Data Analysis Report of Cyber Security Course

James Terence White

23/11/2020

1 Business Understanding

This report is an investigation into a online course hosted by Future Learn. Future Learn is an online platform which has partnered with numerous world leading universities and organizations to delivers a wide range of courses. One of which is Cyber Security, delivered by Newcastle University, where in this report we will undergo a forensic investigation in the form of data analytics.

1.1 Business Objectives

Since Future Learn is an educational site, their interests will fall within anything that will enhance the learning experience, increase the student's interaction with the site, and ultimately have repeat student signing up for more online courses. Thus, we can say a positive outcome from someone taking their course would be to have gain skills or knowledge related to the course they signed up for, and the course was delivered in a way that was stimulating and engaging for the student. Currently, the government are addressing issues, such as "identifying at-risk students" – presented in From Bricks to Clicks.

Although the students welling is paramount, this report will investigate how students interact with the course. By doing so, we aim to measure the courses success and where the course is thriving and where the course is not. Once we tackle these issues, a plan can then be put in place to make improvements to the course where and if needed.

1.2 Assess the Situation

1.2.1 Sources of Data and Knowledge

For this project the data was provided by Newcastle University who are assumed to have direct access to the online courses data.

1.2.2Data Assessment

We are presented a online Cyber Security course Big Data set over the course that has run 7 times, it is fair to say that the course only ran seven time. The data presented is comprised into numerous csv files that can be summarised into the following;

- Survey questions
  - Archetype - relating to to the users psychological traits
  - Weekly sentimental - students feedback on the course
  - Leaving
- Stats
  - Enrollment
  - Step Activity
  - Question Responses
  - Video (>run2)
  - Team members (>run1)

The course content is delivered in the form of videos and notes, of which are separated into steps, i.e. chapters to the Cyber Security course. Within these "steps" are sub-sections, and is what is referred to throughout the data set table headings. However, it is important to note that the video data is only present after run two, it is assumed that videos were provided to the student, but the assimilation of this data was not yet available. Furthermore, there is a Team member file available after run one which contains data on any rolls that were allocated within the course, such as mentors, or course organisers.

Cyber Security ran seven time from 2016 to 2018, for a period of three weeks - see details below Tab. @ref(tab:dates). All information surrounding this was obtained through the course pdf documents - see the data file in the Project Template folder.

Table 1: Summary of start and end dates for each run

Run	StartDate	EndDate
1	05/09/2016	26/09/2016
2	20/03/2017	10/04/2017
3	18/09/2017	09/10/2017
4	13/11/2017	04/12/2017
5	05/02/2018	26/02/2018
6	11/06/2018	02/07/2018
7	10/09/2018	01/10/2018

1.2.3 Requirements

For this project reproducibility is a key requirement of the project. To enforce this all analysis will be done in R, more specifically ProjectTemplate, reports will be compiled in RMarkDown and finally, Git version control will be used. The project lifespan is 4 week, and scheduled to be completed by 4^th of December 2020. With regards to the legality of the data, the assumption is that we have full consent from the data owner and is provided by CSC8631 - Data Management and Exploratory Data at Newcastle University.

1.2.4 Assumptions

The data is believed to be sourced directly from the course online database, so the data can be assume to be reliable. it is also assumed that Future Leans competitors did not have a direct impact on the data set since their largest competitors, i.e. Udemy (2009) we founded before Future Learn (2012). Similarly, economic factors are assumed not to have an impact on the quality of the data. With regards to the presentation of data, all reports should be deduced on RMarkdown and graphical models plotted in R.

1.3 Data Mining Goals

1.3.1 Goals

To answer the business question of “increasing the student interaction with the course”, the data mining aims to satisfy this objective by extrapolating trends in the quiz response data set and how they vary over the number of times this course was run and throughout the course duration. The problem can be answered through a verity of data mining techniques depending on how the data presents itself. After all, this is an exploratory data analysis.

1.3.2 Sucess Criteria

It would be fair to say that the data mining can be deemed a success if, a correlation between the quiz question responses can be related to the interaction in the online course is found, and hows this changes over the number of runs and course duration. For example, how does the participation in the course questions change over the number runs and throughout the course.

1.3 Data Mining Goals

Throughout this data mining process the goal is to explore the data set surrounding the posed question of "identifying students that may need additional support" - is there any evidence within the data provided that support this. A successful outcome of this analysis would be to find a correlation.

1.4 Project Plan

To achieve these goals an analysis will be partaken on the quiz.responses files, where trends will hopefully present themselves. Since the data that was to be analysed was already proved, cut down on a vast amount of the project. Therefore, it is estimated that a vast amount of the project time and effort will be spent understanding the data preparation - cleaning and reformatting of the data.

1. Business understanding
2. Data mining
3. Data preparation
4. Modeling
5. Evaluation
6. Deployment

With regards the the project stages 3 and 4 will be iteratively repeated, where each iteration will make empirical judgments based on the results from the previous model. A majority of the project lifespan is expected to be in the reformatting and modeling stages where numerous iterations will be carried out. This process, depending on the results, will be repeated for a finite number of times, with each cycle further support the previous findings. Equally, if the results are exhausted, consider answering a different data mining question.

