Example 6.2: Dichotomous indicator variables

First, let's import the data and look at the tetrachoric correlations:

```
library(psych)
head(lsat6)
```

```
##      Q1 Q2 Q3 Q4 Q5
## [1,]  0  0  0  0  0
## [2,]  0  0  0  0  0
## [3,]  0  0  0  0  0
## [4,]  0  0  0  0  1
## [5,]  0  0  0  0  1
## [6,]  0  0  0  0  1
```

```
tetrachoric(lsat6)
```

```
## Call: tetrachoric(x = lsat6)
## tetrachoric correlation
##    Q1   Q2   Q3   Q4   Q5
## Q1 1.00
## Q2 0.17 1.00
## Q3 0.23 0.19 1.00
## Q4 0.11 0.11 0.19 1.00
## Q5 0.07 0.17 0.11 0.20 1.00
##
##  with tau of
##    Q1    Q2    Q3    Q4    Q5
## -1.43 -0.55 -0.13 -0.72 -1.13
```

```
cor(lsat6)
```

```
##            Q1         Q2         Q3         Q4         Q5
## Q1 1.00000000 0.07380676 0.09888232 0.04426365 0.02378821
## Q2 0.07380676 1.00000000 0.11478875 0.06229710 0.08621540
## Q3 0.09888232 0.11478875 1.00000000 0.10907504 0.05316847
## Q4 0.04426365 0.06229710 0.10907504 1.00000000 0.09922352
## Q5 0.02378821 0.08621540 0.05316847 0.09922352 1.00000000
```

Beaujean writes that treating ordered categorical variables like continuous ones gives overestimated (i.e., spuriously high) covariances. I disagree. In my experience, correlations are lower when we treat categorical variables as continuous ones. This is what we see in the example, too: tetrachoric correlations are higher than the Pearson correlations calculated with cor().

```
apply(lsat6, 2, mean)
```

```
##    Q1    Q2    Q3    Q4    Q5
## 0.924 0.709 0.553 0.763 0.870
```

Probably, most difficuly item is Q3, easiest item is Q1.

Let's perform an IRT-style analysis using lavaan:

```
library(lavaan)
```

```
## Warning: package 'lavaan' was built under R version 3.4.4
```

```
## This is lavaan 0.6-1
```

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

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

```
model.IRT <- '
  Theta =~ l1*Q1 + l2*Q2 + l3*Q3 + l4*Q4 + l5*Q5
  # label thresholds
  Q1 | th1*t1
  Q2 | th2*t1
  Q3 | th3*t1
  Q4 | th4*t1
  Q5 | th5*t1
  # calculate difficulty parameters:
  b1 := th1/l1
  b2 := th2/l2
  b3 := th3/l3
  b4 := th4/l4
  b5 := th5/l5
  # get logistic from normal estimates:
  a1 := l1*1.7
  a2 := l2*1.7
  a3 := l3*1.7
  a4 := l4*1.7
  a5 := l5*1.7
'
fit.IRT <- cfa(model.IRT, data = data.frame(lsat6), parameterization = "theta", std.lv = TRUE,
           ordered = c("Q1","Q2","Q3","Q4","Q5"))
summary(fit.IRT, standardized = TRUE)
```

```
## lavaan (0.6-1) converged normally after  31 iterations
##
##   Number of observations                          1000
##
##   Estimator                                       DWLS      Robust
##   Model Fit Test Statistic                       4.051       4.740
##   Degrees of freedom                                 5           5
##   P-value (Chi-square)                           0.542       0.448
##   Scaling correction factor                                  0.867
##   Shift parameter                                            0.070
##     for simple second-order correction (Mplus variant)
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model        Unstructured
##   Standard Errors                           Robust.sem
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Theta =~
##     Q1        (l1)    0.423    0.143    2.957    0.003    0.423    0.389
##     Q2        (l2)    0.433    0.107    4.044    0.000    0.433    0.397
```

```
##     Q3        (l3)    0.534    0.128    4.159    0.000    0.534    0.471
##     Q4        (l4)    0.407    0.105    3.892    0.000    0.407    0.377
##     Q5        (l5)    0.364    0.112    3.258    0.001    0.364    0.342
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Q1                0.000                               0.000    0.000
##    .Q2                0.000                               0.000    0.000
##    .Q3                0.000                               0.000    0.000
##    .Q4                0.000                               0.000    0.000
##    .Q5                0.000                               0.000    0.000
##     Theta             0.000                               0.000    0.000
##
## Thresholds:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     Q1|t1    (th1)   -1.555    0.100  -15.586    0.000   -1.555   -1.433
##     Q2|t1    (th2)   -0.600    0.051  -11.809    0.000   -0.600   -0.550
##     Q3|t1    (th3)   -0.151    0.046   -3.297    0.001   -0.151   -0.133
##     Q4|t1    (th4)   -0.773    0.054  -14.232    0.000   -0.773   -0.716
##     Q5|t1    (th5)   -1.199    0.067  -17.798    0.000   -1.199   -1.126
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Q1                1.000                               1.000    0.848
##    .Q2                1.000                               1.000    0.842
##    .Q3                1.000                               1.000    0.778
##    .Q4                1.000                               1.000    0.858
##    .Q5                1.000                               1.000    0.883
##     Theta             1.000                               1.000    1.000
##
## Scales y*:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     Q1                0.921                               0.921    1.000
##     Q2                0.918                               0.918    1.000
##     Q3                0.882                               0.882    1.000
##     Q4                0.926                               0.926    1.000
##     Q5                0.940                               0.940    1.000
##
## Defined Parameters:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     b1               -3.678    1.073   -3.429    0.001   -3.678   -3.678
##     b2               -1.386    0.310   -4.474    0.000   -1.386   -1.386
##     b3               -0.283    0.100   -2.840    0.005   -0.283   -0.283
##     b4               -1.900    0.437   -4.348    0.000   -1.900   -1.900
##     b5               -3.290    0.909   -3.617    0.000   -3.290   -3.290
##     a1                0.719    0.243    2.957    0.003    0.719    0.662
##     a2                0.736    0.182    4.044    0.000    0.736    0.675
##     a3                0.908    0.218    4.159    0.000    0.908    0.801
##     a4                0.692    0.178    3.892    0.000    0.692    0.641
##     a5                0.619    0.190    3.258    0.001    0.619    0.582
```

