lanalytics: a software for analysing online learning quizzes.

Salvador Garcia Gonzalez

Master of Science

Data Science

School of Informatics

University of Edinburgh

2017

Abstract

One of the methods for the measurement of students knowledge and comprehension of a topic is through low-stake high-frequency quizzes. These can provide constant information about the group performance and about the general group understanding about specific concepts and topics. This analysis requires processing the quizzes constantly in order to detect and understand the student learning. For this reason, the lanalytics package is introduced as a tool to administer and analyze quizzes in different grouping levels: per person, per quiz and per group. In addition, the lanalytics dashboard is created as a user interface for the package. Together, they can inform the group necessities to the instructors in order to allow them to adapt their tests difficulties according to these needs.

Acknowledgements

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Salvador Garcia Gonzalez*)

Table of Contents

Chapter 1

Introduction

In the psychometrics and educational literature different ideas for the evaluation of students have been developed. The variety of ideas is diverse, from models that accurately measure the student knowledge and proficiency in a topic to models that help in the quiz and item design. Although the ideas for these models typically arise from social sciences questions and assumptions, they are backed by statistical and mathematical tools that help to measure and interpret the results. This way, with the collaboration from both fields, a more detailed assessment and interpretation of the results can be provided for the students and instructors.

One idea for student assessment is through high-stake and low-stake tests. The basic idea is to apply one or more tests with different frequency level with the purpose to evaluate the knowledge and proficiency of the student for a topic or a set of topics. The frequency of the tests plays an important role in this text, in special the low-stake high-frequency quizzes. Frequent quizzes allow monitoring the student performance during the course, permitting the instructor to take actions based on the obtained information. On the other hand, in the high-stake quizzes, the instructor can only assess the knowledge of the students at the end of the course or when the high stake test is taken.

In this dissertation, some difficulties in the analysis of the low-stake high-frequency tests will be covered. Because the number of tests and the generated information increases, the time to evaluate each item and test difficulty is reduced, so a tool to quickly calculate them is required. Additionally, a tool to manage and constantly analyse the answers from the students is required, both to assess the required time for each quiz and to measure the performance of the group and particular students. All of these having as an objective that the instructor could quickly respond to the needs of the students in the group [18] [14].

For the first point, the estimation of the item difficulty parameter, the use of the Item Response Theory will be proposed, in special of the Rasch models. These models allow to calculate a difficulty parameter for each item and also to provide the student's ability parameter. The most interesting part of this model is that it makes comparable both sets of parameters. This information can potentially benefit the class instructors because they will have a tool to prepare the tests difficult according to the student's abilities. Furthermore, the instructors can perform Just in Time Teaching and adapt their tests difficulties and necessities according to the specific group requirements. Currently, these models are implemented in R packages [1].

For the second point, the management and constant analysis of student answers and test times, a new package is proposed. The main objective of this package is to ease the test analysis and make it feasible for instructors. This package is called *lanalytics*, and its main objective is to administer and analyse quizzes in different grouping levels: per person, per quiz and per group. To include the idea of quizzes administration, the package can create a quiz object and a course object. Then the instructors can export all the data from the quizzes as an R data file or *.csv file. Then, if further analysis is required, they simply have to read the data files to visualise it and make an analysis. The package includes different visualisation tools that work with these quiz and course objects.

To join this two ideas, the management and analysis of the test's information and the estimation of the item difficulty and student abilities, a user interface is proposed. This interface is named *The lanalytics dashboard* and is implemented in Shiny R, that is completely free and open source. As it can be hosted on a web page, it can allow that users that are not familiar with R can have access to the lanalytics package and some functions of the eRm package. As building a comprehensive software that makes analyses data from a wide spectrum of sources is a very long task, this will be just a prototype for further future developments.

In the second chapter of this dissertation, the background for Just in Time Teaching (JiTT) and Item Response Theory (IRT) will be given. In specific, the Rasch models and the plots that will be used in the dashboard will be covered. Later, in the third chapter, the lanalytics package will be introduced, and the motivation for each of the generated plots will be explained. In the fourth chapter, the Shiny interface prototype will be presented, and each component will be explained. In the fifth chapter, a user experience evaluation of the Shiny interface will be documented. Finally, the conclusions

---

[1]For example in the eRm and the ltm packages.

and future work will be provided in the sixth chapter.

All this project is stored in Github. The URL of both the package and the dashboard is: `https://github.com/savrgg/lanalytics`

Chapter 2

Background

The item design is a topic in the assessment process of tests and quizzes for educational courses. A division of these tests can be introduced according to the quiz weight in the final grade and on its periodicity. This way, the high-stake quizzes are tests with low periodicity and a huge reward or penalty for the persons that take it [10]. On the other hand, a more frequent version of these tests are usually under a low-stake idea [1], and then it is allowed to constantly measure the current status and progress of the students. Moreover, through analysing these quizzes, the instructor can have the opportunity to improve the quality of the items and to early detect anomalies in the current group. By analysing the low-stake quizzes, the instructor can find useful information about the current group. For example, if the group misinterpret some core concept for the course, then the instructor can spend more time. Additionally, another factor that can be important is the questions formulations. If some anomaly is detected in the analysis of the answers and the answering time, then the instructor can review the issue in this item. All of these analyses are important because every course has a limited class time, so it is a good idea to spend it on topics or applications that have been problematic for the group. This idea of adapt the class with every quiz application is called Just In Time Teaching (JiTT) [17] and will be explored later.

In the first section of this chapter a brief introduction of the Just in Time Teaching is presented, then a popular method to analyse item-based quizzes is discussed: the Rasch model. This is a model that helps the instructors to find a numerical representation of the difficulty of an item. Also, it finds a numerical representation of the individual ability of every student in the group [4]. The interesting point about this method is that it constructs an ability value per student that is comparable with the difficulty value of

---

[1]between more quizzes are made, then these are ponderated less in the final grade

the item. This way the instructor can determine the level of the quiz not only based on the difficulty of the items, but also on the particular abilities of the current students. This model is explored in the second section of this chapter.

2.1 Just in Time Teaching

When using low stake-frequent quizzes, the instructor can be aware of the current general performance of the group to make more emphasis in conflictive concepts and applications. Furthermore, the instructor can prepare the contents of the next class based on the extracted information from the analysis. For example, if the next class material requires the understanding of some core concepts, and the group is not experienced with them, then the professor can take some time to explain them. This class adaptation of the class according to the detected needings is called Just in Time Teaching [17].

2.2 Item response theory (IRT)

2.2.1 Latent and observable variables

The difference between an observable and a no observable variable relies on the capacity of being directly measured or recorded [8]. The no observable variables are commonly called latent or hidden variables. In the models that have hidden parameters, the inference is not performed as usual, but it needs to be done through information from the observable variables. Examples of latent variables in social sciences are the public opinion, the social class, and the verbal ability [8]. Neither of these variables is directly observed, but are inferred from others quantitative or qualitative variables. For example, the verbal ability can be inferred from tests that involve characteristics of what we think creates the verbal ability.

2.2.2 Latent Variable Models

In general, many statistical models use latent variables in its definitions, for example, the Gaussian mixture models [1] in which it is assumed that each observation corresponds to one class, but this class is unobserved (usually a subpopulation inside a population). Then the joint probability distribution of the observed variables and this

unique hidden variable can be specified as [1]:

$$p(x_o, x_h) = p(x_o|x_h)p(x_h) \tag{2.1}$$

With x_o the set of observable variables and x_h the set of hidden variables. In the above equation, the conditional probability given some specific class (hidden variable) $x_h = k$ is normally distributed. Then the model estimates the probability of belonging given some specific class k as follows:

$$p(x_o|x_h = k) = N(\mu_k, \sigma_k) \tag{2.2}$$

Another popular class of methods is the Hidden Markov Models [1], which are quite similar to the normal Markov Models, but the transition probability is now between the hidden variables instead of the visible ones. Then another probability to link the observable and hidden variables is defined and is known as the emission probability [2]:

$$p(x_o, x_h) = p(x_{o_1}|x_{h_1})p(x_{h_1}) \prod_{i=2}^{n} p(x_{o_i}|x_{h_i})p(x_{h_i}|x_{h_{i-1}}) \tag{2.3}$$

Besides these models that are directly defined with latent variables, there are others that can be expressed with a latent variable structure, such as the commonly used Singular Value Decomposition [9]. As can be observed, one common aspect of the latent variable models is that the joint probabilities can be represented in terms of a conditional distribution of the observed variables given the hidden variables. Then the observed variables are conditionally independent given the hidden variables [8]:

$$p(x_o|x_h) = p(x_{o_1}|x_h) * p(x_{o_2}|x_h) * ... * p(x_{o_n}|x_h) \tag{2.4}$$

Furthermore, these latent variable models can be classified according to the observable and latent variables (continuous or discrete). When the latent variable is considered a continuous variable, we can consider two well-known families of models [3] [8]:

- Factor analysis models (continuous observed variable)

- Item response theory models (discrete observed variable)

---

[2]In this example, there is one hidden variable per visible variable and are assumed to be discrete.
[3]although there are more families of models, but these two are closely related to the topic

In the *Factor Analysis Models* the observed variable is continuous. Then the latent variables are used to explain the variability of the observed variables. Commonly the number of latent variables is less than the number of observed variables. Formally, the factor analysis can be described as [25]:

$$x_o = Ax_h + \mu + \varepsilon \tag{2.5}$$

where:

- $x_o \in \mathbb{R}^n$ is the vector of observable variables

- $x_h \in \mathbb{R}^m$ is the vector of hidden variables and is assumed to be normally distributed $x_h \sim N(0, I_m)$

- $\mu \in \mathbb{R}^n$ is a constant vector that represents the mean of the data

- $\varepsilon \in \mathbb{R}^n$ is called the noise term. It is normally distributed $\varepsilon \sim N(0, \Phi)$ with Φ diagonal

- $A \in \mathbb{R}^{n \times m}$ is called the factor loading matrix.

In this model is important to remark that the observed variables are conditionally independent given the hidden variables in x_h:

$$p(x_{o_1}, ..., x_{o_n} | x_h) = p(x_{o_1} | x_h) * p(x_{o_2} | x_h) * ... * p(x_{o_n} | x_h) \tag{2.6}$$

In the *Item Response Theory (IRT) models* the observed variables are discrete. This is useful because we can use it in models where the observed variables are inherently ordinal or categorical, for example in multiple choice questions with more than one correct answer (polytomous variables), or questions that can be scored as correct or incorrect (dichotomous or binary variables).

In the IRT models the hidden variable is continuous and is called a *latent trait* and it usually represents the ability of the test takers. One of the most common models is the Rasch model that will be introduced in the next section.

2.3 The Rasch model (RM)

2.3.1 The sigmoid function

To explain the Rasch model, we are going to introduce the sigmoid function (which receive the name because of the *S* shaped form). This function has the equation:

$$sigmoid(x) = \frac{1}{1 + \exp(-x)} \tag{2.7}$$

The domain of this function is $(-\infty, \infty)$, and as can be shown, the limit to $-\infty$ is 0 and the limit to ∞ is 1. In consequence, this expression is always bounded between (0,1), as showed in the fig. 2.1. This is very similar to the range of probability. [4]. Because of the last property, this curve is commonly used to model probabilities regarding some independent variable $\mathbf{x}$ [5]. For example, in the logistic regression, the curve is modelled in terms of $Z = B^T X$ where B is a vector of parameters, and X is the matrix that contains the regressors.
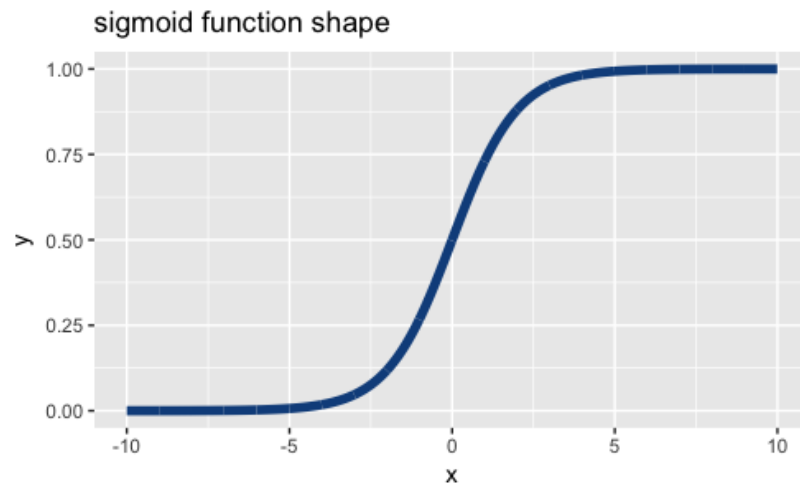
Figure 2.1: Sigmoid function. The domain of this function is $(-\infty, \infty)$ and the range $(0, 1)$

Now, for the Rasch models, we need to analyse further the properties of the sigmoid function. If we subtract some positive scalar $\mathbf{a}$ to $\mathbf{x}$ in the equation 2.7 then it result that the curve in fig. 2.2 moves to the right, and the value of the sigmoid function decrease. Equivalently, if we add the same positive scalar $\mathbf{a}$, then the curve moves to the left.

