Name: Shreya Bhattacharjee
3108
Foundation of data science

Netflix Data Analysis

Description:

The Netflix Originals dataset contains information about original TV shows and movies produced and released by Netflix. The Dataset is having 584 rows and 11 columns. Each observation in the dataset represents a unique Netflix Original and includes various attributes that provide insights into the content and production details.

Predicting IMDb Scores: Build a predictive model using variables such as duration, release year, and genre to predict the IMDb scores of Netflix Originals. This could help identify factors that contribute to higher ratings and guide content creation decisions.

1. **"Title"**: This column likely contains the titles of movies or TV shows. It would include the names of Netflix Originals in your dataset.
2. **"Genre"**: This column typically contains the genre or category to which each title belongs. For example, a movie could belong to genres like "Action," "Comedy," "Drama," "Science Fiction," etc.
3. **"Premiere"**: This column likely contains information about the premiere date of the movie or TV show. It tells you when the title was first made available on Netflix.
4. **"Runtime"**: This column probably contains the duration or length of each title. It indicates how long a movie or TV show is in terms of minutes or hours.
5. **"IMDB Score"**: This column likely contains the IMDb (Internet Movie Database) score or rating for each title. IMDb scores are often used as a measure of a title's popularity or quality.
6. **"Language"**: This column may contain information about the language in which the title is available. For Netflix Originals, this could include various languages.

Name: Shreya Bhattacharjee
3108
Foundation of data science

7. **"Released"**: This column might contain additional information about the release of the title, possibly in a different format or with more details compared to "Premiere."
8. **"Year"**: This column could be derived from the "Premiere" date and represent the year in which each title premiered on Netflix.
9. **"Month"**: This column might be derived from the "Premiere" date and represent the month in which each title premiered.

10. **"Date"**: This column could be derived from the "Premiere" date and represent the specific date on which each title premiered.

11. **"Day"**: This column may represent the day of the week when each title premiered, derived from the "Premiere" date.

Code and Output

```
install.packages("tidyverse")
install.packages("ggplot2")
library(dplyr)
library(caret)
library(ggplot2)
```

Conclusion: Installing and Loading the library.

Reading the CSV file

```
library(readr)
NetflixOriginals <- read_csv("NetflixOriginals.csv")
View(NetflixOriginals)
```

```
head(NetflixOriginals)
```

Title	Genre	Premiere	Runtime	`IMDB Score`	Language	Released	Year
<chr>	<chr>	<chr>	<dbl>	<dbl>	<chr>	<date>	<dbl>
Enter th…	Docu…	August …	58	2.5	English…	2019-08-05	2019
Dark For…	Thri…	August …	81	2.6	Spanish	2020-08-21	2020
The App	Scie…	Decembe…	79	2.6	Italian	2019-12-26	2019
The Open…	Horr…	January…	94	3.2	English	2018-01-19	2018
Kaali Kh…	Myst…	October…	90	3.4	Hindi	2020-10-30	2020
Drive	Acti…	Novembe…	147	3.5	Hindi	2019-11-01	2019

Name: Shreya Bhattacharjee
3108
Foundation of data science

Conclusion: head , R will print the first six rows of the Dataset

```
> tail(NetflixOriginals)
A tibble: 6 x 11
  Title     Genre Premiere Runtime `IMDB Score` Language Released    Year
  <chr>     <chr> <chr>      <dbl>        <dbl> <chr>    <date>      <dbl>
1 Ben Plat… Conc… May 20,…      85          8.4 English  2020-05-20  2020
2 Taylor S… Conc… Decembe…     125          8.4 English  2018-12-31  2018
3 Winter o… Docu… October…      91          8.4 English… 2015-10-09  2015
4 Springst… One-… Decembe…     153          8.5 English  2018-12-16  2018
5 Emicida:… Docu… Decembe…      89          8.6 Portugu… 2020-12-08  2020
6 David At… Docu… October…      83          9   English  2020-10-04  2020
# … 3 more variables: Month <ord>, Date <int>, Day <ord>
```

