CS-767 Assignment 2

Benzon Carlitos Salazar

2024-02-18

Read Data

```
movie_data_df <-
  readr::read_csv(here::here("data", "movie_data.csv"), show_col_types = FALSE)
summary(movie_data_df)
```

```
##       ID              Title               Year          Rating
##  Length:1000        Length:1000        Min.   :1902   Min.   :5.700
##  Class :character   Class :character   1st Qu.:1953   1st Qu.:7.500
##  Mode  :character   Mode  :character   Median :1972   Median :7.800
##                                        Mean   :1975   Mean   :7.805
##                                        3rd Qu.:2001   3rd Qu.:8.100
##                                        Max.   :2019   Max.   :9.300
##       Rank
##  Min.   :   1.0
##  1st Qu.: 250.8
##  Median : 500.5
##  Mean   : 500.5
##  3rd Qu.: 750.2
##  Max.   :1000.0
```

```
rt_movie_data_df <-
  readr::read_csv(here::here("data", "rt_movie_data.csv"), show_col_types = FALSE)
summary(rt_movie_data_df)
```

```
##       ID              Title               Year          Rating
##  Length:1000        Length:1000        Min.   :1902   Min.   : 71.00
##  Class :character   Class :character   1st Qu.:1956   1st Qu.: 93.00
##  Mode  :character   Mode  :character   Median :1974   Median : 95.00
##                                        Mean   :1974   Mean   : 94.86
##                                        3rd Qu.:1994   3rd Qu.: 98.00
##                                        Max.   :2015   Max.   :100.00
##       Rank
##  Min.   :   1.0
##  1st Qu.: 250.8
##  Median : 500.5
##  Mean   : 500.5
##  3rd Qu.: 750.2
##  Max.   :1000.0
```

Schema definition

In both tables, each row represents a movie, with columns representing different attributes such as ID, title, year, rating, and rank.

movie_data.csv

```
movie_data_df %>%
  dplyr::slice_head(n = 10) %>%
  knitr::kable()
```

ID	Title	Year	Rating	Rank
42b83ba8-0168-450c-b224-9115721353cb	The Godfather	1972	9.2	1
40a13538-ff98-4596-8435-b617beadb9d0	Tokyo Story	1953	8.1	2
99619f03-0b4d-43f8-a136-d5a5fd0df7ce	Seven Samurai	1954	8.6	3
52f24199-d46f-49f0-bd3b-ebb4c430e533	The Godfather Part II	1974	9.0	4
0b63ed36-0714-42fe-9b50-f91ad663561a	Casablanca	1942	8.5	5
726f93fd-a197-40c9-b098-f42fe1cf4237	Citizen Kane	1941	8.3	6
d9be7aa8-48d7-4a33-83c0-ae6eeacd3007	Lawrence of Arabia	1962	8.3	7
dda3934d-278b-4790-a378-e8bc2ca940eb	12 Angry Men	1957	9.0	8
2b4b5888-f166-41b7-bdf4-07d87b2c10ac	Sunset Blvd.	1950	8.4	9
967cb3b5-3160-43cf-8585-a3d2010f0422	Shoah	1985	8.7	10

- ID: UUID (Universally Unique Identifier)
- Title: String (Movie title)
- Year: Integer (Year of release)
- Rating: Float (Movie rating)
- Rank: Integer (Ranking of the movie)

rt_movie_data.csv

```
rt_movie_data_df %>%
  dplyr::slice_head(n = 10) %>%
  knitr::kable()
```

ID	Title	Year	Rating	Rank
8ff1c1d4-bd6f-4647-aade-330375db69b6	A Trip to the Moon	1902	100	1
40bce645-7fb8-4584-be9c-e5d894759b1f	Intolerance	1916	97	2
5c4f0a57-84d4-4c7d-9378-11e5be1703b5	Broken Blossoms	1919	95	3
f5b9d713-001a-4b3b-a08e-7a92b410911c	The Cabinet of Dr. Caligari	1920	96	4
db3bab21-c852-4059-9823-55d6e3d951fa	The Kid	1921	100	5
2c1efa6d-3c5e-4d4b-be9e-9dca6c631182	The Phantom Carriage	1921	100	6
732469e1-1f3d-4064-b5b0-d3c6b12520e1	Nanook of the North	1922	100	7
bdabd52e-0bdc-4196-9d2b-9470039a15d9	Nosferatu	1922	97	8
45cf02e5-3458-45e8-a0ea-16a7753a56da	Our Hospitality	1923	97	9
e09519b1-cf26-4b67-ad1b-5353b0631e47	Safety Last!	1923	97	10

