Example 3.3 and 3.4

Part I

First, we load the lavaan package and read in the data:

```
library("lavaan")
```

```
wisc4.cor <- lav_matrix_lower2full(c(1.0,
                                     .72, 1.0,
                                     .64, .63, 1.0,
                                     .51, .48, .37, 1.0,
                                     .37, .38, .38, .38, 1))
colnames(wisc4.cor) <- rownames(wisc4.cor) <- c("Information", "Similarities",
                                                "Word.Reasoning", "Matrix.Reasoning",
                                                "Picture.Concepts")
wisc4.sd <- c(3.01, 3.03, 2.99, 2.89, 2.98)
names(wisc4.sd) <- colnames(wisc4.cor)
wisc4.cov <- cor2cov(wisc4.cor, wisc4.sd)
```

Using the first indicator for identification of the scale of the latent variable is the default in lavaan. In that case, we would specify and fit the model as follows:

```
wisc4.model <- '
  g =~ Information + Similarities + Word.Reasoning + Matrix.Reasoning +
        Picture.Concepts
'
wisc4.fit <- cfa(wisc4.model, sample.cov = wisc4.cov, sample.nobs = 550)
summary(wisc4.fit, standardized = TRUE)
```

```
## lavaan (0.6-1.1188) converged normally after  30 iterations
##
##   Number of observations                           550
##
##   Estimator                                         ML
##   Model Fit Test Statistic                      26.775
##   Degrees of freedom                                 5
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##   Standard Errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   g =~
##     Information       1.000                               2.578    0.857
##     Similarities      0.985    0.045   21.708    0.000    2.541    0.839
##     Word.Reasoning    0.860    0.045   18.952    0.000    2.217    0.742
##     Matrix.Reasnng    0.647    0.047   13.896    0.000    1.669    0.578
##     Picture.Cncpts    0.542    0.050   10.937    0.000    1.398    0.470
##
## Variances:
```

```
##                      Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Information          2.395    0.250    9.587    0.000    2.395    0.265
##    .Similarities         2.709    0.258   10.482    0.000    2.709    0.296
##    .Word.Reasoning       4.009    0.295   13.600    0.000    4.009    0.449
##    .Matrix.Reasnng       5.551    0.360   15.400    0.000    5.551    0.666
##    .Picture.Cncpts       6.909    0.434   15.922    0.000    6.909    0.779
##     g                    6.648    0.564   11.788    0.000    1.000    1.000
```

Additional question

What do the β, Λ, Ψ and Θ matrices look like?

If we want to identify the scale of the latent variable by setting its variance to one, we could use one of the following two approaches:

```
wisc4.fit.std1 <- cfa(wisc4.model, sample.cov = wisc4.cov, sample.nobs = 550,
                      std.lv = TRUE)
ests <- c("lhs", "op", "rhs", "est", "pvalue", "std.lv")
parameterEstimates(wisc4.fit.std1, standardized = TRUE)[ests]
```

```
##                 lhs op             rhs   est pvalue std.lv
## 1                 g =~     Information 2.578      0  2.578
## 2                 g =~    Similarities 2.541      0  2.541
## 3                 g =~  Word.Reasoning 2.217      0  2.217
## 4                 g =~ Matrix.Reasoning 1.669      0  1.669
## 5                 g =~ Picture.Concepts 1.398      0  1.398
## 6       Information ~~     Information 2.395      0  2.395
## 7      Similarities ~~    Similarities 2.709      0  2.709
## 8    Word.Reasoning ~~  Word.Reasoning 4.009      0  4.009
## 9  Matrix.Reasoning ~~ Matrix.Reasoning 5.551      0  5.551
## 10 Picture.Concepts ~~ Picture.Concepts 6.909      0  6.909
## 11                g ~~               g 1.000     NA  1.000
```

```
wisc4.model.std2 <- '
    g =~ NA*Information + Similarities + Word.Reasoning + Matrix.Reasoning +
          Picture.Concepts
    g ~~ 1*g
'
wisc4.fit.std2 <- cfa(wisc4.model.std2, sample.cov = wisc4.cov,
                      sample.nobs = 550)
parameterEstimates(wisc4.fit.std2, standardized = TRUE)[ests]
```

```
##                 lhs op             rhs   est pvalue std.lv
## 1                 g =~     Information 2.578      0  2.578
## 2                 g =~    Similarities 2.541      0  2.541
## 3                 g =~  Word.Reasoning 2.217      0  2.217
## 4                 g =~ Matrix.Reasoning 1.669      0  1.669
## 5                 g =~ Picture.Concepts 1.398      0  1.398
## 6                 g ~~               g 1.000     NA  1.000
## 7       Information ~~     Information 2.395      0  2.395
## 8      Similarities ~~    Similarities 2.709      0  2.709
## 9    Word.Reasoning ~~  Word.Reasoning 4.009      0  4.009
## 10 Matrix.Reasoning ~~ Matrix.Reasoning 5.551      0  5.551
## 11 Picture.Concepts ~~ Picture.Concepts 6.909      0  6.909
```

