Toronto

Initial Analysis

Soni Manan[a]

[a] *College of Professional Studies, Master of Professional Studies in Analytics. Toronto*

Subject: *ALY6070* ***NUID***: *002982645*

Under the guidance of

Prof. Dr. Shahram Sattar

Introduction

This assignment is about to explore the given datasets which are a part of population. This data has 2 data about movies, actors and director. The purpose of this assignment is to present the easy yet effective visualization and squeeze it to gain knowledge what data has. Goal is to identify the datasets according to features and finding target audience, beneficial key findings from Data.

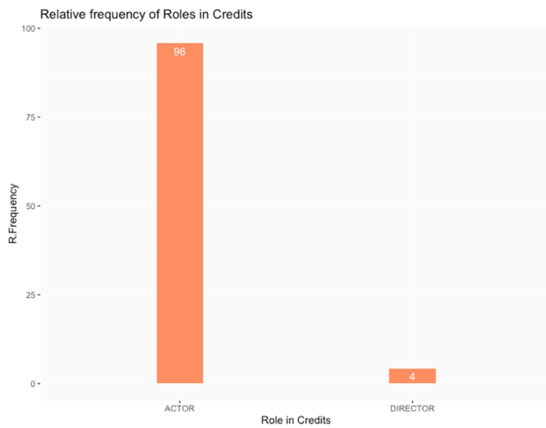
Exploratory Data Analysis

credits.csv has 31802 samples and 5 columns. **Name, character, id** and **role** are categorical and other is **person_id** which is numerical. There are no null values to be cleaned. There is not much data we can gain only by **credits.csv**.

```
# Libraries
library(ggplot2)
library(dplyr)

# Setting the path for current directory
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))

# Head of credits
> head(credits)
  person_id      id          name                 character  role
1     85144 ts20475   Aidy Bryant Self - Various Characters ACTOR
2     85141 ts20475   Michael Che Self - Various Characters ACTOR
3     87723 ts20475 Pete Davidson Self - Various Characters ACTOR
4     99154 ts20475     Mikey Day Self - Various Characters ACTOR
5     26921 ts20475    Colin Jost Self - Various Characters ACTOR
6      3837 ts20475 Kate McKinnon Self - Various Characters ACTOR

# Number of samples
> nrow(credits)
[1] 31802

# Number of Duplicated
> sum(duplicated(credits,by=c('id','person_id')))
[1] 0

# Structure
> str(credits)
'data.frame':	31802 obs. of  5 variables:
 $ person_id: int  85144 85141 87723 99154 26921 3837 116824 85142 4478 36230 ...
 $ id       : chr  "ts20475" "ts20475" "ts20475" "ts20475" ...
 $ name     : chr  "Aidy Bryant" "Michael Che" "Pete Davidson" "Mikey Day" ...
 $ character: chr  "Self - Various Characters" "Self - Various Characters" "Self - Various Characters" "Se
lf - Various Characters" ...
 $ role     : chr  "ACTOR" "ACTOR" "ACTOR" "ACTOR" ...

> unique(credits$role) # 2 Levels actor and director
[1] "ACTOR"    "DIRECTOR"
```

titles.csv has 2398 samples and 15 features.Some are categorical. There are null values to be cleaned such as in age_certification, imdb_score, imdb_votes, tmdb_score and tmdb_popularity.

We can left join this with credits.csv for getting respective movies and shows but reason behind not doing is it can increase noise. Like for 1 movie there are 5 actors.