Conclusion: tail , R will print the last six rows of the Dataset

```
> dim(NetflixOriginals)
[1] 584  11
>
```

Conclusion: the shape of data frame where 584 rows and 11 columns are there.

```
> any(is.na(NetflixOriginals))
[1] FALSE
>
```

Conclusion: as it is showing false so the dataset don’t contain any null values

```
> str(NetflixOriginals)
tibble [584 × 11] (S3: tbl_df/tbl/data.frame)
 $ Title     : chr [1:584] "Enter the Anime" "Dark Forces" "The App" "The Open House" ...
 $ Genre     : chr [1:584] "Documentary" "Thriller" "Science fiction/Drama" "Horror thriller" ...
 $ Premiere  : chr [1:584] "August 5, 2019" "August 21, 2020" "December 26, 2019" "January 19, 201
 $ Runtime   : num [1:584] 58 81 79 94 90 147 112 149 73 139 ...
 $ IMDB Score: num [1:584] 2.5 2.6 2.6 3.2 3.4 3.5 3.7 3.7 3.9 4.1 ...
 $ Language  : chr [1:584] "English/Japanese" "Spanish" "Italian" "English" ...
 $ Released  : Date[1:584], format: "2019-08-05" "2020-08-21" ...
 $ Year      : num [1:584] 2019 2020 2019 2018 2020 ...
 $ Month     : Ord.factor w/ 12 levels "Jan"<"Feb"<"Mar"<..: 8 8 12 1 10 11 12 6 3 5 ...
 $ Date      : int [1:584] 5 21 26 19 30 1 4 5 23 18 ...
 $ Day       : Ord.factor w/ 7 levels "Sunday"<"Monday"<..: 2 6 5 6 6 6 6 6 6 3 ...
>
```

Conclusion: to show all information about dataset

```
> colnames(NetflixOriginals)
 [1] "Title"      "Genre"      "Premiere"   "Runtime"    "IMDB Score" "Language"   "Released"
 [8] "Year"       "Month"      "Date"       "Day"
>
```

Name: Shreya Bhattacharjee
3108
Foundation of data science

Conclusion: to show all the column names

```
> sapply(NetflixOriginals,class)
$Title
[1] "character"

$Genre
[1] "character"

$Premiere
[1] "character"

$Runtime
[1] "numeric"

$`IMDB Score`
[1] "numeric"

$Language
[1] "character"

$Released
[1] "Date"

$Year
[1] "numeric"
```

Conclusion: to show each column name and data type

Name: Shreya Bhattacharjee
3108
Foundation of data science

```
> summary(NetflixOriginals)
   Title              Genre              Premiere            Runtime         IMDB Score
Length:584         Length:584         Length:584         Min.   :  4.00   Min.   :2.500
Class :character   Class :character   Class :character   1st Qu.: 86.00   1st Qu.:5.700
Mode  :character   Mode  :character   Mode  :character   Median : 97.00   Median :6.350
                                                         Mean   : 93.58   Mean   :6.272
                                                         3rd Qu.:108.00   3rd Qu.:7.000
                                                         Max.   :209.00   Max.   :9.000

  Language           Released              Year          Month          Date
Length:584         Min.   :2014-12-13   Min.   :2014   Oct    : 77   Min.   : 1.00
Class :character   1st Qu.:2018-06-27   1st Qu.:2018   Apr    : 63   1st Qu.: 9.00
Mode  :character   Median :2019-10-17   Median :2019   Nov    : 57   Median :16.00
                   Mean   :2019-06-20   Mean   :2019   May    : 53   Mean   :16.35
                   3rd Qu.:2020-09-18   3rd Qu.:2020   Sep    : 53   3rd Qu.:24.00
                   Max.   :2021-05-27   Max.   :2021   Dec    : 51   Max.   :31.00
                                                       (Other):230

       Day
Sunday    :  9
Monday    : 17
Tuesday   : 29
Wednesday: 82
Thursday  : 59
Friday    :383
Saturday  :  5
```

Conclusion: The summary statistics provide an overview of the distribution and range of values for each variable in the dataset.

