CSC8631 Assignment Report

Marc Birkett

03/12/2021

CSC 8631 - Data Management and Exploratory Data Analysis

An investigation into an MDOC dataset using a CRISP-DM model. The model is the industry standard approach to data mining projects, it has six phases and is iterative. It is described as a "set of guardrails to help you plan organise and implement your data science (or machine learning) project."(Data Science Alliance, 2021).

The phases and tasks of a CRISP-DM project are outlined by the Data Science Alliance as per:

1. Business Understanding - what does the business need?

- Understand the business objective. What does the customer want to accomplish?
- Assess the situation. Determine resource availability, requirements, risks and contingencies
- Data Mining Goals - what does success look like?

2. Data Understanding - what data do we have / need? Identify, collect and analyse.

- Collect the initial data, acquire and load into data analysis tool.
- Describe the data, examine and document properties such as data format, number of records and identities
- Explore the data, query and visualise and identify the relationships.
- Verify the data quality and document any issues.

3. Data Preparation - prepare the final data sets for modelling.

- Determine which data set to be used and reasons for inclusion / exclusion.
- Clean the data
- Construct data, carry out any required feature engineering.
- Integrate data, create new data sets by combining data sources
- Re-format data as necessary, cast data types as required.

4. Modelling - what modelling techniques should we apply?

- Determine which models to try.
- Design tests.
- Build and assess the model.

5. Evaluation - which model best meets the business objectives?

- Evaluate the models against the business success criteria identified in step 1.

- Review the processes. Was anything overlooked and everything properly executed?
- Determine next steps. What do we need to do to deploy.

6. Deployment - how do stakeholders access the results?

- Plan deployment, document.
- Plan monitoring and maintenance.
- Produce the final report.

This report will cover the first three steps of this model and a minimum of two iterations. I will attempt to implement all of the associated tasks. With the lack of formal requirements provided by the assignment it will be difficult to define a business need in the first iteration of the model, therefore tasks such as defining the business objective and data mining goals will be defined for the second iteration as a result of the first. Therefore the two iterations of the model will be:

1. Iteration 1

- Data Understanding
- Data Preparation

2. Iteration 2

- Business Understanding - define and understand the hypothesis from Iteration 1.
- Data Understanding
- Data Preparation.

The investigative project will be completed in R using R Studio and has been set up using ProjectTemplate to provide structure and repeatability. Version control is provided by Git and this report created with R Markdown. Various libraries have been imported and used from the Tidyverse regarding data import and management (Dplyr, Readr) and visualisation (GGPlot2).

Iteration 1

A first iteration completing the understanding and preparation step will be carried out. The Future Learn MDOC dataset was downloaded as a zip file and reviewed. The data was supplied in csv files covering 8 different areas of the software over multiple stages. This was imported into R into 8 data frames from the original for easier analysis. The data was then reviewed.

Step 2: Data Understanding Upon initial import it was found that the detected data types were not consistent and so this was dynamically set on import. Particularly, any ID property was set to be an integer and date time properties were converted to the local POSIX time format.

Upon investigation it was decided that some data would be disregarded due to lack of breadth. Therefore, we have data covering the life cycle of a student including:

- **Enrollments**. Enrollment is a categorical dataset with n=37296 items and p=14 variables. It was found that the majority of data for all properties was "unknown" and therefore not suitable for further analysis.
  - **Candidate Keys -** "Learner_ID" of character data type.

– **Fields and data types -** Properties include Country, Gender, Age Range, Highest Education Achieved, Employment Area and Employment Status, Role all of type character. Also date / time of enrolled, un-enrolled, fully participated and purchased statements.

– **Related data sets -** Team Member, Leaving Survey and Step Activity can be related on Learner_ID

– **Data Quality -** The data types across date times in multiple files were inconsistent and therefore they were cast on import. The vast majority of data for each categorical value is "Unknown". Once the data was imported a basic type and missingness check was carried out. This showed type failures of 1 record on each of the date fields (enrolled, unenrolled, fully participated and purchased statement), there were also significant gaps in the same date based data, between 35k and 37k records. However, there were no issues with the categorical data. This was then plotted to investigate the data as shown in Figure 1 "Categorical Enrollment Data".

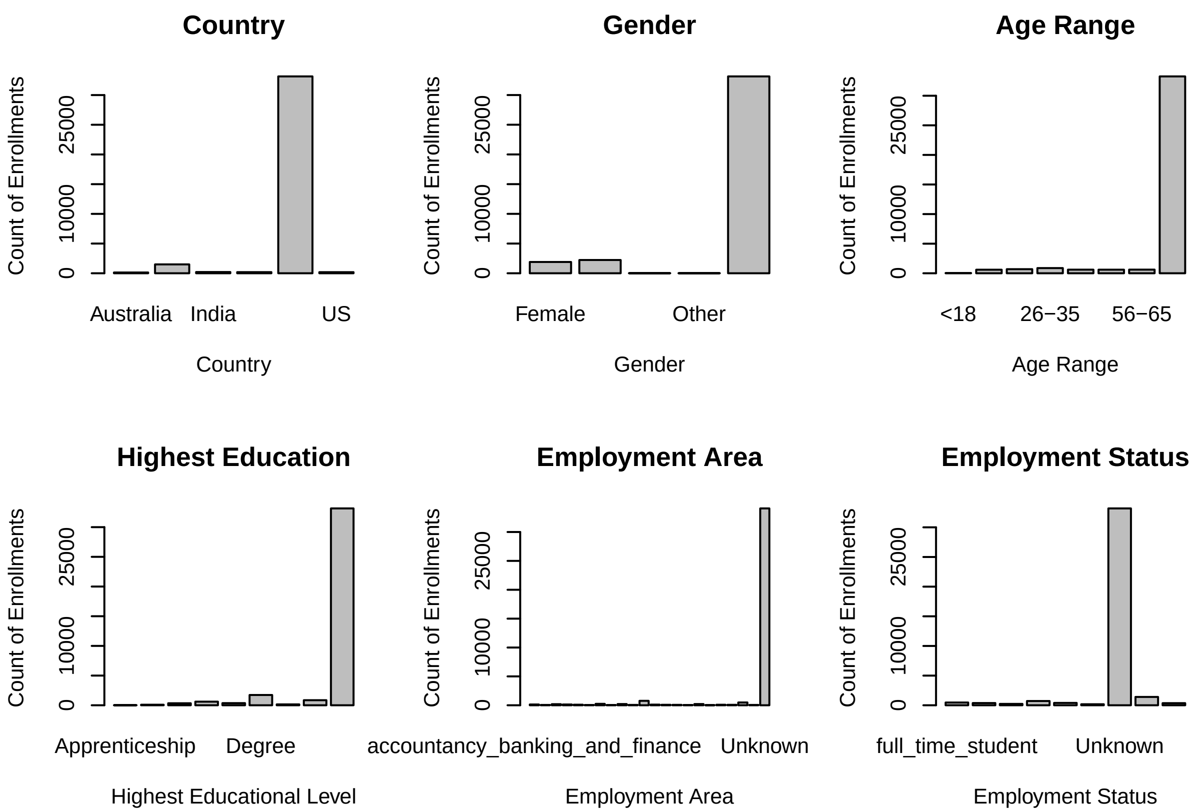

Figure 1: Categorical Enrollment Data

It can be seen that the majority of the data for the categorical data fields is "unknown" and therefore not suitable for further analysis.

- **Step Activity**. This indicates the stage in the course that the related data occurred at. It is principally categorical data with n=423072 items and p=6 variables.

