

**Cloud Counselage**

Live Project Report

**Test Response Analysis**

Name: Tanya Shrivastava

Project manager: Jayanth GS

Profile: Technical Intern

Domain: Artificial Intelligence

Date of joining: 1 March 2020

Duration of internship: 3 months

***INDEX***

1) About

* Overview
* Rasch Model

2) Requirements

* Software requirement
* Hardware requirement

3) Working with sample data

* Necessary Packages installation
* Reading and Viewing in Data
* Rasch model via TAM
* Running the Rasch model

4) Final output

* Item Difficulties
* Modifying output
* Write R script to xlsx
* Visualization
* Summarizing the distribution of difficulties

5) Conclusion

6) References

About

Overview:

Over the time by understanding from the point of view of examiner or faculty this model is created. The model created helps the front user to identify the difficulty level of each question by assigning the different weights to its only by analysing the students response per question.

The approach chosen is the IRT (item responsive theory) from the trending technology of data analysis. Here we make use of a model called as Rasch model which lies under the IRT.

Rasch Model:

The Rasch model is based on a theory of measurement. Whereas one may typically fine-tune a model to fit the data, in the Rasch paradigm, one compares the data to the Rasch model. Under this view, when the data does not fit the Rasch model, it is believed that the data may not be suitable for measurement.

Sometimes it is said that Rasch is difficult or unrealistic to work with because of its assumptions about the underlying data structure. However, these are not assumptions like the assumptions of ordinary least squares (OLS or linear regression). Instead, these “assumptions” - that the data fit the Rasch model - are the very things we are interested in testing to see if our data is suitable for measurement. If we deem that it is, we may proceed to use the results. If we deem that it is not, all is not lost. We can take that information to alter our items, theory, or model.

Requirements

1) Software Requirements

* R :  <http://cran.stat.ucla.edu/>
* R-studio :  <https://www.rstudio.com/products/rstudio/download/#download>

2) Hardware Requirements

* Laptop or Desktop

Working on sample dataset

1) Necessary Packages installation

* For working directory and file paths

1. Install.packages(“tidyverse”)
2. Install.packages(“here”)

Loading the above packages

* For running Rasch model

1. Install.packages(“TAM”)

( Refer point 3 for description of TAM)

2) Reading and Viewing in Data

* The data for this session will be loaded from provided Test data(AI-DataTest.csv). We need to read it in to your R session. This means that it we need to effectively “import it” into R as something that you can now work with. The .csv file will be read in as something called a data frame or (dataframe).

D <- read.csv("AI-DataTest.csv")

* For view the dataframe

str(D)

View(D)

Loading the TAM package

3) Rasch model via TAM

Running the Rasch model via TAM estimates the model:

*Pr (Xi=1|θs,δi*) = exp(θs−δi)

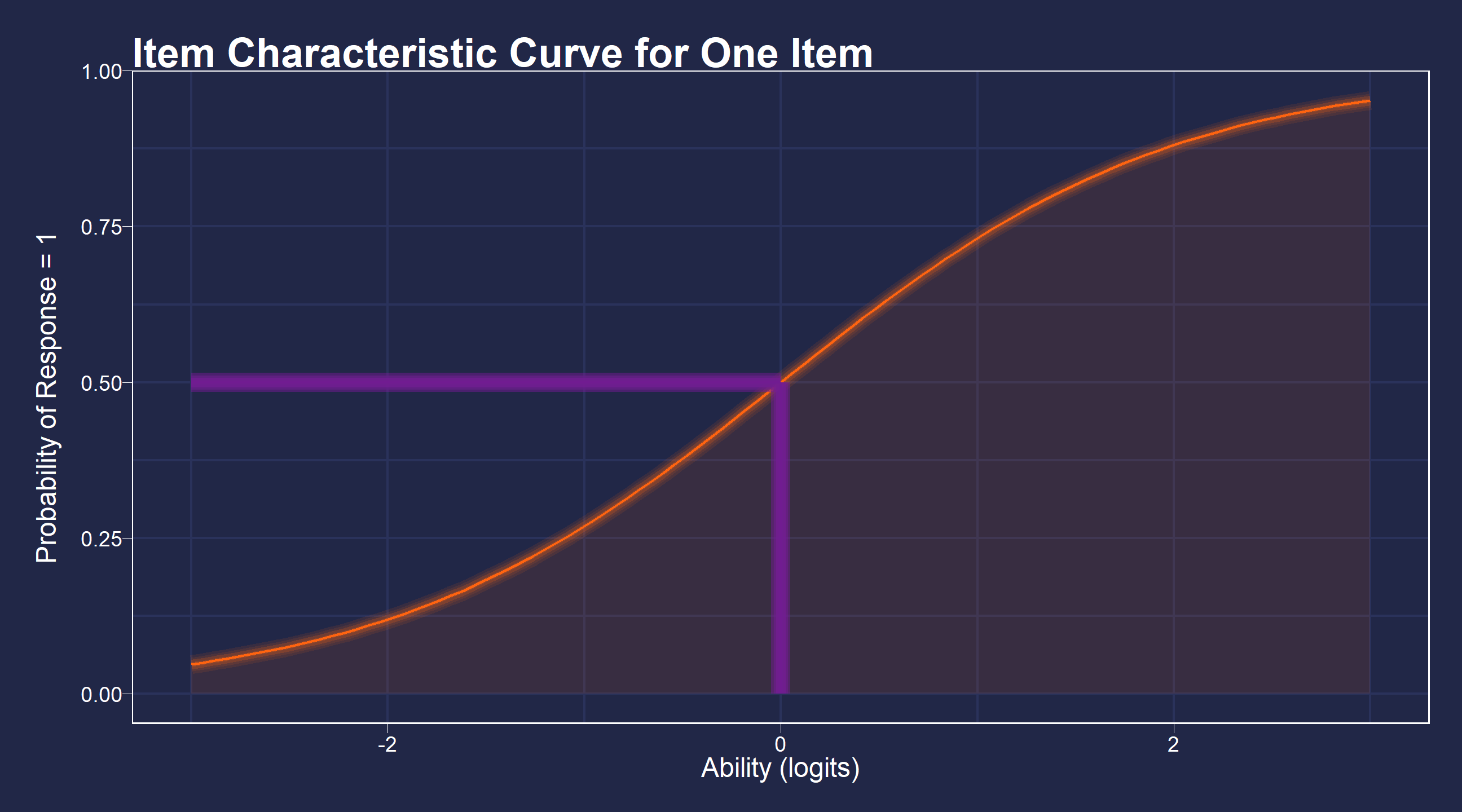
1+exp(θs−δi)

