# CSC8631 Project

Luke Battle

Student No: 200725640

## Introduction/Business Understanding

In this report, we will use the CRISP-DM method to analyse a data set. This will involve us running through several key steps; *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modelling*, *Evaluation* and *Deployment*.

## Business Understanding

The first step in performing our CRISP-DM analysis is to gain an understanding of the business motivation for the analysis. Our primary goal in this case is to enhance Newcastle University's online resources. Specifically, for the past several years, Newcastle University has ran a free online course called *Cyber Security: Safety at Home, Online, in life*. Several data sets have been compiled each time the course has been ran, detailing several facets of the user experience throughout the course. Newcastle University seeks to utilise this data to enhance the quality of material taught outside of the classroom, and ultimately promote learner engagement with the course and Newcastle's wider online resources. To achieve this, different elements of the collected data will be examined over consecutive runs, with the aim of identifying any trends both within and between them that may indicate an area where user engagement could be improved.

## Data Understanding

Next, before beginning our analysis it is important to fully assess the data, so that we can make informed decisions about the portions of the data that should constitute the foundations of our analysis. In total, Newcastle University has provided 53 csv files that describe a range of features of user interaction with the MOOC over 7 runs of the course. These runs span from September 2016 to September 2018. For every run there is a file detailing a different property, with a maximum of 8 per run. All runs have the files "Archetype-survey-reponses", "Enrolments", "Leaving Survey Response", "Question Response", "Step Activity" and "Weekly Sentiment Survey Response". In addition, from run 2 to 7 a file on the "Team Members" is included, and from run 3 to 7 there is also a file on the "Video Stats". However, the data quality within some of these files is quite poor. Using custom function **check_empty()** (note that this function and all custom functions are located in the lib subdirectory in the helpers.r file), we can check the emptiness of these files:

```r
check_empty(dir("data",pattern = "archetype-survey-responses"))
```

```
## [1] "2 out of 7 files are empty"
```

```r
check_empty(dir("data",pattern = "leaving-survey-responses"))
```

```
## [1] "3 out of 7 files are empty"
```

```r
check_empty(dir("data",pattern = "weekly-sentiment-survey-responses"))
```

```
## [1] "4 out of 7 files are empty"
```

Therefore, in the files labelled “archetype-survey-responses”, “leaving-survey-responses” and “weekly-sentiment-survey-responses” we can observe that there are several that are totally blank. The inconsistency of data in these files means they might not be the best candidates for the focus of the data analytics pipeline, and so will not feature in this analysis. The “enrolments” data set does contain data, however the quality of it is poor. For 88.02% of the individuals in the data set, the only data available is the detected country of origin and enrollment date. These individuals are missing useful demographic indicators gender, age range, education level and employment status. Due to the poor quality of the data this will not be used in the analysis. Out of the remaining data sets, a quick look at the video-stats and question-response files suggests that they may warrant further examination and could provide a route to find areas of improvement in the MOOC course.

### Describe Data

The video stats files are included for course runs 3 to 7. We can immediately observe that there are no empty files. Note that, upon loading this file, all video-stats file names have been saved in vector video_files, so we will use this when checking through them all:

```r
check_empty(video_files)
```

```
## [1] "0 out of 5 files are empty"
```

Using the video-stats file from run 3 as an example, let’s check the dimensions of the files:

```r
dim(video_file_raw)
```

```
## [1] 13 28
```

Using custom function **check_dimensions()**, we can check if this is the case for all video-stats files using the following code:

```r
check_dimensions(video_files, dim(video_file_raw))
```

```
## [1] "All files have same dimensions"
```

Therefore, each video-stats file has the same dimensions. The example file from the third run of the course has the following variables:

```r
colnames(video_file_raw)
```

```
##  [1] "step_position"                  "title"
##  [3] "video_duration"                 "total_views"
##  [5] "total_downloads"                "total_caption_views"
##  [7] "total_transcript_views"         "viewed_hd"
##  [9] "viewed_five_percent"            "viewed_ten_percent"
## [11] "viewed_twentyfive_percent"      "viewed_fifty_percent"
## [13] "viewed_seventyfive_percent"     "viewed_ninetyfive_percent"
## [15] "viewed_onehundred_percent"      "console_device_percentage"
## [17] "desktop_device_percentage"      "mobile_device_percentage"
## [19] "tv_device_percentage"           "tablet_device_percentage"
## [21] "unknown_device_percentage"      "europe_views_percentage"
## [23] "oceania_views_percentage"       "asia_views_percentage"
## [25] "north_america_views_percentage" "south_america_views_percentage"
## [27] "africa_views_percentage"        "antarctica_views_percentage"
```

Now, using custom function **check_variables()** we can see that every video-stats file has these variables in the same order.

```
check_variables(video_files, colnames(video_file_raw))
```

```
## [1] "All files have same variables"
```

This means that we can easily compile all of these files into a single master file without any data ended up in the incorrect position. We will do this in the *Data Preparation* phase of our analysis, and this will enable us to more easily perform detailed analysis on the video-stats data set as we will not have to loop through all of the files.