96% role of credits samples are Actor and rest are directors.

```
> ggplot(credits %>% group_by(role) %>% summarise(count = (n()/nrow(credits))*100)
    ,aes(y=count,x=unique(role))) +
  geom_bar(stat='identity',width = 0.2,fill='#FC8E62') +
  geom_text(aes(label=round(count)),vjust=1.5,col='white') +
  labs(x='Role in Credits',y='R.Frequency') +
  ggtitle("Relative frequency of Roles in Credits") +
  theme(panel.background = element_rect(fill="#fafafa"))

# Structure of titles.csv
str(titles)
'data.frame':   2398 obs. of  15 variables:
 $ id                  : chr  "ts20475" "ts20413" "ts20005" "ts20669" ...
 $ title               : chr  "Saturday Night Live" "M*A*S*H" "I Love Lucy" "Taxi" ...
 $ type                : chr  "SHOW" "SHOW" "SHOW" "SHOW" ...
 $ description         : chr  "A late-night live ...
 $ release_year        : int  1975 1972 1951 1978 1970 1969 1972 1965 1970 1972 ...
 $ age_certification   : chr  "TV-14" "TV-PG" "TV-G" "TV-PG" ...
 $ runtime             : int  89 26 30 25 28 25 27 50 27 25 ...
 $ genres              : chr  "['music', 'comedy']" "['war', 'comedy', 'drama']" "['comedy', 'family']" "[
'drama', 'comedy']" ...
 $ production_countries: chr  "['US']" "['US']" "['US']" "['US']" ...
 $ seasons             : num  47 11 9 5 7 5 6 3 5 NA ...
 $ imdb_id             : chr  "tt0072562" "tt0068098" "tt0043208" "tt0077089" ...
 $ imdb_score          : num  8 8.4 8.5 7.7 8.2 6.7 8.1 5.7 7.9 7.9 ...
 $ imdb_votes          : num  47910 55882 25944 13379 8692 ...
 $ tmdb_popularity     : num  54.34 27.31 17.09 14.35 9.29 ...
 $ tmdb_score          : num  6.9 8 8.1 7.3 7.5 7 7.7 7.1 8.1 7.2 ...
```

From the describe we can get some hidden values in data like earliest release year is 1951 and most recent is 2022. Almost all of the variable have skewness. Following variable have less skewness such as runtime, tmdb_score, imdb_score and standard deviation is also around 1.21 and 1.22 respectively. Mean of imdb_score is 6.7 and tmdb_score is 6.89. which looks like co-relating each other for now. We can have idea about normalization from skewness. Next should be getting the null values.

```
> psych::describe(titles)
                     vars    n     mean       sd median trimmed     mad     min       max     range  skew kurtosis      se
id*                     1 2398  1199.50   692.39 1199.5 1199.50  888.82    1.00   2398.00   2397.00  0.00    -1.20   14.14
title*                  2 2398  1190.67   687.52 1191.5 1190.80  883.63    1.00   2379.00   2378.00  0.00    -1.20   14.04
type*                   3 2398     1.55     0.50    2.0    1.57    0.00    1.00      2.00      1.00 -0.22    -1.95    0.01
description*            4 2398  1187.53   692.33 1187.5 1187.50  888.82    1.00   2386.00   2385.00  0.00    -1.20   14.14
release_year            5 2398  2013.42     8.48 2016.0 2014.91    5.93 1951.00   2022.00     71.00 -2.12     6.57    0.17
age_certification*      6 2398     5.46     3.29    6.0    5.40    4.45    1.00     12.00     11.00 -0.16    -1.19    0.07
runtime                 7 2398    61.52    35.10   48.0   59.76   38.55    0.00    229.00    229.00  0.34    -0.99    0.72
genres*                 8 2398   499.39   258.40  451.5  490.40  286.88    1.00   1026.00   1025.00  0.29    -0.78    5.28
production_countries*   9 2398   167.52    50.80  199.0  179.00    0.00    1.00    206.00    205.00 -1.63     1.83    1.04
seasons                10 1330     3.94     5.02    2.0    2.86    1.48    1.00     63.00     62.00  4.16    27.54    0.14
imdb_id*               11 2398  1069.27   686.30 1065.5 1065.50  888.82    1.00   2264.00   2263.00  0.03    -1.23   14.01
imdb_score             12 2232     6.70     1.21    6.8    6.78    1.19    1.00      9.50      8.50 -0.64     0.37    0.03
imdb_votes             13 2231 28286.88 78020.10 3451.0 9912.95 4861.45    5.00 996056.00 996051.00  5.63    41.32 1651.80
tmdb_popularity        14 2348    27.87    92.00   10.7   14.45   10.51    0.27   2989.85   2989.57 18.51   503.13    1.90
tmdb_score             15 2238     6.89     1.22    7.0    6.92    1.19    1.00     10.00      9.00 -0.55     1.69    0.03
```

