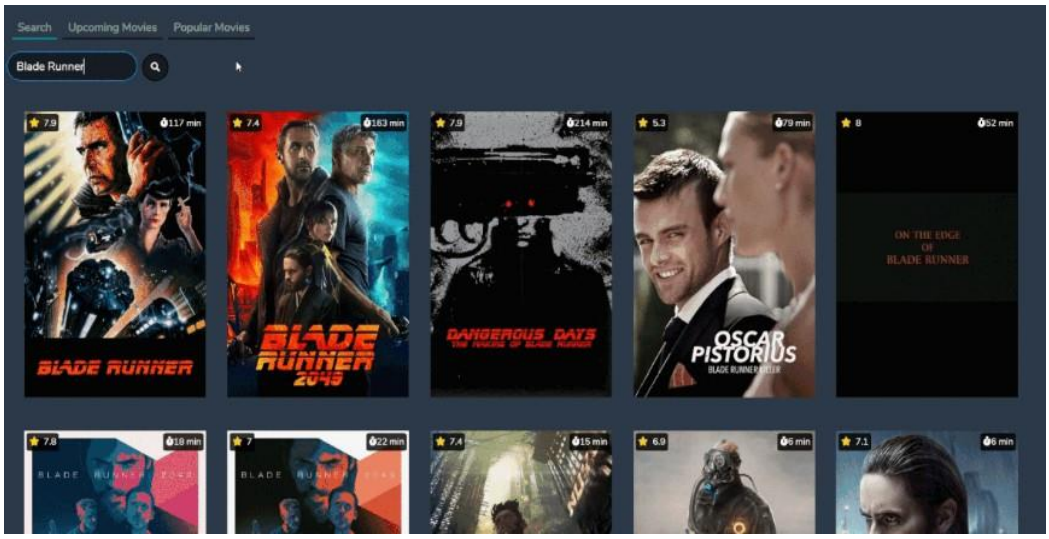
Assessment

Working with Data

KANMANIVISHWAA PARAMASIVAM

Introduction

In general, the film industry has an important role of development and is currently growing at a rapid pace. Even during that challenging market period, the film industry rapidly expanded. The phenomenon occurred again during the 2008 recession. The Movie Database is a film and TV database created by the community. And the database features data from all over the world since 2008. The worldwide box office has been good enough to justify 41.7 billion in 2018, a 2.7 percent growth from the previous year's $40.6 billion and even the second time it has overtaken $40 billion. Revenue from North America and Canada managed to reach an all-time extremely high of $11.9 billion, a 7 percent growth over 2017, while attendance continued to increase by 5 percent. Besides additional categories of filmmakers and demographic details, the database now contains trivia, biographies, and plot summaries.



React App that uses TMDb API to display movie data

TMDb's broad international focus and scope of data are largely unmatched. And the dataset for this whole project was managed to collect from the Movies Database which have all been obtained from the TMDB (The Movie Database) Open API The Movie Database (TMDb) undertook a side project to assist the media community in providing high definition posters and idea art what probably started as a simple image wanting to share community has developed into the most effective user-edited movie databases on the Internet. Early part of 1929, during the Economic Crisis and the Glory Days of Hollywood, an insight related to the consumption of movie tickets started to develop. The purpose of this report is to examine the relationship between features and revenue, where we will be trying to predict revenue from the box office of a movie.

Objectives

Today, film is quite popular in society, and it's growing every year. Every year, there are so many major blockbuster hits releasing, making hundreds of millions of dollars (sometimes over one billion), which are hugely successful. The worldwide box office has been good enough to justify 41.7 billion in 2018, a 2.7 percent growth from the previous year's $40.6 billion and even the second time it has overtaken $40 billion. Revenue from North America and Canada managed to reach an all-time extremely high of $11.9 billion, a 7 percent growth over 2017, while attendance continued to increase by 5 percent. Moreover, for every successful film, there are plenty of others which flop at the box office and are unsatisfying in every case. One of the primary business interests of film studios and their related stakeholders is the prediction of revenue that a new film can yield depending on a few given set of input attributes. The objective was to establish our own mark using an initial data donation from a project called omdb , and we published the first version of the editable database in early 2009.The objective of this project is to evaluate what makes certain movies successful and others not that much, as monitored by worldwide box office revenue.The aim of the report is to examine how popularity, budget, runtime, and other features affect revenue.

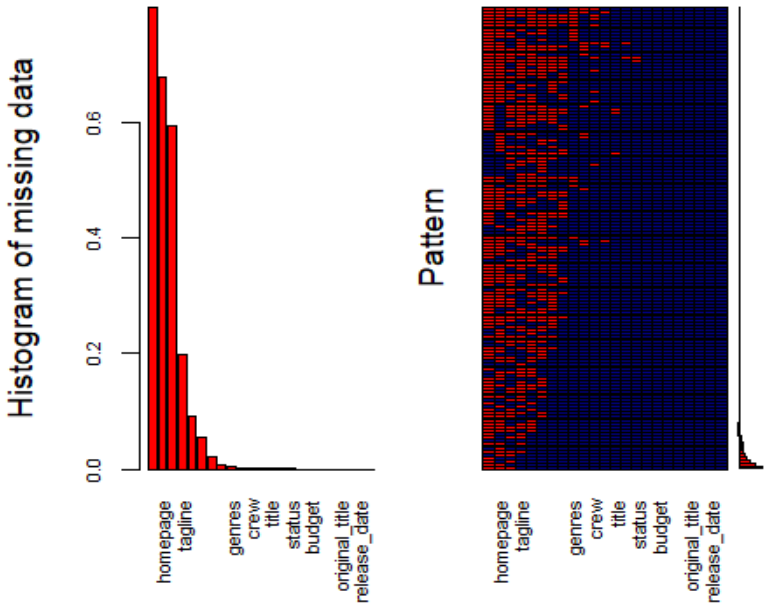
.

