Learner Analytics for Newcastle University Online Course: Cyber Security

Mark R. Tyrrell

14/11/2018

Introduction

The Learning & Teaching Development Service (LTDS) at Newcastle University (NCU) has been operating the online course "Cyber Security: Safety at Home, Online, in Life" since early 2016. The course is delivered through the Futurelearn.com platform as a massive open online course (MOOC). The course operates on a freemium model. Learners can take the full course for free, but evaluation and certification require a payment. Learners are expected to complete the course in nine hours.

Despite the course having a large free-tier user-base, LTDS operates the course as revenue-generating enterprise. This analysis investigates factors contributing to the business performance of the course and makes recommendations on areas of improvement in alignment with the business objectives.

1. Summary of business understanding: background, objectives, and success criteria

Evidence-driven decision making is a key contributor to success in business. Organisations are increasingly leveraging the data-collection capacities of their information technology (IT) systems to understand and respond to their customers. This phenomenon is readily apparent in the growing field of learning analytics; a toolset particularly to online course providers. In this regards, one of the key motivators for analytics is student retention. It is economically more efficient to expend resources on student retention than on recruitment marketing in the highly competitive education sector[1].

Futurelearn provides basic analytics for course providers to evaluate the impact of their courses. These analytics are limited in scope and do not provide the customisation necessary for LTDS to extract meaningful insights from their users. However, the platform does allow access to raw data. These data form the basis of this analysis.

Subsequent to consultations with LTDS, the primary objectives of the business were identified as follows:

1. Increasing enrollment
2. Improving course completion rate
3. Increasing percentage of paid upgrades

In translating the business objectives into analytical inquiry, the following key areas of analysis were identified:

- Profile student demographics
- Evaluate course completion performance metrics
- Identify demographic predictors of course completion
- Identify factors contributing to paid upgrades
- Model predictors of course completion

2. Summary of data mining process

The data made available for this analysis consists of 53 tabular data files in comma-separated value (.csv) format covering seven different aspects of the course. In keeping with the objectives of the analysis, as well as time constraints, only two of the seven aspects were analysed.

The analysis took inspiration from the Open University analytics tool "OU Analyse"[1] which tracks and predicts learner outcomes based on demographics and virtual learning environment (VLE) data. As such, the analysis focuses on the **enrollments** and **step activity** aspects as these data provide sufficient attributes to address the analytical objectives of the report.

2.1 Enrolements Data

The enrollments data tracks learner enrollment and completion dates, as well as general demographics. These data were provided in seven separate csv files containing identical variables with observations tracking learner data over different course runs. The seven files were joined and cleaned to produce a dataset of 37296 observations on 13 variables. The temporal dimensions of the dataset vary from March of 2016 through November 2018. A description of the variables is as follows:

1. **learner_id:** Unique identifier
2. **enrolled_at:** Starting date of course
3. **unenrolled_at:** Date of resignation from course (whether completed or not)
4. **role:** Internal classifier used to distinguish learners from administrative auditors
5. **fully_participated_at:** Course completion date
6. **purchased_statement_at:** Upgrade purchase date (can be anytime during course or after)
7. **gender:** male/female/non-binary/other
8. **country:** country of origin
9. ***age_range:** "<18",">65","18-25","26-35","36-45","46-55","56-65","Unknown"
10. **highest_education_level:** "apprenticeship", "less_than_secondary", "professional", "secondary", "tertiary", "university_degree", "university_doctorate", "university_masters", "Unknown"
11. **employment_status:** "full_time_student", "looking_for_work", "not_working", "retired", "self_employed", "unemployed", "Unknown", "working_full_time", "working_part_time"
12. **employment_area:** 22 sectors
13. **detected_country:** Country identification based on IP address

2.2 Step Activity Data

The step activity data tracks learner progression by course step number, providing an entry each time a learner completes a step. These data were provided in seven separate csv files containing identical variables with observations tracking learner data over different course runs. The seven files were joined and cleaned to produce a dataset of 423072 observations on 6 variables. The temporal dimensions of the dataset vary from May of 2016 through November 2018. A description of the variables is as follows:

1. **learner__id:** Unique identifier

2. **step:** Completed step number (subdivision of 3 sections)

3. **week__number:** Week number from 1-3 associated with completed step according to course schedule

4. **step__number:** Completed step number (absolute progressive step number)

5. **first__visited__at:** Course start date
6. **last__completed__at:** Course completion date

3. Summary of data mining results

The datesets were subject to various processes of cleaning, transformation, subsetting and aggregation to meet the requirements of the individual analyses. The datasets were also joined by key variables for certain analyses.

3.1 Profile of Student Demographics

On enrollment, learners are requested to enter demographic details. These data are contained in the enrollments dataset. Country data was automatically collected in almost all cases by resolving the IP address used by the student to access the service. Unfortunately, as the entry is voluntary, only 10 percent of users entered data pertinent to gender, age, education and employment status. However, after filtering out null data, the resultant dataset contains 3819 observations, and is therefore a relevant sample from which to extract inferences about the larger user base.

