# Tutorial

*Factor analysis and latent trait models*

## Lab 1 - Exercise 2: Social mobility in UK

```
cormat <- read.table("socmob.txt")
cormat
```

```
##       V1    V2    V3    V4    V5    V6    V7    V8    V9   V10
## 1  1.000 0.372 0.231 0.100 0.431 0.171 0.128 0.175 0.077 0.293
## 2  0.372 1.000 0.225 0.134 0.375 0.149 0.096 0.184 0.095 0.283
## 3  0.231 0.225 1.000 0.534 0.354 0.276 0.281 0.318 0.247 0.287
## 4  0.100 0.134 0.534 1.000 0.235 0.234 0.380 0.310 0.348 0.215
## 5  0.431 0.375 0.354 0.235 1.000 0.196 0.138 0.231 0.109 0.444
## 6  0.171 0.149 0.276 0.234 0.196 1.000 0.473 0.260 0.115 0.189
## 7  0.128 0.096 0.281 0.380 0.138 0.473 1.000 0.208 0.191 0.161
## 8  0.175 0.184 0.318 0.310 0.231 0.260 0.208 1.000 0.500 0.438
## 9  0.077 0.095 0.247 0.348 0.109 0.115 0.191 0.500 1.000 0.331
## 10 0.293 0.283 0.287 0.215 0.444 0.189 0.161 0.438 0.331 1.000
```

```
str(cormat)
```

```
## 'data.frame':    10 obs. of  10 variables:
##  $ V1 : num  1 0.372 0.231 0.1 0.431 0.171 0.128 0.175 0.077 0.293
##  $ V2 : num  0.372 1 0.225 0.134 0.375 0.149 0.096 0.184 0.095 0.283
##  $ V3 : num  0.231 0.225 1 0.534 0.354 0.276 0.281 0.318 0.247 0.287
##  $ V4 : num  0.1 0.134 0.534 1 0.235 0.234 0.38 0.31 0.348 0.215
##  $ V5 : num  0.431 0.375 0.354 0.235 1 0.196 0.138 0.231 0.109 0.444
##  $ V6 : num  0.171 0.149 0.276 0.234 0.196 1 0.473 0.26 0.115 0.189
##  $ V7 : num  0.128 0.096 0.281 0.38 0.138 0.473 1 0.208 0.191 0.161
##  $ V8 : num  0.175 0.184 0.318 0.31 0.231 0.26 0.208 1 0.5 0.438
##  $ V9 : num  0.077 0.095 0.247 0.348 0.109 0.115 0.191 0.5 1 0.331
##  $ V10: num  0.293 0.283 0.287 0.215 0.444 0.189 0.161 0.438 0.331 1
```

```
cormat <- as.matrix(cormat)

n <- 713

formula <- "V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10"

f1 <- factanal(formula, factors = 1, covmat = cormat, n.obs = n, rotation = "none")

f2 <- factanal(formula, factors = 2, covmat = cormat, n.obs = n, rotation = "none")

f3 <- factanal(formula, factors = 3, covmat = cormat, n.obs = n, rotation = "none")

f4 <- factanal(formula, factors = 4, covmat = cormat, n.obs = n, rotation = "none")

names(f1)
```

```
##  [1] "converged"    "loadings"     "uniquenesses" "correlation"  "criteria"     "factors"      "dof"
##  [8] "method"       "STATISTIC"    "PVAL"         "n.obs"        "call"
```

```r
Chisq <- round(c(f1$STATISTIC, f2$STATISTIC, f3$STATISTIC, f4$STATISTIC), 3)
Chisq

## objective objective objective objective
##   565.338   316.896   143.799    16.616

df <- c(f1$dof, f2$dof, f3$dof, f4$dof)

pvalues <- round(c(f1$PVAL, f2$PVAL, f3$PVAL, f4$PVAL), 4)
pvalues

## objective objective objective objective
##    0.0000    0.0000    0.0000    0.1198

f4

