Multiple Linear Regression

Waheeb Algabri

Introduction

The aim of this blog-2 is to explore, model and make predictions based on the Movies data set. This data is comprised of 651 randomly sampled movies produced and released before 2016. The **IMDB** and **Rotten Tomatoes** scores for each movie is provided, in addition to several other parameters like **genre**, **oscar nominations**, **runtime** and so on. A detailed description of the data and it's various attributes can be found here.

We first formulate our research question and then explore the data set for relevant trends and patterns. Following this, we build a ***Multiple Linear Regression*** model to answer our question and use it to make meaningful predictions.

Setup

Load Packages

Before getting started, make sure to load the relevant libraries such as `tidyverse` for use in this blog. The tidyverse includes `ggplot2` for plotting and visualization, and `dplyr` for transforming and processing raw data.

```
library(tidyverse)
library(reshape2)
library(knitr)
```

Load Data

```
load("/Users/waheebalgabri/Downloads/movies.Rdata")
attach(movies)
```

The data is present in the `.Rdata` format, which can be loaded using the `load()` function. It can be downloaded from here. We use `attach()` to conveniently use the column names without any operators or indexing.

Understanding the Data

In this section we aim to understand if random sampling was used and if the manner of data collection can be trusted to yield reasonably generalizable results. Further, we also discuss random assignment and causality.

The first step is to determine whether our study is observational or experimental. Here, the data is clearly collected in a manner that does not interfere with how it arises. It is merely observed and there is no **Random Assignment** .

Hence, this is an **Observational** study. Since past data is used, it can be categorized as a **Retrospective Observational Study**.

Generalizability

`Generalizability` is defined as the degree to which the results of a study based on a sample can be said to represent the results that would be obtained from the entire population from which the sample was drawn. In other words, generalizability depends on the degree to which the particular sample in question can be said to be representative of the population.

To understand our sample better, let us take a look at the range of years in which the movies in our data were released. The column ***thtr_rel_year*** is the one we need.

```
# Range of release years
range(thtr_rel_year)
```

```
## [1] 1970 2014
```

We observe that the movies are spread across 44 years and represent a wide range of genres.

However, in this time period, nearly **44,000** movies were produced. Our sample contains merely 651 of those - it is representative of only **1%** of the population.

Therefore, it is unreasonable to assume that this data will produce fully generalizable results. It is mentioned in the data description that random sampling was used. This means that our sample was chosen such that each observation had an equal chance of being selected. This can be observed by exploring the data, and is true in our case.

Hence, we can conclude that random sampling was used and this renders some amount of generalization acceptable

Please note that year of release was just an example to show how we can study generalizability.

Random Assignment

`Random assignment` occurs only in **experimental** settings, where subjects are being assigned to various treatments. Through a random assignment, we ensure that these different characteristics are represented equally in the treatment and control groups. In other words, random assignment allows us to make causal conclusions based on the study.

As already mentioned, our data is obtained in an observational manner and no there is no random assignment. Based on observational studies, we can only establish an association i.e. correlation between the explanatory and the response variables.

Hence, we cannot infer any causal relationships from this data, within limits of the methods used.

Formulating the Research Question

Formulating a proper research question is fundamental because all our analysis is developed to answer it. The aim of this blog is to answer the question given below : ***What are the attributes associated with the popularity of a movie among audiences, as measured by IMDB ratings?*** We have two metrics in the data that reflect popularity among audiences - the audience score from **Rotten Tomatoes** and **IMDB Ratings**. For the purpose of this blog, **IMDB Ratings** is the chosen metric, since their usage of weighted scores make their ratings more representative and accurate. However, the two are very similar for the most part and the discrepancy is limited.

The purpose here is to observe the **linear relationship** between the **IMDB Ratings** of movies and different attributes like `genre`, `critics score`, `runtime` and others specified in the data. We also aim to determine the strength of said relationships, if they exist.

This will ultimately help us to build a **linear regression model** to explore the effects of said associations and make meaningful **predictions** using the same.

It is to be noted that in no way are we attempting to establish any causal relationships here. We are not trying to determine if an attribute **causes** an increase in popularity, only if they are **correlated** with it. Also note that **correlation** only refers to a **linear association** between variables, which is what we explore here. For convenience, the terms **association** and **correlation** are used interchangeably.

