Investigating user engagement with 'Cyber Security: Safety At Home, Online, and in Life'

Luke Kaye

2022-11-16

Introduction

Newcastle University ran a three week online course titled "Cyber Security: Safety At Home, Online, and In Life". It was freely available to the public, and allowed interested learners to develop their understanding around cyber security and its many aspects.

With any programme of study, it is invaluable for those running the programme to understand what drives learner engagement, and therefore satisfaction with the course. Knowing what keeps learners engaged allows continued useful development of the course for future users, and can assist with the production of other programmes of study.

In this analysis, we will attempt to explore engagement with the course by its learners, and factors relating to the course and its learners that may explain behaviour in this pattern of engagement.

Analysis Aims

By exploring engagement with the course and its associated factors, we hope to produce conclusions that will allow a developer of the course, or indeed a similar course, to make decisions that will improve the experience for future learners engaging with the programme, by increasing user engagement.

For example: do certain demographics benefit more than others from the structure of the course? As the course was taught in English, we might wander, among other things, if people from certain continents benefit from learning in English more than those from others. If this turned out to be true, a developer of the programme may wish to offer the course in languages native to areas that struggle to engage with the English-based content.

Due to the open-ended nature of this investigation, there are likely to be many insights revealed from its results so we cannot define a strict set of questions that will be answered.

Available Data

Available are Comma Separated Values (`.csv`) files containing data derived from seven runnings of the course.

The eight `.csv` files available for each period provided data of different interest, namely:

- Archetype Survey Responses
- Enrolment Information
- Course Dropout Survey Responses
- Question Responses

- Course Module Activity
- Programme Lead Members
- Video Statistics
- Weekly Sentiment Survey Responses

As we will see, certain data files are highly useful, whereas others provide negligible information.

The data associated with both learners and course leads are anonymised; there are no personally identifying data that would breach the privacy of people involved with the course.

Additionally, a cover page for each running of the course was provided in `.pdf` format, to assist with interpretation of the data within the `.csv` files. These cover pages detailed the modular structure of the course each period.

Analysis Outline

Our primary analysis of the data will be exploratory in nature, hoping to indicate any significant relationship between *course module activity and demographic information* associated with learners. For example, do learners of certain age brackets engage with more course modules than those of other age brackets?

Our secondary analysis will largely be based on the conclusions reached from the primary analysis. This might entail deeper analysis into any interesting relationships revealed in the data, or analysis from a different perspective if no interesting conclusions are reached.

The analysis will be conducted primarily by using R, a programming language used for statistical computing and graphics. Additional techniques and libraries within R will be employed, which will be introduced as they are required.

Primary Analysis: EDA of Demographic Variables and Module Activity

To structure the analysis detailed within this report, the R library `ProjectTemplate` is used as it provides a system for automating menial parts of data analysis, such as organising files and loading/processing data. `ProjectTemplate` creates several directories used to organise this analysis, one such directory being `data` which we will use to store the raw data files available to us.

Exploration of Data Files

As described previously, each run of the course has data files split into different categories. The two categories of interest for this primary analysis are:

- Enrolment Information (`cyber-security-x_enrolments.csv`)
- Course Module Activity (`cyber-security-x_step-activity.csv`)

where `x` within the file name corresponds to the numbered run of the course.

Course Module Activity Data File

We will begin by exploring the content of `cyber-security-x_step-activity.csv`. This data file contains columns describing if and when each learner has accessed a module within the course. A brief description of each is given below.

`learner_id`

An alphanumeric id associated with each learner.

`step`

The module number associated with a course material.

`week_number`

The week that the course material is contained within, i.e. the part of the module number preceding the decimal point.

`step_number`

The ordered number for the course material in a given week, i.e. the part of the module number succeeding the decimal point.

`first_visited_at`

The date and time in UTC that the learner first accessed the course material.

`last_completed_at`

The date and time in UTC that the learner completed the course material.

From here, we can see the date and time that a learner accessed and completed a course module. An issue with this data file is that any `step` value that should end with a zero is incorrectly stored as not having that final zero. For example, `2.10` is incorrectly stored as `2.1`. `week_number` and `step_number` don't suffer from this issue, therefore as these two columns together represent the same data as `step`, we won't make use of `step`. Furthermore, we can assume that by initially accessing a course module, a user has essentially engaged with that content as they may have forgotten to fulfil the technical criteria for 'completing' that module, and as such we won't make use of `last_completed_at`. Note that if a learner does not have a row in this data file associated with a joint `week_number` and `step_number`, it means that they did not access that course material; we can use this property as our engagement metric. Therefore, since the date and time of accessing a course material isn't important to us, just whether or not a learner did, we can additionally discard `first_visited_at`.

It's worth noting that the course module corresponding to `step = 3.21` was only ran in the first two years of the course, whereas every other course module was ran every year. Therefore, in favour of a balanced analysis, we will remove any rows containing `step = 3.21`

Therefore, within this data file, there are three columns we will make use of. Specifically, `learner_id`, `week_number` and `step_number`. These three columns provide sufficient information to tell us if a learner has engaged with a course material.

Enrolment Information Data File

Next, we will explore the content of `cyber-security-x_enrolments.csv`. This data file contains columns providing enrolment information for learners. A brief description of each is given below.

`learner_id`

An alphanumeric id associated with each learner.

`enrolled_at`

The date and time in UTC that the learner enrolled in the course.

`unenrolled_at`

The date and time in UTC that the learner unenrolled in the course, if they left the course early.

`role`

The role that the course user had. For the vast majority of users, and specifically our users of interest, this column takes the value `learner`.

`fully_participated_at`

The date and time in UTC that the learner fully completed all course materials.

`purchased_statement_at`

The date and time in UTC that the learner purchased the optional certificate stating that they have completed the course.

`gender`

The gender of the learner. Specified by the learner.

`country`

The country of origin of the learner. Specified by the learner.

`age_range`

The age bracket of the learner. Specified by the learner.

`highest_education_level`

The level of education of the learner. Specified by the learner.

`employment_status`

The employment status of the learner. Specified by the learner.

`employment_area`

The employment sector of the learner. Specified by the learner.

`detected_country`

The country of origin of the learner. Detected by the course website.

From here, we can associate several demographic variables with each learner, as well as their unique alphanumeric id number. We aren't interested in `enrolled_at`, `unenrolled_at`, `fully_participated_at` (since we will be using information within `cyber-security-x_step-activity.csv` to see course engagement), and `purchased_statement_at`.

For the vast majority of the learners, information has not been provided for any of the demographic variables and as such learners lacking demographic information will be excluded from the analysis. For some learners, some, but not all, of the demographic variables have been provided. We will retain these observations for our analysis as it would not make sense to discard the rest of their demographic information simply because they don't have a value for all of the variables. The demographic information functions as a set of categorical variables, so we can simply treat unprovided values as an 'Unknown' categorical value, which may be of interest later, or we may simply ignore them in the analysis. Even after choosing not to include observations with wholly unprovided demographic information, we still have a sufficiently large sample size for our analysis, since the raw data files are very large. There are also a handful of `country = NA` values, for which we will discard the entire row due to the unpredictable nature of `NA` values. We also note in `employment_status` that there are observations on `not_working`; we won't merge this with another category as `not_working` could mean many things, e.g. unemployed, retired.

Since our attention is focused on observations that have provided demographic information, we won't require the `detected_country` column, as the same information is provided within `country`.

Therefore, within this data file, there are eight columns of interest: `learner_id`, `role`, `gender`, `country`, `age_range`, `highest_education_level`, `employment_status` and `employment_area`. These columns provide a good selection of demographic information relating to learners that we can explore for relationships with course engagements.

Exploration of Other Data Files

The remaining six data files not looked at so far relate to the other aspects of the data:

- Archetype Survey Responses (`cyber-security-x_archetype-survey-responses.csv`)
- Course Dropout Survey Responses (`cyber-security-x_leaving-survey-responses.csv`)
- Question Responses (`cyber-security-x_question-response.csv`)
- Programme Lead Members (`cyber-security-x_team-members.csv`)
- Video Statistics (`cyber-security-x_video-stats.csv`)
- Weekly Sentiment Survey Responses (`cyber-security-x_weekly-sentiment-survey-responses.csv`)

We will briefly talk about these data files here, and why they are not of interest to our primary analysis.

`cyber-security-x_archetype-survey-responses.csv`

This file includes data relating to archetypes corresponding to learners. The data within this file isn't useful for our analysis.

`cyber-security-x_leaving-survey-responses.csv`

This file includes data relating to learners that left the course early. In theory, the opportunity to analyse data on early course leavers sounds highly useful to our analysis, however the sample size of these data files are relatively small. When taking into account that the vast majority of learners did not provide demographic information, this would result in a preprocessed sample size for this file which would be too small to do any meaningful analysis with.