This idea is close to another formulation of the sigmoid function, which now can have two variables instead of one $(\mathbf{x,a})$. With this modification, the domain and the range are the same. As mentioned before, the sigmoid function can be used to model a probability, that is, a continuous variable in the range [0,1], so then it is equivalent to a regression problem [6] [9]. Now in terms of a classification problem (binary or

---

[4]Except that the probability can take the values of 0 and 1
[5]For now, let's assume that $\mathbf{x}$ is one dimensional
[6]More generally of a generalized linear model
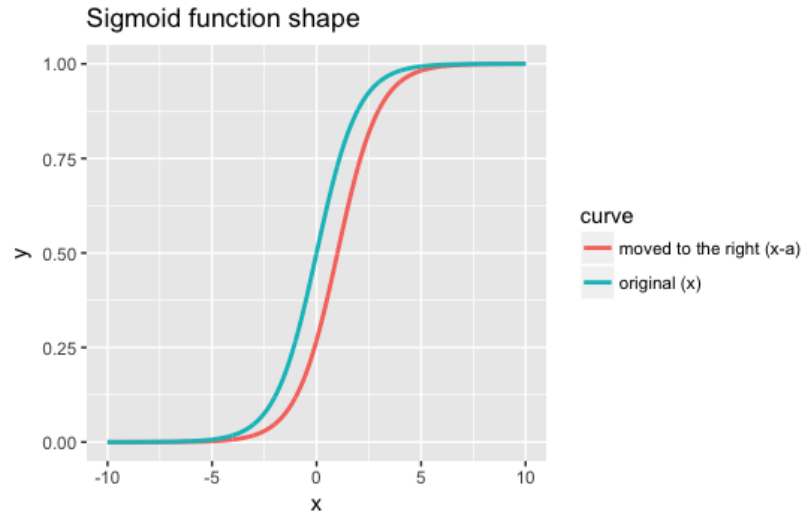
Figure 2.2: Effect of the variable **a** in the sigmoid function.

dichotomous variable), this can be interpreted as the probability to belong to one of the two classes [7]. Then, the expression can be expressed as:

$$P(y = 1 | x, a) = \frac{1}{1 + \exp(-(x-a))}$$

$$= \frac{\exp(x-a)}{1 + \exp(x-a)}$$

(2.8)

Where y is the binary variable. This basic idea is what support the basics of the Rasch model for dichotomous data.

2.3.2 Rasch model

Now, given this background, we can proceed with the definition of the Rasch model. From the last section, we can remember that the IRT models have a continuous latent variable (latent trait or ability) and an observable set of variables. In the next example, let's assume that this variable is binary and that the sigmoid function represents the probability of belonging to one of these classes.

$$P(A_{rc} = 1 | \theta_r, \beta_c) = \frac{\exp(\theta_r - \beta_c)}{1 + \exp(\theta_r - \beta_c)}$$

(2.9)

Where:

- *A* is a matrix where the rows represent the persons and the columns the question of the quiz.

---

[7]Further generalizations for the polytomous case can be done.

- $A_{rc} \in \{0,1\}$ is the entry (r,c) of the matrix A. This entry is a binary variable that represents the answer to the item c of the person r. 0 means that the person answers incorrectly and 1 that answers correctly.

- $\theta_r \in (-\infty, \infty)$ is the ability or trait of the person r.

- $\beta_c \in [0, \infty)$ is the difficulty of the item c.

Now, interpreting the result of the last subsection, if the parameter β_c (difficulty) increases, then the sigmoid curve moves to the right leading to a decrease in the probability of getting correct the question. As in many statistical models, the problems rely on the parameter inference. For this reason, the author defines two auxiliary variables named *Raw Scores*. Each of these scores is the marginalization of A_{rc} with respect to the variables r and c. Then each raw score depends only on one variable:

- **Raw score per person:** $r_r = \sum_c A_{rc}$

- **Raw score per item:** $s_c = \sum_r A_{rc}$

The raw score per person is simply the sum of all the correct questions answered by each person; then it is a measure of the ability. On the other hand, the raw score per item is the sum of all persons that get correct that item, then it can be interpreted as the easiness of the item.

2.3.3 The model assumptions

According to [6] [13] the Rasch model has the following assumptions:

- **1) Unidimensionality:** The unidimensionality assumption is related to the dimensionality of the hidden parameters of the model. For the Rasch model, the ability parameter $\theta_r \in \mathbb{R}$ and the difficulty parameter $\beta_c \in \mathbb{R}$. Other models deal with multidimensional difficulty and ability parameters, but these will be not documented here.

- **2) Conditional independence:** The conditional independence is the same required for the latent variable models. This way, the observed item binary variables are independent among them given the parameters θ_r, β_c

- **3) Sufficiency of the raw score r_r:** It is similar to the sufficiency in statistics. A statistics is sufficient if no other statistic provide some other information about the parameter with the sample.

- **4) Monotonicity in the probability concerning the parameter** θ_r: That is, when having more ability, the probability of getting correct an item increase.

Due to the purpose of the text, the details about the tests for the assumptions will not be explained here. For more details about these tests, please consult [6].

2.3.4 Item-parameter estimation

In this section, a general review on the parameter estimation will be given. The basics of the item parameter estimation are based on the Maximum Likelihood Estimator, but according to [13] there are three ways to estimate the parameters:

Joint Maximum likelihood Estimation (JML)

The joint likehood formula of θ_r and β_c is [26] [13]:

$$
\begin{aligned}
L_{joint} &= \frac{\exp(\theta_r - \beta_c)^{\Sigma_c \Sigma_r a_{rc}}}{\prod_r \prod_i (1 + \exp(\theta_r - \beta_c))} \\[2ex]
&= \frac{\exp(\Sigma_c \Sigma_r a_{rc} \theta_r) \exp(-\Sigma_c \Sigma_v a_{rc} \beta_c))}{\prod_r \prod_c (1 + \exp(\theta_r - \beta_c))} \\[2ex]
&= \frac{\exp(\Sigma_r \theta_r r_r) \exp(-\Sigma_c \beta_c s_c)}{\prod_r \prod_c (1 + \exp(\theta_r - \beta_c))}
\end{aligned}
\tag{2.10}
$$

with $r_r = \sum_c A_{rc}$ and $s_c = \sum_r A_{rc}$. As reported in [11], these estimation for the item parameters are not consistent when the number of the sample increases (there is one parameter per person, so it can increase too much).

Marginal Maximum likelihood Estimation (MML)

If we integrate the person parameter, the joint distribution can be marginalised, and the parameters can be estimated via the Expected-Maximization algorithm [3] [13]:

$$
L_{marg} = \prod_r \left[\exp\left(-\sum_c \beta_c s_c\right) \int \frac{\exp(\theta_r)}{\prod_c^k (1 + exp(\theta - \beta_c))} dG(\theta) \right]^{n_r}
\tag{2.11}
$$

Where it is assumed that the person parameter follows a predefined distribution. The expected maximisation algorithm is an algorithm that helps to maximise the likelihood of difficult problems. The idea is to include latent variables to make easier

the maximisation process. For more information, this method is well documented in different texts [9] [1].

Conditional Maximum likelihood Estimation (CML)

The third idea is to condition the joint likelihood on r_r [13] (that is, conditioned on a sufficient statistic):

$$L_{cond} = \frac{\exp(-\sum_r \beta_c s_c)}{\prod_r \sum_{x|r}(-\exp(\sum_r x_r \beta_c)} \tag{2.12}$$

This way the parameters of the persons do not appear in the conditioning formula.

2.3.5 The Item Characteristic Curve (ICC)

Now the question is how to plot the Rasch model. One way to do it is called the Item Characteristic Curve where we can display one item per curve.

We already have the item and the person parameter, so we can calculate the probability of answering correctly one specific item for one specific person. If we make a plot where in the y-axis is the probability of getting correct this particular item and in the x-axis the person parameter and put one point per person. Then, the parameters of a sigmoid function are adjusted minimising the residuals between the individual points and this sigmoid function. This way we can have an Item Characteristic Curve as a representation of the model expectation [6]:

The plot fig. 2.3 plot is useful because we do not have to analyse all the abilities parameter of the persons, just one function that was fitted with all of these points. This way, we can easily identify which item was the most difficult. It is easy to identify that the leftmost curve is the easiest question and the rightmost curve the most difficult.
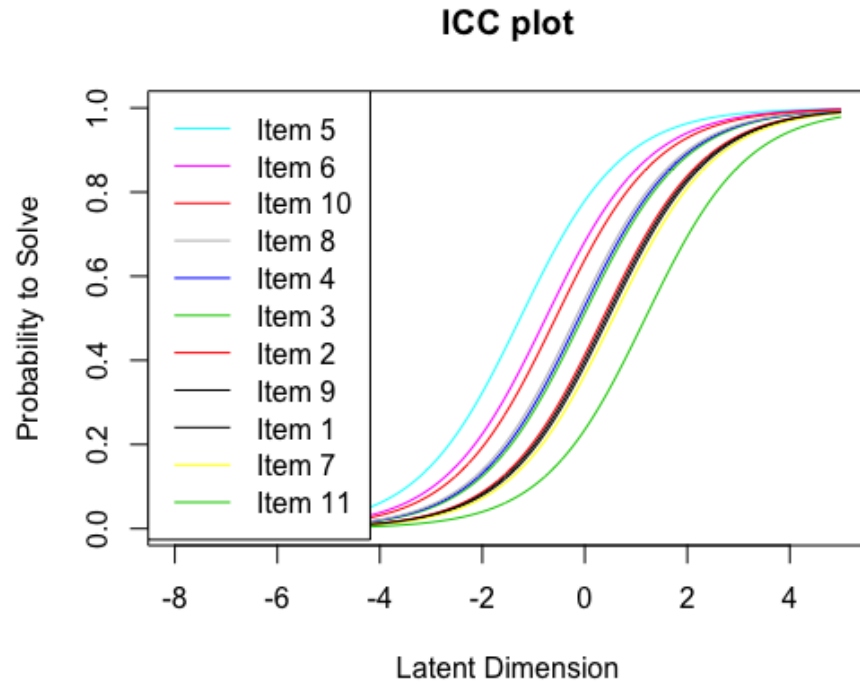
Figure 2.3: Item Characteristic Curve. Each curve represent one item.

2.3.6 The Person-Parameter plot (PPP)

From the inference, we know the exact value of each person parameter, but now we have to answer how does this parameter be related to an observable variable. To answer this question, the Person-parameter plot maps the relation between the values of the ability parameter versus the number of correct answers of each person. This way we can have an idea of the relationship between the latent ability variable and the observable raw scores [4]:

In the fig. 2.4 we can see the raw scores and the person parameter from a synthetic dataset [8]. In this synthetic dataset, there were 11 questions, but the maximum raw score was 10. In the graph, we can observe that this relation is monotone. The idea is that between the number of correct answers increases, the student is more skilled. [4].

2.3.7 The Person-Item Map (PIM)

Now, to show the relationship between the person parameters and the item parameters, the Person-Item map was created. In the uppermost part of this plot, it is shown the

---

[8]the creation of this synthetic dataset are explained in chapter 4

Plot of the Person Parameters
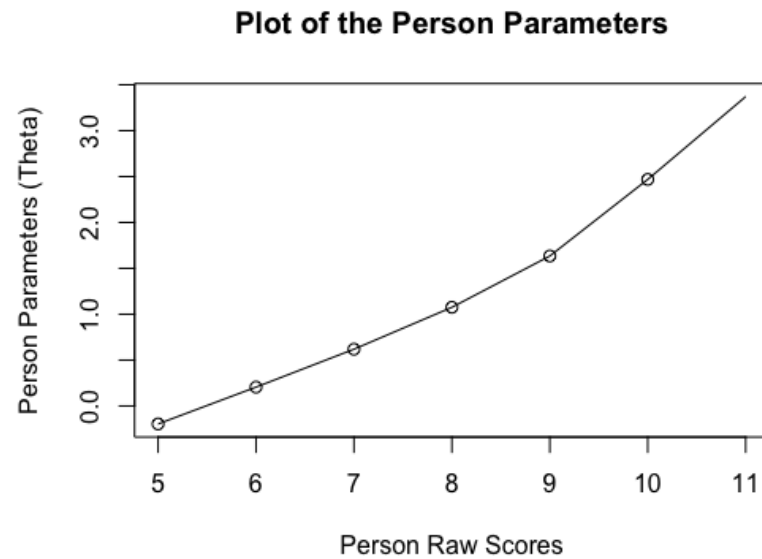
Figure 2.4: Rash model: person parameters. This plot shows the monotone relationship between the raw score per person and the value of the person parameter.

Table 2.1: Raw score per person and the person parameter.

Person raw scores	Estimate person parameter	Std. Error
5	-0.1963	0.6355
6	0.2050	0.6347
7	0.6182	0.6546
8	1.0750	0.7028
9	1.6339	0.8043
10	2.4685	1.0662

histogram of the person parameters regarding the latent dimension. We can see that the mode of this histogram was in the values 1.6339 and 1.0750 that correspond to eight and nine correct answers (see table 2.1). Also in the central part of the plot, we can observe the item parameters values (also regarding the same latent dimension).

