0.000
##     x6                0.925    0.056   16.424    0.000
##     x7                0.196    0.067    2.918    0.004
##     x8                0.186    0.062    2.984    0.003
##     x9                0.279    0.062    4.539    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x4                0.383    0.048    7.903    0.000
##    .x5                0.424    0.059    7.251    0.000
##    .x6                0.368    0.044    8.419    0.000
##    .x7                1.146    0.094   12.217    0.000
##    .x8                0.988    0.081   12.215    0.000
##    .x9                0.940    0.077   12.142    0.000
##     textspeed         0.968    0.112    8.647    0.000
```

Remember that goodness of fit statistics, like the CFI and TLI, should be large (over .90) and close to one, while badness of fit measures like the RMSEA and SRMR should be small (less than .10) and close to zero.

We can see that our fit indices are poor, with low CFI and TLI and high RMSEA and SRMR values. CFI=.789, TLI=.648, RMSE=..228, SRMR=.148

```r
#model with zero degrees of freedom
text.model1 <- 'text =~ x4 + x5 + x6'
text.fit1 <- cfa(model = text.model1, data = HolzingerSwineford1939)
summary(text.fit1)
```

```
## lavaan 0.6-6 ended normally after 15 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                          6
##
##   Number of observations                           301
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 0
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   text =~
##     x4                1.000
##     x5                1.133    0.067   16.906    0.000
##     x6                0.924    0.056   16.391    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .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
##     text              0.969    0.112    8.640    0.000
```

```r
#You should have at least one degree of freedom for any model.
```

If a model has zero degrees of freedom, it means we need to fix the model identification.

Update the model specification by setting two of the coefficient paths to 'a' to set them equal to each other.

```r
#fix model with zero degrees of freedom
text.model2 <- 'text =~ x4 + a*x5 + a*x6'
text.fit2 <- cfa(model = text.model2, data = HolzingerSwineford1939)
summary(text.fit2)
```

```
## lavaan 0.6-6 ended normally after 14 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                          6
##   Number of equality constraints                     1
##
```

```
##   Number of observations                           301
##
## Model Test User Model:
##
##   Test statistic                                11.227
##   Degrees of freedom                                 1
##   P-value (Chi-square)                           0.001
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   text =~
##     x4                1.000
##     x5         (a)    1.009    0.054   18.747    0.000
##     x6         (a)    1.009    0.054   18.747    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .x4                0.383    0.050    7.631    0.000
##    .x5                0.499    0.054    9.164    0.000
##    .x6                0.328    0.045    7.285    0.000
##     text              0.967    0.113    8.585    0.000
```

```
#two equal parameter estimates for x5 and x6
```

You have now created a two-factor model of the reading comprehension and speeded addition factors. Is that better than a one-factor model? Use the cfa() and summary() functions on your new two-factor model of the HolzingerSwineford1939 dataset to show the fit indices.

MULTIFACTOR MODEL:Two factor model

```
#two-factor model of text and speed variables
twofactor.model <- 'text =~ x4 + x5 + x6
speed =~ x7+ x8+x9'

#two-factor model of the reading comprehension and speeded addition factors
```

Is that better than a one-factor model? Use the cfa() and summary() functions to show the fit indices.

```
twofactor.fit <- cfa(model=twofactor.model, data=HolzingerSwineford1939)
summary(twofactor.fit, standardized = TRUE, fit.measures = TRUE)
```

```
## lavaan 0.6-6 ended normally after 24 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         13
##
##   Number of observations                           301
##
## Model Test User Model:
##
```

```
##   Test statistic                                14.354
##   Degrees of freedom                                 8
##   P-value (Chi-square)                           0.073
##
## Model Test Baseline Model:
##
##   Test statistic                               681.336
##   Degrees of freedom                                15
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.990
##   Tucker-Lewis Index (TLI)                       0.982
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2408.414
##   Loglikelihood unrestricted model (H1)      -2401.237
##
##   Akaike (AIC)                                4842.828
##   Bayesian (BIC)                              4891.021
##   Sample-size adjusted Bayesian (BIC)         4849.792
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.051
##   90 Percent confidence interval - lower         0.000
##   90 Percent confidence interval - upper         0.093
##   P-value RMSEA <= 0.05                          0.425
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.039
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   text =~
##     x4                1.000                               0.984    0.847
##     x5                1.132    0.067   16.954    0.000    1.114    0.865
##     x6                0.925    0.056   16.438    0.000    0.911    0.833
##   speed =~
##     x7                1.000                               0.674    0.619
##     x8                1.150    0.165    6.990    0.000    0.775    0.766
##     x9                0.878    0.123    7.166    0.000    0.592    0.587
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
```

```
##   text ~~
##     speed             0.173    0.052    3.331    0.001    0.261    0.261
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x4                0.382    0.049    7.854    0.000    0.382    0.283
##    .x5                0.418    0.059    7.113    0.000    0.418    0.252
##    .x6                0.367    0.044    8.374    0.000    0.367    0.307
##    .x7                0.729    0.084    8.731    0.000    0.729    0.616
##    .x8                0.422    0.084    5.039    0.000    0.422    0.413
##    .x9                0.665    0.071    9.383    0.000    0.665    0.655
##     text              0.969    0.112    8.647    0.000    1.000    1.000
##     speed             0.454    0.096    4.728    0.000    1.000    1.000
```

In comparing the one- and two-factor models, you should see that the fit indices are improved in the two-factor model.

Covariances: was added (amount by which 2 variables change together)

check covariance under std. all

R automatically correlates the latent variables, in order to:

Set covariance = 0, 'speed ~~ 0*visual'

Specify direct relationship between latent variables: 'speed ~ visual'

THREE-FACTOR MODEL:

Three-factor model of personality. This inventory includes 57 questions that measure extraversion, neuroticism, and lying.

Three factor model using the latent variables: extraversion, neuroticism, and lying with four manifest variables on each item.

Remember when you create multiple latent variables, these endogenous variables are automatically correlated. Set the correlation between the extraversion latent variable and neuroticism latent variable to zero, by using the ~~ in model specification code.

EPI

```r
# Load the library and data
library(psych)
```

```
##
## Attaching package: 'psych'

## The following object is masked from 'package:lavaan':
##
##     cor2cov
```

```r
epi <- read.csv("~/Downloads/epi.csv", row.names=1)

# Specify a three-factor model with correlation between extraversion and neuroticism set to zero
epi.model <- 'extraversion =~ V1 + V3 + V5 + V8
neuroticism =~ V2 + V4 + V7 + V9
lying =~ V6 + V12 + V18 + V24
extraversion ~~ 0*neuroticism'

# Run the model
epi.fit <- cfa(model = epi.model, data = epi)
```

```r
# Examine the output
summary(epi.fit, standardized = TRUE, fit.measures = TRUE)
```

```
## lavaan 0.6-6 ended normally after 118 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         26
##
##                                                   Used       Total
##   Number of observations                          3193        3570
##
## Model Test User Model:
##
##   Test statistic                               584.718
##   Degrees of freedom                                52
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                              2196.019
##   Degrees of freedom                                66
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.750
##   Tucker-Lewis Index (TLI)                       0.683
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -23208.145
##   Loglikelihood unrestricted model (H1)     -22915.787
##
##   Akaike (AIC)                               46468.291
##   Bayesian (BIC)                             46626.077
##   Sample-size adjusted Bayesian (BIC)        46543.464
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.057
##   90 Percent confidence interval - lower         0.053
##   90 Percent confidence interval - upper         0.061
##   P-value RMSEA <= 0.05                          0.004
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.058
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
```

```
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   extraversion =~
##     V1                1.000                               0.052    0.115
##     V3                1.360    0.329    4.127    0.000    0.070    0.141
##     V5               -2.829    0.554   -5.109    0.000   -0.146   -0.391
##     V8                7.315    1.832    3.992    0.000    0.377    0.797
##   neuroticism =~
##     V2                1.000                               0.228    0.457
##     V4                0.424    0.053    8.004    0.000    0.097    0.196
##     V7                1.395    0.093   15.023    0.000    0.318    0.648
##     V9                1.205    0.078   15.506    0.000    0.275    0.553
##   lying =~
##     V6                1.000                               0.135    0.272
##     V12              -0.851    0.132   -6.435    0.000   -0.115   -0.291
##     V18              -0.785    0.122   -6.421    0.000   -0.106   -0.289
##     V24               1.086    0.161    6.734    0.000    0.147    0.339
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   extraversion ~~
##     neuroticism       0.000                               0.000    0.000
##     lying            -0.002    0.001   -3.313    0.001   -0.258   -0.258
##   neuroticism ~~
##     lying            -0.014    0.002   -6.867    0.000   -0.469   -0.469
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .V1                0.198    0.005   39.567    0.000    0.198    0.987
##    .V3                0.243    0.006   39.278    0.000    0.243    0.980
##    .V5                0.118    0.005   23.900    0.000    0.118    0.847
##    .V8                0.082    0.026    3.084    0.002    0.082    0.364
##    .V2                0.197    0.006   32.516    0.000    0.197    0.791
##    .V4                0.235    0.006   38.906    0.000    0.235    0.962
##    .V7                0.140    0.007   19.412    0.000    0.140    0.580
##    .V9                0.172    0.006   26.591    0.000    0.172    0.694
##    .V6                0.228    0.007   34.520    0.000    0.228    0.926
##    .V12               0.143    0.004   33.670    0.000    0.143    0.916
##    .V18               0.124    0.004   33.753    0.000    0.124    0.917
##    .V24               0.166    0.005   31.021    0.000    0.166    0.885
##     extraversion      0.003    0.001    2.480    0.013    1.000    1.000
##     neuroticism       0.052    0.005   10.010    0.000    1.000    1.000
##     lying             0.018    0.004    4.500    0.000    1.000    1.000
```

Create a DIRECT PATH

```
#Edit the epi.model to include a direct regression path between lying and neuroticism.
#We might expect that a person's level of neuroticism would predict their level of lying.

epi.model1 <- 'extraversion =~ V1 + V3 + V5 + V8
neuroticism =~ V2 + V4 + V7 + V9
lying =~ V6 + V12 + V18 + V24
lying ~ neuroticism'                   #THIS LINE
```

```r
# Run the model
epi.fit1 <- cfa(model = epi.model1, data = epi)

