ENEM 2018 Studies (In construction)

This is an initial study about Item Theory Response (IRT) and regression models. The main objective of this work is to study the social measures that have an impact on students' proficiency considering the Brazilian High School Exam (Exame Nacional do Ensino Médio - ENEM) in 2018.

Item Theory Response

Installation of packages and data clean for IRT

Installation of the packages to clean and extract knowledge from ENEM dataset.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(tidyr)
library(stringr)
library(splitstackshape)
library(mirt)
```

```
## Loading required package: stats4
```

```
## Loading required package: lattice
```

```
library(GGally)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'GGally'
```

```
## The following object is masked from 'package:dplyr':
##
##     nasa
```

Due the size of the raw data we read the data from a local directory. To illustrate the analysis built here, we use only the first 100k individuals. This will be modefied for all students in other analysis.

```r
setwd("F:/microdados/DADOS")
memory.limit(24576)
```

```
## [1] 24576
```

```r
dat <- data.table::fread(input='enem2018.csv',
                         integer64='character',
                         skip=0,  #Ler do inicio
                         nrows=100000,
                         na.strings = "",
                         showProgress = TRUE)
```

The construction of IRT model is made from four test that compose the exam (CN - Ciencias e Natureza, CH - Ciencias Humanas, Linguagens e Codigos, MT - Matematica e suas Tecnologias which indicate Sciences and Nature, Sciences and Humans, Languages, and Codes, and Mathematic and its Technology, respectively). For this, We build the answers given by students and answer key datasets (`ans` and `gab`, respectively) droping missing values for all items in each test.

```r
ans <- dat %>% select(TX_RESPOSTAS_CN, TX_RESPOSTAS_CH,
                      TX_RESPOSTAS_LC, TX_RESPOSTAS_MT)
gab <- dat %>% select(TX_GABARITO_CN, TX_GABARITO_CH,
                      TX_GABARITO_LC, TX_GABARITO_MT)

ans <- ans %>% drop_na()
gab <- gab %>% drop_na()
```

The number of questions is computed. Also, the answers given by the students for each test are stored in four datasets as can be seen below. Similarly is made with the answers key.

```r
numb_question <- gab[1, ] %>% str_length()

names_cn <- paste('Q', seq_len(numb_question[1]), sep = '')
names_ch <- paste('Q', seq_len(numb_question[2]), sep = '')
names_lc <- paste('Q', seq_len(numb_question[3]), sep = '')
names_mt <- paste('Q', seq_len(numb_question[4]), sep = '')


ans_cn <-  ans[, 1] %>%
  cSplit('TX_RESPOSTAS_CN', sep = '', stripWhite = FALSE)

ans_ch <-  ans[, 2] %>%
  cSplit('TX_RESPOSTAS_CH', sep = '', stripWhite = FALSE)

ans_lc <-  ans[, 3] %>%
  cSplit('TX_RESPOSTAS_LC', sep = '', stripWhite = FALSE)

ans_mt <-  ans[, 4] %>%
  cSplit('TX_RESPOSTAS_MT', sep = '', stripWhite = FALSE)


gab_cn <-  gab[, 1] %>%
  cSplit('TX_GABARITO_CN', sep = '', stripWhite = FALSE)

gab_ch <-  gab[, 2] %>%
  cSplit('TX_GABARITO_CH', sep = '', stripWhite = FALSE)
```

```
gab_lc <-  gab[, 3] %>%
  cSplit('TX_GABARITO_LC', sep = '', stripWhite = FALSE)

gab_mt <-  gab[, 4] %>%
  cSplit('TX_GABARITO_MT', sep = '', stripWhite = FALSE)
```

Next, we renamed the answers and answers key for `Q.`, where `.` indicates the question number.

```
names(ans_ch) <- names_cn
names(ans_cn) <- names_ch
names(ans_lc) <- names_lc
names(ans_mt) <- names_mt

names(gab_ch) <- names_cn
names(gab_cn) <- names_ch
names(gab_lc) <- names_lc
names(gab_mt) <- names_mt
```

To apply the IRT model with three parameters we define as one the correct answers and zero otherwise.

```
#Correction of questions
cor_cn <- + (as.matrix(ans_cn) == as.matrix(gab_cn))
cor_ch <- + (as.matrix(ans_ch) == as.matrix(gab_ch))
cor_lc <- + (as.matrix(ans_lc) == as.matrix(gab_lc))
cor_mt <- + (as.matrix(ans_mt) == as.matrix(gab_mt))
```

IRT Modeling from Three Parameters Model

From `mirt` function the Multivariate Item Response Theory (MIRT) was applied. We highlight that the problem is univariate, i.e. there is only one population in study.

```
model_cn <- mirt(cor_cn, 1, type = '3PL')
```

```
##
Iteration: 1, Log-Lik: -1863789.207, Max-Change: 0.34687
Iteration: 2, Log-Lik: -1854744.863, Max-Change: 0.13563
Iteration: 3, Log-Lik: -1851208.257, Max-Change: 0.10557
Iteration: 4, Log-Lik: -1848970.303, Max-Change: 0.05312
Iteration: 5, Log-Lik: -1848213.395, Max-Change: 0.03254
Iteration: 6, Log-Lik: -1847975.922, Max-Change: 0.01881
Iteration: 7, Log-Lik: -1847904.830, Max-Change: 0.01073
Iteration: 8, Log-Lik: -1847884.414, Max-Change: 0.00600
Iteration: 9, Log-Lik: -1847878.572, Max-Change: 0.00348
Iteration: 10, Log-Lik: -1847876.824, Max-Change: 0.00201
Iteration: 11, Log-Lik: -1847876.306, Max-Change: 0.00118
Iteration: 12, Log-Lik: -1847876.133, Max-Change: 0.00071
Iteration: 13, Log-Lik: -1847876.063, Max-Change: 0.00038
Iteration: 14, Log-Lik: -1847876.049, Max-Change: 0.00022
Iteration: 15, Log-Lik: -1847876.043, Max-Change: 0.00015
Iteration: 16, Log-Lik: -1847876.037, Max-Change: 0.00006
```

```
model_ch <- mirt(cor_ch, 1, type = '3PL')
```

```
##
Iteration: 1, Log-Lik: -2054671.967, Max-Change: 0.36664
```

```
Iteration: 2, Log-Lik: -2048453.537, Max-Change: 0.07017
Iteration: 3, Log-Lik: -2047787.609, Max-Change: 0.02902
Iteration: 4, Log-Lik: -2047544.513, Max-Change: 0.02128
Iteration: 5, Log-Lik: -2047424.893, Max-Change: 0.01548
Iteration: 6, Log-Lik: -2047364.977, Max-Change: 0.01117
Iteration: 7, Log-Lik: -2047334.992, Max-Change: 0.00773
Iteration: 8, Log-Lik: -2047320.067, Max-Change: 0.00537
Iteration: 9, Log-Lik: -2047312.425, Max-Change: 0.00357
Iteration: 10, Log-Lik: -2047308.559, Max-Change: 0.00246
Iteration: 11, Log-Lik: -2047306.456, Max-Change: 0.00180
Iteration: 12, Log-Lik: -2047305.303, Max-Change: 0.00139
Iteration: 13, Log-Lik: -2047303.873, Max-Change: 0.00064
Iteration: 14, Log-Lik: -2047303.738, Max-Change: 0.00059
Iteration: 15, Log-Lik: -2047303.647, Max-Change: 0.00048
Iteration: 16, Log-Lik: -2047303.451, Max-Change: 0.00007
```