We see the most difficult item is Q3, easiest item is Q1. Also, Q3 has the most discriminatory power and Q5 the least.

Let's perform categorical data CFA using lavaan:

```r
model.FA <- '
  Theta =~ l1*Q1 + l2*Q2 + l3*Q3 + l4*Q4 + l5*Q5
'
fit.FA <- cfa(model.FA, data = data.frame(lsat6), std.lv = TRUE,
              ordered = c("Q1","Q2","Q3","Q4","Q5"))
summary(fit.FA, standardized = TRUE)
```

```
## lavaan (0.6-1) converged normally after  24 iterations
##
##   Number of observations                          1000
##
##   Estimator                                       DWLS      Robust
##   Model Fit Test Statistic                       4.051       4.740
##   Degrees of freedom                                 5           5
##   P-value (Chi-square)                           0.542       0.448
##   Scaling correction factor                                  0.867
##   Shift parameter                                            0.070
##     for simple second-order correction (Mplus variant)
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model        Unstructured
##   Standard Errors                           Robust.sem
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Theta =~
##     Q1         (l1)    0.389    0.112    3.486    0.000    0.389    0.389
##     Q2         (l2)    0.397    0.083    4.801    0.000    0.397    0.397
##     Q3         (l3)    0.471    0.088    5.347    0.000    0.471    0.471
##     Q4         (l4)    0.377    0.083    4.536    0.000    0.377    0.377
##     Q5         (l5)    0.342    0.093    3.690    0.000    0.342    0.342
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Q1                0.000                               0.000    0.000
##    .Q2                0.000                               0.000    0.000
##    .Q3                0.000                               0.000    0.000
##    .Q4                0.000                               0.000    0.000
##    .Q5                0.000                               0.000    0.000
##     Theta             0.000                               0.000    0.000
##
## Thresholds:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     Q1|t1            -1.433    0.059  -24.431    0.000   -1.433   -1.433
##     Q2|t1            -0.550    0.042  -13.133    0.000   -0.550   -0.550
##     Q3|t1            -0.133    0.040   -3.349    0.001   -0.133   -0.133
##     Q4|t1            -0.716    0.044  -16.430    0.000   -0.716   -0.716
##     Q5|t1            -1.126    0.050  -22.395    0.000   -1.126   -1.126
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Q1                0.848                               0.848    0.848
```

```
##    .Q2                 0.842                               0.842    0.842
##    .Q3                 0.778                               0.778    0.778
##    .Q4                 0.858                               0.858    0.858
##    .Q5                 0.883                               0.883    0.883
##     Theta              1.000                               1.000    1.000
##
## Scales y*:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     Q1                1.000                               1.000    1.000
##     Q2                1.000                               1.000    1.000
##     Q3                1.000                               1.000    1.000
##     Q4                1.000                               1.000    1.000
##     Q5                1.000                               1.000    1.000
```

Note that model fit is exactly the same, and conclusions about parameter estimates also: Again, most difficult item is Q3, easiest item is Q1. Also, Q3 has the most discriminatory power and Q5 least.