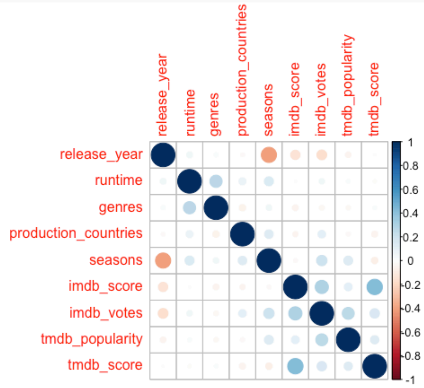
Calculating number of nulls in features and got some values. Seasons have 0 because there is variable named <type> which has movies and found out that seasons and sample of movies are same so logically movies don't have seasons. Removing Null values is not solution so replaced it with mean and mode.

```
# getting nulls
> for (i in colnames(titles)){
+   if(sum(is.na(titles[[i]])) > 1)
+   cat(i,"->",sum(is.na(titles[[i]])),"\n")
+ }
seasons -> 1068
imdb_score -> 166
imdb_votes -> 167
tmdb_popularity -> 50
tmdb_score -> 160

# Number of samples with type MOVIE
> length(titles$type[titles$type == "MOVIE"])
[1] 1068

# MODE function which returns most frequency after NA value
Mode <- function(x){
  ux <- unique(x)
  ux = ux[ux != ""]
  ux = ux[!is.na(ux)]
  ux[which.max(tabulate(match(x, ux)))]
}

# REPLACED age_certification values with MODE of it
# Actual mode is NA because many samples had NA so replace with second MODE value after that.
titles$age_certification[titles$age_certification == ""] = Mode(titles$age_certification)
# replaced mean in place of NA
titles$imdb_score[is.na(titles$imdb_score)] = mean(titles$imdb_score,na.rm=TRUE)
titles$tmdb_score[is.na(titles$tmdb_score)] = mean(titles$tmdb_score,na.rm=TRUE)
```

As we can see that orange to red is opposite correlation and light-blue to blue is for positive. We can note that seasons of shows have negative relation. imdb_score and imdb_votes have positive correlation. tmdb_score and imdb_score has positive correlation which demonstrate that score of movies are actually good for some.

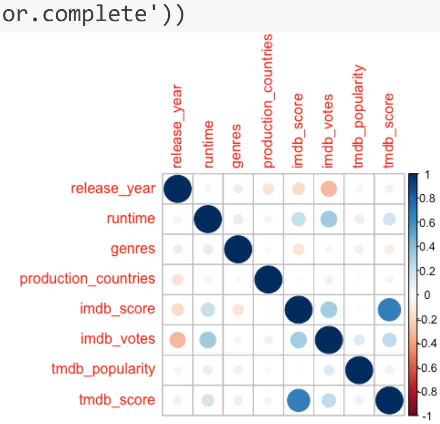
```
# Corrplot of SHOWs
> SHOWs = titles %>% filter(type == 'SHOW')
```

```
> SHOWs = select(SHOWs,-c(id,title,description,type,age_certification,imdb_id,Duration,prd_country_fct))
> SHOWs$genres = as.numeric(as.factor(SHOWs$genres))
> SHOWs$production_countries = as.numeric(as.factor(SHOWs$production_countries))
> corrplot(cor(SHOWs,use='na.or.complete'))
```



Here, for movies scenario is different like imdb_score has strong negative correlation with release year so reason can be elders don't like new movies enough or classics are best. Imdb_score and tmdb_score have strong number than shows had.

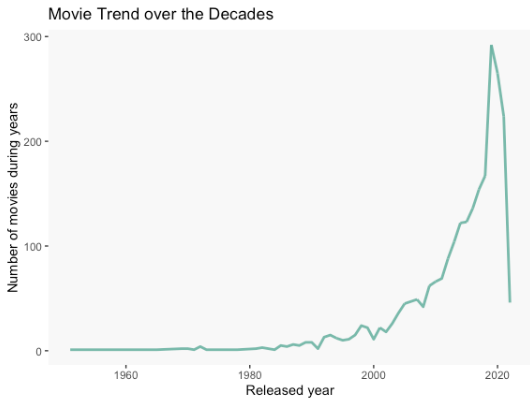
```
# Corrplot of Movies
> movies = titles %>% filter(type == "MOVIE") %>% select(-c(id,title,description,type,age_cert
ification,imdb_id,'seasons',Duration,prd_country_fct))
> movies = select(movies,-c(id,title,description,type,age_certification,imdb_id))
> movies$genres = as.numeric(as.factor(movies$genres))
> movies$production_countries = as.numeric(as.factor(movies$production_countries))
> corrplot(cor(movies,use='na.or.complete'))
```