Data

The purpose of data analysis on these movies is to predict the revenue for movies which may be extremely useful in industry to know what the critical success factors are. Only their own API also tends to give users access to the data about many additional movies, actors and actresses, and crew members. Data points include. The dataset was reduced to two datasets, the train.csv and the test. csv has 3000 movies whereas test.csv has 4398 movies.

The train set has 3000 observations across 23 variables and the test set has 4398 observations from 22 variables. By binding the rows of the train and test datasets together and they may be combined into a single dataset. on both data sets, we are doing all feature engineering and data preparation and by taking glimpse at our train dataset to get a concept of how it appears. Many of these variables (for example, belongs to collection, genres, and cast) appear to be confused. As a result, we will utilize regular expressions to extract the necessary information before incorporating it into our model. Text variables such as belong to collections, genres, crews, casts, and so forth which should be noted that the text structure of those character variables appears to be difficult. Budget, income, popularity, and

runtime are the numerical variables. The test set is one column smaller since it is lacking the'revenue' column, which is our desired value. The test set also includes a lot more rows, which is problematic.




```
> head(train)
     runtime   budget prod_comp_size   top_prod_comp prod_comp_id main_genre    language    collection top_prod_country
503      104 0.000000 Small producer           Other        10255     Action     English No collection    United States
2035     123 7.505150 Small producer           Other         9195      Drama     English No collection    United States
2967      96 7.602060   Big producer New Line Cinema           12     Action     English No collection    United States
470      113 2.056905 Small producer           Other          441  Adventure     English    Collection    United States
1990     123 6.278754 Small producer           Other         3652     Action Non-English    Collection            Other
1540      98 7.000000 Small producer           Other          491      Drama     English No collection    United States
     tagline_presence homepage_presence year_released quarter_released month_released week_released weekday_released
503        No tagline       No homepage      3.304275                2          April            15         Thursday
2035          Tagline       No homepage      3.300161                3           July            27           Friday
2967          Tagline       No homepage      3.300378                3           July            31         Thursday
470           Tagline       No homepage      3.297979                2           June            25        Wednesday
1990       No tagline       No homepage      3.302114                3           July            26           Friday
1540          Tagline       No homepage      3.303412                2           June            25           Friday
     number_of_keywords number_of_prod_companies number_of_genres title_length tagline_length number_of_cast
503                   6                        3                3           9             19             13
2035                  6                        1                4          10             44             16
2967                  5                        3                6           5             35             10
470                   5                        2                5          23             30             11
1990                  3                        1                3           6             12              9
1540                  7                        5                1          13             36             12
     number_of_crew female_cast male_cast female_crew male_crew  revenue
503              39           3         6           4         7 5.500337
2035             10           3        10           1         7 8.181948
2967             16           2         7           2        10 7.943693
470               7           2         6           1         6 8.061090
1990             10           2         3           0         4 6.770852
1540             31           3         5           5        11 6.245328
>
```

The two greatest NA ratio variables are belongs to collection and homepage, both of which are greater than 60%. Budget, which assumes the most significant factors, similarly has a NA value of 27.07 percent. Other taglines, keywords, and production_ companies have a smaller NA value ratio, with less than 20%. Because there are numerous publicly accessible datasets on the internet on budget numbers, movie descriptions, and so on, and the competition regulations

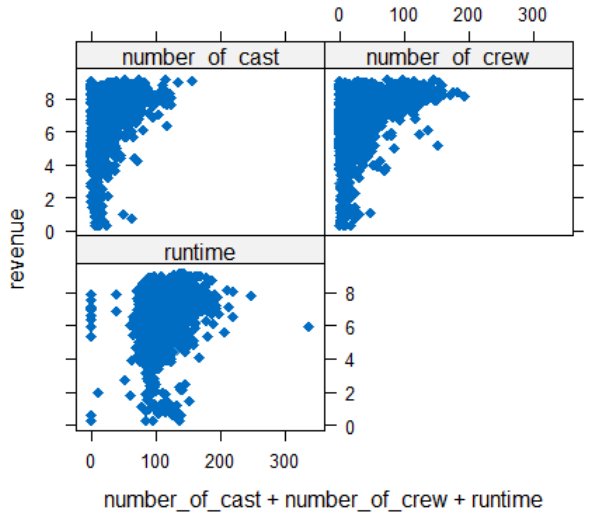
allow for the use of any publicly available publically dataset that is available prior to the movie's release data.