Additional: Fit and compare Rasch and 2PL models

In the Rasch model, the probability of a correct answer is a function of the subject's ability and the item's difficulty:

$$p(Y=1|\theta_j,\beta_i) = \frac{e^{\theta_j-\beta_i}}{1+e^{\theta_j-\beta_i}}$$

where θ_j is the ability of person j, and β_j is the difficulty of item i.

In the 2pl model, the probability of a correct answer is a function of the subject's ability, the item's difficulty, and the item's discriminatory power:

$$p(Y=1|\theta_j,\beta_i,\alpha_i) = \frac{e^{\alpha_i(\theta_j-\beta_i)}}{1+e^{\alpha_i(\theta_j-\beta_i)}}$$

where α_i is the discrimination index of item i.

We can empirically decide between the Rasch and 2pl model, by fitting both models to the data, and testing the difference in model fit.

Let's use lavaan to fit the Rasch and 2pl model:

```
model.2pl <- '
  Theta =~ l1*Q1 + l2*Q2 + l3*Q3 + l4*Q4 + l5*Q5
'
fit.2pl <- cfa(model.2pl, data = data.frame(lsat6), std.lv = TRUE,
               ordered = c("Q1","Q2","Q3","Q4","Q5"))
parameterEstimates(fit.2pl)
```

```
##      lhs  op  rhs label    est    se       z pvalue ci.lower ci.upper
## 1  Theta  =~   Q1    l1  0.389 0.112   3.486  0.000    0.170    0.608
## 2  Theta  =~   Q2    l2  0.397 0.083   4.801  0.000    0.235    0.559
## 3  Theta  =~   Q3    l3  0.471 0.088   5.347  0.000    0.299    0.644
## 4  Theta  =~   Q4    l4  0.377 0.083   4.536  0.000    0.214    0.540
## 5  Theta  =~   Q5    l5  0.342 0.093   3.690  0.000    0.161    0.524
## 6     Q1   |   t1       -1.433 0.059 -24.431  0.000   -1.547   -1.318
## 7     Q2   |   t1       -0.550 0.042 -13.133  0.000   -0.633   -0.468
## 8     Q3   |   t1       -0.133 0.040  -3.349  0.001   -0.211   -0.055
```

```
## 9      Q4    |    t1        -0.716 0.044 -16.430  0.000   -0.801    -0.631
## 10     Q5    |    t1        -1.126 0.050 -22.395  0.000   -1.225    -1.028
## 11     Q1   ~~    Q1         0.848 0.000      NA     NA    0.848     0.848
## 12     Q2   ~~    Q2         0.842 0.000      NA     NA    0.842     0.842
## 13     Q3   ~~    Q3         0.778 0.000      NA     NA    0.778     0.778
## 14     Q4   ~~    Q4         0.858 0.000      NA     NA    0.858     0.858
## 15     Q5   ~~    Q5         0.883 0.000      NA     NA    0.883     0.883
## 16 Theta   ~~ Theta         1.000 0.000      NA     NA    1.000     1.000
## 17     Q1 ~*~    Q1         1.000 0.000      NA     NA    1.000     1.000
## 18     Q2 ~*~    Q2         1.000 0.000      NA     NA    1.000     1.000
## 19     Q3 ~*~    Q3         1.000 0.000      NA     NA    1.000     1.000
## 20     Q4 ~*~    Q4         1.000 0.000      NA     NA    1.000     1.000
## 21     Q5 ~*~    Q5         1.000 0.000      NA     NA    1.000     1.000
## 22     Q1   ~1              0.000 0.000      NA     NA    0.000     0.000
## 23     Q2   ~1              0.000 0.000      NA     NA    0.000     0.000
## 24     Q3   ~1              0.000 0.000      NA     NA    0.000     0.000
## 25     Q4   ~1              0.000 0.000      NA     NA    0.000     0.000
## 26     Q5   ~1              0.000 0.000      NA     NA    0.000     0.000
## 27 Theta   ~1              0.000 0.000      NA     NA    0.000     0.000
```

```
model.rasch <- '
  Theta =~ l*Q1 + l*Q2 + l*Q3 + l*Q4 + l*Q5
'
fit.rasch <- cfa(model.rasch, data = data.frame(lsat6), std.lv = TRUE,
              ordered = c("Q1","Q2","Q3","Q4","Q5"))
parameterEstimates(fit.rasch)
```

```