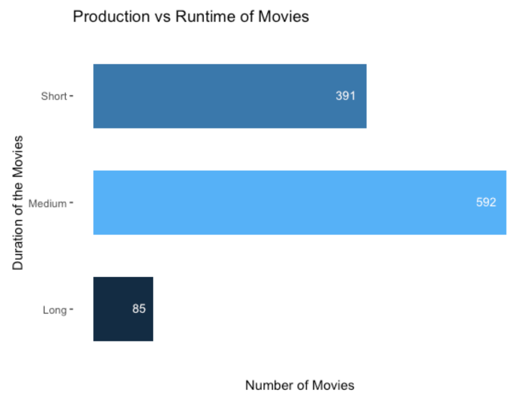



```
# Plotting continuous movie release by years (LINE)
>ggplot(titles %>% group_by(release_year) %>% mutate(count = n()), aes(x=release_year, y=count)) +
  geom_line( color="#69b3a2", size=1, alpha=0.9, linetype=1) +
  labs(x='Released year',y='Number of movies during years')+
  theme(panel.background = element_rect(fill='#f8f8f8'),panel.grid = element_blank()) +
  ggtitle("Movie Trend over the Decades ")
# Comparing movies factorized into length of movie based on runtime
> movies= movies %>%
  mutate(Duration = ifelse(runtime <=90, "Short", ifelse(runtime <= 120, "Medium", "Long")))
> ggplot(data= movies %>% group_by(Duration) %>% filter(type=='MOVIE') %>%
         summarise(count=n()), aes(y=count, x=Duration, fill=count)) +
```

```
        geom_bar(stat= 'identity',width=0.6)+ xlab('Duration of the Movies') + ylab("Number of Movies")  +
        ggtitle("Production vs Runtime of Movies")+
        theme(panel.background = element_rect(fill='#ffffff'), legend.position = 'none',axis.ticks.x = ele
ment_blank(), axis.text.x = element_blank()) +
        coord_flip() + geom_text(aes(label=(count)), vjust=0.4,hjust=1.5, color="white", size=3.5)
```

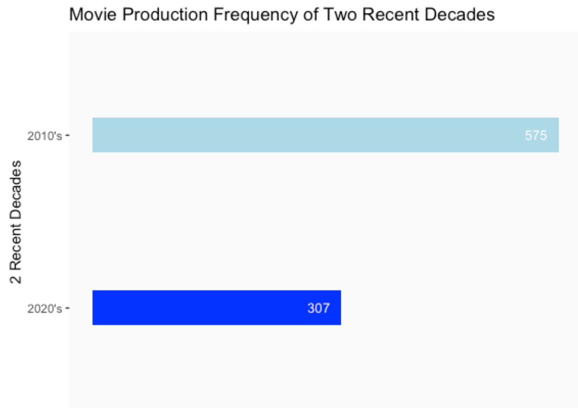
Here is the trend line of movie production according to years. Starting 1951 to 2022. Downfall at 2020 is because 2022 is running and rest is remaining which has current record. In this, **'Short'** has runtime of less than 90, other is less than 120 which is **'Medium'** and **'Long'** is above it 120 minutes.

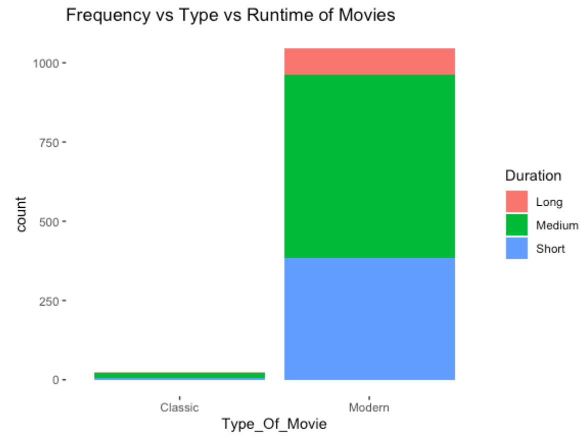





```
> movies <- titles %>% filter(type=='MOVIE') %>%
  mutate(Decade = if_else(release_year >= 2000,
                          paste0(release_year  %/% 10 * 10, "'s"),
                          paste0((release_year - 1900) %/% 10 * 10, "'s")))

> movies %>% group_by(Decade) %>% summarise(count=n()) %>% arrange(desc(count))

> ggplot(movies %>% group_by(Decade) %>%
         summarise(count=n()) %>% filter(count > 200) %>%
         arrange(desc(count)),aes(y=reorder(Decade, count),x=count)) +
  geom_bar(stat = 'identity',width = 0.2,fill=c('lightblue','blue')) +
  theme(panel.background = element_rect("#fafafa"),panel.grid = element_blank(),axis.text.x =
element_blank(),axis.ticks.x = element_blank()) +
  ggtitle("Movie Production Frequency of Two Recent Decades") +
  ylab('2 Recent Decades') + xlab("") +
  geom_text(aes(label=count), vjust=0.4,hjust=1.5, color="white", size=3.5)
```

