Mock exam: Petrella Riccardo

First load the library and the data set

```
library(ltm)
```

```
## Warning: package 'ltm' was built under R version 4.3.2

## Loading required package: MASS

## Loading required package: msm

## Warning: package 'msm' was built under R version 4.3.2

## Loading required package: polycor

## Warning: package 'polycor' was built under R version 4.3.2
```

```
cheating <- read.table("cheating.txt")
```

We have 319 observations and 4 dichotomous variables.

POINT 1_A The total number of theoretical response patterns are:

```
2^4
```

```
## [1] 16
```

POINT 1_B *Which is the item with the highest proportion of no?*

```
dsc <- descript(cheating)
dsc$perc #Contains the percentage of negative and positive responses for each item
```

```
##                  0          1     logit
## LIEEXAM  0.8934169 0.10658307 -2.126129
## LIEPAPER 0.8808777 0.11912226 -2.000769
## FRAUD    0.9341693 0.06583072 -2.652571
## COPYEXAM 0.7868339 0.21316614 -1.305945
```

The item that has the highest percentage (proportion) of NO is FRAUD.

POINT 1_C *Which is the observed frequency of the response pattern (0,0,0,0)? And the ob- served frequency of the response pattern (1,1,1,1)?* It allow to get a numeric matrix that includes the frequencies for each total score

```
dsc$items
```

```
##        0  1  2 3 4
## Freq 207 76 25 9 2
```

Total score equal to zero means that this is the frequency of the response pattern that has all ZERO. So the observed frequency of a response patter all equal to 0 is 207 while in order to have the observed frequency of the response pattern of all 1 we look at the total score of 4= we have 4 ones–> 2

POINT 1_D *Are there pairs of items that are not significantly associated? Which are they?* It is a matrix that contains th p-values of the pairwise association of the p items. The H0 is that the are DEPENDENT. The pair items that are not ASSOCIATED are 2-4 and 1-4.

```
dsc$pw.ass
```

```
##   Item i Item j p.value
## 1      2      4    0.15
## 2      1      4    0.15
## 3      2      3    0.03
## 4      1      3    0.01
## 5      1      2   0.001
## 6      3      4   0.001
```

POINT 3_A *Which is the most difficult item? And the easiest one? Report and comment the values of the difficulty parameters for these two items.* Fit the latent trait model for only one binary factor

```
m1.rip <- ltm(cheating ~ z1,IRT.param=FALSE) #Name of the data set ~ factors
summary(m1.rip)
```

```
##
## Call:
## ltm(formula = cheating ~ z1, IRT.param = FALSE)
##
## Model Summary:
##  log.Lik      AIC      BIC
##  -440.23 896.4599 926.5815
##
## Coefficients:
##                         value std.err  z.vals
## (Intercept).LIEEXAM  -4.4732  1.9143 -2.3368
## (Intercept).LIEPAPER -3.6944  1.1547 -3.1994
## (Intercept).FRAUD    -3.0801  0.3725 -8.2678
## (Intercept).COPYEXAM -1.3791  0.1556 -8.8660
## z1.LIEEXAM            3.1164  1.7328  1.7984
## z1.LIEPAPER           2.5815  1.1390  2.2665
## z1.FRAUD              1.0279  0.3692  2.7839
## z1.COPYEXAM           0.5163  0.2282  2.2626
##
## Integration:
## method: Gauss-Hermite
## quadrature points: 21
##
```

```
## Optimization:
## Convergence: 0
## max(|grad|): 0.0076
## quasi-Newton: BFGS
```

The most difficult item is the one with the lowest intercept so it is LIEEXAM while the easiest is COPYEXAM because it has the higher value for the intercept (difficulty parameter). If we want to fit the 2PL model we need to change IRT.param in TRUE.

POINT 3_B *Rank the items according to the discrimination parameters reporting the values of these coefficients.* Rank the items with respect to the discrimination parameter

```
alpha <- m1.rip$coefficients[,2]
alpha
```

```
##   LIEEXAM  LIEPAPER     FRAUD  COPYEXAM
## 3.1163675 2.5815212 1.0279310 0.5162571
```

We can extract and look at the discrimination parameters. LIEEXAM has the largest discrimination parameter so it is the item that discriminates more between individuals, while COPYEXAM is the one with the smallest discrimination parameter.

POINT 4 *Compute and comment the standardized alpha's and the probabilities of the median individual* Standardize the discrimination parameter

```
stalpha <- alpha/sqrt(1+alpha^2)
stalpha
```

```
##   LIEEXAM  LIEPAPER     FRAUD  COPYEXAM
## 0.9521788 0.9324828 0.7167779 0.4587328
```

What we get is actually the same information of the non-standardized alpha, but these alpha are rescaled so they have values that goes from 0 to 1

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

```
##          (Intercept)        z1 P(x=1|z=0)
## LIEEXAM    -4.473222 3.1163675 0.01128176
## LIEPAPER   -3.694397 2.5815212 0.02425930
## FRAUD      -3.080135 1.0279310 0.04393416
## COPYEXAM   -1.379139 0.5162571 0.20114731
```

In the last column we have the probability of Xi when Z=0 –> definition of the probability of the median individual. The probability that a median individual (y=0) responds positively to COPIEXAM is the highest, while it is the smallest fot LIEXAM. This is natural because COPIEXAM is the easiest item while LIEXAM is the most difficult.

