Analysis of Moodle Metadata

Louis Shibu Thomas
21315313

Final Year Project – 2025
5 ECTS
Double Honours B.Sc. in Data Science

Department of Computer Science,
Maynooth University,
Maynooth, Co. Kildare,
Ireland

A thesis submitted in partial fulfilment of the requirements for the B.Sc. Double Honours in Computer Science.

Double Honours B.Sc. In Data Science

Contents

List of Figures

Declaration of No Plagiarism

0.0.1 Declaration

I hereby certify that this material, which I now submit for assessment on the program of study as part of (add your degree here) qualification, is entirely my own work and has not been taken from the work of others - save and to the extent that such work has been cited and acknowledged within the text of my work.

I hereby acknowledge and accept that this thesis may be distributed to future final year students, as an example of the standard expected of final year projects.

Signed: Date: 16/03/2025

0.0.2 Acknowledgements

I would like to thank Dr. Edgar Galván for guiding me through the process, Levis Shibu Thomas, Owen O Connor and Marie Gwenaelle Severine Penelope Mathieu for supporting throughout my Academic Journey

1 | Abstract

This study anlayses Moodle Metadata to evaluate student engagement patterns and their correlation with academic performance in a university setting. Leveraging R for statistical analysis techniques such as **density plot**. **time series decomposition** and **linear regression** was applied to Moodle Data. Key challenges included: **anonymization**, **cleaning temporal variables** and **addressing missing values**. Time series results revealed peak lab times from 15:00 - 17:00, engagement dipping after breaks, and surges before exams. Linear Regression revealed time-spent statistics. Moreover, this report highlights Moodle's potential in monitoring engagement informing lecturers on education adjustments through data interpretations. Future direction includes time-spent analysis and real-time analytical tools for deeper insights into learning behaviours.

2 | Introduction

As of now I am slowly becoming more of an academic, I have been taking part in the teaching/grading of junior students, While tutoring students I have been interested if I have been impacting students' grades, I believe with Moodle activity data we can see these results more clearly, necessitating a thorough investigation. Especially after covid, where hybrid learning approaches have become more common.

2.1 Why Moodle Metadata

Covid has impacted the usage of online learning environments. There has always been a trend of moving learning on to online platforms, however Covid accelerated that to a necessity. In Maynooth the main learning environment is Moodle, this is mandated to be used by every Lecturer, where the content is left to the discretion of the Lecturer. There are many data collection services provided by Moodle, where you can request access by being a (non-editing) Lecturer, these services are left to their discretion as well. With proper analytical tools, Moodle Metadata has the potential to change lecturer's understanding of their teaching styles, of quizzes of labs and more. With these changes aiding in the student retention and learning process.

2.2 Approach

Our approach combined investigations into Moodle data collection, and into the analysis of their Moodle Data. The Moodle data, given to us by Dr. Edgar Galván, was of a test course and of CS265 Sotware Testing 2023-2024. The test course was setup for an initial investigation if any statistical methods can be applied to create approachable visualizations that might aid a lecturer.

The main coding language used for this project is R. I believe that this language is perfect for running statistical methods and meaningful plots on the Course data. There are many data visualizations tools that are ignored due to the difficulty of creating them from scratch. Due to the reason, I believe that showcasing versatile R packages with plots that are intuitive to create is crucial for this project to be recreated for lecturers.

2.3 Thesis Structure

For the structure the Technical section will contain explanations into the statistical and visual tools that will be applied to the data. This will provide a basic understanding on the disciplines used and the relevance to the usage in later sections, describing characteristics of the Moodle data that we can get and small reviews into what has been used before. It will also explain the main reasons for using R, initially picked due to learning it in my undergrad.

The problem sections contains details into the issues faced with the analysing Moodle Data. That section will be split into the data anonymity, pitfalls of the information on Moodle data variables and questions on if the Moodle data is any good.

Finally the solution section will combine explanations to the codes - which is attached in the appendix and evaluations on what the figures are showing us. With the conclusion further discussing the results of the statistical analysis and some potential uses with the data. Concluding with what the most important information that a Lecturer should know and prospects for the future.

3 | Technical Background

3.1 Use of R

This project was done through R. Base R, can be used for some quick understanding, but would not be as presentable and more confusing to use. This is improved using the many packages that can be attached onto R. `dplyr` [6], this package provides an easier to read data-cleaning and data-aggregation code. That package can be seen with the `magrittr` pipes, %>% or |>. The main use of this is to make neater code. For example, take the code: `Data |> function(., param = param)`, this would take the data and 'push' it into the function. Finally for most plots, that you will see throughout the paper, will be done by using the package `ggplot2` [4]. Other packages like `ggparcoord` is built ontop of the `ggplot2`, which was used in figure B.3.5.

To go into more detail about the packages involved with the statistical methods the correlation plot 5.5 will use base R's `lm` library, this is a basic statistical method, linear model. This is discussed further in the below section. In figure B.3.4, we use a basic time-series decomposition to analyse the trend we are seeing.

A small issue faced when using R was how long the processes would take when running on the Activity Data - this would be no longer than a couple of seconds; however for much larger datasets another language could be needed.

3.2 Statistical Plots and Modelling Techniques

3.2.1 Density Plot

This is the main tool we will be using to visualize the data will be density plots. A density plot is a graphical representation of the distribution of a continuous variable [10]. Since some of the variables, are continuous, we would be losing important information using a histogram, where histograms requires a choice of bins, obscuring important features. Ultimately the smooth curve of the density plot would better capture multimodal data.

$$\widehat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{3.1}$$

Where:

- n is the number of data-points.

- $K(...)$ is the kernel density estimate, R automatically chooses Gaussian if not supplied.
- h is the bandwidth
- x_i is a data-point
- x is the point at which the density is estimated

The bandwidth h is the smoothing parameter, a small h results in a more detailed plot but introduces a lot more noise. Whereas a larger h results in a smoother curve. Base R leaves this bandwidth to the Silverman's rule of thumb [12], this chooses a prediction for a Gaussian Kernel - which we are currently using. In our application we see that the default bandwidth gives a decent curve, so we will stick with it. Furthermore, since we have a countable number of datasets we will be using the summation $\sum$ rather than the integral $\int$.

3.2.2 Time Series Components

For the evaluations made from appendix B.3.4, we will be using some ideas from Forecasting: Principles and Practice [5]. It states that an additive decomposition is appropriate when the magnitude of the seasonal fluctuations around the trend does not vary with the level of the time series. Since there is only fluctuations on times we explained in the solution section we will continue with the additive plot. Which is written in equation 3.2.

$$y_t = S_t + T_t + R_t \tag{3.2}$$

Where :

- y_t is the data
- S_t is the seasonal component, since we know that there is a weekly trend we choose an appropriate window size.
- T_t is the trend-cycle component.
- R_t is the remainder component, what is unexplained after decomposition.

3.2.3 Linear Regression

Here Linear Models [7] was only used in figures 5.5 and used in the line `geom_smooth(..., method = "lm")`, in B.3.6. Usually a linear model is fit such that the vector Y, is explained by some coefficient multiplied the matrix of X predictors plus some error term.

$$\mathbf{Y} = \mathbf{X}\beta + \mathbf{I}\epsilon \tag{3.3}$$

Where:

- Y is a vector of the response variable
- X the predictor variable..

- β is the vector of slopes that will be applied to X.
- ϵ is the error term, this is not known however we will use some statistical methods to minimize this.

In our use of linear models, we will be using the grade of each quiz as $\mathbf{Y}$. We then will use the count data as $\mathbf{X}$. So we will be modelling quiz grade on average amount of time a material use. To calculate our predicted beta's - $\hat{\beta}$, we will us the Maximum Likelihood of $\hat{\beta} = (X^T X)X^T y$. Since we are dealing with a single predictor variable the Maximum Likelihood can be thought of as a line of best fit. In relation with our correlation code in appendix B.3.6, you can see we have mean centered our quiz scores, making our reference β - the intercept - at 0. Therefore the slope would essentially be the correlation. High negative and positive correlations tell us that there might be some connection with X and Y.

3.2.4 Learning Management Systems (LMS) and Market Landscape

Moodle (Modular Object-Oriented Dynamic Learning Environment) is an open-source LMS widely adopted in educational institutions to manage course content, assessments, and learner interactions [1]. Key competitors include proprietary systems like Blackboard, which offers detailed activity dashboards [13], and cloud-based platforms such as Google Classroom. Moodle distinguishes itself through customizability, extensible plugins, and rich data-logging capabilities, making it a preferred choice for educational data mining research.

