SEM: WAIS-III IQ

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
library(data.table)
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
```


Build a four-factor model

```r
library(lavaan)
```

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

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

```
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.
```

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
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
```

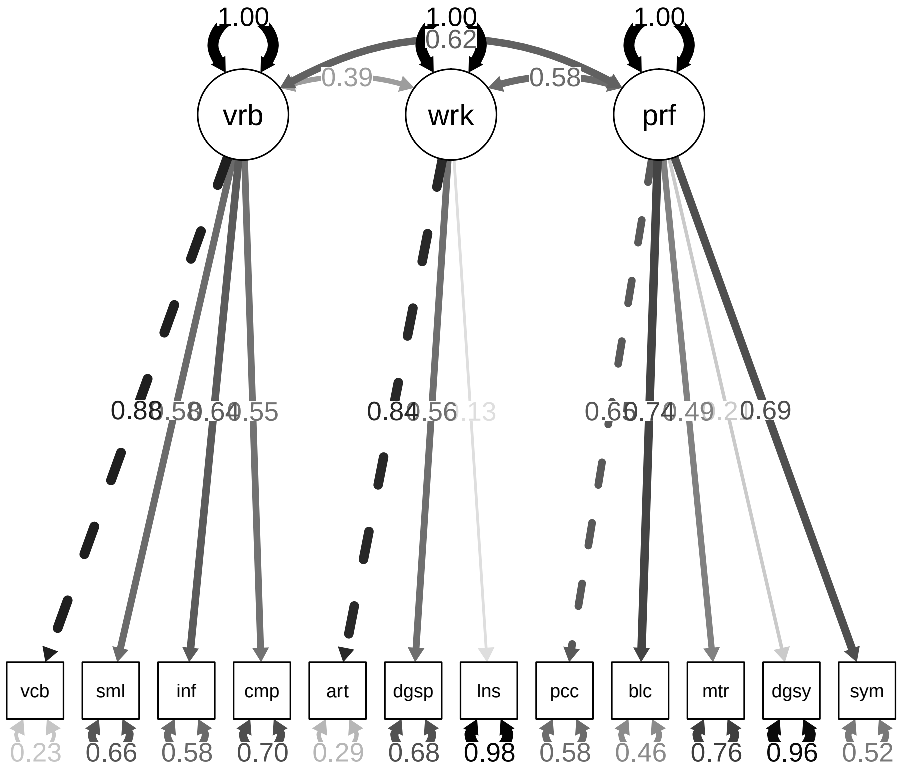
```
##    .simil             6.787    0.620   10.950    0.000    6.787    0.662
##    .inform           11.218    1.085   10.342    0.000   11.218    0.585
##    .compreh           8.962    0.803   11.155    0.000    8.962    0.696
##    .arith             2.571    1.014    2.535    0.011    2.571    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) ## SEM Diagram

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

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

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

Examine modification indices

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

Update the three-factor model

```r
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

```r
wais.fit2 <- cfa(wais.model2, IQdata)
```

Summarize the three-factor model

```r
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:
```

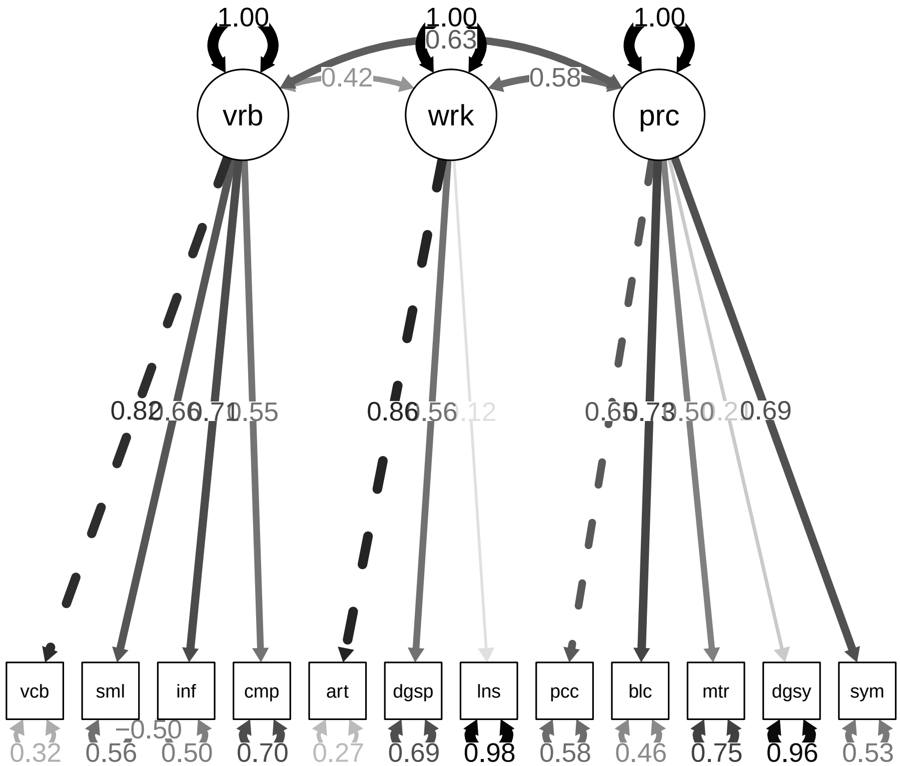
```
##
##   Test statistic                              1042.916
##   Degrees of freedom                                66
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.833
##   Tucker-Lewis Index (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.

Update the default picture