This plot is interesting and useful because we can determine if the difficulty of the topics were lower for the ability of the students. In this synthetic example and the Person-Item map (fig. 2.5), we can see that the difficulty of the topics are on the left side of the plot (most of them are below 0.6182 in the latent dimension), while the ability parameters of the students are in the right zone (most of them are above 0.6182

in the latent dimension). Then, in general terms in this example, the students were very skilled, and their ability was above the item difficulty. This is quite straightforward to deduce from the synthetic data set because to generate the answers, the probability to get right one question was at least 50% for all items and students.
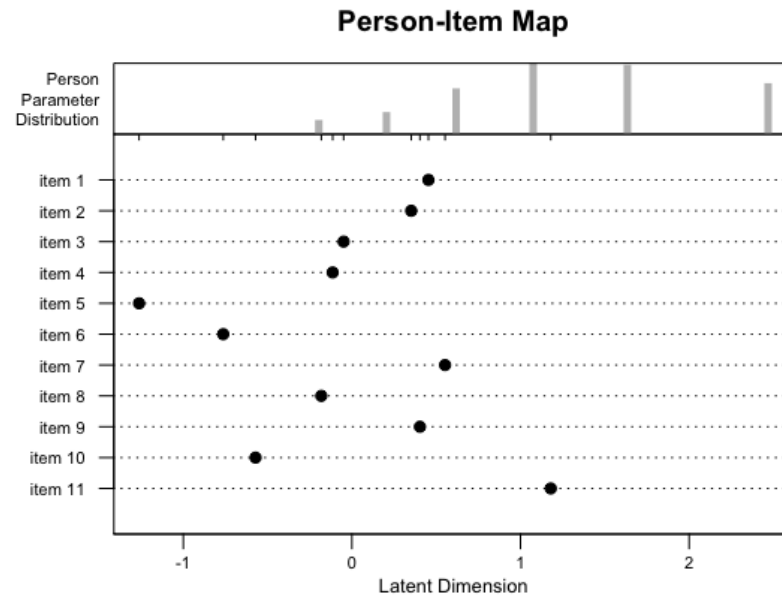

Figure 2.5: Rasch model: Person-Item map

Chapter 3

The lanalytics package

The *lanalytics package*[1] is an R package designed to analyse the answers to online quizzes. The objective of this package is to accomplish one of the main objectives of this text, that was the management and constant analysis of student answers and test times. The package is able to perform different analysis according to the desired grouping level: per quiz, per group and per person. To accomplish this, the data can be plotted in seven different ways, varying on the distinct aggregation levels and the different objectives of the analysis. At the group level, the package contains the *guessing* and the *Time-easiness* plots; at the quiz level, it contains two descriptive statistical graphs, a *boxplot* and a *histogram* of the obtained scores of the online quizzes takers and the *Easiness-Time* and the *Easiness-Time-Level* plots. Finally, at the individual level, the user can plot the grade history of a student. The explanations and objectives of each one of these will be explained throughout this chapter. For all the plots in this chapter, a synthetic dataset was used for illustration purposes; these plots were not generated with any real quiz dataset.

3.1 Design of the package

3.1.1 Initial investigation

Previous work on open source software for analysing quizzes has been done. One of the most popular software is Moodle (Modular Object Oriented Dynamic Learning Environment) that is learning management system [27] [14]. Which allows creating customisable webpages for the courses. With this tool, the instructors can generate

---

[1]The package is open source and it is hosted on Github. `https://github.com/savrgg/lanalytics`

and calculate the grades for their students. Also, it allows the tracking progress of the students.

Some other user interfaces have been proposed to analyse the data from the students, as student engagement, attendance, and interactions with other students. For example, the Desire2Learn software has tools to achieve a personalised learning and also add-ons that allow analysing the engagement and the performance of the students [7]. Other software like learning catalytics was designed to increase student engagement [15]. But, as opposed to Moodle, these two solutions are not freely available.

In the CRAN (Comprehensive R Archive Network) there are some packages that allow the creation and evaluation of exams. One of these tools is the exams2nops function from the *exams* package [27], but according to with the realized research no package was found to manage and analyze quizzes and items of the quizzes in distinct levels that was coded in R. For this reason, the lanalytics package was proposed as a useful tool to produce the required analysis for the instructors.

3.1.2 Cognitive levels

It is important to define the cognitive levels that are used in this package. These levels are based on the levels of the Bloom's taxonomy (remember, understand, apply, analyse, evaluate, create) [2]. The first level that we defined for this analysis include the easiest questions, those that only implies a *factual knowledge*. These are questions that only require memorising and repeat the definition of a concept. The second level is the *understanding of the concept*, where the student should not only know the definition but also to be able to discuss it and explain it. In the last level, *application of concept*, the student should know how to execute and operate the concept in real problems or situations [21].

3.1.3 Quiz and course objects

To allow the instructors to manage the quizzes and course data, the lanalytics package created two exportable objects, the quiz object, and the course object. The basic structure of the lanalytics package is the quiz object which is implemented with a *tibble* [24] [2]. This object contains a tibble in long format with one entry per student-question. For each row, another two variables are required: the score (a binary variable indicating if the answer was correct) and the date-time when the question was answered (in

---

[2]A tibble is data structure build on top of the data.frame class and give it more functionalities.

POSIXct format). All this information should be provided in a **.csv* file.

To allow more interesting analysis, the quiz object can be augmented with other two columns computed by R. The first is the spent time per question and the second is the answering order (a number is assigned to each row according to the order in which it was answered). An example of the quiz object is in the [fig. 3.1].

```
         email.address question           responded.at score
1 ADRXM8548M@gmail.com        1 2017-02-11 22:11:46         1
2 ADRXM8548M@gmail.com        2 2017-02-11 22:12:00         1
3 ADRXM8548M@gmail.com        3 2017-02-11 22:12:16         0
4 ADRXM8548M@gmail.com        4 2017-02-11 22:12:41         1
5 ADRXM8548M@gmail.com        5 2017-02-11 22:13:01         1
6 ADRXM8548M@gmail.com        6 2017-02-11 22:13:49         1
                          quiz order.answer time.per.question
1 datasets/sample_dataset/Q01.csv             1         NA secs
2 datasets/sample_dataset/Q01.csv             2         14 secs
3 datasets/sample_dataset/Q01.csv             3         16 secs
4 datasets/sample_dataset/Q01.csv             4         25 secs
5 datasets/sample_dataset/Q01.csv             5         20 secs
6 datasets/sample_dataset/Q01.csv             6         48 secs
```

Figure 3.1: Quiz object. It has an specified format and columns.

Finally, all the quizzes can be grouped in a course object, which joins them in the same tibble. This is very useful if the instructors want to save the all the records of their course in just one file.

3.2 Group level analysis

The group level analysis can potentially help the instructors to monitor the performance of all the group through all the course. Different interesting aspects of the group are analysed in this section. First, a guessing plot that helps to detect which potentially students that are answering randomly or with some exterior help is presented. After this, an Easiness-Time plot per time-terciles is made, in which the main objective is to understand in general how the group is answering the quiz regarding time. Also, the relation between this invested time and the obtained score in the items is presented. This tab was created to help the instructors to understand all the group behaviour in the quiz (for the guessing plot they can quickly observe the all the anomalous behaviours in the group and for the easiness time per quiz they can observe how the students are answering the questions according to the time).

3.2.1   Guessing plot

The idea of the *guessing plot* [21] [fig. 3.2] is to detect if a student is answering the questions in a very fast pace. Some students can answer the quizzes fast, but even the fastest of them answer the questions above some time threshold. This threshold is relative to the item difficulty and the student's ability, so one way to find a lower bound is to insert a question that every student can answer almost immediately in some section of the quiz and record the time that each student spends on it (For example, an easy a fast question can be: "did you attend this week's lecture?"). If some other question is answered below this time threshold, then it should be analysed. As the fast question answering does not give more information about the student behaviour (the student could be answering randomly or could know the answer from another student), the obtained score should be analysed. In the guessing plot if someone spends less time than the threshold, then a point will appear. If the answer was correct, then the colour of the point will be red. Otherwise, it will be blue.
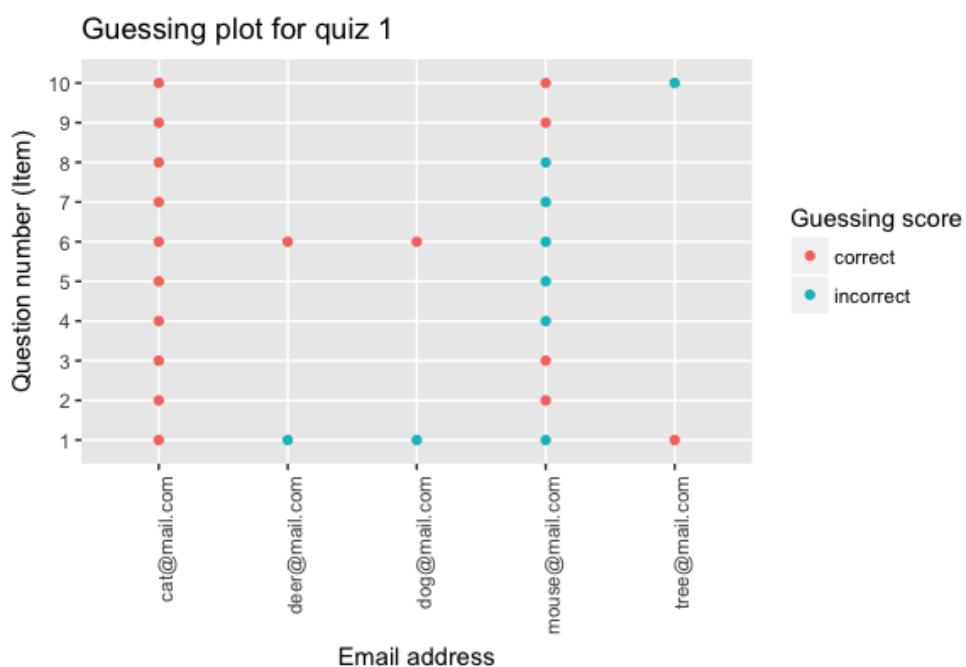
Figure 3.2: Guessing plot. A point indicates that the user answers the item faster than some threshold. If it is red, the answer is correct, otherwise is incorrect.

The intuition of this plot is that, if the student gets many red points in the same quiz, it is possible that the answers to the items were known previous to the quiz. If the student gets many red and blue points, it is possible that the quiz was answered just randomly. The ideal situation is that no or just a few points appear per student

in the graph. For example, in the fig. 3.2 we have 5 users. The users *cat@mail.com* and the user *mouse@mail.com* answered fast all the questions, but the difference is that the user *cat@mail.com* got every question correct while the other user does not. Because of the defined threshold, it is possible that the first student previously knew the answers, while the second one just answers randomly.

3.2.2 Easiness time per quiz and tercils

In the Easiness-Time plot fig. 3.3 the students are grouped by tertiles according to the time that they spent in the question. This way, the first tercile contains the fastest students and the third tercile the slowest. The intuition behind this plot is that, if both the fast and slow answering students score correctly in some question, then it is an easy one. On the other hand, if there is a score gap between the fast and slow students, then it may be interpreted differently. If the fast-answering students get a better result, then it may be interpreted as they know the solution, so they answer it faster. On the other hand, if the slow-answering students have a better result, this item might be a little bit confusing, so the students spend more time trying to figure out the solution.
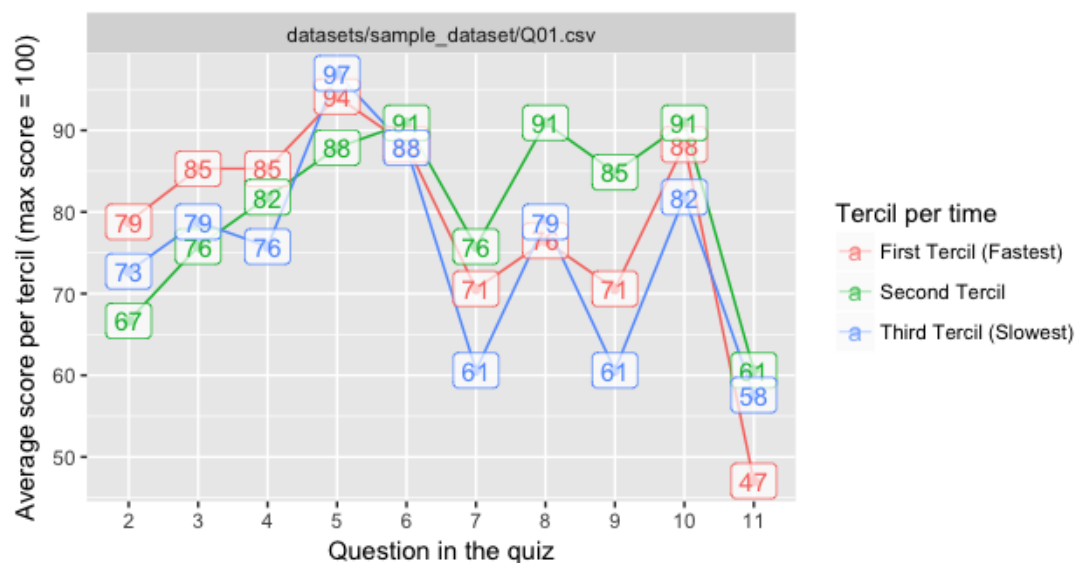

Figure 3.3: Easiness-Time per group and tercil. In the x-axis is the question of the quiz and in the y-axis the average score per time-tercil.

In the fig. 3.3 we can see these patterns. For example, in the questions 5 and 6, all the students got an average of 90. Conversely, in the questions 2,3,4, the faster students got better results than, the slower ones.

3.3 Quiz level analysis

The quiz level analysis can potentially help the instructors to monitor the performance and analyse the data quickly from the perspective of the quiz and items. Different aspects that can be evaluated from this are the easiness of the items and the median spent time in each one of them [3]. First, to give a general idea of the performance of the students in the selected quiz, a histogram and a boxplot of the scores are displayed. This can help to analyse the outliers and the dispersion of the data. After these, two plots analyse the items from a perspective of easiness and time. Conversely to the Easiness-Time plot of the group level, here we are interested in the general performance of the quiz. We would like to observe that difficult items take more time than easy ones, so there is a negative relation between easiness and time. This point can be addressed with the ET-plot [21]. Additionally, we would like to see if this easiness is related with the cognitive level, as intuitively, a higher cognitive level requires more time (For example, to apply a concept, first the student should understand it and remember it).