  – **Candidate Keys -** A compound key of "Learner_ID" of character data type and step (integer). This identifies data about the step the student was at.

  – **Fields and data types -** "Learner_ID" of type character, "week_number" and "step_number" both of which are integer and "step" which is also an integer and is a concatenation of the two. "First_Visited_At" and "Last_Completed_At" are both date times and indicate when the student was at that stage.

  - **Related data sets -** Enrollments via "Learner ID", Question Response, Video Stats and Sentiment Survey via Step. Each step in the step activity data frame has associated quiz questions and responses.
  - **Data Quality -** Data types were consistent across the files to be imported. Date times were cast on import. Basic data quality checks indicate that "First_Visited_At" and "Last_Completed_At" are natively characters and so they were cast to date times. Significant gaps were observed on "last_completed_at" which may be correct.

- **Leaving Survey**. The leaving survey responses comprises of categorical, character and date based data regarding feedback from individuals which have left the course. The data frame has n=403 items and p=10 variables. The primary key of this table is ID which was cast to an integer in the import process.

  - **Candidate Keys -** "ID" which was cast to integer during import.
  - **Fields and data types -** "ID" as integer, "Learner_ID" as character, "Left_at" as datetime, the date and time a student left the course, "leaving_reason" as characters, "last_completed_step" as character and "last_completed_step" as as date time, also the "last_completed_week_number" and "last_completed_step_number".
  - **Related data sets -** Enrollments on "Learner_ID". Investigation into the key fields of "last_completed_step", "last_completed_step_number" and "last_completed_week_number" revealed that "last_completed_step" was a concatenation of the other two fields and could be used to relate to the Step Activity data. This allows the identification of the stage that students leave the course. This can be seen below.

```
## # A tibble: 5 x 3
##   last_completed_step last_completed_step_number last_completed_week_number
##   <chr>                                    <dbl>                      <dbl>
## 1 1.3                                          3                          1
## 2 1.19                                        19                          1
## 3 3.18                                        18                          3
## 4 1.3                                          3                          1
## 5 1.2                                          2                          1
```

  - **Data Quality -** Upon import it was necessary to standardise the data type across multiple files, id was cast to integer, last completed step to character and the last completed step and week numbers to integer. All date times were also cast to date time. Upon investigation of the data it was found that there were multiple leaving reasons which amounted to "lack of time" (Figure 1), this field was merged to make it comparable to other values (Figure 2)

```
##
##                         I don't have enough time
##                                               41
##                               I prefer not to say
##                                               18
##                                            Other
##                                               48
## The course required more time than I realised
##                                               16
##                            The course was too easy
##                                               10
##                            The course was too hard
```

```
##                                              12
##             The course wasn't what I expected
##                                              20
##       The course won't help me reach my goals
##                                              21
```

Once corrected the data is comparable as per below.

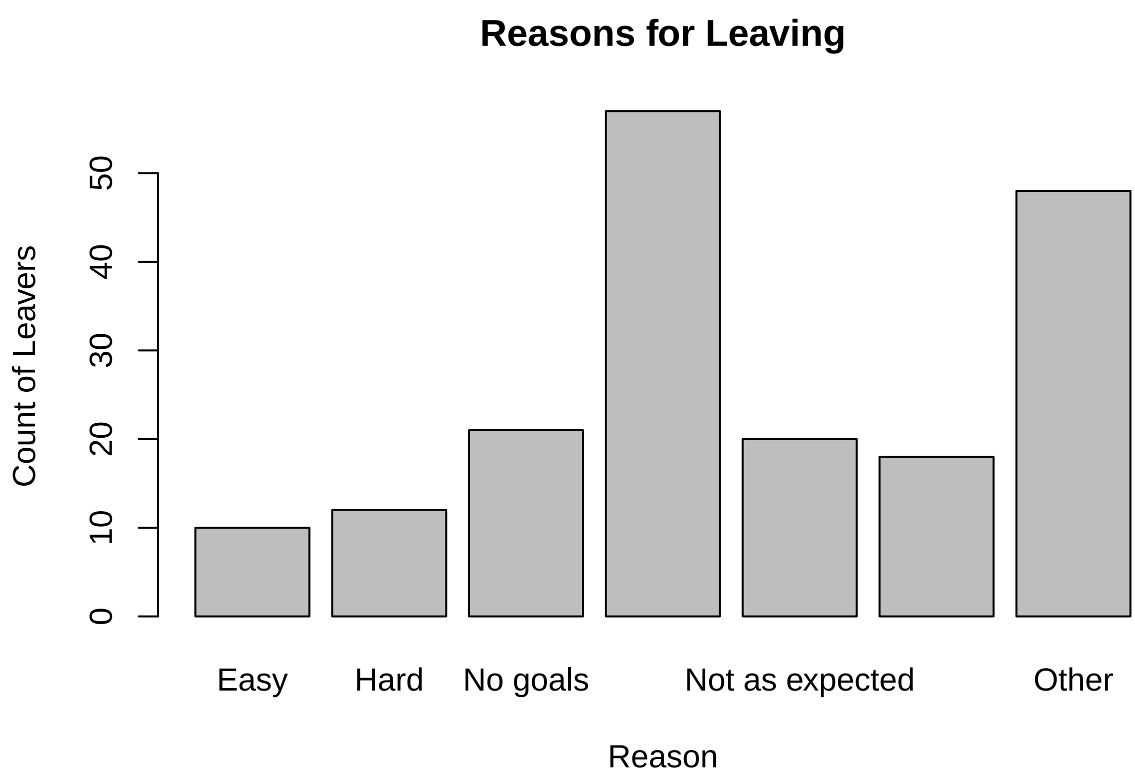

Figure 2: Engineered Reasons for Leaving

- **Question Responses**. Each step in the step activity data frame has associated quiz questions and responses. This is stored in the questions data frame. The question responses data is categorical data with n=176463 items and p=10 variables.
  - **Candidate Keys -** "Learner_ID" of character data type and quiz_question which is a concatenation of week_number, step_number and question_number. By concatenating just the week_number and step_number we are able to calculate the question step and relate to the Step Activity data set.
  - **Fields and data types -** Other properties outside of the keys are "question_type" which is exclusively "MultipleChoice", response of character which is the answer to the question, cloze_response which is always NA, submitted_at which is a datetime of the submitted at date and correct which is a boolean. This would give us access to the scores of each student
  - **Related data sets -** Some feature engineering would allow us to relate to the Step Activity data set and the learner_id allows us to relate to the Enrollments data set and any associated Leaving Surveys.

- **Data Quality** - Simple data quality checks were carried out which showed the correct data types, and that Cloze_Response is always null. A review of week_number, step_number and question_number confirmed that they could be concatenated to form the tables primary key and a step key, as per Figure 4 below, "Question Response Key Construct".

```
## # A tibble: 29 x 4
## # Groups:   quiz_question [29]
##    quiz_question week_number step_number question_number
##    <chr>               <dbl>       <dbl>           <dbl>
##  1 1.7.1                   1           7               1
##  2 1.7.2                   1           7               2
##  3 1.7.3                   1           7               3
##  4 1.7.4                   1           7               4
##  5 1.7.5                   1           7               5
##  6 1.7.6                   1           7               6
##  7 3.11.1                  3          11               1
##  8 3.11.2                  3          11               2
##  9 3.11.3                  3          11               3
## 10 2.8.1                   2           8               1
## # ... with 19 more rows
```

It can be clearly seen that concatenating the columns gives us the key.

