# Data engineering project

Report on the MongoDB + OULAD dataset

**Put together by** **Kory Frankee** (fran0618)

**Class:** COMP 2031

**Page count (for pages that count towards the limit):** ~ 7

**Word count (for sections that count towards the limit):** ~ 2,048words

**Page limit:** 10

**Word limit:** 3,000 words

(Limits exclude my bibliography, references, figures, title page, and appendices)

## What I worked with

**R version** 4.3.3

**R Studio version** 2023.12.1

**Packages I used in one’s code and for what:**

* mongolite {for connecting to a Mongo DB}
* tidyverse
  + ggplot2 {for plotting}
  + tibble {for storing data}
  + dplyr {for data manipulation}
  + tidyr {for rectangling}
  + readr {for reading .csv files}
  + stringr {for working with strings}
  + forcats {for factoring data}
* R6 {for classes since I’m quite comfortable with object-oriented programming}
* sqldf {for SQL}. Note that dplyr *“aims to translate the most common R functions to their SQL equivalents”* and with it, you can even see the SQL query behind code written in dbplyr. I, however, opted to write SQL in some cases and dbplyr in other cases since I knew both
* colourvalues {for converting from an RGB value to a HEX value}

## What I’ve done

#### I. Loading the libraries in

I wrote instructions to check if the packages I want to use are installed and if not, to install them. I then loaded them so they’re available for use

#### II. Loading the data in

To help put the data into R, I developed a *“data loader”* class. **Within it, I wrote functions for:**

* Putting data retrieved from the account of a MongoDB user > from a cluster > from a MongoDB collection into a tibble
* Putting data from a .csv file into a tibble

#### III. Data observations

To help understand the data, I wrote a *“data checker”* class. Within it, I wrote functions for:

* Checking for duplicate primary key IDs
* Checking that the combination of multiple foreign key IDs was unique in each row
* Telling me basic facts about the dataset like the number of rows, columns, and duplicate, missing, and unique values, as well as the data types of the columns in the dataset

After loading in the data from the MongoDB dataset, I discovered that the scores were nested, which is a problem I’ll have to address

A screenshot of a graph

Description automatically generated

Of the 400,000 rows, there were 10,000 students and 501 classes with a unique ID, 3 assignment types (‘exam’, ‘homework’ and ‘quiz’), and 400,000 unique scores in the range of 0 to 100.

(Conclusion) The Mongo DB scores show that the mock data is unrealistic. (P1) Markers don’t tend to provide scores with upwards of 7 decimal places. (P2) Marks don’t tend to be all different to one another when the number of assignments marked is high, which they are. (P3) Marks tend to have a minimum of 0 as some students receive a 0 for not handing in their work. Here, the minimum score is ~ 0.0002. (P4) Marks tend to have a maximum of 100 as some students receive full marks for their work. Here, the maximum is ~ 99.99. (P5) Scores tend to have a normal distribution. I’ve provided a visualization of the distribution of scores for all the assignments to show how uniform it is **(Appendix 0)**

With that in mind, **I can still work with this data**

After looking at the data from the OULAD dataset, I had many problems with it.

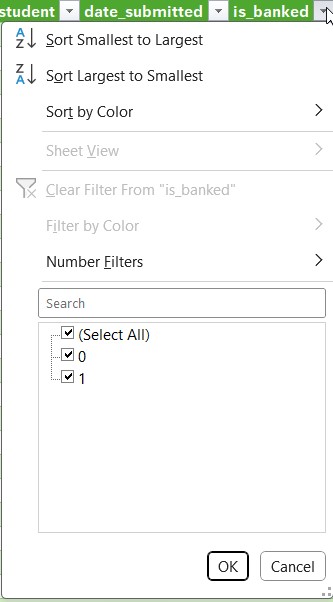
1. Missing student scores (should be 0 as the default mark is zero. Students tend to contact the marker if their mark is missing) *{student\_assessment.csv}*

