Answers to exercises ordered categorical indicator variables

Exercise 6.2

```
gssdat <- read.table("gss2000.dat", header = TRUE)
colMeans(gssdat)
```

```
##    word.a    word.b    word.c    word.d    word.e    word.f    word.g    word.h 
## 0.8555895 0.9296636 0.2694529 0.9503908 0.8345226 0.8487937 0.3880394 0.3948352 
##    word.i    word.j 
## 0.7689433 0.3136256
```

Item `word.c` seems to be most difficult, `word.d` seems most easy.

a)

```
library("lavaan")
gssmod.2PL <- '
  vocab =~ word.a + word.b + word.c + word.d + word.e + word.f + word.g +
            word.h + word.i + word.j
'
gssfit.2PL <- cfa(gssmod.2PL, ordered = paste0("word.", letters[1:10]),
                  data = gssdat)
summary(gssfit.2PL, standardized = TRUE, fit.measures = TRUE)
```

```
## lavaan 0.6-5 ended normally after 25 iterations
## 
##   Estimator                                       DWLS
##   Optimization method                           NLMINB
##   Number of free parameters                         20
##                                                       
##   Number of observations                          2943
##                                                       
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                               285.075     361.903
##   Degrees of freedom                                35          35
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  0.792
##   Shift parameter                                            2.060
##     for the simple second-order correction
## 
## Model Test Baseline Model:
## 
##   Test statistic                              5463.921    4393.714
##   Degrees of freedom                                45          45
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.246
## 
```

```
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.954       0.925
##   Tucker-Lewis Index (TLI)                       0.941       0.903
##
##   Robust Comparative Fit Index (CFI)                            NA
##   Robust Tucker-Lewis Index (TLI)                               NA
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.049       0.056
##   90 Percent confidence interval - lower         0.044       0.051
##   90 Percent confidence interval - upper         0.055       0.062
##   P-value RMSEA <= 0.05                          0.577       0.022
##
##   Robust RMSEA                                                  NA
##   90 Percent confidence interval - lower                        NA
##   90 Percent confidence interval - upper                        NA
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.088       0.088
##
## 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
##   vocab =~
##     word.a            1.000                               0.466    0.466
##     word.b            1.532    0.107   14.286    0.000    0.715    0.715
##     word.c            1.043    0.088   11.871    0.000    0.487    0.487
##     word.d            1.752    0.125   13.998    0.000    0.817    0.817
##     word.e            1.442    0.098   14.661    0.000    0.672    0.672
##     word.f            1.537    0.104   14.767    0.000    0.717    0.717
##     word.g            1.146    0.092   12.419    0.000    0.534    0.534
##     word.h            1.250    0.094   13.341    0.000    0.583    0.583
##     word.i            1.063    0.079   13.399    0.000    0.496    0.496
##     word.j            1.441    0.110   13.160    0.000    0.672    0.672
##
## Intercepts:
##                     Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .word.a            0.000                               0.000    0.000
##    .word.b            0.000                               0.000    0.000
##    .word.c            0.000                               0.000    0.000
##    .word.d            0.000                               0.000    0.000
##    .word.e            0.000                               0.000    0.000
##    .word.f            0.000                               0.000    0.000
##    .word.g            0.000                               0.000    0.000
##    .word.h            0.000                               0.000    0.000
##    .word.i            0.000                               0.000    0.000
##    .word.j            0.000                               0.000    0.000
##     vocab             0.000                               0.000    0.000
##
## Thresholds:
##                       Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     word.a|t1        -1.061    0.029  -37.203    0.000   -1.061   -1.061
##     word.b|t1        -1.473    0.035  -42.114    0.000   -1.473   -1.473
##     word.c|t1         0.614    0.025   24.813    0.000    0.614    0.614
##     word.d|t1        -1.649    0.039  -42.209    0.000   -1.649   -1.649
##     word.e|t1        -0.972    0.028  -35.290    0.000   -0.972   -0.972
##     word.f|t1        -1.031    0.028  -36.600    0.000   -1.031   -1.031
##     word.g|t1         0.284    0.023   12.130    0.000    0.284    0.284
##     word.h|t1         0.267    0.023   11.395    0.000    0.267    0.267
##     word.i|t1        -0.735    0.026  -28.807    0.000   -0.735   -0.735
##     word.j|t1         0.486    0.024   20.129    0.000    0.486    0.486
##
## Variances:
##                       Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .word.a            0.782                               0.782    0.782
##    .word.b            0.489                               0.489    0.489
##    .word.c            0.763                               0.763    0.763
##    .word.d            0.332                               0.332    0.332
##    .word.e            0.548                               0.548    0.548
##    .word.f            0.486                               0.486    0.486
##    .word.g            0.714                               0.714    0.714
##    .word.h            0.660                               0.660    0.660
##    .word.i            0.754                               0.754    0.754
##    .word.j            0.548                               0.548    0.548
##     vocab             0.218    0.028    7.831    0.000    1.000    1.000
##
## Scales y*:
##                       Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     word.a            1.000                               1.000    1.000
##     word.b            1.000                               1.000    1.000
##     word.c            1.000                               1.000    1.000
##     word.d            1.000                               1.000    1.000
##     word.e            1.000                               1.000    1.000
##     word.f            1.000                               1.000    1.000
##     word.g            1.000                               1.000    1.000
##     word.h            1.000                               1.000    1.000
##     word.i            1.000                               1.000    1.000
##     word.j            1.000                               1.000    1.000
```