3.3.1 Histogram of quizzes

To identify the dispersion and skewness of the data we can analyse the histograms per quiz. With these is easy to detect if there are subpopulations in the data (for example if the distribution is similar to a bimodal normal distribution). In the [fig. 3.4] the quiz 01 looks normally distributed, and the quiz 02 looks like a bimodal distribution.
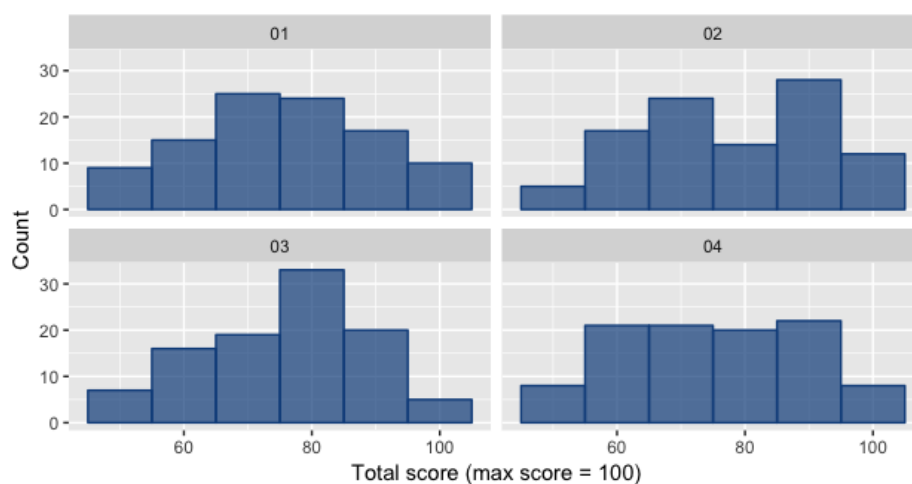

Figure 3.4: Quiz scores histogram. In this plot four quizzes are displayed.

---

[3]It has calculated the median so that the analysis is not affected by outliers

3.3.2 Boxplot of quizzes

Another way to verify the distribution and the skewness of the data is to analyse the boxplots of the scores. A useful aspect of these plots is that they allow us to detect outliers. This is important because the instructor can detect how many students are performing badly or well (depending on what kind of outlier is presented). After that, the instructor can only spend time analysing these points individually and see what topics are confusing for them.

The implemented boxplot in the ggplot2 package [22] is the Tukey boxplots [4], which use the first and third quartile as the hinges and limit the length of the whiskers to 1.5 of the interquartile IQR length. If a point in the data is outside these ranges, then it is classified as an outlier. For the upper whisker, if the value is greater than the maximum value of the data, then the whisker is reduced to this value (An equivalent rule apply for the lower whisker).
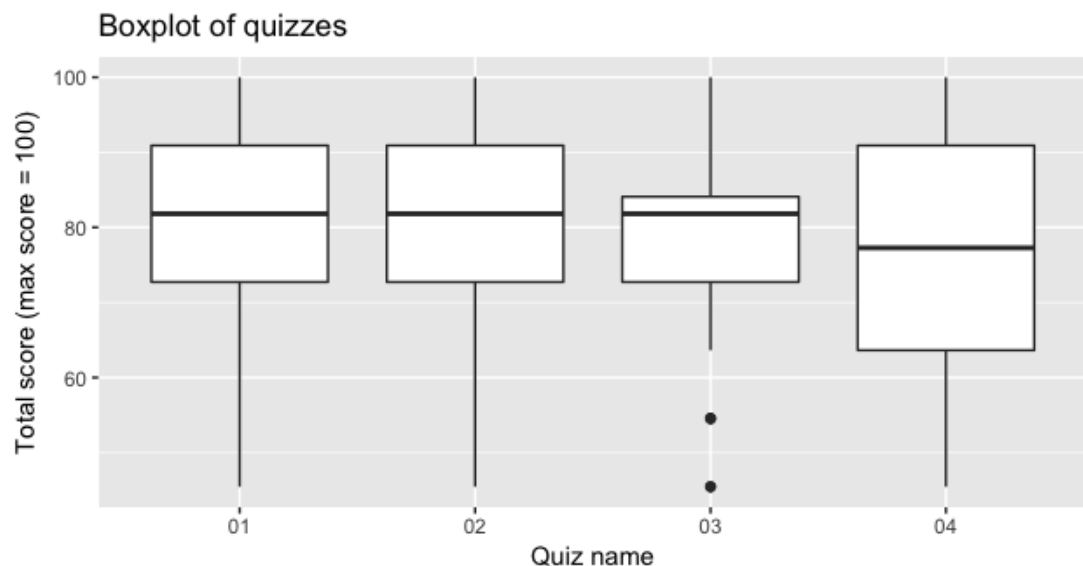
Figure 3.5: Quiz scores boxplot (Also called box and whiskers diagram). This plot display five different summary statistics: median, two whiskers and two hinges. In addition the plot display the outlier points.

For example, in the boxplot of the fig. 3.5 we can see that the lower whisker of the quiz 03 is much smaller than the other quizzes and, as seen in the histogram of the quiz 03, we can identify that this distribution has a much bigger mode than the other 3. As a consequence, the 50% of the data is in contained in a smaller interval, which is reflected in its boxplot.

---

[4]Basically the other types of boxplots differ in the way that the whiskers are constructed.

3.3.3 Easiness-Time plot

The objective of the Easiness-Time plot (ET plot) is to analyse the relationship between the required time for each item versus the average score obtained in the corresponding item. In the quiz design, it is important to consider the required time that each student should spend on each item, this way the instructor can design quizzes according to some pre-established time and discrimination criteria. With this plot, the instructor can analyse both points. Also, a tendency line is displayed in the plot.[5].
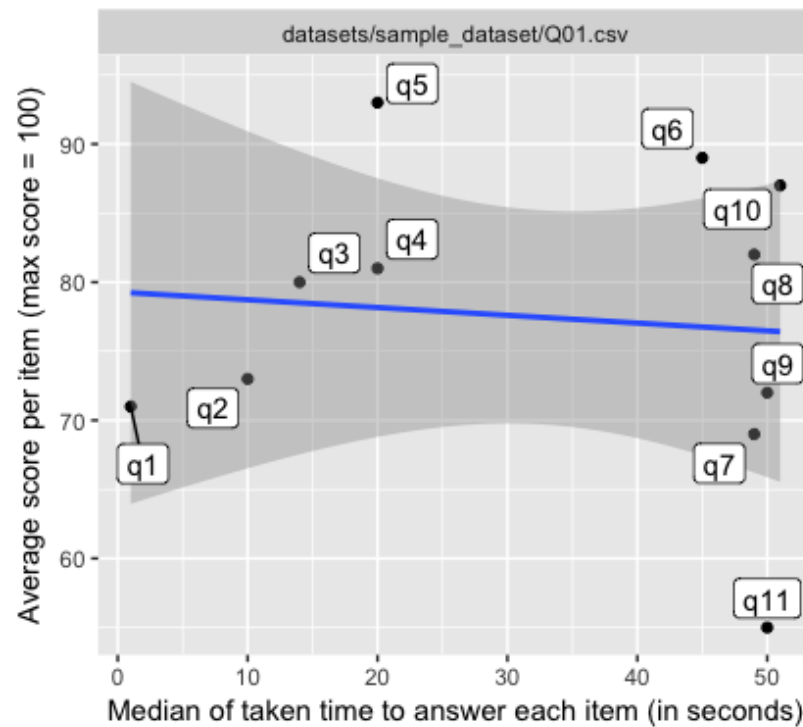
Figure 3.6: Easines-Time plot

For example, in the fig. 3.6 we can see that question 5 is taking 20 seconds to answer (an intermediate time in this example), but the students achieve a high average score. On the other hand, the question 11 is taking too long for the students, and many of them get it wrong. Then, the instructor can examine why the students are getting wrong the question (because it seems that it is not a matter of time).

3.3.4 Easiness-Time-Level plot

The ETL plot [21] adds the extra layer of the cognitive level to the ET plot. As stated at the beginning of the section, in this package we consider three different cognitive

---

[5]To avoid overlay in the labels of the points; the ggrepel package was used.

levels. The hypothesis is that difficult questions (in the sense of cognitive level) should take more time to answer than the easy ones. If we colour the points by the cognitive level, we might see that the high cognitive levels take in average more time to answer them (because we need to know the concept, then apply it). Moreover, in questions that require a high cognitive level, the students may get a lower score. The reasoning behind this idea is that, if the students don't know the factual knowledge (Low cognitive level), they will not know the application of this concept (High cognitive level), so the score is lower.

The importance of this plot is to detect the questions where time and difficulty do not align with instructor-rated pre-established cognitive level. For example, if a low cognitive level question is taking too long to be answered, it could mean that this question is being difficult to understand or that the topic, in general, is difficult for the students (For example, in fig. 3.7, the question 6 and 10). Another example is the behaviour represented in question 1. This is a high cognitive question but the student answer it fast, so almost 30% get it wrong. This can imply that the students have a misconception for the topic related to this question (the students think they know the answer, but they get it wrong).
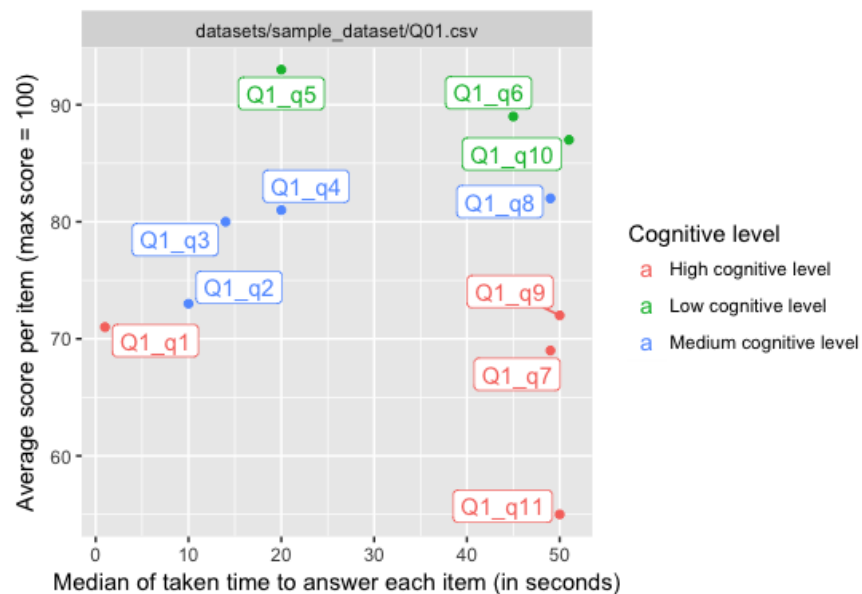

Figure 3.7: Easiness-Time-Level plot.

3.4   Individual plot

Although the main objective of this package was to provide a way quickly analyse the data for all the group and quizzes, the instructors may be interested in visualising one particular student or group of students. In special if these students were outliers in the group and quiz analysis. This way, they can look for a specific way to support these students. To explore and understand the history of these particular students, an individual history plot is implemented. For this plot, the instructor needs to filter the required students from a list and then observe its history and scores in different quizzes.
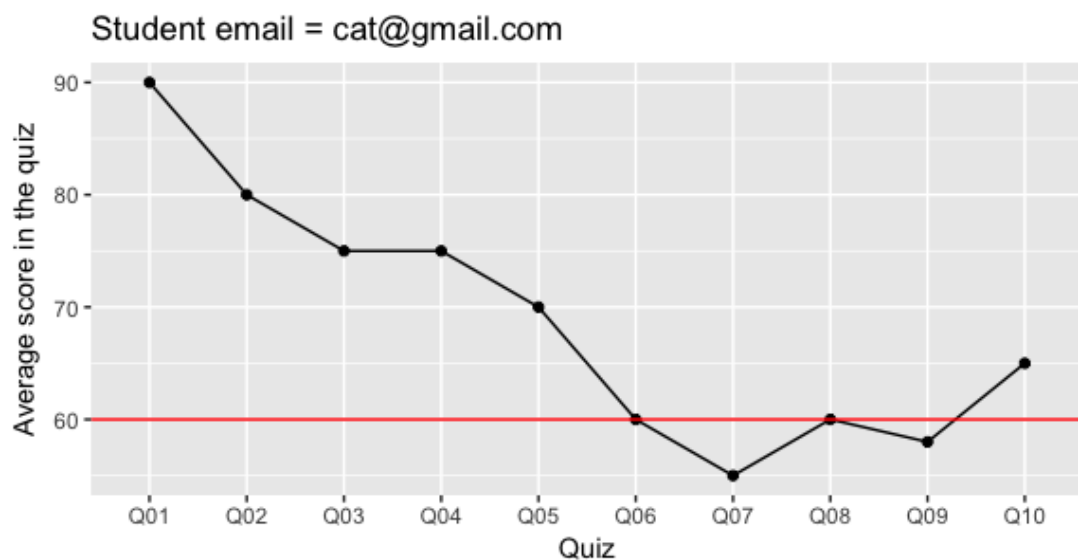
Figure 3.8: Individual performance

Besides, when the course finishes, the instructor can upload a file with the final grade of the students and observe distinct patterns that can lead to failure of the test. An example of this graph is in the fig. 3.8. We can see that this student is getting worse in his evaluations, so an opportune instructor intervention in an early stage of the course could improve the student's performance on the final exam.