```r
model_lc <- mirt(cor_lc, 1, type = '3PL')
```

```
##
Iteration: 1, Log-Lik: -2257650.510, Max-Change: 0.86321
Iteration: 2, Log-Lik: -2227922.516, Max-Change: 0.51755
Iteration: 3, Log-Lik: -2212395.699, Max-Change: 0.39130
Iteration: 4, Log-Lik: -2203864.065, Max-Change: 0.43517
Iteration: 5, Log-Lik: -2199173.798, Max-Change: 0.31647
Iteration: 6, Log-Lik: -2196530.354, Max-Change: 0.19718
Iteration: 7, Log-Lik: -2194768.601, Max-Change: 0.16637
Iteration: 8, Log-Lik: -2193241.779, Max-Change: 0.15290
Iteration: 9, Log-Lik: -2192106.649, Max-Change: 0.15541
Iteration: 10, Log-Lik: -2191263.524, Max-Change: 0.14605
Iteration: 11, Log-Lik: -2190647.320, Max-Change: 0.13450
Iteration: 12, Log-Lik: -2190219.455, Max-Change: 0.11511
Iteration: 13, Log-Lik: -2189944.659, Max-Change: 0.09738
Iteration: 14, Log-Lik: -2189774.773, Max-Change: 0.08150
Iteration: 15, Log-Lik: -2189673.304, Max-Change: 0.06325
Iteration: 16, Log-Lik: -2189616.547, Max-Change: 0.04883
Iteration: 17, Log-Lik: -2189585.303, Max-Change: 0.03740
Iteration: 18, Log-Lik: -2189568.216, Max-Change: 0.02841
Iteration: 19, Log-Lik: -2189558.874, Max-Change: 0.02140
Iteration: 20, Log-Lik: -2189553.744, Max-Change: 0.01598
Iteration: 21, Log-Lik: -2189550.902, Max-Change: 0.01183
Iteration: 22, Log-Lik: -2189547.499, Max-Change: 0.00119
Iteration: 23, Log-Lik: -2189547.347, Max-Change: 0.00111
Iteration: 24, Log-Lik: -2189547.250, Max-Change: 0.00092
Iteration: 25, Log-Lik: -2189546.914, Max-Change: 0.00027
Iteration: 26, Log-Lik: -2189546.906, Max-Change: 0.00025
Iteration: 27, Log-Lik: -2189546.900, Max-Change: 0.00111
Iteration: 28, Log-Lik: -2189546.871, Max-Change: 0.00070
Iteration: 29, Log-Lik: -2189546.857, Max-Change: 0.00064
Iteration: 30, Log-Lik: -2189546.850, Max-Change: 0.00027
Iteration: 31, Log-Lik: -2189546.848, Max-Change: 0.00017
Iteration: 32, Log-Lik: -2189546.845, Max-Change: 0.00046
Iteration: 33, Log-Lik: -2189546.839, Max-Change: 0.00011
Iteration: 34, Log-Lik: -2189546.839, Max-Change: 0.00009
```

```
model_mt <- mirt(cor_mt, 1, type = '3PL')
```

```
##
Iteration: 1, Log-Lik: -1838457.584, Max-Change: 0.43471
Iteration: 2, Log-Lik: -1827632.613, Max-Change: 0.20241
Iteration: 3, Log-Lik: -1823280.926, Max-Change: 0.08799
Iteration: 4, Log-Lik: -1822037.093, Max-Change: 0.04325
Iteration: 5, Log-Lik: -1821642.682, Max-Change: 0.02729
Iteration: 6, Log-Lik: -1821519.331, Max-Change: 0.01617
Iteration: 7, Log-Lik: -1821481.784, Max-Change: 0.00904
Iteration: 8, Log-Lik: -1821470.670, Max-Change: 0.00499
Iteration: 9, Log-Lik: -1821467.389, Max-Change: 0.00284
Iteration: 10, Log-Lik: -1821466.203, Max-Change: 0.00125
Iteration: 11, Log-Lik: -1821466.010, Max-Change: 0.00070
Iteration: 12, Log-Lik: -1821465.946, Max-Change: 0.00049
Iteration: 13, Log-Lik: -1821465.933, Max-Change: 0.00029
Iteration: 14, Log-Lik: -1821465.924, Max-Change: 0.00014
Iteration: 15, Log-Lik: -1821465.920, Max-Change: 0.00009
```

Coefficients and performance are displayed below.

```
#Coefficients
coef_cn <- coef(model_cn, simplify = TRUE, IRTpars = TRUE)
coef_ch <- coef(model_ch, simplify = TRUE, IRTpars = TRUE)
coef_lc <- coef(model_lc, simplify = TRUE, IRTpars = TRUE)
coef_mt <- coef(model_mt, simplify = TRUE, IRTpars = TRUE)

#M2 performance measure
M2(model_cn)
```

```
##             M2  df p      RMSEA   RMSEA_5   RMSEA_95     SRMSR       TLI
## stats 51579.48 945 0 0.02710437 0.0269054 0.02730349 0.0265551 0.7037644
##             CFI
## stats 0.7172297
```

```
M2(model_ch)
```

```
##             M2  df p      RMSEA    RMSEA_5   RMSEA_95      SRMSR       TLI
## stats 158251.9 945 0 0.04777383 0.04757523 0.04797207 0.04335033 0.8105813
##             CFI
## stats 0.8191912
```

```
M2(model_lc)
```

```
##             M2   df p      RMSEA    RMSEA_5  RMSEA_95      SRMSR       TLI
## stats 353037.5 1175 0 0.06407662 0.06389844 0.0642541 0.06153831 0.6480307
##             CFI
## stats 0.6623968
```