3.2.5 Educational Data Mining in LMS Platforms

We can apply data mining techniques to extract insights from LMS data. From previous research we can find what interactions are valuable and what Moodle can offer us.

- **Variables effects using Decision Trees**: relating interaction patterns with academic outcome using a decision tree algorithm [2]. However since a lot of data wrangling is necessary with the Moodle data.
- **Clustering for Personalization:** Using different learner profiles collected from pretest, self-reported questionnaires and more was used to assess learner knowledge, cognitive abilities and control strategies [3]. Bouchet et al, collected details such as log-files, facial expression, diagrams drawn and notes taken from paper to add to their Clustering Algorithm. However, Moodle would be much more sparse in comparison. Nonetheless we can look at what variables were of use in the clustering process and apply it to Moodle. Where the most significant variables, from their ANOVA's of grades:
  - Number of correct answers scored on the pretest.
  - Average number of correct answers on the first attempt of the quiz
  - Number of sub-goals the student took, and changed
  - Number of quizzes answered in an allocated period of time

Fore further context on what we couldn't do, was the amount of time embedded materials was opened during the quiz - this was not attempted during this project.

3.3 Material Information

3.3.1 Data Available in Moodle

We had access to two courses provided to us by Dr. Edgar Galván. One served as a test course to get a brief understanding of the data, it is available to (non-editing) Lecturers. It included information on what time students logged on and what they did, for simplicity we will call this **Activity Data**. The time variable encompassed year, month, day, hour, minute and second. Another notable variable was the event variable, which provides details on student interactions. However, it did not capture every click or action in detail. 3.1 is a table of the data available from Moodle data. There were more variables such as components, eventname and description, these were derivatives of the previously mentioned variables.

3.3.2 Continuous Assessment Data

Similar to getting Moodle Activity data, we can request for grade data. This would include student usernames, id and any graded items -where it has been graded by a (non-editing) lecturers or by Moodle's automated corrections. This would be as useful as the Lecturer designs it where challenging quizzes would be better at showing students' strengths and weaknesses more than quick easy quizzes. (Non-editing) lecturers wont be able to make inferences about teaching via grades until the grading process per activity is completed. This data is one of the most interesting to work with, as you can see students' progresses overtime and you can tie it with the activity data - for further analysis.

Variable	**Description**	**Example**
time	String of the time a student had some activity	"Day/Month/Year, Hour:Minute:Second"
userfullname	Fullname of user who initiated the activity	Unique variable put through a security key
eventcontext	Information of what event: admin, file, quiz and course was used	"Quiz: Lab 11 Quiz"
eventname	brief overview of the eventcontext, without specifics	"Quiz attempte"

Table 3.1: Summary of Moodle Student logs

4 | Problems

4.1 Is the Moodle Logs data good?

From our previous material, we might argue that finding the duration of time spent for material might be a good variable to discern student characteristics; however since we, have activity data we have to create some process to convert it. A much easier variable to calculate might be how many times student has accessed a file for a period of time. Both are generated and discussed further in the solution.

There are two key considerations we have to make before judging if the Moodle Data is good: what variables give us the most relevant information and what student behaviours is available to us. Then we will process the data in order to gleam meaningful information. In the next few subsections we will discuss variables in the Activity Log data and what behaviours can be inferred.

4.1.1 Time:

The initial time variable is written as a string, which would be difficult to translate into usable temporal value. Its format makes it impossible for sorting, plotting and running algorithms in R. This conversion is dependent on what packages we will be using - where most would require time in the `POSIXt` class. Having a date and time variable allows us the freedom to do two things: collecting daily/weekly/monthly information However there needs to be some investigation into it, to see if there are any viable solutions. From these solutions, stems questions on monitoring student retention, if there are patterns worth looking at and when people start dropping interest in academia.

4.1.2 Event:

There are two interesting event variables, as discussed in the technical background, the `eventname` and `eventcontext`. The eventcontext offers us details on materials by searching for words including, "File, Quiz, Attendance, Student Logs accessed", tracking actor activity. This includes administrative activities - which can be deleted before the masking. Whereas, the eventname would offer more general details on what the actors did, for example: "Quiz attempt updated, Quiz attempt viewed, and more". We will need to mine the data to get amount of times events have been triggered. This can be further developed to weight events according to being the most beneficial to students, inspired by paper [8]. However due to constraints we could not develop this further.

4.1.3 Student Information

Another variable that will prove problematic will be the `userfullname` and the `description` variables. This contains identifiable student information, and for ethical reasons we should not be inferring problems associated with students - that should be left with the discretion of the lecturer and the student. There needs to be a process to combine student data, so we are inferring on the population's trends rather than a student's. Or just as effectively create some masking method. Moreover, the `description` column usually contains information such as: "user with id '...' did event with id '...'", for our project deletion of this variable did not affect the analysis, or methods for masking, like abov, should be made for this. Usually the IP-address would be from Maynooth University's pool of Ip addresses]as people connecting via eduroam will be put under a MAC-address. However in the chance they are connecting from their personal IP-addresses it can contain some geographical information. Due to these we have chosen to delete the variable. Additionally, we will not be using this variable in our analysis.

4.1.4 Scale of the Data

Every interaction between a student and the Moodle webpage is taken as a log. Therefore the Moodle dataset will be massive. This would be information of what and when actors clicked on the webpage, so information as material, quizzes and labs would all be taken a log of. Due to the scale there will be too much data to make meaningful analyses from. Necessitating processes into aggregating the data - using means sums and more. This would provide us with a high-level view of the data giving us a more general idea of what information seems to be meaningful to the behaviours we are looking for.

4.1.5 Time Spent

Since we have times when students log on to Moodle we can get the difference between each event to get an idea of how long students spend on Moodle. Combining `eventname` or `eventcontext` with `time` can tell us the time spent. Table 4.2 is the list of events that might prove useful to make meaningful inferences. However there are many issues, we will face with getting time-spent from the activity data. For example, there is no event name or context telling us when students exit out of Moodle. Leaving us with data points telling us that a student spent a nonsensical amount of time reading Lecture slides. Another issue, is that between modules, there is no policies to download or host lecture slides. So some percentage of the students are reading off downloaded notes. Furthermore, some hosted sites wont give us an idea if students are staying active or staying idle.

4.2 Is the masked Continuous Assessment (CA) Data any good?

4.2.1 Initial problems with the data

This data will be as large as the amount of students in the module, making it much more manageable than the activity data. The data is also proportional to the amount of results the students receives from Moodle and Graders, sometimes including attendance as well as honour codes - which necessarily is not useful. Another issue that will be faced is to deal with students absent from labs with prior permission, code generated to create table 4.1 can be found in appendix A.1. Most of there proportion of missing students are around the 10%   20%. Furthermore, since the data was masked, we don't know if this cohort includes (non-editing)lecturers, which would explain a constant rate of missing students. Nonetheless we will give them a 0 grade to see if we can find characteristics of non-attendees. Finally we need some method to connect the CA data with the activity data.

Column Label	nmiss	n	propmiss
Attendance..Attendance.Week.2..Real.	24	233	0.10300429
Attendance..Attendance.W4..Real.	21	233	0.09012876
Attendance..Attendance.W5..Real.	31	233	0.13304721
Attendance..Attendance.W06..Real.	45	233	0.19313305
Attendance..Attendance.W07..Real.	49	233	0.21030043
Attendance..Attendance..Real.	22	233	0.09442060
Attendance..Attendance..Real..1	22	233	0.09442060
Attendance..Attendance..Real..2	22	233	0.09442060
Attendance..Attendance..Real..3	22	233	0.09442060

Table 4.1: Missing Data Summary

Event Name	**Use**
Course viewed	This is when students view the initial website, this can tell us when students are logging for the first time; especially when there are long breaks beforehand.
Quiz Attempted	comparing to the this to when a new activity is made this can possibly show us how long students take to complete quizes.
Quiz report viewed	Would a valuable variable to tell us how long students are spending to understand their grade: possibly telling us when students are trying to actively change their grades.
Event Context	**Use**
"File: "	These unique variables tell us what file was viewed. We can can accurately tell what topics student are finding most difficult, by how much time they spend, especially if Lecturers make lecturer slides with similar difficulty and length
"Quiz: "	This variable paired with the Quiz report viewed, can identify the difficult quizzes.

