Netflix Insights: A Data-Driven Analysis

Shiva Sai Vummaji

July 30, 2025

1 Introduction

In today's world, Digital Streaming is a major part of almost every individual's life, as it provides them with a way to consume entertainment. Among many digital streaming platforms, Netflix is undoubtedly one of the largest, with over 247.15 million current subscribers. The real-life data set that interests me is obtained from Kaggle and includes information about movies and TV shows on Netflix from 1945 to 2022. Specifically, the dataset includes: **title**, **type**, **release year**, **age certification**, **runtime**, **genres**, **production countries**, **seasons**, **imdb id**, **imdb score**, and **imdb votes**.

This dataset can be found at: Netflix Movies and TV Shows (https://www.kaggle.com/datasets/maso0dahmed/netflix-movies-and-shows)

1.1 Dataset

Here, I will attach a screenshot of how a part of the dataset looks like after reading the csv file and using the View() function in R.

	title	type	release_year	age_certification	runtime	genres	production_countries	seasons	imdb_id	imdb_score	imdb_votes
1	Five Came Back: The Reference Films	SHOW	1945	TV-MA	48	['documentation']	['US']	1		NA	NA
2	Taxi Driver	MOVIE	1976	R	113	['crime', 'drama']	['US']	NA	tt0075314	8.3	795222
3	Monty Python and the Holy Grail	MOVIE	1975	PG	91	['comedy', 'fantasy']	['GB']	NA	tt0071853	8.2	530877
4	Life of Brian	MOVIE	1979	R	94	['comedy']	['GB']	NA	tt0079470	8.0	392419
5	The Exorcist	MOVIE	1973	R	133	['horror']	['US']	NA	tt0070047	8.1	391942
6	Monty Python's Flying Circus	SHOW	1969	TV-14	30	['comedy', 'european']	['GB']	4	tt0063929	8.8	72895
7	Dirty Harry	MOVIE	1971	R	102	['thriller', 'crime', 'action']	['US']	NA	tt0066999	7.7	153463
8	My Fair Lady	MOVIE	1964	G	170	['drama', 'music', 'romance', 'family']	['US']	NA	tt0058385	7.8	94121
9	The Blue Lagoon	MOVIE	1980	R	104	['romance', 'drama']	['US']	NA	tt0080453	5.8	69053
10	Bonnie and Clyde	MOVIE	1967	R	110	['drama', 'crime', 'action']	['US']	NA	tt0061418	7.7	111189
11	The Professionals	MOVIE	1966	PG-13	117	['western', 'action', 'european']	['US']	NA	tt0060862	7.3	16168
12	The Guns of Navarone	MOVIE	1961		158	['war', 'action', 'drama']	['US', 'GB']	NA	tt0054953	7.5	50150
13	Lupin the Third: The Castle of Cagliostro	MOVIE	1979	PG	100	['comedy', 'animation', 'action', 'fantasy', 'family']	['JP']	NA	tt0079833	7.6	30277
14	Richard Pryor: Live in Concert	MOVIE	1979	R	78	['comedy', 'documentation']	['US']	NA	tt0079807	8.1	5141
15	The Long Riders	MOVIE	1980	R	99	['western', 'crime']	['US']	NA	tt0081071	6.9	11329
16	White Christmas	MOVIE	1954		115	['romance', 'comedy', 'music']	['US']	NA	tt0047673	7.5	42373
17	Cairo Station	MOVIE	1958		77	['drama', 'crime', 'comedy']	['EG']	NA	tt0051390	7.5	4385
18	The Queen	MOVIE	1968		68	['documentation']	['US']	NA	tt0183686	7.2	1117
19	Hitler: A Career	MOVIE	1977	PG	150	['documentation', 'history', 'european']	['DE']	NA	tt0191182	7.5	2416
20	FTA	MOVIE	1972	R	97	['comedy', 'documentation', 'music']	['US']	NA	tt0068562	6.2	411
21	Saladin the Victorious	MOVIE	1963		186	['drama', 'war', 'action', 'history', 'romance']	['EG']	NA	tt0057357	7.6	2470
22	Singapore	MOVIE	1960		158	['drama', 'thriller', 'crime']	['IN']	NA	tt0268639	6.4	82
23	Dark Waters	MOVIE	1956		120	['drama', 'action', 'romance', 'thriller']	['EG']	NA	tt0049761	6.7	590
24	Alexandria... Why?	MOVIE	1979		133	['drama']	['EG']	NA	tt0077751	7.2	1689
25	Raya and Sakina	MOVIE	1953		105	['drama', 'thriller', 'crime', 'history']	['EG']	NA	tt0316472	6.8	231
26	No Longer Kids	MOVIE	1979		235	['comedy', 'drama']	['EG']	NA	tt8312792	9.0	943
27	Amrapali	MOVIE	1966		120	['fantasy']	['IN']	NA	tt0060104	6.7	225
28	Dostana	MOVIE	1980		161	['drama', 'comedy', 'romance', 'action', 'crime']	['IN']	NA	tt0080653	2.1	25

2 Part 1 - Variables

2.1 Title

The **title** column includes the name of the movie or TV show for each of the 5806 entries. Below is a screenshot of this variable's summary:

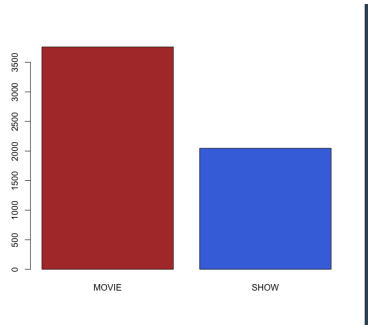
```
> summary(netflix_data$title)
   Length     Class      Mode
     5806 character character
```

2.2 Type

The **type** column includes information about whether an entry is a movie or a TV show. This is a categorical variable. I created separate data frames for movies and tv shows for better clarity:

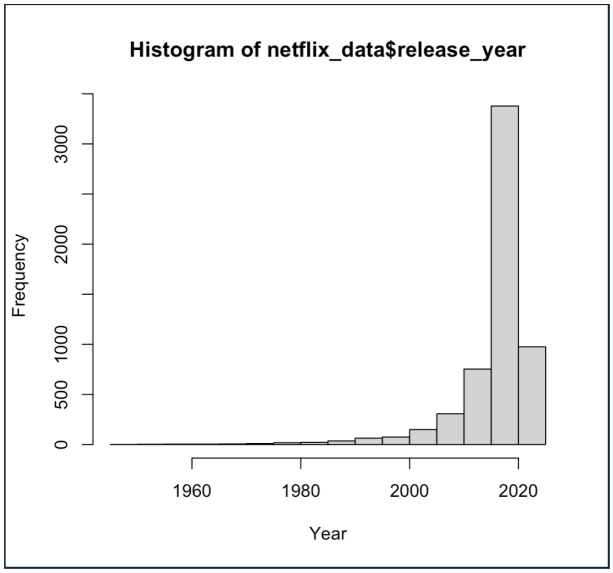
```
# Creating a new dataframe for just shows
movie_data = netflix_data[netflix_data$type == "MOVIE", ]
View(movie_data)

# Creating a new dataframe for just shows
show_data = netflix_data[netflix_data$type == "SHOW", ]
View(show_data)
```

There are 3759 entries in the movies only data frame and 2047 entries in the shows only data frame.
Then, I analyzed the distribution of movies and TV shows using a barplot.