# Examine the output
summary(epi.fit1, standardized = TRUE, fit.measures = TRUE)
```

```
## lavaan 0.6-6 ended normally after 120 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         26
##
##                                                   Used       Total
##   Number of observations                          3193        3570
##
## Model Test User Model:
##
##   Test statistic                               534.426
##   Degrees of freedom                                52
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                              2196.019
##   Degrees of freedom                                66
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.774
##   Tucker-Lewis Index (TLI)                       0.713
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -23183.000
##   Loglikelihood unrestricted model (H1)     -22915.787
##
##   Akaike (AIC)                               46417.999
##   Bayesian (BIC)                             46575.786
##   Sample-size adjusted Bayesian (BIC)        46493.173
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.054
##   90 Percent confidence interval - lower         0.050
##   90 Percent confidence interval - upper         0.058
##   P-value RMSEA <= 0.05                          0.058
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.053
##
## Parameter Estimates:
```

```
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   extraversion =~
##     V1                1.000                               0.052    0.115
##     V3                1.135    0.268    4.230    0.000    0.059    0.118
##     V5               -2.497    0.443   -5.638    0.000   -0.129   -0.346
##     V8                8.223    2.008    4.096    0.000    0.425    0.898
##   neuroticism =~
##     V2                1.000                               0.223    0.447
##     V4                0.462    0.054    8.493    0.000    0.103    0.209
##     V7                1.435    0.093   15.368    0.000    0.320    0.652
##     V9                1.214    0.078   15.570    0.000    0.271    0.545
##   lying =~
##     V6                1.000                               0.125    0.252
##     V12              -0.943    0.150   -6.274    0.000   -0.118   -0.298
##     V18              -0.905    0.143   -6.339    0.000   -0.113   -0.308
##     V24               1.187    0.182    6.509    0.000    0.148    0.342
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   lying ~
##     neuroticism      -0.298    0.043   -6.943    0.000   -0.532   -0.532
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   extraversion ~~
##     neuroticism       0.003    0.001    3.761    0.000    0.240    0.240
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .V1                0.198    0.005   39.671    0.000    0.198    0.987
##    .V3                0.244    0.006   39.651    0.000    0.244    0.986
##    .V5                0.123    0.004   28.256    0.000    0.123    0.881
##    .V8                0.043    0.033    1.302    0.193    0.043    0.193
##    .V2                0.200    0.006   33.262    0.000    0.200    0.800
##    .V4                0.233    0.006   38.804    0.000    0.233    0.956
##    .V7                0.139    0.007   20.087    0.000    0.139    0.575
##    .V9                0.174    0.006   27.907    0.000    0.174    0.703
##    .V6                0.231    0.007   35.398    0.000    0.231    0.936
##    .V12               0.143    0.004   33.349    0.000    0.143    0.911
##    .V18               0.122    0.004   32.825    0.000    0.122    0.905
##    .V24               0.166    0.005   30.854    0.000    0.166    0.883
##     extraversion      0.003    0.001    2.643    0.008    1.000    1.000
##     neuroticism       0.050    0.005    9.947    0.000    1.000    1.000
##    .lying             0.011    0.003    3.970    0.000    0.717    0.717
```

UPDATING POOR MODELS:

if model has CFI and TLI below our criteria (.9) also if bad fit indices RMSEA and SRMS is higher tan criteria (.1)

```r
#CHECK MODEL VARIANCE

#In order to evaluate your three-factor model of the epi, you can examine the variance of the
#manifest(observable) variables to check for potential problems with the model. Very large variances
#can indicate potential issues; however, this value should be compared to the original scale of the dat
# Calculate the variance of V1

var(epi$V1) #0.2017972
```

```
## [1] NA
```

```r
#You can see that your variance from the model (0.199) is very similar to the real variance (0.201)
#which indicates our model does not have variance issues.
```

```r
#Examine MODIFICATION INDICES
#The fit indices for our epi.model are low (in the .70s) for CFI and TLI.
#You can use modification indices to find potential parameters (paths) to add to the model specificatio
#to improve model fit.

# Original model summary
summary(epi.fit, standardized = TRUE, fit.measures = TRUE)
```

```
## lavaan 0.6-6 ended normally after 118 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         26
##
##                                                   Used       Total
##   Number of observations                          3193        3570
##
## Model Test User Model:
##
##   Test statistic                               584.718
##   Degrees of freedom                                52
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                              2196.019
##   Degrees of freedom                                66
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.750
##   Tucker-Lewis Index (TLI)                       0.683
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -23208.145
##   Loglikelihood unrestricted model (H1)     -22915.787
##
##   Akaike (AIC)                               46468.291
##   Bayesian (BIC)                             46626.077
```

```
##   Sample-size adjusted Bayesian (BIC)       46543.464
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.057
##   90 Percent confidence interval - lower         0.053
##   90 Percent confidence interval - upper         0.061
##   P-value RMSEA <= 0.05                          0.004
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.058
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   extraversion =~
##     V1                1.000                               0.052    0.115
##     V3                1.360    0.329    4.127    0.000    0.070    0.141
##     V5               -2.829    0.554   -5.109    0.000   -0.146   -0.391
##     V8                7.315    1.832    3.992    0.000    0.377    0.797
##   neuroticism =~
##     V2                1.000                               0.228    0.457
##     V4                0.424    0.053    8.004    0.000    0.097    0.196
##     V7                1.395    0.093   15.023    0.000    0.318    0.648
##     V9                1.205    0.078   15.506    0.000    0.275    0.553
##   lying =~
##     V6                1.000                               0.135    0.272
##     V12              -0.851    0.132   -6.435    0.000   -0.115   -0.291
##     V18              -0.785    0.122   -6.421    0.000   -0.106   -0.289
##     V24               1.086    0.161    6.734    0.000    0.147    0.339
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   extraversion ~~
##     neuroticism       0.000                               0.000    0.000
##     lying            -0.002    0.001   -3.313    0.001   -0.258   -0.258
##   neuroticism ~~
##     lying            -0.014    0.002   -6.867    0.000   -0.469   -0.469
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .V1                0.198    0.005   39.567    0.000    0.198    0.987
##    .V3                0.243    0.006   39.278    0.000    0.243    0.980
##    .V5                0.118    0.005   23.900    0.000    0.118    0.847
##    .V8                0.082    0.026    3.084    0.002    0.082    0.364
##    .V2                0.197    0.006   32.516    0.000    0.197    0.791
##    .V4                0.235    0.006   38.906    0.000    0.235    0.962
##    .V7                0.140    0.007   19.412    0.000    0.140    0.580
```

```
##    .V9                0.172    0.006    26.591    0.000    0.172    0.694
##    .V6                0.228    0.007    34.520    0.000    0.228    0.926
##    .V12               0.143    0.004    33.670    0.000    0.143    0.916
##    .V18               0.124    0.004    33.753    0.000    0.124    0.917
##    .V24               0.166    0.005    31.021    0.000    0.166    0.885
##     extraversion      0.003    0.001     2.480    0.013    1.000    1.000
##     neuroticism       0.052    0.005    10.010    0.000    1.000    1.000
##     lying             0.018    0.004     4.500    0.000    1.000    1.000
```

```r
# Examine the modification indices
modificationindices(epi.fit, sort=TRUE)
```

```
##              lhs op          rhs      mi    epc sepc.lv sepc.all sepc.nox
## 40   neuroticism =~           V3 152.701 -0.609  -0.139   -0.279   -0.279
## 39   neuroticism =~           V1 122.735  0.493   0.112    0.251    0.251
## 48         lying =~           V3 121.175  1.269   0.171    0.345    0.345
## 58            V1 ~~           V2  76.218  0.032   0.032    0.164    0.164
## 70            V3 ~~           V7  71.613 -0.033  -0.033   -0.178   -0.178
## 13  extraversion ~~  neuroticism  70.230  0.003   0.236    0.236    0.236
## 42   neuroticism =~           V8  68.905  0.372   0.085    0.179    0.179
## 47         lying =~           V1  62.368 -0.819  -0.111   -0.247   -0.247
## 50         lying =~           V8  56.929 -1.095  -0.148   -0.313   -0.313
## 87            V8 ~~           V7  38.504  0.022   0.022    0.203    0.203
## 33  extraversion =~           V7  30.415  1.034   0.053    0.109    0.109
## 59            V1 ~~           V4  28.442  0.021   0.021    0.095    0.095
## 32  extraversion =~           V4  27.525  1.079   0.056    0.113    0.113
## 75            V3 ~~          V24  20.299  0.017   0.017    0.084    0.084
## 52         lying =~           V4  18.610 -0.618  -0.084   -0.169   -0.169
## 103           V4 ~~          V12  17.780  0.014   0.014    0.078    0.078
## 86            V8 ~~           V4  15.339  0.015   0.015    0.109    0.109
## 113           V9 ~~          V18  15.043  0.012   0.012    0.081    0.081
## 53         lying =~           V7  13.292 -0.567  -0.077   -0.156   -0.156
## 35  extraversion =~           V6  10.893 -0.816  -0.042   -0.085   -0.085
## 76            V5 ~~           V8   9.434  0.103   0.103    1.046    1.046
## 116           V6 ~~          V18   9.357  0.011   0.011    0.067    0.067
## 45   neuroticism =~          V18   9.199  0.178   0.041    0.111    0.111
## 74            V3 ~~          V18   8.727 -0.009  -0.009   -0.054   -0.054
## 64            V1 ~~          V18   8.624  0.008   0.008    0.054    0.054
## 68            V3 ~~           V2   8.157 -0.012  -0.012   -0.054   -0.054
## 99            V2 ~~          V24   7.503  0.010   0.010    0.055    0.055
## 51         lying =~           V2   7.304  0.389   0.053    0.105    0.105
## 84            V5 ~~          V24   7.237  0.008   0.008    0.054    0.054
## 89            V8 ~~           V6   6.987 -0.011  -0.011   -0.084   -0.084
## 66            V3 ~~           V5   6.798 -0.010  -0.010   -0.060   -0.060
## 107           V7 ~~           V6   6.068 -0.010  -0.010   -0.057   -0.057
## 61            V1 ~~           V9   6.029  0.009   0.009    0.048    0.048
## 111           V9 ~~           V6   5.999  0.010   0.010    0.051    0.051
## 46   neuroticism =~          V24   5.729  0.180   0.041    0.095    0.095
## 71            V3 ~~           V9   5.614 -0.009  -0.009   -0.046   -0.046
## 54         lying =~           V9   5.263 -0.339  -0.046   -0.092   -0.092
## 56            V1 ~~           V5   5.014  0.007   0.007    0.047    0.047
## 57            V1 ~~           V8   4.821  0.017   0.017    0.136    0.136
## 60            V1 ~~           V7   4.784  0.008   0.008    0.046    0.046
## 117           V6 ~~          V24   4.689  0.010   0.010    0.051    0.051
## 34  extraversion =~           V9   4.329  0.401   0.021    0.042    0.042
```