Results

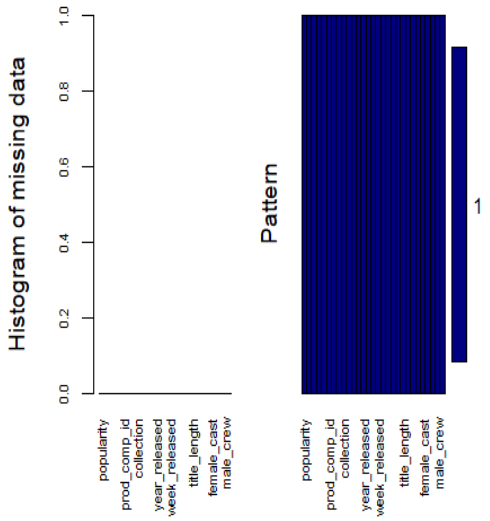
MISSING DATA HANDLING

There are several packages available for imputing missing data, the most prominent of which being Hmisc, missForest, Amelia, and mice. Here, the recommended imputation package is mics. The mice package in R supports with the imputation of missing values with acceptable data values. These potential values are chosen from a distribution customized to each missing datapoint.

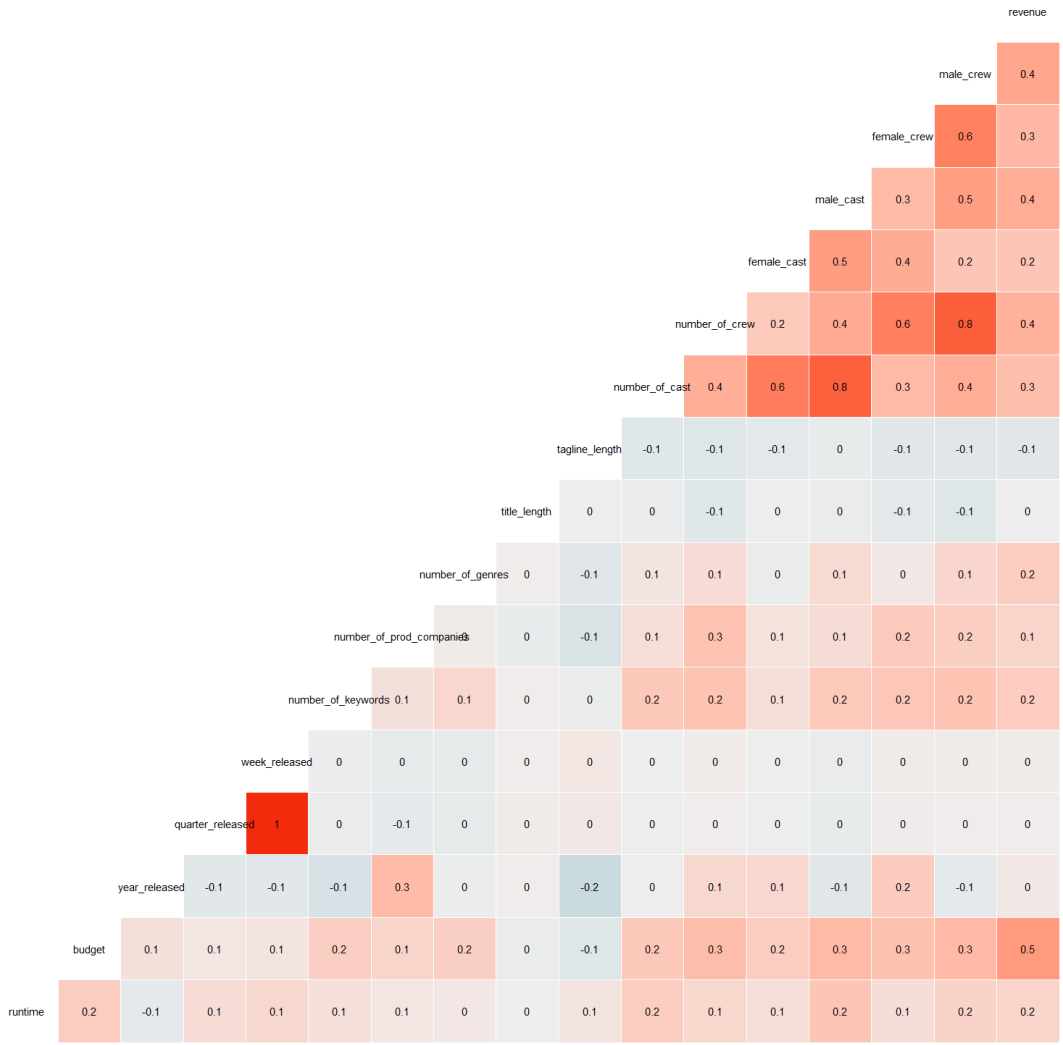
The mice package has a useful function called md.pattern() that can help you analyse the patterns of missing data. The mice() function handles the imputation process. A few words about the parameters: The value m=5 denotes the number of imputed datasets. The default number is five. The imputation technique is denoted by meth= 'pmm'. As an imputation approach, we are employing predictive mean matching in this example. Other imputation techniques are available; enter methods(mice) to get a list of possible imputation methods. The distribution plot of provided original values and imputed missing values in imputed columns is shown as follows.



a)