A screenshot of a search box

Description automatically generatedA screenshot of a search box

Description automatically generated

1. Use of quantitative (‘0’/’1’ values) for if result is transferred from a previous presentation. I think nominal (‘no/’yes’ values) would be clearer and acceptable as no calculations are done on this column *{student\_assessment.csv}*



A screenshot of a computer

Description automatically generated

1. Column names that aren’t that clear to me at least as I had to look up their meaning on the OULAD website {all .csv files}
2. Abbreviation used (e.g. *‘No Formal quals’* could be expanded to *‘No recognized qualifications’*) *{student\_info.csv}.*
3. Wrongly entered age values (e.g. ‘55<=’ should be ‘>55’) *{student\_info.csv}*

A screenshot of a search box

Description automatically generated

A screenshot of a search box

Description automatically generated

1. Missing IMD band values and a wrongly entered IMD band value ’10-20’ with no ‘%’ sign at the end of it) *{student\_info.csv}*

A screenshot of a screenshot of a computer

Description automatically generated

A screenshot of a screenshot of a computer

Description automatically generated

1. Missing when student unregistered values *{student\_registration.csv}*

A screenshot of a search box

Description automatically generated

A screenshot of a search box

Description automatically generated

1. Missing when student registered values *{student\_registration.csv}*

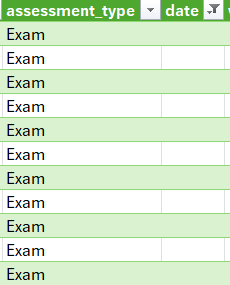
A screenshot of a computer

Description automatically generated

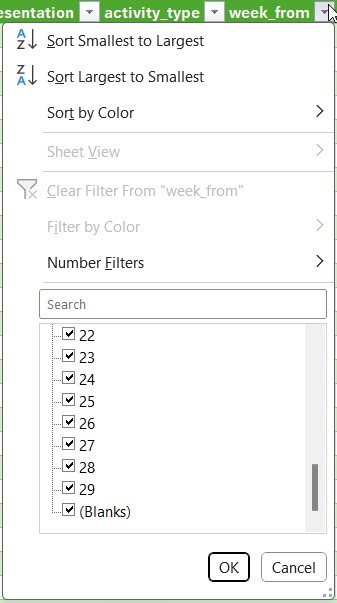
A screenshot of a computer

Description automatically generated

* Missing exam due dates (all these missing values should be the replaced with the last week of the course value) {*assessments.csv}*



1. Missing values for when students use VLE materials *{vle.csv}*



A screenshot of a search box

Description automatically generated

1. Missing values for when students stop using VLE materials *{vle.csv}*

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. The sum of all assessments for a module + presentation combination should add up to 100% and it doesn’t. **There are two reasons it doesn’t add up to 100%:**
   1. The assessments are weighted too low.

* As it is, a student could do well on every single assignment for a module + presentation combination and fail due to all their assessments being weighted very lowly, which is very counterintuitive

E.g. For the student with an ID of *‘465730’* doing module *‘BBB’* and presentation *‘2013J’*, I observed that this student had high scores for all their assessments but failed because the assessment weightings add up to ~ 47.91%, not 100%

A screenshot of a graph

Description automatically generated

* 1. Both the exams and all other assessments cannot be both be weighted at 100% for one case, which is the case of a student doing both an exam and other assessments for a module + presentation combination. This is because this would make grades out of 200% as opposed to 100%, which no university grading system accounts for
* As it is, a student can obtain a weighted score ≥ 101 and ≤ 200.