##
## Call:
## factanal(x = formula, factors = 4, covmat = cormat, n.obs = n,     rotation = "none")
##
## Uniquenesses:
##    V1    V2    V3    V4    V5    V6    V7    V8    V9   V10
## 0.650 0.721 0.592 0.078 0.428 0.005 0.691 0.419 0.525 0.541
##
## Loadings:
##       Factor1 Factor2 Factor3 Factor4
##  [1,]  0.122   0.176   0.475  -0.281
##  [2,]  0.157   0.154   0.423  -0.227
##  [3,]  0.529   0.288   0.193
##  [4,]  0.916   0.253  -0.134
##  [5,]  0.271   0.204   0.571  -0.362
##  [6,]          0.997
##  [7,]  0.279   0.480
##  [8,]  0.343   0.269   0.439   0.445
##  [9,]  0.401   0.124   0.270   0.476
## [10,]  0.268   0.197   0.584
##
##                Factor1 Factor2 Factor3 Factor4
## SS loadings      1.661   1.595   1.393   0.702
## Proportion Var   0.166   0.159   0.139   0.070
## Cumulative Var   0.166   0.326   0.465   0.535
##
## Test of the hypothesis that 4 factors are sufficient.
## The chi square statistic is 16.62 on 11 degrees of freedom.
## The p-value is 0.12

loadings(f3)

##
## Loadings:
##       Factor1 Factor2 Factor3
##  [1,]  0.426   0.403
##  [2,]  0.404   0.343
##  [3,]  0.592           0.116
##  [4,]  0.558  -0.240   0.118
##  [5,]  0.575   0.481
##  [6,]  0.451  -0.126   0.369
##  [7,]  0.477  -0.296   0.462
##  [8,]  0.615  -0.191  -0.289
##  [9,]  0.519  -0.358  -0.381
```

```
## [10,]  0.602   0.168  -0.219
##
##                Factor1 Factor2 Factor3
## SS loadings      2.778   0.866   0.657
## Proportion Var   0.278   0.087   0.066
## Cumulative Var   0.278   0.364   0.430
```

```r
print(f3, cutoff = 0.2)
```

```
##
## Call:
## factanal(x = formula, factors = 3, covmat = cormat, n.obs = n,     rotation = "none")
##
## Uniquenesses:
##    V1    V2    V3    V4    V5    V6    V7    V8    V9   V10
## 0.654 0.719 0.636 0.617 0.437 0.645 0.471 0.502 0.458 0.561
##
## Loadings:
##       Factor1 Factor2 Factor3
##  [1,]  0.426   0.403
##  [2,]  0.404   0.343
##  [3,]  0.592
##  [4,]  0.558  -0.240
##  [5,]  0.575   0.481
##  [6,]  0.451           0.369
##  [7,]  0.477  -0.296   0.462
##  [8,]  0.615          -0.289
##  [9,]  0.519  -0.358  -0.381
## [10,]  0.602          -0.219
##
##                Factor1 Factor2 Factor3
## SS loadings      2.778   0.866   0.657
## Proportion Var   0.278   0.087   0.066
## Cumulative Var   0.278   0.364   0.430
##
## Test of the hypothesis that 3 factors are sufficient.
## The chi square statistic is 143.8 on 18 degrees of freedom.
## The p-value is 1.18e-21
```

```r
comm <- 1 - f3$uniquenesses
comm
```

```
##        V1        V2        V3        V4        V5        V6        V7        V8        V9       V10
## 0.3464168 0.2812792 0.3642287 0.3829988 0.5628355 0.3549912 0.5285937 0.4976275 0.5424113 0.4388895
```

```r
percVar <- sum(comm)/nrow(cormat)
percVar
```

```
## [1] 0.4300272
```

```r
repcorr <- loadings(f3) %*% t(loadings(f3))

round(cormat - repcorr, 3)
```

```
##           V1     V2     V3     V4     V5     V6     V7     V8     V9    V10
##  [1,]  0.654  0.061 -0.017 -0.047 -0.009  0.010  0.019  0.005  0.021 -0.019
##  [2,]  0.061  0.719 -0.006 -0.010 -0.023  0.007  0.001  0.003  0.011 -0.016
##  [3,] -0.017 -0.006  0.636  0.184  0.022 -0.037 -0.062 -0.017 -0.025 -0.040
##  [4,] -0.047 -0.010  0.184  0.617  0.026 -0.091 -0.012 -0.045  0.017 -0.055
##  [5,] -0.009 -0.023  0.022  0.026  0.437 -0.014 -0.008 -0.022 -0.006  0.024
```

```
##  [6,]  0.010  0.007 -0.037 -0.091 -0.014  0.645  0.050  0.065 -0.024  0.019
##  [7,]  0.019  0.001 -0.062 -0.012 -0.008  0.050  0.471 -0.008  0.014  0.024
##  [8,]  0.005  0.003 -0.017 -0.045 -0.022  0.065 -0.008  0.502  0.003  0.037
##  [9,]  0.021  0.011 -0.025  0.017 -0.006 -0.024  0.014  0.003  0.458 -0.005
## [10,] -0.019 -0.016 -0.040 -0.055  0.024  0.019  0.024  0.037 -0.005  0.561
```

```
library(GPArotation)

Varimax(loadings(f3))
```

```
## Orthogonal rotation method varimax converged.
## Loadings:
##       Factor1 Factor2 Factor3
##  [1,]  0.0453  0.5784  0.0992
##  [2,]  0.0880  0.5172  0.0774
##  [3,]  0.3004  0.3363  0.4011
##  [4,]  0.3748  0.1441  0.4709
##  [5,]  0.1164  0.7301  0.1273
##  [6,]  0.0957  0.1740  0.5617
##  [7,]  0.1275  0.0557  0.7136
##  [8,]  0.6507  0.2130  0.1700
##  [9,]  0.7268  0.0205  0.1173
## [10,]  0.4360  0.4932  0.0751
##
## Rotating matrix:
##        [,1]   [,2]   [,3]
## [1,]  0.615 0.6004  0.512
## [2,] -0.451 0.7996 -0.396
## [3,] -0.647 0.0123  0.762
```

```
quartimax(loadings(f3))
```