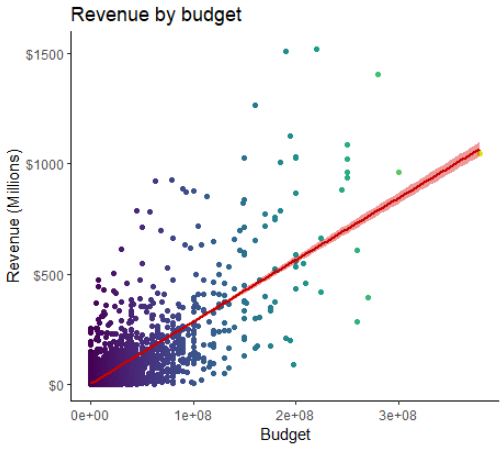


b)

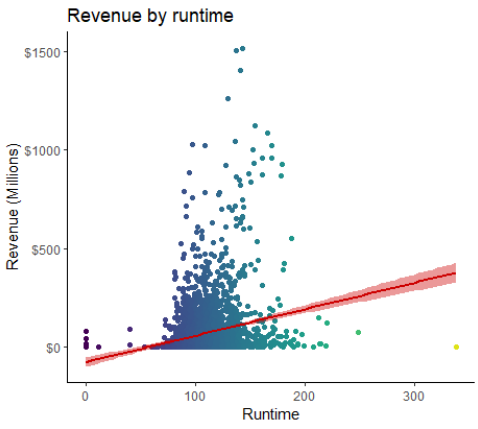


Comparing the distributions of original and imputed data using several relevant charts. The fitting shape indicates that the imputed values are "plausible values."

EXPLORATORY DATA ANALYSIS



a)



b)

LINEAR REGRESSION MODEL

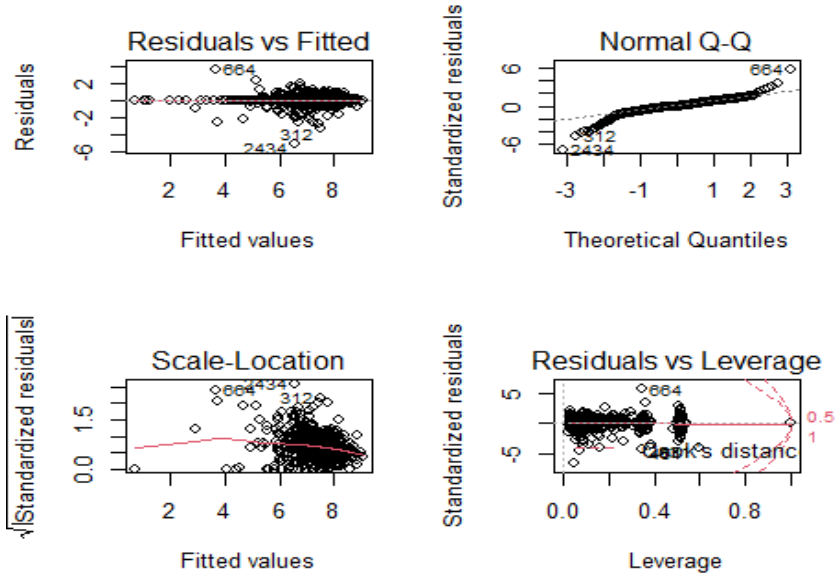
The lm() function in R is used to generate a regression model given a formula in the form of Y~X+X2 where the multiple linear regression model is used to extract the $resid variable from your new model to study the residuals. Residuals are the differences between the prediction and the actual outcomes, which must be analysed in order to enhance your regression model. The results of the linear regression model are reported via the summary function.

The Akaike information criterion (AIC) is a metric for comparing the fit of various regression models. It is calculated by the following equation: AIC = 2K – 2ln(L). The predictor variables that will be used in each model are as follows:

Model selection based on AICc:

	K	AICc	Delta_AICc	AICcWt	Cum.Wt	LL
disp.qsec	17	2686.23	0.00	1	1	-1325.77
disp.hp.wt.qsec	9	2760.48	74.25	0	1	-1371.14
disp.wt	424	3243.52	557.30	0	1	-818.39

Place the models in a list and use the aictab () function to calculate the AIC for each model. The model with the lowest AIC value is always displayed first. According to the results, the next model has the lowest AIC value, The model for this linear regression is lm_model3, and using AIC we find that is optimal.The unexplained variation is referred to as residuals. They are not identical to model error, but they are determined from it, therefore a bias in the residuals would also suggest a bias in the error. The most essential thing to notice is that the red lines showing the mean of the residuals are all horizontal and centred around zero. This indicates that there are no outliers or biases in the data that would invalidate a linear regression.In the top right Normal Q-Qplot, we can see that the real residuals from our model form a nearly perfect one-to-one line with the conceptual residuals from a perfect role model. Based on these residuals, conclusion of our model fulfils the homoscedasticity assumption.



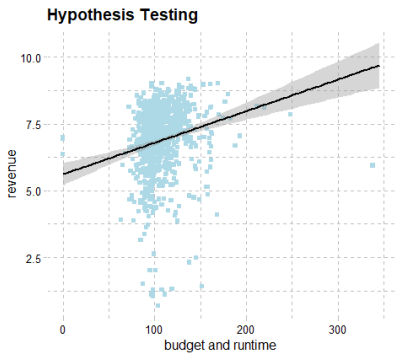
HYPOTHESIS TESTING

A hypothesis test is being conducted to determine whether runtime and budget have an impact on revenue. The revenue can be derived as follows using both variables.