- ID: UUID (Universally Unique Identifier)
- Title: String (Movie title)
- Year: Integer (Year of release)
- Rating: Integer (Movie rating)
- Rank: Integer (Ranking of the movie)

Missing Values

`movie_data.csv` schema

```
missin_values_movie_data <-
  movie_data_df %>%
  dplyr::summarise_all(list(~ sum(is.na(.))))

missin_values_movie_data %>%
  knitr::kable()
```

ID	Title	Year	Rating	Rank
0	0	0	0	0

`rt_movie_data.csv` schema

```
missin_values_rt_movie_data <-
  rt_movie_data_df %>%
  dplyr::summarise_all(list(~ sum(is.na(.))))

missin_values_rt_movie_data %>%
  knitr::kable()
```

ID	Title	Year	Rating	Rank
0	0	0	0	0

There are no missing data in both tables. Each column in the tables contains values for all rows, with no instances where a value is missing (i.e., no NA or NULL values). This indicates that the data is complete and there are no missing values to be addressed in the analysis.

If there happens to be missing values then,

- for the `ID` column, theoretically, this column does not allow any NULL values because this is an identifier. If these were missing, this tells us that there are no corresponding movies, in which case we would just remove these rows.
- for the `Title` column, because we have no way of determining what movie the attributes would correspond to, we can just use a default value of `Unknown Movie` or something similar.
- for the `Year` column, since these are numeric values, we can use a default `1601` value to indicate any missing values. In this case, since we know that no good movies existed in year 1601, it is a good indicator of missing values.

- for the `Rating` column, we may use the average for that column to fill in missing values. Since the table is ordered by ranking, we may get away with finding a somewhat correct rating based on the rows position.
- for the `Rank` column, since the rows are already ordered, we can simply use the rows index to fill in any missing values in the `Rank` column.

Column classification

- ID: UUID (Universally Unique Identifier)
- Title: String (Movie title)
- Year: Integer (Year of release)
- Rating: Integer (Movie rating)
- Rank: Integer (Ranking of the movie)

Character Column Attribution Report for `Title` column in `movie_data.csv`

```
# Helper function to calculate average, min, and max length of textual values
textual_stats <- function(x) {
  if(is.character(x)) {
    avg_length <- mean(nchar(x), na.rm = TRUE)
    min_length <- min(nchar(x), na.rm = TRUE)
    max_length <- max(nchar(x), na.rm = TRUE)
    return(data.frame(avg_length = avg_length, min_length = min_length, max_length = max_length))
  } else {
    return(NULL)
  }
}
```

```
movie_data_summary <-
  movie_data_df %>%
  dplyr::summarise(dplyr::across(Title, textual_stats))
```

- Average length of characters in `Title` column: 15.369 characters.
- Minimum length of characters in `Title` column: 1 character.
- Maximum length of characters in `Title` column: 68 characters.