Or we could use effects coding:

```
wisc4.model.eff <- '
  g =~ NA*Information + a*Information + b*Similarities + c*Word.Reasoning + d*Matrix.Reasoning +
          e*Picture.Concepts
  a + b + c + d + e == 5
'
wisc4.fit.eff <- cfa(wisc4.model.eff, sample.cov = wisc4.cov,
                      sample.nobs = 550)
parameterEstimates(wisc4.fit.eff, standardized = TRUE)[ests]
```

```
##                  lhs op               rhs   est pvalue std.lv
## 1                  g =~       Information 1.239      0  2.578
## 2                  g =~      Similarities 1.221      0  2.541
## 3                  g =~    Word.Reasoning 1.065      0  2.217
## 4                  g =~  Matrix.Reasoning 0.802      0  1.669
## 5                  g =~  Picture.Concepts 0.672      0  1.398
## 6        Information ~~       Information 2.395      0  2.395
## 7       Similarities ~~      Similarities 2.709      0  2.709
## 8     Word.Reasoning ~~    Word.Reasoning 4.009      0  4.009
## 9   Matrix.Reasoning ~~  Matrix.Reasoning 5.551      0  5.551
## 10  Picture.Concepts ~~  Picture.Concepts 6.909      0  6.909
## 11                 g ~~                 g 4.329      0  1.000
```

Let's calculate standardized loadings, commulities and standardized uniquenesses:

```
wisc4.pars <- parameterEstimates(wisc4.fit, standardized = TRUE)
wisc4.pars[wisc4.pars$op == "=~", "std.all"] # standardized loadings
```

```
## [1] 0.8573987 0.8393041 0.7421248 0.5780271 0.4696701
```

```
wisc4.pars[wisc4.pars$op == "=~", "std.all"]^2 # communalities
```

```
## [1] 0.7351326 0.7044313 0.5507492 0.3341153 0.2205900
```

```
1 - wisc4.pars[wisc4.pars$op == "=~", "std.all"]^2 # standardized uniquenesses
```

```
## [1] 0.2648674 0.2955687 0.4492508 0.6658847 0.7794100
```

Additional question

Which subtests have low communalities and high uniquenesses (variance in the subtest scores that is not explained by the common factor)?

Can you already think of a way to improve the model to better explain these subtest scores?