3.5   Documentation

Finally, the documentation is important for the package. For this reason, two different ways to make the documentation were compared. The first way, the codes were commented manually and then a document explaining how to call each function was created. This result to be an inefficient way because, in each change of the functions,

both documents have to be modified. Then an alternative was considered. Then the roxygen2 package [23] solves this problem. With this, it is only necessary to comment the codes with specific fields [6] so that .R detect and compiles the documentation automatically. Also, it is very important this package because it facilitates the creation of the namespace and the exportable objects. Once used, .R automatically creates a *.Rd* file for each function and saves it in the *man* directory.

To make more accessible the documentation, a webpage was considered. One way that works perfectly with the roxygen2 package is the pkgdown package, that helps us to create a website for the package (with the advantage that the examples in the documentation are executed and displayed). This package takes the generated documentation in the man directory and creates a webpage in the docs directory with the README file as the home page of the website. This structure works good with the free webpages storage of Github Pages, so it is used this way.

---

[6]it requires fields as title, description, parameters, return, examples, export

Chapter 4

Shiny interface

Right now, the *lanalytics* package is freely available from Github, but for people that are not R users, an interface should be implemented. The selected option is a Shiny dashboard [20] [5] that contains the plots of the *lanalytics* and the eRm packages. Shiny was selected because it is native to R (although it is a wrapper for HTML and CSS code). Also, it is designed to have the R console as backend (allowing to use any R package).

The suggested dashboard contains five tabs related to distinct aspects of the *lanalytics* package. In this chapter, a quick introduction of the design of the dashboard is provided, the theoretical justification, the technologies for the dashboard and the technical requirements. After that, the each tab will be explained in detail. The first tab only contains general instructions; the next two tabs are focused on importing and displaying the databases, while the last two tabs are focused on analysing the data in three different grouping levels. Furthermore, the last tab is used as an interface to the Rasch models of the eRm (extended Rasch model) package [13]. The usability of this Shiny Dashboard was evaluated with a focus group consisting of 3 people, but future evaluations with more people are suggested to make it usable for users in general.

As a reminder, the objective of the lanalytics package is to provide a tool for management and constant analysis of student answers and test times. Also, with the help of the eRm package, inferences related to the item difficulty and the abilities of the students is provided.

4.1 Design of the dashboard

4.1.1 Initial investigation

The idea behind this dashboard is that the instructors perform Just in Time Teaching and adapt their tests difficulties and necessities according to the specific group require- ments. This idea is expressed in one of the five principles of social constructivism [1]: "A learning environment needs to be flexible and adaptable, so that it can quickly re- spond to the needs of the participants within" [18] [14]. Following this idea, and with the help of the *lanalytics* and *eRm*, the dashboard could be a tool that provides useful information to the instructor so that he takes his understanding of the group and the quizzes.

4.1.2 Technologies for a dashboard

To create a Dashboard, there are distinct alternatives. For this dashboard, two software were studied. The first was Tableau Desktop, where the developers can easily create a dashboard without any programming background. The other analysed dashboard technology is the R Shiny software. This is based in the R-language, and you need to build the dashboard from scratch (Although there are packages that allow some predefined functions).

Tableau Desktop has many benefits. One of the principal is that different filters and rules between the filters can be applied very easily. Moreover, different pre-defined plots are available with just with a drag and drop feature. Also, it has the advantage that the Tableau dashboard can be viewed by everyone (just with the software installa- tion), but to develop or modify a dashboard you need a license [2]. For the connection to databases, you can connect Tableau to distinct sources of data, but this is a disadvan- tage for this particular project because the quiz data is stored in a CSV file or different formats. One major disadvantage of Tableau is that is only for visualisation purposes, so it is not possible to automatically apply a complex model in this software to the data.

On the other hand, *shiny* is a web application framework that is native to R. In general, it is useful to create interactive applications, but if a dashboard is desired, an

---

[1]In the literature, they use the idea that "Social constructivism emphasizes the importance of cul- ture and context in understanding what occurs in society and constructing knowledge based on this understanding" [12] [16]

[2]although there is a free trial for the product

additional package should be installed. For the project, the *shinydashboard* package was selected. The main advantage of this software is that is open source, and any application can be created. Besides, if some computation of a model is required, it is connected to the R console. Another main advantage of this software is that it allows the preprocessing of the files, allowing the flexibility to import quizzes from many formats [3].

4.1.3 Requirements for Shiny

Finally, to connect to the Shiny dashboard there are three alternatives: the first is to connect to the free service provided by RStudio in the website *http://www.shinyapps.io/*, although this is limited by a monthly quota. The second way is to host it in Github and run it locally with the command *runGitHub(repo = "savrgg/lanalytics", subdir = "shinyapp/")* and the third way is to clone the repository and run it locally. Due to the quota restrictions, it is suggested to run locally the project. To do it correctly, a prior installation of some packages need to be done:

Table 4.1: Required packages to run locally the lanalytics Dashboard

Package	Usage
shiny	Provides the functions to create interactive applications.
shinydashboard	Build on top of shiny, provides a template for Dashboards.
DT	Provides useful functions to display tables and interact with them.
tidyverse	A collection of useful packages to analyze data.
stringr	A package to manipulate strings.
eRm	A package that implements extended Rasch models.
ggrepel	Functions to separate overlapping text labels.
devtools	Package with different tools for developers.

---

[3]the current version only allows from R data format and CSV, although it is possible to create functions to use it with any input

4.2   The lanalytics dashboard

In R, under the Psychometric Models and Methods page, there are more than 40 statistical packages related to Item Response Theory. These are hosted in the CRAN across distinct servers and can be downloaded and used within R. Considering the realised investigation, two of the more useful for this project are the *(latent trait models (ltm)* and the *extended Rasch models (eRm).* The first package implements the Rasch model, the two parameter logistic and the Birnbaum's Three-Parameter models. These include one or more of the following parameters: the difficulty of the items, the discrimination of the quiz and the guessing probability that a user had guessed a question [19]. On the other hand, the eRm package has a different estimation procedure of the parameters and introduces tests to verify if the assumptions for the models are met [13]. These packages include many other models, but require that the users know how to code in R. This can be time-consuming for the instructors.

4.2.1   Content of the dashboard

In this section, the main design for the dashboard is explored. As mentioned before, this consists of 5 different tabs that incorporate functions from the lanalytics and the eRm packages. All the screenshots of this chapter are in the appendix A.

Tab 1 and Tab 2: Instructions and import data tab

As mentioned, the first tab contains the general instructions to operate the dashboard. Also, it contains a small explanation for the installation of the lanalytics package. This tab was designed just to introduce the user to the dashboard. The second tab contains input boxes for the quizzes files, the cognitive level file, and the final exam score file (optional). Shiny has the capability to upload many files at a time, but it should be opened in a web browser. Once a file is uploaded and the corresponding upload button is pushed, a list of the uploaded files will appear below the input boxes. The fig. A.1) contains an image of the first tab and the fig. A.2) of the import quizzes tab.

Tab 3: Display quizzes tab

The third tab (*display quizzes*) was created to show the uploaded quizzes in the dashboard. First, it shows the quiz datasets; after that, the user has to select the columns for the cognitive levels file that contains the item and the corresponding cognitive level.

Finally, the cognitive level file and in case that final exam results are uploaded, these are shown in the bottom of the page. To allow interactivity in the tables, such as filters and ordering, the DT package is used. This package is an interface to the DataTables library from JavaScript. The fig. A.3 contains a screenshot of the design of this tab.

Tab 4: Data analysis tab

The fourth tab contains the quiz analysis tab. This is formed with another three sub tabs, each one of them representing distinct grouping level.

Sub tab 1: Group analysis tab

The first sub tab is the group analysis. First, in this group analysis, the quizzes to analyse have to be selected. Because of the nature of the plots, only one quiz at a time can be displayed. After the quiz is selected, two different plots are shown. As a reminder, the plots corresponding to the group level are the guessing plot fig. A.4 and the easiness-time per terciles fig. A.5.

Sub tab 2: Quiz analysis tab

In the second sub tab, the analysis at the quiz level is displayed. Similarly to the previous sub tab, the users have to select a quiz, only that in this tab, they can select one or more (although is suggested to select a few because it takes some time to plot many quizzes). In this tab, the boxplot of the scores, the histogram of the scores, the ET-plot and the ETL-plot are shown [fig. A.6, fig. A.7, fig. A.8]. This tab was designed so that the instructors could analyse all the items from the quiz in the same place.

Sub tab 3: Individual analysis tab

In the last sub tab, it is shown the individual analysis. Here the users have a student email and also many quizzes analyse. If the final exam file is uploaded, then a red horizontal line will be plotted to show how the final grade was related to the grades from the quizzes. This is useful because the instructor can analyse the behaviour of the very good students or very bad students to understand how are they working through the course.

Tab 5: Rasch model

Finally, the last tab of the dashboard contains the plots related to the Rasch model. In this tab, the explained plots in chapter 2 are displayed. The ICC plot fig. A.10, the

person-parameter plot fig. A.11 and the person-item map fig. A.12] are provided. This way, the users can compare the difficulty of the items compared with the ability of the students.

Chapter 5

Evaluation

As stated in previous chapters, the Learning Analytics Software was created with two main objectives. The first is to monitor the performance of the students in different levels: individual, per quiz and per group. The second is to ease the understanding of the items in online quizzes to improve the design of future quiz items [1]. These two objectives were evaluated with a focus group that consisted of 3 persons working with online quizzes and interested in learning analytics topics.

5.1 Focus group structure

The general structure of the focus group consists of two big sections of 30 minutes each. The first was a general introduction to the Learning Analytics objectives, the implemented models, and its capabilities. During this section, a sheet of paper with a series of questions related to objective accomplishment was provided. The next 30 minutes of the focus group consisted of a user interface evaluation, where a synthetic dataset and a series of tasks where given, then the participants had to evaluate some affirmations about the usability.

5.1.1 Participants

For the focus group, eleven persons were invited via email to participate in this focus group. All of them are faculty members of the University of Edinburgh and had shown an interest in online learning tools. Due to their busy agenda, only three of them could assist. All of them professors from the School of Medicine. This focus group was

[1]so that the instructor could quickly adapt the quizzes to the needs of the students in the group

conducted by a moderator.

5.1.2 Synthetic dataset

To test the usability of the first version of the lanalytics package and dashboard, a synthetic dataset was created. This dataset consists of 12 quiz data files, one file for the cognitive levels and one file containing the final grades of the students. These were specially designed to show some specific patterns in the plots, so in this section, the dataset will be described.

To generate the 100 email users a 10-character string was randomly generated, then each of these students was assigned to one of three groups (thinking of subpopulations inside the same group; for example, students that usually get high scores, students that get medium scores and students that get low scores). Then, for each email, 12 quizzes with 11 question each were generated.

Now, the probability of obtaining the right answer for each question was distributed as a Bernoulli distribution with distinct probability according to the group that the user belongs (of the three groups randomly generated). The table 5.1 states the exact probabilities for each subpopulation and item.

Table 5.1: Probability to get correct the answer

Item	High skilled students	Medium skilled students	Low skilled students
1	0.83	0.73	0.63
2	0.85	0.75	0.65
3	0.90	0.80	0.70
4	0.95	0.85	0.75
5	0.95	0.90	0.80
6	0.70	0.65	0.60
7	0.70	0.65	0.60
8	0.95	0.85	0.75
9	0.85	0.80	0.70
10	0.85	0.85	0.85
11	0.70	0.60	0.50

Now, for the answering time per question, two assumptions were taken. The first is that the date for any answer of quiz one might be answered before of the date of

any question of quiz 2 and so on. To ensure this, each quiz was in differents months. Now, for the answering time of the questions in the same quiz, it was assumed that the students answered the quiz in order and in a continuous consecutive interval of time. Then, if we calculate the time difference between the two questions, it would represent the spent time in the question. To create this synthetic variable, a normal distribution with different mean and standard deviation was used.

The final exam file was generated with the same users as the quizzes files, and the final grade was normally distributed with a centre in the student mean score of the quizzes. Finally, the cognitive file was generated based on the probabilities of getting correct a question table 5.1. This way, the items that were easier would have a cognitive level of 1 (the easiest one) and consecutively until the cognitive level 3.

5.2   Focus group

5.2.1   Introduction of the software (30 minutes)

This section of the focus group is divided into two parts, one of 10 minutes and the other of 20 minutes. The first part is to introduce the capabilities of R and Shiny to motivate future uses and further implementations. The second part is an introduction of the Learning Analytics Package objectives and its Shiny interface. In this phase, the motivation of the models and the generated plots were explained.

As part of the evaluation, five questions were asked in a sheet of paper for each of the participants. The objective was to discuss if the presented models were consistent with the objectives. The list of the five questions presented are the following:

- Do the shown figures in the **Individual analysis** help you to monitor how an individual student is performing? What else would you like to understand about the student?

- Do the shown figures in the **Group analysis** help you to monitor how the group is performing? What else would you like to understand about the group?

- Do the shown figures in the **Quiz analysis** help you to understand the difficulty of the items as well as the required time per quiz? What are the main points that you consider when designing an item of a quiz?

- The **Rasch model** helps you to understand the easiness-difficulty of your items?

- The Rasch model plots **(ICC, Person-Item, Person-parameter)** gave you actionable information to improve the design of your quizzes?

5.2.2   User experience evaluation (30 minutes)

Once the presentation of the software is concluded, the synthetic dataset is provided and a 20 minutes user experience evaluation starts. In this part, the users have ten tasks to accomplish [table 5.2], with the central objective to evaluate the usability of the dashboard. In particular, they have to evaluate the easiness of the instructions as well as the easiness of the elements in the dashboard.