1. Minimum: The smallest value observed in the variable.
2. 1st Quartile: The value below which 25% of the data falls.
3. Median: The middle value of the variable, separating the lower 50% from the upper 50%.
4. Mean: The average value of the variable.
5. 3rd Quartile: The value below which 75% of the data falls.
6. Maximum: The largest value observed in the variable.
7. Missing Values: The count or percentage of missing values in the variable.

For categorical variables, the summary may include the following information:

1. Mode: The most frequently occurring category in the variable.
2. Frequency: The count or percentage of observations belonging to each category.
3. Missing Values: The count or percentage of missing values in the variable.

Name: Shreya Bhattacharjee
3108
Foundation of data science

```
is.na(NetflixOriginals)
      Title Genre Premiere Runtime IMDB Score Language Released  Year Month  Date   Day
 [1,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
 [2,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
 [3,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
 [4,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
 [5,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
 [6,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
 [7,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
 [8,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
 [9,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
[10,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
[11,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
[12,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
[13,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
[14,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
[15,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
[16,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
[17,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
[18,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
[19,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
[20,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
[21,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
[22,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
[23,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
[24,] FALSE FALSE    FALSE   FALSE      FALSE    FALSE    FALSE FALSE FALSE FALSE FALSE
```

Conclusion: To identify missing values in dataframe .it return the logical matrix where false indicates no missing values

```
sapply(NetflixOriginals, function(x) length(unique(x)))
    Title      Genre   Premiere    Runtime IMDB Score   Language   Released       Year      Month
      584        115        390        124         54         38        387          8         12
     Date        Day
       31          7
```

Conclusion: count the unique values of each columns

Name: Shreya Bhattacharjee
3108
Foundation of data science

```
> unique(NetflixOriginals$Genre)
  [1] "Documentary"                    "Thriller"
  [3] "Science fiction/Drama"          "Horror thriller"
  [5] "Mystery"                        "Action"
  [7] "Comedy"                         "Heist film/Thriller"
  [9] "Musical/Western/Fantasy"        "Drama"
 [11] "Romantic comedy"                "Action comedy"
 [13] "Horror anthology"               "Political thriller"
 [15] "Superhero-Comedy"               "Horror"
 [17] "Romance drama"                  "Anime / Short"
 [19] "Superhero"                      "Heist"
 [21] "Western"                        "Animation/Superhero"
 [23] "Family film"                    "Action-thriller"
 [25] "Teen comedy-drama"              "Romantic drama"
 [27] "Animation"                      "Aftershow / Interview"
 [29] "Christmas musical"              "Science fiction adventure"
 [31] "Science fiction"                "Variety show"
 [33] "Comedy-drama"                   "Comedy/Fantasy/Family"
 [35] "Supernatural drama"             "Action/Comedy"
 [37] "Action/Science fiction"         "Romantic teenage drama"
 [39] "Comedy / Musical"               "Musical"
 [41] "Science fiction/Mystery"        "Crime drama"
 [43] "Psychological thriller drama"   "Adventure/Comedy"
 [45] "Black comedy"                   "Romance"
 [47] "Horror comedy"                  "Christian musical"
 [49] "Romantic teen drama"            "Family"
```

Conclusion: Extracts the unique all Genre name from the column

```
> summary(is.na(NetflixOriginals))
   Title           Genre          Premiere        Runtime        IMDB Score       Language
 Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical
 FALSE:584       FALSE:584       FALSE:584       FALSE:584       FALSE:584       FALSE:584
  Released          Year           Month            Date            Day
 Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical
 FALSE:584       FALSE:584       FALSE:584       FALSE:584       FALSE:584
>
```