Table 4.2: Summary of Moodle Student logs

5 | Solution

5.1 Masking

The only table of interest which included student names was the userfullname column. Using Edgar's Test Course data I implemented a simple key encryption function in R. See code in appendix B.1.1. Here, we generate a key using the CRAN sodium package [9]. First it generates a key, which you can save using writeBin(), A key can also be given and read in using the readBin() function.

Using the `as.factor()` methods we turn the vector into an ordered list, where R knows what unique variables are associated with the list. We will use these unique encrypted variables and then apply it to the full dataset. The function `encrypt_name()` is used. This a function used to encapsulate some base R and sodium functions. We apply this to the levels and then use the function `recode(data,!!!encrypted)`. The '!!!' will unquote and splice, which is necessary to connect the vectors.

See appendix B.1.2. In order for the decryption, it requires to save the translation table using R's `writeRDS()`. You run code to read this table in. You then make a new translation table, by filling the titles of the old names as the encrypted names, and take the elements in each column as the actual name. Like before, run 'recode(...,!!!old_names)' in order to unquote and splice back to the actual list of names.

Edgar had three students under the Moodle umbrella, where we were initially asked to develop masking methods. We ultimately decided to go further with George Mican's masking data. This was decided by all three of us, incase we wanted to compare our findings together.

5.2 Cleaning

5.2.1 Cleaning Time:

As discussed the time variable is stored as a character variable. This needs to be changed in order to make better statistical methods.

We first take the time variable, then use the strptime method to convert it into a POSIXlt variable, see appendix B.2.1. This function takes the string, one of the parameters you have to specify the format of the string you have given it. Here you get, "Day/Month/Year. Hour: Minute: Second", using the '%'. Due to the year being given in a '/24', for 2024 format, we have added

2000 years on. This will allow us to plot and, potentially, use this as predictive variable.

5.3 Answering Questions

5.3.1 Actions Per Day

One of the question I wanted to solve was what times are students more active, that is done in appendix B.3.1. We load the data in using and filter out every `eventcontext` with "File: Lab ..." and "Quiz: ...". This will give us an idea when students are the most active beside lab-time. We can go even further and see when students are more active during the day. By using the code in appendix B.3.2, in the first plot we can see why we ignored Tuesdays and Thursdays. A simple analysis of figure 5.1 tells us that the more active days are Tuesday and Thursday. Where the mean number of times used far exceed others.

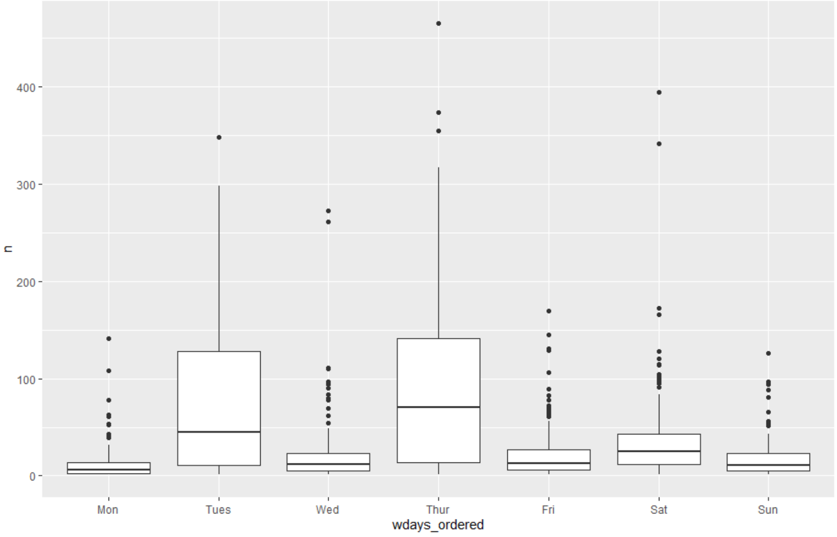
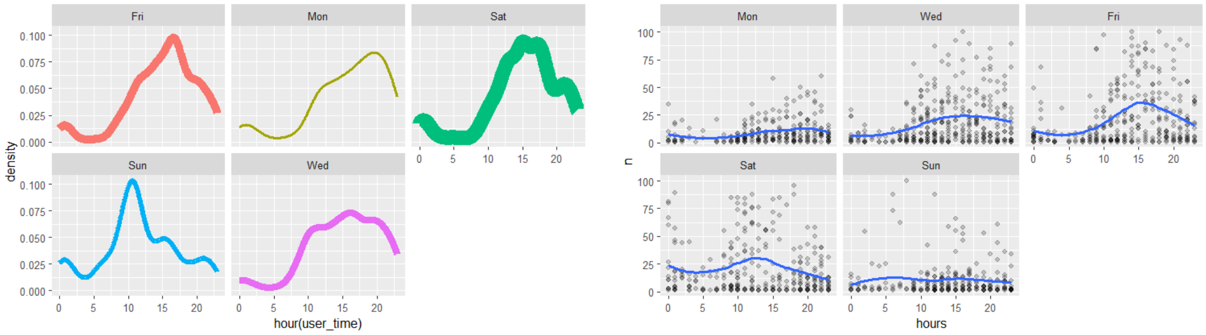


Figure 5.1: Actions Per Each Day

The idea to get look at the plots without Tuesday and Thursday, are confounded by figure 5.1 in appendix B.3.2. In figure B.1, we can see that actions start peaking around 15:00 - 17:00, possibly, around the lab times. People also seem to start performing actions around this time, possibly to prep for the lab.

Using code in appendix B.3.2 we can see that there seems to be two modes. Something to be concerned about is that people are performing actions through out the night, all the way up to 0:00-01:00. Actions peaks around 15:00-17:00 for Monday, Wednesday, Friday and Saturday; whereas on Sunday it peaks around 10:00. Figure 5.2b, comes from code in appendix B.3.2. This is the same as the 5.2a, however we are using the average use of Moodle from counting the amount of times students logged in.



(a) Daily Density Plot (Tuesdays and Thursdays)

(b) Average Use Plot (Other Days)

Figure 5.2: Density plots of student activity on Moodle.

5.4 More about the Data:

5.4.1 Activity Overtime

This section mostly relies on kernel density estimation, previously described in the background. When running it we want to pick a bandwidth that encapsulates two main characteristics of the data, that being the weekly trends and the study breaks. The base R, bandwidth estimator was originally created in MASS which was taken from S.J and Jones book [11]. Due to the base R function describing those two characteristics well we will continue using it.

This module is for the 1st semester of the school year. So we can see that being true in the trend in figure B.2a. This is Makes sense to how the module is structured, where it starts in October ending at the end of January. Looking at the data, looks like some people might have repeated, as there is bumps in July to October. In the the second figure B.2b we look at times before February 2024 in order to get a better idea of the data.

In appendix B.3.3 is code for the density plot of student activity without lab and quiz data, figure B.3. One of the many questions I had was if there is any dips in student retention after long breaks., for example the Halloween break. In this plot it seems to not reach the same peaks as pre-Halloween; but still quite as high as before. And then peaking before the midway point of January. Presumably right before class.

Furthermore, rather than creating a density plot of counting the use per day, it would be more accurate to get an average of it, you can see this in the appendix B.4. What is interesting here is that since we know of the lab times, we can see two spikes every week. Figure 5.3, is a the combined plot of the trend as orange line and the actual as grey. This more clearly describes what has been described before.