b) The easiest item is `word.d`, most difficult is item `word.c`.

c) The strongest indicator is `word.d`, the weakest indicator is `word.a`.

d)

```r
library("ltm")
gssfit.2PL.IRT <- ltm(gssdat ~ z1)
summary(gssfit.2PL.IRT)
```

```
##
## Call:
## ltm(formula = gssdat ~ z1)
```

```
##
## Model Summary:
##     log.Lik      AIC      BIC
##   -13014.36 26068.71 26188.46
##
## Coefficients:
##                  value std.err   z.vals
## Dffclt.word.a -2.1664  0.1493 -14.5121
## Dffclt.word.b -1.9886  0.0934 -21.2975
## Dffclt.word.c  1.1906  0.0781  15.2397
## Dffclt.word.d -1.9808  0.0830 -23.8593
## Dffclt.word.e -1.4070  0.0659 -21.3415
## Dffclt.word.f -1.3768  0.0595 -23.1270
## Dffclt.word.g  0.5111  0.0485  10.5433
## Dffclt.word.h  0.4460  0.0439  10.1641
## Dffclt.word.i -1.4630  0.0944 -15.4925
## Dffclt.word.j  0.6976  0.0427  16.3356
## Dscrmn.word.a  0.9557  0.0799  11.9545
## Dscrmn.word.b  1.9344  0.1612  12.0031
## Dscrmn.word.c  1.0033  0.0712  14.0839
## Dscrmn.word.d  2.6251  0.2542  10.3283
## Dscrmn.word.e  1.6407  0.1143  14.3490
## Dscrmn.word.f  1.9483  0.1419  13.7287
## Dscrmn.word.g  1.0998  0.0714  15.4104
## Dscrmn.word.h  1.2245  0.0781  15.6761
## Dscrmn.word.i  0.9740  0.0719  13.5470
## Dscrmn.word.j  1.6213  0.1058  15.3253
##
## Integration:
## method: Gauss-Hermite
## quadrature points: 21
##
## Optimization:
## Convergence: 0
## max(|grad|): 0.00033
## quasi-Newton: BFGS
```

Most difficult is now `word.a`, easiest is still `word.c`. Best indicator is now `word.d`, weakest indicator is now `word.a`.

e)

```
gssmod.1PL <- '
  vocab =~ lambda*word.a + lambda*word.b + lambda*word.c + lambda*word.d +
              lambda*word.e + lambda*word.f + lambda*word.g + lambda*word.h +
              lambda*word.i + lambda*word.j
'
gssfit.1PL <- cfa(gssmod.1PL, ordered = paste0("word.", letters[1:10]),
                  data = gssdat)
fitinds <- c("chisq.scaled", "df", "pvalue.scaled", "cfi.scaled",
             "rmsea.scaled", "srmr")
fitMeasures(gssfit.1PL, fitinds)
```

```
##   chisq.scaled            df pvalue.scaled    cfi.scaled  rmsea.scaled
##        544.026        44.000         0.000         0.885         0.062
##           srmr
##         0.115
```

```
fitMeasures(gssfit.2PL, fitinds)
```

```
##  chisq.scaled            df pvalue.scaled    cfi.scaled  rmsea.scaled
##       361.903        35.000         0.000         0.925         0.056
##          srmr
##         0.088
```

```
lavTestLRT(gssfit.1PL, gssfit.2PL)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.2000")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##            Df AIC BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## gssfit.2PL 35         285.07
## gssfit.1PL 44         576.98     171.12       9  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

According to CFI, SRMR, RMSEA and the likelihood-ratio test, the 2PL should be preferred.

```
gssfit.1PL.IRT <- rasch(gssdat)
anova(gssfit.1PL.IRT, gssfit.2PL.IRT)
```

```
##
##  Likelihood Ratio Table
##                      AIC      BIC   log.Lik    LRT df p.value
## gssfit.1PL.IRT 26223.83 26289.69 -13100.92
## gssfit.2PL.IRT 26068.71 26188.46 -13014.36 173.12  9  <0.001
```

According to AIC, BIC and the likelihood-ratio test, the 2PL model should be preferred.