Table 5.2: Tasks to accomplish

Tab	Task
Import quizzes tab	Upload the quizzes files
Import quizzes tab	Upload the cognitive file and the final exam file
Import quizzes tab	Remove the cognitive file and upload it again
Display quizzes tab	View if the uploaded files are correct
Display quizzes tab	Select the Item and the Cognitive level column for the cognitive file.
Individual Analysis tab	Search one student and view its grades.
Group Analysis tab	Select one quiz and view the guessers and order plot
Quiz Analysis tab	Observe the histogram and boxplot.
Quiz Analysis tab	Observe the ET and ETL plot. Do you see some pattern with the cognitive level?
eRm package tab	Select one file and view the ICC, the person item map and the person, parameters plot.

To measure the usability of the dashboard, some affirmations are given, and the users have to agree with them according to a Likert scale (1 = Strongly disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, 5 = Strongly agree) [table 5.3].

After this set of affirmations, some general perceptions of the dashboard were asked (on the same Likert scale) table 5.4.

Finally, in the last section of the focus group, a final round of 10 minutes of open

Table 5.3: Likert evaluations

Tab	Question	SD	D	N	A	SA
Import Quizzes	It is easy to upload files.	1	2	3	4	5
Import Quizzes	It is easy to remove files.	1	2	3	4	5
Import Quizzes	It is clear when a file is uploaded.	1	2	3	4	5
Import Quizzes	It is clear when a file is removed.	1	2	3	4	5
Display quizzes	It is clear the format of the quizzes.	1	2	3	4	5
Display quizzes	It is clear how to select the columns of the cognitive file.	1	2	3	4	5
Individual	It is easy to search for an student email.	1	2	3	4	5
Individual	It is easy to visualize which quiz is displayed in the plot.	1	2	3	4	5
Individual	The plot is easy to read.	1	2	3	4	5
Group	It is easy to visualize which quiz is displayed in the plot.	1	2	3	4	5
Group	It is easy to identify possible guessing or cheating misconducts.	1	2	3	4	5
Group	It is easy to understand the relation of time per question versus the grade.	1	2	3	4	5
Quiz	It is easy to visualize which quiz is displayed in the plot.	1	2	3	4	5
Quiz	It is easy to interpret the histogram and boxplot.	1	2	3	4	5
Quiz	The ET and ETL plots are easy to interpret.	1	2	3	4	5
eRm	The Item Characteristic Curve is easily understandable.	1	2	3	4	5
eRm	The person-item plot is easily understandable.	1	2	3	4	5
eRm	The person parameters are easily understandable	1	2	3	4	5

Table 5.4: Likert evaluations

Tab	Question	SD	D	N	A	SA
General	In general the models agree with the general objectives	1	2	3	4	5
General	In general it is easy to use the buttons.	1	2	3	4	5
General	In general it is easy to navigate between tabs.	1	2	3	4	5
General	In general the colors of the dashboard are adequate.	1	2	3	4	5
General	In general the instructions are clear.	1	2	3	4	5

commentaries and suggestions is made. The objective of this final round was to provide
a guideline for future work based on the different necessities of the users.

5.3 Results

The focus group was designed to take 1 hour, but it takes 1 hour, and 20 minutes, that
is 20 minutes more than the expected. This is a sign that the users found difficult to
follow the instructions of the dashboard.

5.3.1 Results for the introduction of the software

From the first part of the focus group, the concerning to the lanalytics package and the
Rasch model the following were the commentaries:

Table 5.5: Main improvement points for the plots

Analysis	Ideas to improve
Individual analysis	- Allow the user to display multiple students in the same plot
Group analysis	- Include a plot indicating the group average score in all quizzes (not only by tercils)
Rasch model	- Needs better explanation to make interpretable and actionable the results.

One point that was difficult to explain and it seems that was not clear enough was

the Rasch model. The idea of an ability parameter per student and a difficulty parameter per item was clear for the participants, but they find difficult the idea of how to improve their quiz design with this model. A point that was especially difficult to understand was the concept of a latent trait as a general ability variable. Another point that was difficult for the Rasch model was the plots. A question that arises in the focus group was related to the advantage to model a difficulty parameter compared with just calculating the number of correct questions per item.

For the other plots in the presentation, there were commentaries about further capabilities in the plots. For example, in the individual plot, one of the participants suggested being allowed to plot more than one student in this plot, to be allowed to evaluate a subset of the students. For the group analysis, one of the participants suggested to include an average for all the group in the easiness-time per time terciles plot, to observe the general performance of the quiz. Also, the labels for some plots were suggested to be more clear.

5.3.2   Results for the user experience evaluation

During this user interface evaluation, the behaviour of the participants was observed with the following comments:

For the Import quizzes tab, it was easy to upload the files and remove the files, but it was complicated to know which files are uploaded or the exact moment of the upload. Also, there should be a functionality to delete just one quiz instead of all of them. The button of "upload quizzes" is not clear, as they do not have the explicit instruction to press it after selecting the files. It would be better if in the instructions is considered the sentence "Now when you press the button the list of uploaded files will appear".

For the Display quizzes tab, it was easy to observe the new format of the quizzes. Also, it was clear how to use the filters that the user can apply. The difficult part for all the users in the focus group was the selection of the cognitive files columns. Because of the layout position of these boxes, it was difficult to know what to press and what columns should they select. One of them mentioned that the position of these boxes in the tab should not be there, but at the beginning of the tab.

For the Individual analysis tab, two of the participants suggested that there should be indicated that they could write the email in the search box, but the instructions were not clear. But the plot was useful, and the task was in general easy in this tab.

For the Group analysis tab, one participant tries to plot information for many

quizzes and the plot shrunk up too much, that can not be readable. Also, the same user detect that each time that the quiz selection box is moved, the plots change (for example, if five quizzes were selected, and then two of them are removed, the dashboard first re-plot 4 quizzes and then re-plot again 3 of them) so the user has to wait until is plotted. Another useful comment was to change the colour of the guessing plot, to indicate with red the possible cheating scenario. Finally, the users think that instructions of this tab were not clear enough.

For the Quiz analysis tab, one person comment that the titles indicating the quiz name should be bigger and also the font for the four plots. Also, one person suggested explaining further the blue line in the ET plot. For this tab, there were no commentaries about the layout of the plots.

For the eRm package tab (Rasch models) all the users suggested that besides the instructions, there should be examples of related analysis that they can do. This with the objective to make valuable this tab. Questions like How it is interpreted? Why is this important to me? Need to be answered in the instructions for each one of these plots.

In the general affirmation section, the users suggested that the dashboard and the models agree with the objectives, also that the general use of the dashboard is easy (move between tabs, use the buttons, the colours, and the design). One of the users suggested that the word *question* should be used instead of *item*. Finally, all of the users agree that the major area of improvement was the instructions specification, in special use lay terms instead of specialised ones, to include a button that explains in a really deep way each of the plots (thinking in all kind of users) and to be like tutorial instructions. (In addition to the suggestion of proofreading the dashboard with a native speaker in order to make the instructions the clearest possible).

In the following table fig. 5.1, we can see the exact Likert evaluation of the tasks:

By analysing the qualitative and the Likert evaluations, we can give more general areas of improvement. Concerning the lanalytics package, the majority of the plots were useful, but improvements regarding the titles, labels and additional characteristics were given. This is important because of shows that the content of the package is useful itself. On the other hand, for the lanalytics dashboard, the general concept and idea of the dashboard are attractive itself, also the general layout and proposed design. The area of improvement is on the instructions and analysis examples. If the instructors find difficult to use a tool that is designed to help them in their quiz analysis, then it is probable that they will not use it. For this reason, a detailed improvement on the

Figure 5.1: Evaluations in likert scale.

instructions is suggested. Then, another user experience evaluations will be required.

Chapter 6

Conclusions

In this dissertation, the use of a tool for analysing low stake-high frequency quizzes was studied. Two objectives for this tool were proposed. The first was the measurement of the difficulty of the items and the abilities of the students, this with the objective to make more suitable the items according to the student abilities. And the second was to propose an open source tool that eases the administration and the analysis of these high-frequency quizzes. To make open source this project, the R language was selected. Finally, for the interactive dashboard, Shiny was selected over other dashboard visualisation tools like Tableau.

In particular, for the first objective two packages for the Rasch models were explored. The first was the *ltm* package that contains the 1-parameter logistic model (1PL) that contains a difficulty parameter, the 2-parameter logistic model (2PL) that contains, also, a discrimination parameter and the 3-parameter logistic model (3PL) that besides contains a guessing parameter. The second explored package was the *eRm*, which contains the simple Rasch model that is similar to the 1PL. Also, it contains some useful plots to compare the item difficulty with the student abilities. Finally, for the included plots and the numerical stability, the eRm package was selected.

From the second point, the lanalytics package and the lanalytics dashboard were proposed, the package includes some basic descriptive analysis, while the dashboard includes the analysis of the lanalytics package and some plots of the eRm package. Initially, the software was thought to include quizzes from different input formats, but the variability of the inputs was too diverse that the homologation could be troublesome. In this aspect, the lanalytics package can be improved in a great sense. Right now the software accepts a specific *.csv format and also it accepts R data files. Also, the results can be output in .csv format and R data format.

A focus group was made to evaluate if the objectives of the project were accomplished. In general, the evaluations for the content of the package and the dashboard were evaluated good, but there is a huge area of improvement on the instructions specification, on the provided examples and the explanation of the importance of each plot. Finally, the users agreed that the plots and the content agree with the general objectives of the project. For future work, a series of improvements in the usability should be done. This with the objective to make more attractive the software and to incentivise its use.