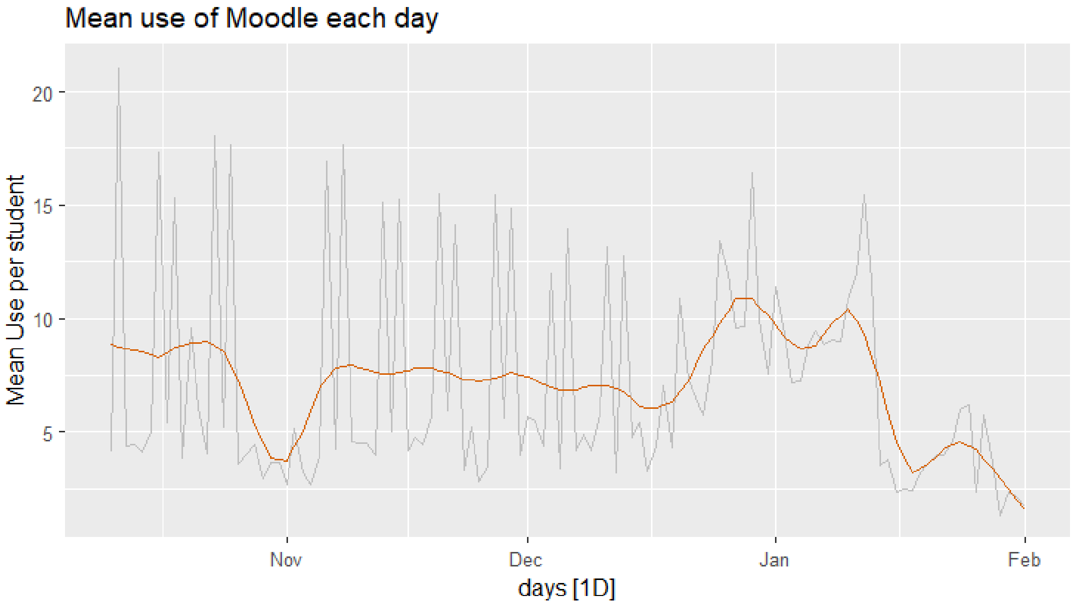


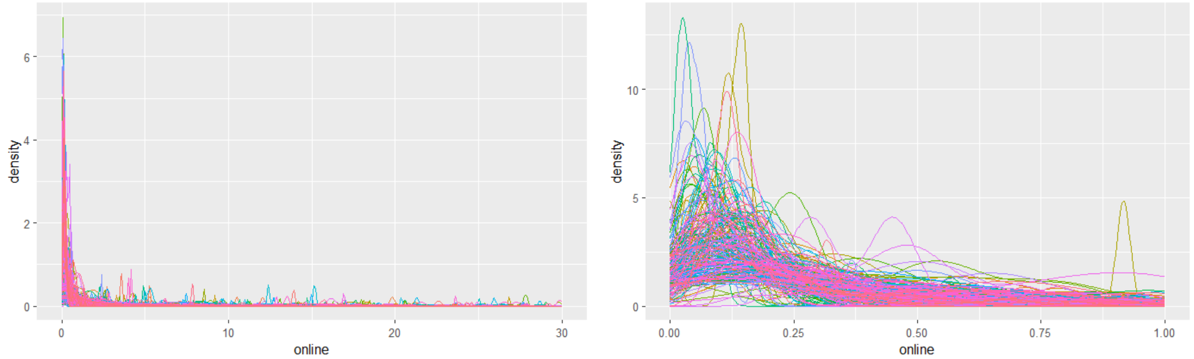
Figure 5.3: Time Series Decomposition of the Mean Use per Day

5.4.2 Minor Time Series Analysis

See appendix B.3.4, or look at the figure 5.3 seen above. Where the main data `json_data.11` mutated the `user_time` variable into days - "DD-MM-YYYY". We want to look at the data without the lab/quiz times, as we have seen them before. We first group by students and days, then using the `count()` function to get a count of use per student and day. Then using the `summarise()` function to get the mean, aggregating the student data: giving us the average amount of student use per day. We then use the `STL()` function to describe the Mean use per day using a seasonal, trend and error model. If we separate the cyclic lab group from the data we can see a trend emerge seeing that the mean use of students continue to be the same, and spiking during the exam period. What's interesting to see is that it seemingly drops 2 days after the peak. This is possibly, telling us that the exam is within these two days..

5.4.3 Investigation into Lagging Data

In this section we will talk about, ultimately why the time spent data was dropped. See appendix B.3.6. In the code, the function `diff_more` is primarily used to find the difference between the current and next event. The function then returns a vector of results with 0 added on at the end. Firstly take not that we are grouping each the data by each user and each day - since we don't know how students logout of Moodle - we assume that they are turned off. In the data there is no `eventcontext` or `eventname` telling us student quit Moodle so must assume as much. Expanding on this idea, people don't just log off at then end of the day, they tend to log off after they have finished working on the material. Would that justify getting rid of large difference? Ignoring these ideas we can get the following plots by filtering for certain file-names - those that would matter more to the lecturer.



(a) less than 30 mins

(b) less than 1 mins

In plot 5.4a we can see that there are only a handful of students going above the 10 minute mark with a few spike randomly on the line. What more noticeable, that most people would spend less than 0.25 minutes on these files. These plots are completely nonsensical and don't make sense for student viewing course material. Therefore we can assume that this data is more than likely student clicking the material - it opening into a new tab; which is hosted by another service. And clicking into the quiz to answer questions related to the material. For these above reasons, I feel like dropping the lagged data might be for the best.

5.5 Masked Continuous Assessment Data

5.5.1 Cleaning the CA data

The biggest issue with the CA data is that some data points are missing, so for some labs it will contain missing data - this could be due to students missing labs. Since we don't have data on how many student miss labs with permissions we will just grade the missing labs as 0. See appendix B.3.5, this is done in the `lapply()` function. In the function, we scale all the grades from 0 to 100, in order to do this we need to collected the `max()` function, we have to account for the missing values so we turn on `na.rm = TRUE`. After scaling the missing data will still follow through, these are found using `res[is.na(res)]`. As mention before, they are turned into 0 values.

After Standardizing we want to collect information about what was the average grade collected for each quiz. This is done for each row, meaning that it is done for each user. These two tables are combined using the user variable, `userfullname`.

After joining the masked CA data and the amount of daily moodle uses not including lab time. We are left with a table of users, average of daily uses, and our scaled grades per week. To make the data points more meaningful they have to aggregated, which is done in the `freq_bands` line. This cuts the average frequency uses into groups of `band_size`, in this case 20. We can look further, but having it at 20 seems to be as meaningful right now. Finally each user is grouped using the frequency bands and the average of each quiz score is found. This method can be used to hide student data, in future use as well. In order to visualize well, we use a parallel coordinate plot from the package `ggparcoord`. Assigning the parameter `scale = "globalminmax"`, to still keep the scaled grades values.

Using joined dataset we make bands of frequencies using the `cut()` function. After plotting with the frequency band, figure B.3.5, we can see that there are some differences between people who would use Moodle the most and the least. Due to there only being a few students in the 80-100 use range they are the ones most affected by outliers, so I have included the line width as how confident the values are. What we can see is that the average follows the same trends. They follow the same dips fro quiz EP, to quiz w6, quiz w7 and so on so forth. Furthermore the lower ends of the frequency bands tends to vary the mostly, as well as dip the most.

5.5.2 Combining Masked Assessment Data and Retention of Lecture Material

Similar to the previous count data we collect per student how much times they have used the lecture material, this is stored in the variable `uses_n_grade`. Unfortunately it would better to get retention, how much time people spend on average on material; however as discussed before it would not work. Now putting the use amount and ca data from the code shown in the appendixB.3.6, and referred to the image 5.5.

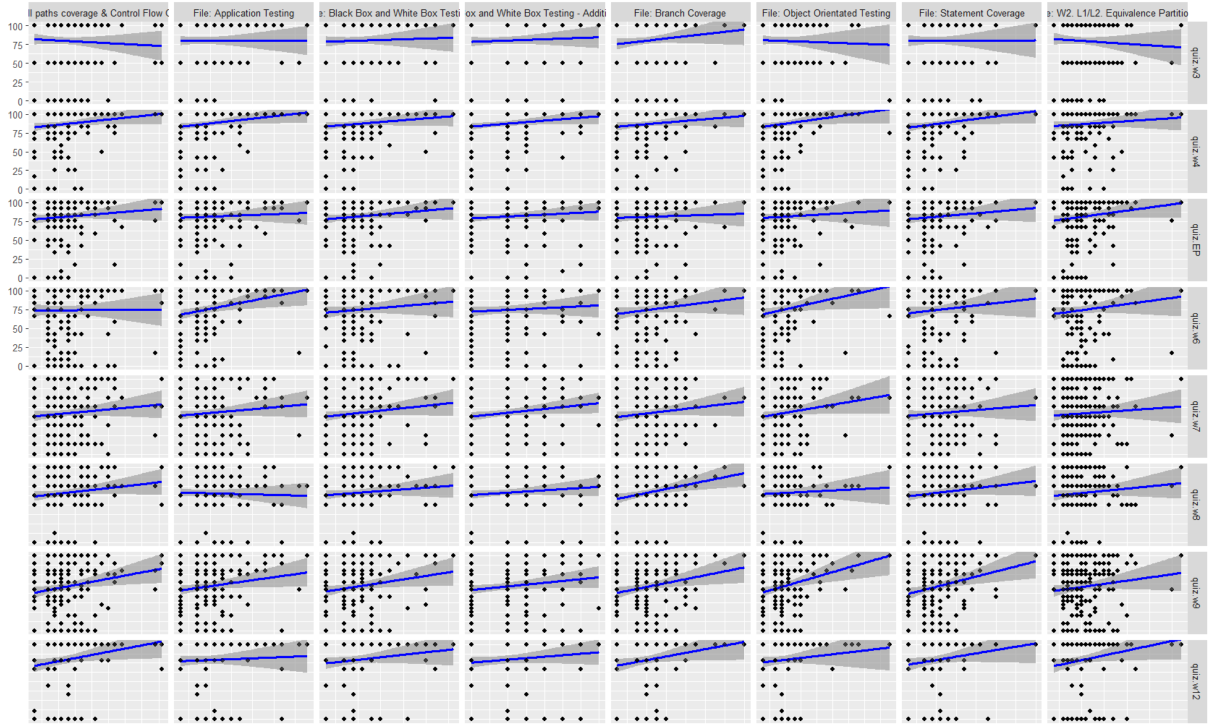


Figure 5.5: Use of lecture material and quiz grade

Every Column is a lecture material and each row is a quiz. Here the blue line is a simple linear regression fitting the n use of material and the grade given. These modelling techniques were discussed previously in the background section.