```
## Orthogonal rotation method Quartimax converged.
## Loadings:
##       Factor1 Factor2 Factor3
##  [1,]  0.0346  0.5817  0.0828
##  [2,]  0.0783  0.5209  0.0620
##  [3,]  0.2998  0.3533  0.3867
##  [4,]  0.3794  0.1646  0.4604
##  [5,]  0.1029  0.7356  0.1056
##  [6,]  0.1012  0.1912  0.5551
##  [7,]  0.1380  0.0778  0.7096
##  [8,]  0.6487  0.2311  0.1530
##  [9,]  0.7280  0.0389  0.1042
## [10,]  0.4265  0.5040  0.0543
##
## Rotating matrix:
##        [,1]   [,2]   [,3]
## [1,]  0.610 0.6268  0.485
## [2,] -0.475 0.7789 -0.410
## [3,] -0.634 0.0196  0.773
```

```
oblimin(loadings(f3))
```

```
## Oblique rotation method Oblimin Quartimin converged.
## Loadings:
##        Factor1 Factor2   Factor3
##  [1,] -0.07079  0.6018  0.023281
##  [2,] -0.00938  0.5332  0.000869
##  [3,]  0.18468  0.2652  0.343808
```

```
## [4,]  0.28383  0.0389  0.433738
## [5,] -0.02550  0.7513  0.022116
## [6,] -0.04299  0.0967  0.573922
## [7,] -0.01981 -0.0558  0.750325
## [8,]  0.63381  0.1168  0.051634
## [9,]  0.76038 -0.0933  0.007663
## [10,]  0.37259  0.4602 -0.054734
##
## Rotating matrix:
##        [,1]     [,2]   [,3]
## [1,]  0.457  0.49093  0.378
## [2,] -0.544  0.97551 -0.466
## [3,] -0.863 -0.00344  0.935
##
## Phi:
##       [,1]  [,2]  [,3]
## [1,] 1.000 0.342 0.383
## [2,] 0.342 1.000 0.326
## [3,] 0.383 0.326 1.000
```

```
library(lavaan)
```

```
## This is lavaan 0.6-4.1342
## lavaan is BETA software!  Please report any bugs.
```

```
socmob.model <- "F1 =~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10
                 F2 =~ V1 + V2 + V4 + V5 + V7 + V9
                 F3 =~ V6 + V7 + V8 + V9 + V10"

fit <- cfa(socmob.model, sample.cov = cormat, sample.nobs = n, std.lv = TRUE)
```

```
## Warning in lav_model_vcov(lavmodel = lavmodel, lavsamplestats = lavsamplestats, :  lavaan WARNING:
##    Could not compute standard errors!  The information matrix could
##    not be inverted.  This may be a symptom that the model is not
##    identified.
```

```
summary(fit, fit.measures = T)
```

```
## lavaan 0.6-4.1342 ended normally after 32 iterations
##
##   Optimization method                           NLMINB
##   Number of free parameters                         34
##
##   Number of observations                           713
##
##   Estimator                                         ML
##   Model Fit Test Statistic                     197.962
##   Degrees of freedom                                21
##   P-value (Chi-square)                           0.000
##
## Model test baseline model:
##
##   Minimum Function Test Statistic             1678.006
##   Degrees of freedom                                45
##   P-value                                        0.000
##
## User model versus baseline model:
##
##   Comparative Fit Index (CFI)                    0.892
##   Tucker-Lewis Index (TLI)                       0.768
```

```
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -9372.006
##   Loglikelihood unrestricted model (H1)      -9273.025
##
##   Number of free parameters                         34
##   Akaike (AIC)                               18812.012
##   Bayesian (BIC)                             18967.375
##   Sample-size adjusted Bayesian (BIC)        18859.416
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.109
##   90 Percent Confidence Interval          0.095  0.123
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.048
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##   Standard Errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   F1 =~
##     V1                0.493       NA
##     V2                0.459       NA
##     V3                0.603       NA
##     V4                0.548       NA
##     V5                0.630       NA
##     V6                0.435       NA
##     V7                0.438       NA
##     V8                0.644       NA
##     V9                0.531       NA
##     V10               0.616       NA
##   F2 =~
##     V1                0.470       NA
##     V2                0.409       NA
##     V4               -0.205       NA
##     V5                0.512       NA
##     V7               -0.199       NA
##     V9               -0.416       NA
##   F3 =~
##     V6                0.319       NA
##     V7                0.375       NA
##     V8               -0.338       NA
##     V9               -0.548       NA
##     V10              -0.332       NA
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   F1 ~~
##     F2               -0.215       NA
```

```
##     F3                 0.240       NA
##   F2 ~~
##     F3                -0.294       NA
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .V1                0.635       NA
##    .V2                0.701       NA
##    .V3                0.635       NA
##    .V4                0.608       NA
##    .V5                0.478       NA
##    .V6                0.641       NA
##    .V7                0.467       NA
##    .V8                0.574       NA
##    .V9                0.422       NA
##    .V10               0.607       NA
##     F1                1.000
##     F2                1.000
##     F3                1.000

rm(list=ls())
```

## Lab 2 - Exercise 2: Law School Admission Test

```
library(ltm)

## Loading required package:  MASS
## Loading required package:  msm
## Loading required package:  polycor

data("LSAT")
?LSAT