`cyber-security-x_question-response.csv`

This file includes data relating to learner performance in question-based exercises. The focus of the analysis here is to analyse engagement with the course, not ability of the learners, and therefore the data here is not useful to us. You might argue that high attainment in question-based exercises shows engagement with the course, but proving this would be an analysis in itself and would digress from the more straightforward results derived from learners accessing course materials.

`cyber-security-x_team-members.csv`

The file includes data relating to programme leads on the course. None of it is useful as our attention is focused on learners.

`cyber-security-x_video-stats.csv`

The file includes data relating to video material within the course and associated metrics, such as viewer retention, video views, viewing device and proportion of views from each geographic region the video received. There is no column within this file for `learner_id`, since it provides aggregated data for each video, and therefore we can't compare it with our demographic data from `cyber-security-x_enrolments.csv`. Since analysing factors around viewer retention for course videos could be useful for gaining insights around course engagement, this data file could in theory be quite useful, but it is more suited to its own independent analysis.

`cyber-security-x_weekly-sentiment-survey-responses.csv`

This file includes data relating to weekly survey responses about how learners felt about the course. It doesn't include a column for `learner_id`, and therefore none of the data here can be contrasted with any explanatory variables, rendering it useless.

How the Data Files of Interest Work Together

When we begin the analysis proper, we will combine all of the columns across the `cyber-security-x_step-activity.csv` data files, producing a 'master' data frame for step activity. This process will be repeated for `cyber-security-x_enrolments.csv`.

The data will then be further preprocessed, detailed later, after which we will eventually be able to associate the rows within our enrolments and step activity data frames through use of the `learner_id` columns. This will give us a profile of each learner, with their demographic variables and their engagement with course modules which we can analyse to identify any interesting trends and insights.

Data Preprocessing - EDA

Within `ProjectTemplate`, we can define a list of libraries to be loaded by R for use within our analysis, by editing `global.dcf` contained within `config`. One such library is `dplyr` contained within the `tidyverse` set of libraries, which is already specified within `global.dcf` by default. We will be using `dplyr` to preprocess our data by the specification mentioned in the previous section.

We run `load.project()` to load the data files within `data` into R, which are then saved to `cache` in an appropriate format for faster future loading. The data files are independently stored as data frames within R, with names similar to their original `.csv` file names.

01-A.R

Within `munge`, we have a preprocessing script `01-A.R` which does our initial data preprocessing on the step activity and enrolment data.

It begins by row binding the entries in each step activity data frame together. Then it removes any rows containing `module_number = 3.21` for reasons explained previously. then discards columns `step`, `first_visited_at` and `last_completed_at`. Afterwards, it uses the information given in `week_number` and `step_number` to make a new column called `module_number`, storing the modules as ordinal numbers, representing their positions in the set of modules. It then removes the `week_number` and `step_number` columns. Preprocessing is finished on the step activity data by collapsing each individual learner_id and module_number pairwise observation into a list, called `module_number_list`, such that the data frame now only contains one observation for each learner_id, associated with a list of its accessed modules.

Separately, it row binds the entries in each enrolment data frame together, then discarding columns `enrolled_at`, `unenrolled_at`, `fully_participated_at`, `purchased_statement_at` and `detected_country`. Afterwards, it filters out non-learners and discards the `role` column. Next, any rows featuring any `NA` values (namely, the four `NA` values within the `country column`) are discarded. It then further removes the rows that have `Unknown` values for all six demographic variables.

A note about duplicate `learner_id` values:

By running different `dplyr` wrangling commands, we can see some basic properties of the combined data. For example, we can see whether we have any duplicate `learner_id` values across the combined enrolment data. Surprisingly, we do have duplicate `learner_id` values within this data, implying that certain learners have in fact taken the course multiple times over different runs of the course. We therefore further clean our enrolment data by removing any rows that feature a `learner_id` that is also present in another row. It is very important we do this or else we will run into problems with the association between step activity and demographic information; since a learner might provide different demographic information on different runs of the course, how will we know which set of demographic information to associate with their step activity? The cleaned sample size is sufficiently big such that simply removing any rows corresponding to multiple appearance `learner_id`s shouldn't affect the analysis meaningfully.

Returning to preprocessing the data, `01-A.R` continues by removing any rows that share a `learner_id` with another row such that any `learner_id`s that had multiple observations corresponding to it are entirely excluded from the analysis, as justified above.

`01-A.R` finishes the preprocessing by creating a new data frame `demographic_step_activity` which contains the combined observations from the step activity and enrolment data frames, joined by association of `learner_id`s. This data frame does not include `learner_id`s which we have excluded from the analysis.

Some of these rows contain values `module_number_list = NULL`, if a learner accessed no modules. `01-A.R` additionally replaces these `NULL` values with the number `0`, as working with `NULL` values can be problematic and a `0` value makes sense with the ordinal structure of this variable. One final step taken for the data preprocessing is to create a new column `module_number_list_length` which states the count of module numbers accessed by a learner, with special consideration for our `0` values within `module_number_list`.

Finally, `01-A.R` caches all the new data frames generated during preprocessing, including `demographic_step_activity`, for quicker loading in the analysis.

The Preprocessed Data Frame; demographic_step_activity

After `01-A.R` has finished running, we have our preprocessed data frame `demographic_step_activity`. This is a data frame containing the cleaned, combined information given in the step activity and enrolment data files. Having all our information of interest in one place makes the upcoming analysis significantly easier to perform.

The differences of note in this final data file are:

- Learners only have one row corresponding to them, with their demographic information
- Only learners with notable demographic information are included
- Columns not of concern within the original data are omitted
- The modules accessed by each learner are stored in ordinal form as a list
- The count of modules accessed by a learner are stored

The way that learner ids and the demographic information are stored within this data frame is identical to `cyber-security-x_enrolments.csv`, with the same column names. The information detailing which modules have been accessed by each learner is contained in a column `module_number_list`, which has list entries containing the modules the learner accessed, each in ordinal form. If a learner accessed no modules, the entry for this column is `0`. We also have a column storing the count of modules accessed by a learner, within `module_number_list_length`. Due to the way we have stored the information on accessed modules, we can treat it as discrete quantitative data, with special consideration for `0` values. We used `learner_id` for associating data across observations and data files however it doesn't have much use for associating course engagement itself and therefore we won't use it in the analysis proper. The demographic information functions as a set of categorical variables.

Analysis - Exploratory Data Analysis

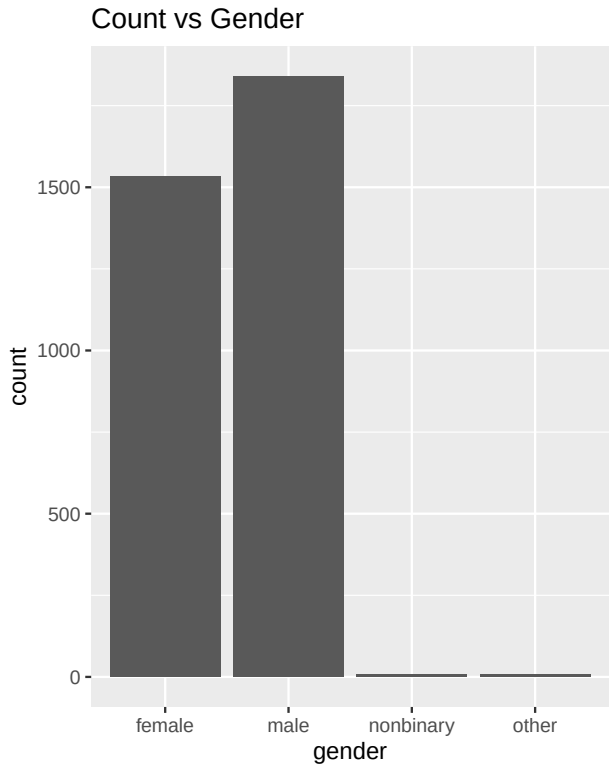
Since we have now finished our data preprocessing, we begin on the analysis proper. This primary analysis is quite open ended in nature as detailed previously, as it attempts to explore general relationships between module engagement and the assorted demographic variables.

