SI 618 Project Part 1 Report

Using PySpark to Explore the Movie Datasets

Motivation

People nowadays love watching movies. Whether a movie is a success is always determined by its director, as directors are in charge of many important aspects of the movies. So in this project, I want to focus my analysis on movie directors. I will explore the movie datasets by PySpark to find out which directors made the most average revenues and which directors produced the highest rated movies. We could learn by merging the relevent datasets together and taking a look at the differences between the directors' average revenues and average movie ratings. Besides, I would also like to discuss the trends of female proportions in movie directors for each genre over time.

Data Sources

I used two different datasets which are described below to conduct data analysis.

1. Movie Basic Information Dataset

Source: https://www.kaggle.com/rounakbanik/the-movies-dataset#movies_metadata.csv

This dataset regarding basic information of movies is available on kaggle and can be downloaded as a CSV file. It contains 24 variables and 45,466 records which cover movies released from 1874 to 2017. Considering the completeness of the movie information, I decided to use the 29,373 records of the movies that were released from 1985 to 2015. Each record mainly shows genres, budget, revenue, original language, original title, overview, popularity, production companies, production countries, release date, runtime, release status, vote average and vote count of a movie.

The variables of interest are *genres*, *revenue*, *release_date* and *vote_average*. Each movie may have several genres, such as adventure, animation, comedy and etc. I chose to use the *vote_average* variable which shows the weighted average ratings of movies for average movie rating comparison instead of calculating the average ratings myself. Vote average applies filters to eliminate and reduce attempts at vote stuffing, which makes the ratings more accurate.

2. Cast and Crew Information Dataset

Source: https://www.kaggle.com/rounakbanik/the-movies-dataset#credits.csv

The dataset that consists of the cast and crew information of movies is also available on kaggle and could be downloaded in CSV format. It has 45,476 records and contains 3 variables which are *id*, *cast* and *crew*. Each record shows a movie's cast and crew information. I am interested in the crew information which mainly shows the department, gender, job, name of every person in the crew. Note that gender has three encoded values 0, 1, 2 which correspondingly represents 'not specified', 'female' and 'male'. I would regard the percentage of females in the total count of females and males as female proportion in movie directors.

Data Manipulation

The datasets were not in a clean and nice format and I still needed to do some data manipulations to prepare the dataset for the analysis and visualization part.

The source code can be found in **si618_project1.py**. The comments indicate the location of code for each specific part.

Step 1: Clean the data

Firstly, I kept only the necessary variables that I am interested in for convenience. I simply dropped the incomplete records and removed the movie records with no specified genres, vote average or crew infomation from the datasets, as they only account for a small part of the whole data. Furthermore, I changed all the variables to the right types because the datasets somehow had unreasonable types for some variables such as *id*.

As I mentioned above, I decided to use the records of the movies that were released from 1985 to 2015 due to the consideration for information completeness. I picked out the movies that have *status* "Released" and created a new variable *release_year* based on *release_date* to select the movies that were released in the chosen time period.

Most movies have unnormal 0 *revenue* values which probably represent missing values out of the difficulty in information acquisition. I chose to retain these movie records because of their large amount. When it comes to calculating the average revenue of each director, I would ignore these records and only take the data with positive recorded revenues into consideration.

All the manipulations in step 1 were accomplished by using the basic functions and operations in pandas.

Step 2: Find the director name and gender for each movie

I applied the eval() function and map operation in pandas to create two new variables called *director* and *gender* which show the name and gender of the director of a movie based on the variable *crew* in the Cast and Crew Information Dataset. For future analysis, I dropped those movie records with no director information.

Step 3: Join the two datasets by shared movie id column

I created RDDs for the datasets by sc.parallelize(df.values.tolist()) and formed key-value pairs by map operation. To find new insights for the movie datasets, I joined the two processed datasets by their shared id column using join operation in PySpark. The merged dataset would be used to support the 3 tasks in the next part.

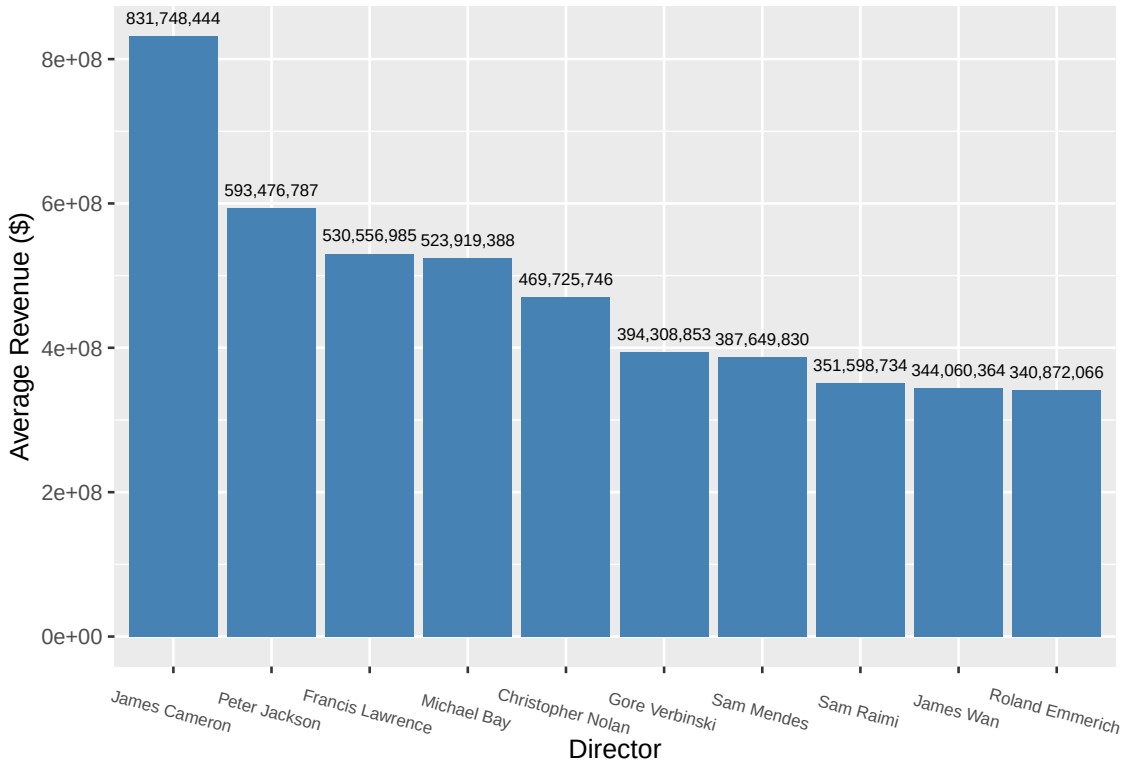

Figure 1: The top 10 average movie revenues of directors.