Let's also look at the difference between the model-implied and sample covariances:

```
fitted(wisc4.fit)$cov # model-implied covariances
```

```
##                  Infrmt Smlrts Wrd.Rs Mtrx.R Pctr.C
## Information      9.044
## Similarities     6.551  9.164
## Word.Reasoning   5.716  5.633  8.924
## Matrix.Reasoning 4.303  4.241  3.700  8.337
## Picture.Concepts 3.606  3.553  3.100  2.334  8.864
```

```
residuals(wisc4.fit)$cov # unstandardized residuals
```

```
##                  Infrmt Smlrts Wrd.Rs Mtrx.R Pctr.C
## Information       0.000
```

```
## Similarities      0.003  0.000
## Word.Reasoning    0.033  0.064  0.000
## Matrix.Reasoning  0.125 -0.045 -0.509  0.000
## Picture.Concepts -0.293 -0.128  0.280  0.933  0.000
```

```r
residuals(wisc4.fit, type="cor")$cor # standardized residuals
```

```
##                  Infrmt Smlrts Wrd.Rs Mtrx.R Pctr.C
## Information       0.000
## Similarities      0.000  0.000
## Word.Reasoning    0.004  0.007  0.000
## Matrix.Reasoning  0.014 -0.005 -0.059  0.000
## Picture.Concepts -0.033 -0.014  0.031  0.109  0.000
```

```r
fitind <- c("chisq", "df", "pvalue", "cfi", "nnfi", "rmsea", "srmr")
```

As a rule of thumb, standardized residuals > .1 are substantial. Can you think of ways how we could reduce those residual correlations > .1?

Example 3.3 and 3.4

Part II

Let's inspect the model fit and modification indices for the unidimensional model:

```
modindices(wisc4.fit)
```

```
##                  lhs op               rhs     mi    epc sepc.lv sepc.all
## 12       Information ~~      Similarities  0.010  0.034   0.034    0.013
## 13       Information ~~    Word.Reasoning  0.279  0.147   0.147    0.047
## 14       Information ~~  Matrix.Reasoning  1.447  0.280   0.280    0.077
## 15       Information ~~  Picture.Concepts  5.493 -0.565  -0.565   -0.139
## 16      Similarities ~~    Word.Reasoning  0.791  0.242   0.242    0.073
## 17      Similarities ~~  Matrix.Reasoning  0.147 -0.089  -0.089   -0.023
## 18      Similarities ~~  Picture.Concepts  0.838 -0.223  -0.223   -0.051
## 19    Word.Reasoning ~~  Matrix.Reasoning  8.931 -0.710  -0.710   -0.151
## 20    Word.Reasoning ~~  Picture.Concepts  2.029  0.365   0.365    0.069
## 21  Matrix.Reasoning ~~  Picture.Concepts 14.157  1.058   1.058    0.171
##    sepc.nox
## 12    0.013
## 13    0.047
## 14    0.077
## 15   -0.139
## 16    0.073
## 17   -0.023
## 18   -0.051
## 19   -0.151
## 20    0.069
## 21    0.171
```

```
fitmeasures(wisc4.fit)
```

```
##                npar                fmin               chisq
##              10.000               0.024              26.775
##                  df              pvalue      baseline.chisq
##               5.000               0.000            1073.427
##         baseline.df     baseline.pvalue                 cfi
##              10.000               0.000               0.980
##                 tli                nnfi                 rfi
##               0.959               0.959               0.950
##                 nfi                pnfi                 ifi
##               0.975               0.488               0.980
##                 rni                logl   unrestricted.logl
##               0.980           -6378.678           -6365.291
##                 aic                 bic              ntotal
##           12777.357           12820.456             550.000
##                bic2               rmsea      rmsea.ci.lower
##           12788.712               0.089               0.058
##      rmsea.ci.upper        rmsea.pvalue                 rmr
##               0.123               0.022               0.298
##          rmr_nomean                srmr        srmr_bentler
##               0.298               0.034               0.034
## srmr_bentler_nomean         srmr_bollen  srmr_bollen_nomean
##               0.034               0.034               0.034
```

```
##          srmr_mplus   srmr_mplus_nomean              cn_05
##               0.034               0.034            228.408
##               cn_01                 gfi               agfi
##             310.899               0.982              0.947
##                pgfi                 mfi               ecvi
##               0.327               0.980              0.085
```

Does the model fit well according to the chi-square? Is that to be expected with this sample size? Does the model fit well according to CFI? SRMR? RMSEA?

Do the modification indices suggest the same changes to the model as the residual covariances did earlier?