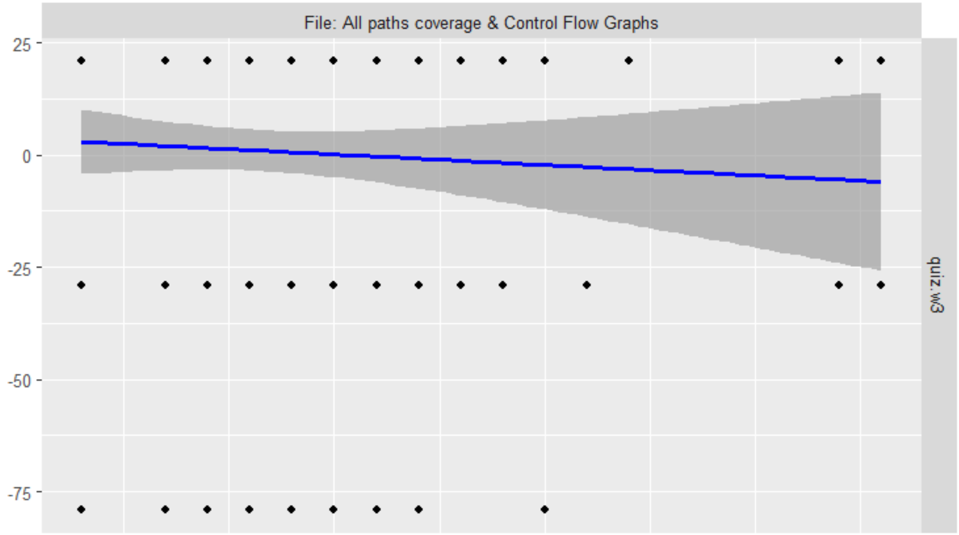


Figure 5.6: Use of lecture material and quiz grade

For example, we will take figure 5.6. Here we can see the scaling and mean centring of y - `quiz.w3` and x - `File:  All paths coverage & Control Flow Graphs`. We can see from the correlation line, that there is no significant change of week 3 grades from the count use of the file. For further clarification, the grey line area describes our confidence of where the true effect lies.

6 | Conclusion

6.1 Results discussion

Regarding the data we have, there are many interesting characteristics that Lecturers can use to improve their teaching.

From the actions per day we can see at points through out the week students are most active.

For the future we can compare it with the retention overtime, we can ask questions like did the the average use per day fall, compared to last year - did it rise proportionally with Maynooth's increase in students. Lecturer's can see themselves, if student engagement with their material and labs dropped after the mid-term break. Although limited, they can see if there is any engagement differences before the exam period. Problems with n amount data not being that usefull would show here.

From the parallel coordinate plot B.3.5, can easily learn their students weakest material.

6.2 Project Approach

The approach to the project was, to investigating what variables are the most important and seeing if there as any statistical methods that find interesting characteristics of the data. Then exploring if there was reasons why there is this factor.

6.3 Future Work

I believe that an application can be made in-order to give lecturer better understanding of student trends. They can learn when students tend to be the most active in order to send messages etc...

More importantly this data can be used in conjunction with IT services in-order to more efficiently use Moodle servers. In this we can tell when students are most active. What times on the weekends there might be spikes. ways to improve data collection.

More time spent data can be collected if we host files via Moodle, and we take note of scrolling - on the material files. A smart retention tracker can be made for lecturers in order to predict difficulty of the material. If students are spending more time and still getting bad grades, it is worth noticing.

For the time series decomposition, although it is more meaningful now to see the trend with the affect of the labs. As this will show that the student's personal engagement is still present. However this analysis from figure B.5 has a weekly periodic effect from October to February. This shouldn't be the case due to the labs not being on during the last week of the break and during January. A fix for this would be to bring the seasonal affect to 0. Bringing the original mean use plot, `meanPday`, after December 20th - due to it being Christmas break - as the trend plot.

A | The Problem chapters appendix

A.1 Information on Masked Continuous Assessment data

Creates a table of the amount of students missing per lab. These missing students are denoted as '-' in the Continuous data.

```r
masked_ca[,grepl("^Attendance", colnames(masked_ca))] -> missing_lab_data
missing_lab_data <- lapply(missing_lab_data, function(column) {
  column[column == "-"] <- NA
  column[column == "100"] <- 1
  return(column)
})

lapply(missing_lab_data, function(x) data.frame(missing=sum(is.na(x)), total=
  length(x), propmiss=sum(is.na(x))/length(x))) -> list_of_dataframes
bind_rows(list_of_dataframes, .id = "column_label")
```

B Solution Appendix

B.1 Masking

B.1.1 Encrypting

Code used to mask the sample Moodle data

```r
key <- sodium::keygen()
# key2 <- readBin("secret.key", what = "raw", n = 32)

encrypt_name <- function(name, key) {
  raw_name <- charToRaw(name)
  encrypted_name <- sodium::data_encrypt(raw_name, key)
  return(base64enc::base64encode(encrypted_name))
}

sampleuserfullname <- as.factor(sample$userfullname)
new_names <- sapply(levels(sample$userfullname), encrypt_name, key)
#Changes the levels of userfullname into an encrypted vector
NAMES <- recode(sample$userfullname, !!!new_names) # Encrypted Data
```

B.1.2 Decrypting

This code here is designed to unmask the encrypted data.

```r
new_names <- readRDS(new_names)
## Decrypting
old_names <- names(new_names)
  names(old_names) <- new_names

# Export Names and BtoA
# ReadIn Names, and use BtoA

old_data <- recode(NAMES, !!!old_names)
```

B.2 Data Cleaning

B.2.1 Time Variable

This code turn the character Time column into a universal time variable.

```
  json_data.11$user_time <- strptime(json_data.11$time,
                              "%d/%m/%Y, %H:%M:%S",
                              tz = "GMT") +
  years(2000)
```

B.3 Answering General Questions

B.3.1 Actions per Day

Here we summarise the json data to get the number of uses per day.

```
test2.1 <- json_data.11 |> filter(!grepl("^(File: Lab|Quiz:)", eventcontext)) |>
        mutate(wdays = days_name[wday(user_time)]) %>%
        group_by(userfullname, wdays) %>%
        summarise(n = length(user_time))
test2.1$wdays_ordered <- factor(test2.1$wdays, levels = c("Mon", "Tues", "Wed",
    "Thur", "Fri", "Sat", "Sun"))


ggplot(test2.1 %>% filter(n <= 500), aes(wdays_ordered, n)) +
geom_boxplot()
```

B.3.2 Density Plot of when students use Moodle

This code generates density plots of when students log in to the Software Testing Moodle Page

```
test2.14 <- json_data.11 |> filter(!grepl("^(File: Lab|Quiz:)", eventcontext))
    |>
  mutate(wdays = days_name[wday(user_time)])

test2.14$wdaysd <- factor(test2.14$wdays,
              levels = c("Mon", "Tues", "Wed", "Thur", "Fri", "Sat", "Sun"))

n_use <- test2.14 |> group_by(wdays) |> count() |> mutate(n = n/63270)
n_uses <- scale(as.matrix(n_use$n), center = FALSE)
rownames(n_uses) <- n_use$wdays

ggplot(test2.14 |> filter(wdays == days_name[c(1,3,5,6,7)]),
                    aes(x = hour(user_time), colour = wdays)) +
  geom_density(aes(linewidth = n_uses[wdays,])) +
  facet_wrap(~wdays)
```

Plot Created by code below.