```
## 69            V3 ~~         V4   3.827  0.008   0.008    0.035    0.035
## 37  extraversion =~        V18   3.057 -0.325  -0.017   -0.046   -0.046
## 106           V7 ~~         V9   2.624 -0.017  -0.017   -0.112   -0.112
## 83            V5 ~~        V18   2.479  0.004   0.004    0.031    0.031
## 96            V2 ~~         V6   2.361  0.006   0.006    0.030    0.030
## 88            V8 ~~         V9   2.253  0.005   0.005    0.046    0.046
## 94            V2 ~~         V7   2.142  0.012   0.012    0.071    0.071
## 92            V8 ~~        V24   2.050  0.006   0.006    0.049    0.049
## 55            V1 ~~         V3   1.617 -0.005  -0.005   -0.023   -0.023
## 43   neuroticism =~         V6   1.585  0.098   0.022    0.045    0.045
## 49         lying =~         V5   1.582  0.116   0.016    0.042    0.042
## 98            V2 ~~        V18   1.192 -0.003  -0.003   -0.022   -0.022
## 65            V1 ~~        V24   1.135  0.004   0.004    0.020    0.020
## 120          V18 ~~        V24   1.004 -0.003  -0.003   -0.024   -0.024
## 110           V7 ~~        V24   0.949  0.004   0.004    0.024    0.024
## 114           V9 ~~        V24   0.942 -0.004  -0.004   -0.021   -0.021
## 63            V1 ~~        V12   0.922  0.003   0.003    0.018    0.018
## 115           V6 ~~        V12   0.905  0.004   0.004    0.021    0.021
## 81            V5 ~~         V6   0.722  0.003   0.003    0.016    0.016
## 100           V4 ~~         V7   0.697 -0.004  -0.004   -0.022   -0.022
## 38  extraversion =~        V24   0.639  0.185   0.010    0.022    0.022
## 44   neuroticism =~        V12   0.585  0.049   0.011    0.028    0.028
## 62            V1 ~~         V6   0.573  0.003   0.003    0.014    0.014
## 80            V5 ~~         V9   0.511 -0.002  -0.002   -0.014   -0.014
## 119          V12 ~~        V24   0.501 -0.003  -0.003   -0.017   -0.017
## 95            V2 ~~         V9   0.439  0.004   0.004    0.024    0.024
## 101           V4 ~~         V9   0.432 -0.003  -0.003   -0.014   -0.014
## 93            V2 ~~         V4   0.420  0.003   0.003    0.013    0.013
## 41   neuroticism =~         V5   0.401 -0.022  -0.005   -0.014   -0.014
## 72            V3 ~~         V6   0.398 -0.003  -0.003   -0.012   -0.012
## 78            V5 ~~         V4   0.355  0.002   0.002    0.011    0.011
## 77            V5 ~~         V2   0.290  0.002   0.002    0.010    0.010
## 36  extraversion =~        V12   0.273 -0.105  -0.005   -0.014   -0.014
## 85            V8 ~~         V2   0.267  0.002   0.002    0.015    0.015
## 105           V4 ~~        V24   0.227  0.002   0.002    0.009    0.009
## 31  extraversion =~         V2   0.206  0.090   0.005    0.009    0.009
## 91            V8 ~~        V18   0.191 -0.001  -0.001   -0.014   -0.014
## 102           V4 ~~         V6   0.158 -0.002  -0.002   -0.007   -0.007
## 97            V2 ~~        V12   0.143 -0.001  -0.001   -0.007   -0.007
## 73            V3 ~~        V12   0.130 -0.001  -0.001   -0.007   -0.007
## 82            V5 ~~        V12   0.115  0.001   0.001    0.007    0.007
## 118          V12 ~~        V18   0.109 -0.001  -0.001   -0.007   -0.007
## 90            V8 ~~        V12   0.107 -0.001  -0.001   -0.011   -0.011
## 67            V3 ~~         V8   0.102 -0.003  -0.003   -0.022   -0.022
## 79            V5 ~~         V7   0.059 -0.001  -0.001   -0.005   -0.005
## 112           V9 ~~        V12   0.054 -0.001  -0.001   -0.005   -0.005
## 108           V7 ~~        V12   0.023 -0.001  -0.001   -0.004   -0.004
## 104           V4 ~~        V18   0.011  0.000   0.000   -0.002   -0.002
## 109           V7 ~~        V18   0.000  0.000   0.000    0.000    0.000
```

```
#Update the model specification code to include the largest mi value.
# Edit the model specification
epi.model2 <- 'extraversion =~ V1 + V3 + V5 + V8
neuroticism =~ V2 + V4 + V7 + V9
```

```
lying =~ V6 + V12 + V18 + V24
neuroticism =~ V3'
```

```
# Reanalyze the model
epi.fit2 <- cfa(model = epi.model2, data = epi)

# Summarize the updated model
summary(epi.fit2, standardized = TRUE, fit.measures = TRUE)
```

```
## lavaan 0.6-6 ended normally after 126 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         28
##
##                                                   Used       Total
##   Number of observations                          3193        3570
##
## Model Test User Model:
##
##   Test statistic                               332.891
##   Degrees of freedom                                50
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                              2196.019
##   Degrees of freedom                                66
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.867
##   Tucker-Lewis Index (TLI)                       0.825
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -23082.232
##   Loglikelihood unrestricted model (H1)     -22915.787
##
##   Akaike (AIC)                               46220.465
##   Bayesian (BIC)                             46390.389
##   Sample-size adjusted Bayesian (BIC)        46301.421
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.042
##   90 Percent confidence interval - lower         0.038
##   90 Percent confidence interval - upper         0.046
##   P-value RMSEA <= 0.05                          0.999
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.040
```

```
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   extraversion =~
##     V1                1.000                               0.068    0.152
##     V3                1.798    0.325    5.532    0.000    0.123    0.246
##     V5               -2.268    0.360   -6.291    0.000   -0.155   -0.414
##     V8                5.077    0.887    5.725    0.000    0.346    0.732
##   neuroticism =~
##     V2                1.000                               0.222    0.445
##     V4                0.432    0.053    8.134    0.000    0.096    0.194
##     V7                1.493    0.093   16.025    0.000    0.331    0.675
##     V9                1.186    0.074   15.938    0.000    0.263    0.530
##   lying =~
##     V6                1.000                               0.135    0.272
##     V12              -0.851    0.127   -6.699    0.000   -0.115   -0.290
##     V18              -0.799    0.119   -6.728    0.000   -0.108   -0.294
##     V24               1.115    0.157    7.087    0.000    0.151    0.347
##   neuroticism =~
##     V3               -0.732    0.066  -11.074    0.000   -0.163   -0.327
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   extraversion ~~
##     neuroticism       0.004    0.001    4.953    0.000    0.283    0.283
##     lying            -0.003    0.001   -4.380    0.000   -0.346   -0.346
##   neuroticism ~~
##     lying            -0.016    0.002   -7.337    0.000   -0.521   -0.521
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .V1                0.196    0.005   39.250    0.000    0.196    0.977
##    .V3                0.217    0.006   34.642    0.000    0.217    0.878
##    .V5                0.116    0.004   29.066    0.000    0.116    0.828
##    .V8                0.104    0.014    7.603    0.000    0.104    0.465
##    .V2                0.200    0.006   33.875    0.000    0.200    0.802
##    .V4                0.235    0.006   39.046    0.000    0.235    0.962
##    .V7                0.131    0.007   19.577    0.000    0.131    0.544
##    .V9                0.178    0.006   29.830    0.000    0.178    0.720
##    .V6                0.228    0.007   34.969    0.000    0.228    0.926
##    .V12               0.144    0.004   34.186    0.000    0.144    0.916
##    .V18               0.123    0.004   34.035    0.000    0.123    0.914
##    .V24               0.166    0.005   31.188    0.000    0.166    0.879
##     extraversion      0.005    0.001    3.265    0.001    1.000    1.000
##     neuroticism       0.049    0.005   10.127    0.000    1.000    1.000
##     lying             0.018    0.004    4.651    0.000    1.000    1.000
```

Your fit indices should improve to the .80s by including this one extra parameter to the model.

(Now, CFI= 0.867, TLI=0.825)

(Before, CFI= 0.750, TLI =0.683)

#COMPARE TWO MODELS

The original model epi.model and the updated model with the modified path epi.model2 can now be compared using the anova() function to determine if the change in fit indices was a large change.

We can use the anova() function because these models are nested, which means they are the same manifest variables with different parameters.