```
M2(model_mt)
```

```
##             M2  df p      RMSEA    RMSEA_5   RMSEA_95     SRMSR       TLI
## stats 69304.51 945 0 0.03149311 0.03129432 0.03169193 0.0305916 0.6773698
##             CFI
## stats 0.6920348
```

Once that the modeling is made we calculate the scores for each student and each test in (0, 1) scale. The (0, 1) means that the scores are have mean zero and standard deviation 100.

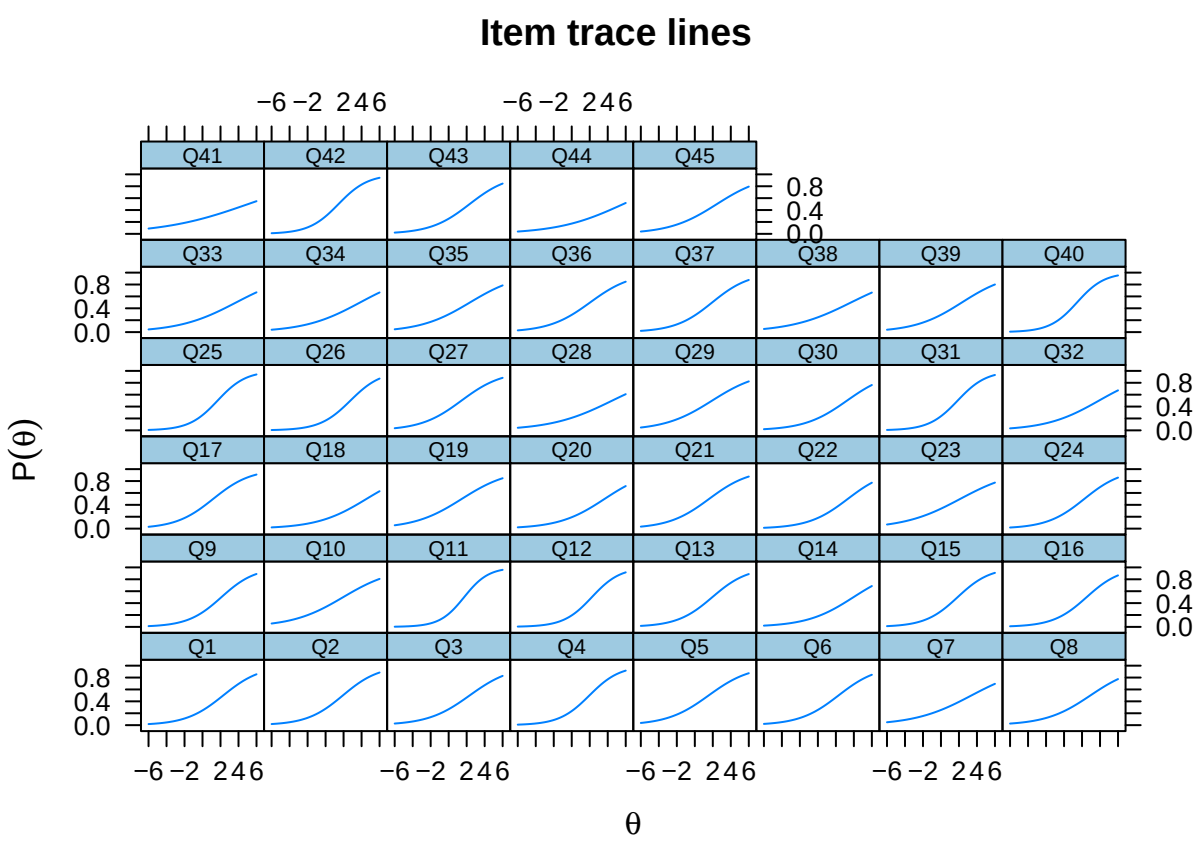
Proficiency and Item Study

```
prof_cn <- fscores(model_cn, method = 'EAP')
prof_ch <- fscores(model_ch, method = 'EAP')
prof_lc <- fscores(model_lc, method = 'EAP')
prof_mt <- fscores(model_mt, method = 'EAP')
```

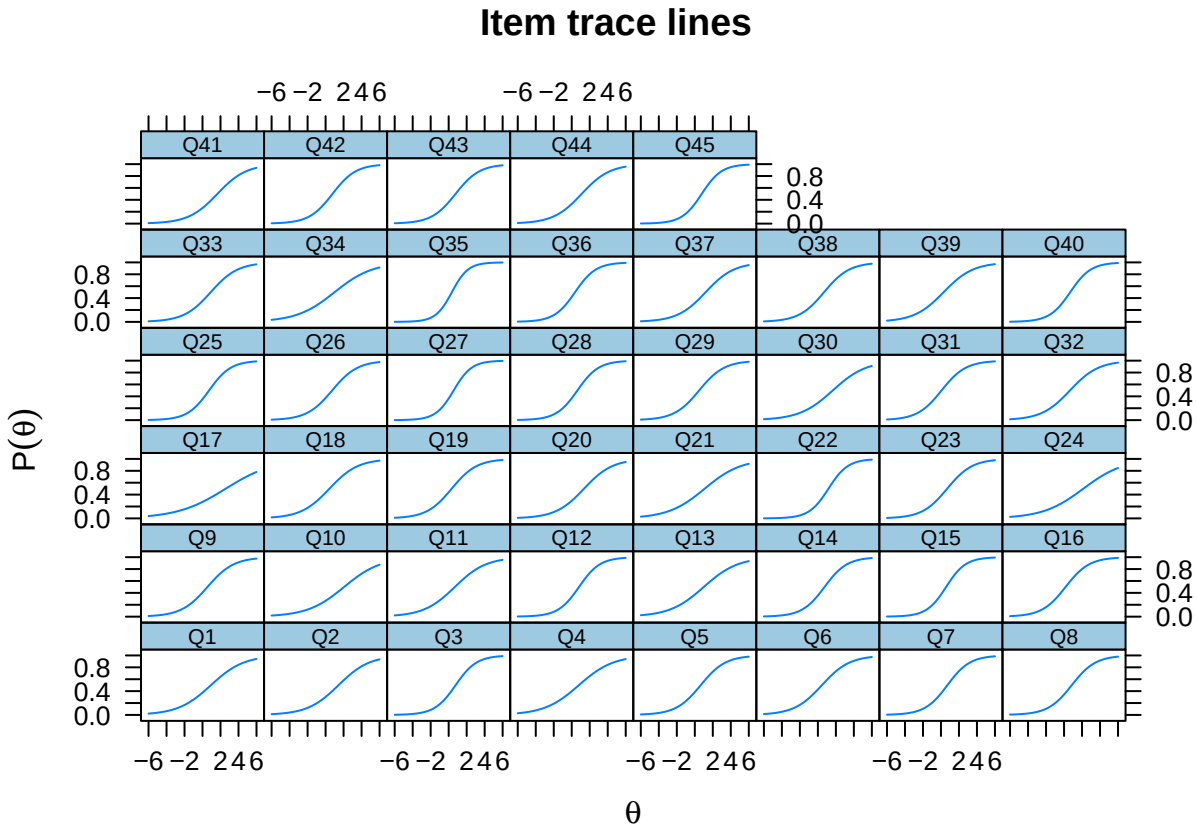
Based on the ENEM IRT approach, we resize the scores. The adopted range is (500, 100); the more or less distant from 500, the student will be considered more or less proficient, respectively.