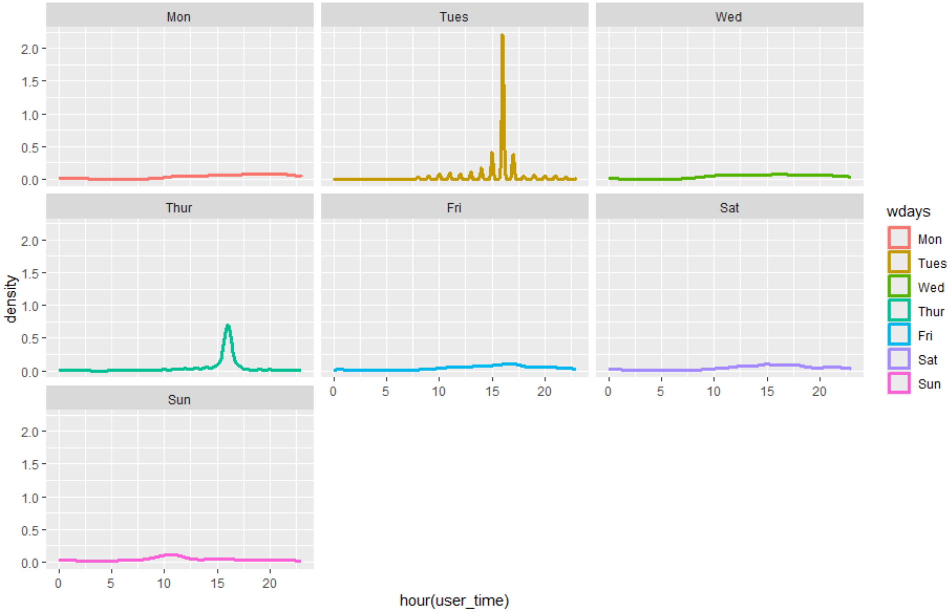


Figure B.1: When students login, to the hour

A measure of average use per hour.

```
test2.14 <- json_data.11 |> filter(!grepl("^(File: Lab|Quiz:)", eventcontext))
  |>
mutate(wdays = days_name[wday(user_time)])

test2.14$wdaysd <- factor(test2.14$wdays,
levels = c("Mon", "Tues", "Wed", "Thur", "Fri", "Sat", "Sun"))

n_use <- test2.14 |> group_by(wdays) |> count() |> mutate(n = n/63270)
n_uses <- scale(as.matrix(n_use$n), center = FALSE)
rownames(n_uses) <- n_use$wdays

ggplot(test2.14 |> filter(wdays == days_name[c(1,3,5,6,7)]),
aes(x = hour(user_time), colour = wdays)) +
facet_wrap(~wdays) +
geom_density(aes(linewidth = n_uses[wdays,])) +
theme(legend.position = "FALSE")
```

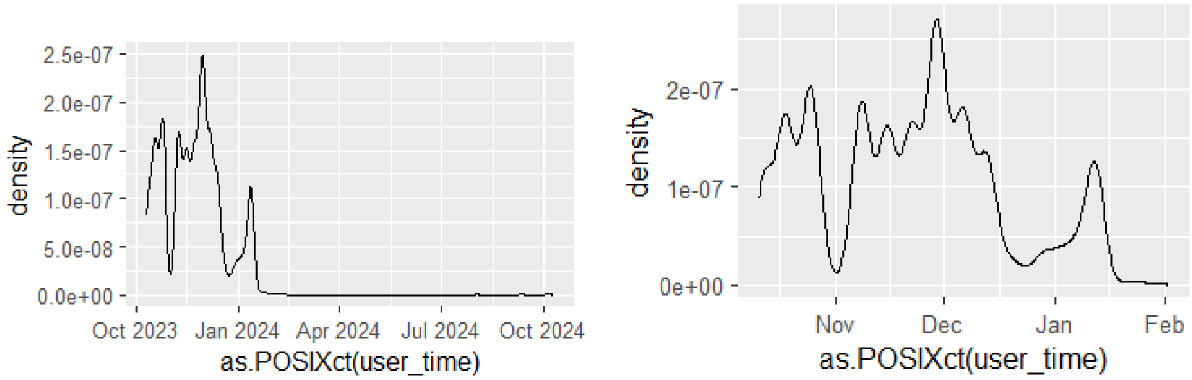
B.3.3 Overall Plot of Activity

This code is plots a density graph of when people log on to Moodle:

```
json_data.11 %>% mutate(days = as.Date(user_time)) %>%
 ggplot(aes(x = as.POSIXct(user_time))) +
 geom_density()
```

To add further to the previous plots we will get rid of lab/quiz data:

```
json_data.11 %>% mutate(days = as.Date(user_time)) %>%
  filter(days <= as.Date("01/02/24", "%d/%m/%y")) %>%
```



(a) Figure 1

(b) Figure 2

Figure B.2: Density plot of student logs over the year

```
  filter(!grepl("^(File: Lab|Quiz:)", eventcontext)) %>%
  ggplot(aes(x = as.POSIXct(user_time))) +
  geom_density()
```

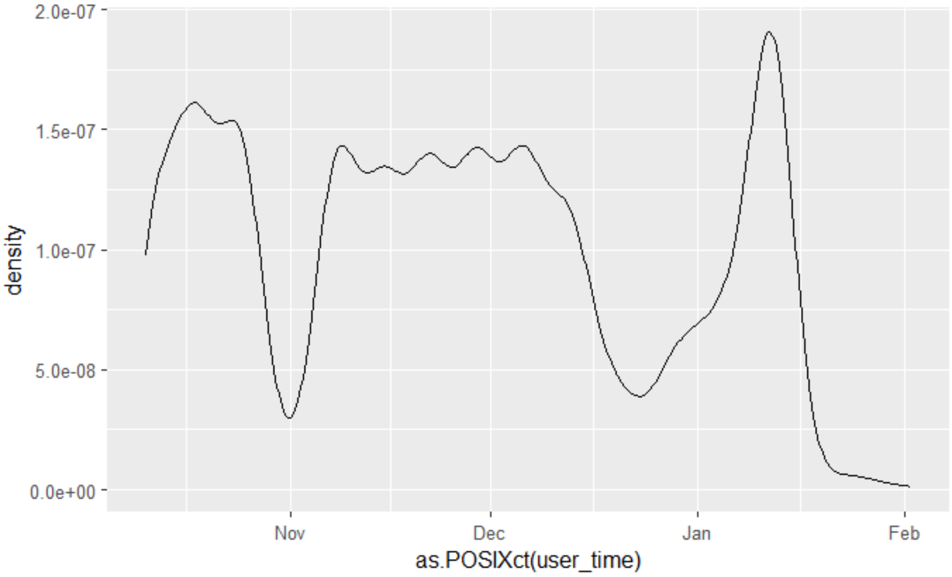


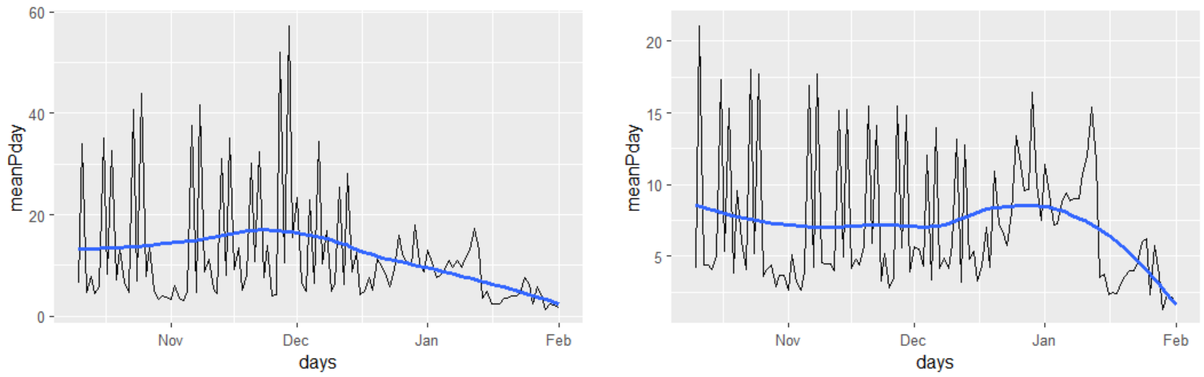
Figure B.3: Figure 1

B.3.4 Plot of the Average uses per day

This is the code for generating and plotting the mean amount of uses per student. The two plots are the plots with and without labs/quizzes.

```
json_data.11 |> mutate(days = as.Date(user_time)) %>%
  filter(days <= as.Date("01/02/24", "%d/%m/%y")) %>%
  filter(!grepl("^(File: Lab|Quiz:)", eventcontext)) %>%
  group_by(userfullname, days) %>% count() -> use_per_day

use_per_day %>% group_by(days) %>% summarise(meanPday = mean(n)) %>%
  ggplot(aes(x = days, y = meanPday)) +
  geom_line() + geom_smooth(se = FALSE, method = "loess")
```