Analysis and Visualization

The three tasks in this part were accomplished by using large-scale computation techniques PySpark. The source code can be found in the part commented as 'analysis' from **si618_project1.py**. The visualization work was done in R using package ggplot2. This part of code is in **si618-project-part-1-xinyej.Rmd**.

Task 1: Which directors made the most average movie revenues?

The first task was to find out which directors made the most average revenue. The code corresponds to the 'analysis task 1' part in **si618_project1.py**.

For convenience, I only kept the needed variables and generated (director, (revenue, count)) pairs by map. As I mentioned before, I retained all the missing values recorded as 0 in *revenue*. So I ruled out these records before computing the average revenue of each director by filter operation. Then I used reduceByKey and mapValues operations to compute the desired statistic and applied sortBy operation to sort the results in descending order. Considering the amount of computation, I only reserved the directors which produced more than 5 movies in the time period of 1985 to 2015 by filter.

Figure 1 shows the top 10 average movie revenues and the corresponding directors who directed more than 5 movies from 1985 to 2015. The top 5 directors are James Cameron, Peter Jackson, Francis Lawrence, Michael Bay and Christopher Nolan.

We see that James Cameron, who directed 'Titanic' and 'Avatar', made the highest average profit about 832 million dollars during the period from 1985 to 2015. The average revenue that he made was around 140% the amount that the second person Peter Jackson made. It was also almost twice as much as the amount that Christopher Nolan who ranked 6th made. This shows that even though many directors are outstanding,

there is great difference between the profits that various directors can make. Moreover, the average profit difference between directors tends to decrease when the average revenue declines.

Task 2: Which directors produced the movies that have the highest average ratings?

In the second task, I tried to figure out which directors made movies with the best average ratings. The code can be found in the 'analysis task 2' part from **si618_project1.py**. I used the *vote_average* variable which shows the weighted average ratings of movies for average rating comparison.

I generated (director, (vote average, count)) tuples by map and again used reduceByKey, filter, mapValues and sortBy operations to get the sorted average movie ratings of directors in descending order. The filter operation that was mentioned above removed the directors which produced less than or equal to 5 movies from 1985 to 2015.

The output is shown in Table 1 below.

Table 1: The top 10 average movie ratings of directors.

Director	Average Rating
Don Hertzfeldt	8.067
Rocco Urbisci	7.822
Hayao Miyazaki	7.717
João César Monteiro	7.683
Krzysztof Kieślowski	7.537
Christopher Nolan	7.536
Quentin Tarantino	7.490
Lance Bangs	7.460
Dominic Brigstocke	7.443
Louis C.K.	7.425

From Table 1, we can see the top 10 average movie ratings and the directors. People gave high evaluations to the movies of Don Hertzfeldt, Rocco Urbisci, Hayao Miyazaki and so on. The highest average rating of Don Hertzfeldt was over 8, and all the top 10 average ratings were bigger than 7.4. The rating levels were quite close. It is interesting to see that the top 10 directors in the average revenue rankings and the top 10 directors in the average rating rankings barely overlapped. It may suggest that the movies that most people love to purchase are not necessarily the ones that people think highly of.

Task 3: The trends of female proportions in movie directors for each genre over time

The last task was focused on the trends of female proportions in movie directors for different genres over time. The code can be seen in the 'analysis task 3' part from **si618_project1.py**.

Firstly I needed to reorganize the dataset to produce one line of movie information for each genre by flatMap. The female proportion in a given genre was computed as the proportion of females in the total count of females and males. In order to calculate the statistic, I first formed ((genre, release year), (gender, count)) pairs by flatMap and then utilized reduceByKey and mapValues to get the desired female percentages in movie directors for each genre and each year. The trends of female proportions are shown in Figure 2 and Figure 3 which are both on the next page.

Considering the genres' large amount (20), I plotted the female proportions for the first ten genres in Figure 2 and the ones for the remaining 10 genres in Figure 3. The division did not have special meaning except for