- **Video Views**. Each step in the step activity data frame also has associated video view data. The video stats data is multivariate data with n=65 items and p=29 variables. There are significantly more data variables in this relation then in those examined previously, and significantly less data items.
  - **Candidate Keys** - "Step_Position" identifies the videos position within, in the course as a whole and could be related to Step Activity. This along with the video title would uniquely identify he video and the rest of the data in the data set is already grouped.
  - **Fields and type** - All of the fields in this dataset, other than "title", are categorical and are integers. The data in this relation can be subsetted into data pertaining to percentage complete, type of device and location of the video view which allows investigation of each specific facet. This will be revisited at the data preparation stage if required.
  - **Related data sets** -The step_position allows us to relate to step_activity which allows relationships through the rest of the dataset.
  - **Data Quality** - Data quality checks for type and missingness did not raise any issues. Upon investigation, the table is wide so, it will be necessary to pivot to see any meaningful data. Also the data is expressed as percentages. To make the data comparable it will be necessary to convert to absolute numbers and compare back to total_views.
- **Weekly Sentiments**. This is completed by each student per week. The sentiment data has n=181 items and p=6 variables. This table has a categorical properly rating and character data to cover the sentiment. I will not be looking to cover sentiment analysis in this work so only the categorical value will be looked at.
  - **Candidate Keys** - "ID" has been cast to integer during the import process, other than week_number I do not see a key to create a relationship into the wider data set.
  - **Fields and data types** - "Experience_rating" is an integer between 1 and 3 which gives the rating of the student for that week. "Week_number" is the number for the week between 1 and 3 and "reason" is a character string to justify the experience_rating.
  - **Related data sets** - None

  - **Data Quality** - ID, Week number and experience rating were cast to integer on import to standardise the data type across multiple data sets. Responded at was also cast from character to date time. Data quality checks revealed correct data types, and reason being missing in 111 cases from 181. There would be value in converting this data to percentages.

The relationships between data sets as discussed above are documented below.

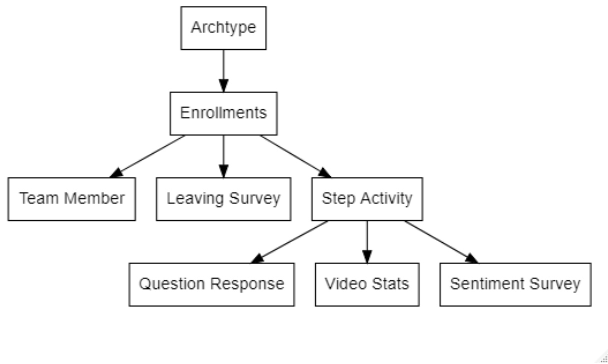

Figure 3: Relationships between data frames.

Data Preparation Each set of data was imported into a dataframe during the data understanding stage. This was done by filtering the file set and reading the contents of each file to a list. This was then bound by Dplyr. The example below binds all *question.response.csv files into a dataframe.

```
#working files
files = list.files(pattern="*question.response.csv")

datalist = lapply(files, function(i){
  csv <- read_csv(i, show_col_types = FALSE)
  csv$stage_id <- substring(i,16,16)
  csv
})
dfQR <-dplyr::bind_rows(datalist)
```

As we still do not have a clear business objective in this first iteration, the data preparation stage will be used to implement any data changes to make a more completed and integrated dataset. The changes implemented here may or may not be required in the analysis carried out in iteration 2, they have merely been observed while carrying out the steps during the data understanding process. Processes such as re-formatting and casting the data have already been carried out in the data understanding step to successfully merge multiple CSV files into one data frame.

Particularly, it was identified that some feature engineering could be carried out to prepare the data for further analysis. These fixes were implemented in the munge process. Particularly:

- **Step Activity**, this dataset has the step start and end date. Some data engineering was completed to add an "isComplete" flag to the dataset, as well as a "completedTime" integer indicating the difference between the start and end date of the step.
  - An isComplete flag was added by checking the last_completed_at date for NA.

```
dfSA$isComplete = !is.na(dfSA$last_completed_at)
```

- A time to complete count in days was added by subtracting the start date from the end date.

```
dfSA$timeToComplete = difftime(dfSA$last_completed_at, dfSA$first_visited_at,
                               units="days") #calculate the difference
```

- Question Responses, it was identified that while the dataset is missing the step column to relate the Step Activity table, this can be created from data which is present. This allows us to integrate responses to steps and to the wider dataset.
  - The quiz_question key was split on . to a list. SApply was then used to concatenate the first two elements together split by a dot. This was then added to the dataset.
  - Dplyr was then used to prove a left join between the question responses and step activity data sets.
- Video Views, it was identified that this dataset has a logical collection of three types of data and could be split into three data sets. It was also observed that a pivoted data set would be required for analysis and the conversion of data from "percentages of percentages" to absolute counts.
  - Splitting out the child dataset was carried out by taking a slice of the original dataset.

```
dfVSTotals = dfVS[,c(1,2,4, 9:15)]
```

  - The data frame was then extended by adding the appropriate properties. In this example a helper function was used to express the percent as raw numbers..

```
dfVSTotals$"05" = as.percent(dfVSTotals$total_views, dfVSTotals$viewed_five_percent)
dfVSTotals$"10" = as.percent(dfVSTotals$total_views, dfVSTotals$viewed_ten_percent)
dfVSTotals$"25" = as.percent(dfVSTotals$total_views, dfVSTotals$viewed_twentyfive_percent)
dfVSTotals$"50" = as.percent(dfVSTotals$total_views, dfVSTotals$viewed_fifty_percent)
dfVSTotals$"75" = as.percent(dfVSTotals$total_views, dfVSTotals$viewed_seventyfive_percent)
dfVSTotals$"95" = as.percent(dfVSTotals$total_views, dfVSTotals$viewed_ninetyfive_percent)
dfVSTotals$"99" = as.percent(dfVSTotals$total_views, dfVSTotals$viewed_onehundred_percent)
```

  - The result was then selected using Dplry to produce a neater table. A pivoted table was then created to support reporting. The figure below shows the transformed data in a logical structure for reporting. This was carried out for all three data sets.

```
  dfVSTotals = select(dfVSTotals, step_position, title, "05", "10", "25", "50", "75", "95", "99")
dfVSTotalsPivot = dfVSTotals %>%
  pivot_longer(!(1:2), names_to = "percentviewed", values_to = "count") #createpivot
```

The formatted data for reporting is as per below.

```
## # A tibble: 455 x 4
##    step_position title                       percentviewed count
##            <dbl> <chr>                       <chr>         <int>
##  1           1.1 Welcome to the course       05             1276
##  2           1.1 Welcome to the course       10             1250
##  3           1.1 Welcome to the course       25             1218
##  4           1.1 Welcome to the course       50             1167
##  5           1.1 Welcome to the course       75             1130
```

```
## 6          1.1  Welcome to the course           95             1102
## 7          1.1  Welcome to the course           99             1056
## 8          1.14 Why would anyone want your data? 05              660
## 9          1.14 Why would anyone want your data? 10              645
## 10         1.14 Why would anyone want your data? 25              623
## # ... with 445 more rows
```

Iteration 2

Throughout iteration 1 a hypothesis to investigate has not become clear.

Business Understanding The dataset is principally a collection of categorical data that could be investigated of which nothing leaps out at this time. I suggest that the objective to accomplish to amass a greater understanding of the data and success looks like some interesting avenues to investigate. A hypothesis may become clearer during the data understanding and preparation phases.