I decided against replacing *”Lower than A Level”* values with *“No recognized qualifications”* since this is a separate category that could, for example, be referring to UK students that completed BTEC *{student\_info.csv}*

#### IV. Data cleanup

I wanted to address the one problem I had with the MongoDB dataset and the 11 problems I had with the OULAD dataset. To help with this, I developed a *“data cleanup”* class. **Within it, I wrote functions for:**

* Unnesting a column
* Finding and replacing values
* Removing junk columns
* Putting values in a column into a vector of length 1 with comma-separated terms and quotation marks between each term. I can use such vectors in conjunction with a SQL IN
* Reordering columns (arrange())
* Setting the names of columns

I also wrote a function to work out the most common IMD value for each of the regions with missing values. I then filled in the missing IMD values for a particular region with the most common IMD value for that region.

How I made it so assessment weightings for a module + presentation combo added up to 100%

* Case 1) Some of the assessment weightings for the module + presentation combo
* Case 2) All of the assessment weightings for the module + presentation combo

As a result of this change, students that do well for all their assessments won’t fail unless they withdrew and obtained a WNF. This means that more students with a high average score will have a higher grade and higher cumulative GPA (**Appendix 06, 07 & 13).** Now there are no module and presentation combinations that **ALL** students fail **(Appendix 06)**

#### V. Data additions

*Everything I do in this section is based on what Australian universities do*

I added a grades column to the MongoDB dataset, which I filled in with tertiary grades (‘F’, ‘P’, ‘Cr’, ‘D’, and ‘HD’) based on what score students received for their assessments

For the OULAD dataset, I addressed the issue with the assessment weightings for the **problematic 2nd case**

**Cases 1) and 2):**

1. Module has assessments but no exams.

* For this case, assessment weights should altogether add up to 100%

1. Module has assessments, including exams.

* For this case, I added a second assessment weight column that scales all the assessment values to a new weighting. I set exams to be worth 40% (can change this value depending on the grading decisions of the university) and other assessments to be worth 100% - exam % so that they both add up to 100%

For both cases, I needed to add columns for the weighted score for each module, which will be equivalent to a grade (e.g. > 85% is a ‘HD’/’7’)

The weighted score (and grade) I decide to give a student is based on if their module had exams or not and if the student withdrew or not

**(Appendix 05)**

From here, I decided to calculate students’ cumulative GPA

#### VI. Data subsets and supersets

To help obtain a subset of the data, I wrote a “data subsetter” class with functions for:

* Getting a few rows of interest (filter())
* Getting a few columns of interest (select())

To help join data together, I wrote a “data supersetter” class with functions for:

* Merging tables

I joined the OULAD tables that I was able to join (without R Studio freezing) into one big table since I found that easier to work with

**(Appendix 04)**

Using a combination of subsetting and supersetting was helpful when, for example, I was working with the large *“student\_vle.csv”* file. I split it up into 8 parts, performed the operations I wanted to perform on each part, and then put the pieces of operated data together into a table

#### VII. Dataset querying

Sometimes I needed the dataset to be in a particular form and to help with this, I wrote a *“dataset querier”* class with functions for:

* Querying the data using SQL

#### VIII. Data analysis

To help analyse the data, I wrote a *“data analysis”* class with functions for:

* Working out statistical information (mean, mean of each group, median, maximum, minimum, quartile value, quartile value for each group, interquartile range, standard deviation, z-score, & range of values). **Here’s an example of what I’m calling the “mean of each group” where each group is a different class:**

A screenshot of a table

Description automatically generated

#### IX. Data visualization

To help visualize the data, I wrote a *“data visualization”* class with functions for:

* Getting the inverted colours for a list of colours (used for distinguishing the colours of labels from colours of the plot objects)
* **Graphing:**
  + Simple scatter plots
  + Complex scatter plots **(Appendix 01)**
  + Bar charts **(Appendix 02, 07)**
  + Bar charts side by side (**Appendix 03**, **06)**
  + Violin plots **(Appendix 08 to 11, 14)**
  + Histogram w/ density plots **(Appendix 0)**
  + Density plots **(Appendix 12)**
  + Pie charts
  + Line graphs
  + Box plot