2 Data Understannding

The data to be used for the project is contained in several files, more specifically in the quiz.response file which is created for each run of the course. Displaying the head of the data frame from run ones quiz responses

```r
head(cyber.security.1_question.response)
```

```
## # A tibble: 6 x 10
##   learner_id quiz_question question_type week_number step_number question_number
##   <chr>      <chr>         <chr>               <int>       <int>           <int>
## 1 77454a73-~ 1.7.1         MultipleChoi~           1           7               1
## 2 77454a73-~ 1.7.1         MultipleChoi~           1           7               1
## 3 a4fa6f89-~ 1.7.1         MultipleChoi~           1           7               1
## 4 a4fa6f89-~ 1.7.1         MultipleChoi~           1           7               1
## 5 a4fa6f89-~ 1.7.1         MultipleChoi~           1           7               1
## 6 f27eec8c-~ 1.7.1         MultipleChoi~           1           7               1
## # ... with 4 more variables: response <chr>, cloze_response <lgl>,
## #   submitted_at <chr>, correct <chr>
```

There are several observations about this data set. Firstly, all the questions are multiple choice, and columns 'quiz_question' is the same as a combination of 'week_number', 'step_number' and 'quiz_number'. Secondly, 'cloze_response' filed has not data in it, thus can be negated. Finally, every attempt by a student is logged making the time, the responses that they selected and whether or not they answered the question correctly or not.

Since all the questions are multiple choice and the cloze_reposne field is empty, these can quite simply be negated. With reference to the quiz question fields, the 'quiz_question' field can be negated as the other field present the information in a more code friendly manner allowing us to pick and choose the relevance of each in the modeling stage .

In particularly it is important when reformatting quiz.responses that we keep reference to which run the data was from. Furthermore, for the reformatting of the quiz data, it is important to keep the entirely of the user id, or question number; therefore, if any further merging is required they are consistent with the rest of the big data set.

In run two onward, there is reference to team members

```
head(cyber.security.2_team.members)
```

```
## # A tibble: 6 x 5
##   id                             first_name last_name team_role    user_role
##   <chr>                          <chr>      <chr>     <chr>        <chr>
## 1 f27eec8c-eaf1-4e6a-90f0-d6d~ FIRST      LAST      host         organisation_a~
## 2 77454a73-6b8b-46a2-8dee-35f~ FIRST      LAST      host         organisation_a~
## 3 a4fa6f89-a596-4d00-9397-420~ FIRST      LAST      lead_educat~ organisation_a~
## 4 21d74c76-2b0d-4dfd-a252-f6d~ FIRST      LAST      educator     learner
## 5 3e58d103-57b3-4d46-ac62-69a~ FIRST      LAST      educator     learner
## 6 85ea97bb-17d6-4bf7-ad74-dfb~ FIRST      LAST      educator     learner
```

These essentially refer to hierarchical rolls in the course and despite potentially having an impact on the quiz responses, i.e., the admin testing the functionality of the course questions, the weighting on the overall results would be negligible. For this reason teams data frames were ignored.

Similarly, as the data mining goals only referred to the quiz questions all survey responses, video.stats, enrollment and step.activity files were not included in this analysis.

Data Quality

The quality of the data seems to be good, and within all the fields appear to have consistent formatting such as capitalisation, method of spacing. However, there are a few rows that have data missing, such as no student id. But, these fields can simply be negated in the data preparation.

```
sum(cyber.security.1_question.response$learner_id == "")
```

```
## [1] 401
```

As seen there are 401 empty fields, however, as a percentage of the overall rows this is very low at 0.5%

```
sum(cyber.security.1_question.response$learner_id == "" )/
  length(cyber.security.1_question.response$learner_id )*
  100
```

```
## [1] 0.5207657
```

Finding abnormalities in the data is quite difficult, simply because of the way the data is presented. However, some variances may become more apparent after reformatting the data.

```
summary(cyber.security.1_question.response)
```

```
##   learner_id        quiz_question      question_type       week_number
##  Length:77002       Length:77002       Length:77002       Min.   :1.000
##  Class :character   Class :character   Class :character   1st Qu.:1.000
##  Mode  :character   Mode  :character   Mode  :character   Median :2.000
##                                                           Mean   :2.085
##                                                           3rd Qu.:3.000
##                                                           Max.   :3.000
##   step_number    question_number   response         cloze_response
##  Min.   : 7.00   Min.   :1.000   Length:77002       Mode:logical
##  1st Qu.: 7.00   1st Qu.:2.000   Class :character   NA's:77002
##  Median : 8.00   Median :3.000   Mode  :character
##  Mean   :11.57   Mean   :3.572
##  3rd Qu.:18.00   3rd Qu.:5.000
##  Max.   :19.00   Max.   :9.000
##  submitted_at         correct
##  Length:77002       Length:77002
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

3 Data Preperation

3.1 Select Data

For the preparation of the data a resultant data frame was desired that could be used to answer numerous variants of the data mining question. Therefore, when referring to the quiz.response csv file, `quiz_question` and `cloze_response` needed to be removed as they could provide no use to modeling. In addition, the time was in the format was presented in `YYYY-MM-DD HH:MM:SS UTC`. For use in this application we are simply interested in the time since the start of the course, and because this was to be standardised around the start of the course, it was easier to present the time in seconds. Therefore, when referencing all runs as one data frame they would be standardised around 0.

3.1.1 Preprocessing

At this stage of the of the data selection, the data was not displayed in a way that any real analysis could be performed on it to determine whether its significance or correlation was high. Therefore, the data was manipulated and was achieved within two functions: `cleanQuizData()` and `quizDataClean()`. Firstly, `cleanQuizData()`, this function took two arguments, the quiz data frame and the course start date, respectively. The long cumbersome heading - making for error prone coding and more difficult representation when plotting data frames. Because of this, they were abriviated. Finally, the date was accounted for and put into seconds - a more manageable format. However, since all the course runs were starting at different start dates the dates were standardised - see @ref(tab:dates). The function was carried out as part of the pre processing in the munging section, where all resultant data frames were stored in the cache. Annoyingly (but necessary), the learner id was incredibly long, but for cross-referencing purposes this was kept. The resultant cleaning from this function resulted in 7 data frames that were in the following format.