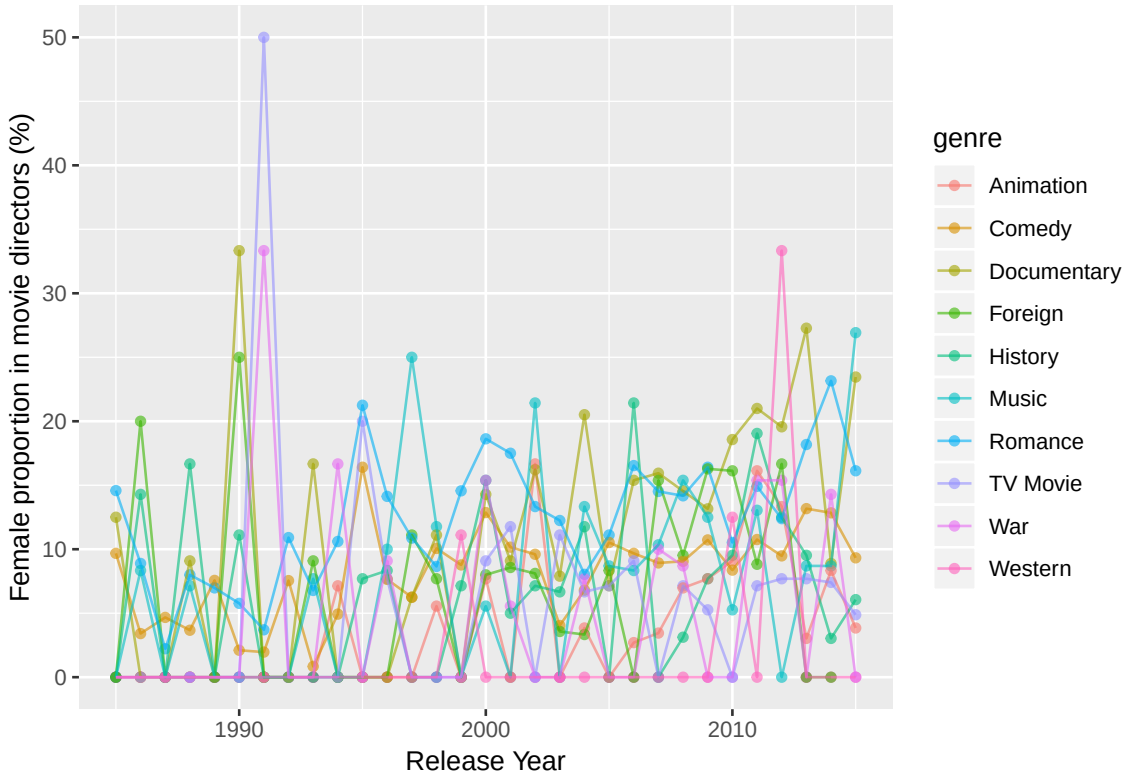

Figure 2: Female proportions in movie directors for each genre (part 1) over time.

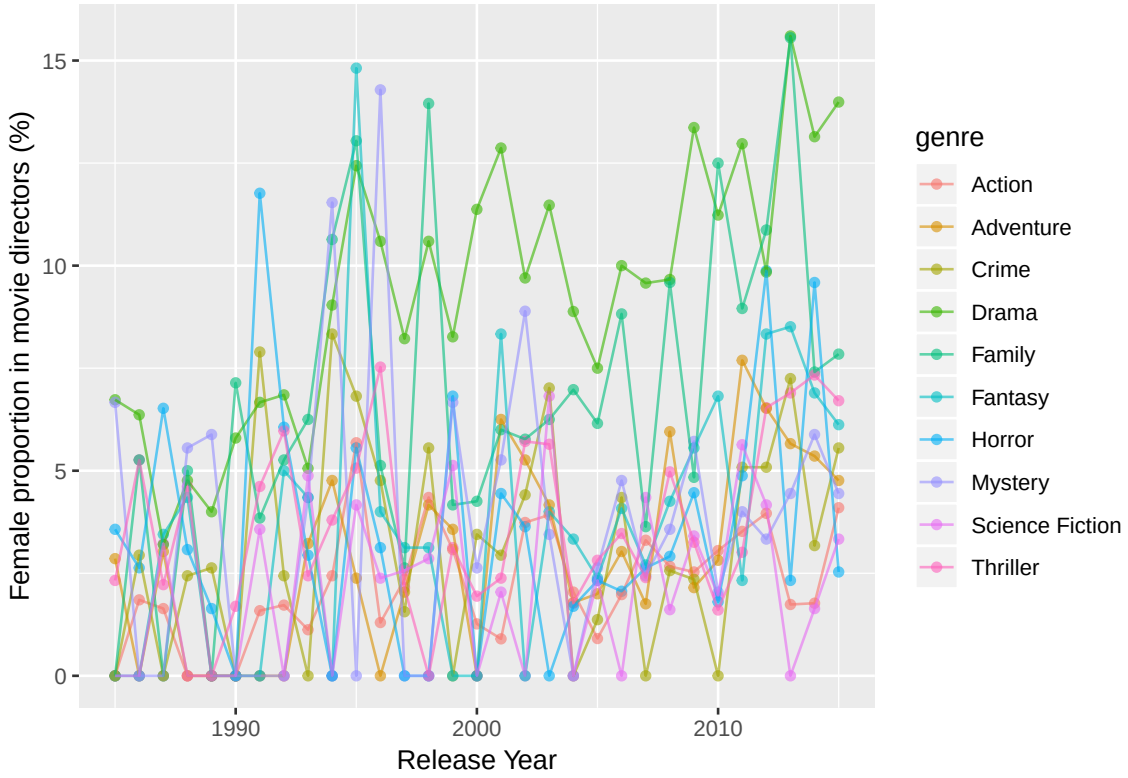

Figure 3: Female proportions in movie directors for each genre (part 2) over time.

better observation. Figure 2 covers genres such as Animation, Comedy, Documentary and so on. Figure 3 covers genres like Action, Adventure, Crime, and etc.

It is easy to see that almost all the female proportions in directors for different genres in each year were below 30%. But the overall female proportions did increase over time. For example, the female proportions for different genres barely passed 10% before 1990, yet quite a few of them were over 10% after 2010. Less genres had 0% female proportions in directors as time went on. We could see that more and more female directors directed movies of Romance, Documentary, Drama, Family, and etc.

Challenges

The biggest challenge was the difficulty in data cleaning. The datasets not only had obvious missing values, but also had missing values recorded as '[]' or 0. Besides, some variables such as id had unreasonable data types when they were read in. It is because some original records somehow mixed a few features. In order to avoid weird bugs in the following parts, I checked each variable of interest extremely carefully and got rid of all the wrong records. The difficulty in data cleaning was also the reason why I first applied pandas to read and process the datasets and then loaded them to RDDs in PySpark by sc.parallelize(df.values.tolist()). I had to investigate the datasets after every single step by interactive computing in jupyter notebook. It was much easier for me to pursue this goal in pandas dataframes than in PySpark RDDs.

The next trouble was caused by the fact that many operations in PySpark such as join and reduceByKey are only defined on Pair RDDs. To detect the problem origin, I logged onto the terminal, entered the interactive PySpark interface and tested my code line by line using the first 100 records of the datasets. After I found out the causes, I solved the problems and ran my code to get the final results.