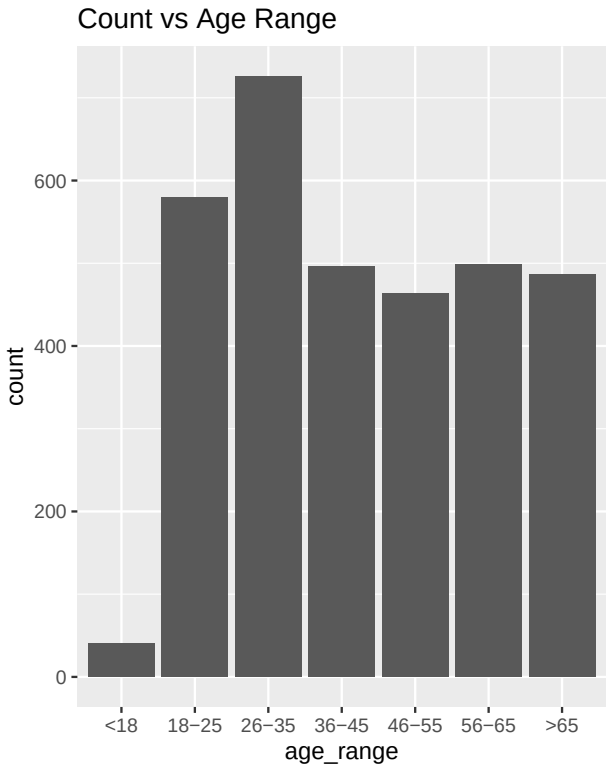
The code behind this analysis can be found in the `src` directory, within the R file `primary_analysis`, however excerpts from the file will be embedded in this report to allow for detailed commenting alongside the individual steps within the analysis.

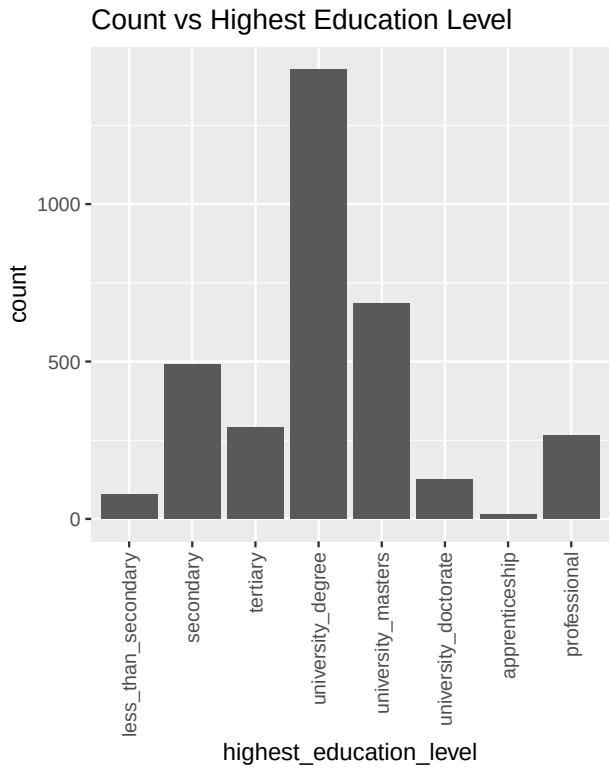
Plots generated for this analysis will be built using `ggplot2`, another library contained within `tidyverse`. As a result, `global.dcf` does not need to be edited to account for `ggplot2` as it is already loaded as part of `tidyverse`, which we loaded previously when making use of `dplyr`. We will also use the `gridExtra` library to allow us to display `ggplot2` plots in grids on this report, which we add to `global.dcf`.

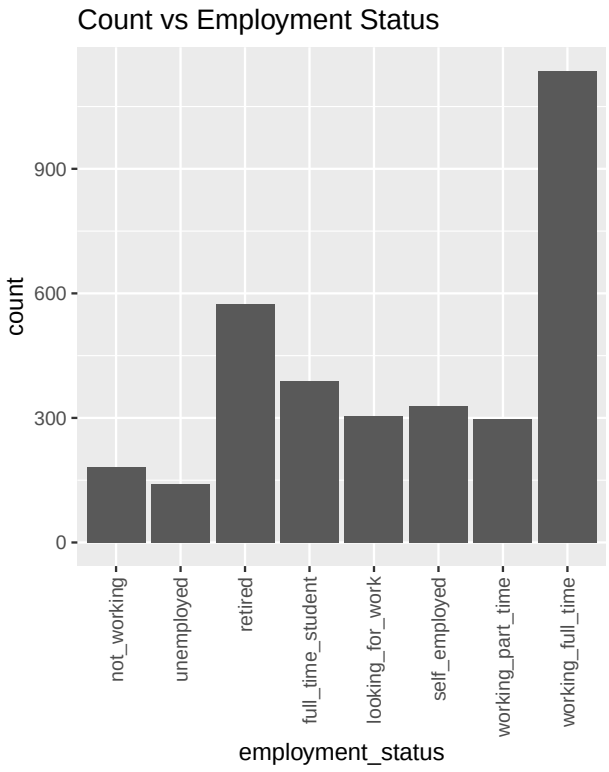
The variables we are trying to explain here are `module_number_list` and `module_number_list_length`, namely various metrics on module engagement such as its count or distribution. We will be modelling our demographic variables as predictors, attempting to explain these metrics of `module_number_list` and `module_number_list_length`.

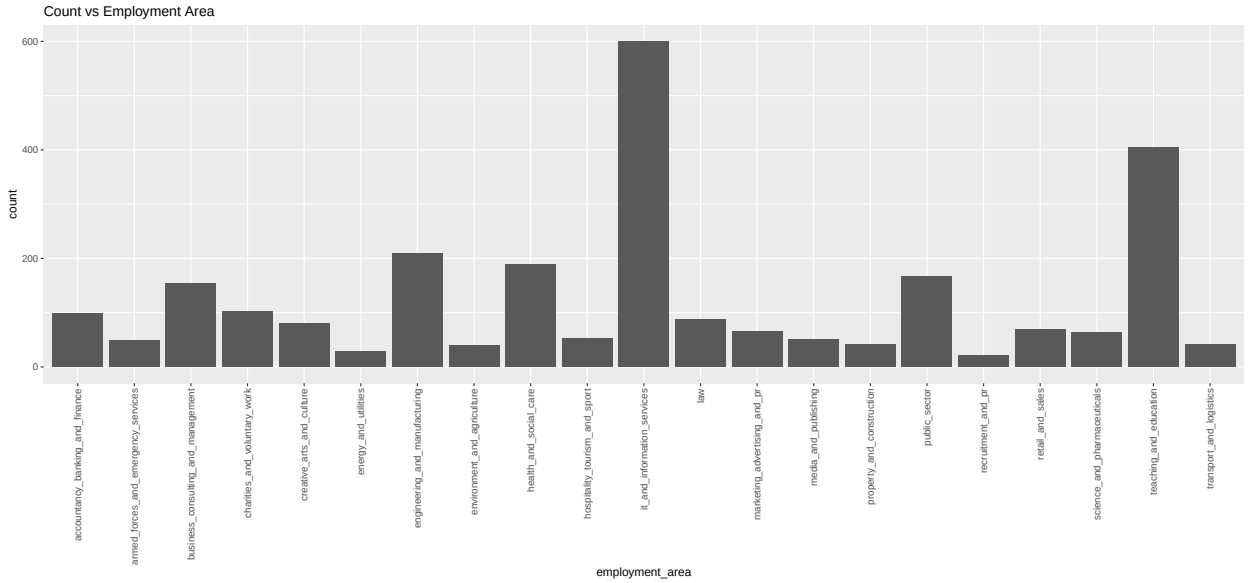
We begin our analysis by first considering a series of bar plots, detailing the counts of each of the demographic predictors. Analysis of these bar plots of the counts will allow us to get a general description of how the demographics are distributed.

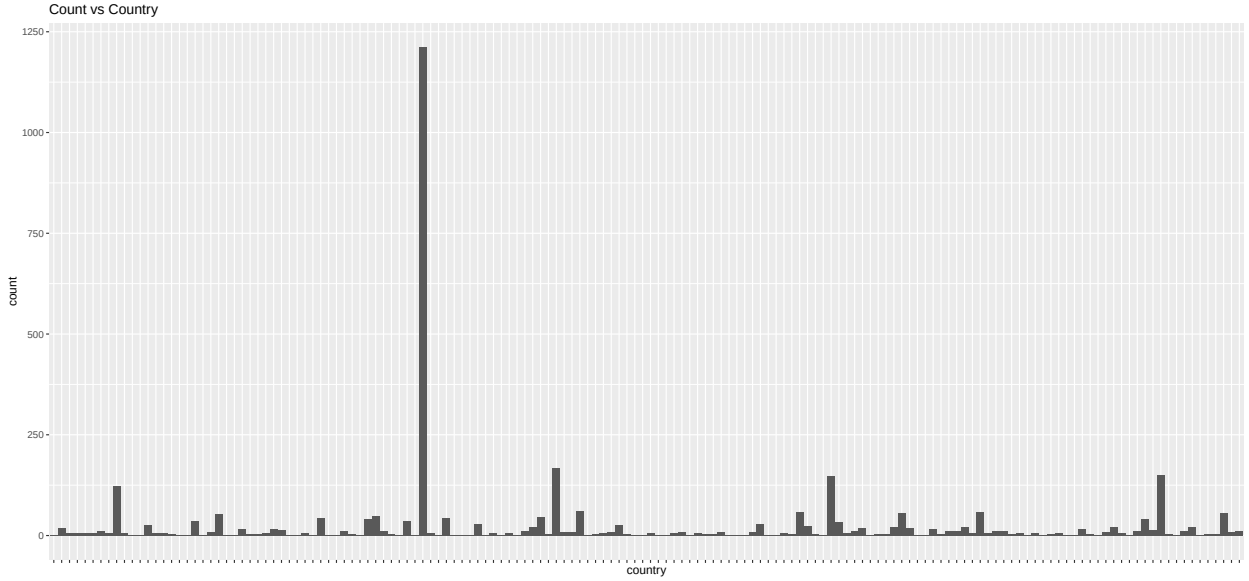












Since there are too many country axis labels to fit on the Count vs Country plot, we instead manually search the data frame to see the countries with noticeably greater enrolment than the rest, specifically the five bars that protrude from the rest.

```
top_5_countries
```

```
## # A tibble: 5 x 2
##   country     n
##   <chr>   <int>
## 1 GB       1211
## 2 IN        168
## 3 US        150
## 4 NG        148
## 5 AU        122
```