```
scale_change <- function(value, mu, sigma){
  sigma * value + mu
}

prof_cn <- apply(prof_cn, 1, function(x) scale_change(x, 500, 100) )
prof_ch <- apply(prof_ch, 1, function(x) scale_change(x, 500, 100) )
prof_lc <- apply(prof_lc, 1, function(x) scale_change(x, 500, 100) )
prof_mt <- apply(prof_mt, 1, function(x) scale_change(x, 500, 100) )

prof <- data.frame(CN = prof_cn, CH = prof_ch, LC = prof_lc, MT = prof_mt)
```

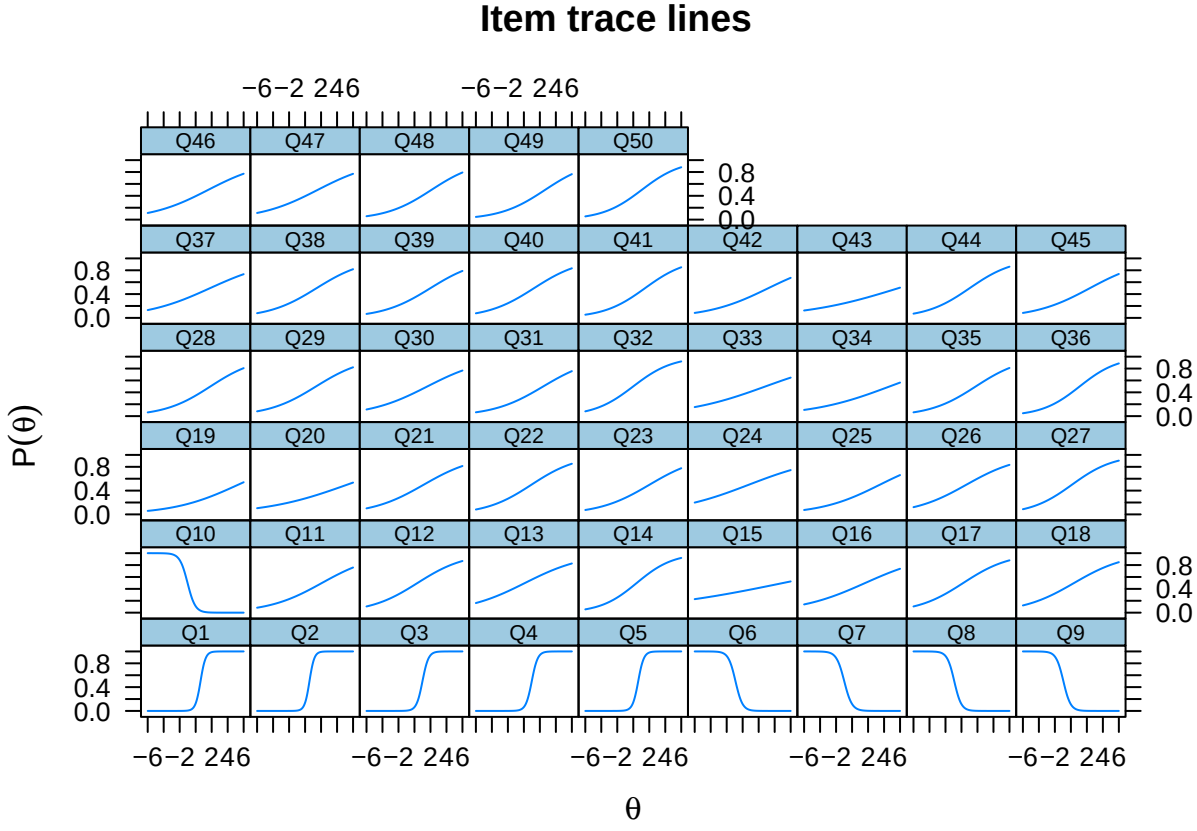
```
plot(model_cn, type = 'trace')
```



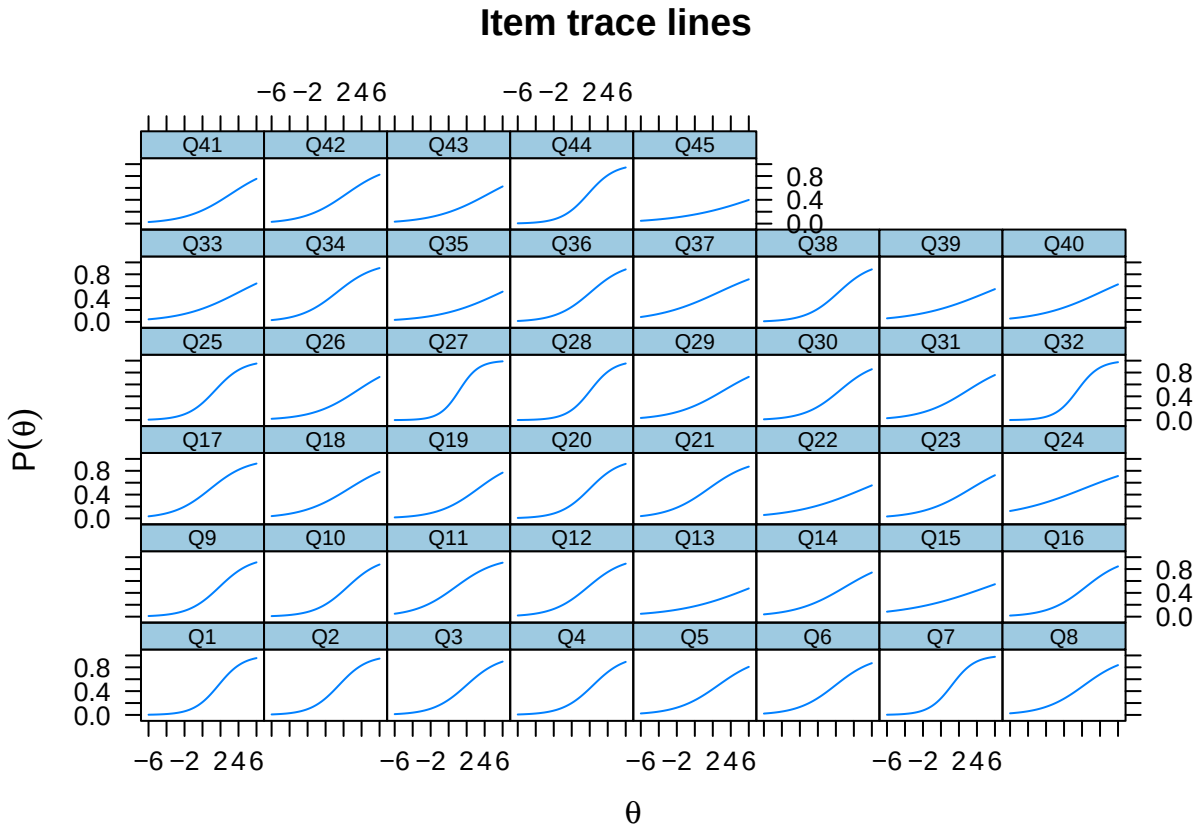