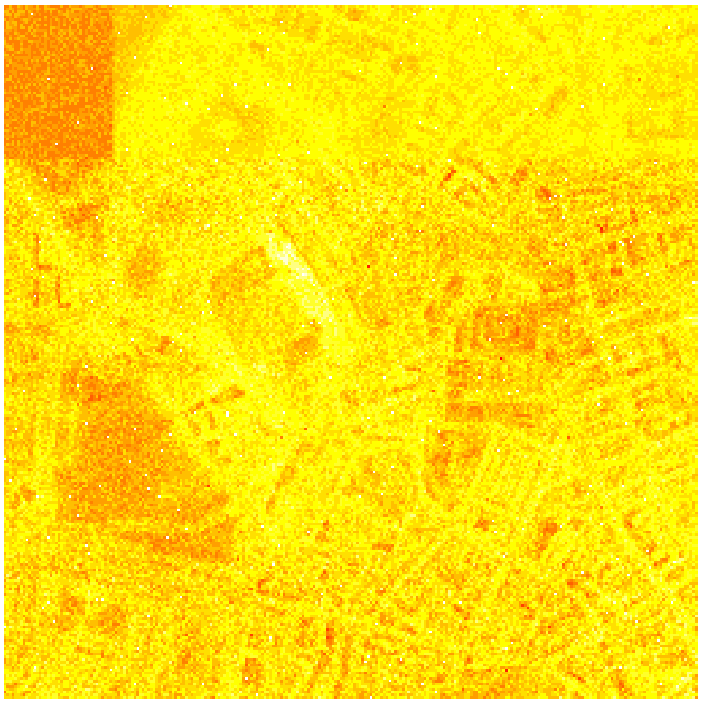
3.1.1 Gender

The current student base is fairly evenly split amongst genders, with slightly more males than females.

3.1.2 Age

The student base is also evenly distributed over the adult age range, from 18 through over 65.

```
## Joining, by = c("taskId", "eventName")
```



3.1.3 Education

The learners are highly educated overall, with almost 66 percent reporting a minimum of a bachelor degree.

3.1.4 Employment Status

A third (34%) of the respondents reported working full time, with another 36 percent reporting not being currently employed (eg. retired, not_working).

3.1.5 Nationality

The table below shows the top 10 student nationalities by number of students. As Future Learn is a British initiative, with the majority of partner Universities based in the United Kingdom, it's not surprising that almost one third (32%) or learners accessed the platform from inside the UK. Fully seven of the countries in the top 10 are English-speaking or have large English-speaking populations. These countries make up 59 percent of the overall student base.

The geographical heat map presented in Figure 1 provides a wider view of the student base by nationality. The countries highlighted on the map represent the top 50 countries by number of students. Although 200 countries in total are represented in the learner nationality dataset, the top 50 countries account for 90 percent of the total.

3.2 Evaluate Course Completion Performance Metrics

Measurement of student performance is critical in any education system, even in a largely cost-free online course. Poor performance can reflect poorly on the student satisfaction, and therefore completion rate. The

enrollment dataset contains numerous interesting data in this regard.

3.2.1 Course Completion Metrics

Of the 37,296 students who enrolled, only 2154 completed the course. This results in a completion rate of 5.8 percent. While low, this rate fits closely with the standard completion rate for MOOCs of 5.5 percent[2]. As the barrier for entrance is small, MOOCs attract a great deal of prospective students.

The enrollment dataset contains a date for enrollment and a separate date for course completion. From this, the number of days taken to complete the course were derived. A histogram of these data is presented in Figure 2.

At registration, the course is estimated to take three weeks to complete at an input level of three hours per week. The data show that this is not the case. According to the enrollments data the median days to completion was 40. The mean was 71, a skew to the right clearly represented in Figure 2. However, it must be noted that the data source for this metric encompasses total time between enrollment and completion, rather than the actual time necessary to complete each section of the course (ref. Section 3.2.2).

It is interesting to note the bi-modal nature of the distribution, with approximately 15 percent of students completing the course around the 125 day mark. This may be the result of auto-scheduled email prompts or other such communication.

3.2.2 Course Step Duration

The step activity dataset provides a record of the 23 course steps completed by each user, including the start and finish dates. From these data, the number of days taken to complete the step were derived. A column chart of these data is presented in Figure 3, showing the average duration for completion of each step. For clarity, the sub-steps between the following sections were removed from the chart:

- step 1.11 through step 1.19
- step 2.11 through step 2.19
- step 2.21 through step 2.29
- step 3.11 through step 3.19

The sum of the average durations for each section displayed in Figure 3 is 25.8 days. However, with the filtered subs-steps added, the total mean is 42.9 days. As with the findings in section 3.2.1, this figure is in excess of the three weeks completion estimate provided at course registration.

