MovieAnalysis_FinalProject

Vasanthakumar Kalaikkovan

22/05/2021

Importing and Cleaning Data

Rating Dataset importing

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.0.5
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
df_ratings <- read_tsv('data.tsv', na = "\\N", quote = '')
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   tconst = col_character(),
##   averageRating = col_double(),
##   numVotes = col_double()
## )
```

```
df_ratings<- na.omit(df_ratings)
head(df_ratings)
```

```
## # A tibble: 6 x 3
##   tconst    averageRating numVotes
##   <chr>             <dbl>    <dbl>
## 1 tt0000001           5.7     1702
## 2 tt0000002           6.1      210
## 3 tt0000003           6.5     1461
## 4 tt0000004           6.2      123
## 5 tt0000005           6.2     2261
## 6 tt0000006           5.1      127
```

Crew Dataset importing

```
df_crews <- read_tsv('crew_data.tsv',na = "\\N")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   tconst = col_character(),
##   directors = col_character(),
##   writers = col_character()
## )
```

```
df_crews<- na.omit(df_crews)
head(df_crews)
```

```
## # A tibble: 6 x 3
##   tconst    directors writers
##   <chr>     <chr>     <chr>
## 1 tt0000009 nm0085156 nm0085156
## 2 tt0000036 nm0005690 nm0410331
## 3 tt0000076 nm0005690 nm0410331
## 4 tt0000091 nm0617588 nm0617588
## 5 tt0000108 nm0005690 nm0410331
## 6 tt0000109 nm0005690 nm0410331
```

Title Dataset importing

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
df_title_temp <- read_tsv('title_data.tsv',na = "\\N",quote = '')
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   titleId = col_character(),
##   ordering = col_double(),
##   title = col_character(),
##   region = col_character(),
##   language = col_character(),
##   types = col_character(),
##   attributes = col_character(),
##   isOriginalTitle = col_double()
## )
```

```
df_title_temp<- na.omit(df_title_temp)
df_title<-df_title_temp %>% filter(ordering<=1)
head(df_title)
```

```
## # A tibble: 6 x 8
##   titleId  ordering title      region language types attributes  isOriginalTitle
##   <chr>       <dbl> <chr>      <chr>  <chr>    <chr> <chr>                 <dbl>
## 1 tt00225~        1 Di shtime~ US     yi       alte~ YIVO trans~               0
## 2 tt00279~        1 Libe un L~ US     yi       alte~ modern tra~               0
## 3 tt00326~        1 Der yidis~ US     yi       alte~ YIVO trans~               0
## 4 tt00651~        1 Altin Han~ TR     tr       alte~ dubbed ver~               0
## 5 tt00668~        1 Kimin Umu~ TR     tr       imdb~ alternativ~               0
## 6 tt00797~        1 Mavile Kr~ TR     tr       imdb~ dubbed ver~               0
```

Final Dataset

Merging all the datasets on the movie id

```
df_combined <- merge(df_crews,df_ratings)
df_final <-merge(df_title,df_combined,by.x="titleId",by.y="tconst")
head(df_final)
```

```
##     titleId ordering                                        title region
## 1 tt0065172        1                                 Altin Hançer     TR
## 2 tt0066854        1 Kimin Umurunda: Teslimatçi Çocugun Anatomisi     TR
## 3 tt0079768        1                               Mavile Kraliçe     TR
## 4 tt0145916        1                                Bekçi Murtaza     TR
## 5 tt0185027        1                                Yilmayan adam     TR
## 6 tt0259685        1                           Yeralti Canavari 3     TR
##   language       types                  attributes isOriginalTitle
## 1       tr alternative              dubbed version               0
## 2       tr imdbDisplay alternative transliteration               0
## 3       tr imdbDisplay              dubbed version               0
## 4       tr imdbDisplay              complete title               0
## 5       tr imdbDisplay                poster title               0
## 6       tr imdbDisplay                   new title               0
##             directors                                 writers averageRating
## 1           nm0387354                     nm0387354,nm2424349           6.3
## 2 nm0267064,nm1293361                               nm0267064           6.9
## 3           nm0640496                               nm0262783           2.5
## 4           nm0059633                     nm0252375,nm0447158           6.7
## 5           nm0040220                               nm1147694           5.2
## 6           nm0534681 nm0934093,nm0534681,nm0731443,nm0924095           5.3
##   numVotes
## 1      128
## 2      128
## 3      116
## 4       68
## 5      301
## 6    16669
```

```
#Modifying the director id for the visualization purpose
df_final$directors[df_final$directors=="nm7132415,nm0880127,nm12374633,nm3123733,nm1699658"]<-"nm7132415
```

Questions for frture steps

What kind of plot are required to show the optimal output

What is the optimal form to represent the result

What information is not self-evident?