Here, *θs* denotes the estimated ability level of student s, *δi* is the estimated difficulty level of item i and both estimates are in logits. Pr(X=1|θs,δi) can be read as the probability of a “correct response” or of a respondent endorsing the “higher” category (if the item is scored dichotomously) for a item i given a student’s ability and item i's difficulty.

TAM will provide estimates for item difficulty and student ability along with a host of other data.

Item difficulties are defined as the point at which a person has a 50% chance of getting an item correct, defined in logits (log of the odds). So, if for an item a person of ability 0 logits has a 50% chance of getting a item correct, that item’s difficulty is defined as 0 logits.

See the figure below for a visualization of this.



4) Running the Rasch model

We now run the Rasch model on the selected dataframe. Here, we store the estimates of the parameters described above in x. X will also contain other information such as model fit criteria, descriptive statistics, and even information about how long it took for the model to converge. It’s essentially a large list.

This is the main computation step, now we just select information that is stored in x or run x through further computation.

Note that the object D has to contain only items and no other information

X <- tam(D)

summary(X)

Output for summary:

|  |
| --- |
| ------------------------------------------------------------  TAM 3.5-19 (2020-05-05 22:45:39)  R version 4.0.2 (2020-06-22) x86\_64, mingw32 | nodename=LAPTOP-4G646999 | login=Tanya Shrivastava  Date of Analysis: 2020-07-30 19:21:33  Time difference of 0.0588429 secs  Computation time: 0.0588429  Multidimensional Item Response Model in TAM  IRT Model: 1PL  Call:  tam.mml(resp = resp)  ------------------------------------------------------------  Number of iterations = 8  Numeric integration with 21 integration points  Deviance = 1365.2  Log likelihood = -682.6  Number of persons = 50  Number of persons used = 50  Number of items = 25  Number of estimated parameters = 26  Item threshold parameters = 25  Item slope parameters = 0  Regression parameters = 0  Variance/covariance parameters = 1  AIC = 1417 | penalty=52 | AIC=-2\*LL + 2\*p  AIC3 = 1443 | penalty=78 | AIC3=-2\*LL + 3\*p  BIC = 1467 | penalty=101.71 | BIC=-2\*LL + log(n)\*p  aBIC = 1383 | penalty=18.02 | aBIC=-2\*LL + log((n-2)/24)\*p (adjusted BIC)  CAIC = 1493 | penalty=127.71 | CAIC=-2\*LL + [log(n)+1]\*p (consistent AIC)  AICc = 1478 | penalty=113.04 | AICc=-2\*LL + 2\*p + 2\*p\*(p+1)/(n-p-1) (bias corrected AIC)  GHP = 0.56688 | GHP=( -LL + p ) / (#Persons \* #Items) (Gilula-Haberman log penalty)  ------------------------------------------------------------  EAP Reliability  [1] 0  ------------------------------------------------------------  Covariances and Variances  [,1]  [1,] 0.001  ------------------------------------------------------------  Correlations and Standard Deviations (in the diagonal)  [,1]  [1,] 0.032  ------------------------------------------------------------  Regression Coefficients  [,1]  [1,] 0  ------------------------------------------------------------  Item Parameters -A\*Xsi  item N M xsi.item AXsi\_.Cat1 B.Cat1.Dim1  1 Q10 50 0.34 0.663 0.663 1  2 Q11 50 0.30 0.847 0.847 1  3 Q12 50 0.52 -0.080 -0.080 1  4 Q13 50 0.16 1.658 1.658 1  5 Q14 50 0.68 -0.754 -0.754 1  6 Q15 50 0.38 0.490 0.490 1  7 Q16 50 0.74 -1.046 -1.046 1  8 Q17 50 0.34 0.663 0.663 1  9 Q18 50 0.64 -0.575 -0.575 1  10 Q19 50 0.80 -1.386 -1.386 1  11 Q20 50 0.50 0.000 0.000 1  12 Q21 50 0.24 1.153 1.153 1  13 Q22 50 0.62 -0.490 -0.490 1  14 Q23 50 0.36 0.575 0.575 1  15 Q24 50 0.44 0.241 0.241 1  16 Q25 50 0.50 0.000 0.000 1  17 Q26 50 0.54 -0.160 -0.160 1  18 Q27 50 0.24 1.153 1.153 1  19 Q28 50 0.42 0.323 0.323 1  20 Q29 50 0.54 -0.160 -0.160 1  21 Q30 50 0.60 -0.405 -0.405 1  22 Q31 50 0.28 0.944 0.944 1  23 Q32 50 0.70 -0.847 -0.847 1  24 Q33 50 0.76 -1.153 -1.153 1  25 Q34 50 0.64 -0.575 -0.575 1  Item Parameters in IRT parameterization  Item alpha beta  1 Q10 1 0.663  2 Q11 1 0.847  3 Q12 1 -0.080  4 Q13 1 1.658  5 Q14 1 -0.754  6 Q15 1 0.490  7 Q16 1 -1.046  8 Q17 1 0.663  9 Q18 1 -0.575  10 Q19 1 -1.386  11 Q20 1 0.000  12 Q21 1 1.153  13 Q22 1 -0.490  14 Q23 1 0.575  15 Q24 1 0.241  16 Q25 1 0.000  17 Q26 1 -0.160  18 Q27 1 1.153  19 Q28 1 0.323  20 Q29 1 -0.160  21 Q30 1 -0.405  22 Q31 1 0.944  23 Q32 1 -0.847  24 Q33 1 -1.153  25 Q34 1 -0.575 |

Output

1) Item Difficulties

We’ll extract difficulties (xsi) from the x object (x is like a large list). We’ll access this via indexing. The $ sign means, access x and extract the object xsi which exists in x.

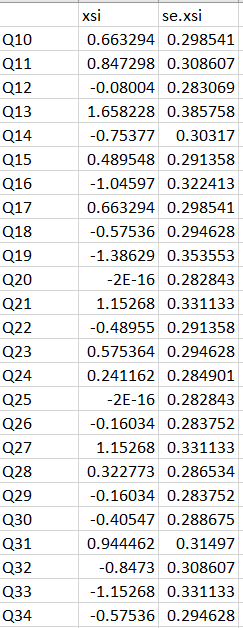
Assign those values to an object in the environment called diffic using <-,

the assignment operator, like before.

diffic <- x$xsi

In the table below, we can see the item difficulties in logits in the column xsi and the standard error for each item se.xsi. One way to think of what the standard error tells us is whether item difficulties may overlap or not.

Higher xsi values indicate more difficult items. For instance, item Q13 is harder than Q14. The values are identified by constraining the mean of item difficulties to zero.



2) Modifying output

For the clear understanding we remane the column name and remove the standard column for knowing accurate item value.

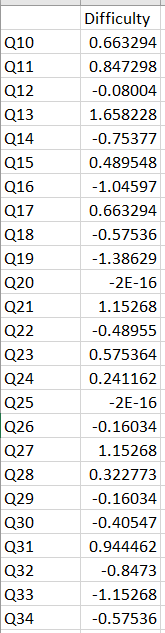
colnames(diffic,do.NULL = FALSE)

colnames(diffic) <- c("Difficulty","x")

df=subset(diffic,select = -c(x))

View(df)