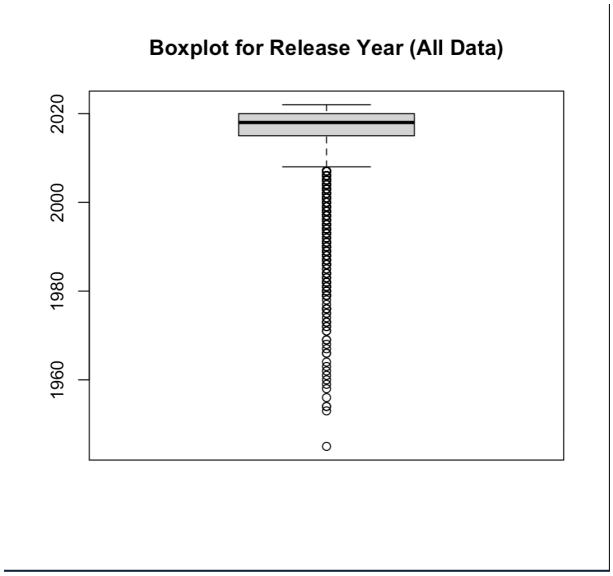


2.3 Release Year

The **release_year** column provides us with information about when a movie/show was released. Specifically, it gives us the year that each entry was released in, from 1945 to 2022. Below is a histogram of release year for all the data:



In order to look more closely at the median, quartiles, and to see if there are any outliers, I also created a boxplot.



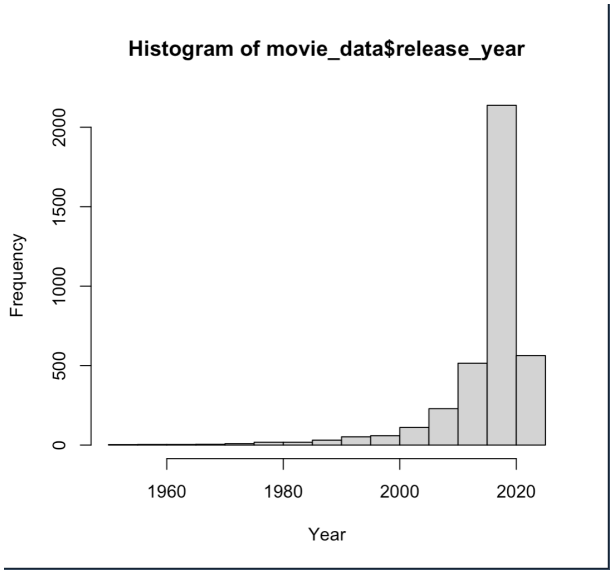
It is clear from the boxplot and the histogram that the median is very recent

(close to 2018-2020). Here are the specific summary statistics:

```
> summary(netflix_data$release_year)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1945    2015    2018    2016    2020    2022
```

These statistics support both the histogram and boxplot since they reveal that the median occurs in 2018, and give the quartile range from 2015 to 2022, which could help explain why there would be a lot of outliers.

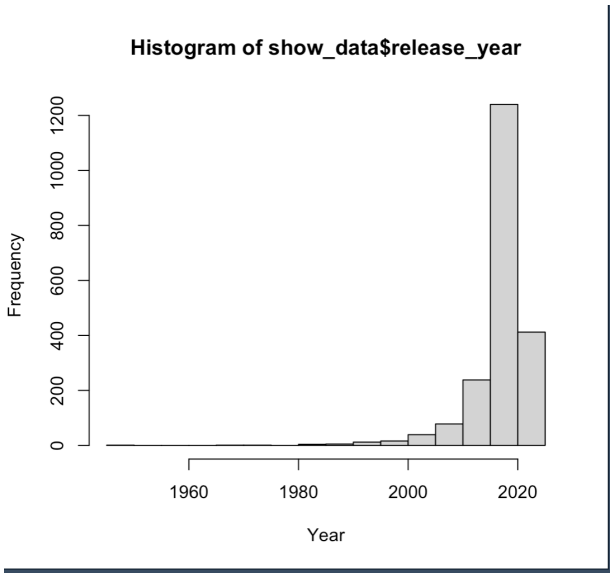
In order to understand the data with more clarity, I created histograms and found summary statistics for only movies data and only shows data. Below is a histogram that shows the distribution of release year for movies only:



Here are the specific summary statistics for release year of movie data:

```
> summary(movie_data$release_year)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1953    2015    2018    2015    2020    2022
```

I did the same for shows data as well. Here is the histogram:



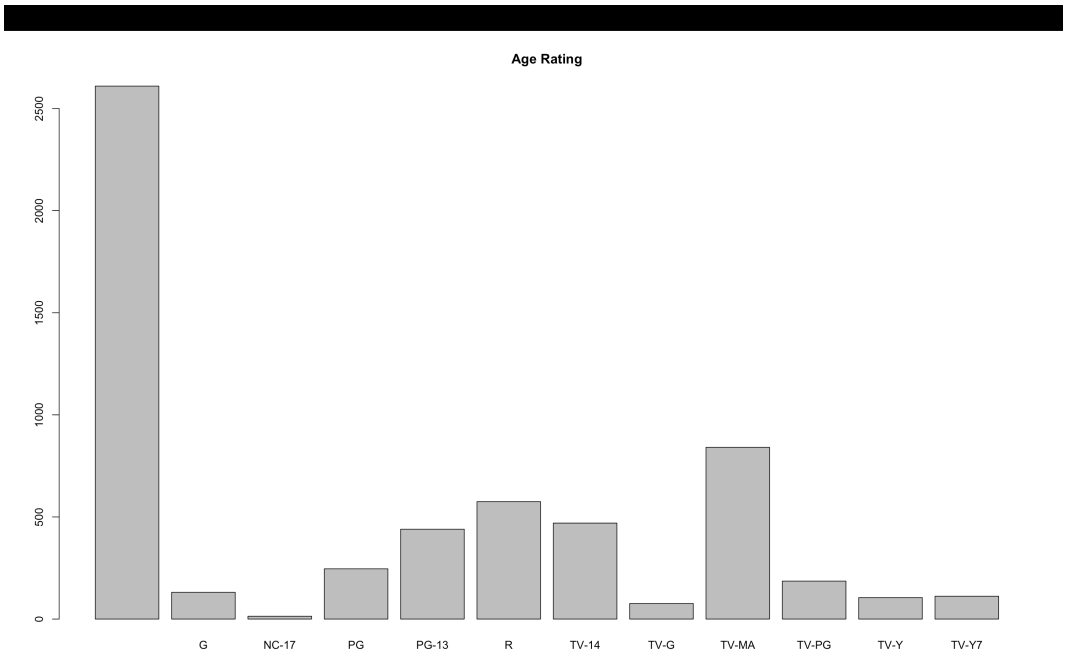
Here is the specific summary statistics:

```
> summary(show_data$release_year)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1945    2016    2019    2017    2020    2022
>
```