To further our understanding of this data. we can now check that each individual in the file is given by the video and step_position that the variables relate to. These are the following:

```
video_file_raw[,c("title","step_position")]
```

```
##                                                     title step_position
## 1                                   Welcome to the course          1.10
## 2                        Why would anyone want your data?          1.14
## 3  Preserving privacy in cloud storage: privacy by design          1.17
## 4              Staying safe online: personal perspectives          1.19
## 5                              Privacy online and offline          1.50
## 6                     Welcome to Week 2: payment security          2.10
## 7            Exploring vulnerabilities in online payments          2.11
## 8                  The million dollar contactless payment          2.17
## 9              The evolving arms race of payment security          2.40
## 10         Welcome to Week 3: security in the future home          3.10
## 11           Exploring security: biometric authentication          3.14
## 12        Exploring security: the Access Control Live Lab          3.15
## 13                             Devices in the future home          3.20
```

Using custom function **check_rows()**, we can confirm that the title and step_position are the same in each video-stats file.

```
check_rows(video_files,video_file_raw$title,which(colnames(video_file_raw)=="title"))
```

```
## [1] "All files have same rows"
```

```
check_rows(video_files,video_file_raw$step_position,which(colnames(video_file_raw)=="step_position"))
```

```
## [1] "All files have same rows"
```

An additional aspect of the video stats files that is important to consider for our analysis is whether the videos used in the course are the same for each run. If they are not, we could not fairly compare the efficacy of each video against each other when considering the results across runs. However, by examining the "video_duration" variable in each file we can see that this is identical for each video in every run, so we can assume for the purposes of this analysis that the videos have not been changed between runs. We can confirm this with the below code, once again using function **check_rows()**:

```
check_rows(video_files,video_file_raw$video_duration,which(colnames(video_file_raw)=="video_duration"))
```

```
## [1] "All files have same rows"
```

For the question-response data set, we can compute the same high level analysis to improve our data understanding. We will perform the same checks as with the video-stats files. This time we will use question_files as our pre-prepared vector of question-response file names, and question_file_raw as an example file from the first run of the course.

```
check_empty(question_files)
```

```
## [1] "0 out of 7 files are empty"
```

```
dim(question_file_raw)
```

```
## [1] 77002    10
```

```
check_dimensions(question_files, dim(question_file_raw))
```

```
## [1] "Not all files have same dimensions"
```

```
colnames(question_file_raw)
```

```
##  [1] "learner_id"      "quiz_question"   "question_type"   "week_number"
##  [5] "step_number"     "question_number" "response"        "cloze_response"
##  [9] "submitted_at"    "correct"
```

```
check_variables(question_files, colnames(question_file_raw))
```

```
## [1] "All files have same variables"
```

So, we can see that none of the question-files are empty, they each have the same 10 variables in the same order but do not have the same number of rows in each file. We can observe that the number of rows in each of these files as related to the learner_id variable, which will vary between runs depending on the number of learners taking part in each course run. However, as the variables are all the same and in the same order, we can easily compile the question-response data files into a single file to compute further analysis on it.

### Data Quality

We can now assess the data quality of each of these data sets using the **diagnose()** function from the**dlookr** package to see if there is any missing data in each file. Using custom function **multiple_data_quality()** we can apply this across multiple files and compute the total count of missing data for both the video-stats files and question-response files:

```
multiple_file_quality(video_files)
```

```
## [1] "Total Missing Count: 0"
```

```
multiple_file_quality(question_files)
```

```
## [1] "Total Missing Count: 176463"
```

Therefore we can observe that there is no missing data in the video-stats files, however a considerable amount missing in the question-response files. To investigate where this is occurring, we can first use the **diagnose()** function on a single data question-response data file.

```
diagnose(question_file_raw)
```

```
## # A tibble: 10 x 6
##    variables      types   missing_count missing_percent unique_count unique_rate
##    <chr>          <chr>           <int>           <dbl>        <int>       <dbl>
##  1 learner_id     charac~             0               0         3410   0.0443
##  2 quiz_question  charac~             0               0           22   0.000286
##  3 question_type  charac~             0               0            1   0.0000130
##  4 week_number    integer             0               0            3   0.0000390
##  5 step_number    integer             0               0            5   0.0000649
##  6 question_numb~ integer             0               0            9   0.000117
##  7 response       charac~             0               0           32   0.000416
##  8 cloze_response logical         77002             100            1   0.0000130
##  9 submitted_at   charac~             0               0        75103   0.975
## 10 correct        charac~             0               0            2   0.0000260
```

The results indicate that all of the data for the “cloze_response” variable is missing. We can examine if this is where the data is missing in all of our question-response files with the following code by counting the NA values within the cloze_response variable for each file.

```
missing_data = numeric(length(question_files))
  for (i in 1:length(question_files)) {
    temp_file = read.csv(paste("data/",question_files[i],sep = ""))
    missing_data[i] = sum(is.na(temp_file$cloze_response))
  }
sum(missing_data)
```

```
## [1] 176463
```

So the amount of data missing in the cloze_response variable in each file is equal to the total data missing across all question-response files. As this is the only part of the file with missing data, we should remove this variable when we prepare our data files for analysis. Aside from this variable, the remainder of the data is all present, and we can now begin to prepare our data.

## Data Preparation

As we have verified in the previous section that all of the video-stats files have the same variable order with no missing data, to prepare the data we can simply stack the data from each file. I initially performed this by creating an empty object and binding each data frame under the preceding one, whilst looping over the files in the data subdirectory. Whilst this did create the desired file containing all of the video-stats data, the method involved incremental object growth. This means that if this data pipeline was used on significantly more video-stats data files in the future, the data preparation phase would rapidly become computationally heavy. We can refine this process by instead inputting the data from each file into a list with pre-defined length, then using function **bind_rows()** from the *dplyr* package so that we can completely avoid incremental object growth. The code from this is shown below:

```r
#create empty list with length equal to amount of video files
video_list = vector("list", length(video_files))

#loop through video files, assign corresponding run to data then add to appropriate
#position in previously defined list
for (i in 1:length(video_files)) {
  video_stats = read.csv(paste("data/",video_files[i],sep = ""))
  Run = rep(video_files_numeric[i], nrow(video_stats))
  video_stats = cbind(video_stats, Run)
  video_list[[i]]=video_stats[,c("Run", "title", "step_position", "video_duration",
                                 "viewed_five_percent","viewed_ten_percent",
                                 "viewed_twentyfive_percent","viewed_fifty_percent",
                                 "viewed_seventyfive_percent", "viewed_ninetyfive_percent",
                                 "viewed_onehundred_percent")]
}

#use bind_rows to take data in list and create data frame, then add new column that finds
#difference between the percentage of learners that viewed 5% and 95% of the videos
video_df = bind_rows(video_list) %>%
  mutate(percentage_drop_off = viewed_five_percent - viewed_ninetyfive_percent)
```