Finally we get output as below table



3) Write R script to xlsx

This gives the output as a xlsx file in the specified working directory

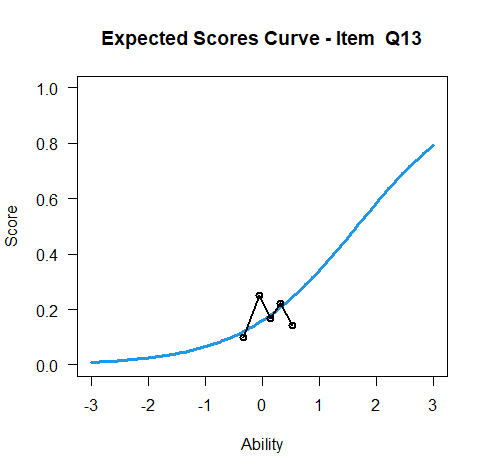
write.xlsx2(df, file="Testoutput.xlsx", sheetName = "result",

append = FALSE)

4) Visualization

We may want to visualize each item characteristic curve (ICC) for each item. These plots plot the expected value (blue, smooth line) given that the data fits the Rasch model, and the observed black line (a binned solution). Each plot represents a single item. They visualize the probability of a respondent getting the item correct given their ability level. For instance, for item Q13, the blue line shows that a person at 1 logit (x-axis) has something like a 95% probability of getting the item correct (predicted). Get Item Characteristic Curves

plot(x)

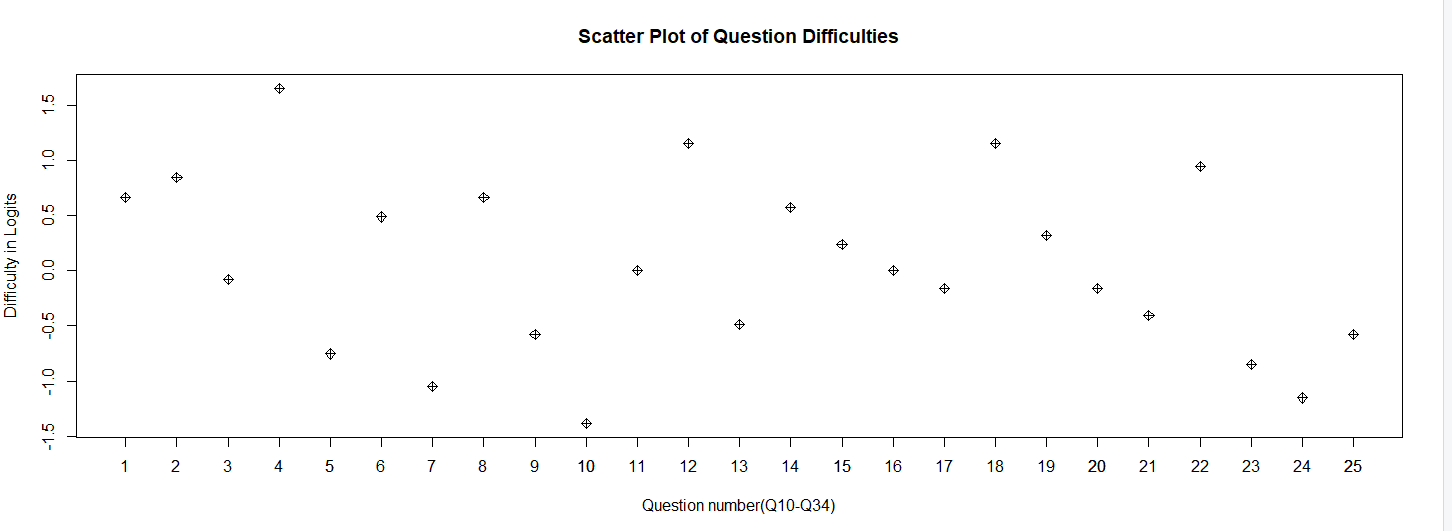


5) Summarizing the distribution of difficulties

We can visualize and summarize the distribution of item difficulties by seeing the items in scatter plot

plot(diffic$Difficulty, main="Scatter Plot of Question Difficulties", xlab="Question Number", ylab = "Difficulty in Logits", pch=9)

axis(side=1, at = c(1:25))



6) Min and Max

Min allows to get the value of item with low difficulty

min(diffic$Difficulty)

Max allows to get the value of item with greater difficulty

max(diffic$Difficulty)

Conclusion

Thus the model helps in analysing the difficulty level per question by assigning unique weight to each question(item).

Also this approach is used in Educational field for the easier understanding to question level.

References

Bond, Trevor G, and Christine Fox. 2015. *Applying the Rasch Model : Fundamental Measurement in the Human Sciences*. Mahwah, N.J.: L. Erlbaum.

Wilson, Mark. 2005. *Constructing Measures : An Item Response Modeling Approach*. Mahwah, NJ: Lawrence Erlbaum Associates.