Appendix A

Screenshots of the dashboard
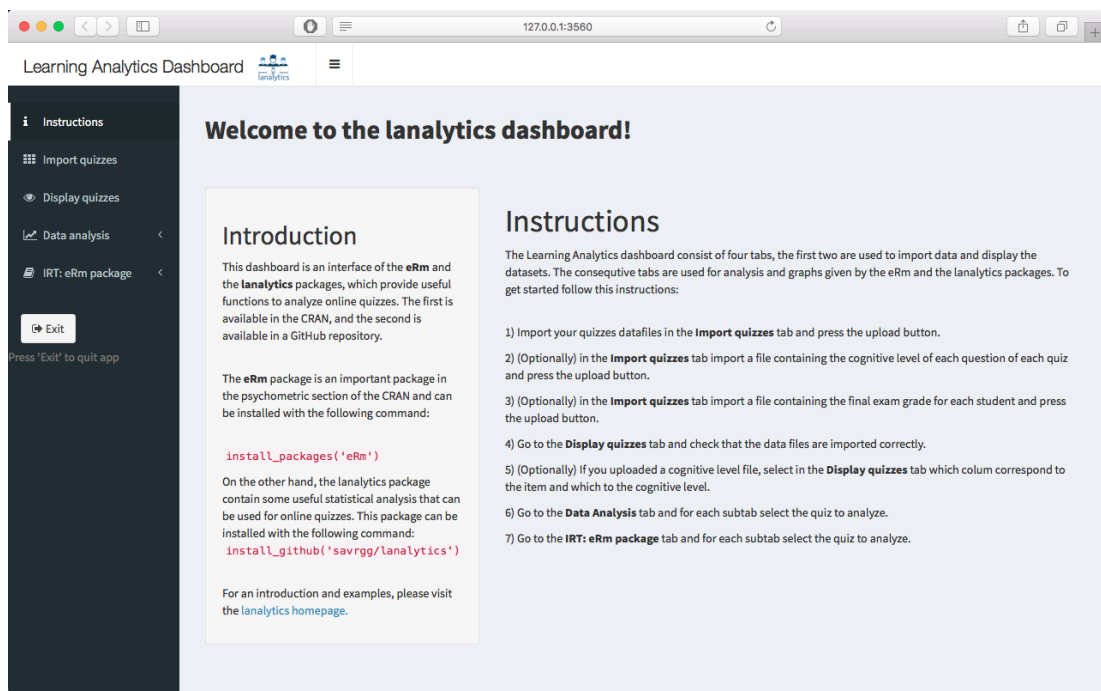
Figure A.1: Instructions tab
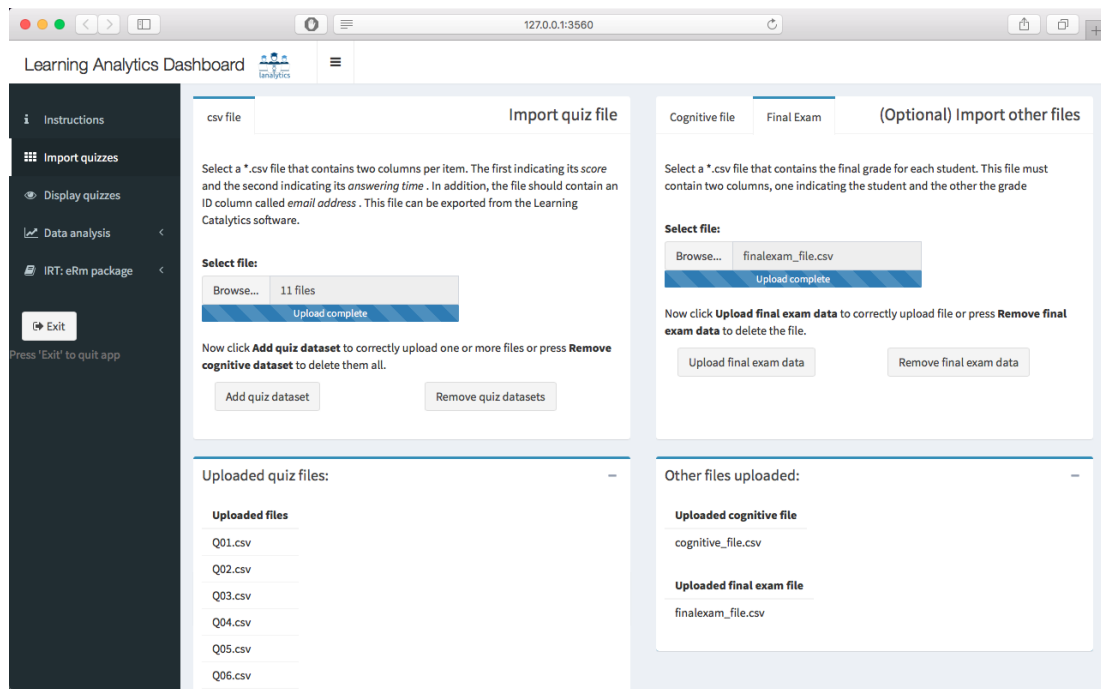
Figure A.2: Import data tab
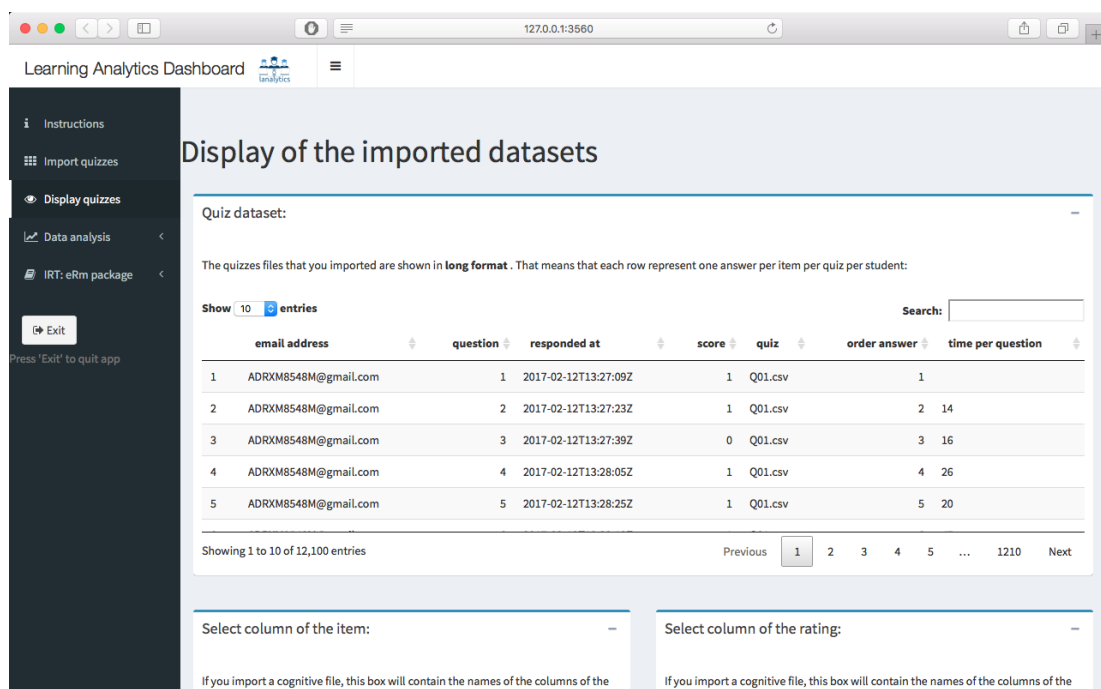
Figure A.3: Display quizzes tab
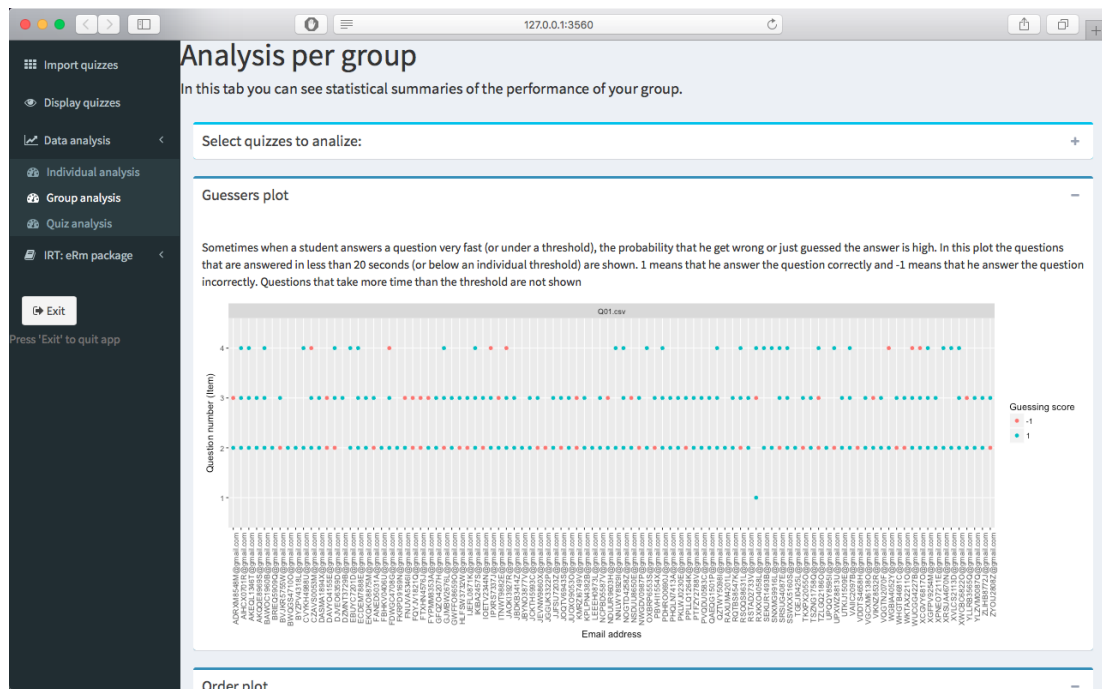
Figure A.4: Guessing plot

Figure A.5: Easiness-Time plot
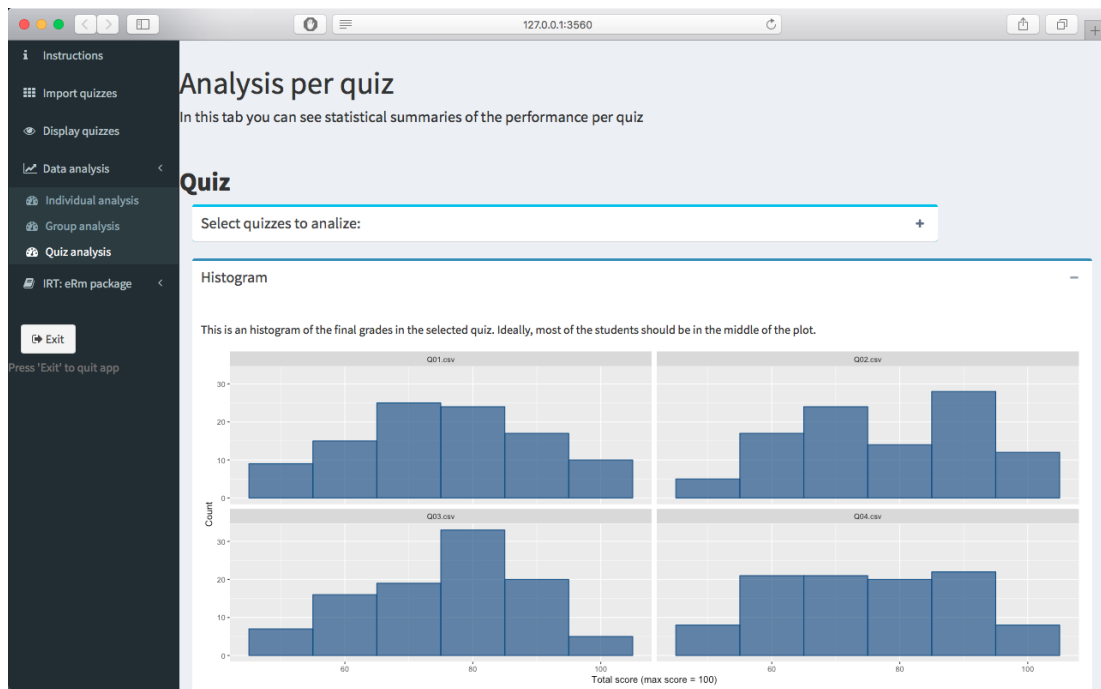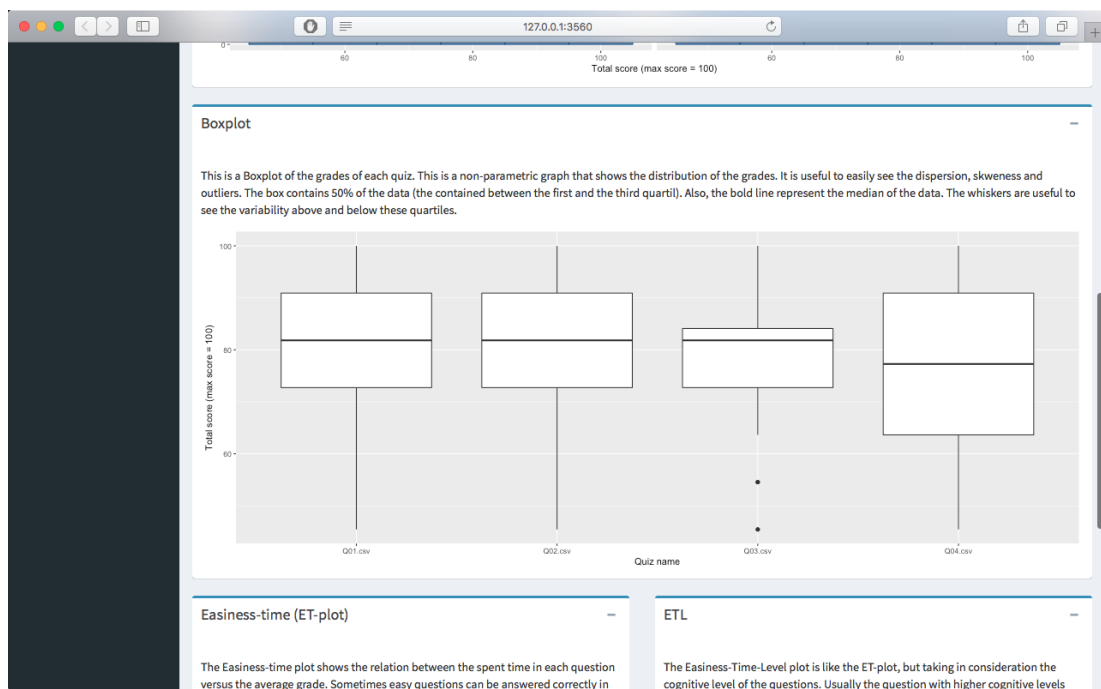
Figure A.6: Histogram of score per quiz