As demonstrated by the chart, section one takes longer for users to complete on average, with a notable downward trend for each section as the course progresses. This trend is with the exception of the outlier (step 3.21) which at approximately 8 days, took significantly longer on average for the students to complete. It is unclear from this analysis why this particular step took so long to complete.

3.2.3 Course Dropout Stage

The inclusion of a unique identifier (learner_id) in the step activity dataset allowed for analysis of the maximum step completed by each learner. As with section 3.2.2, the sub-steps were removed from the analysis for clarity. The findings are presented as a histogram in Figure 4. The distribution count across each step closely follows the average step completion duration data outlined in section 3.3.2, including the outlier (step 3.21), but with the exception of the initial step (1.1). Fully 32 percent of learners fail to continue the course after step 1.1.

3.3 Identify Demographic Predictors of Course Completion

As stated in section 3.1, submission of demographic information for learners is voluntary. As such, only approximately 10 percent of observations contained complete demographic data. However, this resulted in a dataset containing 3819 observations. Assuming the population distribution is normal, and the process for obtaining the data was random (i.e. learners who submitted demographic data were not skewed toward a specific subset), the sample size is large enough to meet the conditions for inference. Therefore substantive conclusions can be made about the overall learner population from this sample with reasonable confidence.

In the following sections, the filtered dataset is analysed by subset, in order to form hypotheses about predictors of success based on demographic data.

3.3.1 Performance by Gender

The data demonstrated in Figure 5 a) show a pronounced tendency for male students to achieve higher step progress than their female counterparts. The difference in medians is fully half a section higher, with a less distinct difference in interquartile range (IQR). Though this finding would seem to support common stereotypes of gender roles, the gap in achievement is notable in that it is significantly less than the overall trend. A recent study in the United States[3] found that women were vastly underrepresented in science, technology, engineering and mathematics (STEM) fields, at only 29 percent. These findings are therefore encouraging.

3.3.2 Performance by Age Range

Perhaps surprisingly for an IT-focused course, the data show a marked progression in step completion with increasing age ranges. Referring to Figure 5 b), the ranges from age 56 upwards demonstrate a clear proclivity for older people to achieve higher completion rates with the course. The median for both ranges is greater than section three; a marked contrast to the medians for the first three age ranges, which fail to complete section one.

3.3.3 Performance by Education

The determinants of appear to be somewhat correlated with step completion outcomes for the course. Referring to Figure 5 c), there is notably upwards trend from less-than secondary level, through post-graduate qualifications. This analysis assumes that the "professional" education level indicates some type of professional-level certification. Though the apprentice level appears to counter this trend, the sample size is very small ($n = 13$) and can therefore not necessarily meaningful.

3.3.4 Performance by Employment Status

The major inference apparent in Figure 5 d) is that retired individuals appear to be significantly more likely to get farther in the course, achieving a median of completion of step 3.2. This finding correlates almost exactly with Figure 5 c) (ref. 3.3.3), in that the median step completion for people of retirement age (> 65) was approximately step 3.2.

3.3.5 Performance by Country

There are no stand-out conclusions to be drawn by Figure 6. No single country appears to perform significantly better than others. However it is notable that of the top 20 countries by step completion, 11 are English speaking or have large English-speaking populations.

3.4 Identify Factors Contributing to Paid Upgrades

The working dataset contains 37,296 observations, however only 289 of these individuals paid for the course certification. Unfortunately the subset of these observations for which full demographic data were available is insufficient for statistical inference (n = 13). Therefore, no meaningful conclusions can be made from analysis of these data with regard to identifying predictors of purchase conversions.

The data do appear to show that conversely, certification purchases drive course completion. As demonstrated by Table 6, 81 percent of learners who paid for certification had finished the course. Of this group (n = 233), 64 percent had purchased the course prior to completion (ref. Table 7). This finding indicates that per-completion purchases are a major motivator towards course completion, in that students who have invested financially in the course are more likely to see it through to completion. Of the subset of students who completed the course, then paid after completion (n = 84), 61 percent paid the day of completion, while 75 percent had paid within one week. This finding possibly reflects automated email reminders to graduates, and should be compared against platform standard practices to measure efficacy.

3.5 Model Predictors of Course Completion

Classification analysis of the the demographic predictors of course completion success was accomplished using a logistic regression model. The results of this analysis identified minimal contribution to model predictive functionality from higher education and employment status. The major predictor was age group, with gender also making a notable contribution. These findings are mostly in keeping with the findings identified in section 3.3. With the exception of 'retired' employment status and a marginal trend upwards for step performance compared to educational achievement, the major correlations identified were indeed gender and age group.