## starting httpd help server ...
##  done

head(LSAT)

##   Item 1 Item 2 Item 3 Item 4 Item 5
## 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

dim(LSAT)

## [1] 1000    5

dsc <- descript(LSAT)
dsc$perc

##            0     1     logit
## Item 1 0.076 0.924 2.4979787
## Item 2 0.291 0.709 0.8905323
## Item 3 0.447 0.553 0.2127994
## Item 4 0.237 0.763 1.1691979
## Item 5 0.130 0.870 1.9009588
```

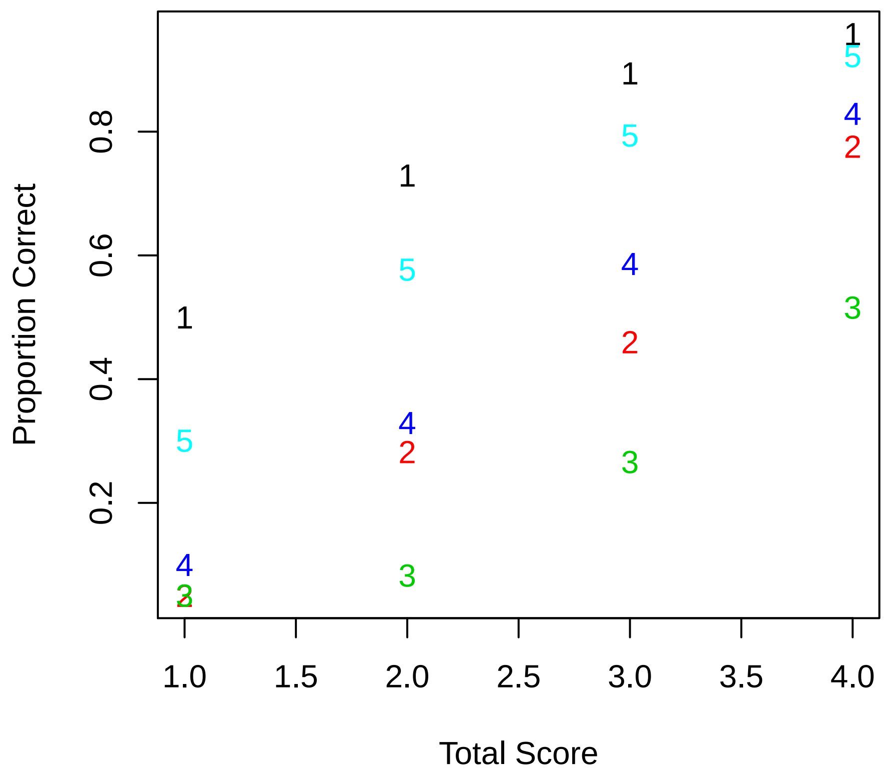
```r
dsc$items

##      0  1  2   3   4   5
## Freq 3 20 85 237 357 298

dsc$pw.ass

##    Item i Item j p.value
## 1       1      5   0.565
## 2       1      4   0.208
## 3       3      5   0.113
## 4       2      4   0.059
## 5       1      2   0.028
## 6       2      5   0.009
## 7       1      3   0.003
## 8       4      5   0.002
## 9       3      4   7e-04
## 10      2      3   4e-04

plot(dsc)
```




```r
m1     <- rasch(LSAT, IRT.param = TRUE, constraint = cbind(ncol(LSAT) + 1, 1))
m1.rip <- rasch(LSAT, IRT.param = FALSE, constraint = cbind(ncol(LSAT)+ 1, 1))

summary(m1.rip)

##
## Call:
```

```
## rasch(data = LSAT, constraint = cbind(ncol(LSAT) + 1, 1), IRT.param = FALSE)
##
## Model Summary:
##    log.Lik      AIC      BIC
##  -2473.054 4956.108 4980.646
##
## Coefficients:
##        value std.err  z.vals
## Item1 2.8720  0.1287 22.3066
## Item2 1.0630  0.0821 12.9458
## Item3 0.2576  0.0766  3.3635
## Item4 1.3881  0.0865 16.0478
## Item5 2.2188  0.1048 21.1660
## z     1.0000      NA      NA
##
## Integration:
## method: Gauss-Hermite
## quadrature points: 21
##
## Optimization:
## Convergence: 0
## max(|grad|): 6.3e-05
## quasi-Newton: BFGS
```

```
coef(m1.rip, prob = TRUE, order = TRUE)
```

```
##           beta.i beta P(x=1|z=0)
## Item 3 0.2576109    1  0.5640489
## Item 2 1.0630294    1  0.7432690
## Item 4 1.3880588    1  0.8002822
## Item 5 2.2187785    1  0.9019232
## Item 1 2.8719712    1  0.9464434
```

```
pval.boot <- GoF.rasch(m1, B = 199, seed = 221019)
pval.boot$Tobs
```

```
## [1] 30.59541
```

```
pval.boot
```

```
##
## Bootstrap Goodness-of-Fit using Pearson chi-squared
##
## Call:
## rasch(data = LSAT, constraint = cbind(ncol(LSAT) + 1, 1), IRT.param = TRUE)
##
## Tobs: 30.6
## # data-sets: 200
## p-value: 0.21
```

```
margins(m1)
```

```
##
## Call:
## rasch(data = LSAT, constraint = cbind(ncol(LSAT) + 1, 1), IRT.param = TRUE)
##
## Fit on the Two-Way Margins
##
## Response: (0,0)
##   Item i Item j Obs   Exp (O-E)^2/E
```

```
## 1      2      4  81 98.69      3.17
## 2      1      5  12 18.45      2.25
## 3      3      5  67 80.04      2.12
##
## Response: (1,0)
##   Item i Item j Obs    Exp (O-E)^2/E
## 1      3      5  63  51.62      2.51
## 2      2      4 156 139.78      1.88
## 3      3      4 108  99.42      0.74
##
## Response: (0,1)
##   Item i Item j Obs    Exp (O-E)^2/E
## 1      2      4 210 193.47      1.41
## 2      2      3 135 125.07      0.79
## 3      1      4  53  47.24      0.70
##
## Response: (1,1)
##   Item i Item j Obs    Exp (O-E)^2/E
## 1      2      4 553 568.06      0.40
## 2      3      5 490 501.43      0.26
## 3      2      3 418 427.98      0.23
```

```r
margins(m1, type = "three-way", nprint = 2)
```

```