```r
quizStu <- function(quiz, courseStartDate){

  #convert the course start date to seconds
  cs = as.numeric(as.POSIXct(courseStartDate ))

  #renaming the df and its columns
  colnames(quiz) = c("id", "qq", "qt", "wn", "sn", "qn", "r", "cr", "t", "ans")

  #removing the columns that are not needed, or give no info
  quiz = select(quiz, -c(qt, cr))
  #displaying the date in seconds and removing substituting the date in which the course started
  quiz$t = as.numeric(as.POSIXct(quiz$t))-cs

  return(quiz)
}
```

Table 2: First two rows of quizStat1 data frame

id	qq	wn	sn	qn	r	t	ans
77454a73-6b8b-46a2-8dee-35f36b6c4fc1	1.7.1	1	7	1	1,2	-5232175	false
77454a73-6b8b-46a2-8dee-35f36b6c4fc1	1.7.1	1	7	1	1,2,3	-5230975	true

As seen in the above data frame the data frame names have been made more compact;

- id - the students id
- qq - quiz question
- wn - week number
- sn - section number
- r - responses
- t - time relative to the start of the course
- ans - was the response correct

It may be worth mentioning that the time is negative because they completed the quiz questions prior to the course even starting.

The seconds function `quizDataClean()` carried out the head duty work reformatting manipulating the data into a format which present some correlations. This function took the resultant data frame from `cleanQuizData()` and outputted a data frame that used the student id as the reference. For example, the function returned the number of answered given by a student (`numAns`); number of different question answered by the student (`numQues`); number of correct answers given (`numCorr`); time of the first and last question answered (`st` and `ft`, respectively).

```
quizStuPre <- function(quizStat){

  #create a data frame with unique user id
  quizData <- data.frame(id = unique(quizStat$id),
                         numAns="",
                         numQues="",
                         numCorr="",
                         ft="",
                         st="")

  for(i in 1:nrow(quizData)){
    count = 0 #count the number of occurrences (i.e. question attempts)
    count2 = 0 #reset the number of correct answers
    count3 = 0 # resets the number of different questions answered
    question = ""
    flag = 1

    #loops the number unique id values
    for(j in 1:nrow(quizStat)){

      if(quizData$id[i] == quizStat$id[j]){
        count = count+1
        if(flag == 1){
          quizData$st[i] = quizStat$t[j] #store the FIRST time student answered question
          flag = 0
        }
        if(quizStat$ans[j] == "true"){
          count2 = count2+1
        }
        if(quizStat$qq[j] != question){
          question = quizStat$qq[j]
          count3 = count3+1
        }
      }
    }
    quizData$numQues[i] = count3#store the number of different questions answered
    quizData$numCorr[i] = count2 # store the number of correct answers
    quizData$numAns[i] = count #store the number of attempts
    quizData$ft[i] = quizStat$t[count] #store the LAST time student answered question