(a) Figure 1

(b) Figure 2

Figure B.4: Density plot of student logs over the year

This code will create the data we can run a time series on. This code generates the STL decomposition, figure B.5, plot and the combined trend and mean use per day plot, from the solutions, at figure 5.3.

```
json_data.11 |> mutate(days = as.Date(user_time)) %>%
  filter(days <= as.Date("01/02/24", "%d/%m/%y")) %>%
  filter(!grepl("^(File: Lab|Quiz:)", eventcontext)) %>%
  group_by(userfullname, days) %>% count() -> use_per_day

use_per_day %>% group_by(days) %>% summarise(meanPday = mean(n)) -> time_series_anl

time_series_anl %>% as_tsibble() %>%
model(STL(meanPday ~ trend(window = 7) + season(window = "periodic"), robust = TRUE)) %>%
components() %>% autoplot() # STL decomposition plot

dcmp <- time_series_anl |> as_tsibble() |>
  model(stl = STL(meanPday))

components(dcmp) |>
  as_tsibble() |>
  autoplot(meanPday, colour="gray") +
  geom_line(aes(y=trend), colour = "#D55E00") +
  labs(
  y = "Mean Use per student",
  title = "Mean use of Moodle each day"
)
```

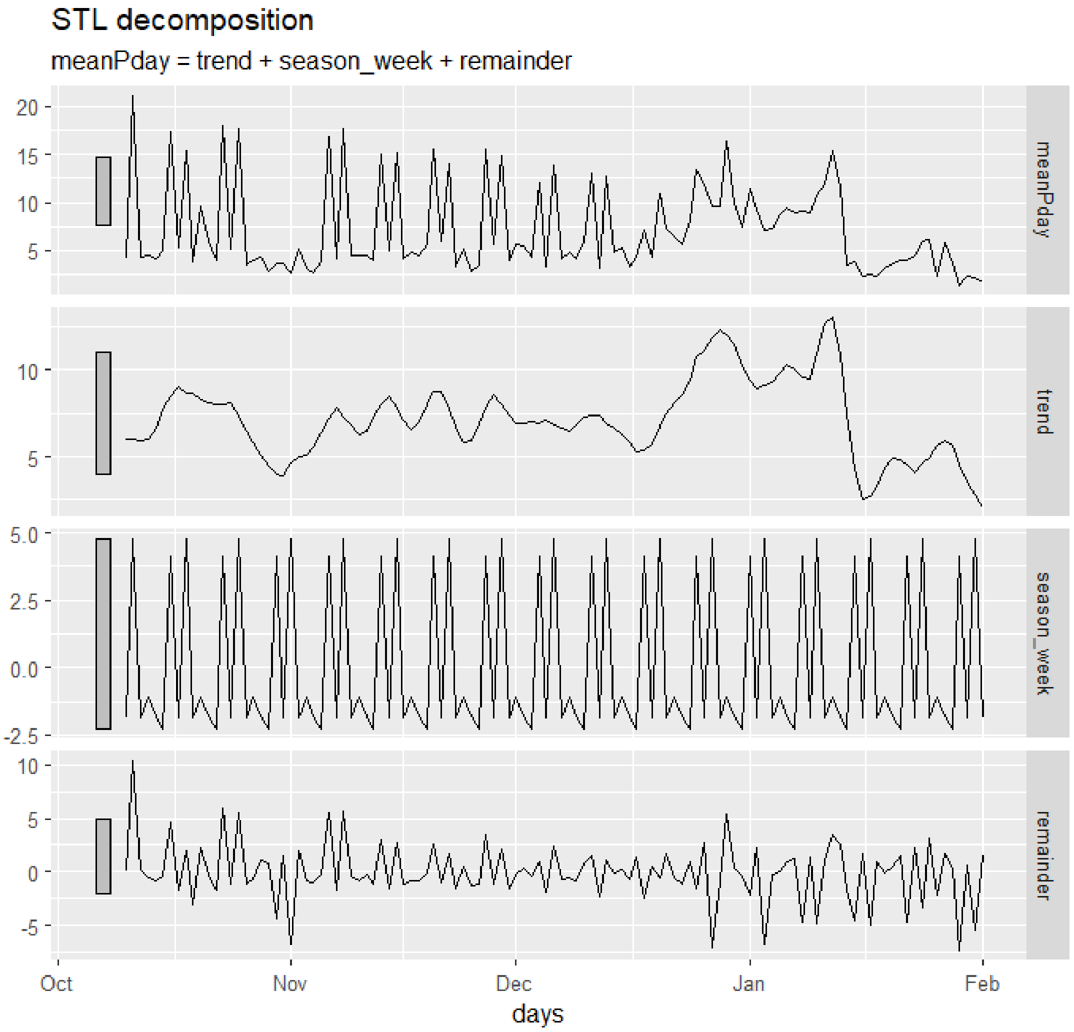


Figure B.5: Time Series Decomposition of the Mean Use per Day

B.3.5 Combining Average Grade Data and Weekly Data

Plot below is the plot of the average grade per week with each line sa a n'amount of events logged on Moodle.

```
test_names2.2.2[,-9] <- lapply(test_names2.2.2[,-9], function(.) {
  max_per_coSl <- max(., na.rm = TRUE)
  res <- (./max_per_col)*100
  res[is.na(res)] <- 0
  return(res)
}) # Standardizing results

test2.2.3 <- json_data.11 %>% mutate(days = as.Date(user_time)) %>%
filter(!grepl("^(File: Lab|Quiz:)", eventcontext)) %>%
group_by(userfullname, days) %>% count()

test2.2.2.1 <- inner_join(test_names2.2.2, test2.2.3, by = "userfullname")

```

```
test2.2.2.1$week <- factor(test2.2.2.1$`week(days)`, levels = c(36:52, 1:35))
band_size <- 20
test2.2.2.1$freq_bands <- cut(test2.2.2.1$n, breaks = seq(0, max(test2.2.2.1$n
  ), by = band_size), include.lowest = TRUE)

test2.2.2.1 %>% filter(n <= 100) %>%
group_by(freq_bands) %>%
summarise_at(vars("quiz.w3", "quiz.w4", "quiz.EP", "quiz.w6", "quiz.w7", "quiz
  .w8", "quiz.w9", "quiz.w12"), mean) -> ggparcoordplot
ggparcoordplot %>%
ggparcoord(columns = 2:9, groupColumn = "freq_bands", scale = "globalminmax")
```