Data Cleaning

Before working with our data, we must clean and transform it to a form which is convenient an meaningful to work with, if it is necessary.

Let us take a look at our data using `str()` and `head()`

```
str(movies)
```

```
## tibble [651 x 32] (S3: tbl_df/tbl/data.frame)
##  $ title           : chr [1:651] "Filly Brown" "The Dish" "Waiting for Guffman" "The Age of Innocenc
##  $ title_type      : Factor w/ 3 levels "Documentary",..: 2 2 2 2 2 1 2 2 1 2 ...
##  $ genre           : Factor w/ 11 levels "Action & Adventure",..: 6 6 4 6 7 5 6 6 5 6 ...
##  $ runtime         : num [1:651] 80 101 84 139 90 78 142 93 88 119 ...
##  $ mpaa_rating     : Factor w/ 6 levels "G","NC-17","PG",..: 5 4 5 3 5 6 4 5 6 6 ...
##  $ studio          : Factor w/ 211 levels "20th Century Fox",..: 91 202 167 34 13 163 147 118 88 84
##  $ thtr_rel_year   : num [1:651] 2013 2001 1996 1993 2004 ...
##  $ thtr_rel_month  : num [1:651] 4 3 8 10 9 1 1 11 9 3 ...
##  $ thtr_rel_day    : num [1:651] 19 14 21 1 10 15 1 8 7 2 ...
##  $ dvd_rel_year    : num [1:651] 2013 2001 2001 2001 2005 ...
##  $ dvd_rel_month   : num [1:651] 7 8 8 11 4 4 2 3 1 8 ...
##  $ dvd_rel_day     : num [1:651] 30 28 21 6 19 20 18 2 21 14 ...
##  $ imdb_rating     : num [1:651] 5.5 7.3 7.6 7.2 5.1 7.8 7.2 5.5 7.5 6.6 ...
##  $ imdb_num_votes  : int [1:651] 899 12285 22381 35096 2386 333 5016 2272 880 12496 ...
##  $ critics_rating  : Factor w/ 3 levels "Certified Fresh",..: 3 1 1 1 3 2 3 3 2 1 ...
##  $ critics_score   : num [1:651] 45 96 91 80 33 91 57 17 90 83 ...
##  $ audience_rating : Factor w/ 2 levels "Spilled","Upright": 2 2 2 2 1 2 2 1 2 2 ...
##  $ audience_score  : num [1:651] 73 81 91 76 27 86 76 47 89 66 ...
##  $ best_pic_nom    : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ best_pic_win    : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ best_actor_win  : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 2 1 1 ...
##  $ best_actress_win: Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ best_dir_win    : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 1 1 1 ...
##  $ top200_box      : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ director        : chr [1:651] "Michael D. Olmos" "Rob Sitch" "Christopher Guest" "Martin Scorsese
##  $ actor1          : chr [1:651] "Gina Rodriguez" "Sam Neill" "Christopher Guest" "Daniel Day-Lewis"
##  $ actor2          : chr [1:651] "Jenni Rivera" "Kevin Harrington" "Catherine O'Hara" "Michelle Pfei
##  $ actor3          : chr [1:651] "Lou Diamond Phillips" "Patrick Warburton" "Parker Posey" "Winona Ry
##  $ actor4          : chr [1:651] "Emilio Rivera" "Tom Long" "Eugene Levy" "Richard E. Grant" ...
##  $ actor5          : chr [1:651] "Joseph Julian Soria" "Genevieve Mooy" "Bob Balaban" "Alec McCowen"
##  $ imdb_url        : chr [1:651] "http://www.imdb.com/title/tt1869425/" "http://www.imdb.com/title/t
##  $ rt_url          : chr [1:651] "//www.rottentomatoes.com/m/filly_brown_2012/" "//www.rottentomatoes
```