##
## Call:
## rasch(data = LSAT, constraint = cbind(ncol(LSAT) + 1, 1), IRT.param = TRUE)
##
## Fit on the Three-Way Margins
##
## Response: (0,0,0)
##   Item i Item j Item k Obs   Exp (O-E)^2/E
## 1      2      3      4  48 66.07      4.94 ***
## 2      1      3      5   6 13.58      4.23 ***
##
## Response: (1,0,0)
##   Item i Item j Item k Obs   Exp (O-E)^2/E
## 1      1      2      4  70 82.01      1.76
## 2      2      4      5  28 22.75      1.21
##
## Response: (0,1,0)
##   Item i Item j Item k Obs   Exp (O-E)^2/E
## 1      1      2      5   3  7.73      2.90
## 2      3      4      5  37 45.58      1.61
##
## Response: (1,1,0)
##   Item i Item j Item k Obs    Exp (O-E)^2/E
## 1      3      4      5  48  36.91      3.33
## 2      1      2      4 144 126.35      2.47
##
## Response: (0,0,1)
##   Item i Item j Item k Obs   Exp (O-E)^2/E
## 1      1      3      5  41 34.58      1.19
## 2      2      4      5  64 72.26      0.94
##
## Response: (1,0,1)
##   Item i Item j Item k Obs    Exp (O-E)^2/E
## 1      1      2      4 190 174.87      1.31
## 2      1      2      3 126 114.66      1.12
```

```
##
## Response: (0,1,1)
##   Item i Item j Item k Obs   Exp (O-E)^2/E
## 1      1      2      5  42 34.35      1.70
## 2      1      4      5  46 38.23      1.58
##
## Response: (1,1,1)
##   Item i Item j Item k Obs    Exp (O-E)^2/E
## 1      3      4      5 397 416.73      0.93
## 2      2      3      4 343 361.18      0.91
##
## '***' denotes a chi-squared residual greater than 3.5
```

```
margins(m1, type = "three-way", nprint = 3)
```

```
##
## Call:
## rasch(data = LSAT, constraint = cbind(ncol(LSAT) + 1, 1), IRT.param = TRUE)
##
## Fit on the Three-Way Margins
##
## Response: (0,0,0)
##   Item i Item j Item k Obs   Exp (O-E)^2/E
## 1      2      3      4  48 66.07      4.94 ***
## 2      1      3      5   6 13.58      4.23 ***
## 3      2      4      5  17 26.43      3.36
##
## Response: (1,0,0)
##   Item i Item j Item k Obs   Exp (O-E)^2/E
## 1      1      2      4  70 82.01      1.76
## 2      2      4      5  28 22.75      1.21
## 3      2      3      4  81 72.98      0.88
##
## Response: (0,1,0)
##   Item i Item j Item k Obs   Exp (O-E)^2/E
## 1      1      2      5   3  7.73      2.90
## 2      3      4      5  37 45.58      1.61
## 3      1      3      4   5  8.61      1.51
##
## Response: (1,1,0)
##   Item i Item j Item k Obs    Exp (O-E)^2/E
## 1      3      4      5  48  36.91      3.33
## 2      1      2      4 144 126.35      2.47
## 3      1      3      5  57  46.76      2.24
##
## Response: (0,0,1)
##   Item i Item j Item k Obs    Exp (O-E)^2/E
## 1      1      3      5  41  34.58      1.19
## 2      2      4      5  64  72.26      0.94
## 3      2      3      4 108 101.01      0.48
##
## Response: (1,0,1)
##   Item i Item j Item k Obs    Exp (O-E)^2/E
## 1      1      2      4 190 174.87      1.31
## 2      1      2      3 126 114.66      1.12
## 3      2      4      5 128 117.03      1.03
##
## Response: (0,1,1)
##   Item i Item j Item k Obs    Exp (O-E)^2/E
```

```
## 1      1      2      5  42  34.35      1.70
## 2      1      4      5  46  38.23      1.58
## 3      3      4      5 281 262.32      1.33
##
## Response: (1,1,1)
##   Item i Item j Item k Obs    Exp (O-E)^2/E
## 1      3      4      5 397 416.73      0.93
## 2      2      3      4 343 361.18      0.91
## 3      2      3      5 377 394.23      0.75
##
## '***' denotes a chi-squared residual greater than 3.5
```