Character Column Attribution Report for `Title` column in `rt_movie_data.csv‘`

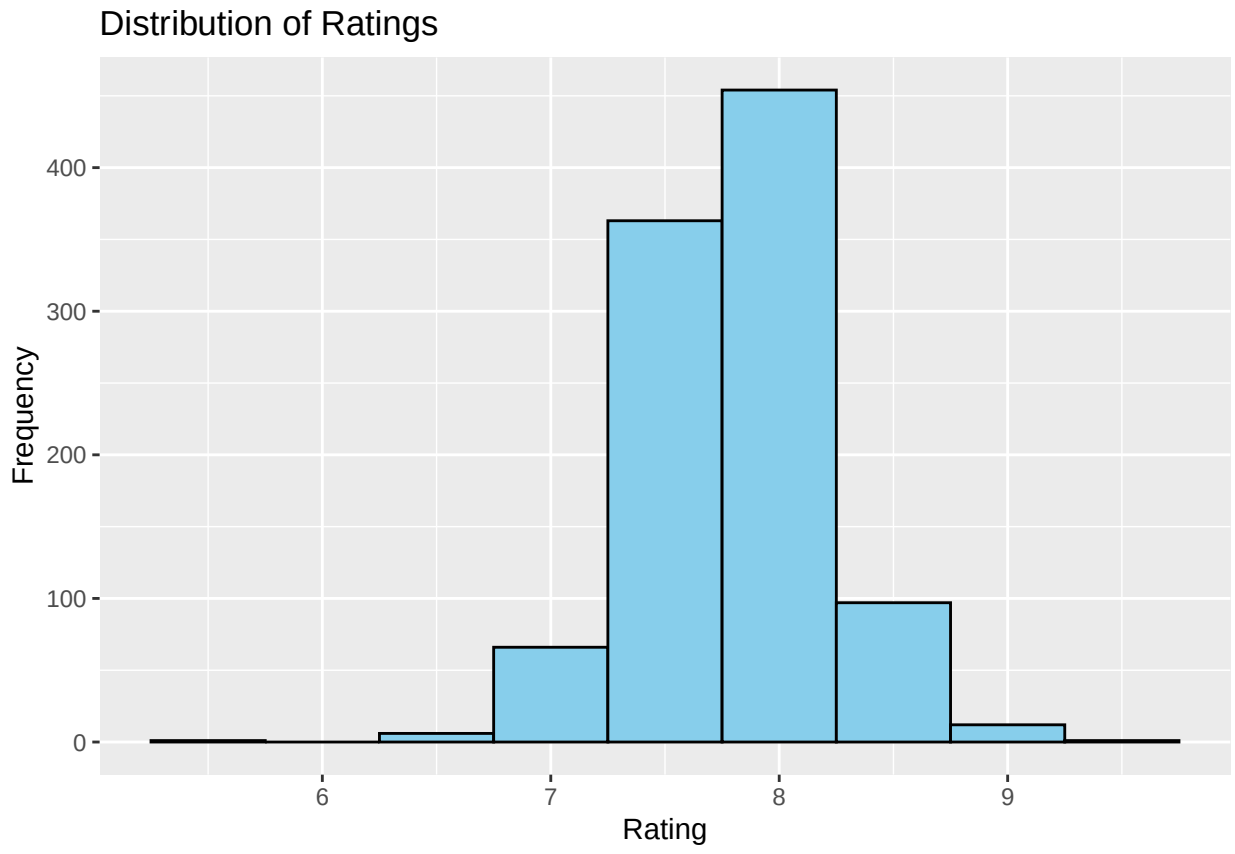
```
rt_movie_data_summary <-
  rt_movie_data_df %>%
  dplyr::summarise(dplyr::across(Title, textual_stats))
```

- Average length of characters in `Title` column: 15.752 characters.
- Minimum length of characters in `Title` column: 1 character.
- Maximum length of characters in `Title` column: 83 characters.