```
# Analyze the original model
epi.fit <- cfa(model = epi.model, data =epi)

# Analyze the updated model
epi.fit2 <- cfa(model = epi.model2, data = epi)

# Compare those models
anova(epi.fit,epi.fit2)
```

```
## Chi-Squared Difference Test
##
##          Df   AIC   BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## epi.fit2 50 46220 46390 332.89
## epi.fit  52 46468 46626 584.72     251.83       2  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The updated model appears better, as the chi-square difference test is significant. (***)

```
#Select Specific Fit Indices
#You can also compare models by using the AIC or ECVI fit indices, rather than the anova() function.
#These fit indices are very useful if your models include different manifest variables.
#When comparing sets of AIC or ECVI values, the best model would have the smallest fit index.

# Find the fit indices for the original model
fitmeasures(epi.fit, c('aic', 'ecvi'))
```

```
##       aic      ecvi
## 46468.291     0.199
```

```
# Find the fit indices for the updated model
fitmeasures(epi.fit2, c('aic', 'ecvi'))
```

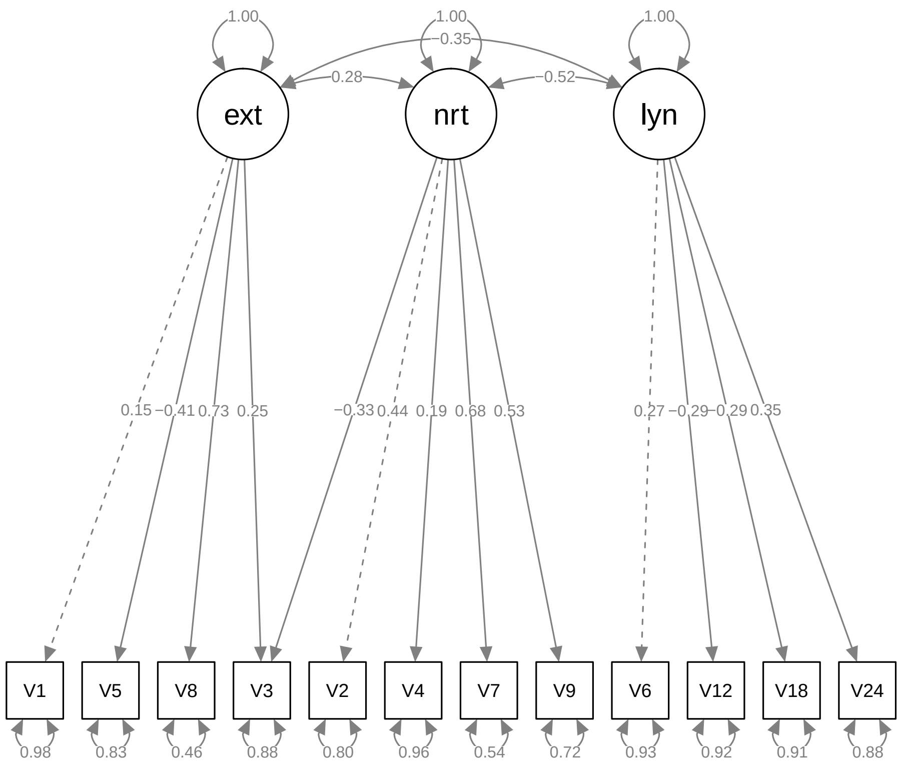
```
##       aic      ecvi
## 46220.465     0.122
```

For both AIC and ECVI, the updated model included the smaller fit indices and would be considered the better model.

```
library(semPlot)
```

```
## Registered S3 methods overwritten by 'huge':
##   method    from
##   plot.sim  BDgraph
##   print.sim BDgraph
```

```
semPaths(epi.fit2, whatLabels = 'std', rotation=1)
```



#EXAMPLE:LOCAL ANIMAL SHELTER A local animal shelter has designed a survey to measure the impact of their Adopt Me program. Viewers rated each dog's picture, background story, and other characteristics to indicate the "adoptableness" of each animal.

The adoptsurvey data contains the six items they rated including pictures, background, loveskids that measure a "good story" latent variable, while energy, wagstail, playful measure an "in person" latent variable. We will build a two-factor model of their survey and examine it for Heywood cases.

```
library(data.table)
library(curl)
adoptsurvey <- fread('https://raw.githubusercontent.com/JiaxiangBU/picbackup/master/adoptsurvey02.csv')
```

```
head(adoptsurvey)
```

```
##     pictures background loveskids    energy   wagstail  playful
## 1:  3.708400 -0.9640867  3.859116 -6.728699 -1.1995000 4.097103
## 2:  1.244440  6.3804313  5.951090  1.606351  0.5322139 1.925454
## 3:  1.192845 -4.3286503  8.231443  4.090618  4.5900018 4.035844
## 4: -1.260835  5.1964583  2.457856  7.596427  3.6990812 4.559570
## 5:  4.575658 -0.1453078  9.527073 -3.134994  2.5460263 3.432766
## 6:  1.959739  6.6615860  5.619911  1.289012  3.3453336 9.074500
```

```
str(adoptsurvey)
```

```
## Classes 'data.table' and 'data.frame':   100 obs. of  6 variables:
##  $ pictures  : num  3.71 1.24 1.19 -1.26 4.58 ...
##  $ background: num  -0.964 6.38 -4.329 5.196 -0.145 ...
##  $ loveskids : num  3.86 5.95 8.23 2.46 9.53 ...
##  $ energy    : num  -6.73 1.61 4.09 7.6 -3.13 ...
##  $ wagstail  : num  -1.199 0.532 4.59 3.699 2.546 ...
##  $ playful   : num  4.1 1.93 4.04 4.56 3.43 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```r
# Build the model
adopt.model <- 'goodstory =~ pictures + background + loveskids
inperson =~ energy + wagstail + playful'

# Analyze the model
adopt.fit <- cfa(model = adopt.model, data = adoptsurvey)
```

```
## Warning in lav_object_post_check(object): lavaan WARNING: some estimated ov
## variances are negative
```

we see an error message warning you that the latent variables are not positive definite.

So, correlation > 1 on the latent variable.

You should fix the Heywood case by collapsing the two latent variables into one latent variable.

```r
#create only one goodstory factor that is measured by all six manifest variables in the adoptsurvey dat

# Edit the original model
adopt.model <- 'goodstory =~ pictures + background + loveskids + energy + wagstail + playful'

# Analyze the model
adopt.fit <- cfa(model = adopt.model, data = adoptsurvey)

# Look for Heywood cases
summary(adopt.fit, standardized = TRUE, fit.measures = TRUE)
```

```
## lavaan 0.6-6 ended normally after 56 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         12
##
##   Number of observations                           100
##
## Model Test User Model:
##
##   Test statistic                                 9.627
##   Degrees of freedom                                 9
##   P-value (Chi-square)                           0.382
##
## Model Test Baseline Model:
##
##   Test statistic                                25.380
##   Degrees of freedom                                15
##   P-value                                        0.045
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.940
##   Tucker-Lewis Index (TLI)                       0.899
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -1651.202
##   Loglikelihood unrestricted model (H1)      -1646.389
```

```
##
##   Akaike (AIC)                                3326.404
##   Bayesian (BIC)                              3357.666
##   Sample-size adjusted Bayesian (BIC)         3319.767
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.026
##   90 Percent confidence interval - lower         0.000
##   90 Percent confidence interval - upper         0.117
##   P-value RMSEA <= 0.05                          0.569
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.061
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   goodstory =~
##     pictures          1.000                               1.343    0.432
##     background        1.468    0.756    1.942    0.052    1.972    0.513
##     loveskids         1.815    0.936    1.939    0.052    2.438    0.515
##     energy            0.067    0.380    0.177    0.859    0.090    0.025
##     wagstail         -0.306    0.521   -0.588    0.556   -0.412   -0.086
##     playful          -0.009    0.356   -0.025    0.980   -0.012   -0.004
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .pictures          7.860    1.503    5.228    0.000    7.860    0.813
##    .background       10.873    2.659    4.089    0.000   10.873    0.737
##    .loveskids        16.491    4.052    4.069    0.000   16.491    0.735
##    .energy           12.677    1.794    7.066    0.000   12.677    0.999
##    .wagstail         22.674    3.232    7.016    0.000   22.674    0.993
##    .playful          11.181    1.581    7.071    0.000   11.181    1.000
##     goodstory         1.804    1.287    1.402    0.161    1.000    1.000
```

You will look for a Heywood cases on one of the manifest variables, rather than on the latent variable.

(negative variance)

```r
# Build the model
adopt.model <- 'goodstory =~ pictures + background + loveskids
inperson =~ energy + wagstail + playful'

# Analyze the model and include the data argument
adopt.fit <- cfa(adopt.model, adoptsurvey)
```

```
## Warning in lav_object_post_check(object): lavaan WARNING: some estimated ov
## variances are negative
```

```r
# Summarize the model to view the negative variances
summary(adopt.fit, standardized=TRUE, fit.measures = TRUE)
```

```
## lavaan 0.6-6 ended normally after 300 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         13
##
##   Number of observations                           100
##
## Model Test User Model:
##
##   Test statistic                                 7.134
##   Degrees of freedom                                 8
##   P-value (Chi-square)                           0.522
##
## Model Test Baseline Model:
##
##   Test statistic                                25.380
##   Degrees of freedom                                15
##   P-value                                        0.045
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    1.000
##   Tucker-Lewis Index (TLI)                       1.156
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -1649.956
##   Loglikelihood unrestricted model (H1)      -1646.389
##
##   Akaike (AIC)                                3325.912
##   Bayesian (BIC)                              3359.779
##   Sample-size adjusted Bayesian (BIC)         3318.722
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.000
##   90 Percent confidence interval - lower         0.000
##   90 Percent confidence interval - upper         0.109
##   P-value RMSEA <= 0.05                          0.686
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.050
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
```

```
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   goodstory =~
##     pictures          1.000                               1.360    0.437
##     background        1.471    0.763    1.928    0.054    2.000    0.521
##     loveskids         1.746    0.892    1.958    0.050    2.375    0.501
##   inperson =~
##     energy            1.000                               0.208    0.058
##     wagstail         45.278 1090.877    0.042    0.967    9.410    1.969
##     playful           0.869    1.110    0.783    0.434    0.181    0.054
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   goodstory ~~
##     inperson         -0.014    0.332   -0.041    0.967   -0.048   -0.048
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .pictures          7.814    1.514    5.162    0.000    7.814    0.809
##    .background       10.762    2.695    3.993    0.000   10.762    0.729
##    .loveskids        16.791    3.936    4.266    0.000   16.791    0.749
##    .energy           12.642    2.066    6.119    0.000   12.642    0.997
##    .wagstail        -65.707 2125.647   -0.031    0.975  -65.707   -2.876
##    .playful          11.148    1.760    6.335    0.000   11.148    0.997
##     goodstory         1.850    1.310    1.411    0.158    1.000    1.000
##     inperson          0.043    1.046    0.041    0.967    1.000    1.000
```

we can see variance is negative for wagstail variable, which is a Heywood case. (-65.707)

HEIWOOD CASES=> Correlations that are out of bounds, Negative variances

Fix the Manifest Heywood Model:

To fix the error in the last model, we can use the var() function to calculate the variance of the manifest variable that is estimated as negative.

```r
# Summarize the model to view the negative variances
summary(adopt.fit, standardized = TRUE, fit.measures = TRUE, rsquare=TRUE)
```

```
## lavaan 0.6-6 ended normally after 300 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         13
##
##   Number of observations                           100
##
## Model Test User Model:
##
##   Test statistic                                 7.134
##   Degrees of freedom                                 8
##   P-value (Chi-square)                           0.522
##
## Model Test Baseline Model:
##
##   Test statistic                                25.380
##   Degrees of freedom                                15
```

```
##   P-value                                       0.045
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                   1.000
##   Tucker-Lewis Index (TLI)                      1.156
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -1649.956
##   Loglikelihood unrestricted model (H1)     -1646.389
##
##   Akaike (AIC)                               3325.912
##   Bayesian (BIC)                             3359.779
##   Sample-size adjusted Bayesian (BIC)        3318.722
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                         0.000
##   90 Percent confidence interval - lower        0.000
##   90 Percent confidence interval - upper        0.109
##   P-value RMSEA <= 0.05                         0.686
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                          0.050
##
## Parameter Estimates:
##
##   Standard errors                            Standard
##   Information                                Expected
##   Information saturated (h1) model         Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   goodstory =~
##     pictures          1.000                               1.360    0.437
##     background        1.471    0.763    1.928    0.054    2.000    0.521
##     loveskids         1.746    0.892    1.958    0.050    2.375    0.501
##   inperson =~
##     energy            1.000                               0.208    0.058
##     wagstail         45.278 1090.877    0.042    0.967    9.410    1.969
##     playful           0.869    1.110    0.783    0.434    0.181    0.054
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   goodstory ~~
##     inperson         -0.014    0.332   -0.041    0.967   -0.048   -0.048
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .pictures          7.814    1.514    5.162    0.000    7.814    0.809
##    .background       10.762    2.695    3.993    0.000   10.762    0.729
##    .loveskids        16.791    3.936    4.266    0.000   16.791    0.749
```

```
##    .energy           12.642    2.066    6.119    0.000   12.642    0.997
##    .wagstail        -65.707 2125.647   -0.031    0.975  -65.707   -2.876
##    .playful          11.148    1.760    6.335    0.000   11.148    0.997
##     goodstory         1.850    1.310    1.411    0.158    1.000    1.000
##     inperson          0.043    1.046    0.041    0.967    1.000    1.000
##
## R-Square:
##                    Estimate
##     pictures          0.191
##     background        0.271
##     loveskids         0.251
##     energy            0.003
##     wagstail             NA
##     playful           0.003
```

```
# View the variance of the problem manifest variable
var(adoptsurvey$wagstail)
```

```
## [1] 23.07446
```

```
# Update the model using 5 decimal places
adopt.model2 <- 'goodstory =~ pictures + background + loveskids
inperson =~ energy + wagstail + playful
wagstail ~~ 23.07446 * wagstail'              #THIS LINE

# Analyze and summarize the updated model
adopt.fit2 <- cfa(model = adopt.model2, data = adoptsurvey)

# Summarize the model to view the negative variances
summary(adopt.fit2, standardized = TRUE, fit.measures = TRUE, rsquare=TRUE)
```

```
## lavaan 0.6-6 ended normally after 69 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         12
##
##   Number of observations                           100
##
## Model Test User Model:
##
##   Test statistic                                 8.493
##   Degrees of freedom                                 9
##   P-value (Chi-square)                           0.485
##
## Model Test Baseline Model:
##
##   Test statistic                                25.380
##   Degrees of freedom                                15
##   P-value                                        0.045
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    1.000
##   Tucker-Lewis Index (TLI)                       1.081
##
```

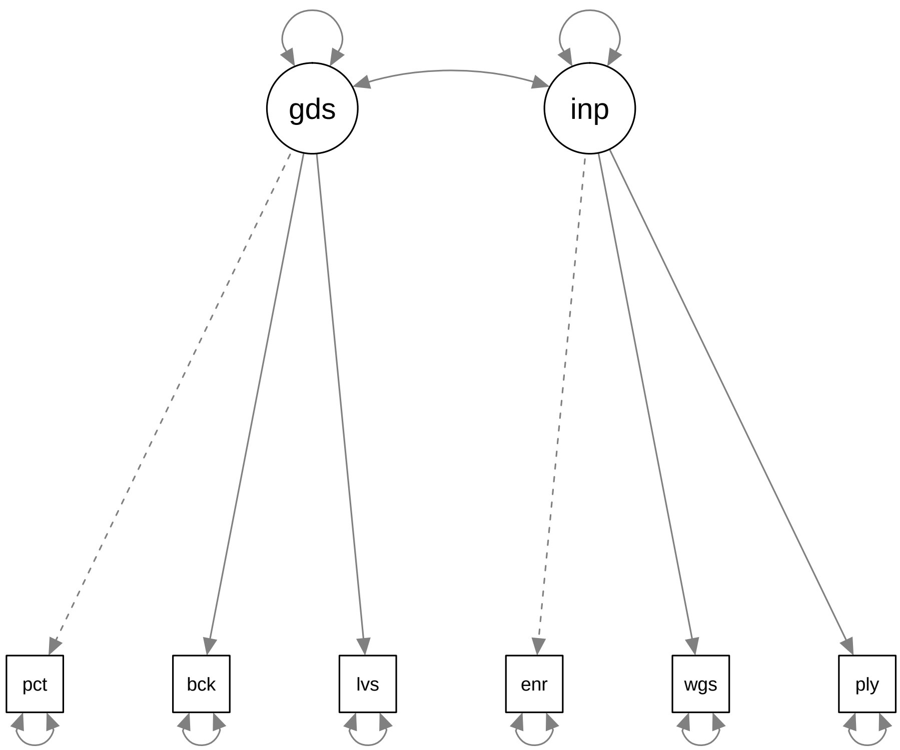
```
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -1650.635
##   Loglikelihood unrestricted model (H1)     -1646.389
##
##   Akaike (AIC)                               3325.270
##   Bayesian (BIC)                             3356.532
##   Sample-size adjusted Bayesian (BIC)        3318.633
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                         0.000
##   90 Percent confidence interval - lower        0.000
##   90 Percent confidence interval - upper        0.108
##   P-value RMSEA <= 0.05                         0.664
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                          0.058
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   goodstory =~
##     pictures          1.000                               1.344    0.432
##     background        1.461    0.758    1.928    0.054    1.964    0.511
##     loveskids         1.818    0.947    1.919    0.055    2.444    0.516
##   inperson =~
##     energy            1.000                               0.959    0.269
##     wagstail          1.391    2.244    0.620    0.535    1.334    0.268
##     playful           0.807    1.640    0.492    0.623    0.774    0.231
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   goodstory ~~
##     inperson         -0.077    0.450   -0.172    0.863   -0.060   -0.060
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .wagstail         23.074                              23.074    0.928
##    .pictures          7.857    1.510    5.203    0.000    7.857    0.813
##    .background       10.906    2.672    4.082    0.000   10.906    0.739
##    .loveskids        16.461    4.103    4.012    0.000   16.461    0.734
##    .energy           11.765    2.683    4.385    0.000   11.765    0.928
##    .playful          10.582    2.082    5.084    0.000   10.582    0.946
##     goodstory         1.807    1.296    1.395    0.163    1.000    1.000
##     inperson          0.920    2.209    0.416    0.677    1.000    1.000
##
## R-Square:
```

```
##                  Estimate
##     wagstail        0.072
##     pictures        0.187
##     background      0.261
##     loveskids       0.266
##     energy          0.072
##     playful         0.054
```

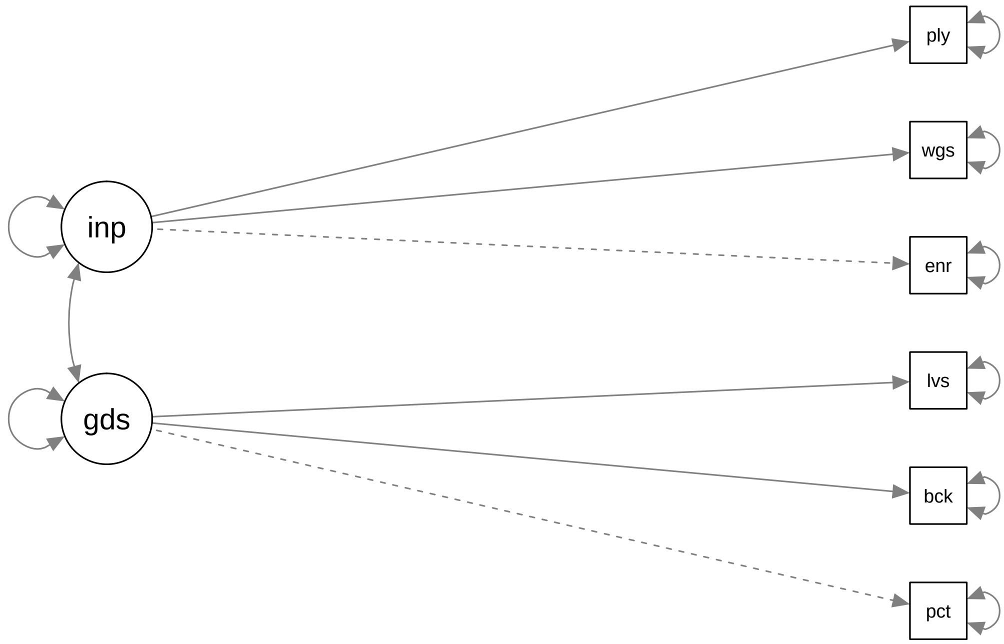
problem fixed!