Conclusion: To check the null values this data set has no null values.

```
> colSums(is.na(NetflixOriginals))
    Title      Genre   Premiere    Runtime IMDB Score   Language   Released       Year      Month
        0          0          0          0          0          0          0          0          0
     Date        Day
        0          0
>
```

Conclusion: To check the null values in number format

Name: Shreya Bhattacharjee
3108
Foundation of data science

```
> min(NetflixOriginals$`IMDB Score`)
[1] 2.5
> max(NetflixOriginals$`IMDB Score`)
[1] 9
> sd(NetflixOriginals$`IMDB Score`)
[1] 0.9792564
>
```

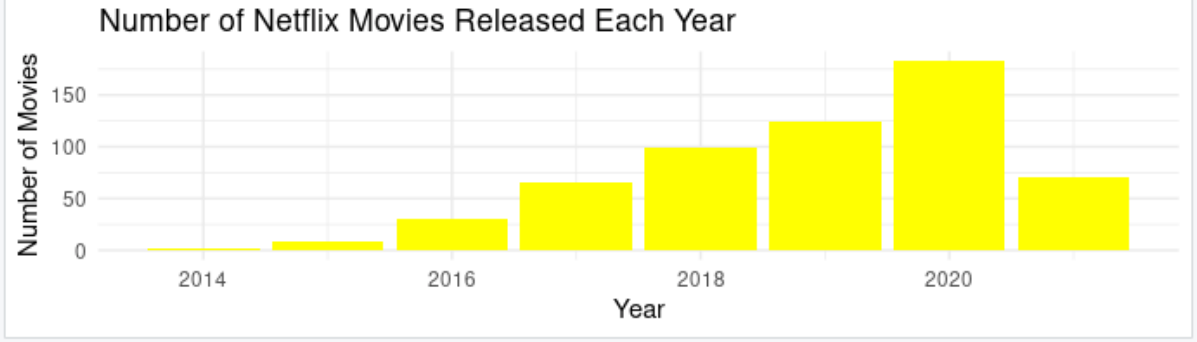
Conclusion: to calculate min,max,sd of IMDB SCORE COLUMN.

```
# Now you can use the %>% operator
NetflixOriginals <- NetflixOriginals %>%
  mutate(Released = mdy(Premiere))
NetflixOriginals <- NetflixOriginals %>%
  mutate(Year = year(Released)) %>%
  mutate(Month = month(Released, label=TRUE)) %>%
  mutate(Date = day(Released)) %>%
  mutate(Day = wday(Released, label=TRUE, abbr=FALSE))
```

Conclusion: By performing these operations, the code adds new variables (`Released`, `Year`, `Month`, `Date`, and `Day`) to the `NetflixOriginals` dataset, which provide more detailed information about the release dates. These new variables can be used for further analysis, such as exploring trends in release dates over time or analyzing the distribution of releases by month or day of the week.

```
# Create a bar chart for the number of movies per year
n_year <- NetflixOriginals %>% group_by(Year) %>% summarise(total=n())
n_year_graph <- ggplot(data=n_year, aes(x=Year, y=total)) +
  geom_bar(stat="identity", fill="yellow") +
  labs(title="Number of Netflix Movies Released Each Year",
       x="Year", y="Number of Movies") +
  theme_minimal()
n_year_graph
```