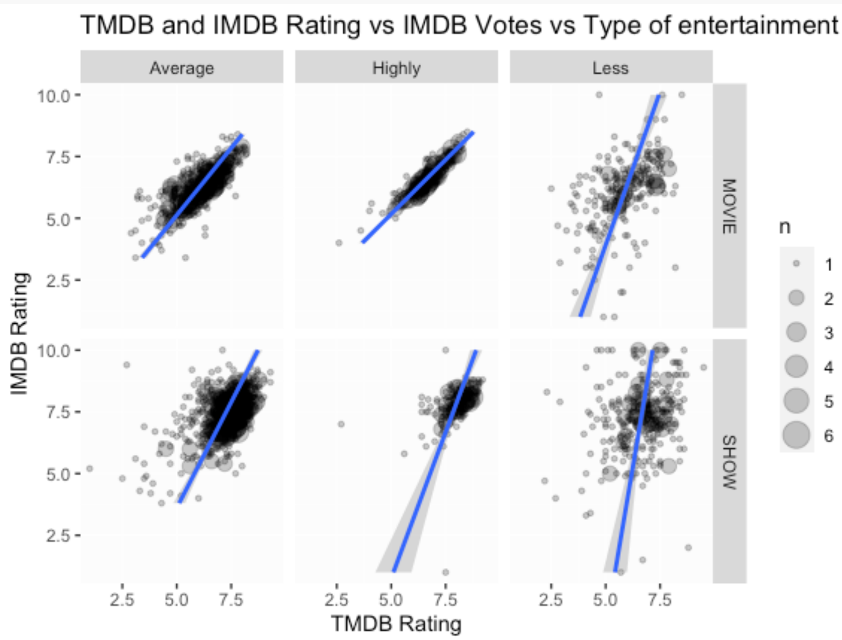
As above bar plot was showing kind of exponential growth and in just 2 years of 2020 it reaches more than half count of 2010's number according to sample.





Above vertical bar graph is according to **release_year**, I've categorized as **'Old', 'Classics' and 'Modern'** in which movies older than 1953, 1990 and greater than 1990 are respectively. And result that There are almost 0 movies before 1953 so not mentioned in graph. In addition, I've illustrated Runtime duration of movies which were set earlier.

```
> movies <- movies %>%
  mutate(Type_Of_Movie = ifelse(release_year <=1953, "Old", ifelse(release_year <= 1990, "Classic", "Moder
n")))

> ggplot(movies %>%
         group_by(Type_Of_Movie, Duration) %>%
         summarise(count=n()), aes(y=count, x=Type_Of_Movie, fill=Duration)) +
  geom_bar(stat='identity') +
  ggtitle("Frequency vs Type vs Runtime of Movies")
```