After eliminating the missing data and combining based on the movie id, the final dataset become very less comparing to the initial one. The result is going to be based on the available dataset which is very small.

What are different ways you could look at this data?

We can make the prediction based on the movie and its ratings but again we can predict the director's success rate. Please suggest some other way which will be opt in different ways.

How do you plan to slice and dice the data?

Slicing and dicing the data is happened in the final dataset merging itself.

How could you summarize your data to answer key questions?

Data has the movie title id, director id, number of votes, and average rating. Based on these column we can able to answer our questions.

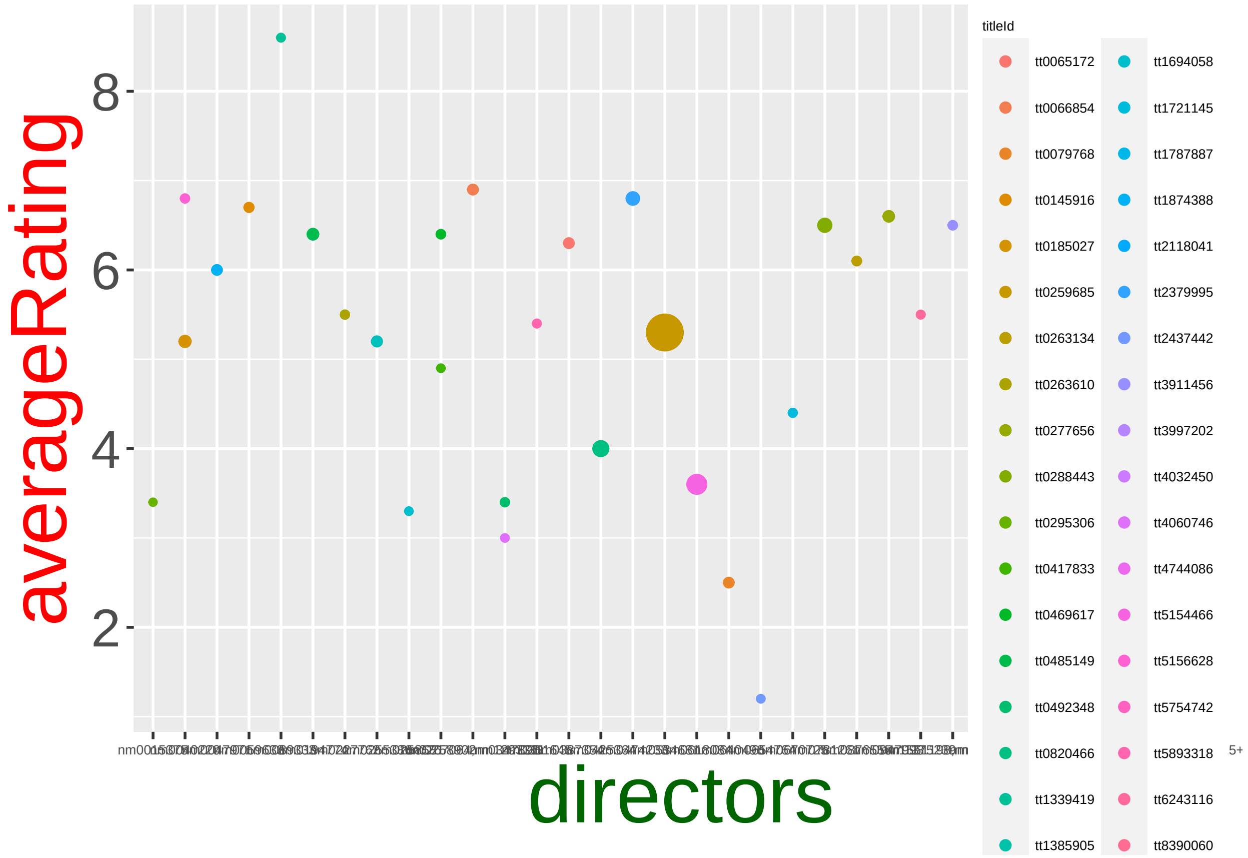
What types of plots and tables will help you to illustrate the findings to your questions?