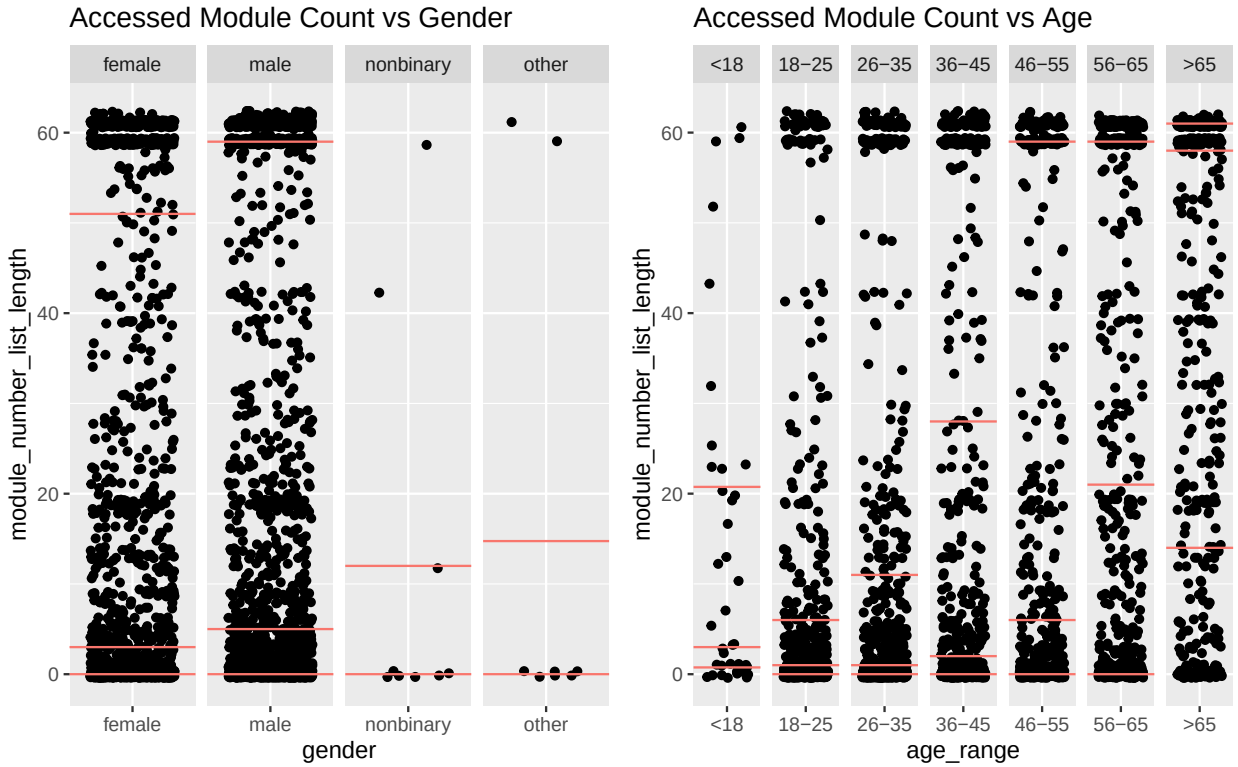
This states that, overwhelmingly, most learners are from Great Britain, with also noticeable, but smaller, counts from India, USA, Nigeria and Australia.

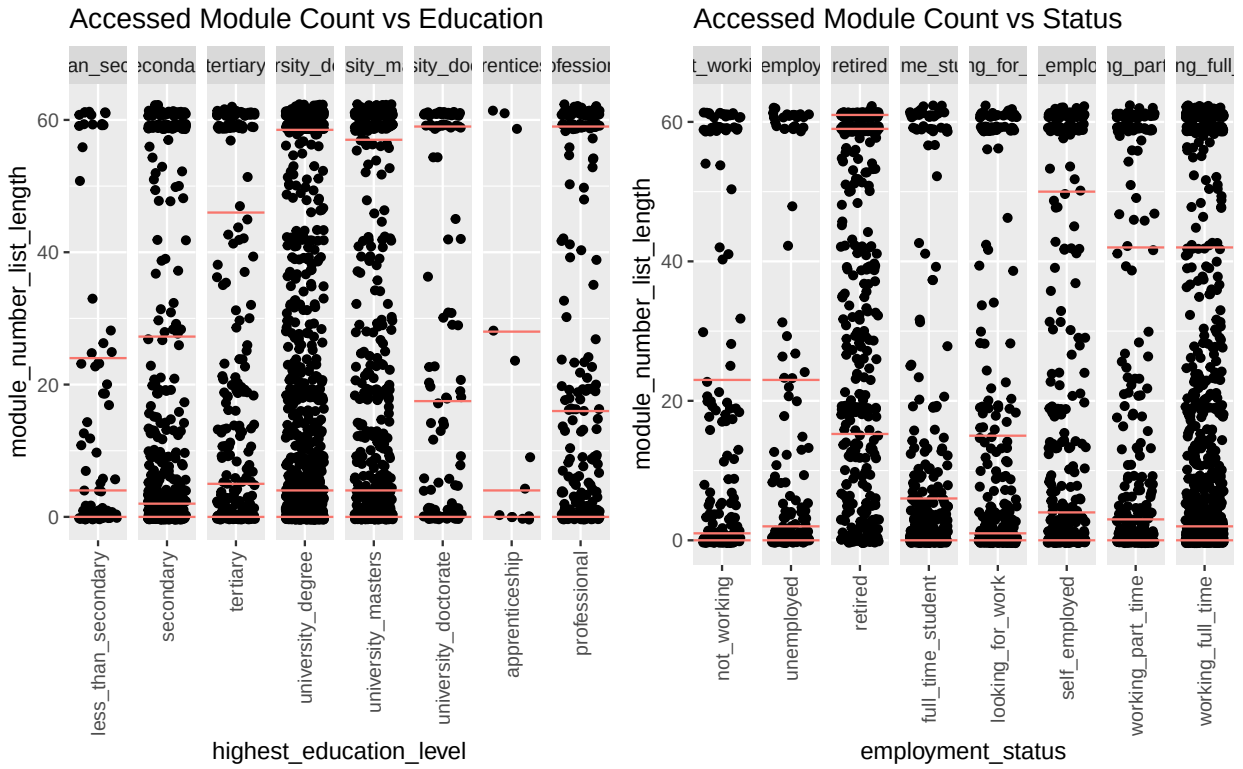
Looking at the rest of the plots, we see a reasonable balance between female and male learners, with occasional learners identifying as non-binary or other. For ages, 26-35 is the largest group, then secondarily 18-25. Beyond that, ages greater than or equal to 36 are fairly evenly distributed, then finally a small amount of learners are less than 18 years old. Most learners are university educated, with a strong proportion having only an undergraduate degree and noticeable amounts of people also possessing a masters degree. There seems to be a spread of educational backgrounds that took an interest in the course, although the number of people taking it with apprenticeships or a less than secondary education is quite small. Most learners work full time, although there is a significant proportion of retirees. Beyond that, we see that people from all employment backgrounds have enrolled in the course. The most prominent area of employment for learners is within IT, as you might expect from a course of this nature, but there are also large amounts of learners from teaching and education which also makes sense. Beyond this, we see a spread of learners from all manners of employment area.