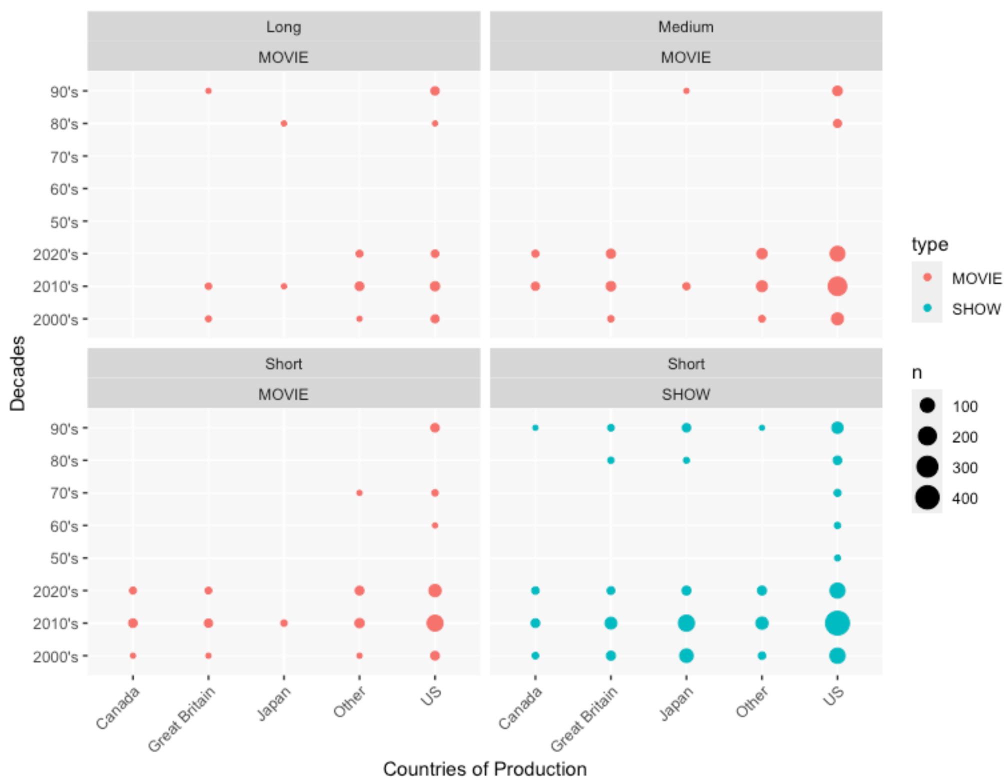


As we've seen previously, we have correlation between tmdb_rating and imdb_rating. So plotting it with number of imdb votes and type of entertainment which are Movies and Shows.

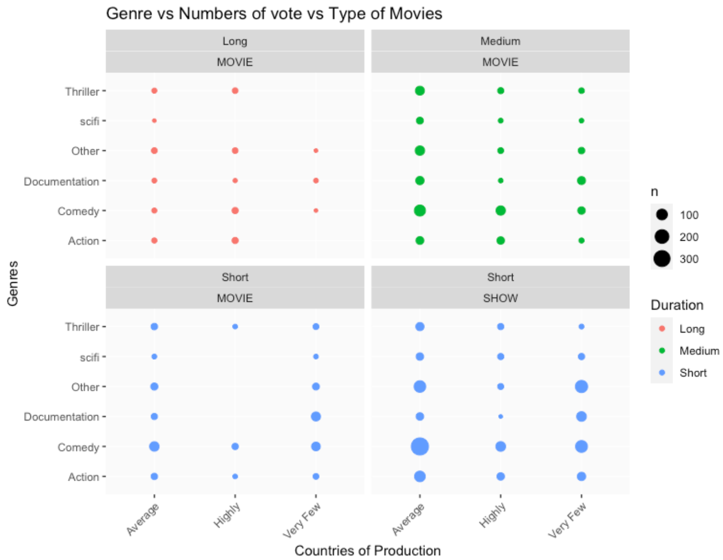
So, we can note from above graph according to categorized of votes based on mean of votes. Spread data is concentrating around mean of tmdb and imdb. And It's been seen that for shows which are voted highly have good concentration and can use for prediction as well.

```
>titles <- titles %>% mutate(voted = ifelse(imdb_votes <=1000, "Less", ifelse(imdb_votes <= mean(titles$imdb_votes,na.rm = T), "Average", "Highly")))
> ggplot(titles %>% filter(!is.na(voted)),aes(x=tmdb_score,y=imdb_score)) +
  geom_count(alpha=0.25)+ coord_flip()+ stat_smooth(method='lm')+
  facet_grid(rows=vars(type),cols=vars(voted)) +
  ggtitle('TMDB and IMDB Rating vs IMDB Votes vs Type of entertainment') +
  theme(panel.background = element_rect(fill='#fcfcfc')) +
  xlab('IMDB Rating') + ylab("TMDB Rating")
```