According to the preceding relationship, the Null Hypothesis (i.e., revenue is unaffected by budget and runtime) and the Alternate Hypothesis (Ha) (i.e., revenue is affected by budget and runtime). I'm setting the significance level at 5% and performing the hypothesis test with the Anova test. The P value, which runs from 0 to 1, reflects the chance that the null hypothesis will occur. The null hypothesis can be rejected if the P value is less than the significance level (0.05). If the P value is greater than the significance threshold, the null hypothesis can be rejected, implying that revenue is affected by budget and runtime.

	Df	f value	p value
runtime	1	26.39	3.42e-07
budget	1	242.54	< 2e-16
Residuals	897		

a)



b)

MULTICOMPARISON MODELS

SUMMARY FOR LINEAR REGRESSION MODEL:

```
Call:
lm(formula = revenue ~ budget + runtime + number_of_cast + year_released +
    week_released + prod_comp_id + quarter_released + number_of_crew +
    title_length + tagline_length + number_of_keywords + female_cast +
    female_crew + male_crew + male_cast + number_of_keywords +
    number_of_prod_companies + number_of_genres, data = test_step)

Residuals:
    Min      1Q  Median      3Q     Max
-5.0566 -0.0886  0.0000  0.1691  3.6568

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)             -1.504e+02  5.085e+01  -2.958 0.003245 **
budget                   8.375e-02  1.421e-02   5.893 7.16e-09 ***
runtime                 -5.543e-04  1.766e-03  -0.314 0.753700
number_of_cast          -6.825e-03  4.269e-03  -1.599 0.110500
year_released            4.668e+01  1.539e+01   3.034 0.002549 **
week_released            2.406e-02  1.000e-02   2.405 0.016539 *
prod_comp_id1            3.736e+00  9.918e-01   3.767 0.000186 ***
prod_comp_id10           4.251e-01  1.174e+00   0.362 0.717413
prod_comp_id10000        1.262e+00  8.448e-01   1.494 0.135935
prod_comp_id10105        2.791e+00  1.182e+00   2.361 0.018649 *
prod_comp_id1011         1.933e+00  1.176e+00   1.644 0.100865
prod_comp_id10146        2.700e+00  1.019e+00   2.649 0.008335 **
prod_comp_id10163        2.542e+00  1.021e+00   2.491 0.013075 *
prod_comp_id10167        1.960e+00  1.187e+00   1.652 0.099275 .
prod_comp_id10253        2.196e+00  1.210e+00   1.814 0.070252 .
prod_comp_id10308        2.433e+00  1.185e+00   2.053 0.040635 *
```

```
Residual standard error: 0.8252 on 477 degrees of freedom
Multiple R-squared:  0.7866,    Adjusted R-squared:  0.5979
F-statistic: 4.167 on 422 and 477 DF,  p-value: < 2.2e-16
```

SUMMARY FOR RANDOM FOREST:

```
Call:
 randomForest(formula = revenue ~ ., data = my_data_clean, ntree = 501,      replace = TRUE, nodesize = 9, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 501
No. of variables tried at each split: 9

          Mean of squared residuals: 0.8691644
                    % Var explained: 50.81
> |
```

Random Forests, in general, give superior results, perform well on huge datasets, and can deal with missing data by generating estimates for it. They do, however, present a significant issue in that they cannot extrapolate beyond previously viewed data. In a moment, we'll go further into these problems. Random Forest is better dealing with missing data than Linear Regression model. The Mean of Squared residuals in Random Forest is having higher value than the the Mean of Squared residuals in Linear Regression . So The Random is preferable for this dataset .

PREDICTION

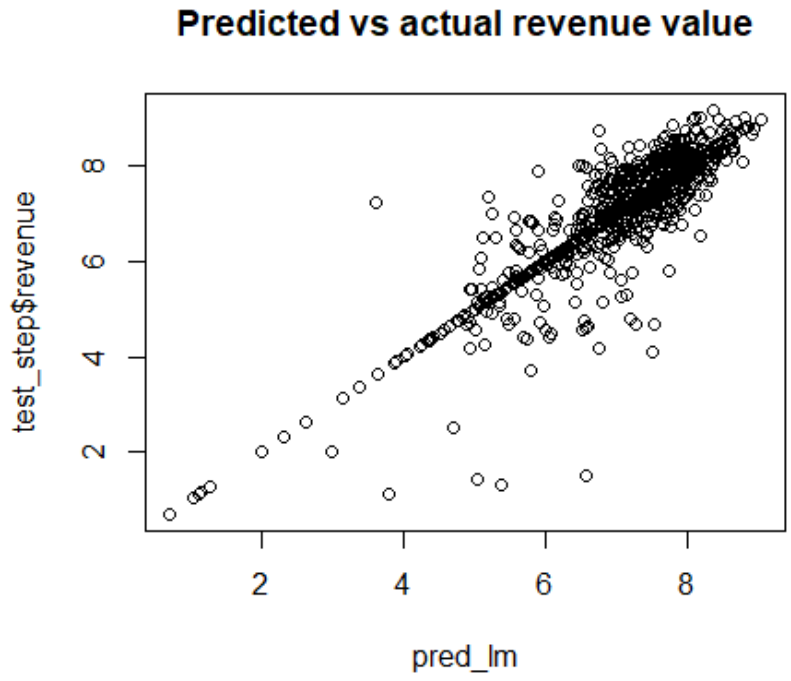
The linear model can be used to create predictions either if the null hypothesis, which states that there is no relationship between our variables, is rejected or if the model works well with our data. Our model's output is termed to as summary ().