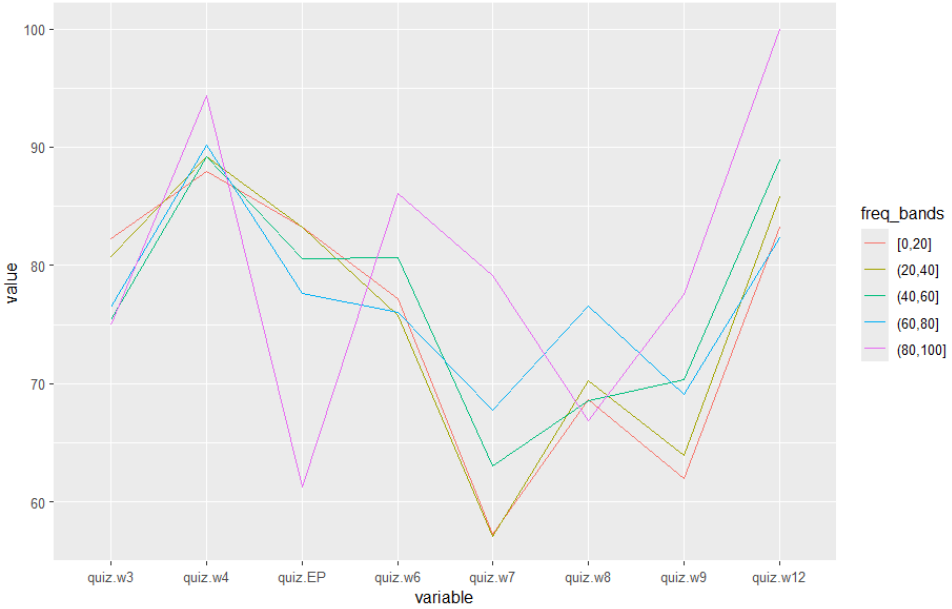


Figure B.6: Average Grade per week, with freq of use included

B.3.6 Time spent on course material

This code is a function to calculate time spent on each event.

```
diff_more <- function(x) {
  x1 <- x[1:length(x)-1]
  x2 <- x[2:length(x)]

  return(c(difftime(x1, x2, units = "mins"),0))
}

test_set <- json_data.11 |> group_by(userfullname, as.Date(user_time)) |>
  mutate(online = diff_more(user_time))
just.looking <- test_set |> select(eventcontext, user_time, userfullname,
  online)
pattern <- paste(
  c(
    "File: Notes during lectures v2",
    "W2\\. L1/L2\\. Equivalence Partitions",
  "2022 CS265 Exam Paper",
  "Application Testing",
  "Object Orientated Testing",
  "Black Box and White Box Testing",
  "All paths coverage & Control Flow Graphs",
  "Branch Coverage",
  "Statement Coverage",
  "Decision Table Testing",
  "BVA"
 ),
 collapse = "|"
)

ggplot(test_set |> filter(grepl(pattern, eventcontext) & online <= 1),
aes(x = online, colour = userfullname, alpha = 0.05)) +
geom_density() + theme(legend.position="none")
```

Below code generates the image 5.5.

```
pattern2 <- paste(
c(
 "W2\\. L1/L2\\. Equivalence Partitions",
 "Application Testing",
 "Object Orientated Testing",
 "Black Box and White Box Testing",
 "All paths coverage & Control Flow Graphs",
 "Branch Coverage",
 "Statement Coverage"
 ),
 collapse = "|")

json_data.11  |> group_by(userfullname) |>
filter(grepl(pattern2, eventcontext)) |> group_by(eventcontext, userfullname)
  |> summarise(use_n = n()) -> uses_n_grade
uses_n_grade |> pivot_wider(names_from = eventcontext, values_from = use_n) ->
   uses_n_grade
uses_n_grade <- uses_n_grade %>% replace(is.na(.), -1)
uses_n_grade <- inner_join(uses_n_grade, test_names2.2.2)
#plot_data <- uses_n_grade[,-1]
ggpairs(uses_n_grade[,-1], lower = list(continuous = "smooth")) +
geom_smooth(color = "blue", method = "lm") +
geom_point(alpha = 0.4) -> plot

uses_omit_grade <- na.omit(uses_n_grade)
uses_omit_grade <- inner_join(uses_omit_grade, test_names2.2.2)
ggpairs(uses_omit_grade[,-1], lower = list(continuous = "smooth")) +
geom_smooth(color = "blue", method = "lm") +
geom_point(alpha = 0.4) -> plotB
plotB$plots <- plot$plots[sort(rep(16*(8:15), 8)) + c(1:8)]
plotB$yAxisLabels <- plot$yAxisLabels[9:16]
plotB$xAxisLabels <- plot$xAxisLabels[1:8]
plotB
```

B.4 Evidence Supporting Use of LLM

LLM's have been used sparingly for the project, due it being applying knowledge learned from different Statistics Modules that studied in the last few years. The only time I have been using LLM's is to write this paper. Dr. Edgar G. advised me to learn LaTex as it will be necessary when continuing in academia. I have briefly used it before, having the application TexStudio installed and pre-requisites for it installed before. Since I don't have much experience I have been using AI to understand LaTex. Mostly has been using Bing AI - copilot, but now switched to Deepseek.

B.4.1 Prompts and Responses

In order to link references as a .bib file, exported from the application Zotero, AI had to be used to understand it.

For the technical background section, for the use of KDE's I used AI to convert the formula on wikipedia into mathematical expression in LaTex:

```
{\displaystyle {\widehat {f}}_{h}(x)={\frac {1}{n}}\sum _{i=1}^{n}K_{h}(x-x_{i})={\frac {1}{nh}}\sum _{i=1}^{n}K{\Big (}{\frac {x-x_{i}}{h}}{\Big )},}
can you right this is math notation in Latex. And describe how I can do it on my Latex File
```

Where Deepseek responded with

```
\widehat{f}_{h}(x) = \frac{1}{n} \sum_{i=1}^{n} K_{h}(x - x_{i}) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_{i}}{h}\right)
```

In order to match a .docx template given to us by the course co-ordinator I used AI to fix a contents table error I was havign

```
How to add a 0 chapter for a decleration of no plagarism and an acknowledgement section
```

Where Deepseek responded with:

```
...
% Chapter 0: Acknowledgement
\cleardoublepage % Ensure it starts on a new page
\chapter*{Acknowledgement}
\addcontentsline{toc}{chapter}{Acknowledgement} % Add to table of contents
...
```

Bibliography

[1] *About Moodle - MoodleDocs*. url: `https://docs.moodle.org/405/en/About_Moodle` (visited on 02/19/2025).

[2] Brijesh Baradwaj and Saurabh Pal. “Mining Educational Data to Analyze Students’ Performance”. In: *International Journal of Advanced Computer Science and Applications* 2 (Oct. 16, 2011), pp. 63–69. doi: `10.14569/IJACSA.2011.020609`.

[3] François Bouchet et al. “Clustering and Profiling Students According to Their Interactions with an Intelligent Tutoring System Fostering Self-Regulated Learning”. In: *Journal of Educational Data Mining* 5 (May 1, 2013), pp. 104–146.

[4] *Create Elegant Data Visualisations Using the Grammar of Graphics*. url: `https://ggplot2.tidyverse.org/` (visited on 03/05/2025).

[5] *Forecasting: Principles and Practice (3rd Ed)*. url: `https://otexts.com/fpp3/` (visited on 03/09/2025).

[6] *Group by One or More Variables — Group_by*. url: `https://dplyr.tidyverse.org/reference/group_by.html` (visited on 03/06/2025).

[7] Henrik Madsen and Poul Thyregod. *Introduction to General and Generalized Linear Models*. Milton, UNITED KINGDOM: Taylor & Francis Group, 2010. isbn: 978-1-4398-9114-8. url: `http://ebookcentral.proquest.com/lib/nuim/detail.action?docID=1648247` (visited on 03/12/2025).

[8] Dr Leo Magno. “Using Facebook Metrics to Measure Student Engagement in Moodle”. In: *International Journal on Open and Distance e-Learning (IJODeL)* (Jan. 1, 2016). url: `https://www.academia.edu/33518638/Using_Facebook_Metrics_to_Measure_Student_Engagement_in_Moodle` (visited on 03/15/2025).

[9] Jeroen Ooms [aut and cre. *Sodium: A Modern and Easy-to-Use Crypto Library*. Version 1.4.0. Dec. 16, 2024. url: `https://cran.r-project.org/web/packages/sodium/index.html` (visited on 02/27/2025).

[10] Emanuel Parzen. “On Estimation of a Probability Density Function and Mode”. In: *The Annals of Mathematical Statistics* 33.3 (Sept. 1962), pp. 1065–1076. issn: 0003-4851, 2168-8990. doi: `10.1214/aoms/1177704472`. url: `https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-33/issue-3/On-Estimation-of-a-Probability-Density-Function-and-Mode/10.1214/aoms/1177704472.full` (visited on 03/07/2025).

[11] S. J. Sheather and M. C. Jones. "A Reliable Data-Based Bandwidth Selection Method for Kernel Density Estimation". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 53.3 (July 1, 1991), pp. 683–690. issn: 1369-7412, 1467-9868. doi: 10.1111/j.2517-6161.1991.tb01857.x. url: https://academic.oup.com/jrsssb/article/53/3/683/7028194 (visited on 03/07/2025).

[12] Bernard W. Silverman. *Density Estimation for Statistics and Data Analysis*. New York: Routledge, Feb. 19, 2018. 176 pp. isbn: 978-1-315-14091-9. doi: 10.1201/9781315140919.

[13] *Student Activity Details for Courses*. url: https://help.blackboard.com/Learn/Instructor/Ultra/Performance/Course_Reports/Course_Activity_Related_to_Grades/View_Individual_Student_Course_Activity_Data (visited on 02/20/2025).