Due to the iterative relationship between the *Data Preparation* and *Modelling* phases that is inherent to CRISP-DM, other components in the above code such as the specific variable selection and introduction of derived attributes were not applied at first. My initial runs involved performing the compilation of data for all variables and then using the **group_by()** and **summarise()** functions from *dplyr* to create new summary data frames to examine and plot the impact of different variables. This allowed me to hone in on the specific variables of importance to the data analysis pipeline and remove the rest. Additionally, the data is still summarised but this is now done when creating the plots, as opposed to creating different data frames for different plots and in doing so cluttering the data analysis pipeline. The only exception to this is the data frame "highest_audience_drop", from which the videos with the highest percentage decrease in audience can be derived, as well as an informative table used within this report.

In the data preparation for the video-stats files, several new variables have also been added. The addition of the "Run" variable has been included in every iteration of this phase, as it would be intuitively useful in our subsequent analysis for us to include an identifier of what run of the course the data corresponds to. With the inclusion of this variable, I have taken care to ensure this was done in a fully reproducible manner with no user input required. The below code, located in 01-A.R in the munge subdirectory, takes the course run from the file name:

```r
#create empty vector of length equal to video files length
video_files_numeric = numeric(length(video_files))
```

```
#remove known patterns in file names to identify run number associated with
#each file in data subdirectory
for (i in 1:length(video_files)) {
  run_num = str_remove(video_files[i],"cyber-security-")
  run_num = str_remove(run_num, "_video-stats.csv")
  video_files_numeric[i] = as.integer(run_num)
}
```

Another variable addition that was added in later versions of this analysis was the "percentage_drop_off", calculated by finding the difference between the variables "viewed_five_percent" and "viewed_ninetyfive_percent". The motivation for this inclusion will be discussed when reviewing the analysis results.

The final step taken in our data preparation of the video-stats files, is to reformat the data frame from "wide" into "long" format using *dplyr* function **pivot_longer()**. This allowed us to plot all of the variables relating to the percentage of the video viewed together. An example of the final data frame can be seen below:

```
head(video_df_long)
```

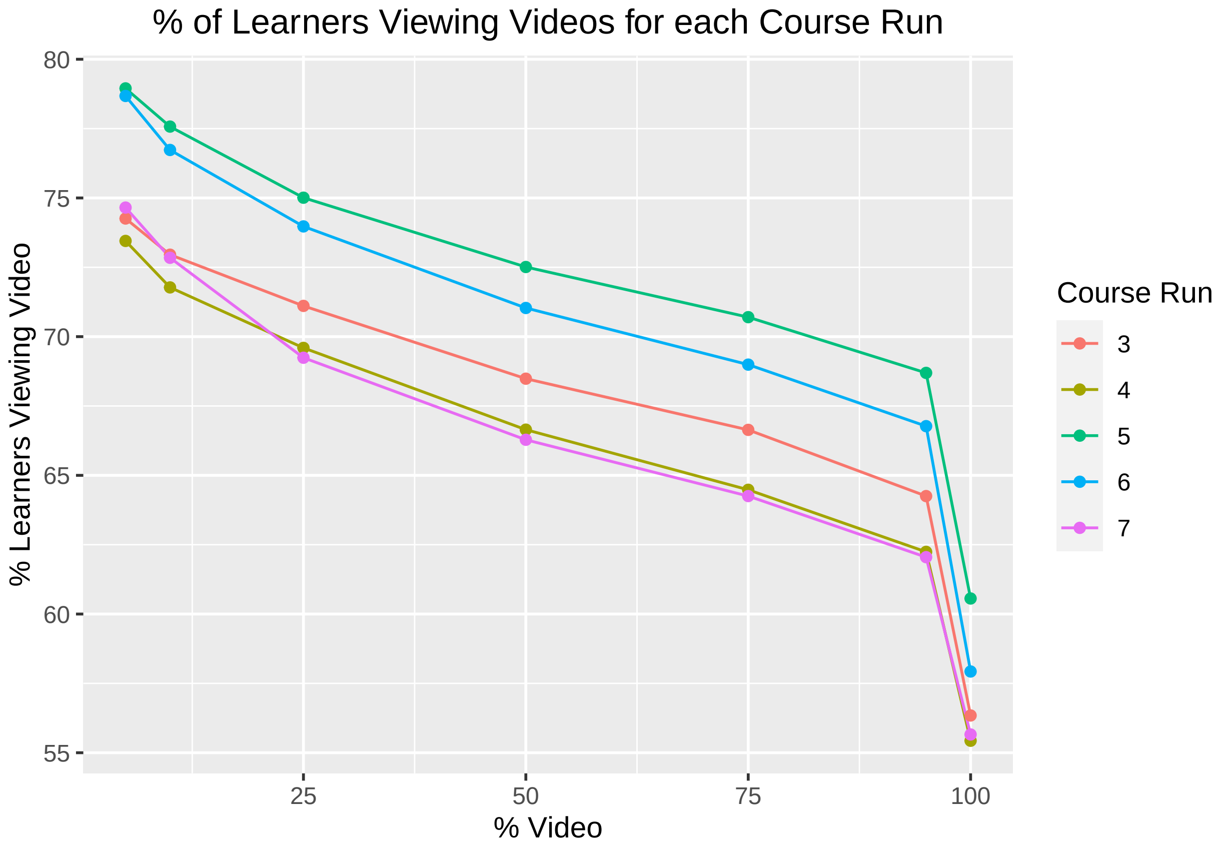
```
## # A tibble: 6 x 7
##     Run title         step_position video_duration percentage_drop_~ viewed value
##   <dbl> <chr>                <dbl>          <int>             <dbl>  <int> <dbl>
## 1     3 Welcome to ~           1.1             99              10.5      5  77.0
## 2     3 Welcome to ~           1.1             99              10.5     10  75.4
## 3     3 Welcome to ~           1.1             99              10.5     25  73.4
## 4     3 Welcome to ~           1.1             99              10.5     50  70.4
## 5     3 Welcome to ~           1.1             99              10.5     75  68.2
## 6     3 Welcome to ~           1.1             99              10.5     95  66.4
```