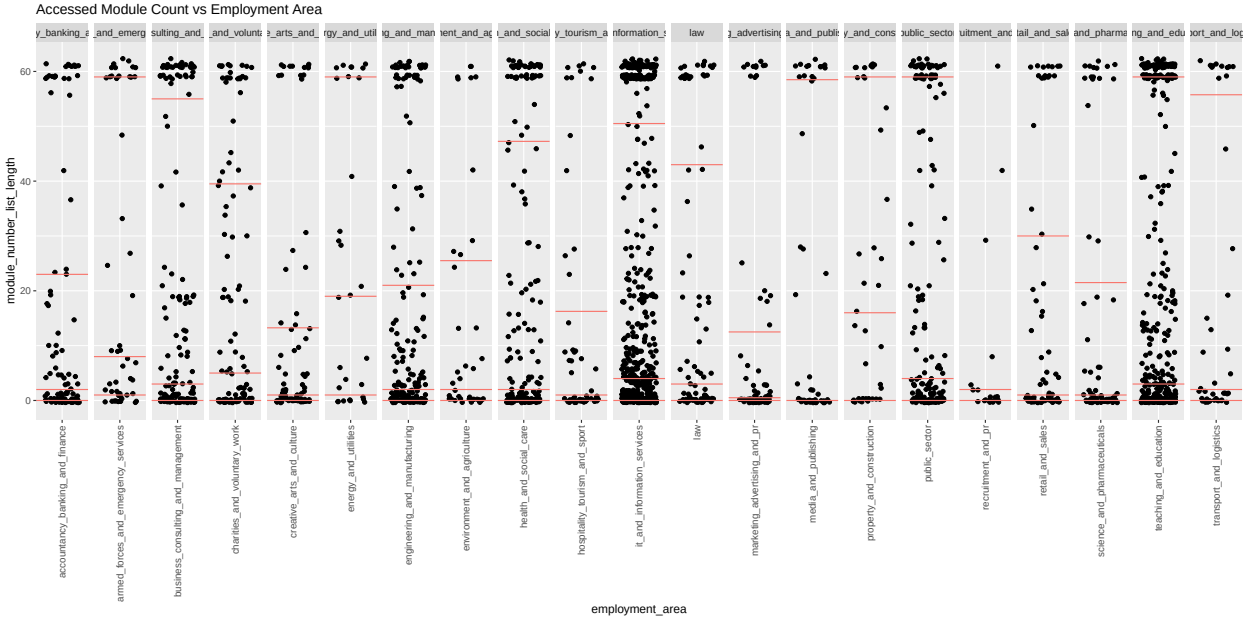
While the analysis so far doesn't provide any insights regarding course engagement, it is useful to see the overall distributions of predictors.

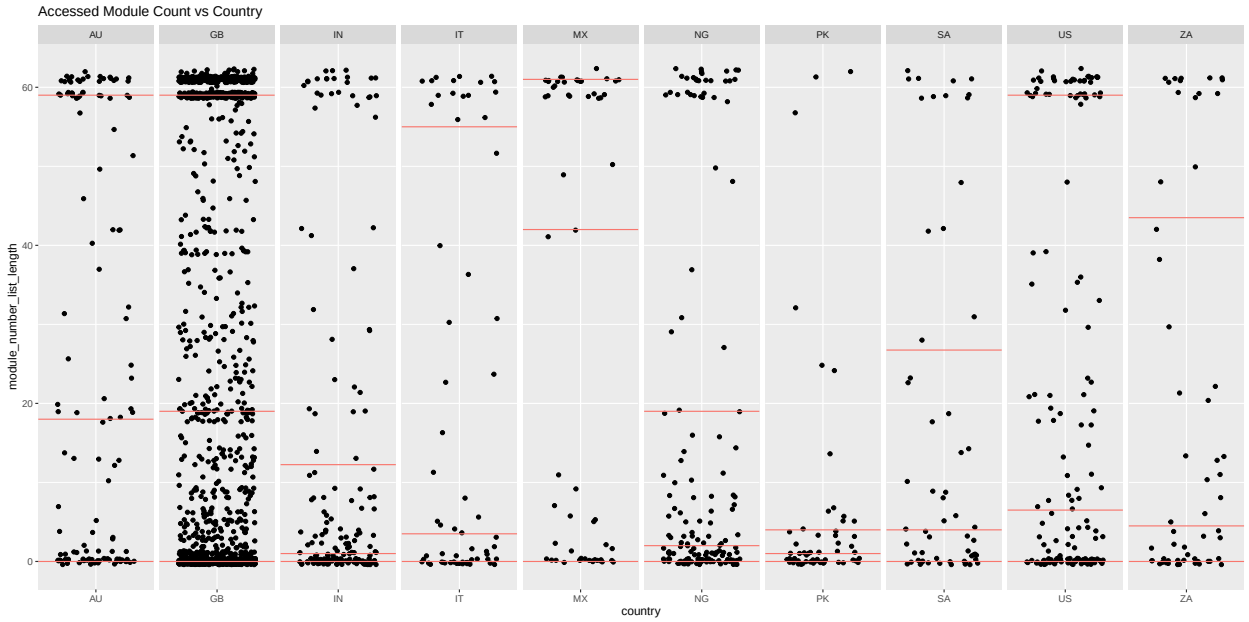
Next, we produce scatterplots of `module_number_list_length`, that is, the number of accessed modules, vs the different predictors. Each plot is generated by giving each `module_number_list_length` value a small amount of random noise, allowing us to see the density of `module_number_list_length` values across the predictors. We are required to do this since `module_number_list_length` is discrete quantitative; without making this transformation we would have several observations at each value overlapping on the plot. We additionally plot a red line at every quartile for each category to see the skewness of the distributions in `module_number_list_length`.

For our scatterplot against country, we will only consider the countries with the ten highest learner counts since there are far too many countries to visualise on one plot, as we saw previously. We note the proportion of learners overall from these ten countries is 0.6108677, which should provide a sufficient sample size to see differentiation in module engagement across different countries.




Accessed Module Count vs Education
module_number_list_length
highest_education_level
less_than_secondary
secondary
tertiary
university_degree
university_masters
university_doctorate
apprenticeship
professional
Accessed Module Count vs Status
employment_status
not_working
unemployed
retired
full_time_student
looking_for_work
self_employed
working_part_time
working_full_time


Accessed Module Count vs Employment Area
module_number_list_length
employment_area
accountancy_banking_and_finance
armed_forces_and_emergency_services
business_consulting_and_management
charities_and_voluntary_work
creative_arts_and_culture
energy_and_utilities
engineering_and_manufacturing
environment_and_agriculture
health_and_social_care
hospitality_tourism_and_sport
it_and_information_services
law
marketing_advertising_and_pr
media_and_publishing
property_and_construction
public_sector
recruitment_and_pr
retail_and_sales
science_and_pharmaceuticals
teaching_and_education
transport_and_logistics



There are some interesting conclusions to be inferred from the plots here, which we discuss. Firstly, we note that the distribution for modules accessed among female and male learners is quite similar. The sample size for the other gender categories is too small to make any definitive conclusions. In terms of ages, the average number of modules accessed seems to increase fairly linearly as the age of the learner increases, as shown by the quartiles. This could be that people have more free time for the course as they get older, or perhaps that older people feel they are unknowledgeable in the course content compared to those that are younger, which would make sense as technology use is more ubiquitous among younger people. For education level, the learners with university or professional qualifications tend to engage with the course more than others. Since the course has a modular structure quite similar to a university degree course, this could be that someone with a tertiary education is used to the format for the course and is therefore more likely to engage with it further. Looking at employment status, we see that retirees stand out as having very high course engagement, with the median value sitting at almost the maximum. The remaining statuses all have much more positively skewed distributions. This could be that retirees have more free time to pursue a course like this one. After retirees, those in work have the next highest course engagement. For employment area, in terms of how it relates to module engagement, there doesn't appear to be much in terms of correlation, with the spreads of module engagement all positively skewed in some way. Finally, looking at our top ten countries in terms of learner counts, we have the highest engagement from learners originating in Australia, Great Britain, Mexico and USA. Mexico is not predominantly an English speaking country, so perhaps the language spoken natively within a country isn't a significant factor in course engagement.

While we could explore more summaries within this dataset, the information given in these scatterplots already serves sufficiently to narrow down the predictors we should be be considering in more detail. There is evidence of correlation among course engagement within `age_range`, `highest_education_level` and `employment_status`. There is also some spread in `country`, although the data within this category is binned too thinly to really explore it in any more depth and from what we've seen regarding `MX`, it's quite likely that whether or not a country is predominantly English speaking is irrelevant.

Conclusion of Analysis - Exploratory Data Analysis

The main purpose of this primary analysis was to identify general relationships between the predictors and module engagement. While we originally intended to look at both `module_number_list` and `module_number_list_length`, it was sufficient to look only at `module_number_list_length`, to identify enough information to be able to choose a subset of the predictors to analyse in more detail.

We identified that the age, education level, and employment status of a learner plays a significant factor in their engagement with the course. The older a learner gets, the better their engagement is. Those with tertiary education or professional qualifications also tend to engage with course modules more. Finally, retirees tend to engage with the course much more thoroughly than others, followed by those who are working.