CREATE DIAGRAMS w/ semPlot library and semPaths() function

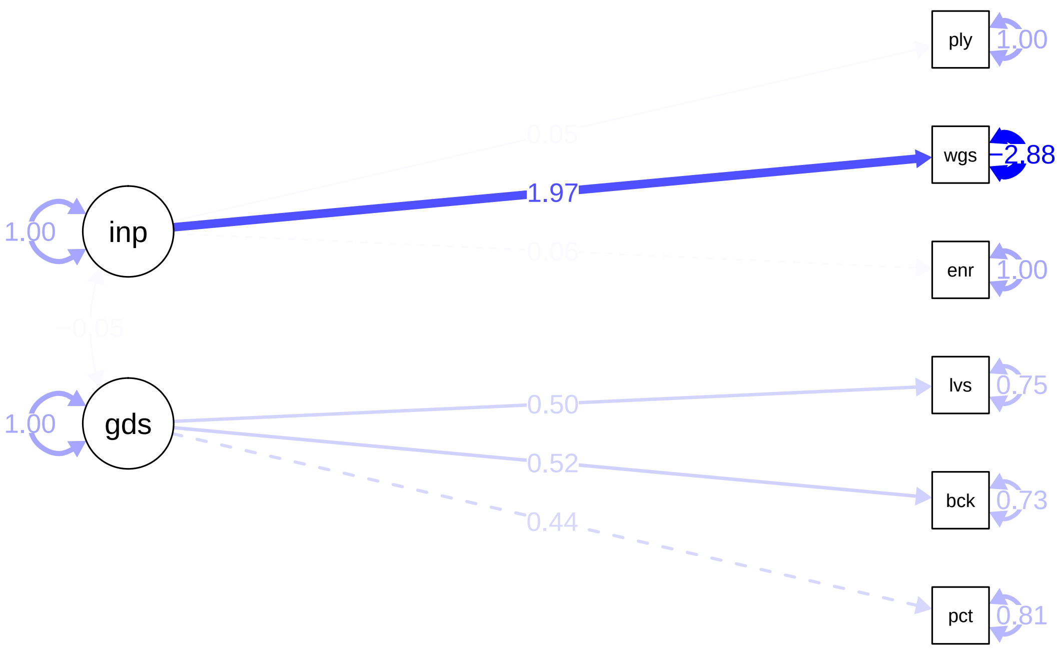
```r
#basic diagram

# Load the semPlot library
library(semPlot)

# Create a default picture
semPaths(adopt.fit)
```




```r
# Update the default picture
semPaths(object = adopt.fit,
         layout= 'tree',
         rotation = 2)
```




```r
# Update the default picture
semPaths(object = adopt.fit,
         layout = "tree",
         rotation = 2,
         whatLabels= 'std',
         edge.label.cex = 1,
         what = 'std',
         edge.color = 'blue')
```



WAIS-III IQ EXAMPLE:

The WAIS-III IQ scale has a proposed four-factor model structure with verbal comprehension, working memory, perceptual organization, and processing speed. You should analyze this structure to determine if the model fits the data and that there are no problems with the model.

```r
#load data
IQdata <- fread('https://raw.githubusercontent.com/JiaxiangBU/picbackup/master/IQdata.csv')
```

```r
head(IQdata)
```

```
##    V1 inform simil vocab compreh digspan arith piccomp block matrixreason
## 1:  1     31    23    63      27      20    18      18    50           21
## 2:  2     15    20    44      21      13    12      13    29           17
## 3:  3     13    22    40      28      14    13      13    28           16
## 4:  4     13    21    51      21      22    13      16    36           14
## 5:  5     22    21    55      28      17    10      13    22           13
## 6:  6     25    22    61      27      20    20      18    59           18
##    symbolsearch digsym lnseq
## 1:           38     57    15
## 2:           24     56    12
## 3:           25     72    13
## 4:           27     67    18
## 5:           27     60    15
## 6:           38     78    16
```

```r
head(IQdata)
```


```
##    V1 inform simil vocab compreh digspan arith piccomp block matrixreason
## 1:  1     31    23    63      27      20    18      18    50           21
## 2:  2     15    20    44      21      13    12      13    29           17
## 3:  3     13    22    40      28      14    13      13    28           16
## 4:  4     13    21    51      21      22    13      16    36           14
## 5:  5     22    21    55      28      17    10      13    22           13
## 6:  6     25    22    61      27      20    20      18    59           18
##    symbolsearch digsym lnseq
## 1:           38     57    15
## 2:           24     56    12
## 3:           25     72    13
## 4:           27     67    18
## 5:           27     60    15
## 6:           38     78    16
```


```r
# Build a four-factor model
wais.model <- 'verbalcomp =~ vocab + simil + inform + compreh
workingmemory =~ arith + digspan + lnseq
perceptorg =~ piccomp + block + matrixreason
processing =~ digsym + symbolsearch'

# Analyze the model and include the data argument
wais.fit <- cfa(wais.model, IQdata)
```

```
## Warning in lav_object_post_check(object): lavaan WARNING: covariance matrix of latent variables
##                 is not positive definite;
##                 use lavInspect(fit, "cov.lv") to investigate.
```

```r
# Summarize the model with fit.measures and standardized loadings
summary(wais.fit, standardized = TRUE, fit.measures=TRUE)
```

```
## lavaan 0.6-6 ended normally after 153 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
```

```
##   Number of free parameters                         30
##
##   Number of observations                           300
##
## Model Test User Model:
##
##   Test statistic                               233.268
##   Degrees of freedom                                48
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                              1042.916
##   Degrees of freedom                                66
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.810
##   Tucker-Lewis Index (TLI)                       0.739
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -9939.800
##   Loglikelihood unrestricted model (H1)      -9823.166
##
##   Akaike (AIC)                               19939.599
##   Bayesian (BIC)                             20050.713
##   Sample-size adjusted Bayesian (BIC)        19955.570
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.113
##   90 Percent confidence interval - lower         0.099
##   90 Percent confidence interval - upper         0.128
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.073
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   verbalcomp =~
##     vocab             1.000                               6.282    0.879
##     simil             0.296    0.031    9.470    0.000    1.859    0.581
##     inform            0.450    0.043   10.483    0.000    2.825    0.645
##     compreh           0.315    0.035    8.986    0.000    1.979    0.551
```

```
##   workingmemory =~
##     arith             1.000                               2.530    0.845
##     digspan           0.875    0.137    6.373    0.000    2.213    0.561
##     lnseq             0.225    0.106    2.130    0.033    0.570    0.142
##   perceptorg =~
##     piccomp           1.000                               1.391    0.596
##     block             3.988    0.421    9.477    0.000    5.546    0.719
##     matrixreason      0.909    0.127    7.171    0.000    1.264    0.494
##   processing =~
##     digsym            1.000                               2.809    0.239
##     symbolsearch      1.065    0.300    3.547    0.000    2.990    0.724
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   verbalcomp ~~
##     workingmemory     6.120    1.232    4.969    0.000    0.385    0.385
##     perceptorg        5.644    0.868    6.503    0.000    0.646    0.646
##     processing       10.050    3.150    3.190    0.001    0.570    0.570
##   workingmemory ~~
##     perceptorg        2.437    0.371    6.561    0.000    0.693    0.693
##     processing        2.701    0.984    2.745    0.006    0.380    0.380
##   perceptorg ~~
##     processing        4.027    1.200    3.356    0.001    1.031    1.031
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .vocab            11.573    2.656    4.357    0.000   11.573    0.227
##    .simil             6.792    0.620   10.951    0.000    6.792    0.663
##    .inform           11.201    1.084   10.330    0.000   11.201    0.584
##    .compreh           8.969    0.804   11.157    0.000    8.969    0.696
##    .arith             2.560    0.901    2.842    0.004    2.560    0.286
##    .digspan          10.653    1.102    9.666    0.000   10.653    0.685
##    .lnseq            15.750    1.294   12.173    0.000   15.750    0.980
##    .piccomp           3.505    0.323   10.851    0.000    3.505    0.644
##    .block            28.761    3.207    8.968    0.000   28.761    0.483
##    .matrixreason      4.957    0.431   11.509    0.000    4.957    0.756
##    .digsym          130.314   10.847   12.014    0.000  130.314    0.943
##    .symbolsearch      8.127    2.480    3.277    0.001    8.127    0.476
##     verbalcomp       39.459    4.757    8.294    0.000    1.000    1.000
##     workingmemory     6.399    1.122    5.703    0.000    1.000    1.000
##     perceptorg        1.934    0.371    5.211    0.000    1.000    1.000
##     processing        7.889    4.309    1.831    0.067    1.000    1.000
```

```
#there is a problem with the correlation between perceptual organization and processing speed (std. all
```

To fix a highly correlated set of latent variables, you should collapse those two variables into one latent variable. You should make a performance variable that combines the manifest variables for the perceptorg and processing latent variables.

```
# Edit the original model
wais.model <- 'verbalcomp =~ vocab + simil + inform + compreh
workingmemory =~ arith + digspan + lnseq
performance =~ piccomp + block + matrixreason + digsym + symbolsearch'

# Analyze the model and include the data argument
```

```r
wais.fit <- cfa(wais.model, IQdata)

# Summarize the model
summary(wais.fit, standardized= TRUE, fit.measure=TRUE)
```