  }

  return(quizData)
}
```

Due to the heavy computational requirements of this function, it is part of the pre-processing, stored in the munge folder. As seen below, the format of this is much more like what we would expect in order to question the relationships each field has.

Table 3: First two rows of quizStuPre1 data frame

id	numAns	numQues	numCorr	ft	st
77454a73-6b8b-46a2-8dee-35f36b6c4fc1	39	17	18	7831	-5232175
a4fa6f89-a596-4d00-9397-420a348c398d	40	19	19	7859	-4805410

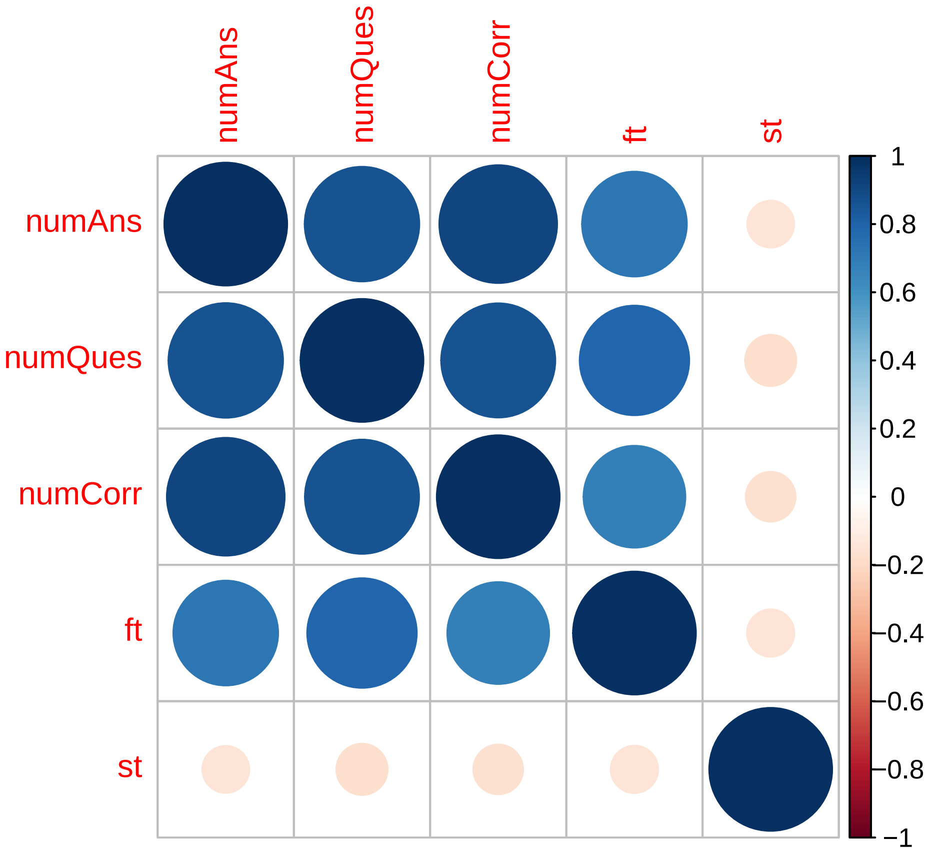
It is clear that these number appear to be more user friendly for a data analyst and correspond to the following;

- id - the students id
- numAns - the total number attempts at all questions
- numQues - the total number of questions answered
- numCorr - the number of correct answers provided
- ft - final time a question was answered
- st - first time a question was answered

3.1.2 Relationships

To gain an understanding of the data a correlation matrix was plotted for run one (excluded the student id).

```
  quizStuPre1 <- select(quizStuPre1, -c(id))
  quizStuPre1 <- as.data.frame(sapply(quizStuPre1, as.numeric))
  corrplot(cor(quizStuPre1), method="circle")
```



In this plot there is some strong correlation particularly in `numCorr` vs `numAns` and we can also see that there is a negative correlation in all of the start time `st`. Below are the highest correlations, ranked in order.

1. Question Attempts vs Correct answers (0.92)
2. Question Attempts vs Questions answers (0.87)
3. Correct answers vs Questions answers (0.86)

3.2 Clean Data

```
summary(quizStuPre1)
```

```
##      numAns          numQues         numCorr             ft
##  Min.   :  1.00   Min.   : 1.00   Min.   :  0.00   Min.   :-5232175
##  1st Qu.:  9.00   1st Qu.: 6.00   1st Qu.:  6.00   1st Qu.:-2795990
##  Median : 17.00   Median :10.00   Median : 10.00   Median :    3204
##  Mean   : 22.58   Mean   :13.36   Mean   : 12.37   Mean   :-1106865
##  3rd Qu.: 36.00   3rd Qu.:22.00   3rd Qu.: 20.00   3rd Qu.:    7287
##  Max.   :401.00   Max.   :22.00   Max.   :236.00   Max.   :   44231
##        st
##  Min.   :-5232175
##  1st Qu.:  112605
##  Median :  306961
##  Mean   :  523251
##  3rd Qu.:  724466
##  Max.   : 2893905
```

To gain a full understanding of the paired relationships that our constructed data presented a pairs plot of the data frame was plotted for the run1 set of data. it apparent that there was still some anomalies when first plotted presenting us from seeing the full scope of the pairs plots - this was seen in `numCorr` and `numAns`, so they were removed through the following code.

```
quizStuPre1 <- quizStuPre1[!(quizStuPre1$numAns > 75), ]
quizStuPre1 <- quizStuPre1[!(quizStuPre1$numCorr > 22), ]
```

Note:

- There was only 22 questions so some student might have answered the question more than once).
- The same arguments must be carried out on the other runs to ensure consistency.

This reduced any change of skewing data. this was decided to be any value >75 for the number of answers and 22 for the number of correct answers.