A summary of the reduced model is provided below, after removing education and employment status. The model test error was computed using k=10 folds cross validation. The resulting calculated error is 13 percent. However, as demonstrated by the confusion matrix, the model is rather pessimistic. For all 382 test observations, the model predicted probabilities for zero course completions. This result is unlikely of course, as 13 percent of observations in the sample had completed the course. Further evaluation using additional data could confirm the viability of the model.

```
# # Display summary of model parameters
# summary(lr_fit_final)
# # Model Confusion Matrix
# conf_matrix
# # Model Test Error
# lr_test_error
# # Cross Validated Test Error (k = 10)
# cross_val_test_error
```

4. Summary of Results Evaluation and Business Conclusions

The analysis returned numerous results relevant to the business objectives. These results are summarised and evaluated below. Summary conclusions and reccomendations are included where applicable.

4.1 Increase Enrolment

Gender Gap
The difference in course enrollment between genders demonstrates that there is room for growth in female student usage. It may be possible to boost female enrollment by differential marketing strategies or tailoring the VLE in a gender-focused way.

Overly Educated Student Base
There is room for growth in lesser educated demographics

Largely UK Focused
There is opportunity for student base expansion to English-speaking developing countries. Review of platform multi-language options could provide insights into how non-english audiences could be better reached.

4.2 Improve Course Completion Rate

Completion Rate
At 5.8 percent, the completion rate is standard performance for the sector, but lessons learned by other MOOCs could be applied here to increase completion rates. For instance, is the platform providing the latest in terms of user interface and pedagocical approaches?

Course Duration
The 71 day median (43 day mean) for course completion contrasts with the course estimate of 3 weeks. It is possible students sign up expecting an easier or quicker course. The stated completion estimate should be reviewed and either changed, or course structure change to more accurately reflect the real course duration.

Day 90 Course Completion Upswing
The bimodal nature of the dropout by step findings raises questions about why there is such a notable increase in course completions from the 3-month mark. Is it in fact auto-scheduled email reminders? This should be reviewed as it would prove efficacy of this functionality and provide an evidence base for other such automated reminders.

Step 3.21 (Duration Uutlier)
Taking on average 8 days to complete, this step was a major outlier. The particulars of this step should be reviewed and the step possibly reformatted, as it correlates with a high dropout for the same step number.

Downward Trend in Dropout Follows Step Durations
Though this could be simply a platform-based learning curve, it could signify a less than optimum structuring. For instance, easier steps earlier in the course might result in greater step achievement and course completion on average.

Gender Performance Trend
Review gender-focused course accessibility for females. Why are women doing worse than men? It may be possible to tailor the VLE in a gender-focused way based on provided demographic student data.

Age Performance Trend
Review course age-focused attributes. For instance, could a different interface for younger learners improve performance?

Retirees Performance Trend
Retirees show above average completion rates. Marketing to this demographic could improve overall course completion rates.

Nationality Performance Trend
Review platform multi-language options as well as current VLE user interface with regards to english as a second language (ESL) users.

4.3 Increase Percentage of Paid Upgrades

Course Purchase Correlation with Completion
Increased data from future cohorts could further elucidate the dynamics behind these metrics. A review of causation could be possible with an online-based (operational) randomised control trial (RCT). This could involve offering free/discounted evaluation/certification to random learners and comparing results with a control group.

Conclusions for future data mining

The large quantity of missing demographic data (approximately 90%) limited meaningful analysis of subsets of the sample. For instance, though there were 289 observations in the dataset of paid learners, only 13 of these obervations had complete demographic variables. Therefore it was impossible to draw statistical inference about demographic factors contributing to paid upgrades. If there was someway of enticing users to enter demographic data more frequently, it could vastly increase the utitlity of the learning analytics for this course.

References

1. http://www.policyconnect.org. uk/hec/sites/site_hec/files/ report/419/fieldreportdownload/ frombrickstoclicks- hecreportforweb.pdf
2. Chuang, Isaac and Ho, Andrew, HarvardX and MITx: Four Years of Open Online Courses – Fall 2012-Summer 2016 (December 23, 2016). Available at SSRN: https://ssrn.com/abstract=2889436 or http://dx.doi.org/10.2139/ssrn.2889436
3. National Science Foundation, National Center for Science and Engineering Statistics, Scientists and Engineers Statistical Data System (SESTAT) and National Survey of College Graduates (NSCG) (1993, 2013), http://sestat.nsf.gov.