Scatter plot

```
library(ggplot2)
```

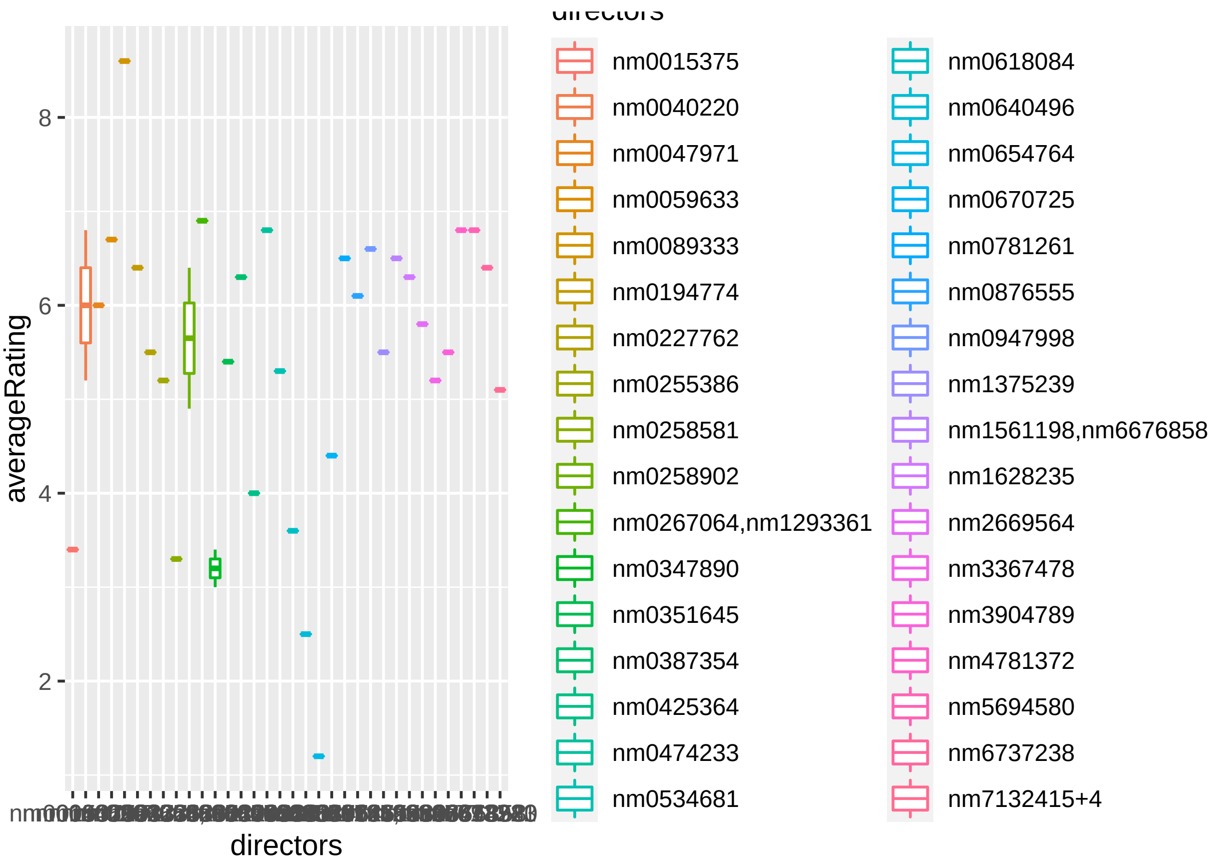
```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
scatter_plot <- ggplot(data=df_final,aes(x=directors,y=averageRating,size=numVotes))+geom_point(aes(colo
  theme(axis.title.x=element_text(colour="DarkGreen",size = 30),
        axis.title.y = element_text(colour = "Red",size = 30),
        axis.text.x = element_text(size=5),
        axis.text.y = element_text(size=20),
        legend.title = element_text(size=5),
        legend.text=element_text(size=5),
        legend.position = c(1,1),
        legend.justification = c(1,1))
scatter_plot
```