```
summary(quizStuPre1)
```

```
##      numAns         numQues         numCorr            ft
##  Min.   : 1.00   Min.   : 1.00   Min.   : 0.00   Min.   :-5232175
##  1st Qu.: 9.00   1st Qu.: 6.00   1st Qu.: 6.00   1st Qu.:-2795990
##  Median :17.00   Median :10.00   Median :10.00   Median :    3204
##  Mean   :22.44   Mean   :13.33   Mean   :12.27   Mean   :-1110810
##  3rd Qu.:36.00   3rd Qu.:22.00   3rd Qu.:20.00   3rd Qu.:    7287
##  Max.   :67.00   Max.   :22.00   Max.   :22.00   Max.   :   14975
##        st
##  Min.   :-5232175
##  1st Qu.:  112605
##  Median :  307068
##  Mean   :  523730
##  3rd Qu.:  725191
##  Max.   : 2893905
```

3.3 Construct Data

From the results in the previous stage the following derivations were compiled that may be of use for an analysis. This consisted of

- dt - the change in time between the first and last question
- acc - ratio of correct to false answers given
- scr - ratio of correct answers against to different questions
- tot - percentage of questions completed

All the above fields were added through quizStuCon() function where the resultant data frames were names `quizStuConX` where `X` goes from `1:7`.

```
quizStuCon <- function(quizData){
  quizData <- data.frame(quizData,
              tot = (as.numeric(quizData$numQues) / max(as.numeric(quizData$numQues))),
              dt <- Mod((as.numeric(quizData$ft) - as.numeric(quizData$st))),
              acc <- (as.numeric(quizData$numCorr)/as.numeric(quizData$numAns)),
              scr <- (as.numeric(quizData$numQues)/as.numeric(quizData$numAns))
              )
  return(quizData)
}
```

Furthermore, the data frame consisted of `char` values that needed to be transformed into `nums` variables.

```
#converts the df variables to a num other than id
dfToNum <- function(data){
  df <- data
  df <- select(df, -c(id))
  df <- as.data.frame(sapply(df, as.numeric))
  df <- data.frame(data$id, df )
  return(df)
}
```

It made sense to scale all the time fields as their value was already scaled around 0 and not the actual data. This removed complication of the larger numbers that were harder to digest. Again this was run for all 7 `quizStuConX` df's.

```
quizStuCon1$dt <- scale(quizStuCon1$dt)
quizStuCon1$ft <- scale(quizStuCon1$ft)
quizStuCon1$st <- scale(quizStuCon1$st)
```

Below shows the resultant data frame for run one, i.e. `quizStuCon1`.

Table 4: The constructed data frame (excluding id)

numAns	numQues	numCorr	ft	st	tot	dt	acc	scr
39	17	18	0.2604139	-10.188771	0.7727273	3.570679	0.4615385	0.4358974
40	19	19	0.2604399	-9.421665	0.8636364	3.233106	0.4750000	0.4750000

3.4 Interogate Data

After all the relevant transformation were made to the quiz response df's, now required us to combine all the runs into one final data frame that could be used to model the data. From the scatter plot matrix provided before, showed weak correlation in the start times `st`, thus these were removed. The student `id` was also removed since we no longer need reference this. Furthermore, when merging the df's, it was important to retain the run that the data originated from, and was done as follows.

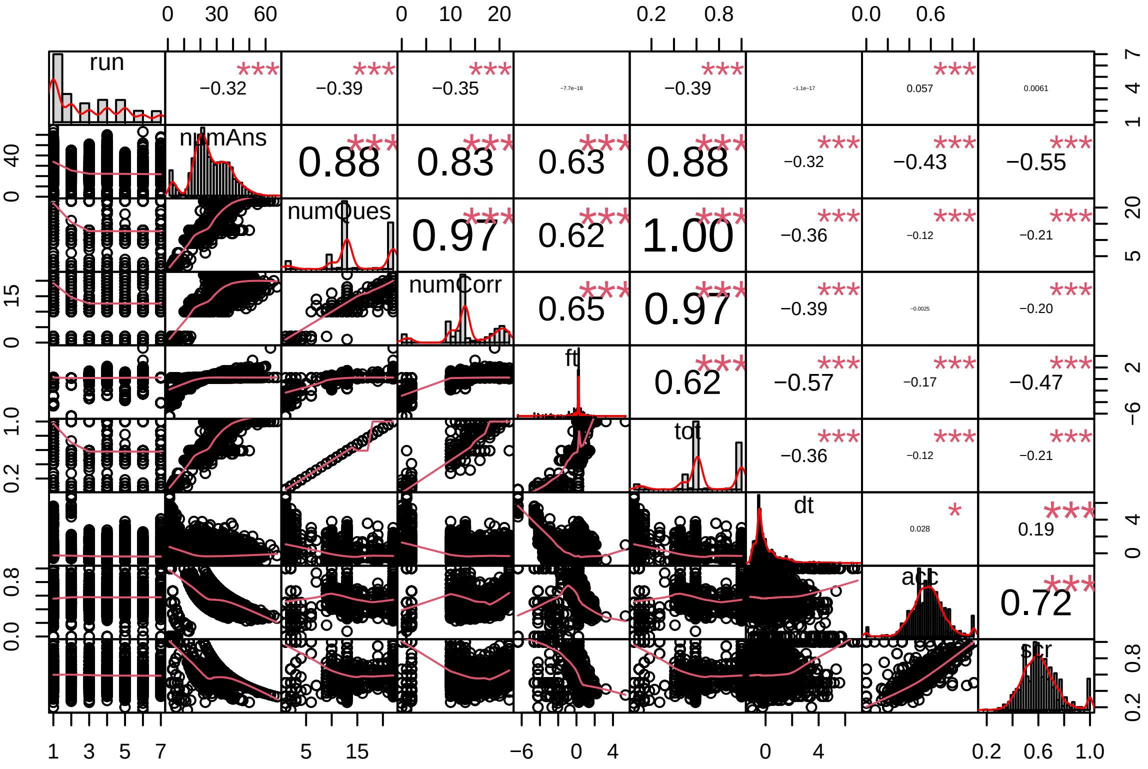
```
#Merge all quiz data df's