##      lhs  op   rhs label   est    se       z pvalue ci.lower ci.upper
## 1  Theta  =~    Q1     l  0.400 0.032  12.682  0.000    0.338    0.461
## 2  Theta  =~    Q2     l  0.400 0.032  12.682  0.000    0.338    0.461
## 3  Theta  =~    Q3     l  0.400 0.032  12.682  0.000    0.338    0.461
## 4  Theta  =~    Q4     l  0.400 0.032  12.682  0.000    0.338    0.461
## 5  Theta  =~    Q5     l  0.400 0.032  12.682  0.000    0.338    0.461
## 6     Q1   |    t1       -1.433 0.059 -24.431  0.000   -1.547   -1.318
## 7     Q2   |    t1       -0.550 0.042 -13.133  0.000   -0.633   -0.468
## 8     Q3   |    t1       -0.133 0.040  -3.349  0.001   -0.211   -0.055
## 9     Q4   |    t1       -0.716 0.044 -16.430  0.000   -0.801   -0.631
## 10    Q5   |    t1       -1.126 0.050 -22.395  0.000   -1.225   -1.028
## 11    Q1  ~~    Q1        0.840 0.000      NA     NA    0.840    0.840
## 12    Q2  ~~    Q2        0.840 0.000      NA     NA    0.840    0.840
## 13    Q3  ~~    Q3        0.840 0.000      NA     NA    0.840    0.840
## 14    Q4  ~~    Q4        0.840 0.000      NA     NA    0.840    0.840
## 15    Q5  ~~    Q5        0.840 0.000      NA     NA    0.840    0.840
## 16 Theta  ~~ Theta        1.000 0.000      NA     NA    1.000    1.000
## 17    Q1 ~*~    Q1        1.000 0.000      NA     NA    1.000    1.000
## 18    Q2 ~*~    Q2        1.000 0.000      NA     NA    1.000    1.000
## 19    Q3 ~*~    Q3        1.000 0.000      NA     NA    1.000    1.000
## 20    Q4 ~*~    Q4        1.000 0.000      NA     NA    1.000    1.000
## 21    Q5 ~*~    Q5        1.000 0.000      NA     NA    1.000    1.000
## 22    Q1  ~1              0.000 0.000      NA     NA    0.000    0.000
## 23    Q2  ~1              0.000 0.000      NA     NA    0.000    0.000
## 24    Q3  ~1              0.000 0.000      NA     NA    0.000    0.000
## 25    Q4  ~1              0.000 0.000      NA     NA    0.000    0.000
## 26    Q5  ~1              0.000 0.000      NA     NA    0.000    0.000
## 27 Theta  ~1              0.000 0.000      NA     NA    0.000    0.000
```

```r
anova(fit.rasch, fit.2pl)
```

```
## Scaled Chi Square Difference Test (method = "satorra.2000")
##
##           Df AIC BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## fit.2pl    5         4.0511
## fit.rasch  9         4.9433     0.8764       4     0.9279
```

```r
fitinds <- c("cfi.scaled", "rmsea.scaled", "srmr")
fitMeasures(fit.rasch, fitinds)
```

```
##   cfi.scaled rmsea.scaled         srmr
##        1.000        0.000        0.041
```

```r
fitMeasures(fit.2pl, fitinds)
```

```
##   cfi.scaled rmsea.scaled         srmr
##        1.000        0.000        0.036
```

CFI and RMSEA indicate perfect model fit for each model. According to the SRMR, the 2pl model fits slightly better, but SRMR does not take parsimonity in to account. The chi-square difference test indicates that the Rasch model does not fit significantly worse than the 2pl model. As the Rasch model has less estimated parameters, it should be preferred.

Analysis of ordered categorical items with > 2 categories

For ordered items with > 2 ordered response categories, the code is the same. Just make sure you declare the items as ordered in applying the cfa() function. Automatically, a threshold for every category - 1 is estimated. Reverde coding is not even necessary (items that should be reverse coded just get a negative loading, but you have to make sure that all categories within an item are ordered in the same direction).