SI 618 Project Part 2 Report

Exploratory Data Analysis of the Movie Data

Xinye Jiang

1 Motivation

People nowadays love watching movies. Whether a movie is a success can always be shown by its revenue in a way, as it is not likely for an unsuccessful movie to make money. So the general goal of this project is focused on movie revenue. To be more specific, I want to find out the trend of movie revenue, what are the influencing or related factors of revenue and how these factors influence or relate to revenue. I will accomplish this goal by performing exploratory data analysis such as extracting and visualizing the relationships between movie revenue and release time, genres and so on by data.table operations and ggplot2 functions.

The three specific questions that I decide to explore are as follows.

Question 1: How do temporal factors influence movie revenue?

Question 2: What is the relationship between movie revenue and movie budget for each genre?

Question 3: How do vote count and vote average relate to movie revenue?

2 Data Source

Movie Basic Information Dataset

Source: https://www.kaggle.com/rounakbanik/the-movies-dataset#movies_metadata.csv

This dataset regarding the basic information of movies is available on kaggle and can be downloaded as a CSV file. It contains 24 variables and 45,466 records which cover movies released from 1874 to 2017. Considering the completeness of the movie information, I decided to use the 29,373 records of the movies that were released from 1985 to 2015. Each record mainly shows the genres, budget, revenue, original language, popularity, production companies, production countries, release date, runtime, release status, vote average and vote count of a movie.

The variables of interest in this project are *id*, *revenue*, *release_date*, *runtime*, *genres*, *budget*, *vote_average* and *vote_count*. *release_date* has a data type of date. *genres* shows a list of dictionaries with a disparate genre in each dictionary. Each movie may have several genres, such as adventure, animation, comedy and etc. Other important variables are all numeric. The *vote_count* variable displays how many ratings a film's score is based on and the *vote_average* variable shows the weighted average ratings of movies. *vote_average* is a reliable measure of the movie ratings, as it applies filters to eliminate and reduce attempts at vote stuffing.

3 Methods

The dataset was not in a clean and nice format and thus some data manipulations were still needed to prepare the dataset for the three proposed questions.

The source code can be found in the corresponding part of **si618-project-part-2-xinyej.Rmd**.

3.1 Question 1: How do temporal factors influence movie revenue?

3.1.1 Question 1: Manipulation

As for data preprocessing, I firstly changed all these variables into the right data types, because some original records mixed a few features and the dataset thus had unreasonable data types for some variables like *id*. I then picked out the movies that have *status* "Released" and were released from 1985 to 2015 based on *release_date*. Finally, I kept only the necessary variables that I am interested in for convenience, i.e., *id*, *revenue*, *release_date* and *runtime* in this case. These manipulations were accomplished by the basic functions regarding the data type transformation as.integer(), as.numeric(), as.Date() and basic data.table operations in R.

To prepare for the analysis, I computed the movie counts and total movie revenues for each year by data.table operations and length(), sum(), year() functions. Besides, I calculated the average revenues respectively for each year and each month, each year and each day of week, and each month and each day of week by data.table operations and mean(), year(), month(), weekdays() functions.

3.1.2 Question 1: Missing/Incomplete/Noise

I simply dropped the incomplete records, as they only account for a small part of the whole data. Most movies have the abnormal value 0 for *revenue* which might probably represent missing values out of the difficulty in information acquisition. I chose to keep these movie records because of their large amount. When it comes to calculate the mean revenue of movies, all the retained records would be under consideration though those abnormal records did slightly affect the results.

3.1.3 Question 1: Challenges

I encountered a challenge that some variables such as *id* had unreasonable data types when they were read in. It is because some original records mixed a few features. In order to avoid weird bugs in the following parts, I checked every variable of interest extremely carefully, changed them into the right data types and got rid of all the wrong records.

3.2 Question 2: What is the relationship between movie revenue and movie budget for each genre?

3.2.1 Question 2: Manipulation

In order to prepare the data for analysis, after changing the variables into the right data types, I selected the movies that were released from 1985 to 2015 based on *status* and *release_date*. I removed those records that had "[]" as their *genres* from the dataset and retained only the variables *id*, *revenue*, *budget* and *genres* for convenience.

To carry out this analysis regarding *genres*, the dataset needed to be reorganized to have one line of movie information for each genre. I used gsub() function and regular expression to replace all the unnecessary characters with the empty string and then applied strsplit(), unlist() and sapply() functions to grasp the

genre(s) for each movie into a vector. Finally I utilized the pipe operator %>% and tidyr::unnest() function to make each genre its own row, generating the desired dataset for question 2.

3.2.2 Question 2: Missing/Incomplete/Noise

The incomplete records were dropped due to their small amount. Most movies have the abnormal value 0 for *revenue* or *budget* which might probably represent missing values. I decided to keep these records because they account for a large part of the data.

3.2.3 Question 2: Challenges

The biggest challenge was to reorganize the dataset to make each genre its own row for all the movies. It is easy to do this manipulation by flatMap in PySpark. R also provides packages to pursue this goal similarly. I used another way that processed the data mainly by string manipulation functions. I used regular expression and gsub(), strsplit(), unlist() and sapply() functions to get the genre(s) as a vector for each movie and applied pipe operator %>% and tidyr::unnest() function to obtain the desired dataset.

Another challenge was that the variable *budget* had wrong data type "factor" when it was read in and I got inconsistent numbers when I changed it into type "numeric". In order to fix this problem, I added an argument "stringsAsFactors = FALSE" when I read the original dataset using read.csv() function. In this way, I could get consistent numbers when correcting the type of *budget*.

3.3 Question 3: How do vote count and vote average relate to movie revenue?

3.3.1 Question 3: Manipulation

Similar to the manipulations in the previous questions, I changed the variables into the correct data types and retained only the movies that were released from 1985 to 2015 based on *status* and *release_date*. I reserved the variables *id*, *revenue*, *vote_count* and *vote_average* to facilitate the following computation. These data manipulations in question 3 were accomplished by the basic data.table operations and the data type transformation functions as.integer() and as.numeric() in R.