```
plot(model_ch, type = 'trace')
```



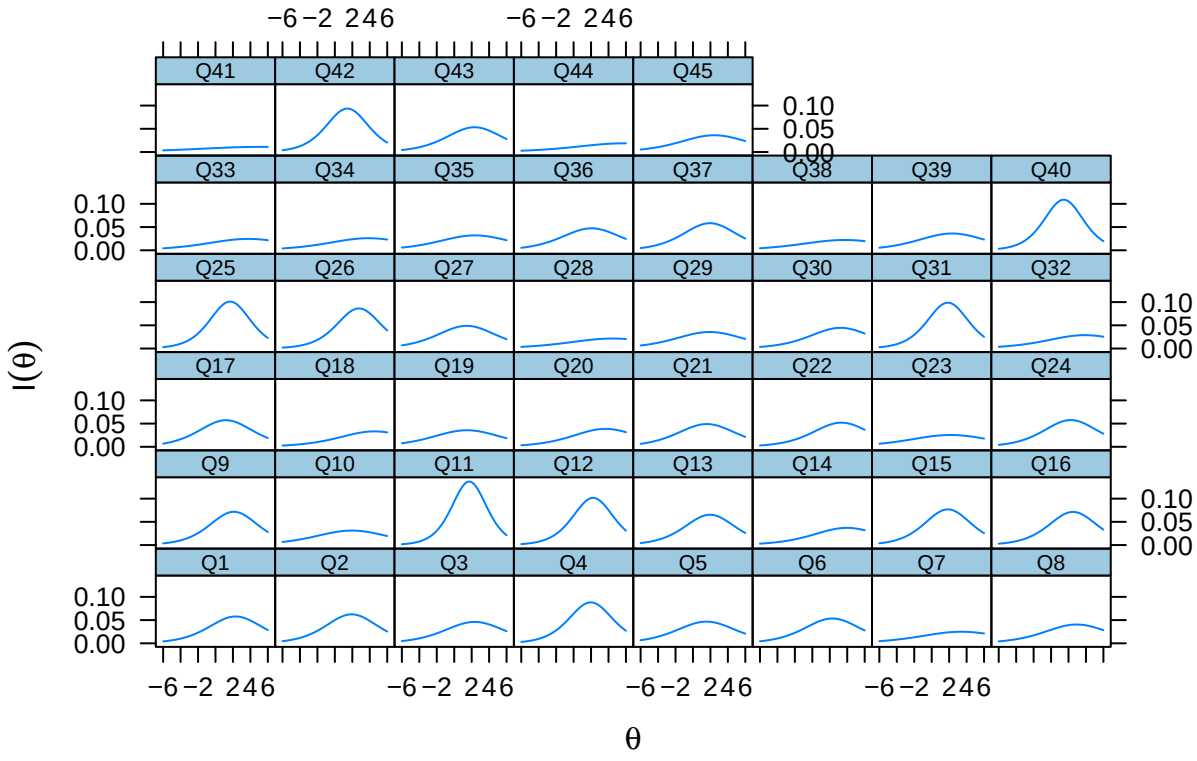

```
plot(model_lc, type = 'trace')
```



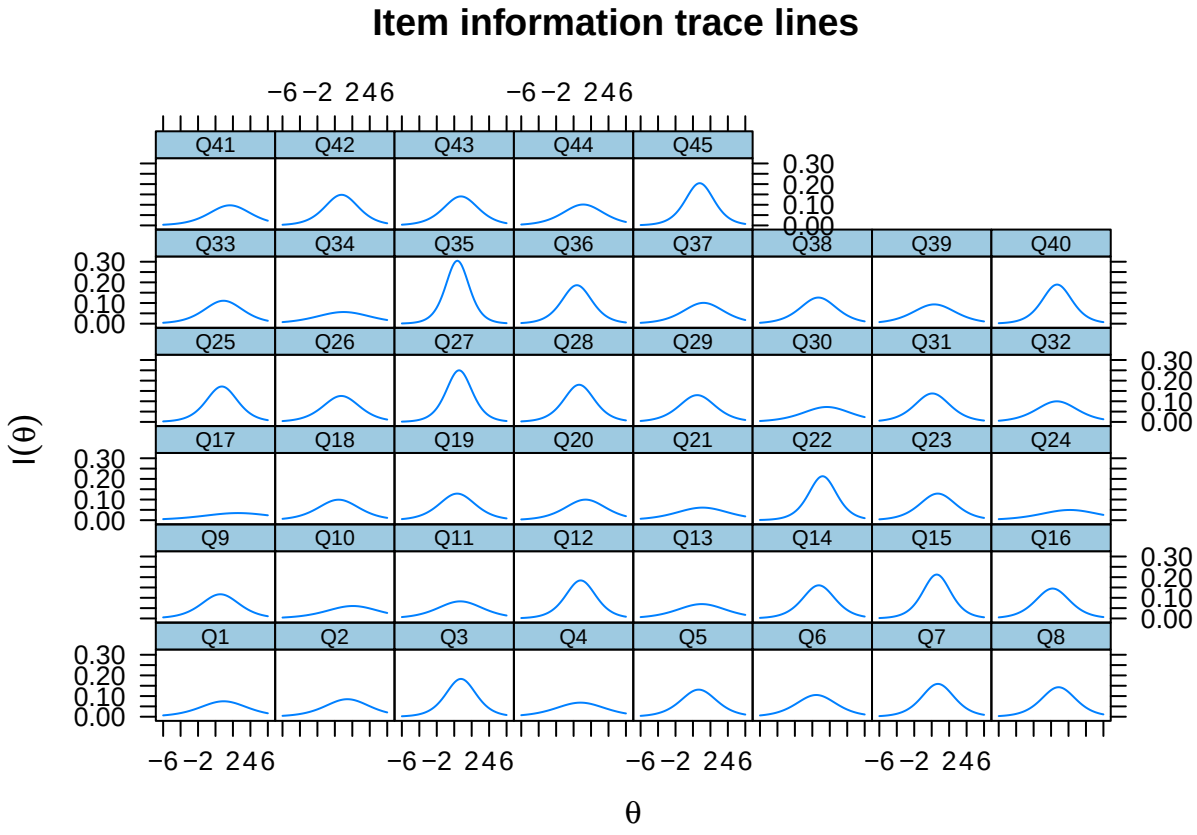