Outliers

Histogram for the Rating column in `movie_data.csv`

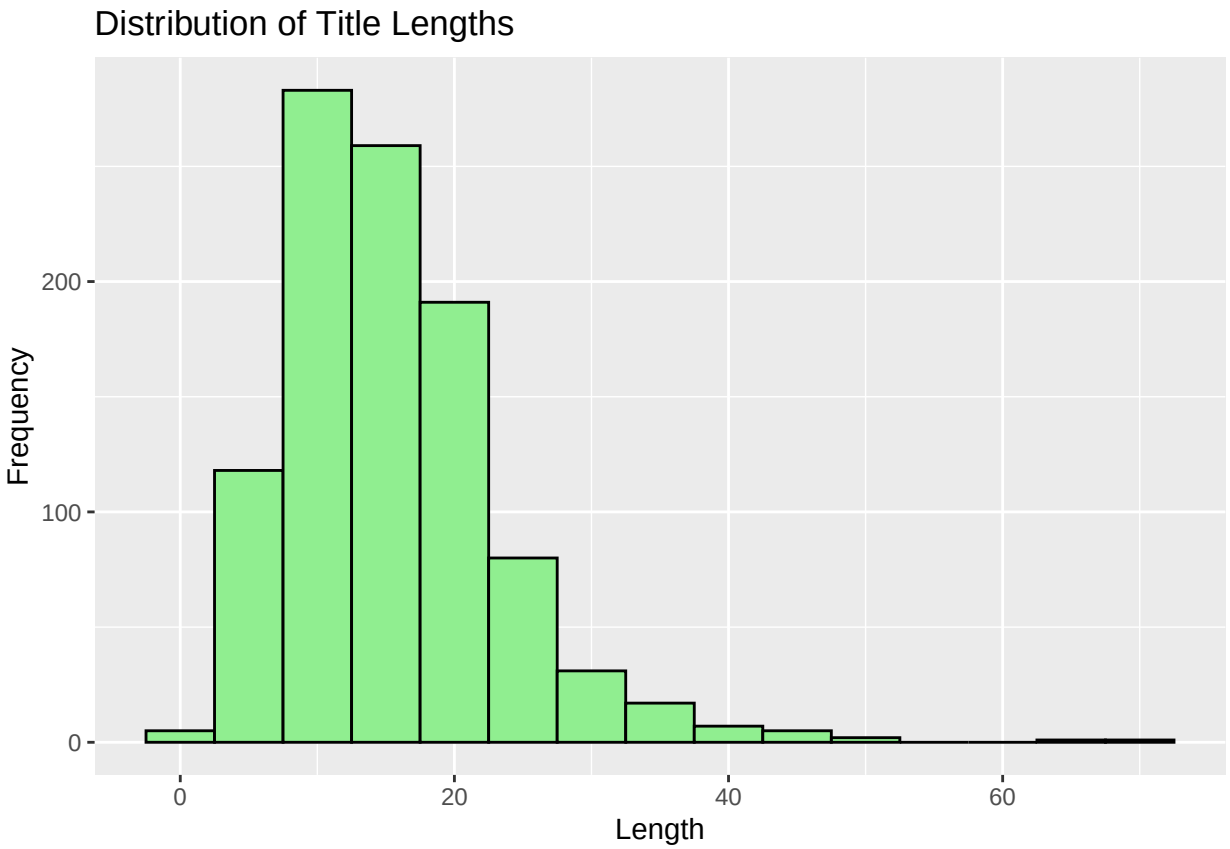
```
ggplot2::ggplot(movie_data_df, ggplot2::aes(x = Rating)) +
  ggplot2::geom_histogram(binwidth = 0.5, fill = "skyblue", color = "black") +
  ggplot2::labs(title = "Distribution of Ratings", x = "Rating", y = "Frequency")
```