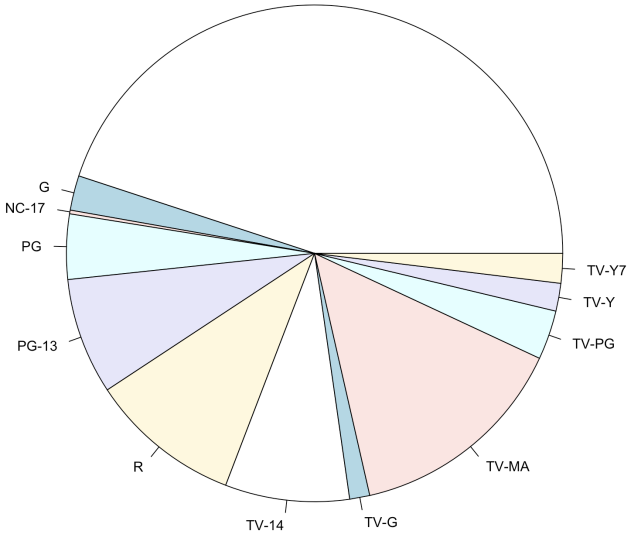
There isn't much difference between the distribution of release year for shows and movies, since they both follow similar trends. In both cases, it is clear that the majority of movies are released later (after 2015/2016). Additionally, their mean and median years are also very similar, as they are only different by 1 year. It is also important to note that all three histograms (for all data, just for movies, and just for shows) are left-skewed, indicating the fact that there are more entries that are recent, as there is more data on the right side of the graph, and we can also see that the median is greater than the mean in all cases.

2.4 Age Certification

The **age_certification** variable provides us with information about the age rating for each movie. The ratings that are used are similar to the ones used on Netflix: G, NC-17, PG, PG-13, R, TV-MA, TV-14, TV-G, TV-Y, TV-Y7. Additionally, it is important to note that some entries do not have an age rating associated with them. Below is a barplot that shows the distribution of the age rating:



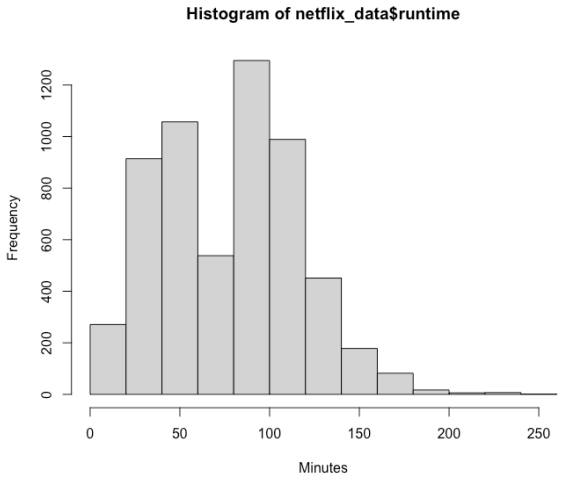
I also created a pichart:



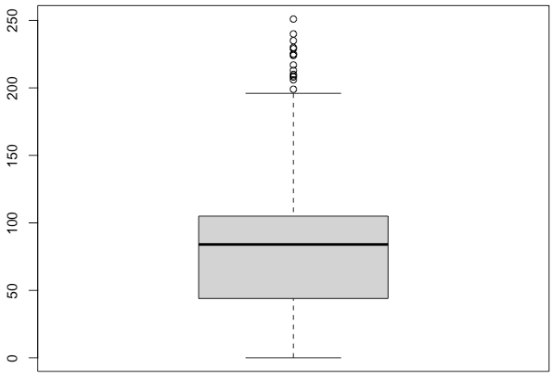
The barplot and pichart above reveal that many of the entries do not have an age rating associated with them. However, among the ones that do, it looks like TV-MA, TV-R, PG-13, and TV-14 are the most common ones.

2.5 Runtime

The **runtime** variable provides information about how long a movie or show is in minutes. This is a continuous numerical variable. It is important to see that there is a clear variation in terms of the runtime for movies and shows, which makes sense because movies are typically longer than an episode in a show. Below is a histogram for runtime of all the data (both movies and shows included):



The histogram above looks right-skewed. In order to further explore this data, I created a boxplot:

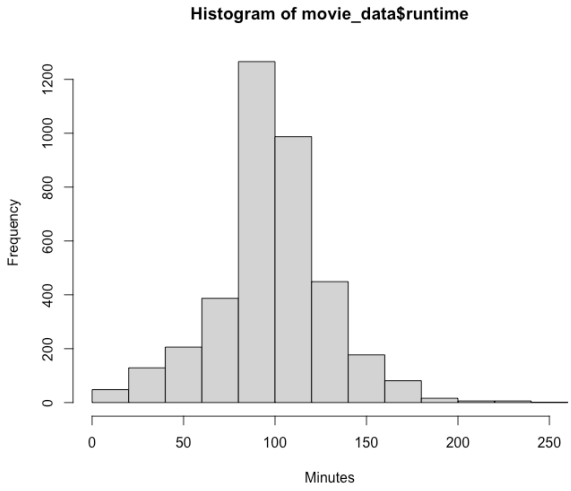


It is clear from the boxplot and the histogram that the median is between 50 to 100. It can also be seen that there are many outliers above the 3rd Quartile in the boxplot graph. Here are the specific summary statistics:

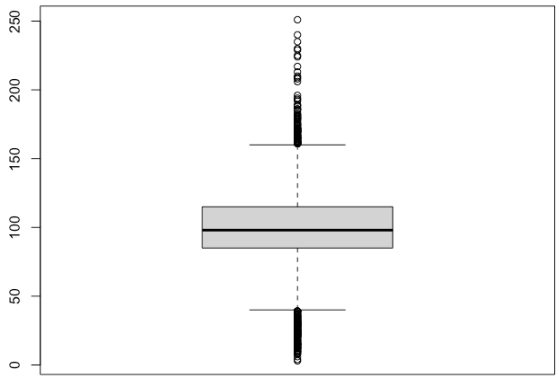
```
summary(netflix_data$runtime)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.00   44.00   84.00   77.64  105.00  251.00
```

These statistics support both the histogram and the boxplot above because they reveal that the median occurs at 84. The boxplot also shows the minimum and maximum values, which match the output from the summary statistics.

In order to understand the data with more clarity and better understand the difference between runtime for shows and movies, I created histograms, boxplots, and found summary statistics for data with only movies and data with only shows. Below is a histogram that shows the distribution of runtime for movies only data:



Here is my boxplot for movies only data:

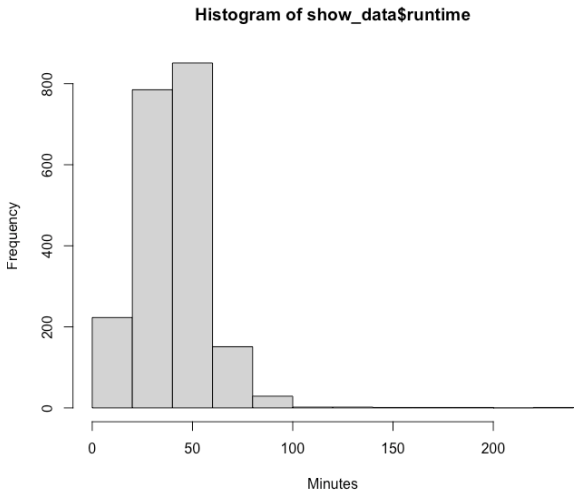


Here are my summary statistics for movie only data:

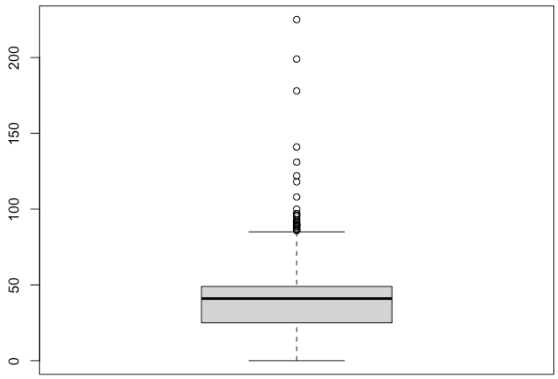
```
summary(movie_data$runtime)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 3.00   85.00   98.00   98.79  115.00  251.00
```

As we can see above, the summary statistics match both the boxplot and the histogram as the graphs show the median to be very close to 100. Additionally, the boxplot reveals that there seem to be outliers on both the minimum and maximum ends of the runtime data. It can also be seen here that the mean and medians are higher for data with movies only compared to overall data, which makes sense because movies are typically longer than episodes.