```
plot(model_mt, type = 'trace')
```



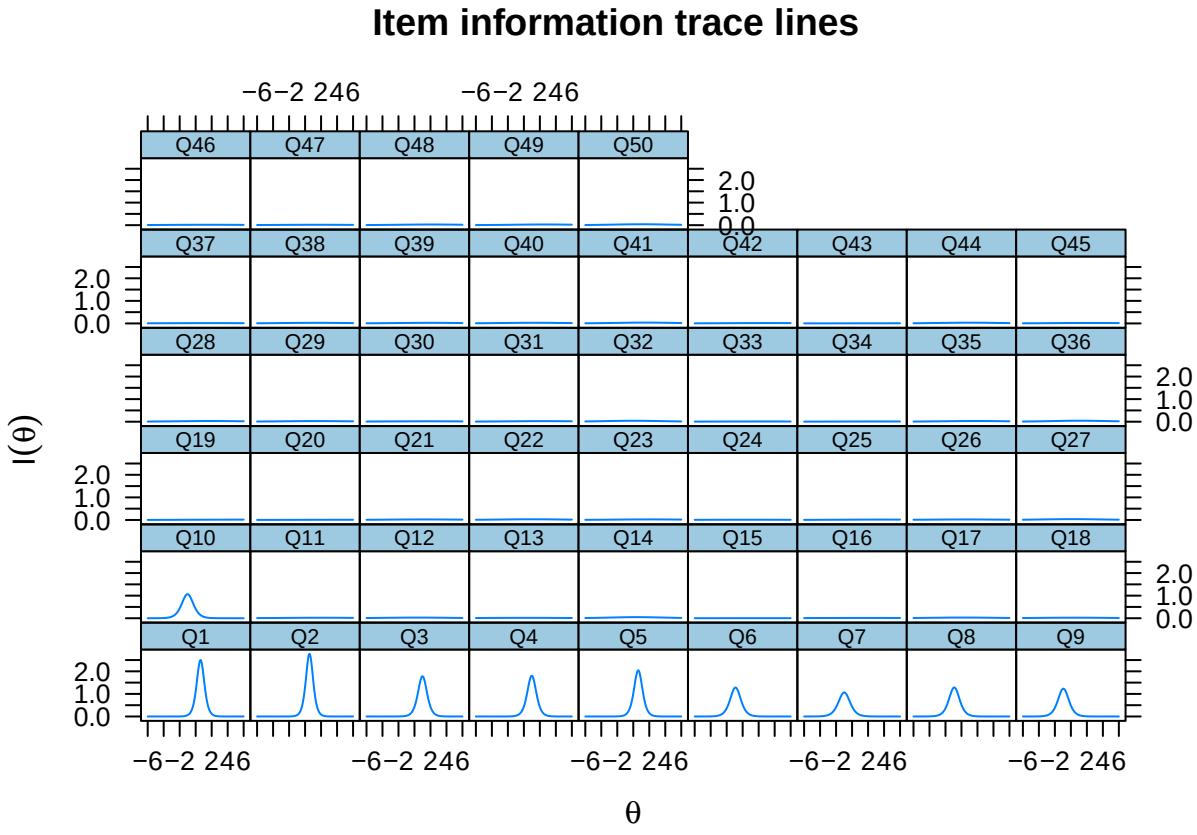

```
plot(model_cn, type = 'infotrace')
```



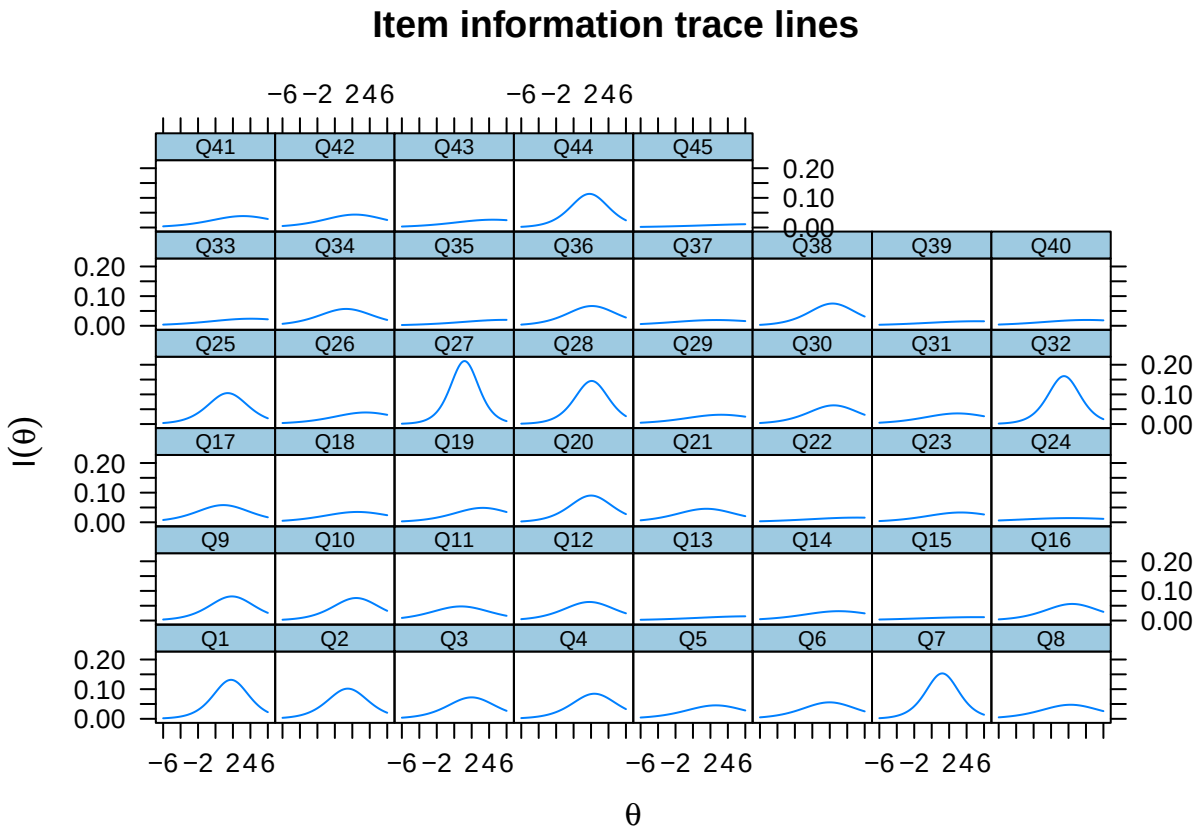

```
plot(model_ch, type = 'infotrace')
```




```
plot(model_lc, type = 'infotrace')
```




```
plot(model_mt, type = 'infotrace')
```



It is fundamental the quantitaty of information stored by each item (question). Then, we calculate the information for each item from some theta values.

```
theta <- matrix(seq(-4, 4, by = .1))
seq_cn <- matrix(seq_len(length(names_cn)), ncol = 1)
seq_ch <- matrix(seq_len(length(names_ch)), ncol = 1)
seq_lc <- matrix(seq_len(length(names_lc)), ncol = 1)
seq_mt <- matrix(seq_len(length(names_mt)), ncol = 1)

extr_item_cn <- lapply(seq_cn, function(x) extract.item(model_cn, x))
extr_item_ch <- lapply(seq_ch, function(x) extract.item(model_ch, x))
extr_item_lc <- lapply(seq_lc, function(x) extract.item(model_lc, x))
extr_item_mt <- lapply(seq_mt, function(x) extract.item(model_mt, x))

info_item_cn <- lapply(extr_item_cn, function(x) iteminfo(x, theta))
info_item_ch <- lapply(extr_item_ch, function(x) iteminfo(x, theta))
info_item_lc <- lapply(extr_item_lc, function(x) iteminfo(x, theta))
info_item_mt <- lapply(extr_item_mt, function(x) iteminfo(x, theta))
```

The `b` parameter indicates the difficulty of item. The first most hard question for each test are: (i) CN - 44, 41, 18, 28, and 14; (ii) CH - 17, 24, 10, 30, and 41; (iii) LC - 43, 19, 20, 15, and 34; (iv) MT - 45, 13, 35, 39, and 15.