For the question response files, we can facilitate easier cross-run analysis by compiling all files into one, and adding an identifier labeling the run that the data pertains to. This can be performed in the same fashion as with the video-stats data frame, however we established in the previous section that the variable "cloze response" in the question-response data is empty in all files. We should therefore remove this variable from our final data set.

In order to calculate the percentage of questions answered correctly per run, it would be useful to transform the "correct" variable into a more calculation-friendly format. We will add a new column, "correct_binary", where if a question has been answered correctly this variable will contain a 1, or 0 for the converse. This will enable us to easily calculate the percentage of questions answered correctly within each run of the course. Using this, we can input create a small data frame reflecting the run and the corresponding percentage of questions answered correctly.

## Modelling

Now that both data sets are appropriately constructed, we will begin our analysis of the data. To motivate our analysis of the video-stats data, we should first assess if watching the videos can improve learning on the course. To achieve this, we will investigate if there is a relationship between the percentage of people who watched the videos in the course, and their results in the multiple choice quiz. To examine this, we will first plot the percentage of the videos viewed on average by learners on the course, and show how this varies per course run.

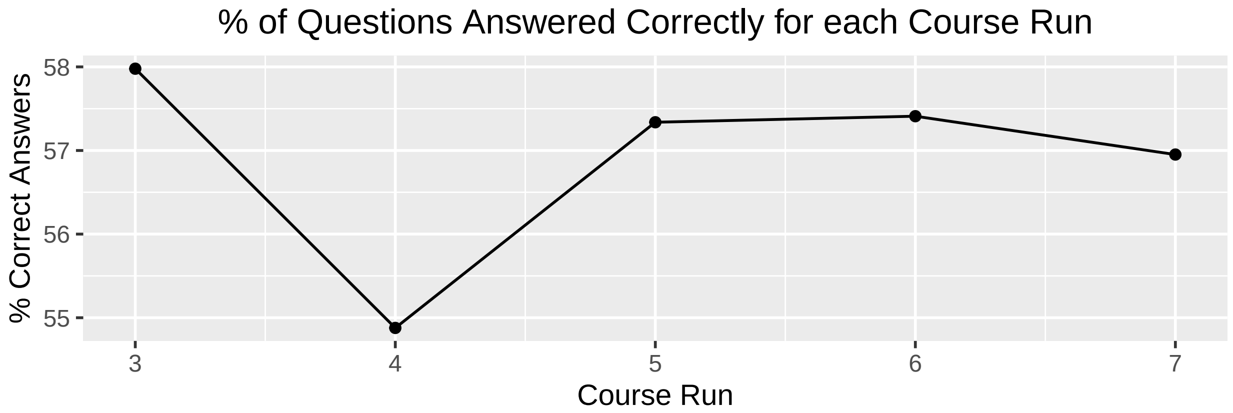


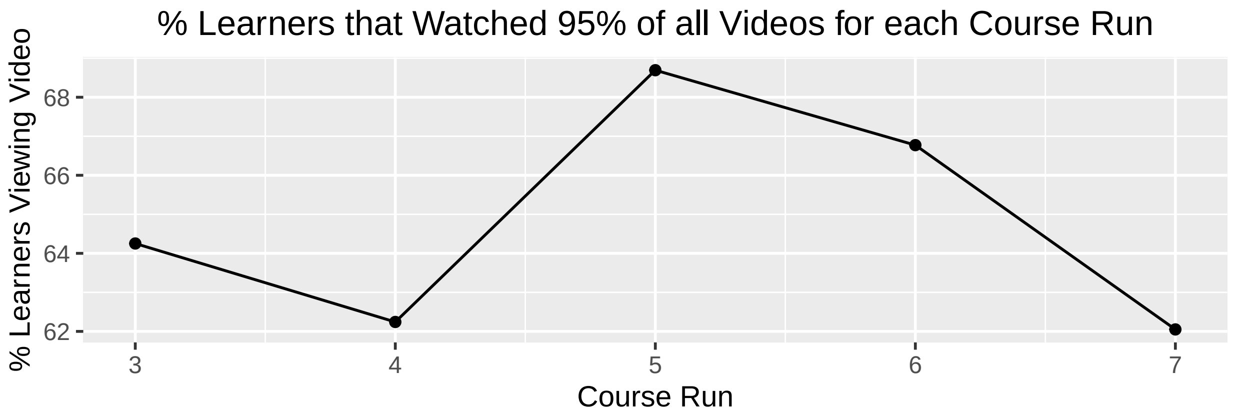
In the above plot, we can observe a steady decline of people watching the video as it progresses up until 95% of the video. A steep decline can then be seen after this to the percentage of learners on the course who viewed 100% of the video. This intuitively makes sense as many viewers may end the video when there are only a few seconds left when it is obvious that the the important information in the video has been given, and so would not register as having watched 100% of the video. This means that the percentage of people that viewed the entire video is not the best metric by which we can judge how many people watched the video. The percentage of users that watched 95% of the video would be a more suitable metric, as they have still effectively watched the entirety of the video just may have not watched the final few seconds.