Appendix: R-Code

The code below is provided for reference and reproducibility purposes. The sections detail the individual data files used for the analysis, and the order reflects the loading seqence.

```
################################### Munge: 01-A.R ######################################

##### Read csv data files, combine by type (rbind), clean/transform as required and cache as .RData

##### Enrolments
# Load and combine all datasets
list_sel = list.files(path = "data/", pattern = "*enrolments*")
x <- lapply(list_sel, function(i) read.csv(file = paste("data/",i,sep = "")))
df_enrolments <- do.call('rbind', x)

# Clean & Transform
df_enrolments = df_enrolments %>%
        #Convert date columns to POSIX UTC
        mutate(enrolled_at = as.character(enrolled_at)) %>%
        mutate(unenrolled_at = as.character(unenrolled_at)) %>%
        mutate(fully_participated_at = as.character(fully_participated_at)) %>%
        mutate(purchased_statement_at = as.character(purchased_statement_at)) %>%
        mutate(enrolled_at = ymd_hms(enrolled_at, tz="UTC")) %>%
        mutate(unenrolled_at = ymd_hms(unenrolled_at, tz="UTC")) %>%
        mutate(fully_participated_at = ymd_hms(fully_participated_at, tz="UTC")) %>%
        mutate(purchased_statement_at = ymd_hms(purchased_statement_at, tz="UTC")) %>%
        #Convert country to character vector
        mutate(detected_country = as.character(detected_country)) %>%
        # Add derived variables to display Days Enrolled, Days to Completion, and binary completion var
        mutate(days_enrolled = as.numeric(round(difftime(unenrolled_at, enrolled_at, unit="days"),0))) %
        mutate(days_to_completion = as.numeric(round(difftime(fully_participated_at, enrolled_at, unit=
```

```
        mutate(completed = ifelse(is.na(fully_participated_at), 0, 1)) %>%
        mutate(learner_id = as.character(learner_id))

cache("df_enrolments")

##### Step Activity
# Load and combine all datasets
list_sel = list.files(path = "data/", pattern = "*activity*")
x <- lapply(list_sel, function(i) read.csv(file = paste("data/",i,sep = "")))
df_step_activity <- do.call('rbind', x)

# Clean & Transform
df_step_activity = df_step_activity %>%
        #Convert date columns to POSIX UTC
        mutate(first_visited_at = ymd_hms(first_visited_at, tz="UTC")) %>%
        mutate(last_completed_at = ymd_hms(last_completed_at, tz="UTC")) %>%
        mutate(learner_id = as.character(learner_id))

cache("df_step_activity")


##### Functions
#Tabulate function (shows percentage)
tblFun <- function(x){
        tbl <- table(x)
        res <- cbind(tbl,round(prop.table(tbl)*100,2))
        colnames(res) <- c('Count','Percentage')
        res
}

cache("tblFun")

################################ EDA EDA_Enrolment.R #####################################

# DF Step Activity

# Create working object from cached dateset
df_enrol = df_enrolments

cache("df_enrol")


### 3.1 Profile existing student base

#Student data summary statistics (filtering out unknowns and insufficient samples)
df_enrol_dt = filter(df_enrol, gender != "Unknown" & gender != "nonbinary" & gender != "other"
                     & age_range != "Unknown" & detected_country != "--" & highest_education_level != "

#Reorder factor levels and remove unknown
df_enrol_dt$age_range = factor(df_enrol_dt$age_range, c("<18","18-25","26-35","36-45","46-55","56-65","
df_enrol_dt$gender = factor(df_enrol_dt$gender, c("male", "female"))
df_enrol_dt$highest_education_level = factor(df_enrol_dt$highest_education_level, c("apprenticeship","l
df_enrol_dt$employment_status = factor(df_enrol_dt$employment_status, c("full_time_student","looking_fo
```

```r
#Rename factor levels
df_enrol_dt$highest_education_level = mapvalues(df_enrol_dt$highest_education_level,
        from = c("apprenticeship","less_than_secondary","professional","secondary","tertiary","universi
        to = c("apprentice","< secondary","professional","secondary","tertiary","bachelors","masters","
df_enrol_dt$employment_status = mapvalues(df_enrol_dt$employment_status,
        from = c("full_time_student","looking_for_work","not_working","retired","self_employed","unempl
        to = c("student","seeking work","not working","retired","self-employed","unemployed","full-time


cache("df_enrol_dt")


#Gender histogram
ggplot(df_enrol_dt) + stat_count(aes(gender), fill=I("#0066CC"), col=I("white"), alpha=I(0.8)) +
        labs(x = "Gender", y = "Count", title = "Gender")

tblFun(df_enrol_dt$gender)

#Age group histogram
ggplot(df_enrol_dt) + stat_count(aes(age_range), fill=I("#0066CC"), col=I("white"), alpha=I(0.8)) +