There will clearly be some correlation among the predictors here; retirees tend to skew older for example. Therefore further analysis will be required to identify which subsets of the categories within these predictors are most important. For example, is it the older age of a learner that drives their engagement with the course, or is it actually the fact that they are retired?

Since we will be performing another analysis which goes into more depth into these predictors, we will hold off making any definitive conclusions for how a course developer could improve the course, as we are likely to generate better insights after this following analysis.

Secondary Analysis - Regression Model

As we found some interesting relationships in our dataset, we will focus this secondary analysis on that same dataset, within a smaller subset of it and in more detail.

Our original exploration was fine to only use general plots, but to truly pick apart what it is within this dataset that differentiates module engagement, we will make use of a regression model. Our chosen three predictors we will be considering further, `age_range`, `highest_education_level` and `employment_status`, are all categorical variables. `module_number_list_length`, our count of accessed modules by a learner, provided some useful insights previously and as such this is the variable we will be attempting to explain.

Data Preprocessing - Regression Model

We will be retaining `demographic_step_activity` from the previous analysis since it has already had most of the preprocessing done on it that we will be requiring here. There are a few additional steps we will need to take here, however. Once again, we will use `dplyr` to preprocess the data.

02-A.R

Within `munge`, we have a second script called `02-A.R`, which will do our additional preprocessing to form our subsetted data for our regression model.

It begins by duplicating `demographic_step_activity`, forming a new data frame called `model_data`. It then deletes our now unneeded columns, namely `gender`, `country` and `employment_area`. While it is likely that we won't be requiring `learner_id` or `module_number_list`, we retain them anyway. Since, in the previous analysis, we retained various rows that contained `Unknown` values in certain predictors, it is almost certain that we will have `Unknown` values in our data now, even after removal of some columns. This is undesirable for our later regression model, so `02-A.R` removes any row that features even one `Unknown` value. This retains approximately 3200 observations, only dropping a comparably small amount.

`02-A.R` then finishes by caching `model_data` for quicker loading during the analysis.

The Preprocessed Data Frame; model_data

What we have in `model_data` is a subset of `demographic_step_activity`. It is identical except:

- It drops columns `gender`, `country` and `employment_area`,

- It drops any rows containing one or more `Unknown` values in a predictor.

While `model_data` is largely similar to `demographic_step_activity`, dropping all instances of `Unknown` values is important since there aren't any useful insights to be generated from patterns on values which are ambiguous. Dropping unnecessary columns assists with readability of the data frame.

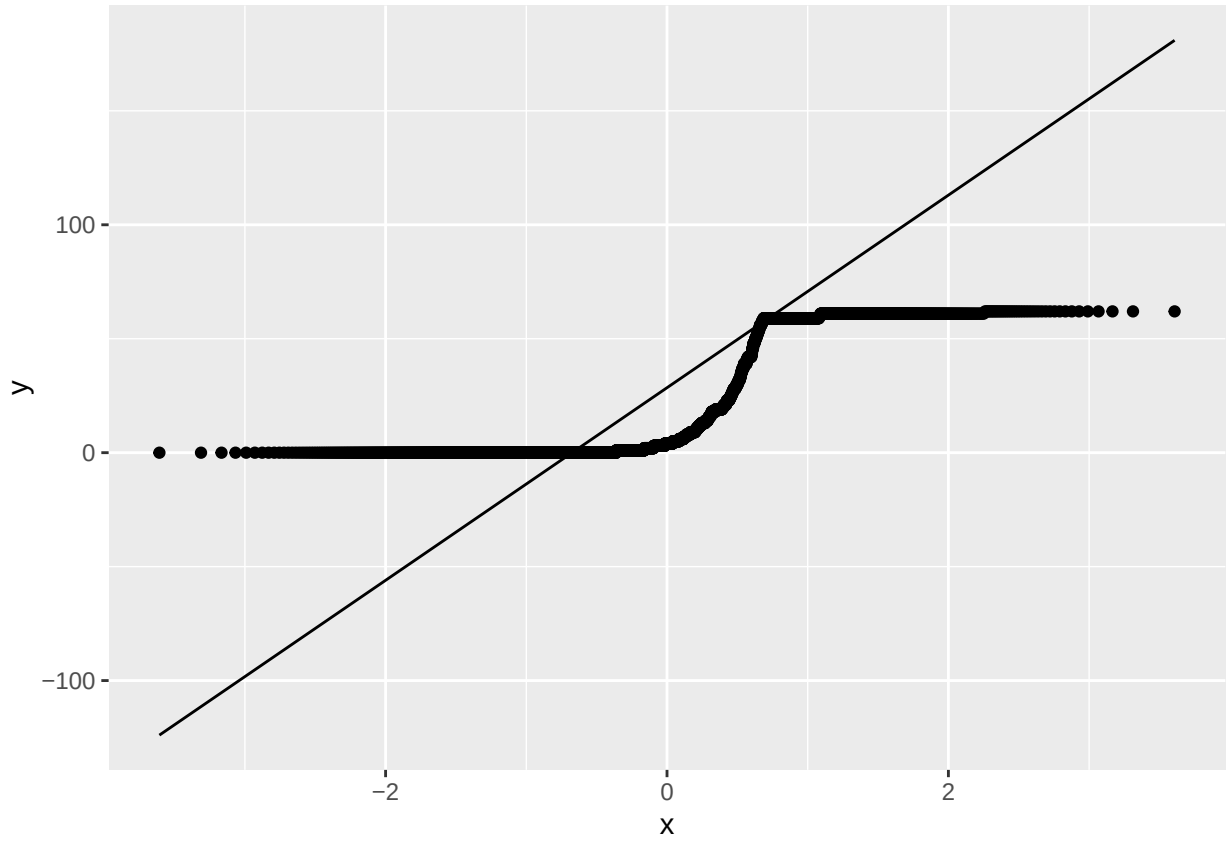
Analysis - Regression Model

This secondary analysis is more specific than the primary analysis. Instead of considering general exploratory techniques, such as with scatterplots, we will be structuring this analysis around a regression model, attempting to place explanatory behaviour for `module_number_list_length` more specifically.

Within the `src` directory, we have an R file `secondary_analysis` which contains the code for this secondary analysis, although certain components of the analysis will be computed directly in this report. Like before, excerpts from the file will be embedded in this report to allow commenting of analysis steps.

If its assumptions are satisfied, a three-way ANOVA model could be useful here, since we have three categorical predictors. One assumption we must check is the normality of the residuals of the data fitted to a linear model. To do this, We fit a full multiplicative linear model and then make use of `ggplot2` again to view a quantile-quantile plot of the residuals, as well as performing a Shapiro-Wilk normality test.

```
## lm(formula = module_number_list_length ~ age_range * highest_education_level *
##     employment_status, data = model_data)
```




```
##
```

```
##  Shapiro-Wilk normality test
##
## data:  residuals(linear_model)
## W = 0.9377, p-value < 2.2e-16
```

We see in both the quantile-quantile plot and the Shapiro-Wilks test a very clear violation of normality, therefore we won't consider an ANOVA model further.

Another alternative that could be suitable is a Poisson regression model. As a generalised linear model, it is generally well suited to count data, which our data is since `module_number_list_length` takes non-negative integer values. It is also safe to assume that each learner functions as an independent observation. One important property we must check, however, is that the mean and variance of a Poisson regression model in `module_number_list_length` are roughly equal, otherwise we would have an overdispersed model which would result in poor identification of structure. To do this, we will use the `dispersiontest` function within the `AER` package, which we add to `config.dcf`.

```
poisson_model$call
```

```
## glm(formula = module_number_list_length ~ age_range + highest_education_level +
##     employment_status, family = poisson, data = model_data)
```

```
dispersiontest(poisson_model)
```

```
##
##  Overdispersion test
##
## data:  poisson_model
## z = 32.885, p-value < 2.2e-16
## alternative hypothesis: true dispersion is greater than 1
## sample estimates:
## dispersion
##   30.23805
```

Clearly the mean and variance differ substantially so we cannot use Poisson regression. If we recall our exploratory data analysis earlier, we note that a large amount of values within `module_number_list_length` take zero values. Therefore, a model that is equipped to deal with this concentration of zero values could be well suited to the underlying process generating the data. Two candidates of this nature are zero-inflated models and hurdle models. Both models are mixture models, assuming a point mass binomial model part at zero and a generalised linear model (GLM) part geared towards count data. They differ in how they assume zeros are generated.

- Both models first make use of the binomial part to generate either an 'off' value, returning a zero, or generate an 'on' value, to then sample from the GLM part geared towards count data.
- Afterwards, a zero-inflated model will generate a value from the GLM part, which can still return a value of zero. However, with the hurdle model, this generated value from the GLM part must be nonzero. To this end, a zero-inflated model uses a usual discrete probability distribution, whereas the hurdle model uses a zero-truncated discrete probability distribution (that is, it takes values greater than zero).