SUMMARY OF PREDICTION OF LINEAR REGRESSION MODEL:

summary(pred_lm)

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.699	6.417	7.176	6.949	7.751	9.041

Here the graph is plotted for the predicted value and actual value using prediction of linear regression model. Here the actual and predicted values are shown



Limitations:

The scope of the project is reduced to exploratory data analysis utilising plots and charts. To draw any findings, statistical analysis must be performed.There might be more elements influencing gross income that have not been explored in the project. Modeling can be performed, and the data set can be segmented into train and test data sets to validate the model. Assigning more variable or attribute must be reduced and categorical or factorial data also should be reduced.

Conclusion:

It is not a 100% sure resolution that this formula will work, but it does show us that we have a strong probability of producing big profits if we share comparable traits. When we release a film with these traits, it raises people's anticipation for the film. This is only one example of an influential aspect that might lead to differing outcomes; there are many more that must be addressed. In this Project, the NA value of categorical data can not be imputed so the data had been replaced for processing the regression and Accuracy of the model. This is the most vital reason for the future project's imputation of missing value.

APPENDIX:

```
pacman::p_load(tidyverse,
               lubridate,
               ggthemes,
               plotly,
               corrplot,
               gridExtra,
               VIM,
               viridis,
               readr,
               tidyr,
               infotheo,
               outliers,
               Hmisc,
               randomForest,
               modelr,
               caret,
               tidymodels,
```

```
        DMwR2,
        ggplot2,
        dplyr,
        broom,
        datarium,
        DataExplorer,
        ggpubr,
        Amelia,
        rpart,
        rpart.plot,
        GGally,
        AICcmodavg,
        xgboost,
        mice)
full_data  <- bind_rows(train, test)
#missing data status:
DataExplorer::profile_missing(full_data)
#plot the missing data status
library(VIM)
aggr_plot <- aggr(full_data, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE, labels=names(data), cex.axis=.7, gap=3, ylab=c("Histogram of missing data","Pattern"))
#train data
glimpse(train)
#head of train data
head(train)
```