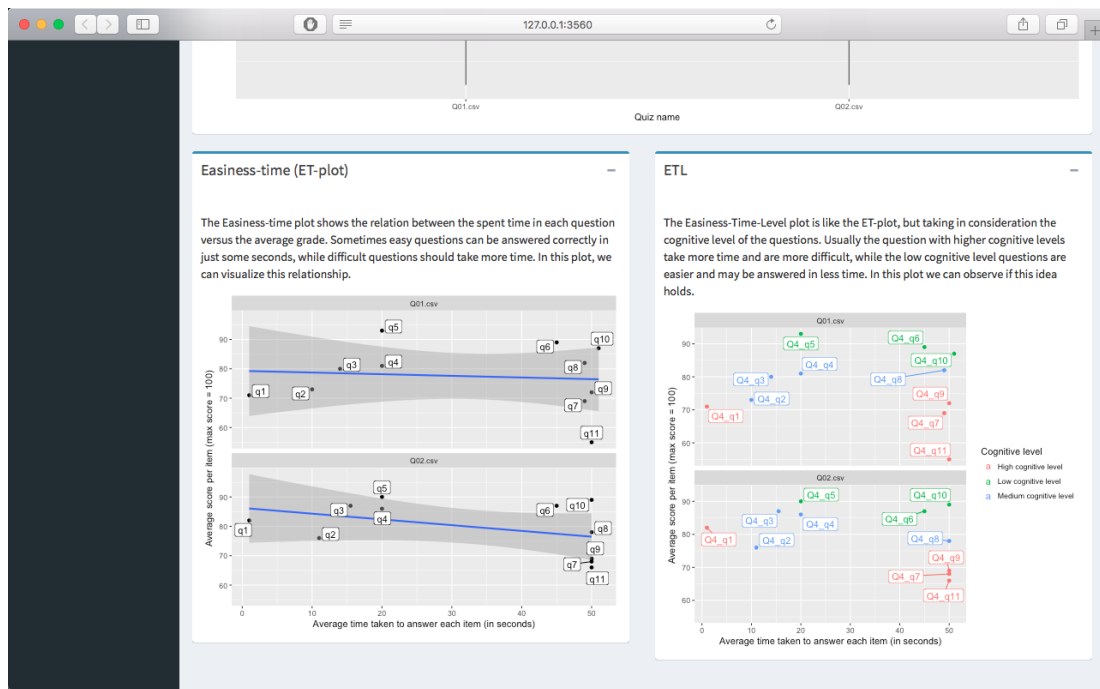Figure A.7: Boxplot of score per quiz
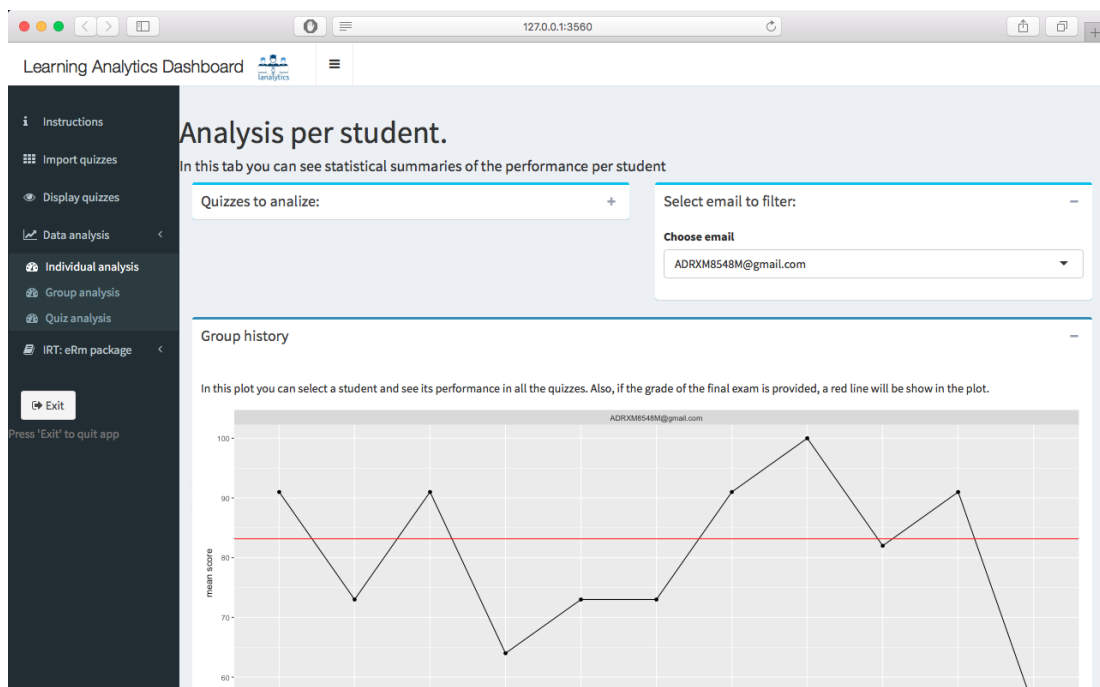
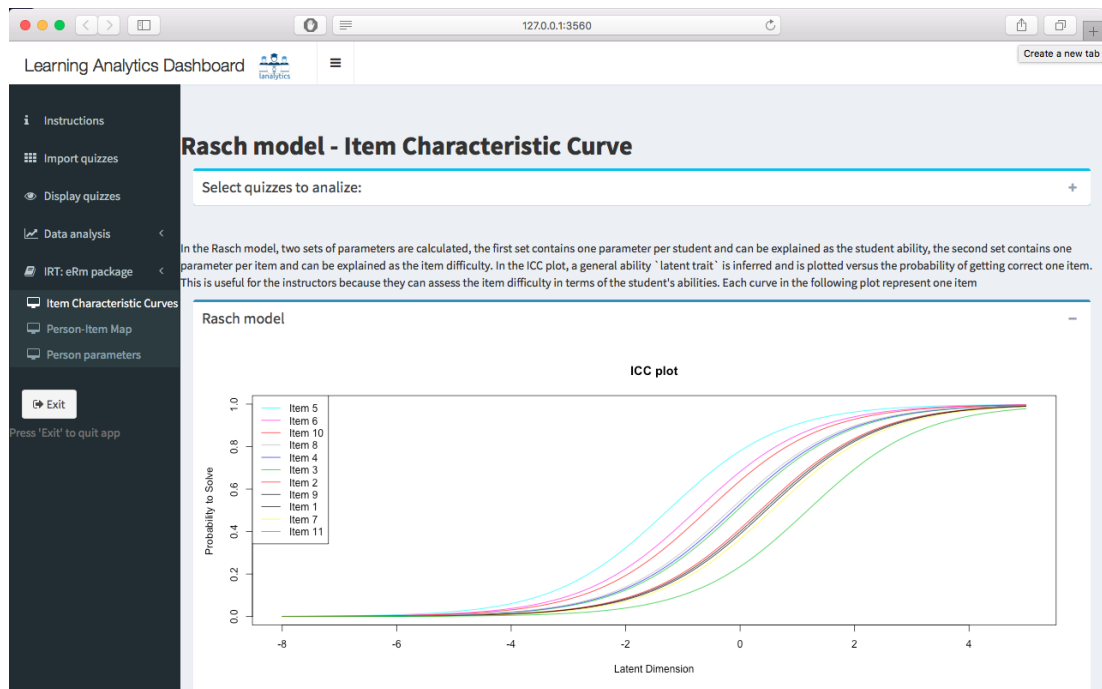Figure A.8: ET and ETL plots

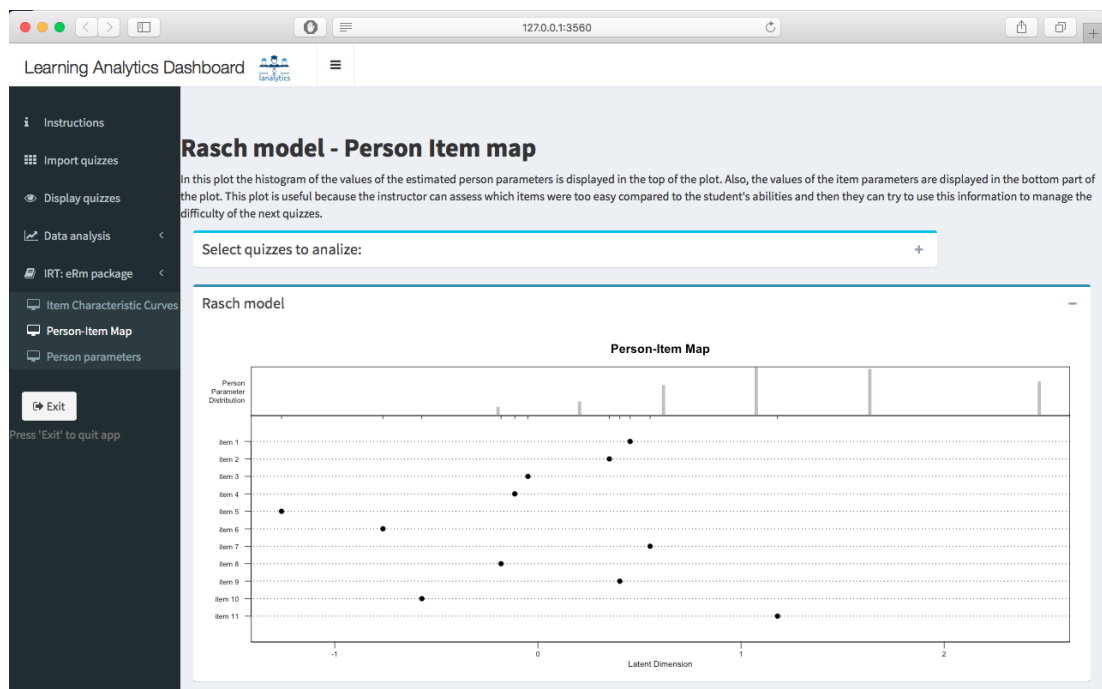Figure A.9: Individual analysis tab

Figure A.10: Rasch model - ICC
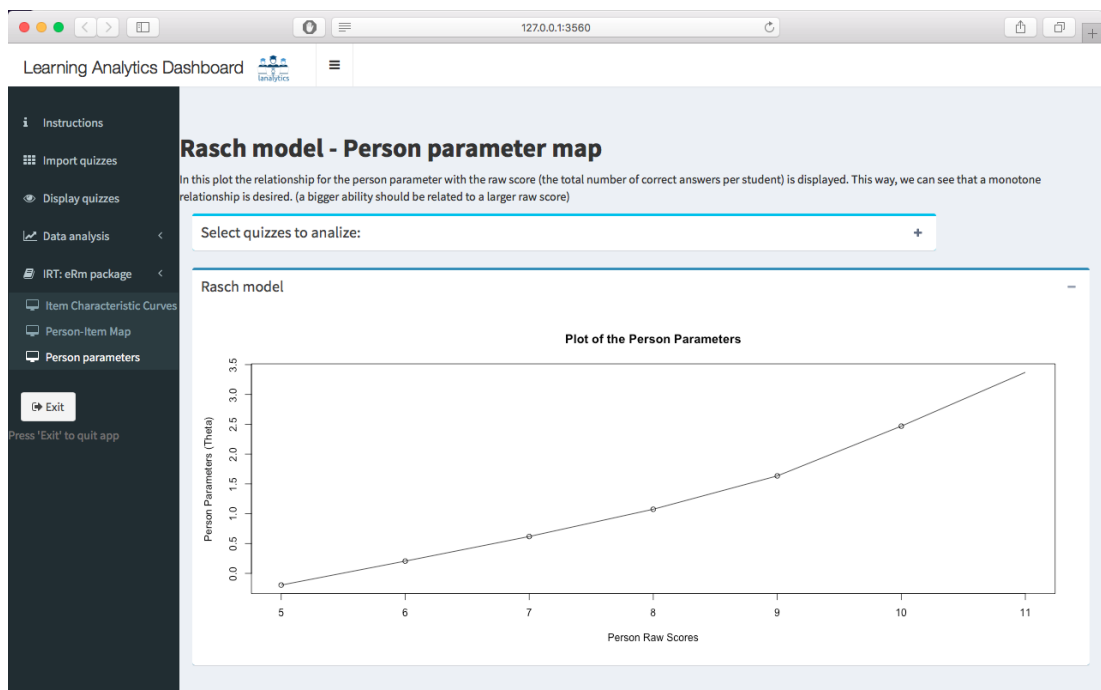

Figure A.11: Rasch model - Person Item map

Figure A.12: Rasch model - Person parameter map

Bibliography

[1] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[2] B Bloom. A taxonomy of cognitive objectives. *New York: McKay*, 1956.

[3] R Darrell Bock and Murray Aitkin. Marginal maximum likelihood estimation of item parameters: Application of an em algorithm. *Psychometrika*, 46(4):443–459, 1981.

[4] Trevor Bond and Christine M Fox. *Applying the Rasch model: Fundamental measurement in the human sciences*. Routledge, 2015.

[5] W Chang. shinydashboard: Create dashboards with shiny. *R package version 0.5*, 1, 2015.

[6] Christine DeMars. *Item response theory*. Oxford University Press, 2010.

[7] Desire2Learn. Desire2learn. `https://www.d2l.com/en-eu/`. [Online; accessed 1-August-2017].

[8] B Everett. *An introduction to latent variable models*. Springer Science & Business Media, 2013.

[9] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2009.

[10] Salvador Garcia Gonzalez. Developing a software for analyzing data from online learning quizzes. *Informatics Review Proposal*, 2017.

[11] Shelby J Haberman. Maximum likelihood estimates in exponential response models. *The Annals of Statistics*, pages 815–841, 1977.

[12] Beaumie Kim. Social constructivism. *Emerging perspectives on learning, teaching, and technology*, 1(1):16, 2001.

[13] Patrick Mair, Reinhold Hatzinger, and Marco J Maier. Extended rasch modeling: The r package erm. *PDF-Dateianhang zum Programmpaket eRm*, 2009.

[14] Dougiamas Martin. Moodle. `https://moodle.org/`. [Online; accessed 1-August-2017].

[15] Eric Mazur, Gary King, and Lukoff Brian. Learning catalytics. `https://www.pearson.com/us/higher-education/products-services-teaching/learning-engagement-tools/learning-catalytics.html`. [Online; accessed 1-August-2017].

[16] Mark McMahon. Social constructivism and the world wide web-a paradigm for learning. In *ASCILITE conference. Perth, Australia*, volume 327, 1997.

[17] Gregor M. Novak. Just-in-time teaching. *New Directions for Teaching and Learning*, 2011(128):63–73, 2011.

[18] Carla R Payne. *Information Technology and Constructivism in Higher Education: Progressive Learning Frameworks: Progressive Learning Frameworks*. IGI Global, 2009.

[19] Dimitris Rizopoulos. ltm: An r package for latent variable modeling and item response theory analyses. *Journal of statistical software*, 17(5):1–25, 2006.

[20] RStudio, Inc. *Easy web applications in R.*, 2013. URL: `http://www.rstudio.com/shiny/`.

[21] Melanie I Stefan and Yan Liu. lanalytics. *GitHub repository*, 2015.

[22] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2009.

[23] Hadley Wickham, Peter Danenberg, and Manuel Eugster. roxygen2: In-source documentation for r, 2015.

[24] Hadley Wickham and Garrett Grolemund. R for data science, 2016.

[25] Chris Williams. Probabilistic modelling and reasoning factor analysis and beyond. `http://www.inf.ed.ac.uk/teaching/courses/pmr/docs/fadoc.pdf`. [Online; accessed 10-August-2017].

[26] Benjamin Wright and Nargis Panchapakesan. A procedure for sample-free item analysis. *Educational and Psychological measurement*, 29(1):23–48, 1969.

[27] Achim Zeileis, Nikolaus Umlauf, and Friedrich Leisch. Flexible generation of e-learning exams in r: Moodle quizzes, olat assessments, and beyond. Technical report, Working Papers in Economics and Statistics, 2012.