To prepare for analysis in question 3, I rounded the vote average to the nearest 0.5 and calculated the mean profit for each vote average group by round() and data.table operations. Similar manipulation was also done to the vote count that I rounded the vote count to the nearest 1000 and computed the mean revenue for each vote count group.

3.3.2 Question 3: Missing/Incomplete/Noise

I also removed the incomplete records as a result of their small amount. As mentioned before, most movies have the abnormal value 0 for *revenue* which might represent missing values. I chose to keep them as they account for most of the data.

3.3.3 Question 3: Challenges

The challenge that I had was also the same challenge that I had in question 1. Some records mixed a few features and caused some variables to have wrong data types. I solved this problem by carefully checking the variables one by one and changing them into the right types.

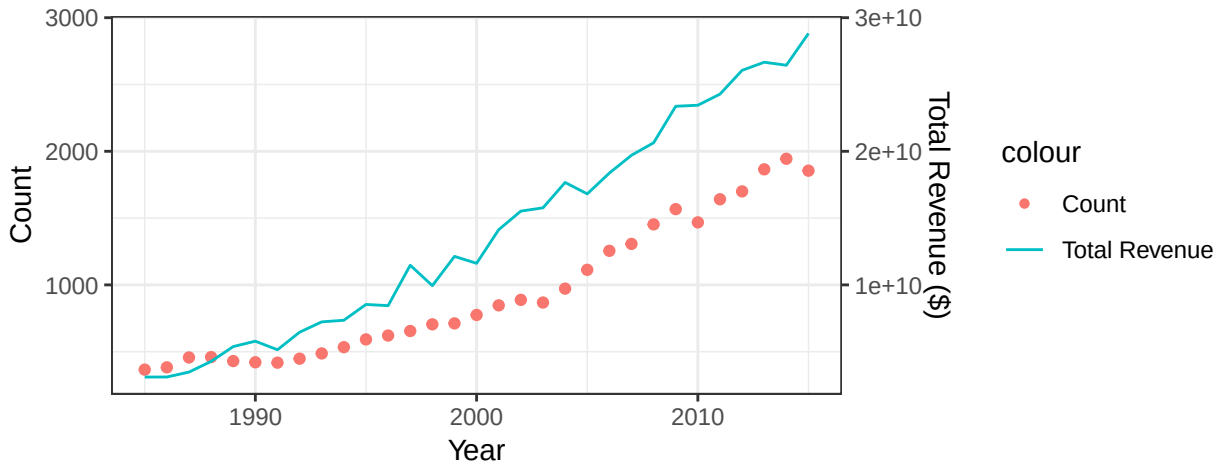

Figure 1: Overall Trend of Movie Revenues and Counts

4 Analysis and Results

The analysis and visualization work was done in R using data.table methods and ggplot2 functions. The source code can be found in the corresponding part of **si618-project-part-2-xinyej.Rmd**.

4.1 Question 1: How do temporal factors influence movie revenue?

The first data analysis was to investigate the importance of temporal factors to movie revenue and how do temporal factors influence movie revenue.

4.1.1 Question 1: Workflow of the source code

To have an overall idea of the trend of movie revenues, I first computed the total movie revenues and the number of movies for each year by data.table operations and visualized the result using lines and points in Figure 1 by ggplot2 functions. Note that I adjusted the y scales and added two different y axes on the same plot to make the results shown in one plot by setting the sec.axis argument in the scale_y_continuous() function.

Secondly, in order to find out the temporal patterns of movie revenues, I calculated the average revenues respectively for each year and each month, each year and each day of week, and each month and each day of week by data.table methods and mean(), year(), month(), weekdays() functions. I then displayed the results in Figure 2 using ggplot2 functions and grid.arrange() by three heat maps with darker color representing higher mean profits.

I finally examined the effect of a movie's runtime to its profit by a scatterplot in Figure 3 with runtime as x and revenue as y, using mainly ggplot() and geom_point() functions.

4.1.2 Question 1: Result and Visualization

Figure 1 shows that both the movie counts and total revenues have increased steadily over years, and the total revenues grow faster than movie counts. So we can conclude that more movies are produced and they overally tend to make more money as time goes on.

The first two heat maps in Figure 2 also indicate that the average profitability of movies has increased in the recent years, as in general the color is darker on the right side of the figures.