#Data Visualisation

```
# Budget
ggplot(train, aes(x = budget, y = revenue, color = budget)) +
  geom_point() +
```

```
  # scale_color_gradient(low = "grey10", high = "grey75") +
  scale_color_viridis(begin = 0, end = .95, option = 'D') +
  geom_smooth(method = 'lm', color = 'red3', fill = 'red3') +
  scale_y_continuous(breaks = c(0, 500000000, 1000000000, 1500000000),
                     labels = c('$0', '$500', '$1000', '$1500')) +
  theme_classic() +
  theme(legend.position = 'none') +
  labs(title = 'Revenue by budget', x = 'Budget', y = 'Revenue (Millions)')
# Runtime
ggplot(train, aes(x = runtime, y = revenue, color = runtime)) +
  geom_point() +
  # scale_color_gradient(low = "grey10", high = "grey75") +
  scale_color_viridis(begin = 0, end = .95, option = 'D') +
  geom_smooth(method = 'lm', color = 'red3', fill = 'red3') +
  scale_y_continuous(breaks = c(0, 500000000, 1000000000, 1500000000),
                     labels = c('$0', '$500', '$1000', '$1500')) +
  theme_classic() +
  theme(legend.position = 'none') +
  labs(title = 'Revenue by runtime', x = 'Runtime', y = 'Revenue (Millions)')
# Collection
train$collection_name <- str_extract(train$belongs_to_collection,
                                     pattern = "(?<=name\\'\\:\\s{1}\\').+(?=\\'\\,\\s{1}\\'poster)")
train %>%
  group_by(collection_name) %>%
  summarise(movie_count = n()) %>%
  arrange(desc(movie_count)) %>%
  filter(!is.na(collection_name)) %>%
  head(10)
train$collection[!is.na(train$belongs_to_collection)] <- 'Collection'
```

```
train$collection[is.na(train$belongs_to_collection)] <- 'No collection'

# Main genre

genres_matching_point <- "Comedy|Horror|Action|Drama|Documentary|Science Fiction|
        Crime|Fantasy|Thriller|Animation|Adventure|Mystery|War|Romance|Music|
        Family|Western|History|TV Movie|Foreign"

train$main_genre <- str_extract(train$genres, genres_matching_point)

# Production company id

train$prod_comp_id <- str_extract(train$production_companies,
                        pattern = "([0-9]+)")

#Top production companies

train$prod_comp_name <- gsub('(^\\[\\{\'name\'\\:\\s\'|\'\\,\\s\'id.*)', '',
                    train$production_companies)

train %>%
  group_by(prod_comp_name) %>%
  summarise(movie_count = n()) %>%
  arrange(desc(movie_count)) %>%
  filter(!is.na(prod_comp_name)) %>%
  head(10)

train$top_prod_comp[train$prod_comp_name=='Universal Pictures'] <- 'Universal Pictures'

train$top_prod_comp[train$prod_comp_name=='Paramount Pictures'] <- 'Paramount Pictures'

train$top_prod_comp[train$prod_comp_name=='Twentieth Century Fox Film Corporation'] <-
'Twentieth Century Fox Film Corporation'

train$top_prod_comp[train$prod_comp_name=='Columbia Pictures'] <- 'Columbia Pictures'

train$top_prod_comp[train$prod_comp_name=='New Line Cinema'] <- 'New Line Cinema'

train$top_prod_comp[train$prod_comp_name=='Warner Bros.'] <- 'Warner Bros.'

train$top_prod_comp[train$prod_comp_name=='Walt Disney Pictures'] <- 'Walt Disney Pictures'

train$top_prod_comp[is.na(train$top_prod_comp)] <- 'Other'

#Production company size

train$prod_comp_size[train$prod_comp_name=='Universal Pictures'] <- 'Big producer'
```

```
train$prod_comp_size[train$prod_comp_name=='Paramount Pictures'] <- 'Big producer'
train$prod_comp_size[train$prod_comp_name=='Twentieth Century Fox Film Corporation'] <- 'Big producer'
train$prod_comp_size[train$prod_comp_name=='Columbia Pictures'] <- 'Big producer'
train$prod_comp_size[train$prod_comp_name=='New Line Cinema'] <- 'Big producer'
train$prod_comp_size[train$prod_comp_name=='Warner Bros.'] <- 'Big producer'
train$prod_comp_size[train$prod_comp_name=='Walt Disney Pictures'] <- 'Big producer'
train$prod_comp_size[is.na(train$prod_comp_size)] <- 'Small producer'
#Top production countries
train$prod_country <- str_extract(string = train$production_countries,
                                  pattern = "[:upper:]+")
train %>%
  group_by(prod_country) %>%
  summarise(movie_count = n()) %>%
  arrange(desc(movie_count)) %>%
  filter(!is.na(prod_country)) %>%
  head(10)
train$top_prod_country[train$prod_country=='US'] <- 'United States'
train$top_prod_country[train$prod_country=='GB'] <- 'Great Britain'
train$top_prod_country[train$prod_country=='FR'] <- 'France'
train$top_prod_country[is.na(train$top_prod_country)] <- 'Other'
# IMDB id
train$imdb_id_2 <- str_extract(train$imdb_id, '[0-9]+')
#Language
train %>%
  group_by(original_language) %>%
  summarise(movie_count = n()) %>%
  arrange(desc(movie_count)) %>%
  filter(!is.na(original_language)) %>%
```

```
  head(10)
train$language[train$original_language=='en'] <- 'English'
train$language[is.na(train$language)] <- 'Non-English'
# Year, quarter, month, week, and weekday released
which(is.na(full_data$release_date))
full_data[3829, c('title', 'runtime')]
full_data$release_date[3829] <- '3/20/01'
train$release_date_mod <- parse_date_time2(train$release_date, "mdy",
                                           cutoff_2000 = 20)
# Create year, quarter, month, week, and weekday released using the LUBRIDATE package.
train$year_released <- ymd(train$release_date_mod) %>%
  lubridate::year()  # Grab year.
train$quarter_released <- ymd(train$release_date_mod) %>%
  lubridate::quarter()  # Grab quarter.
train$month_released <- ymd(train$release_date_mod) %>%
  lubridate::month(label = TRUE, abbr = FALSE)  # Grab month.
train$week_released <- ymd(train$release_date_mod) %>%
  lubridate::week()  # Grab week.
train$weekday_released <- ymd(train$release_date_mod) %>%
  lubridate::wday(label = TRUE, abbr = FALSE)  # Grab weekday.
#Tagline presence
train$tagline_presence[is.na(train$tagline)] <- 'No tagline'
train$tagline_presence[is.na(train$tagline_presence)] <- 'Tagline'
#Homepage presence
train$homepage_presence[is.na(train$homepage)] <- 'No homepage'
train$homepage_presence[is.na(train$homepage_presence)] <- 'Homepage'
#Gender of cast & crew
# Total cast count and by gender
train$number_of_cast <- str_count(train$cast, 'name')
```

```
train$female_cast <- str_count(train$cast, ('gender\'\\:\\s1'))

train$male_cast <- str_count(train$cast, ('gender\'\\:\\s2'))

train$unspecified_cast <- str_count(train$cast, ('gender\'\\:\\s0'))

# Total crew count and by gender

train$number_of_crew <- str_count(train$crew, 'name')

train$female_crew <- str_count(train$crew, ('gender\'\\:\\s1'))

train$male_crew <- str_count(train$crew, ('gender\'\\:\\s2'))

train$unspecified_crew <- str_count(train$crew, ('gender\'\\:\\s0'))

#Number of (1) genres, (2) production companies, (3) production countries, (4) spoken languages, and (5) keywords.

train$number_of_genres <- str_count(train$genres, 'name')

train$number_of_prod_companies <- str_count(train$production_companies, 'name')

train$number_of_prod_countries <- str_count(train$production_countries, 'name')

train$number_of_spoken_languages <- str_count(train$spoken_languages, 'name')

train$number_of_keywords <- str_count(train$Keywords, 'name')

#Length of (1) title_length, (2) overview_length, and (3) tagline_length by extracting the lengths of the strings of the variables.

train$title_length <- str_length(train$title)

train$tagline_length <- str_length(train$tagline)

train$overview_length <- str_length(train$overview)
```