```
m2     <- rasch(LSAT)
m2.rip <- rasch(LSAT,IRT.param=FALSE)

summary(m2.rip)
```

```
##
## Call:
## rasch(data = LSAT, IRT.param = FALSE)
##
## Model Summary:
##    log.Lik      AIC      BIC
##  -2466.938 4945.875 4975.322
##
## Coefficients:
##        value std.err  z.vals
## Item1 2.7300  0.1304 20.9291
## Item2 0.9986  0.0792 12.6123
## Item3 0.2399  0.0718  3.3418
## Item4 1.3065  0.0846 15.4357
## Item5 2.0994  0.1054 19.9099
## z     0.7551  0.0694 10.8757
##
## Integration:
## method: Gauss-Hermite
## quadrature points: 21
##
## Optimization:
## Convergence: 0
## max(|grad|): 2.9e-05
## quasi-Newton: BFGS
```

```
summary(m2)$coefficients[1,1]*summary(m2)$coefficients[6,1]
```

```
## [1] -2.730013
```

```
summary(m2.rip)$coefficients[1,1]
```

```
## [1] 2.730013
```

```
anova(m1,m2)
```

```
##
##  Likelihood Ratio Table
##        AIC     BIC  log.Lik   LRT df p.value
## m1 4956.11 4980.65 -2473.05
## m2 4945.88 4975.32 -2466.94 12.23  1  <0.001
```

```
margins(m2)
```

```
##
## Call:
## rasch(data = LSAT)
##
## Fit on the Two-Way Margins
##
## Response: (0,0)
##   Item i Item j Obs   Exp (O-E)^2/E
## 1      1      3  47 42.47      0.48
## 2      1      5  12 14.55      0.45
## 3      2      4  81 87.21      0.44
##
## Response: (1,0)
##   Item i Item j Obs    Exp (O-E)^2/E
## 1      3      5  63  58.48      0.35
## 2      2      4 156 149.79      0.26
## 3      4      5  85  88.43      0.13
##
## Response: (0,1)
##   Item i Item j Obs    Exp (O-E)^2/E
## 1      1      3  29  33.53      0.61
## 2      2      4 210 203.79      0.19
## 3      1      5  64  61.45      0.11
##
## Response: (1,1)
##   Item i Item j Obs    Exp (O-E)^2/E
## 1      2      4 553 559.21      0.07
## 2      3      5 490 494.53      0.04
## 3      1      3 524 519.47      0.04
```

```
margins(m2, type = "three-way", nprint = 2)
```

```
##
## Call:
## rasch(data = LSAT)
##
## Fit on the Three-Way Margins
##
## Response: (0,0,0)
##   Item i Item j Item k Obs   Exp (O-E)^2/E
## 1      1      3      5   6  9.40      1.23
## 2      3      4      5  30 25.85      0.67
##
## Response: (1,0,0)
##   Item i Item j Item k Obs   Exp (O-E)^2/E
## 1      2      4      5  28 22.75      1.21
## 2      2      3      4  81 74.44      0.58
##
## Response: (0,1,0)
##   Item i Item j Item k Obs  Exp (O-E)^2/E
## 1      1      2      5   3 7.58      2.76
## 2      1      3      4   5 9.21      1.92
##
## Response: (1,1,0)
##   Item i Item j Item k Obs   Exp (O-E)^2/E
## 1      2      4      5  51 57.49      0.73
## 2      3      4      5  48 42.75      0.64
##
## Response: (0,0,1)
```

```
##   Item i Item j Item k Obs    Exp (O-E)^2/E
## 1      1      3      5  41  33.07      1.90
## 2      2      3      4 108 101.28      0.45
##
## Response: (1,0,1)
##   Item i Item j Item k Obs    Exp (O-E)^2/E
## 1      2      3      4 210 218.91      0.36
## 2      1      2      4 190 185.56      0.11
##
## Response: (0,1,1)
##   Item i Item j Item k Obs   Exp (O-E)^2/E
## 1      1      3      5  23 28.38      1.02
## 2      1      4      5  46 42.51      0.29
##
## Response: (1,1,1)
##   Item i Item j Item k Obs    Exp (O-E)^2/E
## 1      1      2      4 520 526.36      0.08
## 2      1      2      3 398 393.30      0.06
```