A successful project would outline more than one avenue for further exploration with an indication of what may be found. Continuous data is available via the Video View data which may be of further interest to investigate from a complexity point of view.

The risks of such an approach are that nothing leaps out of interest or for further analysis. However, this is a valuable finding in itself.

Data Understanding To support this iteration, I felt that I needed to review the data sources and further investigate the structure. It was found that the Archetype and Team Member data sets had been missed in iteration 1. Upon further investigation, it was found that they relate to Enrollment on Learner_ID on a one to neo basis. Therefore I have merged the Team Member, Archetype and Enrollment data sets to create separate Person and Student Info data sets, these are related on Learner_ID.

```
#create dfStudent and person from enrollments, archetypes, team members
dfStudent <- left_join(dfE, dfAR, by = c("learner_id" = "learner_id")) #enrollment
dfStudentTM <- left_join(dfStudent, dfTM, by = c("learner_id" = "learner_id")) #team member
dfStudentLeavers <- left_join(dfStudentTM, dfLSR, by = c("learner_id" = "learner_id")) #leavers
dfStudentLeavers$left = !is.na(dfStudentLeavers$last_completed_step) #set leaver flag
```

This gave me a complete dataset regarding the students, which I subsequently split into Student Info and Person data sets. I also flagged any leavers to allow this to be easily visualised.

```
#select out two tables
dfStudentInfo = select(dfStudentLeavers, learner_id, enrolled_at, unenrolled_at,
                  fully_participated_at, purchased_statement_at,
                  archetype, role, team_role, user_role, stage_id.x)

dfPerson = select(dfStudentLeavers, learner_id, gender, country, age_range,
                   highest_education_level, employment_status, employment_area,
                   detected_country, left)
```

Person contains details about the person including age, age_range, education level. Student Info contains student centric information such as enrollment dates and various roles, archetypes etc. I have also slimmed down the Step and Answers data sets.

```
## # A tibble: 6 x 6
##   learner_id                stage_id  step step_number isComplete timeToComplete
##   <chr>                        <int> <dbl>       <dbl> <lgl>      <drtn>
## 1 77454a73-6b8b-46a2-8dee-~        1   1.1           1 FALSE      NA days
## 2 c1a75ae7-c76f-43ac-ad9f-~        1   1.1           1 FALSE      NA days
## 3 a4fa6f89-a596-4d00-9397-~        1   1.1           1 FALSE      NA days
## 4 60b56cea-ad2d-4dcb-95c1-~        1   1.1           1 TRUE        0 days
## 5 05a815ce-3c40-4005-a3eb-~        1   1.1           1 TRUE        0 days
## 6 5553e67a-1780-42f3-8355-~        1   1.1           1 TRUE        1 days
```

```
## # A tibble: 6 x 7
##   learner_id         step week_number quiz_question response submitted_at
##   <chr>             <dbl>       <dbl> <chr>            <dbl> <dttm>
## 1 77454a73-6b8b-46~   1.7           1 1.7.1               12 2016-07-06 00:00:00
## 2 77454a73-6b8b-46~   1.7           1 1.7.1              123 2016-07-06 00:00:00
## 3 a4fa6f89-a596-4d~   1.7           1 1.7.1              123 2016-07-11 00:00:00
## 4 a4fa6f89-a596-4d~   1.7           1 1.7.1               12 2016-07-11 00:00:00
## 5 a4fa6f89-a596-4d~   1.7           1 1.7.1               23 2016-07-11 00:00:00
## 6 f27eec8c-eaf1-4e~   1.7           1 1.7.1              123 2016-07-27 00:00:00
## # ... with 1 more variable: correct <lgl>
```

We are now left with the following structure which I am confident can be correctly queried.

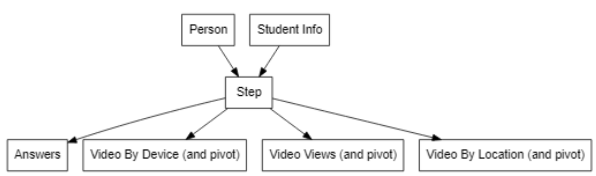

Figure 4: Relationships between data frames.

These transformations were subsequently loaded into the analysis tool, and an extra step was added to the munge process to handle these transformations.

People and Students My initial investigation queried the categorical data regarding people and students. Each categorical item was plotted on a bar chart with "Unknown" removed to allow a reasonable view on the rest of the data. As identified in iteration 1, unknown makes up the vast quantity of this data. Students split by gender is roughly a 50 / 50 split. Country offers GB as the clear leader where the data is known. Age range offers 26-35 as the most common group, but get a general spread over all range other than under 18. So far so predictable. The only clear outliers to give any kind of insight into the type of student attracted is:

- Highest education = University degree
- Employment Status = Working full time
- Employment area = IT and Information Services

Detected country also found GB to be the most common value after Unknown. However, students would travel to take part in their course so this may be artificially bloated. So we have learned that individuals taking a cyber security course generally work in IT, full time, and the gateway to such a career is a university degree.

Of potential more interest in the multiple roles for each student. We can see that there is a clear number of vitalisers over the other archetypes, and very few flourishes and hobbyists. This is shown in the figure below.