#### X. Data modelling

##### Linear models

**Point of using one:**

Seeing if there’s a relationship between one or more variables and an outcome variable

**Seeing if two variables are related:**

* Can use correlation for this, it’s defined as

Correlation can range between -1 and 1. Higher positive numbers mean there’s a closer relationship. Lower negative numbers mean there’s an inverse relationship. Numbers near 0 mean that there’s no relationship

**Conditions for using one:**

* Continuous outcome variable
* Linear relationship
* Independent observations
* Residuals follow a normal distribution
* There should be no extreme outliers (the variance of the residuals should be the same). Might be able to find outliers using the outlierTest() function in the car package

**Interpreting one:**

* If the line has a positive slope, the predictor has a positive association with the outcome

**Considerations when using one:**

* Should factor categorical variables
* Want to find the set of predictors that best explain the outcome (find the best fitting model) whilst avoiding including so many predictors so that it fits the current data well but does not predict future data well

**Evaluating the performance of a model:**

* Looking at the root mean square error (RMSE)
* Looking at the difference between the actual value and the value the model predicts for a given observation

##### Simple linear model

A simple linear regression model has 1 outcome (y) and 1 explanatory variable (x)

**Equation for a simple linear model**

Simple linear model

**How well this model performed:**

**How I interpreted it:**

##### Multiple linear model

A multiple linear regression model has 1 outcome (y) and 2 or more explanatory variables (x)

**Equation for a multiple linear model**

Example of model where x consists of categorical variables with only a few possible outcomes

**How well this model performed:**

**How I interpreted it:**

Example of model where x consists of continuous variables that consist of a range of real numbers

**How well this model performed:**

**How I interpreted it:**

Example of model where x consists of categorical and continuous variables

**How well this model performed:**

**How I interpreted it:**

#### XI. Data findings

The OULAD dataset is more interesting to talk about, so I’ll be focusing on it for this section

Both genders score about as good as each other

A screenshot of a computer

Description automatically generated

Students without a disability have a slight edge over students with one

A screenshot of a computer

Description automatically generated

Students that are older have a slight edge over students that are younger

A screenshot of a computer

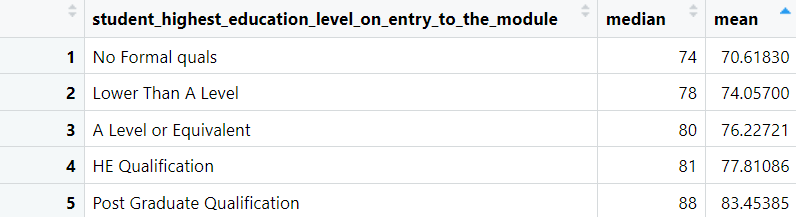
Description automatically generated

Students that are less impoverished have a slight edge over students that are more impoverished (two exceptions can be seen. The 10-20% band outdoes the 20-30% band and the 40-50% band outdoes the 50-60% IMD band)

A screenshot of a computer

Description automatically generated

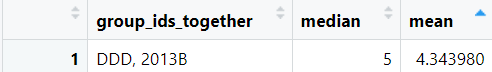
Students that are more well-education on entry have an edge over less educated students



Students perform the best in module + presentation combo *“EEE, 2014J”* and the worst in module + presentation combo *“DDD, 2013B”*