We will now display our calculated values of the percentage of questions answered correctly for the each run of the course.

| Run | % Correct |
|---|---|
| 1 | 54.78 |
| 2 | 58.67 |
| 3 | 57.98 |
| 4 | 54.88 |
| 5 | 57.34 |
| 6 | 57.41 |
| 7 | 56.95 |

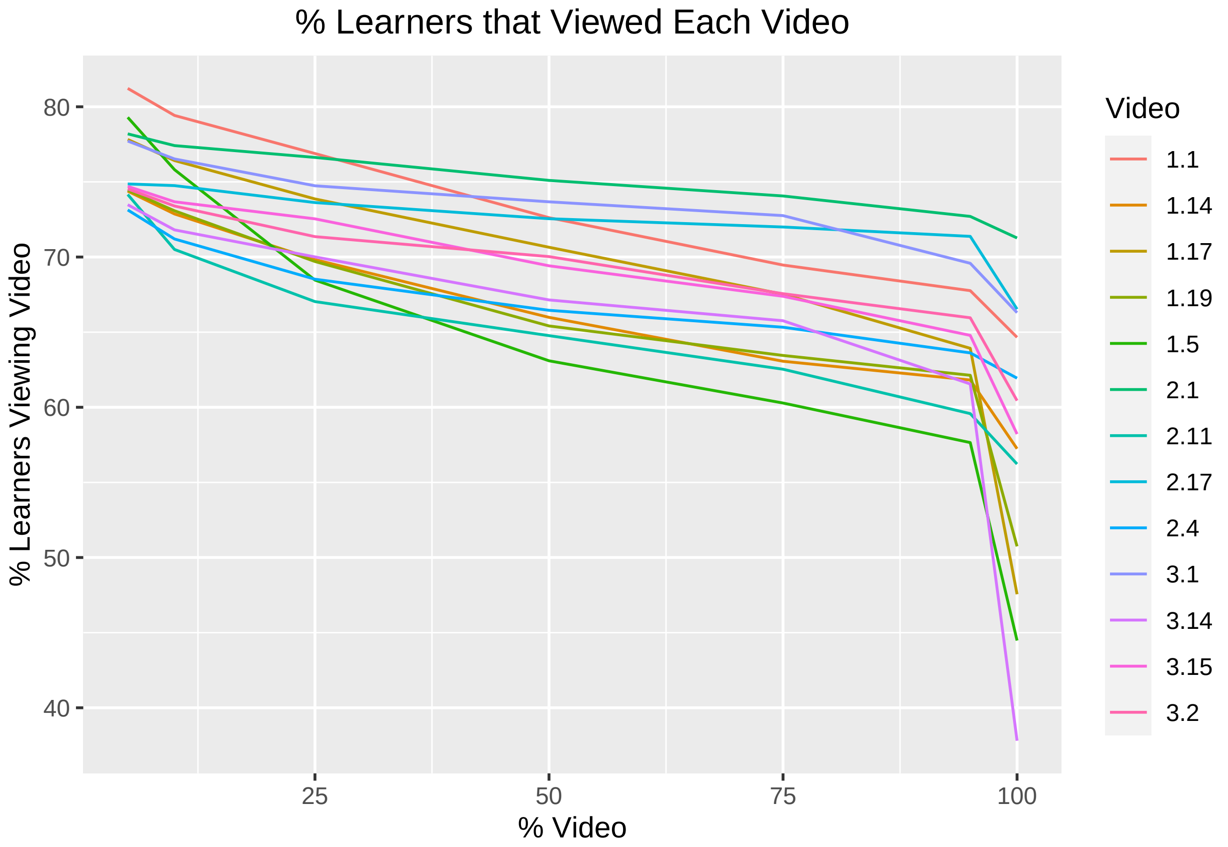
To compare this to the percentage of people who watched the videos we will now visualise the percentage of users that viewed 95% of all of the videos on average for each run of the course, and compare this to a plot of how many questions were answered correctly for each course iteration:



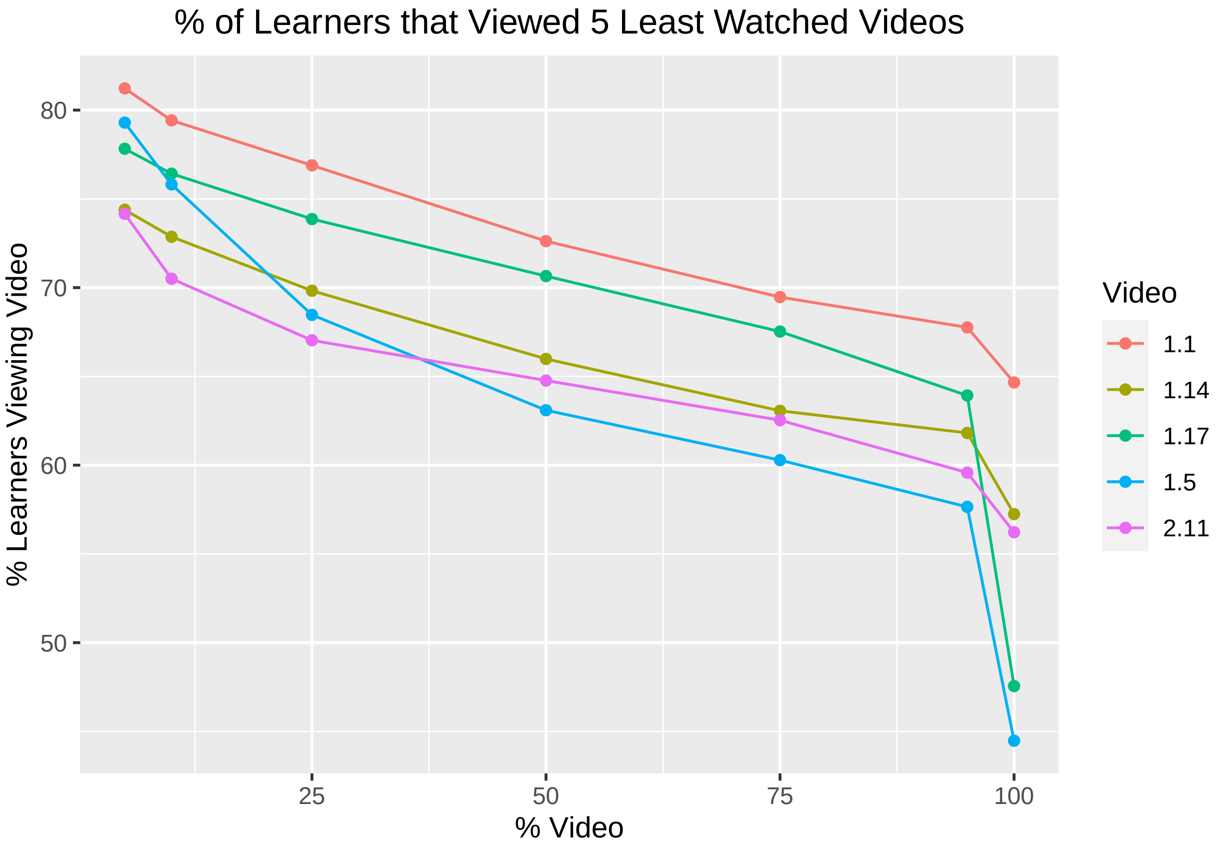


From this plot we can see clearly that in the fourth run of the MOOC course, there was a drop both in the percentage of questions answered correctly and the amount of people who viewed the course videos. Additionally, in the fifth run both the percentage of people who watched 95% of the videos increased along with the percentage of questions answered correctly. In the sixth run, both the video viewership and questions answered correctly stay relatively high. However, in the seventh run the questions answered correctly stays high whilst the percentage of people who fully watched the videos plummets. In the third run, we can also observe that the highest percentage of questions are answered correctly whilst the percentage of users that watched 95% of the videos remained quite low. These observations would suggest that there is a loose relationship between these two, as we see a similar trend specifically from run 4 to 6. Having said this, rather than causing more questions to be answered correctly, a higher percentage of users that viewed the videos may indicate more interested users in that run who may be more interested in the course and thus excel more in the questions on average.

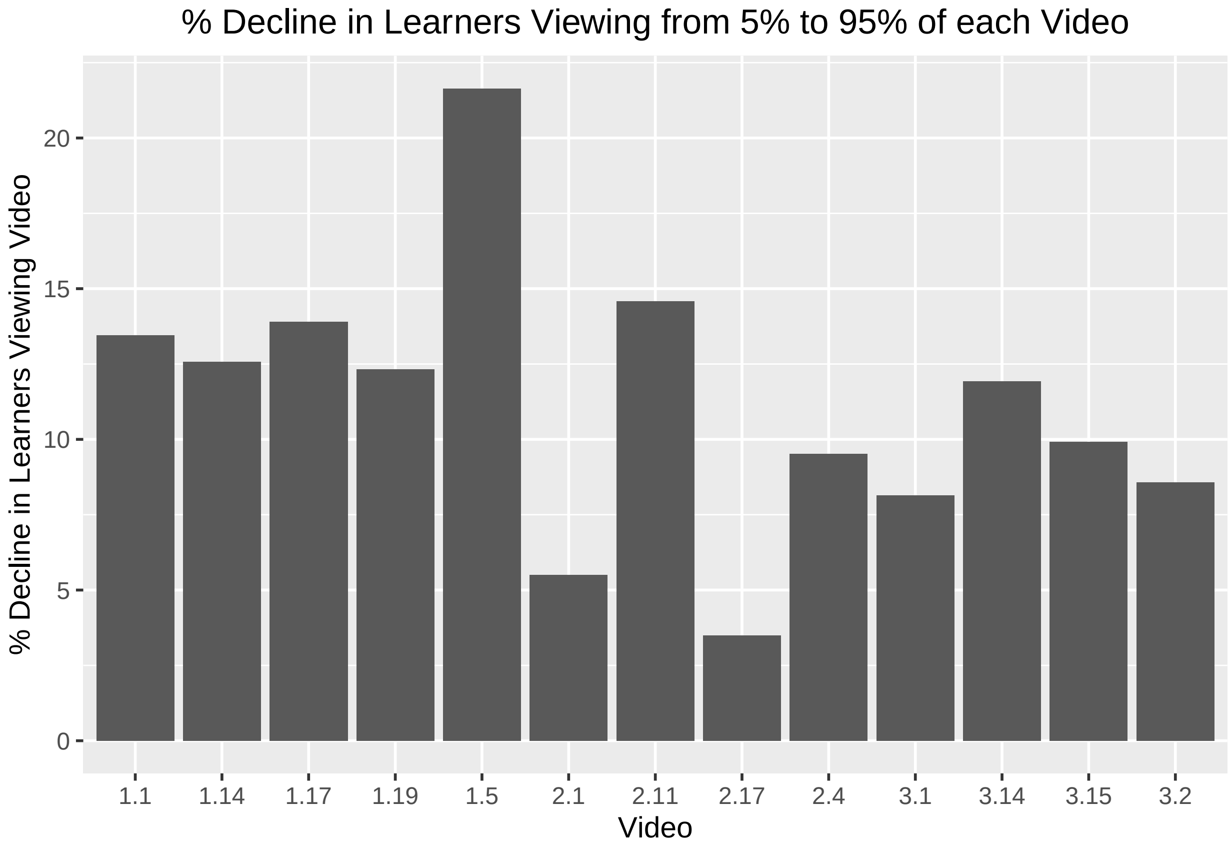
Now that we have demonstrated there is some relationship between the use of the video and outcome of the multiple choice quiz, we should look further into how the videos might be improved. In order to assess this, we will assess the percentage of the videos viewed on average by learners on the course for each video. As previously noted, we are assuming that the videos have remained unaltered over runs due to the identical duration, so these averages will be taken for each video over every run. In addition, recall from our earlier *Data Understanding* section that the "step_position" differentiates each video, and will match one-to-one with the video titles. In this plot we will use it to be more space efficient.