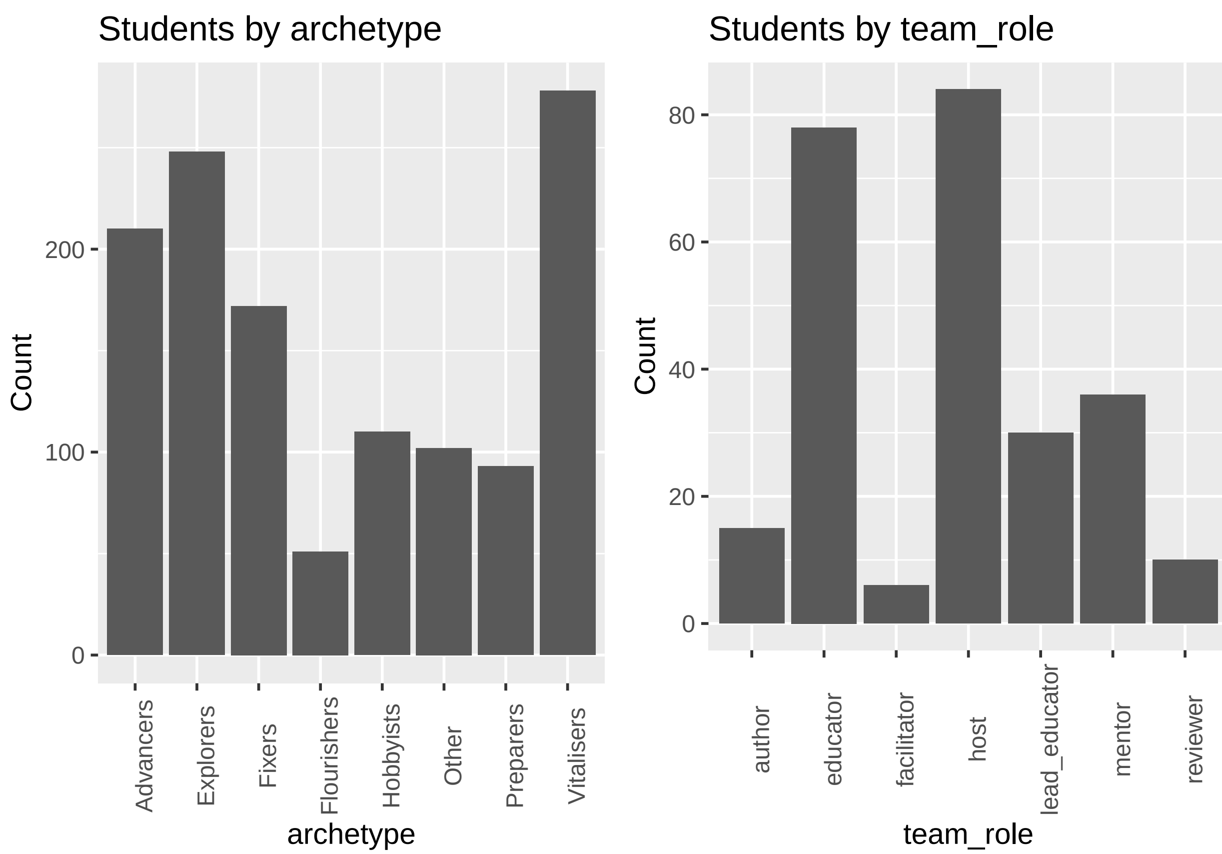

Figure 5: Student by Archetype and Team Role

The vast majority of students on the course are learners rather than admins and the team roles reflect educator and host being the team roles. Those don't particularly sound like student roles and the low numbers in each role could also reflect this. Some investigation of archetype vs team roles in further iterations could be of interest.

Course Progress I adopted the completed steps of the course to reflect course progress. The completed date had already been used to indicate a completed step. When counting the number of completed steps across all cohorts as per the chart below, you can see the numbers are significant. It can also be seen that 1.1 is the most completed step by far, and the completed steps across the week reduces. There is then a spike at the beginning of the next week and so on until the end of the course.

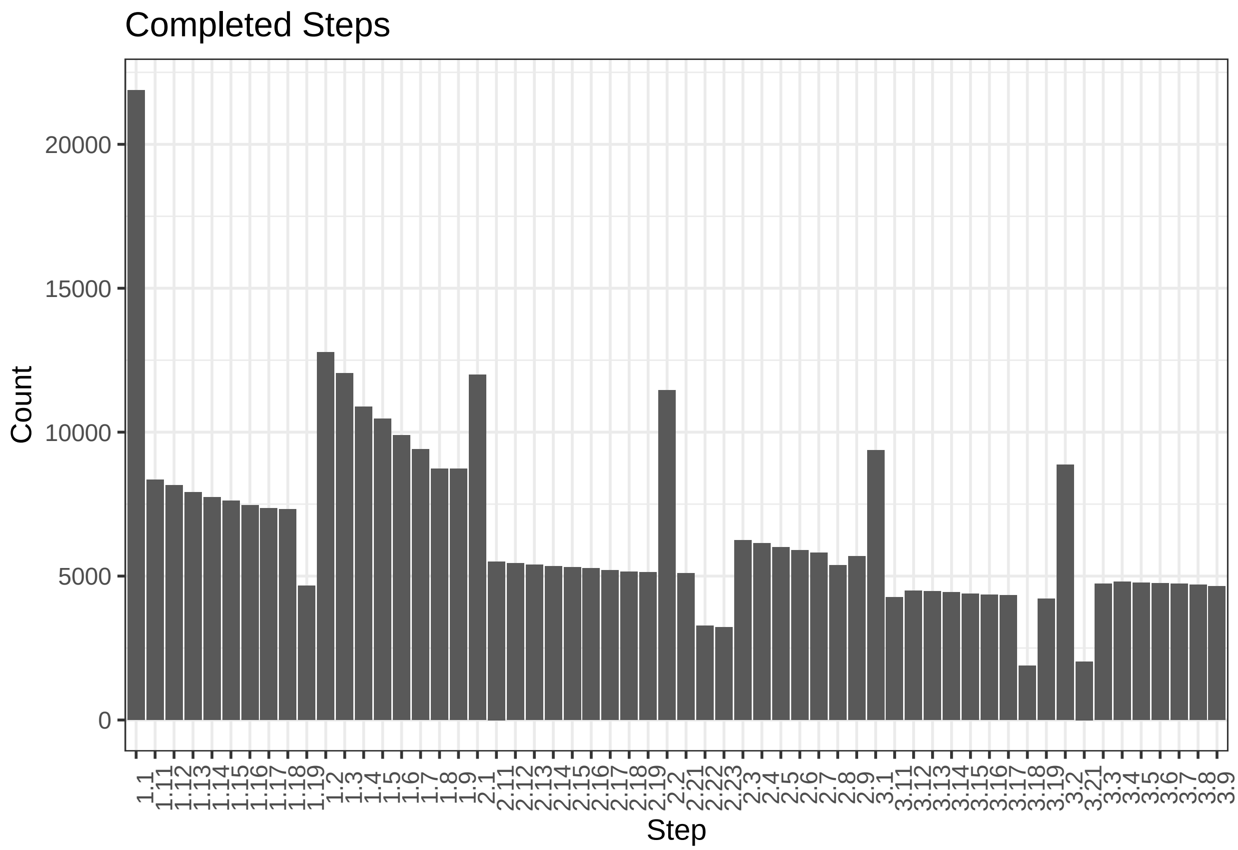

Figure 6: Completed Steps

The numbers completing the final step are significantly lower than those starting which could signify a high drop out rate, however, the numbers look generally high so this will be revisited.