        labs(x = "Age Groups", y = "Count", title = "Age Groups")

tblFun(df_enrol_dt$age_range)

#Education histogram
ggplot(df_enrol_dt) + stat_count(aes(highest_education_level), fill=I("#0066CC"), col=I("white"), alpha
        labs(x = "Education", y = "Count", title = "Education")

tblFun(df_enrol_dt$highest_education_level)

#Employment histogram
ggplot(df_enrol_dt) + stat_count(aes(employment_status), fill=I("#0066CC"), col=I("white"), alpha=I(0.8
        labs(x = "Employment Status", y = "Count", title = "Employment Status")

tblFun(df_enrol_dt$employment_status)

#Country
#Transform data for detected country, aggregate by count for each country, select top countries by x am
df_enrol_ct = df_enrol %>%
        dplyr::select(detected_country) %>%
        filter(detected_country != "--") %>%
        mutate(count = 1) %>%
        group_by(detected_country) %>%
        summarise(sum(count)) %>%
        rename(Country = detected_country, Count = "sum(count)") %>%
        mutate(Percentage = round(Count/sum(Count)*100,2)) %>%
        filter(Count > 95) %>%
        arrange(desc(Count))

cache("df_enrol_ct")

#Convert to names for table
df_enrol_ct_name = df_enrol_ct
```

```
df_enrol_ct_name$Country = countrycode(df_enrol_ct_name$Country, "iso2c", "country.name")
cache("df_enrol_ct_name")

#Plot and table
ggplot(df_enrol_ct_name, aes(Country, Count)) + geom_col(fill=I("#0066CC"), col=I("white"), alpha=I(0.8)
        labs(x = "Country", y = "Count", title = "Nationality")

df_enrol_ct_name[1:10,]

#Create map object
map_country = invisible(joinCountryData2Map(df_enrol_ct, joinCode = "ISO2", nameJoinColumn = "Country",
cache("map_country")
#creating a user defined colour palette
# op = palette(c('green','yellow','orange','red'))
# #find quartile breaks
# cutVector = quantile(df_enrol_ct$Count, prob = seq(0, 1, length = 11), type = 5)
# #classify the data to a factor
# df_enrol_ct$Count = cut(df_enrol_ct$Count, cutVector, include.lowest = TRUE)
# #rename the categories
# levels(df_enrol_ct$Count) = c('low', 'med', 'high', 'vhigh', 't1', 't2','t3','t4','t5','t6')
# #mapping by shade
# par(mai=c(0,0,0.2,0),xaxs="i",yaxs="i")
mapCountryData(map_country, nameColumnToPlot = "Count", catMethod='logFixedWidth', mapTitle='Learners by

#mapping by bubble
# par(mai=c(0,0,0.2,0),xaxs="i",yaxs="i")
# mapBubbles(dF=map_country, nameZSize="Count", nameZColour="Count", numCats = 10, catMethod='categoric

### 3.2 Performance Metrics

##### Completion Data (n = 2154)
df_enrol_days_cp = filter(df_enrol, !is.na(days_to_completion))

cache("df_enrol_days_cp")

#Plot Course Completion Duration frequency
ggplot(df_enrol_days_cp, aes(days_to_completion)) + geom_histogram(binwidth = 10, fill=I("#0066CC"), col
        labs(x = "Days", y = "Count", title = "Histogram: Successful Course Completion Duration (days)"
        scale_x_continuous(breaks = round(seq(min(df_enrol_days_cp$days_to_completion), max(df_enrol_day
ggsave(file.path('graphs', 'course_completion_duration.pdf'))


################################## EDA EDA_Step_Activity.R ###################################

#DF Step Activity

# Create working object from cached dateset
df_step = df_step_activity

# 1. Analysis of Completion Duration by Step
#Compute number of average days take to complete each step
df_step_avg = df_step %>%
        #Add derived variable for duration per step
```

```
        mutate(total_days = as.numeric(round(difftime(last_completed_at, first_visited_at, unit="days")
        #Strip all variables except step and total_days
        dplyr::select(step, total_days) %>%
        #Aggregate
        group_by(step) %>%
        summarise(total_days = round(mean(total_days, na.rm = TRUE),1)) %>%
        #Remove sub-levels
        filter(step < 1.11 | step > 1.19) %>%
        filter(step < 2.11 | step > 2.19) %>%
        filter(step < 2.21 | step > 2.29) %>%
        filter(step < 3.11 | step > 3.19)

cache("df_step_avg")

#Plot average days per step
ggplot(df_step_avg, aes(step, total_days)) + geom_col(size = 1, fill=I("#0066CC"), colour = "#0066CC", a
        title = "Average Completion Duration per Step (days)")  +
        scale_x_continuous(breaks = round(seq(min(df_step_avg$step), max(df_step_avg$step), by = 0.1),1
ggsave(file.path('graphs', 'avg_comp_duration_step.pdf'))


# 2. Analysis of Dropout Frequency by Step
df_step_max = df_step %>%
        #Compute maximum level completed for each unique learner ID
        filter(is.na(last_completed_at)) %>%
        dplyr::select(learner_id, step) %>%
        group_by(learner_id) %>%
        summarise(step = max(step))

cache("df_step_max")

df_step_drop = df_step_max %>%