I did the same for shows only data as well. Here is the histogram:



Here is my boxplot for shows only data:



Here are my summary statistics for the show only data:

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00   25.00   41.00   38.82   49.00  225.00
```

As we can see above, the summary statistics match both the boxplot and the histogram as the graphs show the median to be very close to 40. Additionally, the boxplot reveals that there seem to be outliers on the maximum end of the data (greater than third quartile). It can also be seen that the mean and medians for data with shows only are lower compared to movies only and overall data, which makes sense because episodes are typically shorter than movies.

2.6 Genre

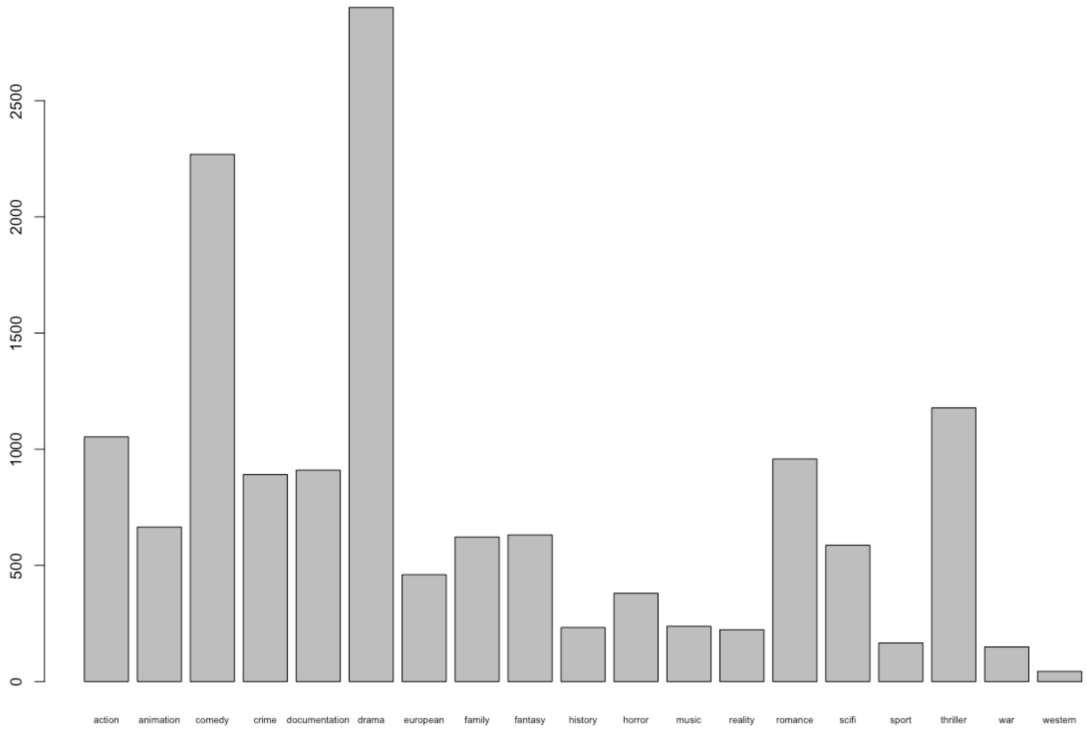
The **genre** variable provides information about what type of genre a given movie or show is such as drama, crime, comedy, etc. The format of it is a list of strings, separated by commas. It is important to note that the number of genres for each entry varies because some entries have 1, some have 4, etc. This is a categorical variable.

Since the genre column does not just have one genre for each entry and instead included multiple separated by commas, I split them and created a new data frame that holds these genres separately:

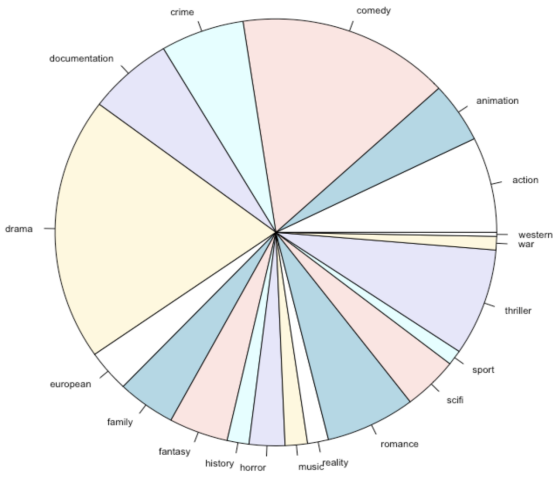
```
g = unlist(strsplit(gsub("\\[|\\]|'", "", netflix_data$genres), ", "))
g_df = data.frame(genre = g)
View(g_df)
g_c = table(g_df$genre)
```

This allows for the counting of each genre individually, so the distribution of each individual genre can be analyzed. Here is a barplot that shows the distri-

bution among different genres:



Additionally, I also created a piechart:



As both the barplot and piechart reveal, the most common genres among the netflix dataset seem to be drama, comedy, action, crime, thriller, romance, and documentation.

Here are the specific counts for each genre:

action	animation	comedy	crime	documentation	drama	european	family	fantasy	history	horror
1053	665	2269	891	910	2901	460	622	631	233	380
music	reality	romance	scifi	sport	thriller	war	western			
238	223	958	587	166	1178	149	44			

As we can see, these count values match with the piechart and barplot shown above.

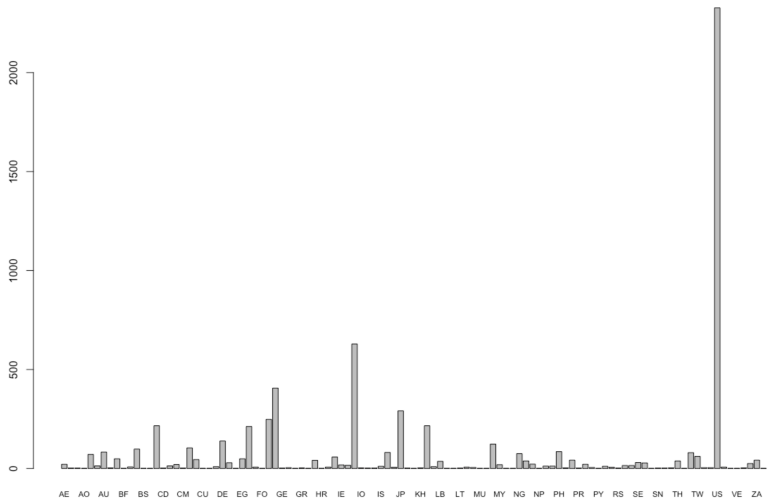
2.7 Production Countries

The **production_countries** variable provides information about the country where the specific movie or show was produced in. The format of it is similar to that of genres, where countries are separated by commas, as some movies or shows can be produced in multiple countries. This is a categorical variable.

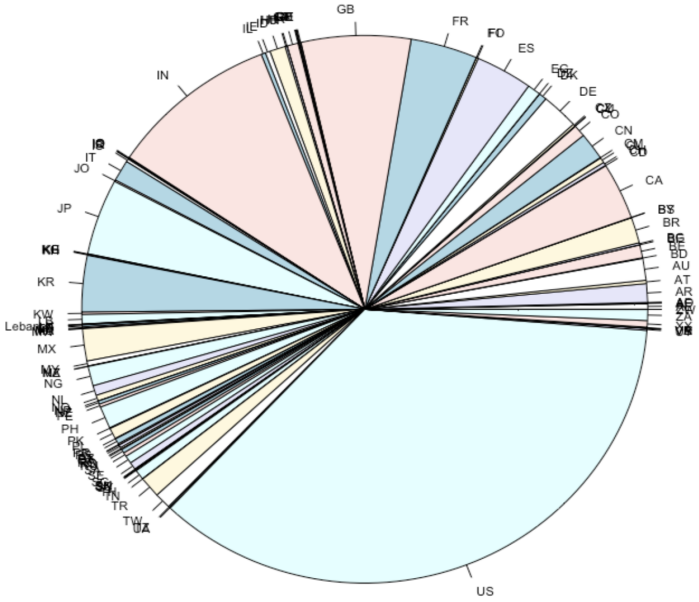
Since the production_countries column does not just have one country for each entry and instead includes multiple separated by commas, I split them and created a new dataframe that holds countries separately:

```
p = unlist(strsplit(gsub("\\[|\\]|'", "", netflix_data$production_countries), ", "))
p_df = data.frame(country = p)
p_c = table(p_df$country)
```

This allows for the counting of each country individually, so the distribution of each individual country can be analyzed. Here is a barplot that shows the distribution of each country:



Here is a pichart that I made for this variable:



As both the barplot and piechart reveal, the most common production countries among the netflix dataset seem to be U.S, India, and Great Britian.

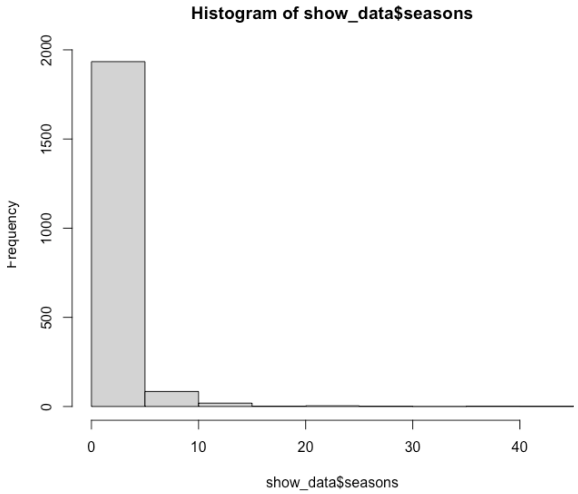
Here are the specific counts for each country:

AE	AF	AL	AO	AR	AT	AU	BD	BE	BF	BG	BR	BS	BY	CA	CD	CH	CL	CM
21	2	2	1	71	13	83	3	49	1	8	98	1	1	216	2	13	20	2
CN	CO	CU	CY	CZ	DE	DK	DZ	EG	ES	FI	FO	FR	GB	GE	GH	GL	GR	GT
104	45	1	1	9	139	29	1	49	212	7	1	248	406	2	4	1	3	1
HK	HR	HU	ID	IE	IL	IN	IO	IQ	IR	IS	IT	JO	JP	KE	KG	KH	KR	KW
41	1	7	58	18	16	629	3	2	2	11	81	6	291	2	1	3	216	9
LB	Lebanon	LK	LT	LU	MA	MU	MW	MX	MY	MZ	NA	NG	NL	NO	NP	NZ	PE	PH
36	1	1	2	7	5	1	1	123	19	1	1	75	38	22	1	12	12	85
PK	PL	PR	PS	PT	PY	QA	RO	RS	RU	SA	SE	SG	SK	SN	SU	SY	TH	TN
3	42	2	21	5	1	11	6	2	15	15	30	28	1	2	2	3	38	1
TR	TW	TZ	UA	US	UY	VA	VE	VN	XX	ZA	ZW							
80	61	4	4	2327	7	1	1	3	25	42	1							

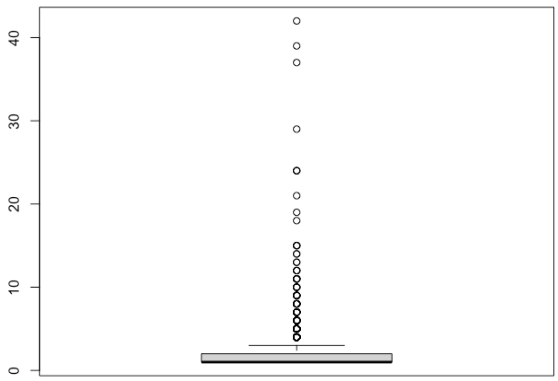
As we can see, the count values match with the ones in the piechart and barplot shown above.

2.8 Seasons

The **seasons** variable provides information about how many seasons each entry has. For instance, a show could have four seasons, 1 season, etc. It is also important to note that for movies, the seasons is *NA*, which makes sense because movies typically do not have seasons. This is a discrete variable since it can be countable. Since the seasons column only really applies for shows only, I visualized it with the shows only data frame that I created earlier. Below is a histogram that shows the distribution of seasons:



I also created a boxplot to understand the median, quartiles about the seasons information.



As we can see above, it is clear that the majority of shows have very low number of seasons. However, there do seem to be outliers that have many seasons.

Here are the summary statistics:

```
> summary(show_data$seasons)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   1.000   2.166   2.000  42.000
>
```

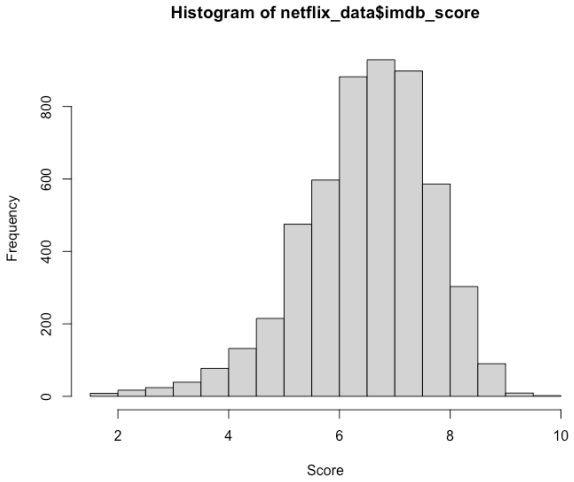
It is clear that the resulting information from the summary statistics matches both the boxplot and the histogram since the median is very low (1 season) and the mean is also low (approximately 2 seasons).

2.9 IMDb ID

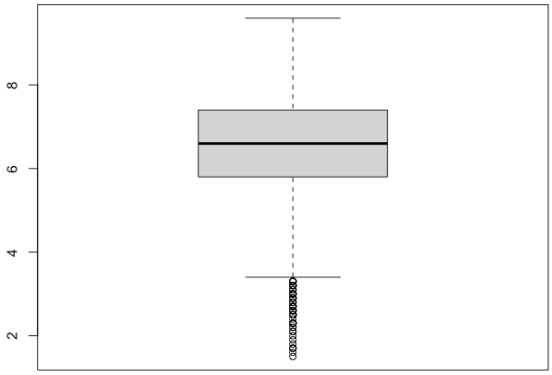
The **imdb_id** variable provides information about the specific imdb identifier for the specific movie or show that can be matched on the IMDb database. This id is unique to each entry and an example looks like: *tt9184982*. This is a categorical variable.

2.10 IMDb Score

The **imdb_score** variable provides information about the rating found on IMDb database for each entry. The score is on a 10-point scale that is rounded to one decimal place, so it is a rounded continuous variable. Here is a histogram that shows the distribution for each score:



Here is a boxplot for the score:



As we can see from the histogram and boxplot above, it appears that the median score rating is between 6 and 8. Additionally, the histogram slightly looks like a bell-curve, although it is very slightly left-skewed.

Here are the summary statistics for score:

```
summary(netflix_data$imdb_score)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
1.500   5.800   6.600   6.533   7.400   9.600     523
```

The resulting information from this summary statistics supports the boxplot and histogram as it reveals that the median is 6.6. Additionally, the quartiles, minimum, and maximum values also match the boxplot values.

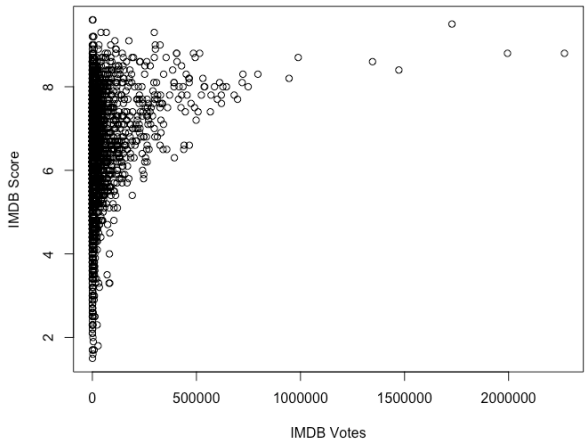
2.11 IMDb Votes

The **imdb_votes** variable provides information about the number of votes a specific movie or show has received on the IMDb database. This is a discrete variable. Here are the summary statistics for votes:

```
> summary(netflix_data$imdb_votes)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
    5     521    2279   23407   10144 2268288     539
```

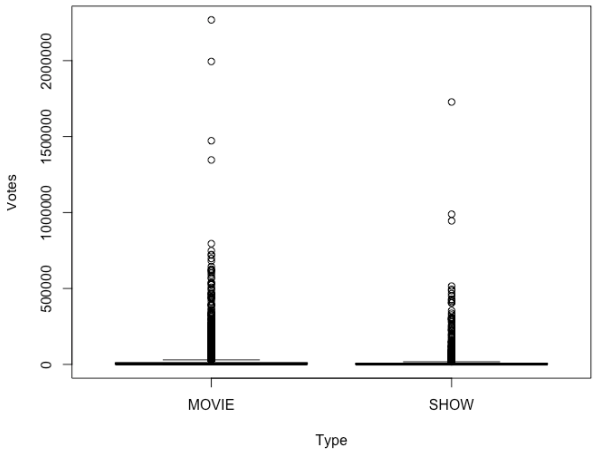
As we can see from these statistics, the median number of votes is 2279, and the range is very large as it ranges from 5 to 2268288.

In order to explore the votes variable and see if there is a relationship with the score, I plotted a scatterplot:



As we can see above, most of the entries have below 500000 votes.

I also wanted to see if the number of votes varied based based on the type of each entry (whether, on average, movies received higher number of votes than shows):



Although identifying the quartiles and median is not clear from the graph above, it is evident that they follow similar numbers, below 500000, as can be matched with the scatterplot above.

3 Part 2 - Research Questions

3.1 Question 1

Question: Is there a significant difference in IMDb scores between movies and TV shows?

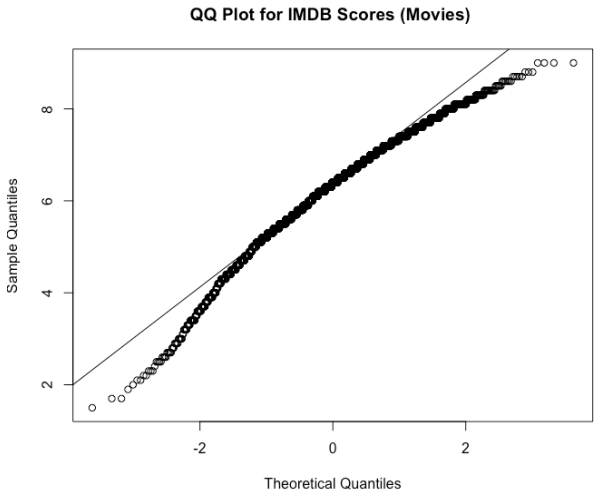
This question is of interest for this data since analyzing the difference in scores between the two types of entries provides an understanding of individual's preferences, whether they like movies more than TV shows or vice versa. Additionally, understanding this difference can also help people managing Netflix see what type is preferred more so they can add more entries of that type. For instance, if the scores for movies are significantly higher, then Netflix could include more movies than shows, since users tend to rate the movies higher, so they would watch more movies.

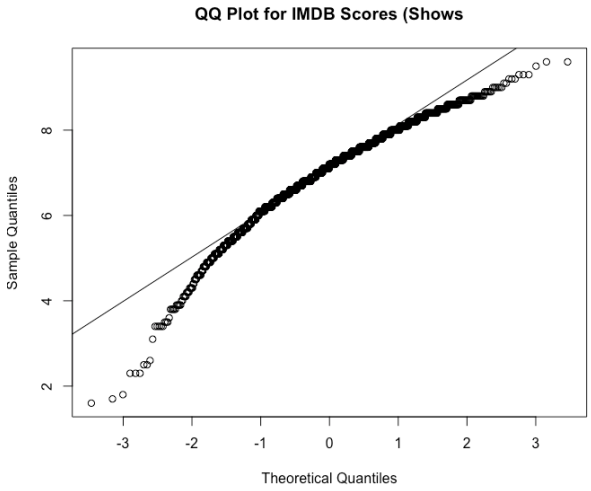
I will use an independent two-sample t-test in order to answer this question as I have two independent groups (movies and TV shows), and I can compare the means of IMDb scores for each.

The assumptions made in applying this method is that both groups of data (movies and TV shows) are normally distributed with similar variances, and that the two groups are independent. In order to check the validity of the normality assumption, I can examine the QQ plots for the two groups. In order to do so, I created vectors that have the imdb score for each type of data:

```
imdb_movies = netflix_data$imdb_score[netflix_data$type == "MOVIE"]
imdb_shows = netflix_data$imdb_score[netflix_data$type == "SHOW"]
```

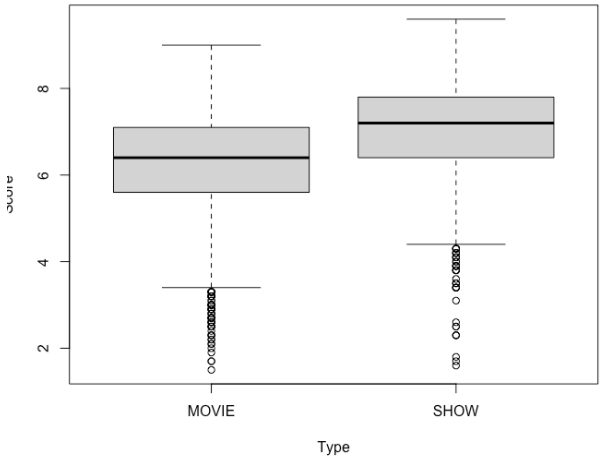
Then, I created QQ plots:





As it is clear from the graphs above, both the QQplots follow the pattern that they are similar to a straight line, which supports the fact that both groups are normally distributed.

Before I perform the test, I want to consider a boxplot that models the difference between IMDb score between the two different types of entries:



Null Hypothesis: There is not a difference in average IMDb scores between movies and TV shows.

Alternative Hypothesis: There is a significant difference in average IMDb scores between movies and TV shows.

Using the vectors above, I ran a two-sample t-test:

```
        Welch Two Sample t-test

data:  imdb_movies and imdb_shows
t = -23.875, df = 3976.8, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.8120173 -0.6887780
sample estimates:
mean of x mean of y
 6.266980  7.017377
```

This test-results in a p-value that is very small: 2.2e-16. Considering that the standard significance level is with an alpha of 0.05, it is clear that since the p-value is lower than 0.05, we can reject the null hypothesis. As a result, there is sufficient evidence that there is a significant difference between the average IMDb scores between movies and TV shows. From above, the sample estimates of mean score of movies is 6.266980 and the mean score of shows is 7.017377. It can also be seen that the 95% confidence interval for the difference in average scores lies between -0.8120173 and -0.6887780, which reveals that we are 95% confident that the true difference in average IMDb scores between the two groups is between -0.8120173 and -0.6887780. The degrees of freedom is 3976.8 and the t-statistic is -23.875.

Since the test shows that there is sufficient evidence that there is a significant difference between the average IMDb scores between movies and TV shows, it is clear that, on average, these two different groups are voted differently by users on the IMDb database. Since TV shows have a higher average score than movies, it is evident that individuals like TV shows more than movies, as they rated higher. As a result, Netflix should consider supporting TV shows more as the average IMDb score for shows is higher than that of movies.

4 Question 2

Question: Do Runtime, Release Year, and the number of IMDb votes contribute to IMDb score?

This question is of interest for this data since analyzing how these factors contribute to the IMDb score provides movie-makers and show-makers what to prioritize and how long each movie/episode should be to receive a higher score. For instance, if results show that individuals prefer shorter movies over longer movies, then it would make sense that movie-makers prioritize shorter movies. Additionally, it provides information about how the quantity of people voting is related to the score.

I will use a multiple regression model in order to answer this question as I have a dependent variable (IMDb score) and multiple independent variables (Runtime, Release Year, IMDb Votes), and I can find the relationship between the dependent and independent variables.

The assumptions made in applying this method is that the Runtime, Release Year, and IMDb votes are independent, the residuals are normal distributed, and the fact that the independent variables are not strongly correlated with eah other. In order to check if the residuals are normally distributed, I can plot the QQplot of the residuals obtained from the multi-regression model.

Null Hypothesis: There is no relationship between the independent variables (Runtime, Release Year, IMDb votes) and IMDb score.

Alternative Hypothesis: There is relationship between at least one independent variable (Runtime, Release Year, or IMDb votes) and IMDb score.

Here is how I created the model:

```
reg_model = lm(netflix_data$imdb_score ~
                 netflix_data$runtime + netflix_data$release_year +
                 netflix_data$imdb_votes)
summary(reg_model)
```

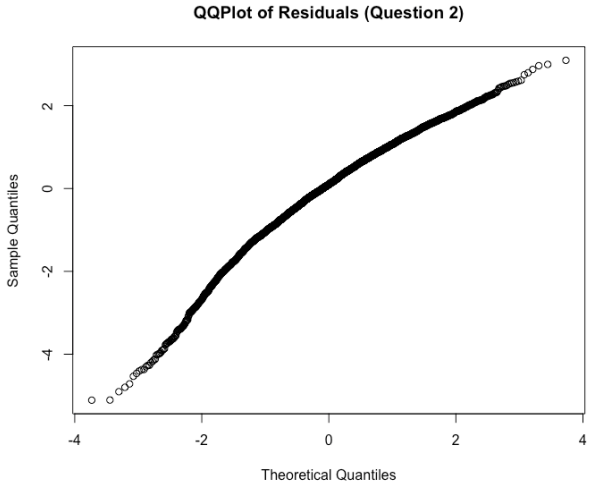
Below is the result:

```
Call:
lm(formula = netflix_data$imdb_score ~ netflix_data$runtime +
    netflix_data$release_year + netflix_data$imdb_votes)

Residuals:
    Min      1Q  Median      3Q     Max
-5.1096 -0.6528  0.0991  0.7858  3.0940