Course progress was then revisited by the same categorical variables as those used for students outlined previously. Course progress by team role indicated some interesting number as per the figure below. This was also cross referenced with organisation admin. As per figure 7 below, the large number of incomplete steps against each team role at first indicated that team roles aren't associated with learners, but are more likely to be associated with organisation admin. The user role graph does indicate a lot of incomplete steps for the admin roles, but surprisingly, also for the user roles when I would have expected this to be higher. Also the count of steps against the admin roles appear to be exceptionally large at this point. I became concerned at the quality of the steps data so this would be revisited in the data preparation step of this task.

```
## Note: Using an external vector in selections is ambiguous.
## i Use ‘all_of(column)‘ instead of ‘column‘ to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

Question Answers Inclusion of the question and answers data allowed me to investigate the number of questions answered and the result. I would also be able to link this to step data via the feature engineered

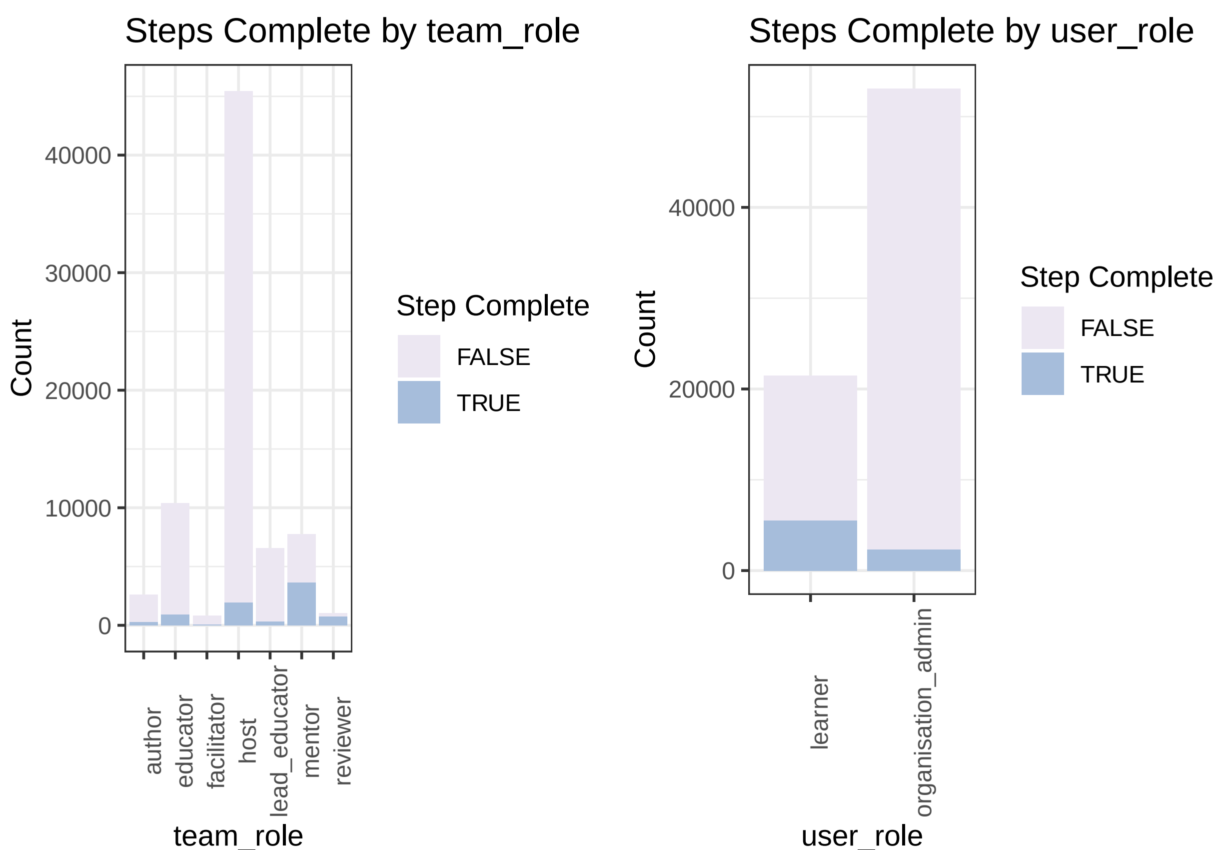

Figure 7: Progress by Team and User Roles

“step” number in the question data set. However, my concerns above regarding the quality of that data meant that was left for a potential later iteration. I was however able to investigate the number of questions which students had attempted to answer and what the outcome was, as per the figure below.

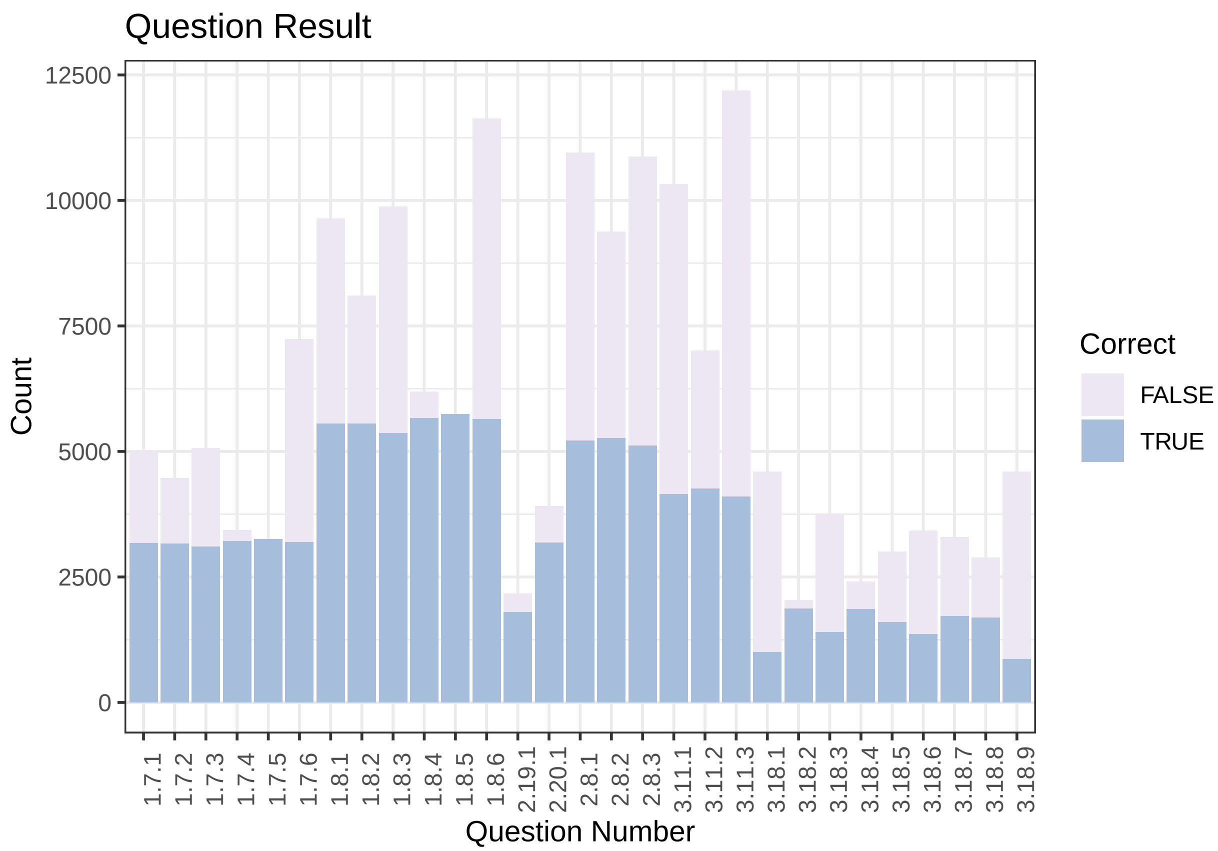

Figure 8: Questioned Answered with Result

It can be seen that the attempts at answering the quiz questions vary wildly throughout the course, although there is some consistency between the numbers of correct answers. Week three quiz attempts and correct answers are markedly lower.

At this point I chose to investigate the step data set which looked excessive. This was revisited in the data preparation task.

Data Preparation During the data understanding task I established that the step data I was working with did not seem to be correct. There was a lot of data and what seemed to be an unfeasible amount of steps per student, given that this is only a three week course. I also wanted to explore some continuous data as the majority of data explored so far had been categorical.

I established that most of the continuous data in the complete dataset was regarding video views. This dataset is a count of views by various metrics against each video in the step. However, given that I hadn’t particularly explored it I was keen to use some data I had viewed previously. In the student info dataset I had enrolled and un enrolled dates and use this to feature engineer a total enrolled time metric per student, expressed as enrolled date - un enrolled date, in days. I then attempted to correlate the enrolled time against the number of steps completed. This left me with 4374 students to work with and 385558 completed steps, a ratio of 1 to 88.147, which is larger than the unique steps in the course, at 58. The workings are shown in the next figure.

```r
#Student Info - Enrolled length of time
dfStudentInfo$totalEnrolledTime = difftime(dfStudentInfo$unenrolled_at,
                                           dfStudentInfo$enrolled_at,
                                           units="days") #calculate the difference

#get the time enrolled
df <- dfStudentInfo %>% drop_na(totalEnrolledTime)
df <- filter(df, totalEnrolledTime !=0)
df <- select(df, learner_id, enrolled_at, unenrolled_at, totalEnrolledTime)
df$totalEnrolledTime = as.double(round(df$totalEnrolledTime, digits=0))
count(df) #4374

#get the steps passed
dfCompletedSteps = filter(dfStep, isComplete==TRUE)
dfCompletedSteps$step = as.character(dfCompletedSteps$step)
count(dfCompletedSteps) #385558

#unique steps
length(unique(dfStep$step)) #58
```

To investigate, I created some plots showing total steps by learner, which was a count of steps completed against learner id. The result was giving me count of up to approximately 1500 steps, so I chose to query the data more directly as per below.