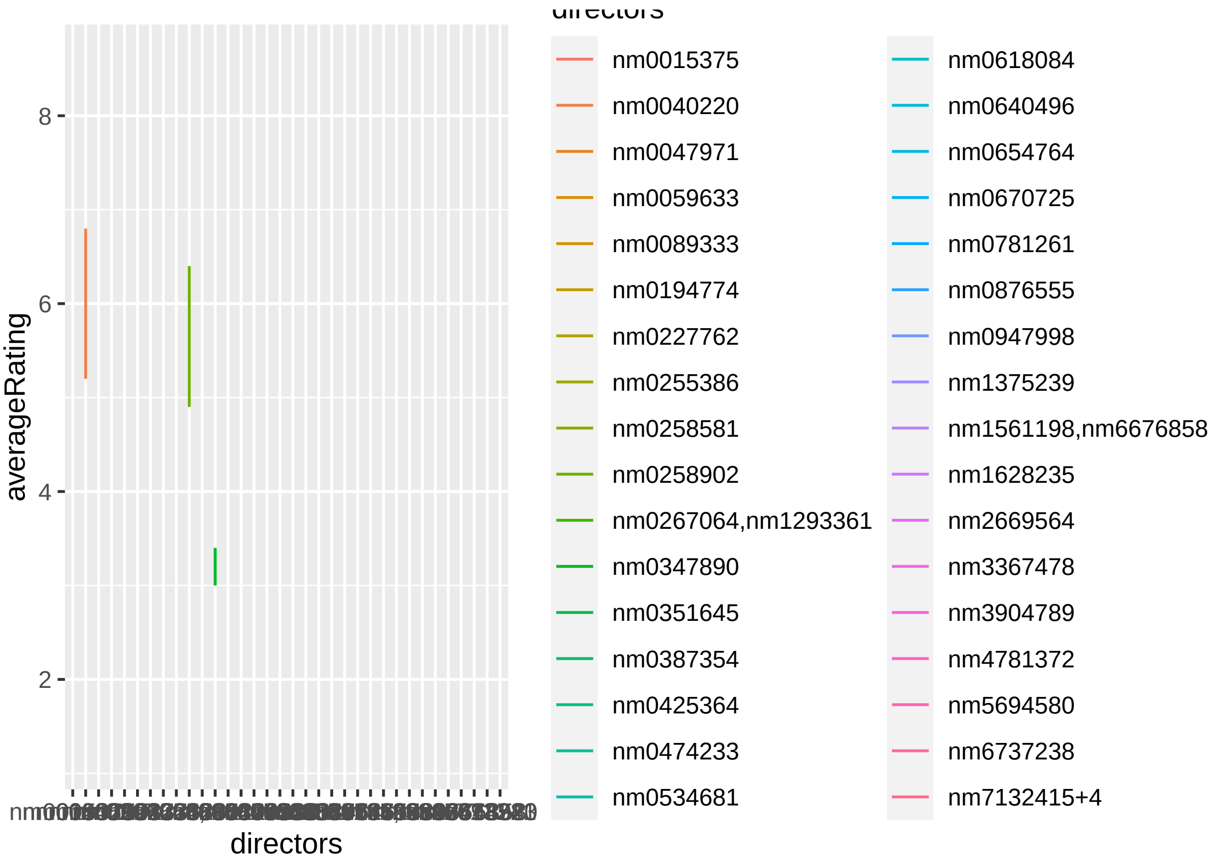
Boxplot

```
boxplot<-ggplot(data=df_final,aes(x=directors,y=averageRating,colour=directors))+geom_boxplot(aes(colou
  theme(axis.title.x=element_text(colour="DarkGreen",size = 30),
        axis.title.y = element_text(colour = "Red",size = 30),
        axis.text.x = element_text(size=2),
        axis.text.y = element_text(size=20),
        legend.title = element_text(size=2),
        legend.text=element_text(size=2),
        legend.position = c(1,1),
        legend.justification = c(1,1)))
boxplot
```