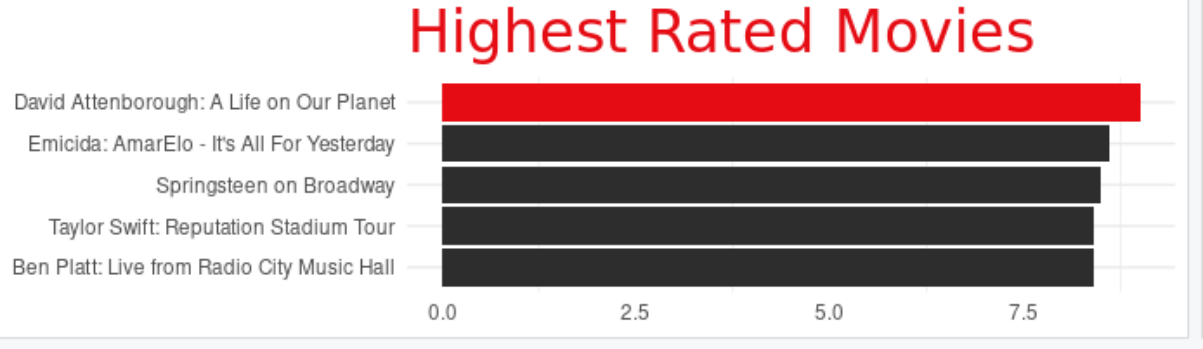
Name: Shreya Bhattacharjee
3108
Foundation of data science



Conclusion: The x-axis represents the years, the y-axis represents the count of movies, and each bar represents the number of movies released in a specific year. The plot is titled "Number of Netflix Movies Released Each Year" and has a minimalistic theme applied to it.

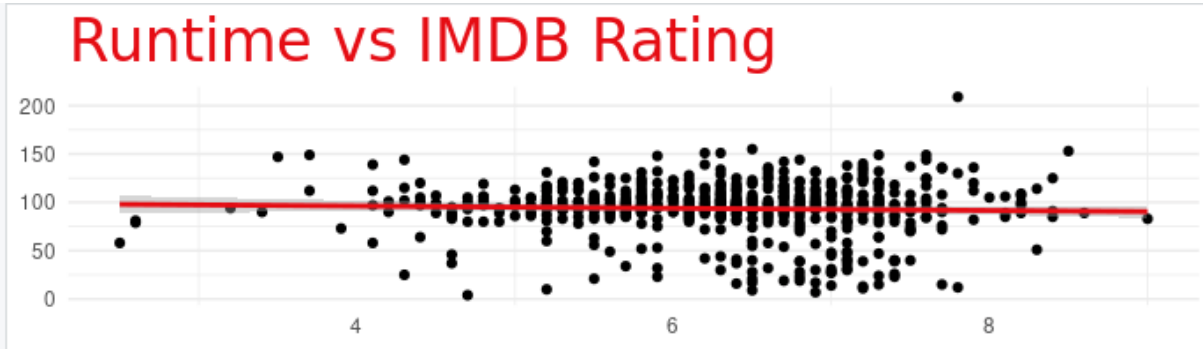
```
n <- NetflixOriginals %>% arrange(desc(`IMDB Score`)) %>% head(5)
> n_graph <- ggplot(data=n)+
+   geom_col(mapping=aes(
+     x=reorder(`Title`,`IMDB Score`),
+     y=`IMDB Score`,
+     fill=ifelse(
+       `IMDB Score`==max(`IMDB Score`),
+       "red","black")))+
+   labs(title="Highest Rated Movies")+
+   theme_minimal()+
+   scale_fill_manual(values = c("#2d2d2d","#E50914"))+
+   coord_flip()+
+   theme(
+     legend.position="none",
+     plot.title = element_text(
+       family="Bebas Neue",
+       size=25,
+       color="#E50914"),
+     axis.title.x=element_blank(),
+     axis.title.y=element_blank(),
+     panel.grid.major.x=element_blank()
+   )
> n_graph

>
```

Name: Shreya Bhattacharjee
3108
Foundation of data science



Conclusion: It creates a horizontal bar chart showing the top 5 highest-rated Netflix Originals based on IMDb scores, with bars color red for the highest-rated title and black for the others.