Histogram for the length of the Title column in `movie_data.csv`

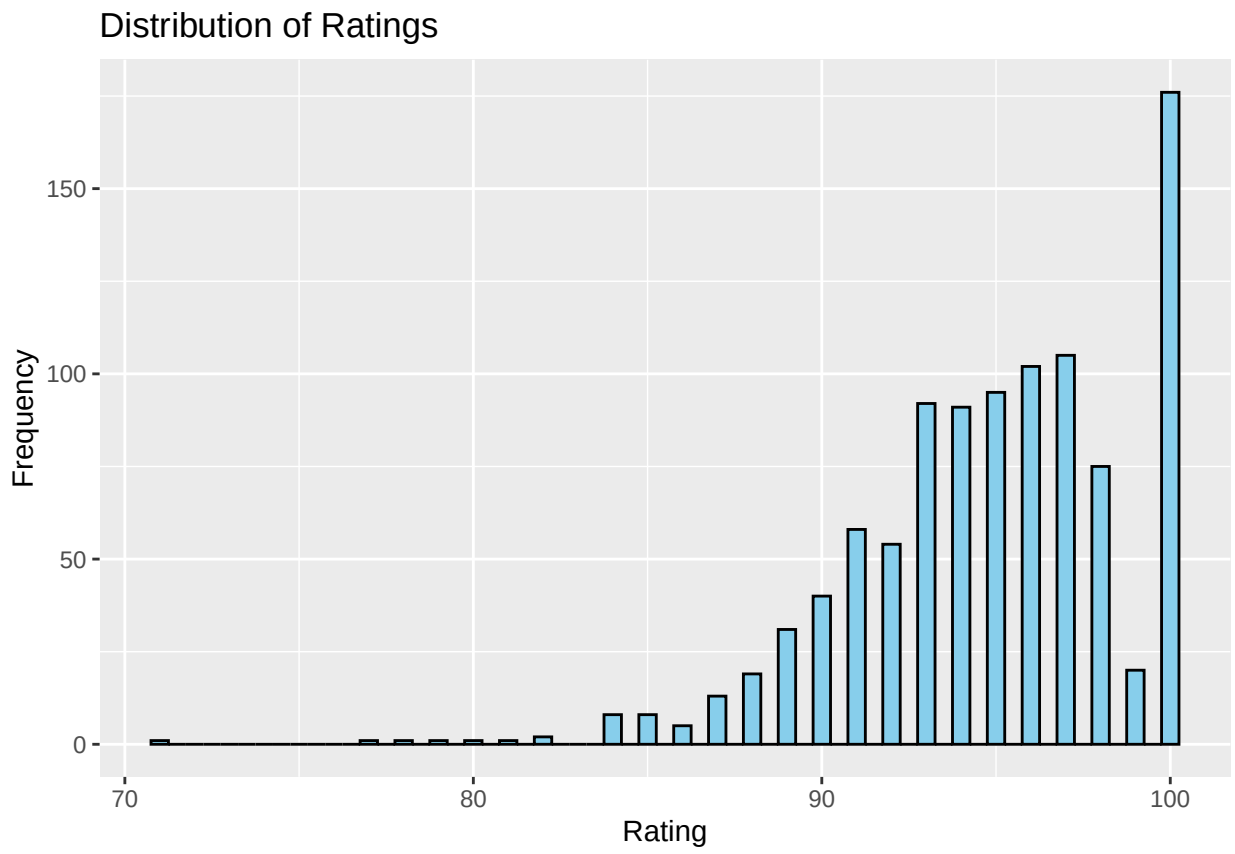
```
movie_data_df$Title_length <- nchar(movie_data_df$Title)

ggplot2::ggplot(movie_data_df, ggplot2::aes(x = Title_length)) +
  ggplot2::geom_histogram(binwidth = 5, fill = "lightgreen", color = "black") +
  ggplot2::labs(title = "Distribution of Title Lengths", x = "Length", y = "Frequency")
```