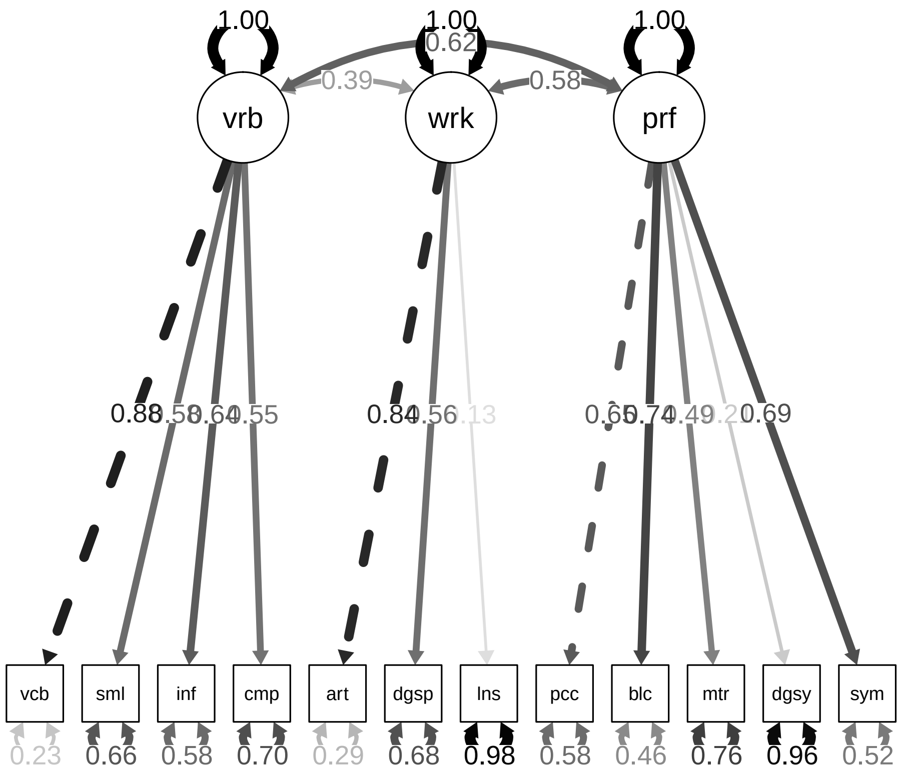
```
## lavaan 0.6-6 ended normally after 110 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         27
##
##   Number of observations                           300
##
## Model Test User Model:
##
##   Test statistic                               252.809
##   Degrees of freedom                                51
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                              1042.916
##   Degrees of freedom                                66
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.793
##   Tucker-Lewis Index (TLI)                       0.733
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -9949.570
##   Loglikelihood unrestricted model (H1)      -9823.166
##
##   Akaike (AIC)                               19953.141
##   Bayesian (BIC)                             20053.143
##   Sample-size adjusted Bayesian (BIC)        19967.515
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.115
##   90 Percent confidence interval - lower         0.101
##   90 Percent confidence interval - upper         0.129
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.076
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
```

```
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   verbalcomp =~
##     vocab             1.000                               6.281    0.879
##     simil             0.296    0.031    9.483    0.000    1.861    0.581
##     inform            0.449    0.043   10.481    0.000    2.822    0.644
##     compreh           0.315    0.035    8.999    0.000    1.981    0.552
##   workingmemory =~
##     arith             1.000                               2.528    0.844
##     digspan           0.881    0.152    5.786    0.000    2.227    0.565
##     lnseq             0.205    0.107    1.920    0.055    0.518    0.129
##   performance =~
##     piccomp           1.000                               1.517    0.650
##     block             3.739    0.390    9.583    0.000    5.672    0.735
##     matrixreason      0.832    0.117    7.099    0.000    1.262    0.493
##     digsym            1.603    0.507    3.160    0.002    2.431    0.207
##     symbolsearch      1.880    0.204    9.236    0.000    2.852    0.690
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   verbalcomp ~~
##     workingmemory     6.132    1.234    4.970    0.000    0.386    0.386
##     performance       5.892    0.886    6.647    0.000    0.618    0.618
##   workingmemory ~~
##     performance       2.227    0.362    6.149    0.000    0.581    0.581
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .vocab            11.577    2.651    4.367    0.000   11.577    0.227
##    .simil             6.787    0.620   10.950    0.000    6.787    0.662
##    .inform           11.218    1.085   10.342    0.000   11.218    0.585
##    .compreh           8.962    0.803   11.155    0.000    8.962    0.696
##    .arith             2.571    1.014    2.535    0.011    2.571    0.287
##    .digspan          10.590    1.161    9.121    0.000   10.590    0.681
##    .lnseq            15.807    1.297   12.183    0.000   15.807    0.983
##    .piccomp           3.138    0.317    9.913    0.000    3.138    0.577
##    .block            27.343    3.226    8.476    0.000   27.343    0.459
##    .matrixreason      4.960    0.441   11.243    0.000    4.960    0.757
##    .digsym          132.291   10.925   12.109    0.000  132.291    0.957
##    .symbolsearch      8.936    0.957    9.333    0.000    8.936    0.524
##     verbalcomp       39.455    4.754    8.299    0.000    1.000    1.000
##     workingmemory     6.388    1.215    5.259    0.000    1.000    1.000
##     performance       2.301    0.408    5.646    0.000    1.000    1.000
```

this solves the Heywood case(Correlations that are out of bound)

```r
# Load the library
#library(semPlot)

# Update the default picture
semPaths(object = wais.fit,
         layout = "tree",
         rotation = 1,
```

```r
         whatLabels = 'std',        #standardized loading as labels
         edge.label.cex = 1,
         what = 'std',              #shading
         edge.color = 'black')      #color of shading
```



Our three-factor model picture indicates that some of the loadings are not very strong, which indicates manifest(observable) variables that are not measuring their latent variable.

#Add Paths to Improve Fit The three-factor model of the WAIS-III showed poor fit when examining the fit indices. You can use the modification indices to view potential parameter estimates to add to the model to improve fit. Correlated error terms are normal estimates to add, as the variance of the manifest variables on the same factor can be related to each other.

```r
#View the modification indices output and add the highest mi value to update the model.

# Examine modification indices
modificationindices(wais.fit, sort = TRUE)
```

```
##                lhs op          rhs     mi     epc sepc.lv sepc.all sepc.nox
## 66           simil ~~       inform 35.879  -3.757  -3.757   -0.431   -0.431
## 56           vocab ~~       inform 28.377   9.783   9.783    0.858    0.858
## 48     performance =~        vocab 21.865  -2.077  -3.151   -0.441   -0.441
## 115          block ~~ matrixreason 16.209  -3.622  -3.622   -0.311   -0.311
## 96           arith ~~        block 15.061   3.679   3.679    0.439    0.439
## 117          block ~~ symbolsearch 13.144   5.725   5.725    0.366    0.366
## 47   workingmemory =~ symbolsearch 12.272  -0.467  -1.181   -0.286   -0.286
## 81          inform ~~        block 12.269   4.358   4.358    0.249    0.249
## 64           vocab ~~       digsym 11.578 -11.261 -11.261   -0.288   -0.288
## 40   workingmemory =~        simil 11.383   0.278   0.703    0.220    0.220
## 72           simil ~~        block 10.605  -3.084  -3.084   -0.226   -0.226
## 45   workingmemory =~ matrixreason  9.685   0.267   0.675    0.264    0.264
```

```
## 95          arith ~~      piccomp  9.463  -0.892  -0.892   -0.314   -0.314
## 60          vocab ~~        lnseq  9.425  -3.486  -3.486   -0.258   -0.258
## 67          simil ~~      compreh  9.356   1.587   1.587    0.203    0.203
## 44  workingmemory =~        block  9.258   0.765   1.933    0.251    0.251
## 51    performance =~      compreh  9.177   0.601   0.912    0.254    0.254
## 62          vocab ~~        block  8.712  -5.377  -5.377   -0.302   -0.302
## 73          simil ~~ matrixreason  8.672   1.065   1.065    0.184    0.184
## 106         lnseq ~~      piccomp  8.620   1.298   1.298    0.184    0.184
## 91        compreh ~~       digsym  8.155   5.908   5.908    0.172    0.172
## 59          vocab ~~      digspan  8.127   2.849   2.849    0.257    0.257
## 37     verbalcomp =~       digsym  7.803  -0.464  -2.917   -0.248   -0.248
## 68          simil ~~        arith  7.534   1.064   1.064    0.255    0.255
## 99          arith ~~ symbolsearch  7.468  -1.391  -1.391   -0.290   -0.290
## 57          vocab ~~      compreh  7.107  -3.508  -3.508   -0.344   -0.344
## 87        compreh ~~        lnseq  7.001   1.887   1.887    0.159    0.159
## 97          arith ~~ matrixreason  6.391   0.848   0.848    0.237    0.237
## 107         lnseq ~~        block  5.677   3.289   3.289    0.158    0.158
## 34     verbalcomp =~      piccomp  5.507   0.071   0.447    0.192    0.192
## 78         inform ~~      digspan  5.435  -1.649  -1.649   -0.151   -0.151
## 33     verbalcomp =~        lnseq  5.250  -0.104  -0.652   -0.163   -0.163
## 54    performance =~        lnseq  4.644   0.512   0.777    0.194    0.194
## 39  workingmemory =~        vocab  4.638  -0.406  -1.025   -0.143   -0.143
## 102       digspan ~~        block  4.564  -2.689  -2.689   -0.158   -0.158
## 35     verbalcomp =~        block  4.551  -0.218  -1.371   -0.178   -0.178
## 88        compreh ~~      piccomp  4.455   0.728   0.728    0.137    0.137
## 112       piccomp ~~ matrixreason  4.306   0.568   0.568    0.144    0.144
## 101       digspan ~~      piccomp  4.218   0.808   0.808    0.140    0.140
## 46  workingmemory =~       digsym  4.139  -0.852  -2.152   -0.183   -0.183
## 71          simil ~~      piccomp  4.029   0.607   0.607    0.132    0.132
## 76         inform ~~      compreh  3.789  -1.367  -1.367   -0.136   -0.136
## 70          simil ~~        lnseq  3.693  -1.200  -1.200   -0.116   -0.116
## 50    performance =~       inform  3.487   0.444   0.673    0.154    0.154
## 58          vocab ~~        arith  3.451  -1.457  -1.457   -0.267   -0.267
## 55          vocab ~~        simil  3.393   2.239   2.239    0.253    0.253
## 113       piccomp ~~       digsym  3.375   2.419   2.419    0.119    0.119
## 93          arith ~~      digspan  3.274   7.960   7.960    1.526    1.526
## 86        compreh ~~      digspan  3.234  -1.110  -1.110   -0.114   -0.114
## 80         inform ~~      piccomp  2.871  -0.672  -0.672   -0.113   -0.113
## 104       digspan ~~       digsym  2.754  -3.822  -3.822   -0.102   -0.102
## 114       piccomp ~~ symbolsearch  2.677  -0.731  -0.731   -0.138   -0.138
## 89        compreh ~~        block  2.551   1.725   1.725    0.110    0.110
## 90        compreh ~~ matrixreason  2.342  -0.632  -0.632   -0.095   -0.095
## 74          simil ~~       digsym  2.021  -2.575  -2.575   -0.086   -0.086
## 43  workingmemory =~      piccomp  1.899  -0.104  -0.262   -0.113   -0.113
## 49    performance =~        simil  1.675   0.227   0.345    0.108    0.108
## 92        compreh ~~ symbolsearch  1.646   0.764   0.764    0.085    0.085
## 111       piccomp ~~        block  1.591  -1.084  -1.084   -0.117   -0.117
## 85        compreh ~~        arith  1.350  -0.514  -0.514   -0.107   -0.107
## 32     verbalcomp =~      digspan  1.224   0.058   0.365    0.092    0.092
## 79         inform ~~        lnseq  0.998  -0.815  -0.815   -0.061   -0.061
## 69          simil ~~      digspan  0.996   0.540   0.540    0.064    0.064
## 53    performance =~      digspan  0.942  -0.710  -1.077   -0.273   -0.273
## 77         inform ~~        arith  0.890   0.480   0.480    0.089    0.089
## 116         block ~~       digsym  0.805   3.770   3.770    0.063    0.063
```

```
## 120        digsym ~~ symbolsearch  0.724  1.948  1.948   0.057   0.057
## 100       digspan ~~        lnseq  0.703 -0.688 -0.688  -0.053  -0.053
## 83         inform ~~       digsym  0.667  1.935  1.935   0.050   0.050
## 36     verbalcomp =~ matrixreason  0.543  0.025  0.159   0.062   0.062
## 61          vocab ~~      piccomp  0.529  0.414  0.414   0.069   0.069
## 105       digspan ~~ symbolsearch  0.481 -0.475 -0.475  -0.049  -0.049
## 52    performance =~        arith  0.478 -0.694 -1.052  -0.352  -0.352
## 98          arith ~~       digsym  0.474 -1.135 -1.135  -0.062  -0.062
## 94          arith ~~        lnseq  0.430 -0.496 -0.496  -0.078  -0.078
## 31     verbalcomp =~        arith  0.237 -0.029 -0.182  -0.061  -0.061
## 103       digspan ~~ matrixreason  0.226  0.221  0.221   0.030   0.030
## 42  workingmemory =~      compreh  0.190 -0.041 -0.103  -0.029  -0.029
## 75          simil ~~ symbolsearch  0.188 -0.227 -0.227  -0.029  -0.029
## 63          vocab ~~ matrixreason  0.143 -0.253 -0.253  -0.033  -0.033
## 109         lnseq ~~       digsym  0.128 -0.951 -0.951  -0.021  -0.021
## 38     verbalcomp =~ symbolsearch  0.077  0.015  0.094   0.023   0.023
## 118  matrixreason ~~       digsym  0.060 -0.380 -0.380  -0.015  -0.015
## 41  workingmemory =~       inform  0.037  0.021  0.053   0.012   0.012
## 119  matrixreason ~~ symbolsearch  0.031 -0.085 -0.085  -0.013  -0.013
## 108         lnseq ~~ matrixreason  0.017  0.069  0.069   0.008   0.008
## 110         lnseq ~~ symbolsearch  0.009  0.072  0.072   0.006   0.006
## 65          vocab ~~ symbolsearch  0.005 -0.068 -0.068  -0.007  -0.007
## 84         inform ~~ symbolsearch  0.004 -0.045 -0.045  -0.004  -0.004
## 82         inform ~~ matrixreason  0.004  0.029  0.029   0.004   0.004
```

```r
# Update the three-factor model
wais.model2 <- 'verbalcomp =~ vocab + simil + inform + compreh
workingmemory =~ arith + digspan + lnseq
perceptorg =~ piccomp + block + matrixreason + digsym + symbolsearch
simil ~~ inform'

# Analyze the three-factor model where data is IQdata
wais.fit2 <- cfa(wais.model2, IQdata)

# Summarize the three-factor model
summary(wais.fit2, standardized=TRUE, fit.measures=TRUE )
```