```
m3     <- ltm(LSAT ~ z1)
m3.rip <- ltm(LSAT ~ z1, IRT.param = FALSE)

summary(m3.rip)
```

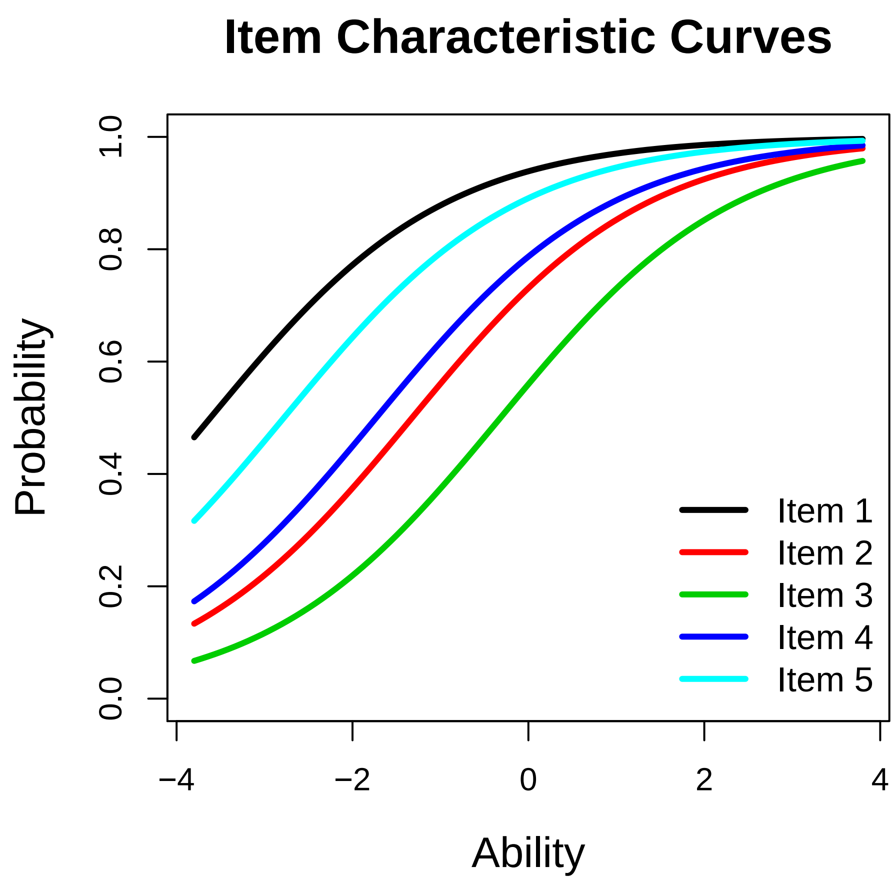
```
##
## Call:
## ltm(formula = LSAT ~ z1, IRT.param = FALSE)
##
## Model Summary:
##    log.Lik      AIC      BIC
##  -2466.653 4953.307 5002.384
##
## Coefficients:
##                      value std.err  z.vals
## (Intercept).Item 1 2.7730  0.2057 13.4824
## (Intercept).Item 2 0.9902  0.0900 10.9987
## (Intercept).Item 3 0.2492  0.0763  3.2681
## (Intercept).Item 4 1.2848  0.0990 12.9711
## (Intercept).Item 5 2.0536  0.1354 15.1620
## z1.Item 1          0.8254  0.2581  3.1983
## z1.Item 2          0.7229  0.1867  3.8721
## z1.Item 3          0.8905  0.2326  3.8281
## z1.Item 4          0.6886  0.1852  3.7186
## z1.Item 5          0.6575  0.2100  3.1306
##
## Integration:
## method: Gauss-Hermite
## quadrature points: 21
##
## Optimization:
## Convergence: 0
## max(|grad|): 0.024
## quasi-Newton: BFGS
```

```
anova(m2,m3)
```

```
##
##  Likelihood Ratio Table
##        AIC     BIC  log.Lik  LRT df p.value
```

```
## m2 4945.88 4975.32 -2466.94
## m3 4953.31 5002.38 -2466.65 0.57  4   0.967

plot(m2, legend = TRUE, cx = "bottomright", lwd = 3, cex.main = 1.5,
     cex.lab = 1.3, cex = 1.1)
```

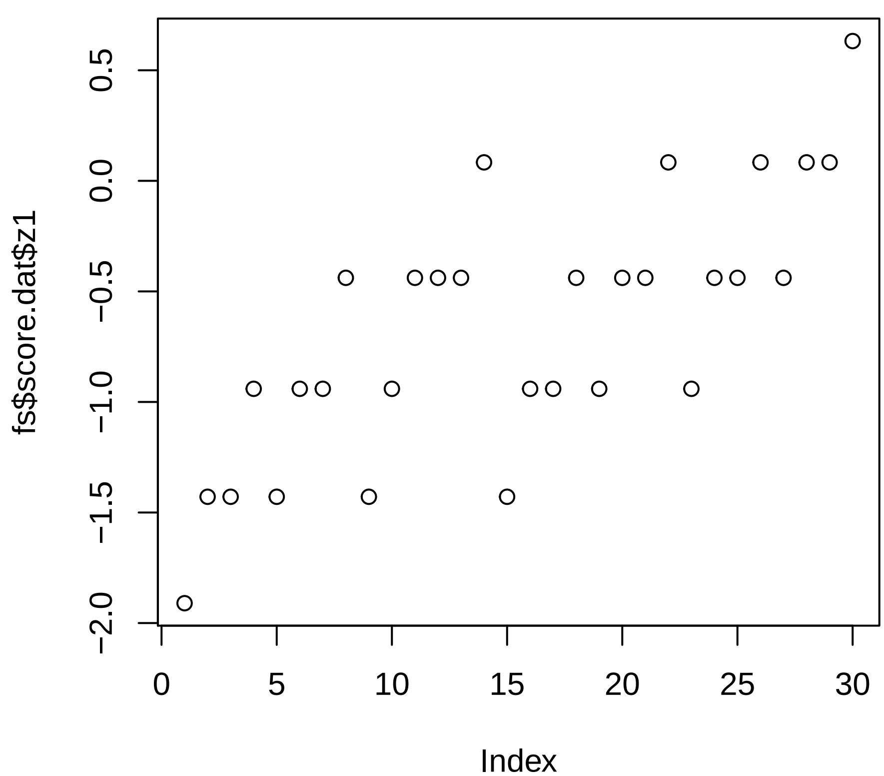



```
fs<-factor.scores(m2, method="EAP")
fs

##
## Call:
## rasch(data = LSAT)
##
## Scoring Method: Expected A Posteriori
##
## Factor-Scores for observed response patterns:
##    Item 1 Item 2 Item 3 Item 4 Item 5 Obs    Exp     z1 se.z1
## 1       0      0      0      0      0   3  2.364 -1.910 0.797
## 2       0      0      0      0      1   6  5.468 -1.429 0.800
## 3       0      0      0      1      0   2  2.474 -1.429 0.800
## 4       0      0      0      1      1  11  8.249 -0.941 0.809
## 5       0      0      1      0      0   1  0.852 -1.429 0.800
## 6       0      0      1      0      1   1  2.839 -0.941 0.809
## 7       0      0      1      1      0   3  1.285 -0.941 0.809
## 8       0      0      1      1      1   4  6.222 -0.439 0.823
## 9       0      1      0      0      0   1  1.819 -1.429 0.800
## 10      0      1      0      0      1   8  6.063 -0.941 0.809
## 11      0      1      0      1      1  16 13.288 -0.439 0.823
```

```
## 12      0      1      1      0      1   3   4.574 -0.439 0.823
## 13      0      1      1      1      0   2   2.070 -0.439 0.823
## 14      0      1      1      1      1  15  14.749  0.084 0.841
## 15      1      0      0      0      0  10  10.273 -1.429 0.800
## 16      1      0      0      0      1  29  34.249 -0.941 0.809
## 17      1      0      0      1      0  14  15.498 -0.941 0.809
## 18      1      0      0      1      1  81  75.060 -0.439 0.823
## 19      1      0      1      0      0   3   5.334 -0.941 0.809
## 20      1      0      1      0      1  28  25.834 -0.439 0.823
## 21      1      0      1      1      0  15  11.690 -0.439 0.823
## 22      1      0      1      1      1  80  83.310  0.084 0.841
## 23      1      