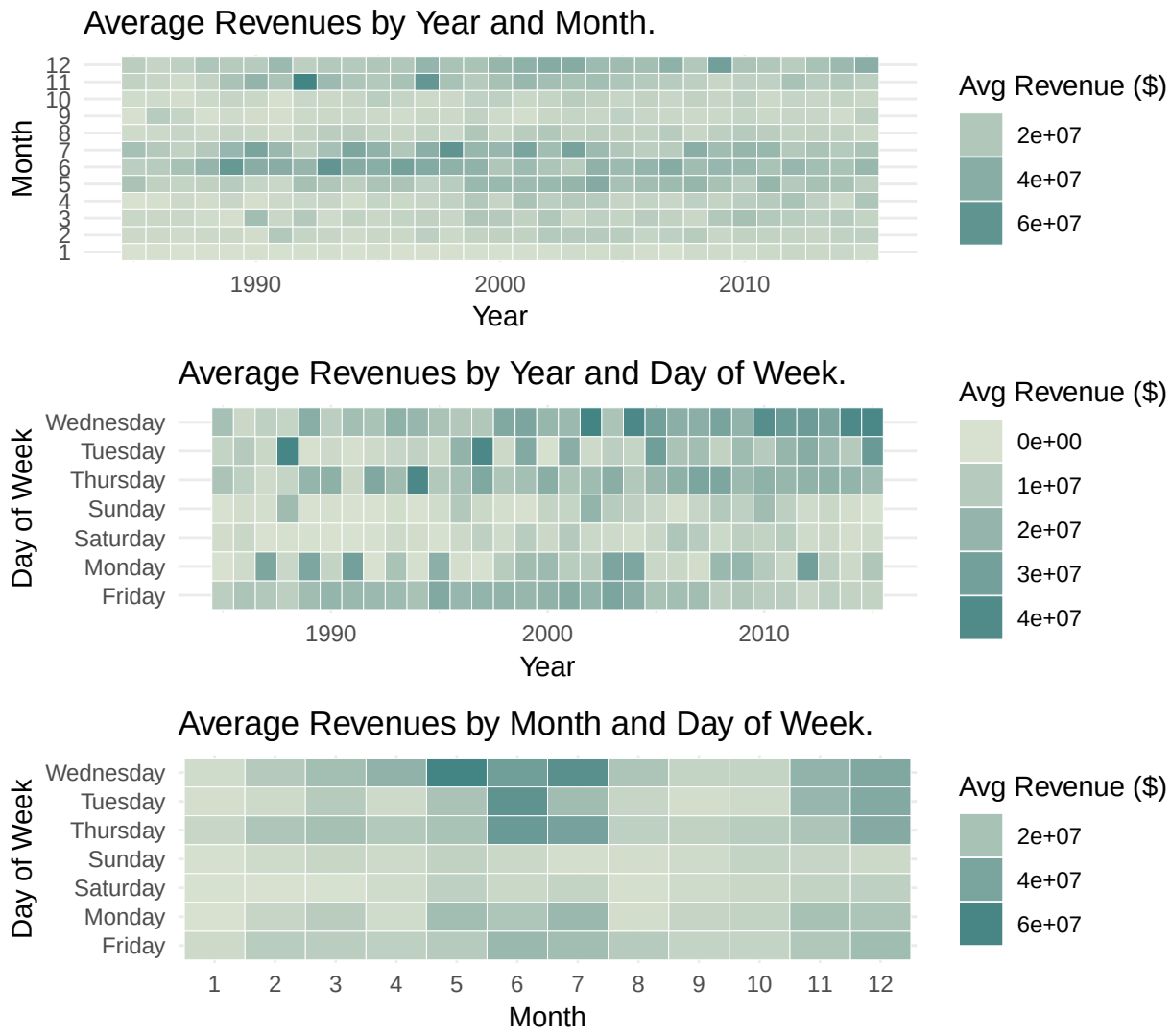

Figure 2: Temporal patterns of average revenues

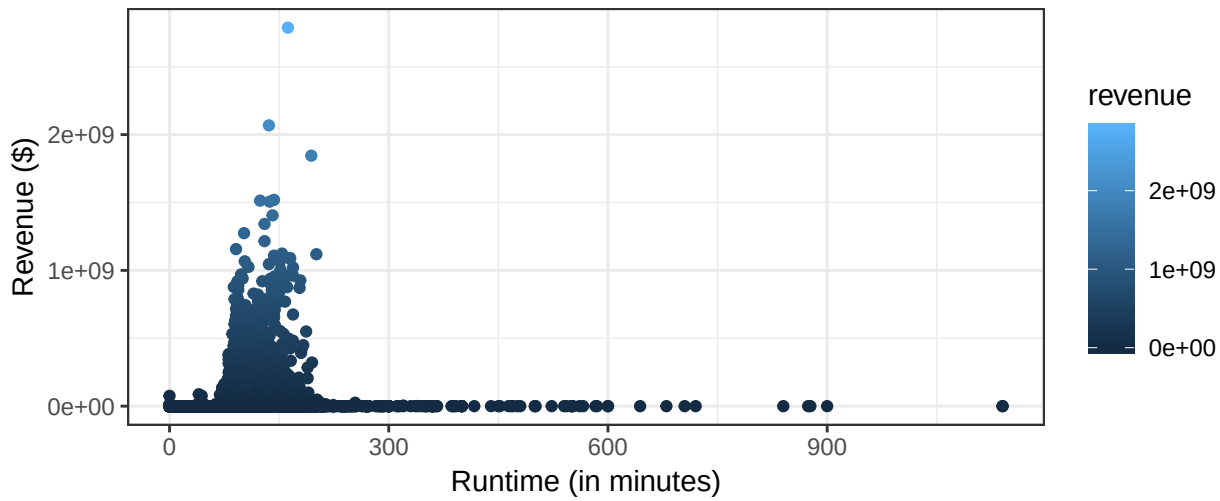

Figure 3: Revenue vs Runtime

The first and third heat maps show that months like June and July, followed by May, November and December, are particularly important for movie releases, as the average revenue of movies released in these months can earn 2 times more than the average revenue in other months. It might result from the fact that people, especially some groups such as students, may have more free time to watch movies during summer and winter. From the first heat map, the average revenue differences between months decrease over years, probably due to the convenience to watch films nowadays.

The last two heat maps indicate that movies released on Tuesday through Thursday are most likely to earn better profits, while those released during the weekends have a tendency to make less money. This might be because people can make plans in advance to see the movies that are released and promoted for a few days on Friday night or at the weekend.

Figure 3 has no specific pattern and the range of revenues seem random for movies of different runtimes. Movies that have extremely long or short runtimes tend to have no record for *revenue*, i.e., revenue value 0 in this case. Runtime is not likely to have much effect on movie revenue.

In conclusion, movies have increasing numbers and average profitability over years. Movies released on Tuesday through Thursday in June and July overally earn best revenues. The runtime of a movie does not have much effect on its revenue.

4.2 Question 2: What is the relationship between movie revenue and movie budget for each genre?

In the second analysis, we looked into the relationship between revenue and budget for each genre.

4.2.1 Question 2: Workflow of the source code

To check how budget influences revenue for various genres, I utilized scatterplots to show the relationships. Considering the intelligibility of the plots, I showed the plots of 20 genres in four rows by using %in% (to filter data), data.table operations and grid.arrange(..., nrow=4). I also added grey points and a black "y=x" line for better comparison of budget and revenue for each genre by geom_point(color="grey"), geom_abline(intercept=0, slope=1) and facet_grid(.~genres).