Histogram for the Rating column in `rt_movie_data.csv`

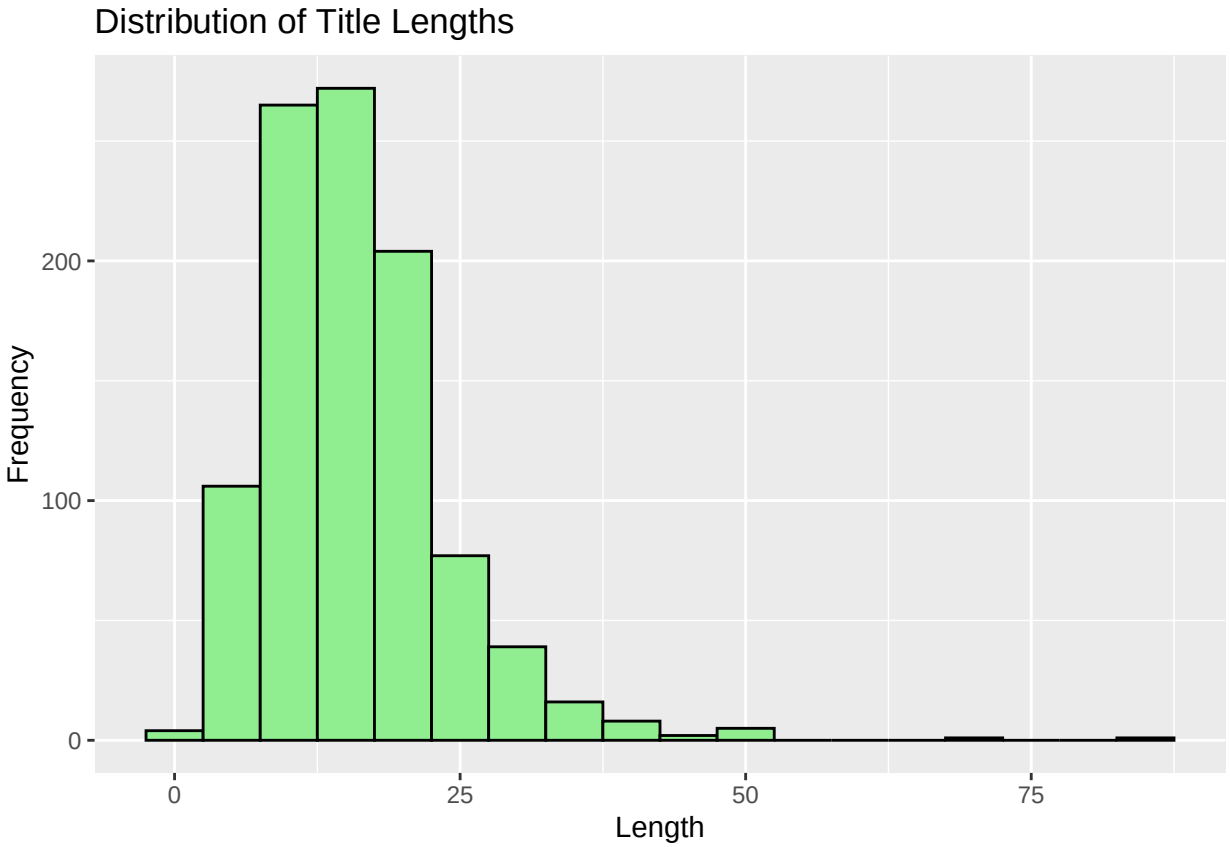
```
ggplot2::ggplot(rt_movie_data_df, ggplot2::aes(x = Rating)) +
  ggplot2::geom_histogram(binwidth = 0.5, fill = "skyblue", color = "black") +
  ggplot2::labs(title = "Distribution of Ratings", x = "Rating", y = "Frequency")
```



Histogram for the length of the Title column in `movie_data.csv`

```
rt_movie_data_df$Title_length <- nchar(rt_movie_data_df$Title)

ggplot2::ggplot(rt_movie_data_df, ggplot2::aes(x = Title_length)) +
  ggplot2::geom_histogram(binwidth = 5, fill = "lightgreen", color = "black") +
  ggplot2::labs(title = "Distribution of Title Lengths", x = "Length", y = "Frequency")
```