1      0      0      0  16  11.391 -0.941 0.809
## 24      1      1      0      0      1  56  55.171 -0.439 0.823
## 25      1      1      0      1      0  21  24.965 -0.439 0.823
## 26      1      1      0      1      1 173 177.918  0.084 0.841
## 27      1      1      1      0      0  11   8.592 -0.439 0.823
## 28      1      1      1      0      1  61  61.235  0.084 0.841
## 29      1      1      1      1      0  28  27.709  0.084 0.841
## 30      1      1      1      1      1 298 295.767  0.632 0.864
```

```
plot(fs$score.dat$z1)
```




```
resp.pattern <- fs$score.dat[,1:5]
total.score  <- apply(resp.pattern,1,sum)
total.score

##  [1] 0 1 1 2 1 2 2 3 1 2 3 3 3 4 1 2 2 3 2 3 3 4 2 3 3 4 3 4 4 5
```

```r
round(fs$score.dat[order(total.score),],3)
```

```
##    Item 1 Item 2 Item 3 Item 4 Item 5 Obs     Exp     z1 se.z1
## 1       0      0      0      0      0   3   2.364 -1.910 0.797
## 2       0      0      0      0      1   6   5.468 -1.429 0.800
## 3       0      0      0      1      0   2   2.474 -1.429 0.800
## 5       0      0      1      0      0   1   0.852 -1.429 0.800
## 9       0      1      0      0      0   1   1.819 -1.429 0.800
## 15      1      0      0      0      0  10  10.273 -1.429 0.800
## 4       0      0      0      1      1  11   8.249 -0.941 0.809
## 6       0      0      1      0      1   1   2.839 -0.941 0.809
## 7       0      0      1      1      0   3   1.285 -0.941 0.809
## 10      0      1      0      0      1   8   6.063 -0.941 0.809
## 16      1      0      0      0      1  29  34.249 -0.941 0.809
## 17      1      0      0      1      0  14  15.498 -0.941 0.809
## 19      1      0      1      0      0   3   5.334 -0.941 0.809
## 23      1      1      0      0      0  16  11.391 -0.941 0.809
## 8       0      0      1      1      1   4   6.222 -0.439 0.823
## 11      0      1      0      1      1  16  13.288 -0.439 0.823
## 12      0      1      1      0      1   3   4.574 -0.439 0.823
## 13      0      1      1      1      0   2   2.070 -0.439 0.823
## 18      1      0      0      1      1  81  75.060 -0.439 0.823
## 20      1      0      1      0      1  28  25.834 -0.439 0.823
## 21      1      0      1      1      0  15  11.690 -0.439 0.823
## 24      1      1      0      0      1  56  55.171 -0.439 0.823
## 25      1      1      0      1      0  21  24.965 -0.439 0.823
## 27      1      1      1      0      0  11   8.592 -0.439 0.823
## 14      0      1      1      1      1  15  14.749  0.084 0.841
## 22      1      0      1      1      1  80  83.310  0.084 0.841
## 26      1      1      0      1      1 173 177.918  0.084 0.841
## 28      1      1      1      0      1  61  61.235  0.084 0.841
## 29      1      1      1      1      0  28  27.709  0.084 0.841
## 30      1      1      1      1      1 298 295.767  0.632 0.864
```

```r
factor.scores(m2,resp.pattern=rbind(c(0,1,1,0,0), c(0,1,0,1,0)))
```

```
##
## Call:
## rasch(data = LSAT)
##
## Scoring Method: Empirical Bayes
##
## Factor-Scores for specified response patterns:
##   Item 1 Item 2 Item 3 Item 4 Item 5 Obs   Exp     z1 se.z1
## 1      0      1      1      0      0   0 0.944 -0.959 0.801
## 2      0      1      0      1      0   0 2.744 -0.959 0.801
```