        # Add Count variable (= 1 for each row) and drop learner_id
        mutate(Count = ifelse(!is.na(learner_id), 1, 0)) %>%
        mutate(learner_id = NULL) %>%
        # Remove sub-levels
        filter(step < 1.11 | step > 1.19) %>%
        filter(step < 2.11 | step > 2.19) %>%
        filter(step < 2.21 | step > 2.29) %>%
        filter(step < 3.11 | step > 3.19)

cache("df_step_drop")

#Plot frequency of drop out by step number
ggplot(df_step_drop, aes(step)) + geom_histogram(binwidth = 0.1, fill=I("#0066CC"), col=I("white"), alp
        labs(x = "Step", y = "Count", title = "Histogram: Last Step Completed") +
        scale_x_continuous(breaks = round(seq(min(df_step_drop$step), max(df_step_drop$step), by = 0.1)
        theme(text = element_text(size=10))
ggsave(file.path('graphs', 'dropout_by_step.pdf'))

#Percentage dropout by step 1.1
dim(filter(df_step_drop, step == 1.1))[1]/dim(df_step_drop)[1]
```

```
################################## EDA EDA_Join.R ########################################


### Join enrolments data (age, gender etc) with step activity on learner_id and look at performance base
# Left join on enrolements data (as opposed to step activity data) because demographic (interesting) da
#Dataset containing full demographic variables (n = 3819)
df_step_perf = left_join(df_enrol_dt, df_step_max, copy = FALSE)
df_step_perf = df_step_perf %>%
        #Drop unused columns
        dplyr::select(-(enrolled_at:fully_participated_at)) %>%
        mutate(country = NULL) %>%
        #Convert all "completed" steps to 3.9 (otherwise shows NA in some cases)
        mutate(step = ifelse(is.na(step) & completed == 1, 3.9, step)) %>%
        #Create new binary var to show course purchase
        mutate(purchased = ifelse(!is.na(purchased_statement_at),1,0))

cache("df_step_perf")

####Predictor Gender
p1 = ggplot(data = df_step_perf, mapping = aes(x = gender, y = step)) + geom_boxplot(fill=I("#0066CC"),
        labs(x = "Gender", y = "Step", title = "a) Gender") +
        scale_y_continuous(breaks = round(seq(min(df_step_perf$step, na.rm = TRUE), max(df_step_perf$st
dat1 = ggplot_build(p1)$data[[1]]
p1 = p1 + geom_segment(data=dat1, aes(x=xmin, xend=xmax, y=middle, yend=middle), colour="white", size=0
cache("p1")

####Predictor Age
p2 = ggplot(data = df_step_perf, mapping = aes(x = age_range, y = step)) + geom_boxplot(fill=I("#0066CC
        labs(x = "Age Range", y = "Step", title = "b) Age Range") +
        scale_y_continuous(breaks = round(seq(min(df_step_perf$step, na.rm = TRUE), max(df_step_perf$st
dat2 = ggplot_build(p2)$data[[1]]
p2 = p2 + geom_segment(data=dat2, aes(x=xmin, xend=xmax, y=middle, yend=middle), colour="white", size=0
cache("p2")

####Predictor Education
p3 = ggplot(data = df_step_perf, mapping = aes(x = highest_education_level, y = step)) + geom_boxplot(f
        labs(x = "Education Level", y = "Step", title = "c) Education") +
        scale_y_continuous(breaks = round(seq(min(df_step_perf$step, na.rm = TRUE), max(df_step_perf$st
        theme(axis.text.x=element_text(angle=90,hjust=1,vjust=0.5))
dat3 = ggplot_build(p3)$data[[1]]
p3 = p3 + geom_segment(data=dat3, aes(x=xmin, xend=xmax, y=middle, yend=middle), colour="white", size=0
cache("p3")

####Predictor Employment Status
p4 = ggplot(data = df_step_perf, mapping = aes(x = employment_status, y = step)) + geom_boxplot(fill=I(
        labs(x = "Employment Status", y = "Step", title = "d) Employment Status") +
        scale_y_continuous(breaks = round(seq(min(df_step_perf$step, na.rm = TRUE), max(df_step_perf$st
        theme(axis.text.x=element_text(angle=90,hjust=1,vjust=0.5))
dat4 = ggplot_build(p4)$data[[1]]
p4 = p4 + geom_segment(data=dat4, aes(x=xmin, xend=xmax, y=middle, yend=middle), colour="white", size=0
cache("p4")

pred_grid = grid.arrange(p1, p2, p3, p4, ncol=2)
```

```
##Country boxplot
#Derive and sort top 10 countries from df_enrol_ct dataset
ordered <- order(df_enrol_ct$Count, decreasing = TRUE)
top_countries = df_enrol_ct[ordered,][1:20,]
#Filter dataset for top 10 countries
df_step_perf_ct = df_step_perf %>%
        filter(detected_country %in% top_countries$Country) %>%
        mutate(step = ifelse(is.na(step), 0, step)) %>%
        group_by(detected_country) %>%
        summarise(step = mean(step))
        # arrange(step)
df_step_perf_ct$detected_country = countrycode(df_step_perf_ct$detected_country, "iso2c", "country.name
cache("df_step_perf_ct")
####Predictor Country
ggplot(data = df_step_perf_ct, mapping = aes(x = reorder(detected_country, step), y = step)) + geom_col
        theme(axis.text.x=element_text(angle=90,hjust=1,vjust=0.5)) +
        labs(x = "Country", y = "Step", title = "Step Performance by Country (Top 20)")