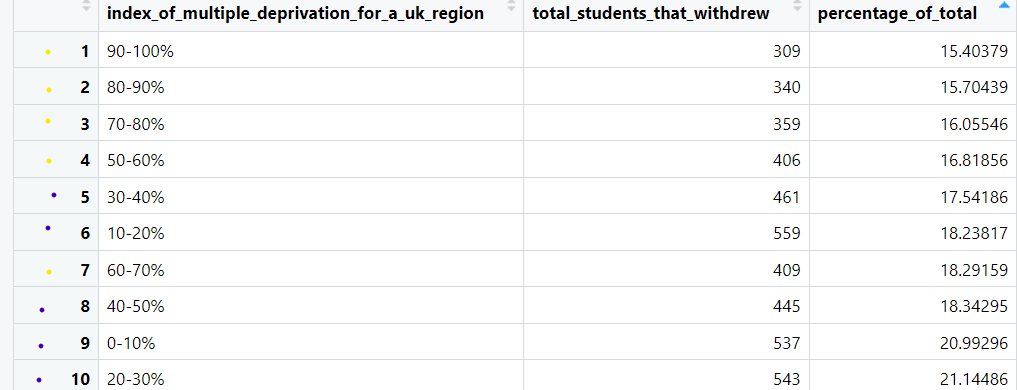
A close up of a word

Description automatically generated



~ 59.99% of students have a high enough {> 4.5} cumulative GPA for postgraduate studies

Students that are more impoverished are generally but not always likelier to withdraw



#### XII. Bibliography

**Sources I used to develop a deeper understanding of working with data and R in R Studio**

* Chang, W, 2024, *“Introduction to R6 classes”*, <https://r6.r-lib.org/articles/Introduction.html>, viewed March 2024
* Cotton, R 2013, *“Learning R”*, <http://duhi23.github.io/Analisis-de-datos/Cotton.pdf>, viewed March 2024
* Freeman, M 2019, *“Programming skills for data science: Start writing code to wrangle, analyze, and visualize data with R”*, viewed March 2024
* Grolemund, G 2014, *“Hands-On Programming with R”*, <https://rstudio-education.github.io/hopr/>, viewed March 2024.
* Huynh, H 2019, *“R for graduate students”*, <https://bookdown.org/yih\_huynh/Guide-to-R-Book/>, viewed March 2024
* IBM 2024, *“R Programming Basics for Data Science”*, <https://www.edx.org/learn/r-programming/ibm-r-programming-basics-for-data-science>, viewed March 2024
* Irizarry, R 2019, *“Introduction to Data Science: Data Analysis and Prediction Algorithms with R”*, <https://rafalab.dfci.harvard.edu/dsbook/>, viewed March 2024
* Ismay, C; Kim, A; McConville, K, 2024, *“Statistical Inference via Data Science A ModernDive into R and the Tidyverse”*, <<https://moderndive.com/>>, viewed April 2024
* Lander, J 2013, *“R for everyone: Advanced analytics and graphs”*, <https://www.researchgate.net/profile/David-Booth-7/post/Best\_approach\_for\_model\_selection/attachment/62588e5933d0f000f76d1352/AS%3A1144874569072657%401649970776919/download/R+for+Everyone+Advanced+Analytics+and+Graphics+by+Jared+P.+Lander+%28z-lib.org%29.pdf>, viewed March 2024
* Long, J & Teetor, P 2019, *“R Cookbook, 2nd Edition”*, <https://rc2e.com/>, viewed March 2024
* Matloff, N 2017, “The Art of R Programming: A Tour of Statistical Software Design”, <https://diytranscriptomics.com/Reading/files/The%20Art%20of%20R%20Programming.pdf>, viewed March 2024
* Mecklin, C 2023, *“STA 440/441 Notes (Mathematical Statistics I)”*, <<https://bookdown.org/cmecklin/sta440/>>, viewed April 2024
* Monkman, M, 2024, *“The Data Preparation Journey Finding Your Way With R”*, <https://bookdown.org/martin\_monkman/DataPreparationJourney/>, viewed April 2024
* Nahhas, R 2024, *“Introduction to Regression Methods for Public Health Using R”*, <https://www.bookdown.org/rwnahhas/RMPH/>, viewed March 2024
* Neth, H, 2024, *“Data Science for Psychologists”*, viewed April 2024, <https://bookdown.org/hneth/ds4psy/>
* Ooms, J 2018, *“Mongolite user manual”*, <https://jeroen.github.io/mongolite/>, viewed March 2024
* Peng, R, Kross, S & Anderson, B, 2020, *“Mastering software development in R”*, <<https://bookdown.org/rdpeng/RProgDA/>>, viewed April 2024
* Roback, P & Legler, J, 2021, *“Beyond Multiple Linear Regression Applied Generalized Linear Models and Multilevel Models in R”*, < <https://bookdown.org/roback/bookdown-BeyondMLR/>>, viewed April 2024
* Scott, J, 2021, *“Data Science in R: A Gentle introduction”*, <<https://bookdown.org/jgscott/DSGI/>>, viewed April 2024
* Shang, Y, 2022, *“Making Sense of Data with R”*, viewed April 2024, <<https://bookdown.org/yshang/book/>>
* Wickham, H 2014, *“Advanced R”*, <https://adv-r.hadley.nz/index.html>, viewed March 2024.
* Wickham, H; Rundel, M; Grolemund, G 2023, *“R for data science (2e)”*, <https://r4ds.hadley.nz/>, viewed March 2024.
* Xie, Y; Altaire, J; Grolemund, G, 2023, *“R Markdown: The Definitive Guide”*, <<https://bookdown.org/yihui/rmarkdown/>>, viewed April 2024
* Zieffler, A, 2012, *“Statistical Thinking: A Simulation Approach to Modeling Uncertainty (STAT 216 edition)”*, <https://bookdown.org/frederick\_peck/textbook/>, viewed April 2024