```
#The items are sorted from difficulty parameters b
dif_cn <- data.frame(question = seq_cn, coef_cn$items) %>% arrange(desc(b))
dif_ch <- data.frame(question = seq_ch, coef_ch$items) %>% arrange(desc(b))
dif_lc <- data.frame(question = seq_lc, coef_lc$items) %>% arrange(desc(b))
```

```r
dif_mt <- data.frame(question = seq_mt, coef_mt$items) %>% arrange(desc(b))

dif_cn
```

```
##    question         a        b g u
## 1        44 0.2736131 5.722439 0 1
## 2        41 0.2102391 5.064263 0 1
## 3        18 0.3646191 4.550572 0 1
## 4        28 0.2935206 4.496155 0 1
## 5        14 0.3848404 3.959543 0 1
## 6        32 0.3396802 3.879636 0 1
## 7        34 0.3223968 3.855409 0 1
## 8        33 0.3128808 3.754086 0 1
## 9        38 0.2976220 3.704189 0 1
## 10       20 0.3932157 3.654624 0 1
## 11        7 0.3159116 3.394761 0 1
## 12       22 0.4560492 3.327088 0 1
## 13       30 0.4222999 3.234281 0 1
## 14        8 0.4024587 2.934273 0 1
## 15       26 0.5878114 2.756467 0 1
## 16       16 0.5343742 2.512226 0 1
## 17       45 0.3800679 2.448881 0 1
## 18       35 0.3582549 2.383126 0 1
## 19       39 0.3803480 2.346198 0 1
## 20        3 0.4292997 2.336653 0 1
## 21       43 0.4620022 2.325417 0 1
## 22        1 0.4808084 2.306046 0 1
## 23        6 0.4625448 2.296820 0 1
## 24       24 0.4798350 2.257286 0 1
## 25       12 0.6373896 2.236825 0 1
## 26       23 0.3188198 2.145760 0 1
## 27        9 0.5353627 2.102445 0 1
## 28       36 0.4348441 2.055135 0 1
## 29        4 0.5939997 1.965931 0 1
## 30       10 0.3532767 1.955637 0 1
## 31       13 0.5110171 1.948143 0 1
## 32       37 0.4829969 1.941957 0 1
## 33        2 0.4996131 1.922262 0 1
## 34       29 0.3773944 1.921043 0 1
## 35       15 0.5544505 1.852979 0 1
## 36       31 0.6285061 1.851756 0 1
## 37       11 0.7392769 1.719120 0 1
## 38       25 0.6358293 1.674237 0 1
## 39       21 0.4424970 1.590961 0 1
## 40        5 0.4322788 1.540267 0 1
## 41       19 0.3774920 1.461663 0 1
## 42       40 0.6597612 1.451856 0 1
## 43       42 0.6118530 1.432335 0 1
## 44       27 0.4423782 1.422094 0 1
## 45       17 0.4784984 1.188980 0 1
```

```r
dif_ch
```

```
##    question         a         b g u
## 1        17 0.3712957 2.5829667 0 1
```

```
## 2        24 0.4448539 2.1463865 0 1
## 3        10 0.4909497 2.0608289 0 1
## 4        30 0.5379026 1.6597434 0 1
## 5        41 0.6236978 1.6524706 0 1
## 6         2 0.5823677 1.4353212 0 1
## 7        20 0.6311380 1.3672536 0 1
## 8        37 0.6347616 1.2085651 0 1
## 9        22 0.9226570 1.1758278 0 1
## 10       44 0.6361197 1.1042386 0 1
## 11       21 0.4934124 1.0966692 0 1
## 12       34 0.4747776 1.0047350 0 1
## 13       13 0.5273607 0.9996144 0 1
## 14        1 0.5465508 0.9354617 0 1
## 15       33 0.6661682 0.9180928 0 1
## 16        8 0.7557462 0.8606925 0 1
## 17       12 0.8581275 0.8106197 0 1
## 18        4 0.5220950 0.7917373 0 1
## 19        3 0.8555971 0.7700860 0 1
## 20       45 0.9041268 0.7697245 0 1
## 21       42 0.7698201 0.7681003 0 1
## 22       43 0.7494261 0.7646267 0 1
## 23       14 0.8009308 0.7405460 0 1
## 24       26 0.7092462 0.7336600 0 1
## 25       25 0.8283480 0.7330305 0 1
## 26        7 0.7970768 0.7132217 0 1
## 27       38 0.7111031 0.6921923 0 1
## 28       23 0.7177611 0.6921074 0 1
## 29       40 0.8701288 0.6900061 0 1
## 30       11 0.5753282 0.6899819 0 1
## 31       32 0.6317620 0.6642475 0 1
## 32        5 0.7241853 0.6531374 0 1
## 33       28 0.8484112 0.6328157 0 1
## 34       15 0.9220726 0.5749770 0 1
## 35       27 0.9996631 0.5577797 0 1
## 36        9 0.6842776 0.5538882 0 1
## 37       29 0.7202028 0.4850987 0 1
## 38        6 0.6486444 0.4462238 0 1
## 39       18 0.6299640 0.4296137 0 1
## 40       36 0.8629388 0.3737931 0 1
## 41       35 1.1035027 0.3589727 0 1
## 42       19 0.7172973 0.3369543 0 1
## 43       39 0.6107946 0.3001366 0 1
## 44       16 0.7609942 0.1756724 0 1
## 45       31 0.7424125 0.0722013 0 1
```

```
dif_lc
```

```
##    question          a           b g u
## 1        43  0.1654961  5.805041922 0 1
## 2        19  0.2442078  5.330344285 0 1
## 3        20  0.1907692  5.269432985 0 1
## 4        15  0.1103563  5.097312549 0 1
## 5        34  0.2003837  4.710480546 0 1
## 6        25  0.2660041  3.486116163 0 1
## 7        42  0.2599288  3.218596438 0 1
```

```
## 8        33  0.1938458  2.859542763 0 1
## 9        49  0.3518455  2.653685693 0 1
## 10       31  0.3170719  2.429664229 0 1
## 11       45  0.2862465  2.380098561 0 1
## 12       48  0.3456099  2.124303564 0 1
## 13       11  0.2955444  2.120317978 0 1
## 14       23  0.3137941  2.038735235 0 1
## 15       39  0.3299782  1.997624193 0 1
## 16       28  0.3443880  1.849558549 0 1
## 17       37  0.2424978  1.801890610 0 1
## 18       35  0.3446595  1.793649874 0 1
## 19       16  0.2386403  1.660942726 0 1
## 20       30  0.2718210  1.619347568 0 1
## 21       47  0.2731605  1.578270976 0 1
## 22       46  0.2742423  1.564134038 0 1
## 23       41  0.3844992  1.476012332 0 1
## 24       38  0.3301915  1.442623735 0 1
## 25       29  0.3297299  1.376650728 0 1
## 26       40  0.3411044  1.279665257 0 1
## 27       21  0.3045747  1.164873341 0 1
## 28       36  0.4191364  1.104474538 0 1
## 29       50  0.4045726  1.064786396 0 1
## 30       44  0.3658488  1.026739911 0 1
## 31        3  2.6706596  1.001045565 0 1
## 32        4  2.6888400  0.987515891 0 1
## 33       22  0.3464222  0.956478320 0 1
## 34       24  0.2053435  0.789239387 0 1
## 35       26  0.2989356  0.623121585 0 1
## 36        5  2.8611482  0.619048160 0 1
## 37        1  3.1636215  0.608304094 0 1
## 38        2  3.3328276  0.550740918 0 1
## 39       14  0.4394879  0.439720766 0 1
## 40       18  0.3085721  0.413848827 0 1
## 41       12  0.3363837  0.391644545 0 1
## 42       17  0.3447480  0.233265690 0 1
## 43       13  0.2672098  0.154754182 0 1
## 44       27  0.3829957  0.144155288 0 1
## 45       32  0.4059186 -0.003672932 0 1
## 46        6 -2.2662209 -0.901173800 0 1
## 47        8 -2.2680339 -0.902692038 0 1
## 48        9 -2.2208652 -0.937826075 0 1
## 49        7 -2.0630042 -0.981339003 0 1
## 50       10 -2.0640044 -1.018507031 0 1
```