```
## # A tibble: 350,144 x 3
##    learner_id                           step      n
##    <chr>                                <chr> <int>
##  1 f605eb94-981c-4e4b-bafa-17ca61606acb 1.1       8
##  2 eccf7f6e-5364-47b3-ada7-aaffe578d636 1.1       7
##  3 36025d73-6f7f-4a3f-8ebf-6794b82cb0a6 1.1       6
##  4 36025d73-6f7f-4a3f-8ebf-6794b82cb0a6 2.1       6
##  5 36025d73-6f7f-4a3f-8ebf-6794b82cb0a6 2.2       6
##  6 36025d73-6f7f-4a3f-8ebf-6794b82cb0a6 3.1       6
##  7 36025d73-6f7f-4a3f-8ebf-6794b82cb0a6 3.2       6
##  8 36e9e17f-220d-43c6-aa94-bbb0aca3797b 1.1       6
##  9 3befdad2-f0ad-445c-ab55-4c73c50d5a8e 1.1       6
## 10 40207781-50bb-4d85-8367-83c3dd043d4b 1.1       6
## # ... with 350,134 more rows
```

This count of steps by learner shows multiple rows of the same learner_id, i.e one per step, but it also shows multiple enrollments on that step. This is a clear difference to what I assumed that the data would be. To check this i removed the step number from the count and then also queried the original data as per the figure below.

```
## # A tibble: 14,459 x 2
##    learner_id                               n
##    <chr>                                <int>
##  1 55d8eb62-927e-4d74-888c-59aa86ee4fd9   183
##  2 d416c199-cf68-4c94-9ad2-23742bbd282d   183
##  3 d670791b-49b2-4bd5-bac1-312528d53a26   183
##  4 36025d73-6f7f-4a3f-8ebf-6794b82cb0a6   182
##  5 75ba2d07-bfc5-4049-b622-7ab91b6d5baa   182
##  6 ad4a3395-6123-4772-a6dc-ad9f10e74418   182
```

```
##  7 7b30f56d-593b-43e6-bac2-9397dedfecce   179
##  8 f605eb94-981c-4e4b-bafa-17ca61606acb   165
##  9 ee42384b-f618-4e5e-8a8c-0f6e19840ffd   152
## 10 ae4c1068-3314-4a2e-a2f0-2671508140d9   151
## # ... with 14,449 more rows
```

```
## # A tibble: 20,285 x 2
##    learner_id                               n
##    <chr>                                <int>
##  1 77454a73-6b8b-46a2-8dee-35f36b6c4fc1   304
##  2 f27eec8c-eaf1-4e6a-90f0-d6d5b653285d   192
##  3 55d8eb62-927e-4d74-888c-59aa86ee4fd9   184
##  4 75ba2d07-bfc5-4049-b622-7ab91b6d5baa   183
##  5 d416c199-cf68-4c94-9ad2-23742bbd282d   183
##  6 d670791b-49b2-4bd5-bac1-312528d53a26   183
##  7 36025d73-6f7f-4a3f-8ebf-6794b82cb0a6   182
##  8 7b30f56d-593b-43e6-bac2-9397dedfecce   182
##  9 ad4a3395-6123-4772-a6dc-ad9f10e74418   182
## 10 f605eb94-981c-4e4b-bafa-17ca61606acb   170
## # ... with 20,275 more rows
```

As can be seen there are a lot of learners with an unfeasible amount of steps for a 3 week course with a max of 58 steps. The highest number of 183 is approximately 3 times the maximum of the course. It can also be seen that the same issue is seen in the original data, with a maximum of 304 steps, 5.2 times the maximum steps on the course. Since the issue was in the original data I concluded that I had not made any relational errors in creating my data tables. My misunderstanding was on the import from the source files into R in my munge process.

The original data had been provided in 7 data files where the number had been provided in the file name. Given that this potentially could multiple the data, depending on the learner_id in each file, I decided to investigate whether I had multiple enrollments across files. For want of a better phrase I had added that value into the source data table as "stage_id", and then has used it no further.

To explore this the stage_id was added to munge step for both the enrollment data and the step data. I then visualised completed steps and enrollment against the stage id.

```
## # A tibble: 4,221 x 2
##    learner_id                           stage_id.x
##    <chr>                                     <int>
##  1 00353a59-73a5-41e7-9b9e-fcd7a04accc9          7
##  2 005e96bf-d97f-4b85-abb5-c6cbc0ed3451          4
##  3 006a7609-762b-4a65-b434-fbc0d6db79e5          5
##  4 00a60ddd-87cf-48bb-8dbe-c3682ef413dd          2
##  5 00b80d03-2eb5-4f60-98c5-5fa2868bcd35          2
##  6 0100d15a-f66a-4615-91aa-efe5e2ad9ece          3
##  7 010ac5eb-d195-42c5-a463-d45ea7e12f15          3
##  8 011c29c5-9609-45ab-95f1-e4f78e4f5887          1
##  9 011dc6d1-1af0-42e4-9639-9d01d167fd89          5
## 10 011dc6d1-1af0-42e4-9639-9d01d167fd89          6
## # ... with 4,211 more rows
```

You can see from the figure above that there are multiple enrollments against each stage, which possibly indicates late joiners to the course. By sorting the data by learner and stage it can also be seen that there are multiple enrollments of the same learner across all stages but not all students enrolled on each stage. This has introduced multiplicity which should have been found in the previous iteration.

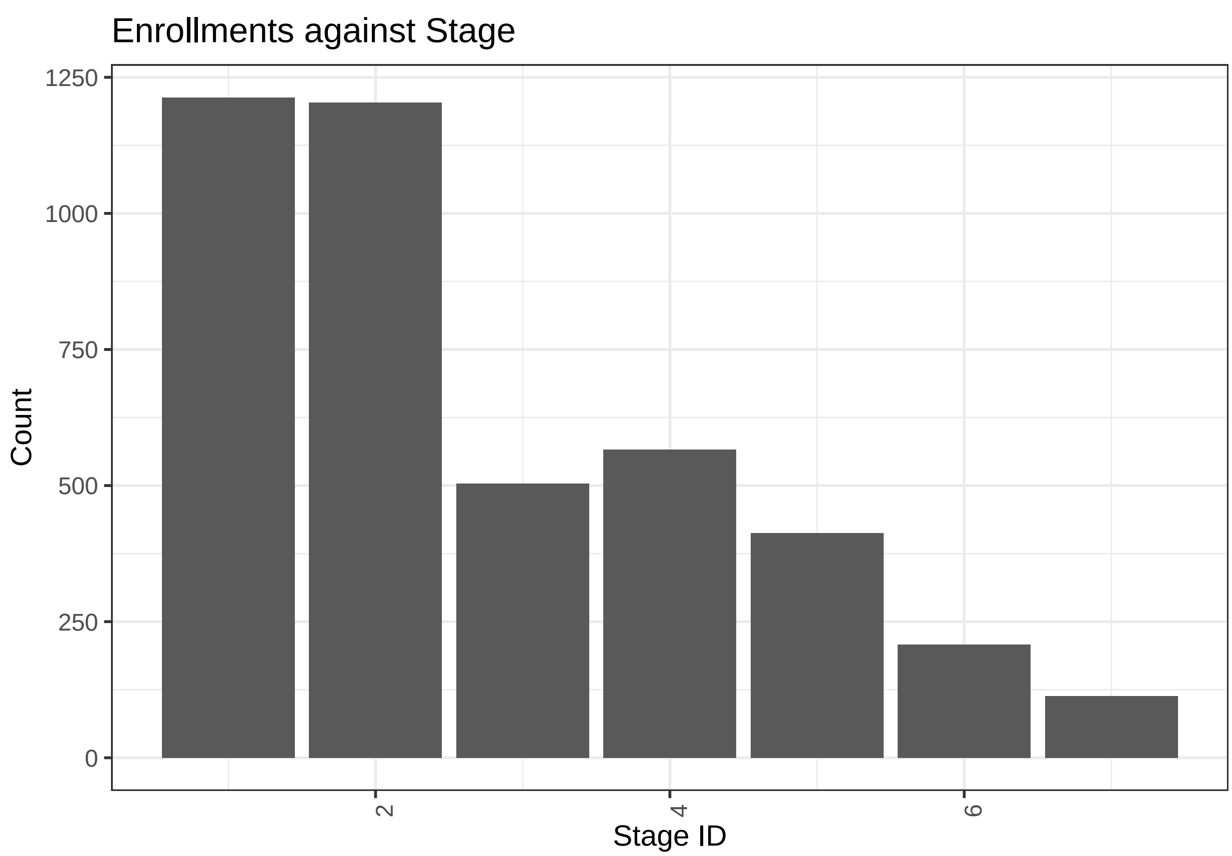

Figure 9: Enrollments against Stage