#### XIII. References

**Where I obtained the datasets:**

* Kuzilek, J; Hlosta, M; Zdrahal, Z 2017, *“Open University Learning Analytics dataset”*, <<https://analyse.kmi.open.ac.uk/open_dataset>>, viewed March 2024
* MongoDB 2024, *“Sample training dataset | Grades”*, <https://www.mongodb.com/docs/atlas/sample-data/sample-training/#sample\_training.grades>, viewed March 2024

**Where I obtained information about university grading decisions:**

* Australian Education Info 2024, *“Grading system in Australia”,* <https://www.australiaeducation.info/Education-System/Grading-System.html>, viewed March 2024
* University of Southern Queensland 2023, *“Grades procedure”*, <https://policy.usq.edu.au/documents/20592PL>, viewed March 2024.
* University of Southern Queensland 2024, *“How do I work out my final grade?”*, <<https://usqassist.custhelp.com/app/answers/detail/a_id/2615/~/how-do-i-work-out-my-final-grade%3F>>, viewed March 2024.
* University of Sydney, *“How do I work out my final grade?”,* <https://usqassist.custhelp.com/app/answers/detail/a\_id/2615/~/how-do-i-work-out-my-final-grade%3F>, viewed March 2024
* University of Adelaide 2023, *“How do Withdraw No Fails (WNF) and Withdraw Fails (WF) affect my Grade Point Average (GPA)?”*, <https://student.ask.adelaide.edu.au/app/answers/detail/a\_id/1684/~/how-do-withdraw-no-fails-%28wnf%29-and-withdraw-fails-%28wf%29-affect-my-grade-point>, UofA, viewed March 2024

**Where I obtained information about a cut-off GPA for postgraduate study:**

* University of Adelaide 2023, *“Master of Artificial Intelligence and Machine Learning”*, <https://www.adelaide.edu.au/degree-finder/maiml\_maiml.html/>, UofA, viewed March 2024

**Where I obtained information that dbplyr is like a SQL alternative for R programmers:**

* Wickham, H & Ruiz, E 2024, *“Introduction to dbplyr”*, <https://dbplyr.tidyverse.org/articles/dbplyr.html>, viewed April 2024
* Wickham, H & Ruiz, E 2024, *“SQL translation”*, <<https://dbplyr.tidyverse.org/articles/sql-translation.html>>, viewed April 2024