```r
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.

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
```

View the fit indices for the original and updated models

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

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

```
# 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.

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)
```

View the fit indices RMSEA and SRMR for the original and updated models

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

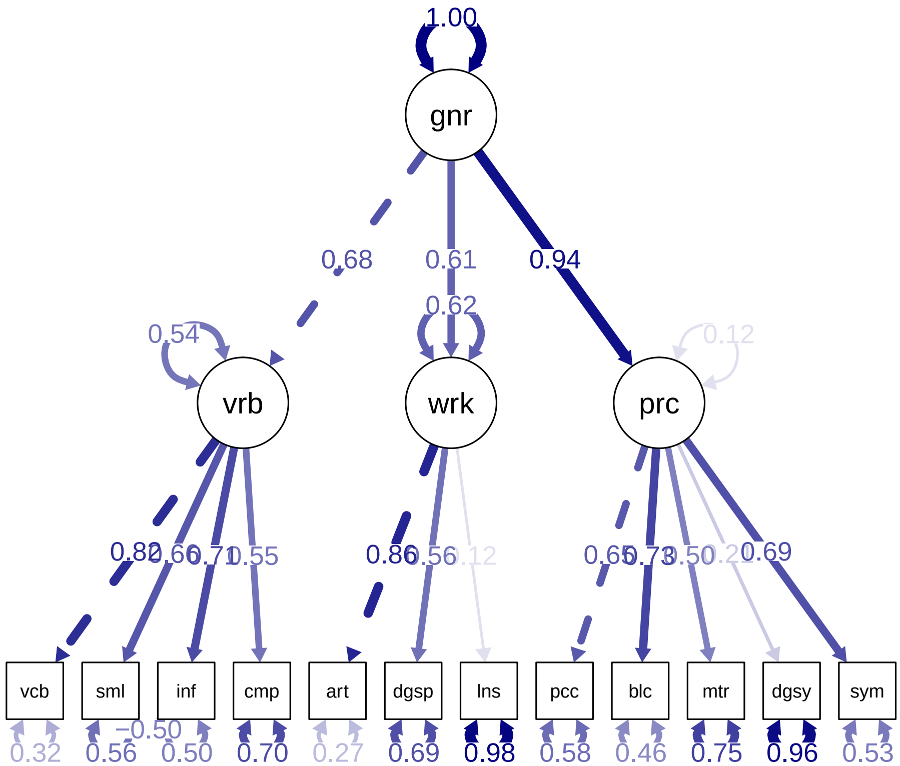
```
## rmsea  srmr
## 0.104 0.071
```

```
# 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')
```



1.00
gnr
0.68
0.61
0.94
0.62
0.54
0.12
vrb
wrk
prc
0.82
0.66
0.71
0.55
0.86
0.56
0.12
0.65
0.73
0.50
0.21
0.69
vcb
sml
inf
cmp
art
dgsp
lns
pcc
blc
mtr
dgsy
sym
−0.50
0.32
0.56
0.50
0.70
0.27
0.69
0.98
0.58
0.46
0.75
0.96
0.53