Evaluation Warning: The document was created with Spire.Doc for Python.

Movie Data Analysis Project

This project involves analyzing movie ratings, genres, and runtime data using R. Below is a summary of the steps involved in the project.

**Contents**   
1. 1. Installing Necessary Libraries   
2. 2. Loading the Necessary Libraries   
3. 3. Reading the Data   
4. 4. Joining Datasets   
5. 5. Exploring the Data   
6. 6. Data Cleaning   
7. 7. Frequency Distribution Analysis   
8. 8. Movie Ratings Analysis   
9. 9. Bayesian Rating Analysis   
10.10. Genre and Average Rating Correlation

**1. Installing Necessary Libraries**   
```r   
install.packages('readr')   
install.packages('dplyr')   
install.packages('ggplot2')   
install.packages('tidyr')   
install.packages('plotly')   
```

**2. Loading the Necessary Libraries**   
```r   
library(readr)   
library(dplyr)   
library(ggplot2)   
library(tidyr)   
library(plotly)   
```

**3. Reading the Data**   
```r   
ratings <- read\_tsv('raw\_data/title.ratings.tsv')

basics <- read\_tsv('raw\_data/title.basics.tsv')   
```

**4. Joining Datasets**   
```r   
joined\_data <- full\_join(basics, ratings)   
View(joined\_data)   
```

**5. Exploring the Data**   
```r   
test <- joined\_data   
```

**6. Data Cleaning**

**6.1 Counting Missing Values**   
```r   
sum(is.na(test))   
na\_counts <- lapply(test, function(test) sum(is.na(test))) ```

**Evaluation Warning: The document was created with Spire.Doc for Python.**

```r   
boxplot(test, col = 'lightblue', main = 'Distribution by Column', xlab = 'Variables', ylab = 'Values')   
```

**6.3 Removing NA Values**   
```r   
test <- na.omit(test)   
```

**6.4 Removing Duplicates**   
```r   
total\_duplicates <- sum(duplicated(test))   
duplicates <- test %>%  
 group\_by(across(everything())) %>%  
 filter(n() > 1) %>%  
 ungroup()   
test <- test[!duplicated(test),]   
sum(duplicated(test))   
```

**6.5 Cleaning Columns**   
```r   
cleaned\_v2 <- cleaned\_v1 %>%  
 select(-tconst, -originalTitle, -isAdult, -endYear)   
View(cleaned\_v2)   
```

**7. Frequency Distribution Analysis**

**7.1 Frequency Table of `titleType`**   
```r   
View(fdtQl(cleaned\_v2$titleType))   
```

**7.2 Selecting Movies Only**   
```r   
only\_movies = filter(cleaned\_v2, titleType == 'movie' | titleType == 'short' | titleType == 'tvMovie')   
```

**8. Movie Ratings Analysis**

**8.1 Top 100 Movies**   
**Evaluation Warning: The document was created with Spire.Doc for Python.** ```r   
 sorted\_by\_num\_votes <- only\_movies %>% arrange(desc(numVotes))   
 top\_100\_best\_rated\_movies <- sorted\_by\_num\_votes[1:100, ]   
 top\_100\_best\_rated\_movies\_according\_to\_rating <-   
 top\_100\_best\_rated\_movies %>% arrange(desc(averageRating))   
 ```

**8.2 Average Rating by Genre**   
```r   
ggplot(average\_rating\_of\_movies, aes(x=genres, y=avg\_rating, fill=avg\_rating)) + geom\_bar(stat = 'identity') +  
 scale\_fill\_gradient(low = 'blue', high = 'red')   
```

**8.3 Genre Analysis**   
```r   
ggplot(genre\_analysis, aes(x = genres, y = avg\_rating, fill = genres)) + geom\_bar(stat = 'identity', show.legend = FALSE)   
```

**8.4 Average Rating by Genre Over the Years**   
```r   
plot <- average\_ratings\_by\_genre\_year %>%

ggplot(aes(startYear, avg\_rating, color = genres)) + geom\_line() +  
 facet\_wrap(~genres) +  
 theme(legend.position = 'none')   
ggplotly(plot) %>% layout(hovermode = 'x unified') ```

**8.5 Correlation of Average Rating and Length Over the Years**   
```r   
ggplot(genre\_data\_scaled, aes(x = startYear, y = ScaledValue, color = Factor, linetype = Factor)) +  
 geom\_line(aes(group = Factor, genres), size = 1.2) +  
 facet\_wrap(~genres)   
```

**8.6 Number of Movies Released Per Year**   
```r   
ggplot(number\_of\_movies\_per\_year, aes(x=startYear, y=count)) + geom\_line() +  
 facet\_wrap(~)   
```

**Evaluation Warning: The document was created with Spire.Doc for Python.**

```r   
ggplot(number\_of\_movies\_each\_year, aes(x=startYear, y=count)) + geom\_line(aes(group = genres, color = genres), size=0.5) + facet\_wrap(~genres, scales='free\_y')   
```

**9. Bayesian Rating Analysis**   
```r   
C <- mean(horror\_movies$rating, na.rm = TRUE)   
m <- 100   
horror\_movies <- horror\_movies %>%  
 mutate(adjusted\_rating = (C \* m + horror\_movies$averageRating \* horror\_movies$numVotes) / (horror\_movies$numVotes + m))   
```

**10. Genre and Average Rating Correlation**   
```r   
grouped\_by\_genre$runtimeMinutes <-   
as.numeric(as.character(grouped\_by\_genre$runtimeMinutes)) ```