As this plot is cluttered and so difficult to fully interpret, it would be useful to only focus on a selection of the videos that had the highest drop off in audience as the video progressed. We will therefore filter the above plot to show only the top 5 videos with the highest percentage drop in audience to produce the below plot:



This plot indicates that video with step position 1.5 loses the highest percentage of it's audience. This corresponds to the video called "Privacy online and offline". What is particularly notable with this video, is that there is a steep drop in audience from 5% to 25% of the video. Other videos in the above two plots have a fairly consistent gradient of audience decline as the video progresses. This video displays a significantly higher drop in audience that any other video on the course over all runs from 5% to 95% of the video, as shown in the below column chart:

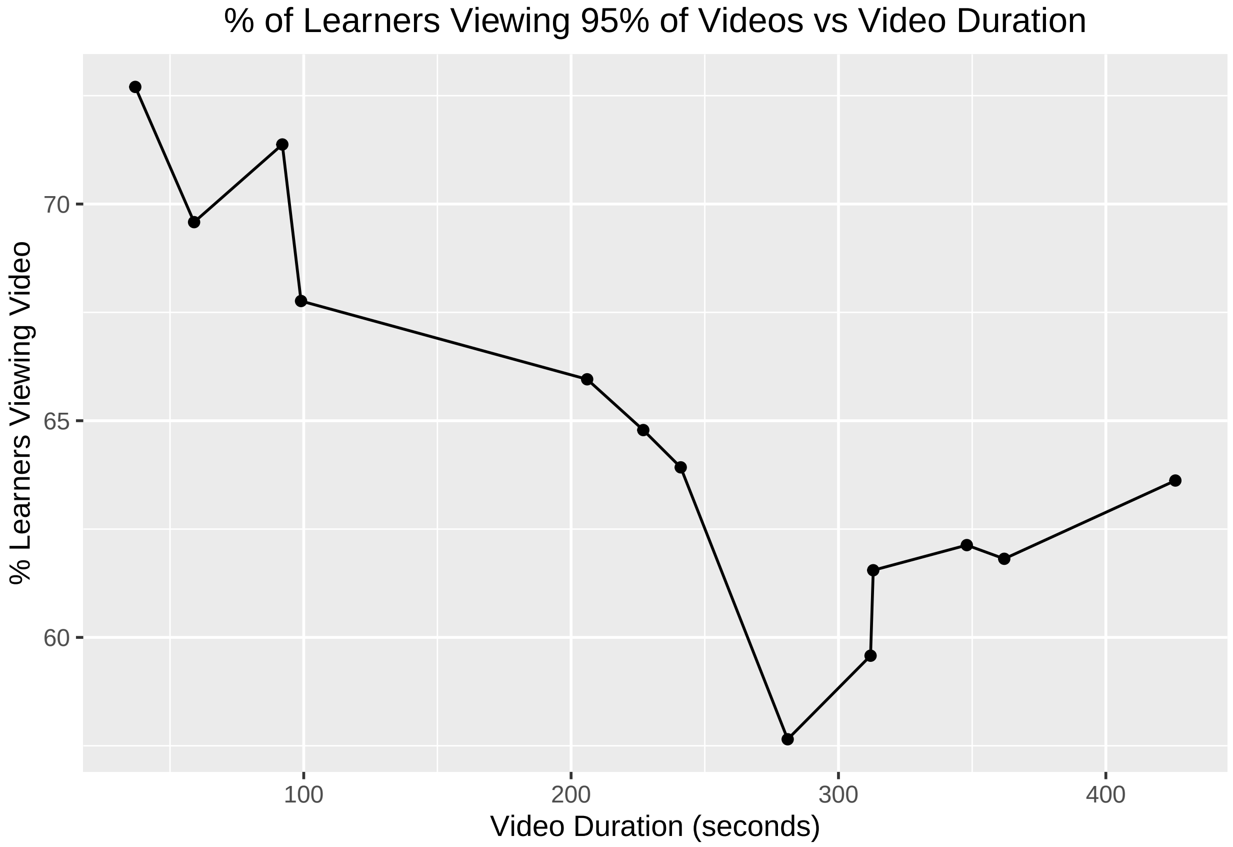


We can also represent this data explicitly and with the inclusion of the video titles, as below:

| Step Position | Video Title | % Drop in Learner Audience |
|---|---|---|
| 1.50 | Privacy online and offline | 21.65 |
| 2.11 | Exploring vulnerabilities in online payments | 14.58 |
| 1.17 | Preserving privacy in cloud storage: privacy by design | 13.90 |
| 1.10 | Welcome to the course | 13.46 |
| 1.14 | Why would anyone want your data? | 12.57 |
| 1.19 | Staying safe online: personal perspectives | 12.32 |
| 3.14 | Exploring security: biometric authentication | 11.93 |
| 3.15 | Exploring security: the Access Control Live Lab | 9.92 |
| 2.40 | The evolving arms race of payment security | 9.51 |
| 3.20 | Devices in the future home | 8.58 |
| 3.10 | Welcome to Week 3: security in the future home | 8.14 |
| 2.10 | Welcome to Week 2: payment security | 5.50 |
| 2.17 | The million dollar contactless payment | 3.49 |

Again, this shows that the video with step position 1.5, “Privacy online and offline”, loses over 20% of the audience from 5% of the video to 95% of it. Comparatively, the video with step position 2.11, “Exploring vulnerabilities in online payments” has slightly less than a 15% loss in viewers. In contrast to these videos, we can observe that the video with step position 2.17, “The million dollar contactless payment”, has the highest viewer retention and loses less than 5% of the audience. The mean average percentage loss is 11.2%.