In the book, Beaujean decided to improve the model by including Verbal and Fluid intelligence factors:

```r
wisc4.model2 <- '
  V =~ Information + Similarities + Word.Reasoning
  F =~ Matrix.Reasoning + Picture.Concepts
'
wisc4.fit.2 <- cfa(wisc4.model2, sample.cov = wisc4.cov,
                   sample.nobs = 550)
```

Additional question

As a rule-of-thumb, an LV model needs at least 3 indicators to be identified. How come the above LV model is identified?

Also, why was the covariance between the two latent factors included in the model, although we did not specify it in the syntax?

We could also include a structural model, where we assume a causal relationship between the two types of intelligence:

```r
wisc4.model3 <- '
  V =~ Information + Similarities + Word.Reasoning
  F =~ Matrix.Reasoning + Picture.Concepts
  V ~ F
'
wisc4.fit.3 <- cfa(wisc4.model3, sample.cov = wisc4.cov, sample.nobs = 550)
```

Or, we could have done exactly what the modification indices suggested:

```r
wisc4.model4 <- '
  g =~ Information + Similarities + Word.Reasoning + Matrix.Reasoning +
        Picture.Concepts
  Matrix.Reasoning ~~ Picture.Concepts
'
wisc4.fit.4 <- cfa(wisc4.model4, sample.cov = wisc4.cov,
                   sample.nobs = 550)
```

Which of the models fits best?

```r
fitMeasures(wisc4.fit, fitind)
```

```
##  chisq     df pvalue    cfi   nnfi  rmsea   srmr
## 26.775  5.000  0.000  0.980  0.959  0.089  0.034
```

```r
fitMeasures(wisc4.fit.2, fitind)
```

```
##  chisq     df pvalue    cfi   nnfi  rmsea   srmr
## 12.687  4.000  0.013  0.992  0.980  0.063  0.019
```

```r
fitMeasures(wisc4.fit.3, fitind)
```

```
##  chisq     df pvalue    cfi   nnfi  rmsea   srmr 
## 12.687  4.000  0.013  0.992  0.980  0.063  0.019
```

```r
fitMeasures(wisc4.fit.4, fitind)
```

```
##  chisq     df pvalue    cfi   nnfi  rmsea   srmr 
## 12.687  4.000  0.013  0.992  0.980  0.063  0.019
```

```r
residuals(wisc4.fit.2, type = "cor")$cor
```

```
##                   Infrmt Smlrts Wrd.Rs Mtrx.R Pctr.C
## Information        0.000                            
## Similarities      -0.003  0.000                     
## Word.Reasoning     0.001  0.005  0.000              
## Matrix.Reasoning   0.023  0.003 -0.051  0.000       
## Picture.Concepts  -0.020 -0.001  0.043  0.000  0.000
```

```r
residuals(wisc4.fit.3, type = "cor")$cor
```

```
##                   Infrmt Smlrts Wrd.Rs Mtrx.R Pctr.C
## Information        0.000                            
## Similarities      -0.003  0.000                     
## Word.Reasoning     0.001  0.005  0.000              
## Matrix.Reasoning   0.023  0.003 -0.051  0.000       
## Picture.Concepts  -0.020 -0.001  0.043  0.000  0.000
```

```r
residuals(wisc4.fit.4, type = "cor")$cor
```

```
##                   Infrmt Smlrts Wrd.Rs Mtrx.R Pctr.C
## Information        0.000                            
## Similarities      -0.003  0.000                     
## Word.Reasoning     0.001  0.005  0.000              
## Matrix.Reasoning   0.023  0.003 -0.051  0.000       
## Picture.Concepts  -0.020 -0.001  0.043  0.000  0.000
```

Note that the last three models are equivalent: They have exactly the same number of estimated parameters, model fit and residuals. In other words, the data cannot discriminate between the three models. Only the researcher can, by using theory and interpreting the model.

Additional question

What do the Λ, β, Ψ and Θ matrices look like in the last three fitted models?