4.2.2 Question 2: Result and Visualization

Figure 4 shows the relationship between revenue and budget for each genre. We see that various genres produce different numbers of movies. The documentary, foreign and tv movie genres have the least number of movies. Movies of the adventure, fantasy, science fiction and action genres, however, account for a large part of all the movies.

The scatterplots also show the positive correlation between profit and budget. In general, movies that have higher budgets seem to actually see the benefits of these budgets through their higher profits. It is understandable that higher budget movies normally could attract more people, as the high budget might be used to hire well-known actors or actresses, build elaborate sets, or acquire rights regarding some famous topics.

Most genres except history, foreign and tv movie produce movies that overally have revenues higher than their budgets, so generally movies are profitable. Adventure, fantasy, science fiction and action movies normally have larger budgets along with extremely high profits. Popular movies, such as Avatar, Avengers and Pirates of the Caribbean, mostly belong to these genres and could make quite a fortune despite their large budget. Movies of the genres animation, comedy, drama, family, romance and so on have moderate budgets and high revenues. And documentary and music movies have small budgets and moderate revenues.

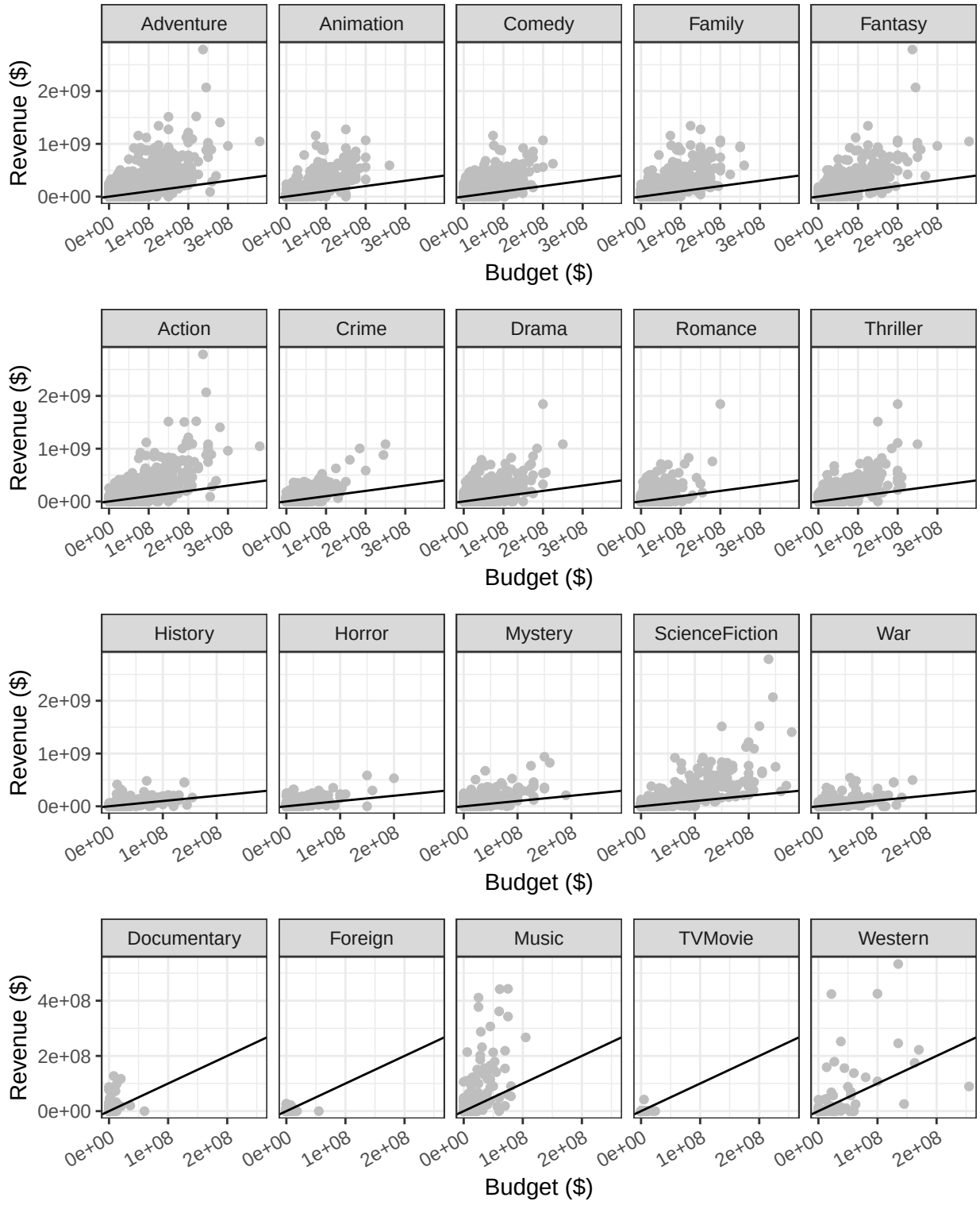

Figure 4: The relationship between revenue and budget for each genre

4.3 Question 3: How do vote count and vote average relate to movie revenue?

The goal of the third question was to examine the relationships between movie revenue and vote count, vote average. *vote_count* here displays the number of ratings and *vote_average* shows the weighted average ratings of movies.

4.3.1 Question 3: Workflow of the source code

I applied the scatterplot to show the desired result by setting vote average as x, movie revenue as y and vote count as point size and point color using ggplot(..., aes(x=vote_average, y=revenue)) and geom_point(aes(size=vote_count, color=vote_count), alpha=0.8). I also added scale_color_gradient(low="blue", high="red") to create a two color gradient showing the values of vote count from low to high.

In order to see the relationships more clearly, I showed the results for vote average and vote count separately by bar plots in Figure 6. I rounded the vote average to the nearest 0.5 and calculated the mean profit for each vote average group by round() and data.table operations and visualized the result by geom_bar(stat="identity"). Similar manipulation was also done to vote count that I rounded the vote count to the nearest 1000 and visualized the mean profit for each vote count group.