#Data Wrangling

```
# Subsetting the data


my_data_subset <- subset(train,

                         select = c(popularity, runtime, budget, prod_comp_size,

                                    top_prod_comp, prod_comp_id, main_genre, language, collection,

                                    top_prod_country, tagline_presence, homepage_presence,

                                    year_released, quarter_released, month_released, week_released,
```

```
                    weekday_released, number_of_keywords, number_of_prod_companies,
                    number_of_genres, title_length, tagline_length, number_of_cast,
                    number_of_crew, female_cast, male_cast, female_crew, male_crew,
                    # number_of_prod_countries, number_of_spoken_languages,
                    # imdb_id_2, overview_length, unspecified_cast, unspecified_crew,
                    revenue))
my_data_subset <- mutate(my_data_subset,
                    budget = log10(budget + 1),
                    year_released = log10(year_released),
                    popularity = log10(popularity + 1),
                    revenue = log10(revenue + 1))
```

#Imputation (mice)

```
set.seed(12345)
# pattern of Data
md.pattern(my_data_subset)
# Plot themissing values in our data set.
aggr_plot <- aggr(my_data_subset, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE, labels=names(data), cex.axis=.7, gap=3, ylab=c("Histogram of missing data","Pattern"))
#mice to impute
mice_my_data_subset <- mice(my_data_subset,m=5,maxit = 50,method = "pmm",seed = 500)
# distribution plot of given original values and imputed missing values in imputed columns
xyplot(mice_my_data_subset,revenue ~ number_of_cast+number_of_crew+runtime,pch=18,cex=1)
xyplot
#Complete the mice imputation
newDATA <- complete(mice_my_data_subset, 1)
# Treating missing values of categorical data in dataset
newDATA$prod_comp_id[is.na(my_data_subset$prod_comp_id)] <- 10000
newDATA$main_genre[is.na(my_data_subset$main_genre)] <- "Drama"
```

```
#missing data status after imputation

DataExplorer::profile_missing(newDATA)

#plot for the missing data status after imputation

aggr_plot <- aggr(newDATA, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE, labels=names(data), cex.axis=.7, gap=3, ylab=c("Histogram of missing data","Pattern"))
```

#Data cleaning

```
my_data_clean <- newDATA %>%

  mutate(revenue)
```

#Missing values

```
colSums(prop.table(is.na(my_data_clean)))

#Data modeling

library(rsample)

set.seed(100)

#Splitter

index <- initial_split(my_data_clean %>%

                         select(-popularity),prop = 0.70)

train <- training(index)

test <- testing(index)

test_step <- testing(index)

prop.table(table(train$revenue))

prop.table(table(test$revenue))

prop.table(table(test_step$revenue))

 #summary

summary(test_step)

#Plot

GGally::ggcorr(test_step %>%

                 select_if(is.numeric),label = T,legend.position = "none")
```

#Linear regression

```
lm_model1 <- lm( revenue ~ budget+runtime+ number_of_cast  + year_released + week_released
+ quarter_released +

              number_of_genres,data = test_step)


lm_model2 <- lm( revenue ~ budget+runtime+ number_of_cast  + year_released + week_released
+ quarter_released + number_of_crew

          +title_length  +  tagline_length  +  number_of_keywords  +  female_cast  +  female_crew
+male_crew +male_cast

          +  number_of_genres,data = test_step)


lm_model3 <- lm( revenue ~ budget+runtime+ number_of_cast  + year_released + week_released
+prod_comp_id + quarter_released + number_of_crew

          +title_length  +  tagline_length  +  number_of_keywords  +  female_cast  +  female_crew
+male_crew +male_cast + number_of_keywords + number_of_prod_companies

           +  number_of_genres,data = test_step)

summary(lm_model1)

summary(lm_model2)

summary(lm_model3)
```

#AIC

```
#define list of models

models <- list(lm_model1, lm_model2, lm_model3)

#specify model names

mod.names <- c('disp.hp.wt.qsec', 'disp.qsec', 'disp.wt')

#calculate AIC of each model

aictab(cand.set = models, modnames = mod.names)

#Make predictions on the test set using the linear regression model.

pred_lm = predict(lm_model3, test_step)

summary(pred_lm)

summary(lm_model3)
```

```
plot(pred_lm, test_step$revenue, main="Predicted vs actual revenue value")
#hypothesis testing
cor.test(test_step$runtime,test_step$revenue)
cor.test(test_step$budget,test_step$revenue)
hypoout=aov(test_step$revenue~test_step$runtime+test_step$budget)
summary(hypoout)
confint(hypoout)


ggplot(test_step,aes(x=runtime,y=revenue))+
  geom_point(colour="lightblue",shape=15)+
  geom_smooth(aes(x=budget+runtime,y=revenue),method=lm,colour="black")+
  labs(title="Hypothesis Testing",
       x="budget and runtime",y="revenue")+
  theme_pander()+
  theme(axis.title=element_text())
#Multicomparison
#random forest
set.seed(222)
rf_model <- randomForest(revenue ~ .,
                         my_data_clean,
                         ntree = 501,
                         replace = TRUE,
                         nodesize = 9,
                         importance = TRUE);

print(rf_model)
plot(rf_model)
```