df1 <- data.frame(run=1, select(quizStuCon1, -c(st, id)))
df2 <- data.frame(run=2, select(quizStuCon2, -c(st, id)))
df3 <- data.frame(run=3, select(quizStuCon3, -c(st, id)))
df4 <- data.frame(run=4, select(quizStuCon4, -c(st, id)))
df5 <- data.frame(run=5, select(quizStuCon5, -c(st, id)))
df6 <- data.frame(run=6, select(quizStuCon6, -c(st, id)))
df7 <- data.frame(run=7, select(quizStuCon7, -c(st, id)))

quizStuMod <- rbind( df1,  df2, df3, df4, df5, df6, df7)
```

The pairs plot showed several fields which seemed to have a linear correlations, where the `tot` seemed to presented the most linear correlations. referring back to the business objective, it would be

4 Modeling

In this modeling stage there will be no specific modeling techniques used, but rather an exploratory analysis, thus another scatter plot matrix was plotted which hosted all the additional fields added in the data construction stage

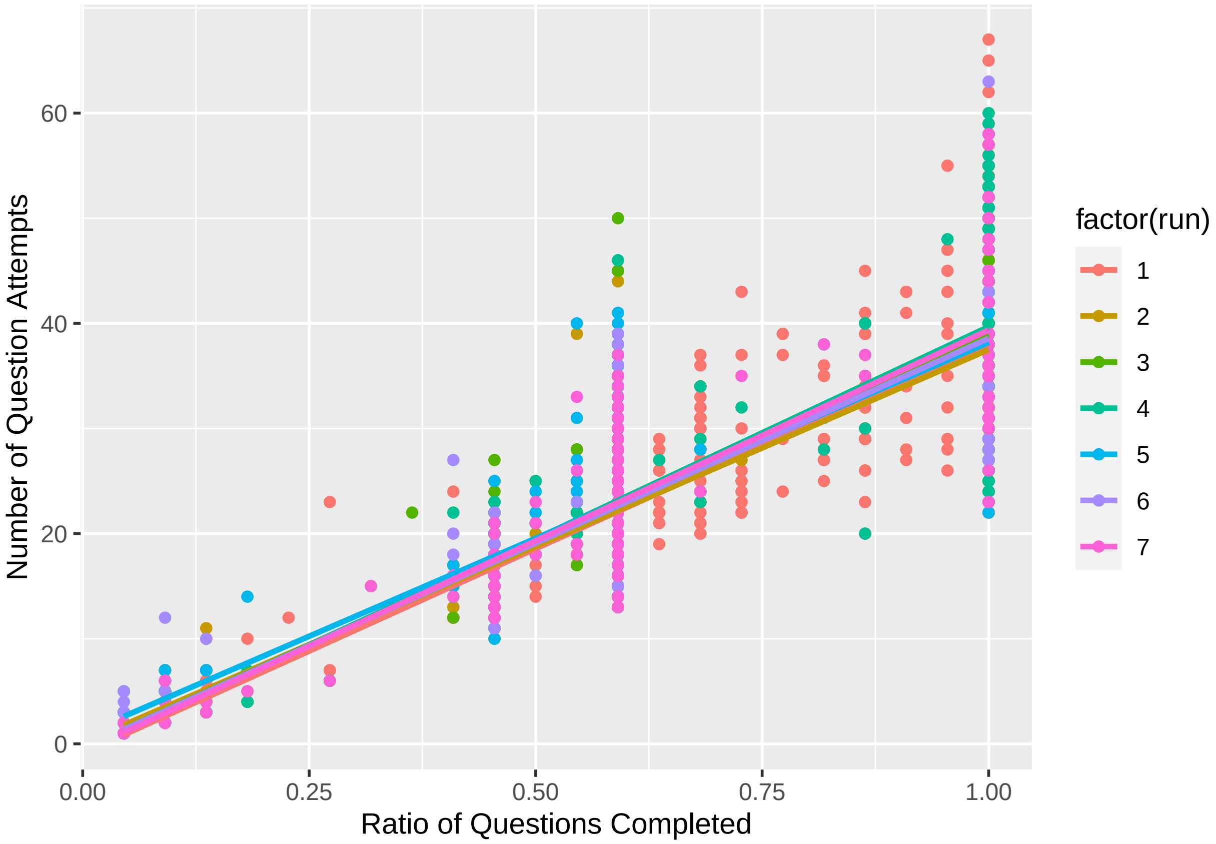
```
chart.Correlation(quizStuMod, histogram=TRUE, pch=19)
```



The data mining goal looked at how the quiz question interaction varies over the number of runs that the course was run. Therefore, it would interesting to see how the total percentage of the course questions completed vs the number of correct answers provided matched up across all seven runs, as these had a strong correlation. The resultant fields were plotted in a scatter plot was plotted with a linear line of correlation

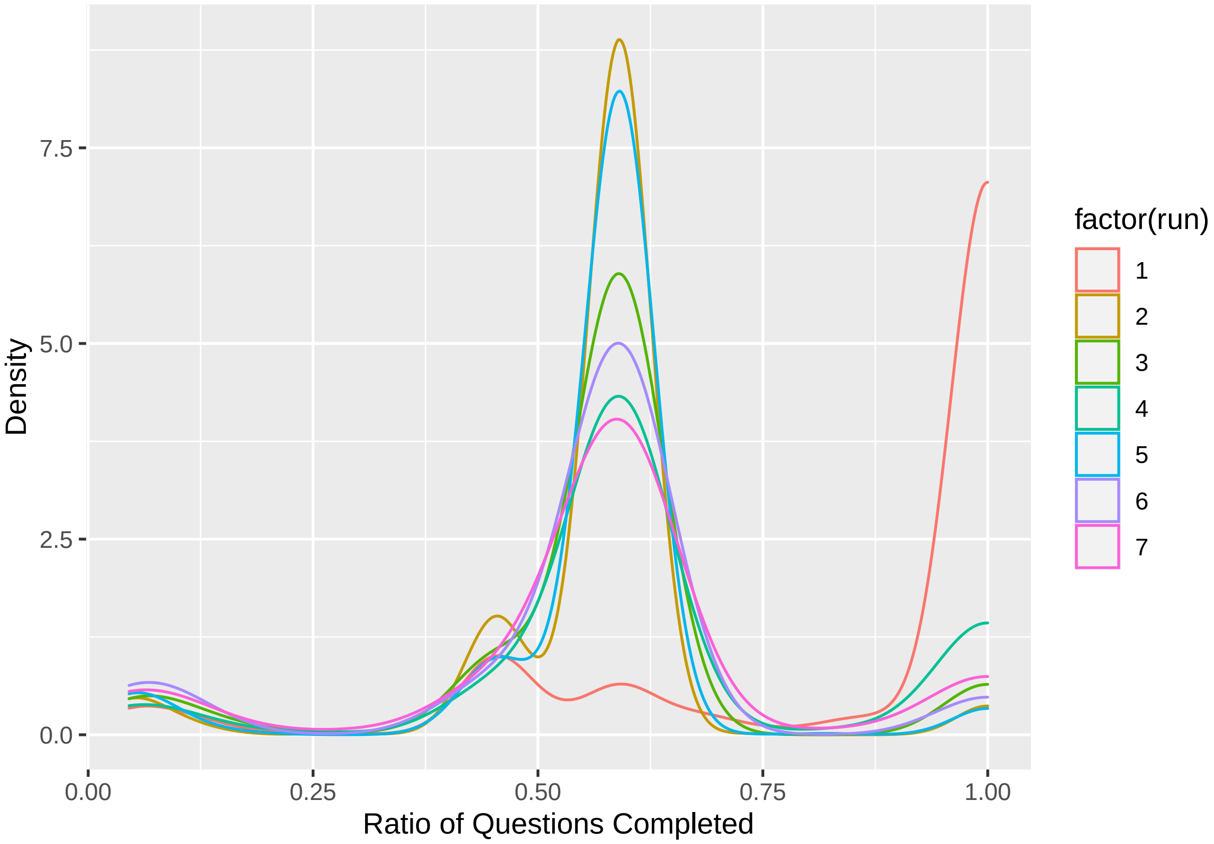
```
#plot a scatter plot with linear line of correlation between runs
ggplot(quizStuMod, aes(x = tot, y = numAns, col = factor(run))) +
  geom_point() +
  stat_smooth(method = "lm", se=F) +
    xlab("Ratio of Questions Completed") +
    ylab("Number of Question Attempts")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



There is a clear linear relationship between the percentage of course questions completed and the number of answers given. Therefore, we can say that the error rate in the questions remains the same throughout, and from the correlation graph before we can say as the percentage of the course increase as does the number of correct answers, the number of questions completed and the number of attempts. But, also we can people who complete >60% of the course are more likely to complete 100% of the questions. Maybe they feel invested? But in terms of density, what is the spread of total course questions being completed by the students.

```
# Plot the density by runs
ggplot(quizStuMod, aes(x=tot, color=factor(run))) +
  geom_density() +
  xlab("Ratio of Questions Completed") +
  ylab("Density")
```



The results of this is incredibly surprising, other than run one there is very few students that are competing the course, with the majority of the dropout appearing at the same percentage of questions completed year by year. There is some catalyst event occurring at this point that is triggering people to no longer participate in the course, particularly the larger spike occurring after the students have completed over 50% of the course. The following table highlights this as a percentage of students completing more than 75% of the questions for the number of times this course was run.