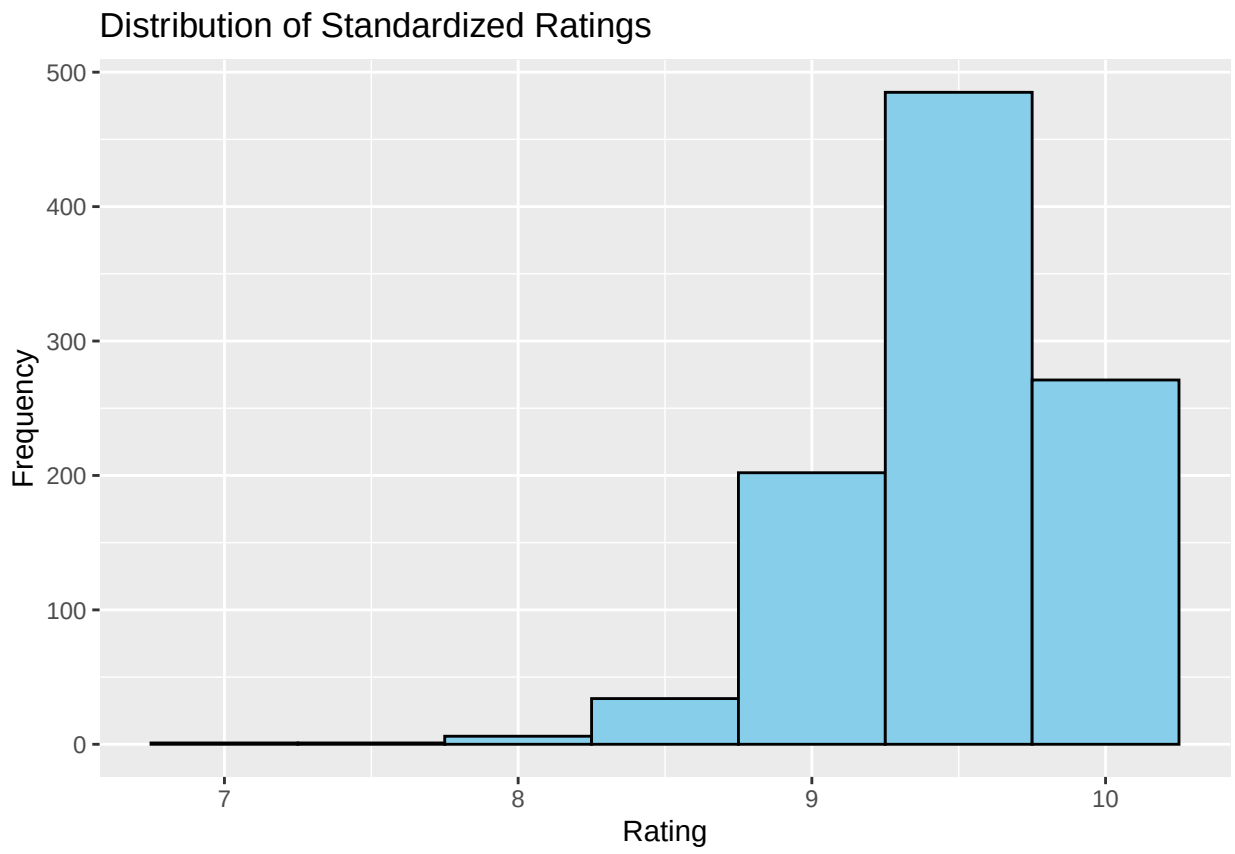


We don't really see much outliers, except for the length of characters in titles. The ratings seem to be correctly formatted and are within the range. The only difference is the fact that ratings in `movie_data.csv` are in the range of 1 - 10, while the ratings in the `rt_movie_data.csv` are in the range of 0 - 100.

We do see ratings around 70 - 75 in the `rt_movie_data.csv`, and quite a few of 100s in the `rt_movie_data.csv`. What we can do to match the range of ratings for both dataframes is to divide each rating in the `rt_movie_data.csv` by 10 to standardize it to be between 0 - 10. Then, we can re-run the histogram and see if it worked.

```
rt_movie_data_df <-
  rt_movie_data_df %>%
  dplyr::mutate(Rating = Rating / 10)

ggplot2::ggplot(rt_movie_data_df, ggplot2::aes(x = Rating)) +
  ggplot2::geom_histogram(binwidth = 0.5, fill = "skyblue", color = "black") +
  ggplot2::labs(title = "Distribution of Standardized Ratings", x = "Rating", y = "Frequency")
```



After standardization, we now see that the majority of ratings are in the 9.5 range.

Formatting

All the columns do follow the same formatting. For example, all of the `Year` values are properly formatted as `YYY`. The `Rank` values are all integers, and so are the `Rating` values.

Additional Data Quality Problems

Upon initial review, both datasets appear to be free of inherent data quality issues. The only discrepancy noted is the difference in ratings between the two tables, indicating a potential inconsistency in the movies listed. This suggests that one table may include movies not present in the other, highlighting a need for further investigation to ensure data completeness and consistency across the datasets.

Packages used

The following R packages are used in this report:

```
## Finding R package dependencies ... Done!

## $.
## [1] "rmarkdown" "magrittr"  "here"      "readr"     "dplyr"     "knitr"
## [7] "ggplot2"   "renv"
```