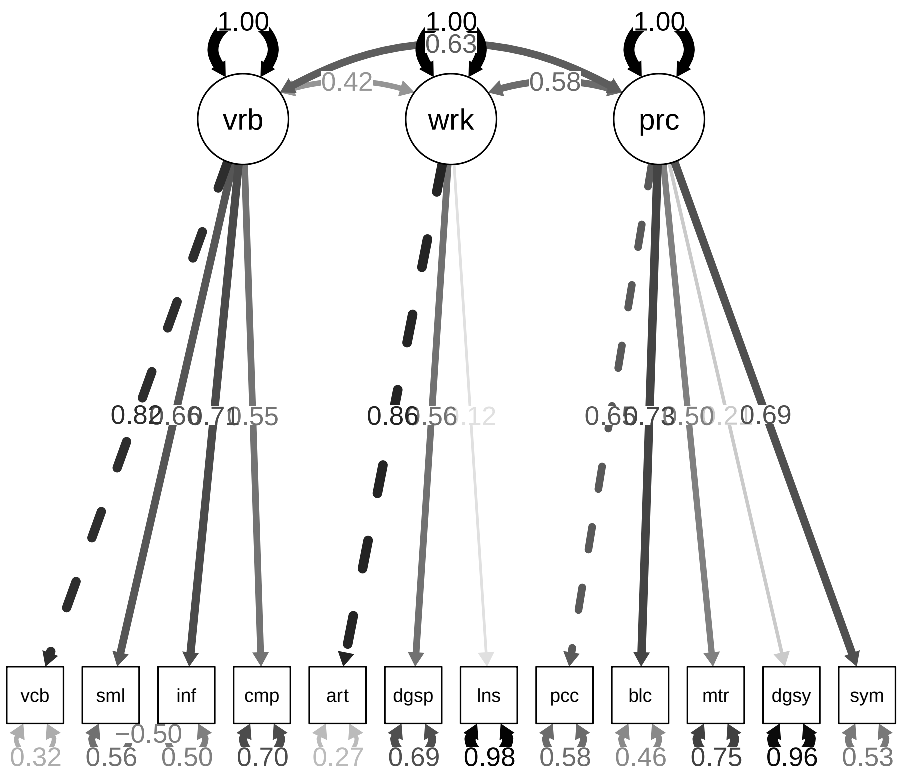
```
## lavaan 0.6-6 ended normally after 114 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         28
##
##   Number of observations                           300
##
## Model Test User Model:
##
##   Test statistic                               212.813
##   Degrees of freedom                                50
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                              1042.916
##   Degrees of freedom                                66
```

```
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.833
##   Tucker-Lewis Index (TLI)                       0.780
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -9929.572
##   Loglikelihood unrestricted model (H1)      -9823.166
##
##   Akaike (AIC)                               19915.144
##   Bayesian (BIC)                             20018.850
##   Sample-size adjusted Bayesian (BIC)        19930.051
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.104
##   90 Percent confidence interval - lower         0.090
##   90 Percent confidence interval - upper         0.119
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.071
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   verbalcomp =~
##     vocab             1.000                               5.888    0.824
##     simil             0.361    0.035   10.184    0.000    2.125    0.664
##     inform            0.525    0.048   10.857    0.000    3.090    0.706
##     compreh           0.334    0.036    9.349    0.000    1.965    0.547
##   workingmemory =~
##     arith             1.000                               2.565    0.857
##     digspan           0.857    0.149    5.768    0.000    2.199    0.558
##     lnseq             0.193    0.104    1.850    0.064    0.495    0.123
##   perceptorg =~
##     piccomp           1.000                               1.515    0.650
##     block             3.737    0.390    9.581    0.000    5.662    0.734
##     matrixreason      0.843    0.118    7.176    0.000    1.278    0.499
##     digsym            1.615    0.508    3.181    0.001    2.446    0.208
##     symbolsearch      1.875    0.203    9.218    0.000    2.841    0.688
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##  .simil ~~
```

```
##    .inform           -3.738    0.606   -6.169    0.000   -3.738   -0.503
##   verbalcomp ~~
##     workingmemory     6.278    1.181    5.315    0.000    0.416    0.416
##     perceptorg        5.654    0.859    6.583    0.000    0.634    0.634
##   workingmemory ~~
##     perceptorg        2.237    0.363    6.172    0.000    0.576    0.576
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .vocab            16.365    2.375    6.892    0.000   16.365    0.321
##    .simil             5.734    0.610    9.399    0.000    5.734    0.560
##    .inform            9.635    1.095    8.801    0.000    9.635    0.502
##    .compreh           9.026    0.791   11.413    0.000    9.026    0.700
##    .arith             2.380    1.037    2.294    0.022    2.380    0.266
##    .digspan          10.715    1.154    9.282    0.000   10.715    0.689
##    .lnseq            15.830    1.298   12.193    0.000   15.830    0.985
##    .piccomp           3.143    0.316    9.937    0.000    3.143    0.578
##    .block            27.457    3.220    8.527    0.000   27.457    0.461
##    .matrixreason      4.921    0.439   11.216    0.000    4.921    0.751
##    .digsym          132.218   10.920   12.108    0.000  132.218    0.957
##    .symbolsearch      8.996    0.958    9.393    0.000    8.996    0.527
##     verbalcomp       34.667    4.408    7.865    0.000    1.000    1.000
##     workingmemory     6.579    1.239    5.309    0.000    1.000    1.000
##     perceptorg        2.296    0.407    5.643    0.000    1.000    1.000
```

This model appears to have better fit indices than the previous model.

```r
# Update the default picture
semPaths(object = wais.fit2,
         layout = "tree",
         rotation = 1,
         whatLabels = 'std',        #standardized loading as labels
         edge.label.cex = 1,
         what = 'std',              #shading
         edge.color = 'black')      #color of shading
```



Use the anova() function and the aic and ecvi fit indices outlined previously to help determine if model fit was significantly improved.

```r
# Compare the models
anova(wais.fit, wais.fit2)
```

```
## Chi-Squared Difference Test
##
##           Df   AIC   BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## wais.fit2 50 19915 20019 212.81
## wais.fit  51 19953 20053 252.81     39.996       1  2.545e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# View the fit indices for the original model
fitmeasures(wais.fit, c('aic', 'ecvi'))
```

```
##       aic      ecvi
## 19953.141     1.023
```

```r
# View the fit indices for the updated model
fitmeasures(wais.fit2, c('aic', 'ecvi'))
```

```
##       aic      ecvi
## 19915.144     0.896
```

The three-factor model with the added correlated error fits better than the original model!

#HIERARCHICAL MODELS The underlying theory about intelligence states that a general IQ factor predicts performance on the verbal comprehension, working memory, and perceptual organization subfactors. Therefore, you should create a hierarchical model that demonstrates that relationship between the second order latent variable and the first layer of latent variables.

```r
# Update the three-factor model to a hierarchical model
wais.model3 <- 'verbalcomp =~ vocab + simil + inform + compreh
```

```
workingmemory =~ arith + digspan + lnseq
perceptorg =~ piccomp + block + matrixreason + digsym + symbolsearch
simil ~~ inform
general =~ verbalcomp + workingmemory + perceptorg'   #THISLINE

# Analyze the hierarchical model where data is IQdata
wais.fit3 <- cfa(model = wais.model3, data = IQdata)

# Examine the fit indices for the old model
fitmeasures(wais.fit2, c('rmsea', 'srmr'))
```

```
## rmsea  srmr
## 0.104 0.071
```

```
# Examine the fit indices for the new model
fitmeasures(wais.fit3, c('rmsea', 'srmr'))
```

```
## rmsea  srmr
## 0.104 0.071
```

```
# Update the default picture
semPaths(object = wais.fit3,
         layout = 'tree',
         rotation = 1,
         whatLabels = 'std',
         edge.label.cex = 1,
         what = 'std',
         edge.color = 'navy')
```