Table 5: Summary of start and end dates for each run

Run	Percentage
1	74.91
2	3.20
3	7.84
4	21.68
5	3.41
6	6.98
7	13.07

Again, this shows that this varies somewhat randomly.

5 Data Mining

5.1 Data Mining Goals

To try and identify the reasoning why the majority of students were only completing 55%-60% of the course question, the data needed to be re manipulated to allow the modeling to quiz whether students lack of participation was occurring at a particular question or week in the course. This time, instead of taking the student id as the subject, the question number was.

6 Data Preperation

For all the data preparation code please refer to `preprocessing2.R` file stored in the scr folder withing the project files. ## 6.1 Pre-Processing Since this was using the same data sets as before most the pre-processing was already complete. The `quizQuePre()` function was run on before taking the argument from the earlier found `quizStat` data frames.

```r
quizQuePre <- function(quizStat){

  #create a dataframe with unique user qq
  quizData <- data.frame(qq = unique(quizStat$qq),
                         numAns="",
                         numStu="",
                         numCorr="",
                         wn="",
                         sn="",
                         qn="")

  for(i in 1:nrow(quizData)){
    count = 0 #count the number of occurrences (i.e. question attempts)
    count2 = 0 #reset the number of correct answers
    count3 = 0 # resets the number of different questions answered
    student = ""
    flag = 1

    #loops the number unique qq values
    for(j in 1:nrow(quizStat)){

      if(quizData$qq[i] == quizStat$qq[j]){
        count = count+1
        if(flag == 1){
          quizData$wn[i] = quizStat$wn[j] #store the FIRST time student answered question
          quizData$sn[i] = quizStat$sn[j]
          quizData$qn[i] = quizStat$qn[j]
          flag = 0
        }
        if(quizStat$ans[j] == "true"){
          count2 = count2+1
        }
        if(quizStat$id[j] != student){
          student = quizStat$id[j]
          count3 = count3+1
        }
      }
    }
    quizData$numStu[i] = count3#store the number of different questions answered
    quizData$numCorr[i] = count2 # store the number of correct answers
    quizData$numAns[i] = count #store the number of attempts


  }

  return(quizData)
}
```

The resultant data frame for run one can be seen below, where all runs were stored in the munge file.

Table 6: First two rows of quizQuePre1 data frame

qq	numAns	numStu	numCorr	wn	sn	qn
1.7.1	5019	3443	3172	1	7	1
1.7.2	4468	3316	3167	1	7	2

It is clear that these number appear to be more user friendly for a data analyst and correspond to the following;

- qq - question
- numAns - the total number attempts for the question

- numStu - the total number of students that answered the question
- numCorr - the number of correct answers provided for the question

- wn - week number
- sn - section number
- qn - question number

6.2 Construct Data

The addition of the ratio of student that completed the qustion, the accuracy of the reuslts for each queestion and the ratio of students to answers we derived and added to the data frame using the `quizQueCon()` function.

```
quizQueCon <- function(quizData) {
  quizData <- data.frame(quizData,
                         tot = (as.numeric(quizData$numStu) / max(as.numeric(quizData$numStu))),
                         acc = (as.numeric(quizData$numCorr)/as.numeric(quizData$numAns)),
                         scr = (as.numeric(quizData$numStu)/as.numeric(quizData$numAns))
                         )
  return(quizData)
}
```

The resultant df was then transformed converting the numeric values from `char` to `num` variables - essentially all columns other than the quiz questions `qq`.

```
#converts the df variables to a num other than id
dfToNum <- function(data){
  df <- data
  df <- select(df, -c(qq))
  df <- as.data.frame(sapply(df, as.numeric))
  df <- data.frame(qq=data$qq, df )
  return(df)
}
```

Since each run of the course varied with the number of students quite significantly, the number of students was normalised for each run

```
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

quizQueCon1$numStu <- normalize(quizQueCon1$numStu)
```

6.3 Interrogating Data

All seven runs were combined into one function, making sure to keep note of the run that the originated from

```
# run one was ignored as the section question varied from the other
df1 <- data.frame(run=1, quizQueCon1)
df2 <- data.frame(run=2, quizQueCon2)
df3 <- data.frame(run=3, quizQueCon3)
df4 <- data.frame(run=4, quizQueCon4)
df5 <- data.frame(run=5, quizQueCon5)
df6 <- data.frame(run=6, quizQueCon6)
df7 <- data.frame(run=7, quizQueCon7)
quizQueMod <- rbind( df1, df2,df3, df4, df5, df6, df7)
```

```
  # Histogram plot
#  ggplot(df, aes(x=tot, color=run, fill=run)) +
#    geom_histogram(aes(x=tot), alpha=0.3, position="identity") +
#                  xlab("Ratio of Questions Completed") +
#                  ylab("Number of Students")
```

Assuming that this is time related, and going back to the data construction stage I wanted to format the data and see if we can correlate this to a specific question.

4.1 Selecting Modelling Technique

#Evaluation

#Deployment