This has some good theoretical backing, since someone who has progressed into the course beyond the first few modules is probably more likely to stick with it than someone who has just started, so in theory

the process generating these zero values for `module_number_list_length` could be different to the process generating the rest of the data.

As it is not immediately obvious which zero-generating behaviour would be better suited to the data, we will fit models of both classes. In the name of model simplicity as we have no strong reason to believe otherwise, for our models, we make the assumption that the probability someone 'escapes' this zero accessed module count is constant, that is, it doesn't depend on any of their demographic variables. We will also assume a Poisson distribution for our GLM part, as it is a simple example of a model used with count data.

To fit these models, we make use of the `countreg` library which we add to `config.dcf`. Note that this library must be installed manually as it is not located in Rs CRAN, therefore if you wish to reproduce this analysis you should run `install.packages("countreg", repos = "http://R-Forge.R-project.org")` before running `library('ProjectTemplate')` and `load.project()`.

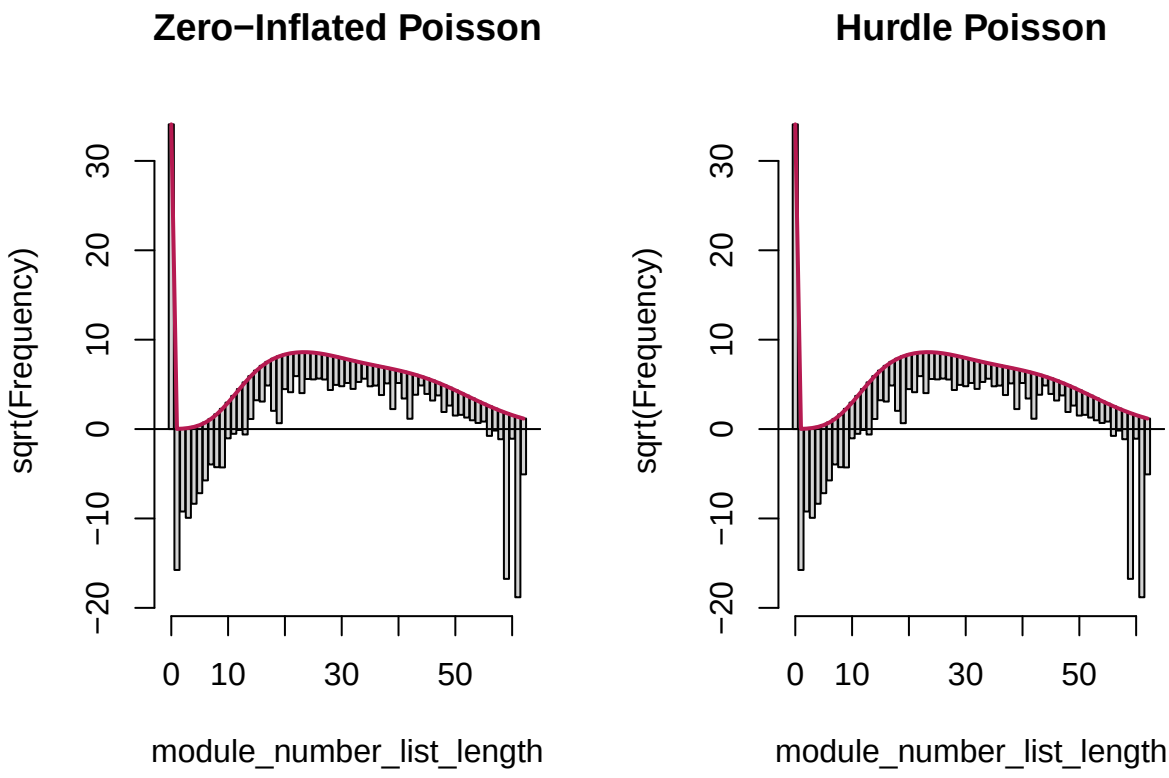
```
zi_poisson_model$call
```

```
## zeroinfl(formula = module_number_list_length ~ age_range + highest_education_level +
##     employment_status | 1, data = model_data, dist = "poisson")
```

```
h_poisson_model$call
```

```
## hurdle(formula = module_number_list_length ~ age_range + highest_education_level +
##     employment_status | 1, data = model_data, dist = "poisson")
```

We will then analyse rootograms of these two models to assess their fits. We won't be using `ggplot2` here since these rootogram functions are bespoke to `countreg`.



We see from the bumps in both rootograms that the models are overdispersed, and therefore indicate poor fit. We will additionally consider zero-inflated and hurdle models that make use of negative binomial distributions, as they generalise the Poisson distribution by incorporating a parameter that can account for this overdispersion by adjusting for the variance independently from the mean.

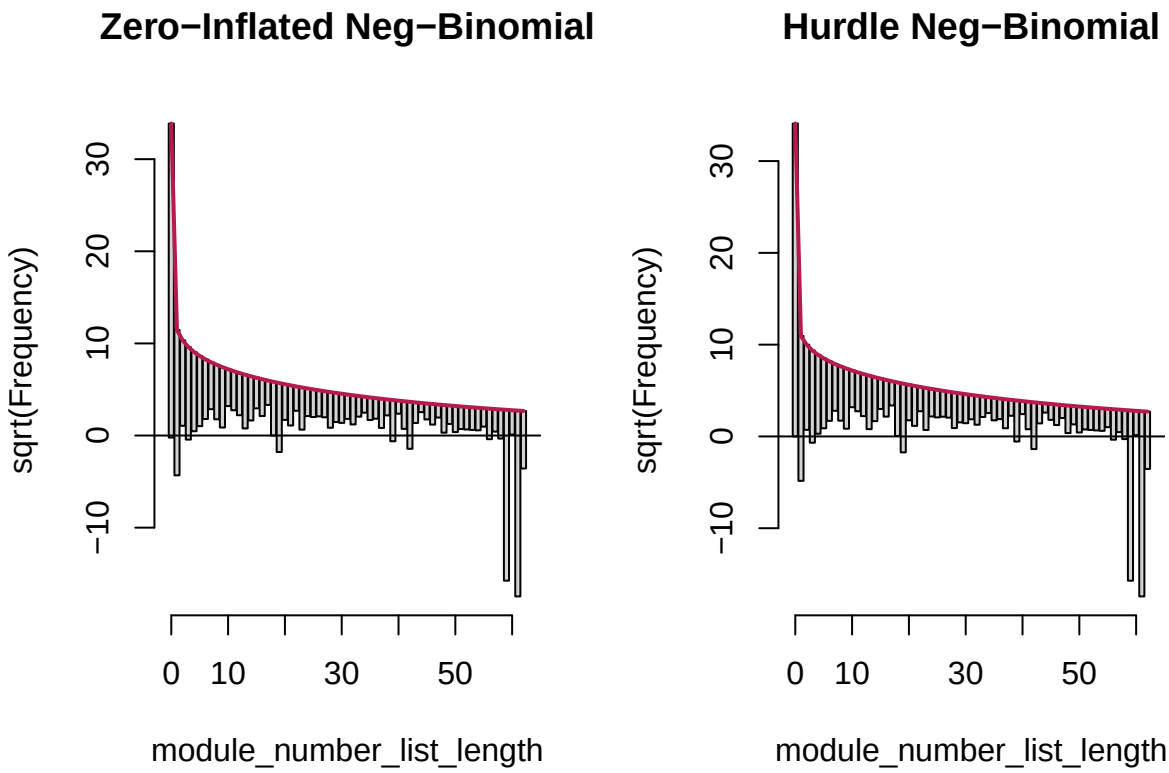
```
zi_negbin_model$call
```

```
## zeroinfl(formula = module_number_list_length ~ age_range + highest_education_level +
##     employment_status | 1, data = model_data, dist = "negbin")
```

```
h_negbin_model$call
```

```
## hurdle(formula = module_number_list_length ~ age_range + highest_education_level +
##     employment_status | 1, data = model_data, dist = "negbin")
```

We then plot their rootograms.