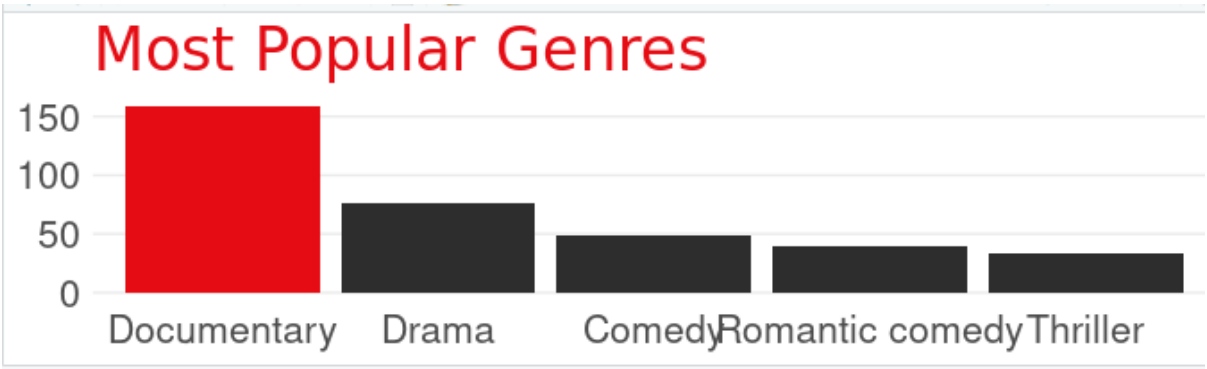
```
n_graph <- ggplot(data=NetflixOriginals,aes(x = `IMDB Score`, y = Runtime))+
+   geom_point()+
+   geom_smooth(method = "lm", color="#E50914")+
+   labs(title="Runtime vs IMDB Rating")+
+   theme_minimal()+
+   scale_fill_manual(values=c("#2d2d2d","#E50914"))+
+   theme(
+     legend.position = "none",
+     plot.title=element_text(
+       family="Bebas Neue",
+       size=25,
+       color="#E50914"),
+     axis.title.x=element_blank(),
+     axis.title.y=element_blank(),
+     panel.grid.major.x=element_blank()
+   )
> n_graph
```



Conclusion: IT generates a scatter plot with a linear regression line showing the relationship between IMDb scores and runtime for Netflix Originals. The plot has a minimalistic appearance with a customized title and no legend. The linear regression line helps visualize any potential linear relationship between these two variables.

Name: Shreya Bhattacharjee
3108
Foundation of data science

```
#Most popular Genres
> n <- NetflixOriginals%>% group_by(Genre) %>%
+   summarise(Movies=n()) %>%
+   arrange(desc(Movies)) %>%
+   head(5)
> n_graph <-
+   ggplot(data=n)+
+   geom_col(mapping = aes(
+     x=reorder(Genre, -Movies),
+     y=Movies,
+     fill=ifelse(Movies == max(Movies),"red","black")))+
+   labs(title="Most Popular Genres")+
+   theme_minimal()+
+   scale_fill_manual(values = c("#2d2d2d","#E50914"))+
+   theme(
+     legend.position="none",
+     plot.title = element_text(
+       family="Bebas Neue",
+       size=25,
+       color="#E50914"),
+     axis.title.x=element_blank(),
+     axis.title.y=element_blank(),
+     panel.grid.major.x=element_blank(),
+     panel.grid.minor = element_blank(),
+     text = element_text(size=20)
+   )
> n_graph
```



Conclusion: The output is a bar chart titled "Most Popular Genres" that displays the top 5 genres with the most movies in the NetflixOriginals dataset. Where Documentary is the highest

```
# Data cleaning and preprocessing
> NetflixData <- NetflixOriginals %>%
+   select(Year, Genre, `IMDB Score`) %>%
+   na.omit()  # Remove rows with missing values
> # Perform one-hot encoding for Genre
```

Name: Shreya Bhattacharjee
3108
Foundation of data science

```
> NetflixData <- NetflixData %>%
+   separate_rows(Genre, sep=", ") %>%
+   pivot_wider(names_from = Genre, values_from = Genre, values_fn = l
ength, values_fill = 0)
```