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                 4.055e+01  4.367e+00   9.285  < 2e-16 ***
netflix_data$runtime       -6.182e-03  4.049e-04 -15.265  < 2e-16 ***
netflix_data$release_year -1.666e-02  2.163e-03  -7.703 1.58e-14 ***
netflix_data$imdb_votes    2.637e-06  1.806e-07  14.604  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.113 on 5263 degrees of freedom
  (539 observations deleted due to missingness)
Multiple R-squared:  0.08123,   Adjusted R-squared:  0.08071
F-statistic: 155.1 on 3 and 5263 DF,  p-value: < 2.2e-16
```

From the results above, it is clear that the independent variables Runtime, Release Year, and IMDb votes contribute to the IMDb score because the p-values are very small, less than the standard significance level with an alpha of 0.05. Additionally, since the p-value is less than 0.05, we can reject the Null Hypothesis. We have sufficient evidence that that at least one of the independent variable has a relationship with IMDb score. In fact, since the p-values for all Runtime, Release Year, and IMDb votes is smaller than 0.05, which suggests that each independent variable has a relationship with IMDb score. In terms of the coefficients, we seen that Runtime has a estimate of -6.182e-03, which means that for every minute of runtime, the IMDb score decreases by approximately 0.006 points, which is a very slight negative relationship between Runtime and IMDb score, signifiying that lower runtimes have slightly higher ratings (which matches because the previous question revealed that shows have higher average rating than movies). For Release Year, the coefficient estimate is -1.666e-02, which means that for every next year, the IMDb score decreaes by approximately 0.017 point, which is a slight negative negative correlation, signifying that older entries seem to have slightly higher ratings. In terms of

IMDb votes, the coefficient estimate is 2.637e-06, which means that for every additional vote, the IMDb score increaes by 0.00000264 points, which is a very, very, very slight positive relationship, signifying that more votes could lead to a very, very, very slightly higher rating. Since the Multiple R-Squared produced is 0.08123, it means that approximately 8.123% of the variability in IMDb scores can be explained by Runtime, Release Year, and IMDb votes. Although the R-Squared produced is low, this still makes sense because the IMDb scores are also dependent on other preferences that individuals may have, cast, budget, current cultural trends, etc. The adjusted R-Squared is very similar to the Multiple R-Squared. The equation for my Multiple Regression Model is:
IMDb Score = 40.55 - 0.006 * Runtime - 0.017 * Release Year + 0.00000264 * IMDb Votes.

For checking whether the residuals obtained from the model are normally distributed, the QQPlot of the residuals can be plotted:



As it is clear from the graph above, the plot follows similar pattern to that of a straight line, indicating that the residuals are mostly normal.

From the results, it is clear that there is a very slight negative relationship between Runtime and IMDb Score, which suggests that shorter entries have very slight higher ratings, while longer entries have very slight lower ratings. The reason for this could be due to the fact that entries with shorter Runtime are shows (episodes), and from the previous question, it is clear that average rating for shows was higher than average rating for movies, and shows have lower Runtime than movies. Another possible explanation for this relationship is the fact that humans tend to get bored or distracted, or might even have other tasks to work on, which would cause entries with longer Runtimes to have lower ratings. The relationship between Release Year and IMDb Score is also slightly negative, which suggests that older entries (classics) are rated higher

than newer entries. The relationship between IMDb Votes and IMDb Score is a very, very, very slight positive relationship, which suggests that if there are more voters, then the IMDb score would be higher. The reason for this could be that if there are more voters, then the entry (movie or show) would be more popular, which would cause it to have a higher rating.

5 Question 3

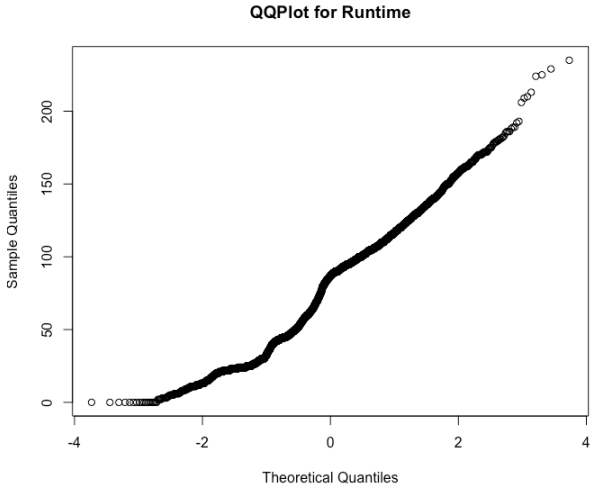
Question: Is there a correlation between Runtime and IMDb Score?

This question aims to check whether Runtime (the length of a movie or show in minutes) is correlated (has a linear relationship) with IMDb Score. This question is of interest for this data since understanding the relationship between Runtime and IMDb Score can provide valuable information about individuals' preferences, which platforms like Netflix can use in order to improve the content they display. Additionally, as mentioned in the previous question, the correlation between Runtime and IMDb Score is also useful because it provides movie-makers or show-makers whether to produce long or short entries.

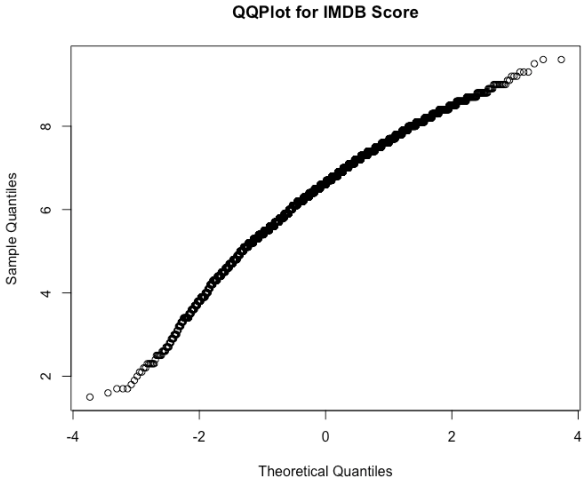
I will use the Pearson Correlation in order to answer this question as I have continuous variables (the IMDb score is a rounded continuous and Runtime is continuous as well), and I aim to test if there is a linear relationship between them. The main assumption I will be making are that both variables are normally distributed, which I can find using the QQPlots.

It is important to note that with the current dataset, I would not be able to find the Pearson Correlation Coefficient since some entries have an IMDb score of NA. As a result, I created a new dataframe that excludes the entries with NA values:

In order to check for the normality assumption for both variables, here are the QQPlots:



Here is the QQPlot for IMDb Score:



As we can see from the plots above, it is clear that both plots follow a pattern that is similar to a straight line. As a result, the IMDb score data and the Runtime data is normally distributed.

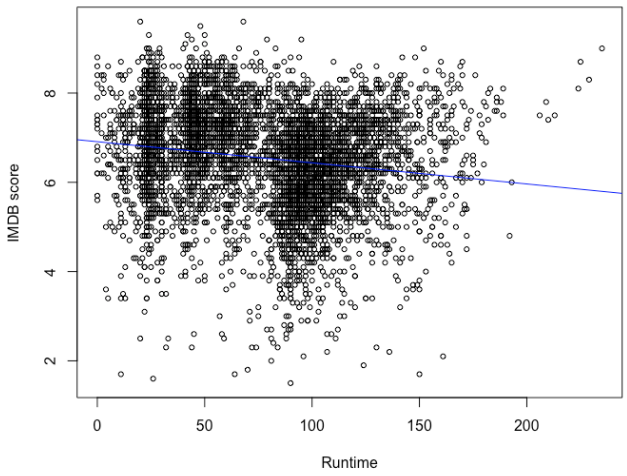
Null Hypothesis: There is no linear relaitonship between Runtime and IMDb Score.

Alternative Hypothesis: There is a linear relationship between Runtime and IMDb Score.

In order to find the Pearson Correlation Coefficient, I can cor() function and specify the method as **pearson**:

```
cor3 = cor(new_netflix$runtime, new_netflix$imdb_score, method = "pearson")
```

This outputs in a correlation of -0.1592965. The negative value infront of the result indicates that there is an inverse relationship between the two variables. For instance, if Runtime increases, then IMDb Score decreases. Similarly, if Runtime decreases, IMDb Score increases. This means that individuals tend to prefer movies or shows with lower Runtime. Additionally, since the result is closer to 0 than it is to 1, it indicates that there is a weak linear relationship between the two variables. In order to visualize these results, I plotted the relationship between the two variables and added a linear regression line.



As we can see from the graph above, it matches with the resulting correlation because the linear regression line has a very flat slope and is in the negative direction.

In order to find the probability of observing the relationship above and obtain statistical significance of this relationship above, I calculated the t-value and p-value based on the correlation coefficient.

```
total = nrow(new_netflix)
t_value = cor3 * sqrt((n - 2) / (1 - cor3^2))
p_value = 2 * pt(-abs(t_value), df = n - 2)
p_value
```

This resulted in a p-value of 0.8407035. Considering the standard significance level with an alpha of 0.05, it is clear that the p-value is higher than 0.05. As a result, we fail to reject the Null Hypothesis. As a result, we do not have sufficient evidence that there is a linear relationship between Runtime and IMDb Score. This indicates that there is approximately an 84% probability that the observed relationship between the two variable was by chance.