From this, I've subsetted data which have production country such as US and Japan. Japanese film makers are tending to short shows and animes are famous in japan which are less than 60 minutes. Others are combined country of productions which includes US, Japan, Canada, Dutch, France and Great Britain. The plot is illustrated below. According to this dataset, 50's ,60's, 70's have very less production.




```
>titles = titles %>% mutate(prd_country = if_else(grepl("'US'",titles$production_countries,fixed=T),'US',
                                    if_else(grepl("JP",titles$production_countries,fixed=T),'Japan',
                              ifelse(grepl("GB",titles$production_countries,fixed=T),'Great Britain',
                      ifelse(grepl("CA",titles$production_countries,fixed=T),'Canada','Other')))))
> ggplot(titles,aes(Decade,prd_country,col=type)) +
  geom_count(position = 'identity')+ facet_wrap(Duration ~ type ) + coord_flip() +
  xlab('Decades') +  ylab('Countries of Production') +
  theme(axis.text.x = element_text(angle=45,vjust=1,hjust=1),panel.background = element_rect(fill="#fafafa"))
```



Movie with short length are less likely to get higher rating. Long action, sci-fi and thrillers are more likely to be unvoted. People's preferences are Medium movie and short shows.

```
> titles <- titles %>% mutate(voted = ifelse(imdb_votes <=1000, "Less", ifelse(imdb_votes <= mean(titles$imdb_votes,na.rm = T), "Average", "Highly")))
> ggplot(titles %>% filter(!is.na(voted)),aes(x=tmdb_score,y=imdb_score)) +
  geom_count(alpha=0.25)+ coord_flip()+ stat_smooth(method='lm')+
  facet_grid(rows=vars(type),cols=vars(voted)) +
  ggtitle('TMDB and IMDB Rating vs IMDB Votes vs Type of entertainment') +
  theme(panel.background = element_rect(fill='#fcfcfc')) +
  xlab('IMDB Rating') + ylab("TMDB Rating")
```

Conclusion

Primarily, after this of analysis, **for the production houses**, I believe **people are ought to vote more** for **medium movies** and **shows that are short** so they need to focus at these categories. In addition, **giving a thought about viewers**, they are always **coming under the target audience whether it is product or service**. Japanese shows as known as animes are getting highest imdb as us shows and movies. People are enjoying and voting also because there are a lot of voters from US and Japanese shows getting popularity in all over world.

From **the perspective of imdb rating and modeling**, it uses '**Bayesian estimate**' and according to that the algorithm of top movies and ratings work. So, number of votes are important for a movie and shows. But for 250 top movie list movies required to have at least 25 hundred votes. Therefore, Movies and shows who are short and medium gets very good amount of rating which is near to average or above average as we have seen in above plotting's. **Dataset is lack of revenue of movies and shows.** In fact, trend has much more increased toward shows and series because it increases hype due to runtime.

TMDB's popularity is something else like a variable. By mean, it is calculated using many variables. So if it's high numbers for that movie like number of votes for today, number of views for today, previous day score and etc. Therefore, it's something we can't be relay.

From the decades, movie industry all over world facing many issues like fear of wars especially with nuclear in 40s-50s, racism, etc.

Modification

Comment:	**Action:**
Explanation of Audience is missing in the report.	I've added two paragraph containing audience perspective. There are two opposite perspectives targeting more votes and popularity. Some of the key points which requires to focus are formatted as bold.

Bibliography

Exploratory Data Analysis with R. (n.d.). Retrieved from Bookdown.org: https://bookdown.org/rdpeng/exdata/exploratory-graphs.html

Hayden, L. (2018, Mar 30). *Graph density with scatterplot ggplot2 in R*. Retrieved from Stackover Flow: https://stackoverflow.com/questions/49573013/graph-density-with-scatterplot-ggplot2-in-r

Hadley, et al. "Subset rows using column values." *the Grammar of Data Manipulation • dplyr*, RStudio, https://dplyr.tidyverse.org/index.html.

Hadley, et al. "Subset rows using column values." *the Grammar of Data Manipulation • dplyr*, RStudio, https://dplyr.tidyverse.org/index.html.

Zach, "How to Print Multiple Variables on the Same Line in R." Statology, Zach, 13 Apr. 2022, https://www.statology.org/r-print-multiple-variables

TMDB. "Popularity - Getting-Started." Api Docs, https://developers.themoviedb.org/3/getting-started/popularity.