Conclusion: The final NetflixData dataset will have the "Year" and "IMDB Score" columns along with additional binary columns representing each genre as one-hot-encoded features. If a movie belongs to a particular genre, the corresponding genre column will have a value of 1; otherwise, it will have a value of 0.
This preprocessing is useful for preparing the data for machine learning tasks where you want to use genre as a feature, and most machine learning algorithms require numerical data, not categorical data like genre names. One-hot encoding helps convert categorical data into a suitable format for modelling

```
# Load the necessary library for splitting
> library(caret)
> # Set a seed for reproducibility
> set.seed(123)
> # Split the data into training (70%) and testing (30%) sets
> splitIndex <- createDataPartition(NetflixData$`IMDB Score`, p = 0.7,
list = FALSE)
> trainData <- NetflixData[splitIndex, ]
> testData <- NetflixData[-splitIndex, ]
```

Conclusion: The training set (trainData) is used to train a machine learning model, while the testing set (testData) is used to evaluate the model's performance. By splitting the data, you can assess how well the model generalizes to unseen data and make conclusions about the model's predictive accuracy and potential usefulness for making predictions about IMDb scores for Netflix content**.**

```
# Load the necessary library for modeling
> library(lmtest)
> # Build a linear regression model
> model <- lm(`IMDB Score` ~ ., data = trainData)
> # Make predictions on the test dataset
> predictions <- predict(model, newdata = testData)

# Calculate RMSE
> rmse <- sqrt(mean((testData$`IMDB Score` - predictions)^2))
> cat("Root Mean Squared Error (RMSE):", rmse, "\n")
```

Name: Shreya Bhattacharjee
3108
Foundation of data science

```
Root Mean Squared Error (RMSE): 1.477801

# Calculate MAE
> mae <- mean(abs(testData$`IMDB Score` - predictions))
> cat("Mean Absolute Error (MAE):", mae, "\n")
Mean Absolute Error (MAE): 1.198614

>
```

Conclusion: A lower RMSE and MAE indicate that the model is making predictions that are closer to the actual IMDb scores, which is generally desirable.

The RMSE value of approximately 1.478 indicates that, on average, the model's predictions are about 1.478 IMDb score points away from the actual scores. The MAE value of approximately 1.199 indicates that, on average, the model's predictions are about 1.199 IMDb score points away from the actual scores. These values can be used to assess the model's predictive accuracy and make conclusions about its suitability for predicting IMDb scores based on the given features. Further analysis and fine-tuning of the model may be necessary to improve its performance if needed.

```
model <- lm(data=NetflixOriginals, formula = Runtime ~ `IMDB Score`)
> summary(model)

Call:
lm(formula = Runtime ~ `IMDB Score`, data = NetflixOriginals)

Residuals:
    Min      1Q  Median      3Q     Max
-91.399  -7.439   3.398  14.467 117.195

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   100.849      7.453  13.531   <2e-16 ***
`IMDB Score`   -1.159      1.174  -0.987    0.324
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 27.76 on 582 degrees of freedom
Multiple R-squared:  0.001673,  Adjusted R-squared:  -4.283e-05
F-statistic: 0.975 on 1 and 582 DF,  p-value: 0.3238
```

Name: Shreya Bhattacharjee
3108
Foundation of data science

```
res <- cor.test(NetflixOriginals$Runtime, NetflixOriginals$`IMDB Score`,method
="pearson")
> res

        Pearson's product-moment correlation

data:  NetflixOriginals$Runtime and NetflixOriginals$`IMDB Score`
t = -0.98744, df = 582, p-value = 0.3238
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.12162699  0.04037194
sample estimates:
        cor
-0.04089629

> #P-Value
> res$p.value
[1] 0.3238393
> #Correlation Coefficient
> res$estimate
        cor
-0.04089629

>
```

Conclusion: The correlation analysis suggests that there is a weak and statistically non-significant negative correlation between Runtime and IMDB Score for Netflix original movies or TV shows in the dataset. This means that changes in IMDB Score are not a strong predictor of changes in Runtime