POINT 5 *Report the values of the goodness of fit tests, the degrees of freedom and the p-values and comment the goodness of fit of the model. Are the tests reliable? If not why?*

```
E <- fitted(m1.rip)[,5] #Expected frequencies
O <- m1.rip$patterns$obs #Observed frequencies
cbind(m1.rip$patterns$X, O, E)
```

```
##                   O           E
##  [1,] 0 0 0 0 207 205.6982523
##  [2,] 0 0 0 1  46  47.1411191
##  [3,] 0 0 1 0   7   9.1736797
##  [4,] 0 0 1 1   5   2.8187912
##  [5,] 0 1 0 0  13  12.5501381
##  [6,] 0 1 0 1   4   5.1930619
##  [7,] 0 1 1 0   1   1.6705821
##  [8,] 0 1 1 1   2   0.7982847
##  [9,] 1 0 0 0  10   9.6501979
## [10,] 1 0 0 1   3   4.3174125
## [11,] 1 0 1 0   1   1.4908193
## [12,] 1 0 1 1   2   0.7622705
## [13,] 1 1 0 0  11   8.2686623
## [14,] 1 1 0 1   4   5.1611361
## [15,] 1 1 1 0   1   2.4914920
## [16,] 1 1 1 1   2   1.8141004
```

```
Chisq <- sum((E-O)^2/E)
Chisq
```

```
## [1] 9.268555
```

```
DOF <- 16-4*2-1 #2^p - p(q+1) - 1
DOF
```

```
## [1] 7
```

```
pvalueC <- 1 - pchisq(Chisq, DOF)
pvalueC #According to the Chi-square test our model has a good fit to the data
```

```
## [1] 0.2339457
```

```
LR <- 2 * sum(O * log(O/E))
LR
```

```
## [1] 8.169943
```

```
pvalueLR <- 1 - pchisq(LR, DOF)
pvalueLR #According to the LR test our model has a good fit to the data
```

```
## [1] 0.3178474
```

But there are some patterns that have expected frequencies lower than 1. So we shouldn't rely on the values of these tests because the Chi-square result doesn't hold anymore.

POINT 6 *Which alternative measures of fit can be considered? What do they suggest?* We inspect the residuals and in particular the two-way margins

```
margins(m1.rip)
```

```
##
## Call:
## ltm(formula = cheating ~ z1, IRT.param = FALSE)
##
## Fit on the Two-Way Margins
##
## Response: (0,0)
##   Item i Item j Obs    Exp (O-E)^2/E
## 1      3      4 241 236.17      0.10
## 2      1      4 228 229.09      0.01
## 3      2      4 225 226.01      0.00
##
## Response: (1,0)
##   Item i Item j Obs   Exp (O-E)^2/E
## 1      3      4  10 14.83      1.57
## 2      1      4  23 21.90      0.06
## 3      2      4  26 24.98      0.04
##
## Response: (0,1)
##   Item i Item j Obs   Exp (O-E)^2/E
## 1      3      4  57 61.81      0.37
## 2      2      3  15 14.25      0.04
## 3      1      3  15 14.46      0.02
##
## Response: (1,1)
##   Item i Item j Obs   Exp (O-E)^2/E
## 1      3      4  11  6.19      3.73 ***
## 2      1      4  11 12.05      0.09
## 3      2      3   6  6.77      0.09
##
## '***' denotes a chi-squared residual greater than 3.5
```

There is just one value greater than 3.7 but our cutoff is 4 so we can conclude that our univariate latent trait model has a good fit to the data.

POINT 7 *Give an interpretation of the latent variable, and illustrate the different methods for scaling individuals.*

```
fs1 <- factor.scores(m1.rip, method = "Component") #Compute the component scores
fs <- factor.scores(m1.rip, method = "EAP") #Compute the posterior mean scores
```

We get the classification, a score for each pattern observed in the model. To response pattern all equal to zero corresponds the lowest score while to the pattern with all ones corresponds the highest score of the latent variable.

So the latent variable can be interpreted as the cheating behavior, so people that don't have a cheating behavior and have responded NO to all the items have a very low score while people that have responded YES have a strong cheating behavior and a highest score of the latent variable.

The two results (Component and EAP) give exactly the same ranking result.

There is another way to compute the scores – TOTAL SCORE The rank that we got before (Components and EAP) is different from the following one (TOTAL SCORE).

```
resp.pattern <- fs$score.dat[,1:4]
total.score <- apply(resp.pattern, 1, sum)
total.score #Total score of the patterns
```

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

```
round(fs$score.dat[order(total.score),],3) #We get the components scores but what change is the order o
```

```
##    LIEEXAM LIEPAPER FRAUD COPYEXAM Obs     Exp     z1 se.z1
## 1        0        0     0        0 207 205.698 -0.354 0.836
## 2        0        0     0        1  46  47.141 -0.019 0.773
## 3        0        0     1        0   7   9.174  0.262 0.711
## 5        0        1     0        0  13  12.550  0.883 0.564
## 9        1        0     0        0  10   9.650  1.043 0.530
## 4        0        0     1        1   5   2.819  0.502 0.654
## 6        0        1     0        1   4   5.193  1.038 0.531
## 7        0        1     1        0   1   1.671  1.176 0.508
## 10       1        0     0        1   3   4.317  1.182 0.507
## 11       1        0     1        0   1   1.491  1.310 0.494
## 13       1        1     0        0  11   8.269  1.690 0.508
## 8        0        1     1        1   2   0.798  1.305 0.494
## 12       1        0     1        1   2   0.762  1.434 0.490
## 14       1        1     0        1   4   5.161  1.828 0.526
## 15       1        1     1        0   1   2.491  1.976 0.550
## 16       1        1     1        1   2   1.814  2.140 0.580
```