4.3.2 Question 3: Result and Visualization

The relationships between movie revenue and vote count, vote average are shown in Figure 5 and Figure 6.

Figure 5 indicates that despite that some high-rated movies have very small vote count value, there is positive correlation between vote count and vote average. In other words, high rating always comes along with moderate or high vote count. We could see that movies with higher vote count tend to make more profits, as the bigger and redder points representing the movies with higher vote count seem to correspond to larger revenues in Figure 5. Vote average also seems to be positively related to movie revenue.

Figure 6 shows the results for vote average and vote count separately. We see a pattern that the average revenue first increases when vote average increases until vote average reaches around 7.5 and then it begins to decrease when the vote average increases. After checking the data, I found out several factors that lead to this pattern. One is that the extremely high average rating like 10.0 for a movie is always made by only one person, i.e., vote count of this movie equals 1. So in this case, this rating is doubtable, as it does not reflect an honest overall evaluation of the movie. Another reason is probably that movies with high average rating over 8.5 are not necessarily the movies that are popular and are loved by everyone, and thus they have merely mediocre performance on revenue.

The average revenue also first increases and then decreases when vote count increases in Figure 6. I checked the data and found that this change might result from the fact that there is only one movie with the high vote count over 14,000 in this dataset. So the observed pattern is unreliable. We suppose that movie revenue normally first increases and later flattens when vote count increases.

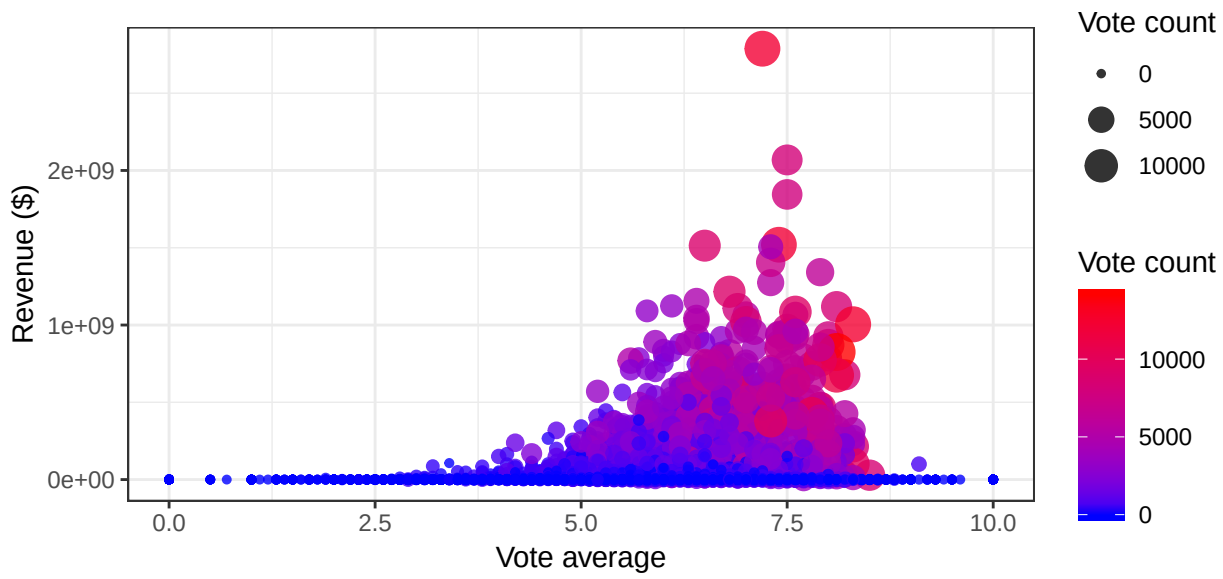

Figure 5: Revenue vs Vote average and Vote count

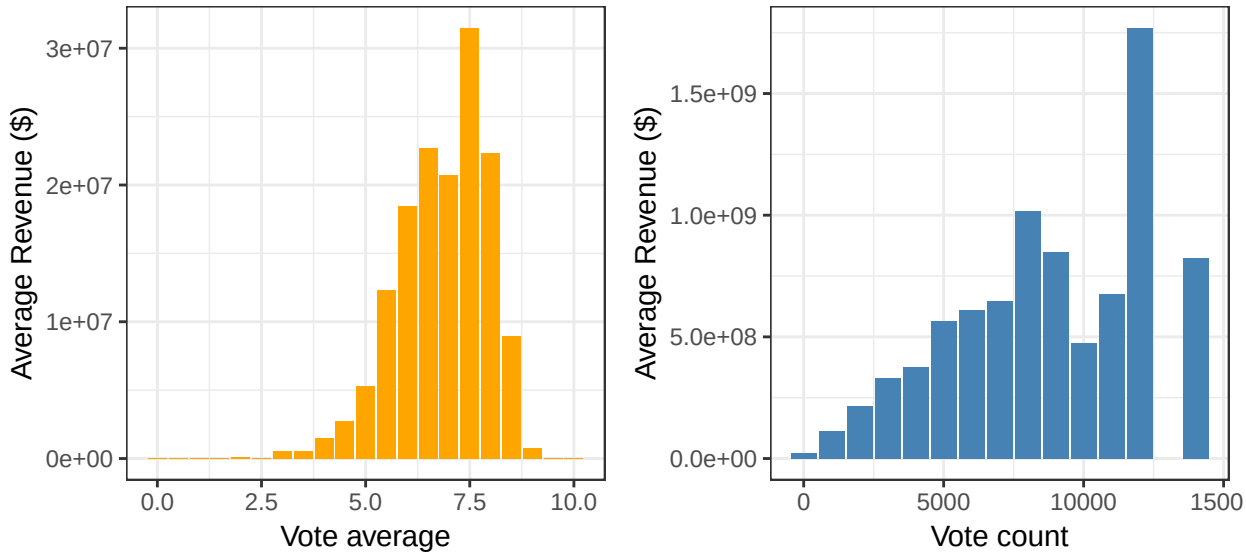

Figure 6: The relationships between revenue and vote average, vote count separately