Whilst many factors could contribute to how engaged a learner is with a video, an aspect from the video-stats data that we could investigate as a driver of this is the duration of the video. To visualise this, we will plot the percentage of people that watched 95% of the videos on average over all course runs against the video duration.



The above plot shows a clear negative correlation between the length of the video and the percentage of learners on the course who watched 95% of it. Interestingly, this correlation is particularly prevalent in the first 300 seconds. After 300 seconds, the percentage of retained learners throughout the video increased again, but not in the vicinity of retention shown in the videos with small duration.

## Evaluation

We initially set out to seek measures that Newcastle University could take in order to improve the efficacy of their online teaching resources, specifically the cyber security MOOC that they run. Thus far in this report, we have assessed the quality and suitability of the data supplied by Newcastle University, prepared data sets for the video stats and question response files and performed several pieces of analysis on them. Our analysis has shown several things. Firstly, we established that there is a slight positive correlation between the percentage of learners who watch the videos and the percentage of questions answered correctly in the course quiz. Our intention was to motivate the subsequent analysis on the video stats data, so that before seeking improvements in the course videos, we should assess the efficacy of the videos as a learning tool. Although our analysis suggests there is a correlation between the two course components, we could attribute this to many things. For instance, it may not be the videos themselves that can impact learning on the course, but poor quiz results like in run 4 could simply reflect learners less engaged with the course, irrespective of whether they watched the video or not.

The second component in our analysis was a comparison of learner engagement between videos. As noted previously, we assumed that the videos remained unchanged in each run of the course, so when comparing videos we can take an average of learning engagement with each video over all course runs. This analysis highlighted several videos that had a particularly low retention of viewers, but of particular note was the video entitled “Privacy online and offline”, which had a far lower audience retention rate than any other video, with a high drop off in audience during the first quarter of the video. It is therefore clear that this video could be improved with the goal of enhancing learner engagement.

The final part of our analysis consisted of an investigation into whether the duration of the videos could be contributing factor in audience retention. Our results indicated that shorter videos do tend to increase learner engagement, with far less learners leaving videos early that were less than 100 seconds long. However, as with our first analysis this interpretation is far from conclusive and there are many other potential reasons that learners would be less engaged with some videos than others.

Out of the results above from our analysis, the second part as an identification for an area of improvement in the course is the most significant. There is clearly scope to promote the efficacy of the video “Privacy online and offline” as a tool for learning, either by condensing the material into a shorter video or by another means. Then, the result of the video duration vs audience retention gave the second most compelling result. Although the result was not conclusive, it showed a clear correlation between the length of the videos and the amount of learners that watched them. Where the second analysis identified the area of improvement, the third provided a method that this area could potentially be enhanced, therefore satisfying the business objectives that we initially set. Finally, as the results only slightly agreed for runs 4,5 and 6, the comparison of the questions answered correctly in the quiz and the percentage of learners who watched 95% of the videos yielded the most unsuccessful results. Even if our comparison was more successful, there are so many factors that could influence learners performance in the quiz that they would still be far from conclusive.

These analyses could be improved and built upon in several ways. Our first analysis could be taken to a more granular level, and consider comparing specific sections of relevance between the quiz and the videos, i.e. comparing a video to the section of the quiz that it relates to. Additionally, continually updating this data pipeline as more course runs occur would improve all aspects of analysis, as more data could either disprove or reinforce some of our interpretations with regard to the analysis outcomes.

For the majority of the data preparation, steps have been taken where possible to promote computing efficiency. When empty vectors have been created to store results, their length has been specified so that incremental object growth is avoided. A specific example of this is the method used in preparing the data frames that we have performed analysis on. Rather than using rbind() with an empty data frame, we used a list with a specified length to store the data from each file and then the function bind_rows() to compile them. These methods allow us to avoid increasing object size incrementally in our data preparation, which would not scale well if we were to perform the analysis for significantly more files. Having said that, parallel computing has not been used in any data preparation steps, so if this were to be improved it could be by using parallel computing to enhance the efficiency of the data preparation process.

## Deployment

The analysis performed is fully reproducible, and any new course runs can be added in the same location should the data pipeline be used for analysis on new data. Loading the project will then facilitate any new data files for video-stats or question-response to be filtered through the analysis and update the appropriate plots. Note that the reproducibility of the project is dependent on the both the files and filenames being in the format they are for this analysis. An an example, in run 3 the video-stats and question-response filenames would be “cyber-security-3_video-stats.csv” and “cyber-security-3_question-response.csv”, respectively. Furthermore, by taking care to avoid code that incrementally grows objects, the steps taking to prepare the data are computationally robust should significantly more data files be ran through the data analysis pipeline. Once any new data files have been formatted appropriately and filtered through the corresponding graph, they will also appear in an interactive presentation created using shiny. The plots in the presentation are linked to the plots prepared in this analysis, and so any new data will be reflected there. None of this requires any user input beyond loading the data and the project, and so the process is fully reproducible with minimal effort. The presentation surmises the three key points from the analysis in the comparison of video viewing vs question response, analysis of learner retention per video and the effect of video duration on this.