################################## LM Var Tests  ##########################################

#Ages
levs = levels(df_predictors$age_range)
p = c()
for(i in 1:length(levs)){
        x = dim(filter(df_predictors, age_range == levs[i] & completed == 1))[1]
        y = dim(filter(df_predictors, age_range == levs[i]))[1]
        z = x/y
        p[i] = log(z/ (1 - z))
}
plot(p)

#Education
levs = levels(df_predictors$highest_education_level)
p = c()
for(i in 1:length(levs)){
        x = dim(filter(df_predictors, highest_education_level == levs[i] & completed == 1))[1]
        y = dim(filter(df_predictors, highest_education_level == levs[i]))[1]
        z = x/y
        p[i] = log(z/ (1 - z))
}
plot(p)

################################ Linear Modelling  ########################################

### Predictive Modelling

df_predictors = df_step_perf

#Transform predictor dataset
df_predictors = df_predictors %>%
        #Remove extra vars
```

```
        dplyr::select(-(learner_id:purchased_statement_at)) %>%
        dplyr::select(-(employment_area:days_to_completion)) %>%
        dplyr::select(-(step:purchased))


#Convert nominal categorical var to binary integer
df_predictors$gender = (as.integer(df_predictors$gender)-1)
#Convert categorical vars to integers as per results of ordinalt tests (ref. Ordinal_test.R))
df_predictors$age_range = as.integer(df_predictors$age_range)
df_predictors$highest_education_level = as.integer(df_predictors$highest_education_level)



###### PRELIM MODEL
#Deconstruct, standardise and rebuild predictors dataset (& remove employment status because of convers
df_predictors_std = scale(as.matrix(df_predictors[,1:3]))
df_predictors_lm = data.frame(df_predictors_std, df_predictors[,5])
names(df_predictors_lm) = c("gender","age_range","highest_education_level","completed")

#Fit the Linear model using the dataset
lr_fit = glm(completed ~ ., data = df_predictors_lm, family = "binomial")
summary(lr_fit)
#Compute prediction probabilities
phat = predict(lr_fit, df_predictors_lm, type = "response")
#Compute fitted (i.e. predicted) values
yhat = ifelse(phat > 0.5, 1, 0)
#Calculate confusion matrix
table(Observed=df_predictors_lm$completed, Predicted = yhat)
#Compute training error
1 - mean(df_predictors_lm$completed == yhat)




###### FINAL MODEL

#Final model Age + Gender
df_predictors_lm_red = df_predictors_lm[,-3]
names(df_predictors_lm_red) = c("gender","age_range","completed")

set.seed(20)
#Create training and testing sets
len_set = dim(df_predictors_lm_red)[1]
train_qt = round(len_set*0.9,0)
random_sel = sample(len_set, len_set, replace = FALSE)
#Create test/training indices
train_sel = random_sel[1:train_qt]
test_sel = random_sel[(train_qt+1):len_set]
#Slice predictors df into test/training dfs
df_train = filter(df_predictors_lm_red, row_number() %in% train_sel)
df_test = filter(df_predictors_lm_red, row_number() %in% test_sel)

#Fit the Linear model using the dataset
```

```
lr_fit_final = glm(completed ~ ., data = df_train, family = "binomial")

cache("lr_fit_final")

summary(lr_fit_final)
#Compute prediction probabilities
phat_test = predict(lr_fit_final, df_test, type = "response")
#Compute fitted (i.e. predicted) values
yhat_test = ifelse(phat_test > 0.5, 1, 0)
#Calculate LR confusion matrix
conf_matrix = table(Observed=df_test$completed, Predicted = yhat_test)
#Compute LR test error
lr_test_error = 1 - mean(df_test$completed == yhat_test)

cache("conf_matrix")
cache("lr_test_error")



###### CROSS VALIDATION

n=nrow(df_predictors_lm_red)

set.seed(20)
#10-fold cross validation
nfolds = 10
#Sample fold-assingment index
fold_index = sample(nfolds, n, replace=TRUE)
fold_sizes = numeric(nfolds)
#Compute fold sizes
for(k in 1:nfolds){
        fold_sizes[k] = length(which(fold_index==k))
        fold_sizes
}

#Assign vector for avg MSE
cv_lsq_errors = numeric(nfolds)

#Loop through folds fitting model to k-1 training data and predicting values for k, assign errors to ve
for(k in 1:nfolds){
        #Fit model by least squares using all but the k-th fold
        lsq_tmp_fit = glm(completed ~ ., data=df_predictors_lm_red[fold_index!=k,], family = "binomial"
        #Compute fitted values for the k-th fold
        phat = predict(lsq_tmp_fit, df_predictors_lm_red[fold_index == k,], type = "response")
        #Work out the MSE for the k-th fold
        yhat = ifelse(phat > 0.5, 1, 0)
        cv_lsq_errors[k] = mean((df_predictors_lm_red[fold_index==k,]$completed - yhat)^2)
}

#Take mean of error vector
cross_val_test_error = weighted.mean(cv_lsq_errors, w=fold_sizes)

cache("cross_val_test_error")
```