We see from these rootograms that both models seem to fit the data well. We will additionally look at Bayes Information Criterion (BIC) for these four models to see a summary of their fits

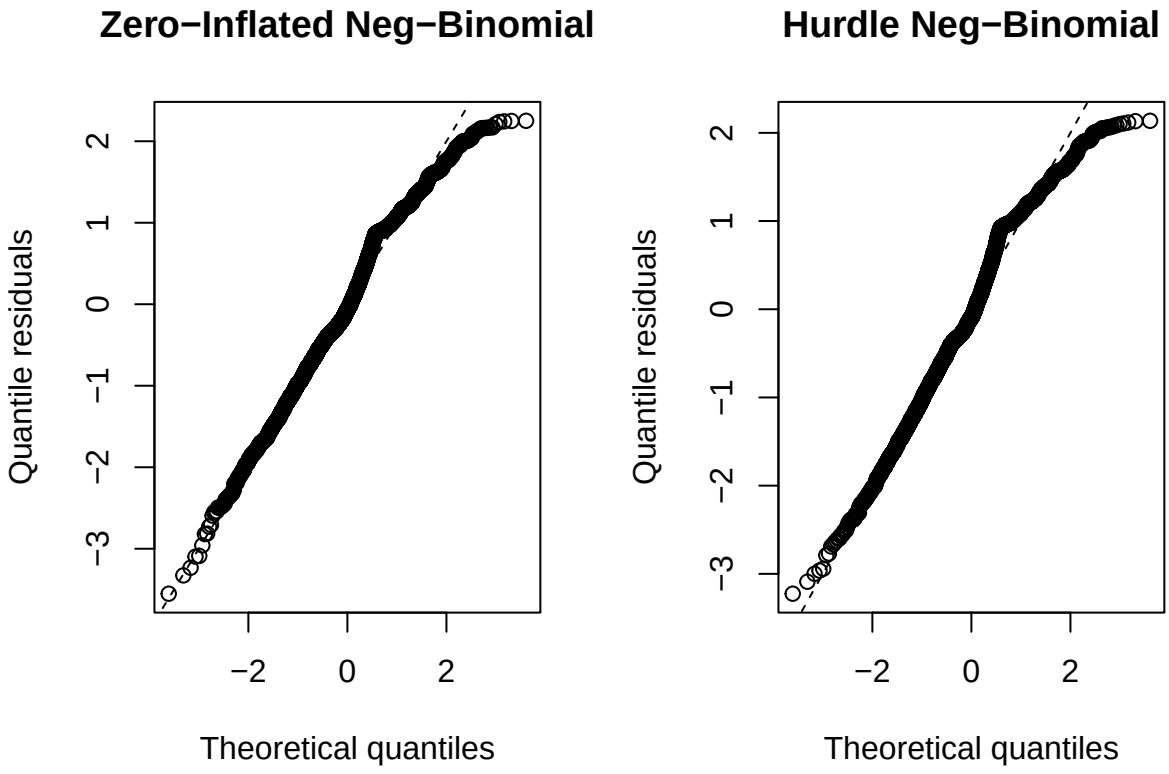
```
BIC(zi_poisson_model, h_poisson_model, zi_negbin_model, h_negbin_model)
```

```
##                  df      BIC
## zi_poisson_model 22 58637.02
```

```
## h_poisson_model  22 58637.02
## zi_negbin_model  23 22523.07
## h_negbin_model   23 22566.77
```

We clearly see from the lower BIC scores of the negative binomial models that they fit significantly better than the Poisson models. The BIC scores for the negative binomial models are very similar, but we note the zero-inflated model BIC score takes a slightly lower value.

We additionally look at quantile-quantile plots for the negative binomial models. Again the function we use is bespoke to `countreg` so we won't use `ggplot2` for this purpose.



We see that both quantile-quantile plots show no unusual departures from the model. While they do have a small bump roughly in the middle and slight curvature at the top end, this isn't a significant worry.

As both these negative binomial models seem to fit well, we choose the zero-inflated model purely due to its slightly lower BIC score. We finally make use of Vuong's non-nested hypothesis test to check that it is a better fit to our data than the original Poisson regression model we explored. We use the `vuongtest` function from the `nonnest2` package, which we add to `config.dcf`.

```
vuongtest(poisson_model, zi_negbin_model)
```

```
## Warning in imhof(n * omega.hat.2, lamstar^2): Note that Qq + abserr is positive.
```

```
##
## Model 1
##  Class: glm
##  Call: glm(formula = module_number_list_length ~ age_range + highest_education_level + ...
```

```
##
## Model 2
##  Class: zeroinfl
##  Call: zeroinfl(formula = module_number_list_length ~ age_range + highest_education_level + ...
##
## Variance test
##   H0: Model 1 and Model 2 are indistinguishable
##   H1: Model 1 and Model 2 are distinguishable
##     w2 = 164.042,   p = <2e-16
##
## Non-nested likelihood ratio test
##   H0: Model fits are equal for the focal population
##   H1A: Model 1 fits better than Model 2
##     z = -54.666,   p = 1
##   H1B: Model 2 fits better than Model 1
##     z = -54.666,   p = < 2.2e-16
```

We therefore have by this test that our zero-inflated negative binomial model is a significant improvement over the regular Poisson regression model.

Selecting the zero-inflated negative binomial model as our final model for the data, we summarise it below.

```
##
## Call:
## zeroinfl(formula = module_number_list_length ~ age_range + highest_education_level +
##     employment_status | 1, data = model_data, dist = "negbin")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -0.6288 -0.6225 -0.4766  0.5422  3.7308
##
## Count model coefficients (negbin with log link):
##                                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)                                2.849135   0.553274   5.150 2.61e-07 ***
## age_range>65                               0.732387   0.284657   2.573   0.0101 *
## age_range18-25                            -0.281794   0.264386  -1.066   0.2865
## age_range26-35                             0.025001   0.271959   0.092   0.9268
## age_range36-45                             0.239017   0.277172   0.862   0.3885
## age_range46-55                             0.452726   0.276359   1.638   0.1014
## age_range56-65                             0.638346   0.274899   2.322   0.0202 *
## highest_education_levelless_than_secondary -0.165832   0.523276  -0.317   0.7513
## highest_education_levelprofessional        -0.019244   0.494826  -0.039   0.9690
## highest_education_levelsecondary            0.127581   0.491436   0.260   0.7952
## highest_education_leveltertiary            -0.061563   0.494649  -0.124   0.9010
## highest_education_leveluniversity_degree    0.005166   0.488294   0.011   0.9916
## highest_education_leveluniversity_doctorate -0.031511   0.505072  -0.062   0.9503
## highest_education_leveluniversity_masters  -0.034973   0.490835  -0.071   0.9432
## employment_statuslooking_for_work           0.018247   0.141669   0.129   0.8975
## employment_statusnot_working                0.052427   0.157759   0.332   0.7396
## employment_statusretired                    0.247980   0.157807   1.571   0.1161
## employment_statusself_employed              0.141295   0.146386   0.965   0.3344
## employment_statusunemployed                 0.078083   0.169141   0.462   0.6443
## employment_statusworking_full_time          0.162674   0.126237   1.289   0.1975
## employment_statusworking_part_time         -0.004713   0.151123  -0.031   0.9751
```

```
## Log(theta)                                  -0.390725   0.048664  -8.029 9.82e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.8835     0.0555  -15.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 0.6766
## Number of iterations in BFGS optimization: 27
## Log-likelihood: -1.117e+04 on 23 Df
```

The model states that the `age_range` factors `>65` and `56-65` are both significant at the 5% level. We also have `age_range` factor `46-55` and `employment_status` factor `retired` approaching significance at the 10% level. An increase in any of these predictors results in a larger value of `module_number_list_length` for a learner. We also have that our zero-inflation coefficient is highly significant and we can therefore state that the behaviour of the overall distribution around zero values is different to the rest of the model. Note that in the summary given above, one of the factors for each predictor is not included as it is used by R as the base case for the factors; the rest of the factors of that category of predictor have their estimations on the dependent variable relative to that base case.

Conclusion of Analysis - Regression Model

Our final model has indicated that only a small subset of the original predictors we considered in the secondary analysis are important for describing the number of modules a learner engages with, namely if a learner ages 56 or greater. Additionally, the higher an age bracket an `age_range` factor corresponds to, the more significant it becomes. We can interpret this as older learners generally engaging with the course content more. Therefore, predominantly, the age of a learner is the most important factor in assessing a learners engagement with the course modules. The education level and working status of a learner is not important for describing their module engagement when age has been factored in.

While we could proceed further with regularised regression or a subset selection method to further improve the model, this would entail a tertiary analysis which is beyond the scope of this report.

Report Conclusion

Within this report, we considered a wide range of demographic variables corresponding to learners. We attempted to use these demographic variables to explain how a learner would engage with the online course through engaging with its individual modules. We measured engagement with modules by the quantity of which a learner accessed.

Through use of an exploratory data analysis and a regression model, we found that the age of a learner is the most important factor in dictating how they would engage with the course. The older a learner was, the more likely they were to engage with more of the course.

Therefore, a course developer should be focusing on the factors that make younger learners less likely to engage with the course, or conversely what makes older learners more likely to engage with the course. Perhaps a survey could be conducted which attempts to contrast user experience among the age groups to pinpoint precisely these factors that make younger learners less likely to engage with the course. From the data we have available currently, we cannot make sufficient judgements about this discrepancy across ages and it will require further data to identify the root causes of this age disparity in course engagement. We do note, however, that we have some data on learner feedback, within `cyber-security-x_leaving-survey-responses.csv` and `cyber-security-x_weekly-sentiment-survey-responses.csv`, but it is far too incomplete to be of any use.