```
dif_mt
```

```
##    question         a         b g u
## 1        45 0.2157861 7.9071391 0 1
## 2        13 0.2399718 6.4295978 0 1
## 3        35 0.2826795 5.8861728 0 1
## 4        39 0.2471891 5.1750045 0 1
## 5        15 0.2142830 5.1554684 0 1
## 6        22 0.2516305 5.1249196 0 1
## 7        43 0.3246423 4.4304166 0 1
## 8        40 0.2802242 4.0920761 0 1
```

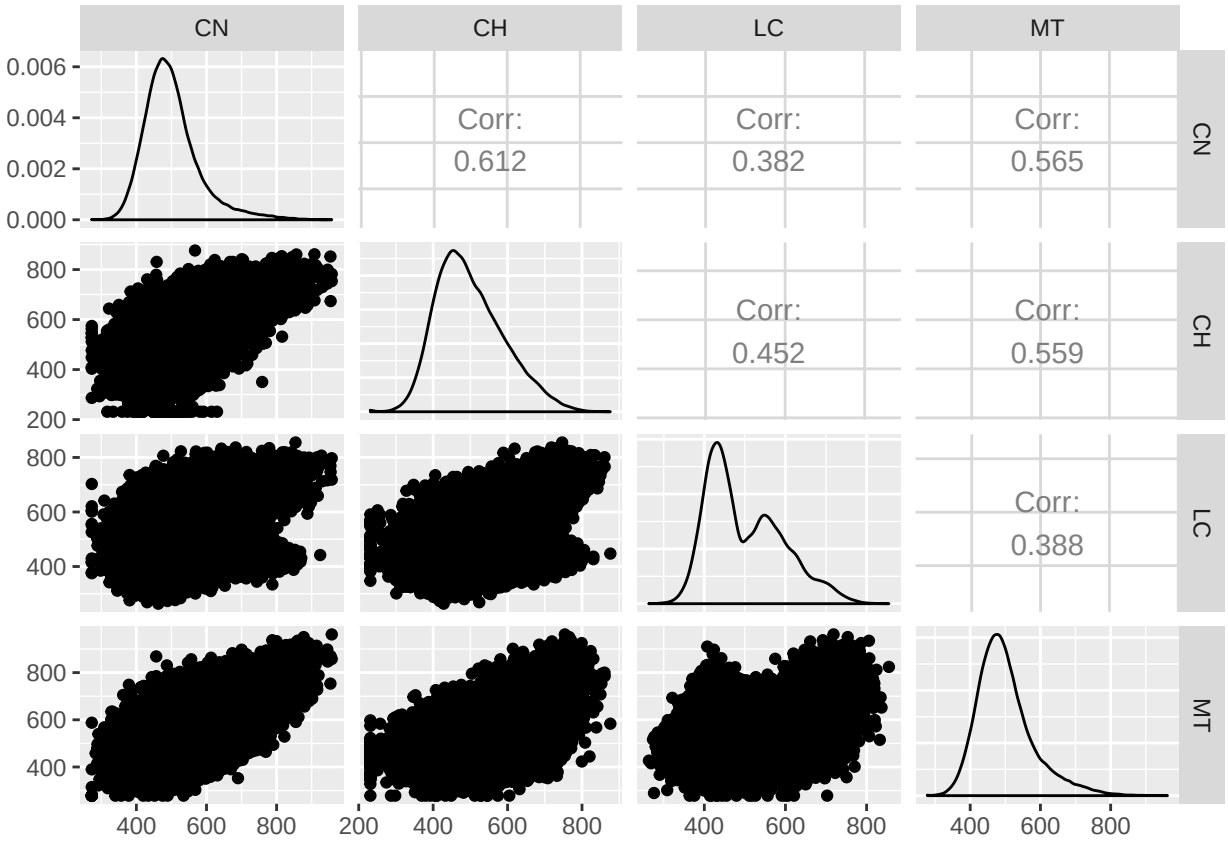
```
## 9        33 0.3086251 4.0558499 0 1
## 10       26 0.3949155 3.5372582 0 1
## 11       23 0.3638034 3.3255925 0 1
## 12       19 0.4412557 3.2705650 0 1
## 13       29 0.3539539 3.2240039 0 1
## 14       41 0.3928172 3.1632732 0 1
## 15       14 0.3557447 3.0323329 0 1
## 16       31 0.3785380 2.9736860 0 1
## 17       37 0.2783362 2.6824774 0 1
## 18        5 0.4261017 2.6218347 0 1
## 19       18 0.3737093 2.5963001 0 1
## 20       30 0.5020252 2.4532880 0 1
## 21       10 0.5514149 2.4409848 0 1
## 22       16 0.4734618 2.4173305 0 1
## 23        4 0.5817465 2.3814554 0 1
## 24       42 0.4174522 2.3184478 0 1
## 25       38 0.5476557 2.3171469 0 1
## 26        8 0.4355288 2.2420199 0 1
## 27       24 0.2381070 2.1885394 0 1
## 28       36 0.5162423 2.1058542 0 1
## 29       28 0.7623556 2.0728181 0 1
## 30       20 0.6007116 2.0053542 0 1
## 31        3 0.5379103 1.9892934 0 1
## 32        6 0.4713948 1.9649562 0 1
## 33        9 0.5707738 1.8817179 0 1
## 34       44 0.6736351 1.8231810 0 1
## 35       12 0.5007560 1.8099738 0 1
## 36        1 0.7251569 1.7787841 0 1
## 37       21 0.4270283 1.5383551 0 1
## 38       32 0.8039669 1.4860010 0 1
## 39        2 0.6380390 1.4819496 0 1
## 40       25 0.6459446 1.4002422 0 1
## 41       34 0.4767368 1.2863731 0 1
## 42        7 0.7827100 1.2113747 0 1
## 43       27 0.9215764 1.1818669 0 1
## 44       17 0.4817512 0.8962571 0 1
## 45       11 0.4368183 0.7780760 0 1
```

Correlation analysis

I NEED TO WORK HERE

Low correlation between variables, then we use independent modeling analysis. In other words, due the response variables are not correlated the recommended is to use a regression model for each variable, i.e. four regression problems

```
prof %>% ggpairs()
```




```
cor(prof)
```

```
##           CN        CH        LC        MT
## CN 1.0000000 0.6120054 0.3822004 0.5653981
## CH 0.6120054 1.0000000 0.4515429 0.5585100
## LC 0.3822004 0.4515429 1.0000000 0.3882719
## MT 0.5653981 0.5585100 0.3882719 1.0000000
```