```
## # A tibble: 1 x 1
##        n
##    <int>
## 1 385558
```

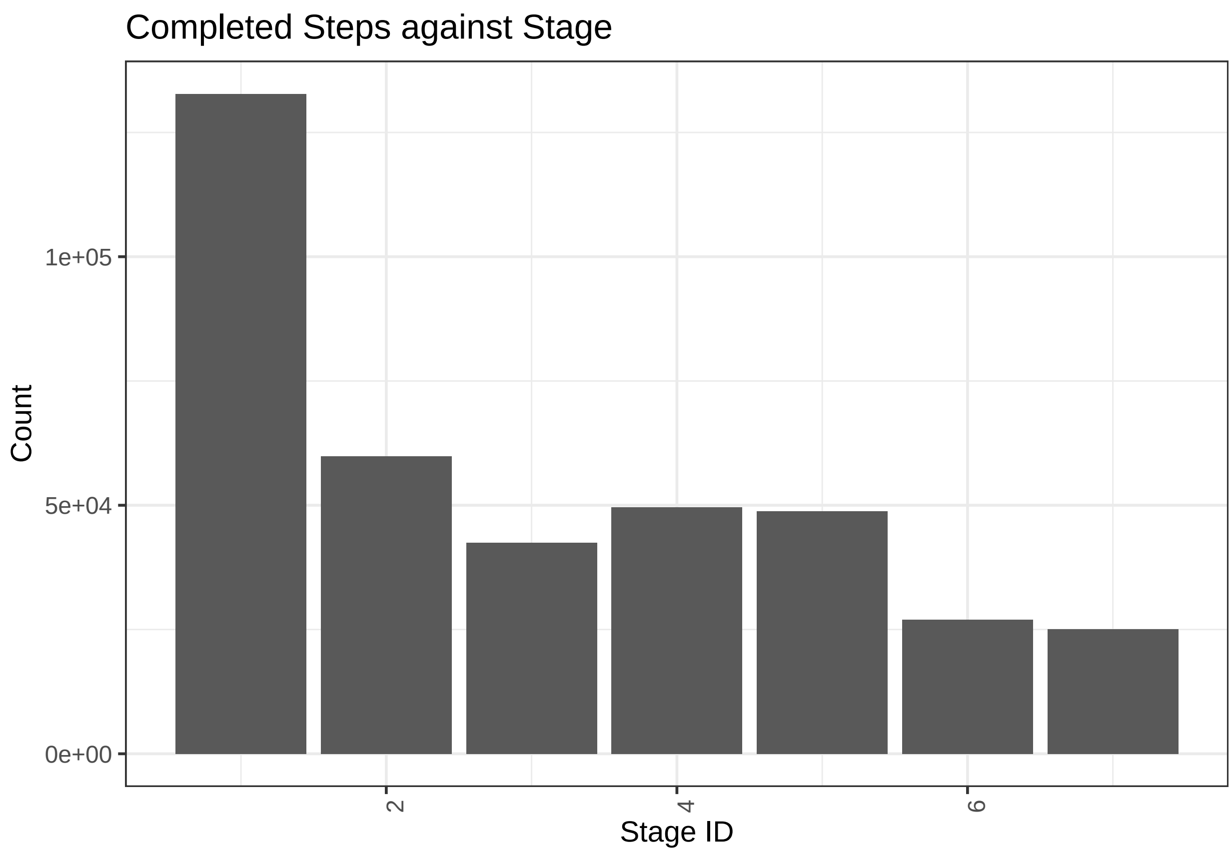

Figure 10: Completed Steps by Stage

```
## # A tibble: 385,558 x 3
##    learner_id                           stage_id step
##    <chr>                                   <int> <chr>
##  1 000a49d0-39c8-4cef-848c-a37d3574d179        1 1.1
##  2 000a49d0-39c8-4cef-848c-a37d3574d179        1 1.2
##  3 000a49d0-39c8-4cef-848c-a37d3574d179        1 1.3
##  4 000a49d0-39c8-4cef-848c-a37d3574d179        1 1.4
##  5 000a49d0-39c8-4cef-848c-a37d3574d179        1 1.5
##  6 000a49d0-39c8-4cef-848c-a37d3574d179        1 1.6
##  7 000a49d0-39c8-4cef-848c-a37d3574d179        1 1.7
##  8 000a49d0-39c8-4cef-848c-a37d3574d179        1 1.8
##  9 000a49d0-39c8-4cef-848c-a37d3574d179        1 1.9
## 10 000a49d0-39c8-4cef-848c-a37d3574d179        1 1.1
## # ... with 385,548 more rows
```

As you can see from the figure above there is also multiple steps against each stage, and there are steps per learner per stage. This also introduces multiplicity in the visualisations.

The stage_id would perhaps have been better referred to as Cohort ID. This finding would improve the dataset in any subsequent iteration of this data exploration.

Conclusions

The second iteration provided a couple of potential avenues for exploration, even though it only examined half of the dataset. Particularly:

1. Team Role vs Archetypes - do archetypes reflect the student cohort and team roles the teaching staff?
2. Do students with previous higher education attainment complete more steps of the course successfully? Or does the data merely prove there is more of them on the course?
3. Questions and Answers could be related to Steps and therefore Students to get greater insight on student results.

What we learned from Iteration two is that I had failed to take into account an idea of a student cohort when I imported the data. If I had asked that question earlier in the project the data would have made more sense. The range of new skills and approaches during this assignment and unit was so wide and detailed I feel like I have barely scratched the surface. However, time is against me.

References

Data Science Alliance, 2021, What is CRISP-DM, https://www.datascience-pm.com/crisp-dm-2/, Accessed: 26/11/2021