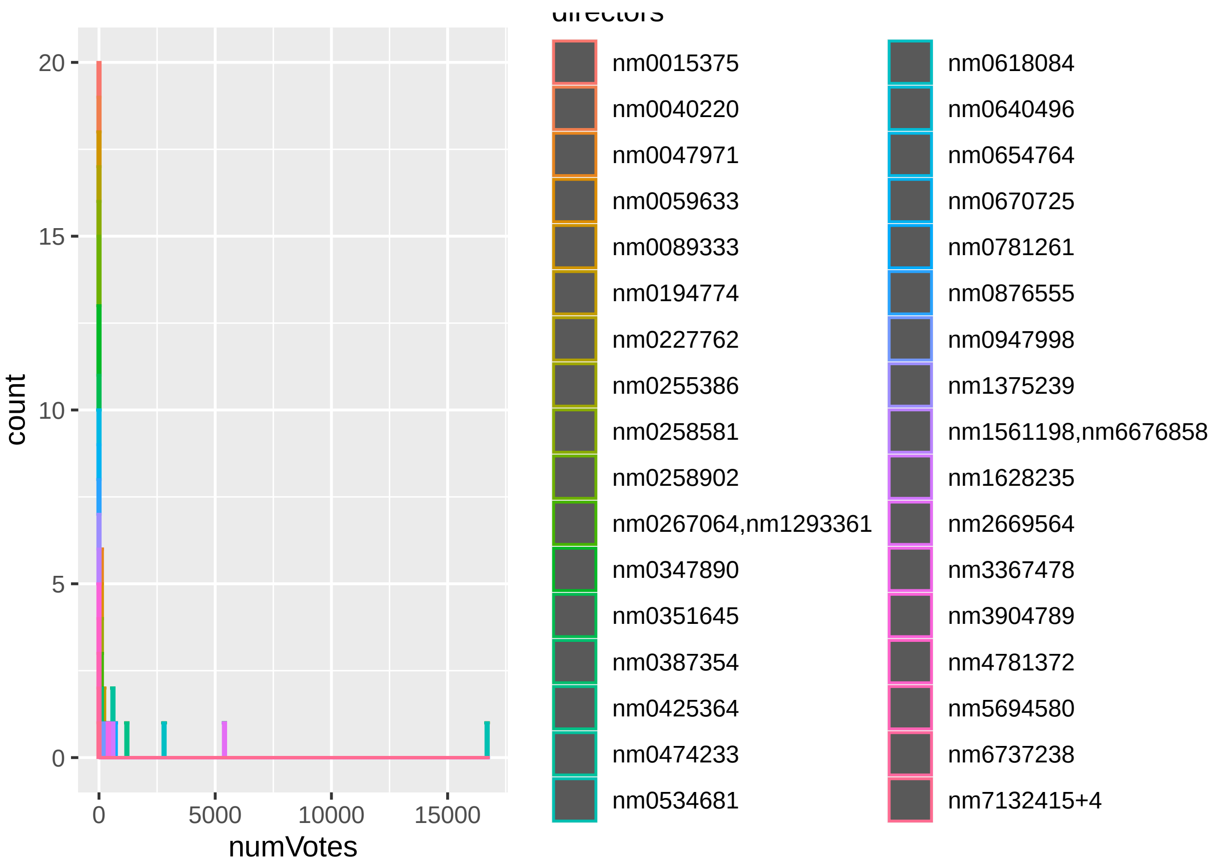
Trend lines

```
trend_line<-ggplot(data=df_final,aes(x=directors,y=averageRating,colour=directors))+geom_line()
trend_line
```



Histogram

```
histogram<-ggplot(data=df_final,aes(x=numVotes,colour=directors))+geom_histogram(binwidth = 100)
histogram
```



Do you plan on incorporating any machine learning techniques to answer your research questions? Explain.

With the help the plot only, we can answer our question and there wont be any requirement for machine learning