title	title_type	genre	runtime	mpaa_rating	studio	thtr_rel_year	thtr_rel_month	thtr_rel_day	dvd_rel_year	dvd_rel_month	dvd_rel_day	imdb_rating	imdb_num_votes	critics_rating	critics_score	audience_rating	audience_score	best_pic_nom	best_pic_win	best_actor_win	best_actress_win	best_dir_win	top200_box	director	actor1	actor2	actor3	actor4	actor5	imdb_url	rt_url
Filly Brown	Feature Film	Drama	80	R	Indomina Media Inc.	2013	19	2013	7	30	5.5	899	Rotten	45	Upright	73	no	no	no	no	no	no	Michael D. Olmos	Gina Rodriguez	Jenni Rivera	Lou Diamond Phillips	Emilio Rivera	Joseph Julian Soria	http://www.imdb.com/title/tt1869425/	//www.rottentomatoes.com/m/filly_brown_2012/
The Dish	Feature Film	Drama	101	PG-13	Warner Bros. Pictures	2001	14	2008	8	28	7.3	12227	Certified Fresh	95	Upright	81	no	no	no	no	no	no	Rob Sitch	Sam Neill	Kevin Harrington	Patrick Warburton	Tom Long	Genevieve Mooy	http://www.imdb.com/title/tt0205873/	//www.rottentomatoes.com/m/dish/
Waiting for Guffman	Feature Film	Comedy	84	R	Sony Pictures Classics	1996	21	2008	8	21	7.6	22381	Certified Fresh	91	Upright	91	no	no	no	no	no	no	Christopher Guest	Christopher Guest	Catherine O'Hara	Parker Posey	Eugene Levy	Bob Balaban	http://www.imdb.com/title/tt0118111/	//www.rottentomatoes.com/m/waiting_for_guffman/
The Age of Innocence	Feature Film	Drama	139	PG	Columbia Pictures	1993	1	2001	11	6	7.2	35005	Certified Fresh	86	Upright	76	no	no	yes	no	yes	no	Martin Scorsese	Daniel Day-Lewis	Michelle Pfeiffer	Winona Ryder	Richard E. Grant	Alec McCowen	http://www.imdb.com/title/tt0106226/	//www.rottentomatoes.com/m/age_of_innocence/
Malevolence	Feature Film	Horror	90	R	Anchor Bay Entertainment	2004	10	2005	5	19	5.1	2388	Rotten	33	Spilled	27	no	no	no	no	no	no	Stevan Mena	Samantha Dawn	Brad Johnson	Brando Johnson	Heather Magee	Richard Glover	http://www.imdb.com/title/tt0388230/	//www.rottentomatoes.com/m/malevolence/

title	title_type	genre	runtime	mpaa_rating	studio	thtr_rel_year	thtr_rel_month	thtr_rel_day	dvd_rel_year	dvd_rel_month	dvd_rel_day	imdb_rating	imdb_num_votes	critics_rating	critics_score	audience_rating	audience_score	best_pic_nom	best_pic_win	best_actor_win	best_actress_win	best_dir_win	top200_box	director	actor1	actor2	actor3	actor4	actor5	imdb_url	rt_url
Old Partner	Documentary	Documentary	78	Unrated	Global Media Group	2009	9	15	2010	9	20	7.8	333	Fresh	91	Upright	86	no	no	no	no	no	no	Chung ryoul Leekyus	Lee Won soon	Moon Sam-soon	NA	NA	NA	http://www.imdb.com/title/tt1334549/	http://www.rottentomatoes.com/partner/

At this stage, we can identify our **dependent variable** the attribute **imdb_rating** which is measured on a `scale of 10` (Note that ***audience_score*** and **critics_score** are measured on a **scale of 100**).

A cursory look at the data tells us that the columns studio, director, urls and list of actors are likely to have too many categories and may not be very relevant to our analysis. Hence, it is prudent to remove them from our data so that we have cleaner data to work with. We also omit any dates involved.

Since the attributes **audience_score** and **audience_rating** are simply the Rotten Tomatoes version of our dependent variable, we drop these as well.

The `select()` command of `dplyr` is used to drop the said columns, using their indices. We use **complete.cases()** to check for any missing values.

```
# Count missing values for each column
missing_count <- movies %>%
  summarise_all(~sum(is.na(.)))
```

```
movies = select(movies,-c(25:32,6:12,17,18))
sum(complete.cases(movies))
```

```
## [1] 650
```

It can be seen that 650 out of 651 observations are complete i.e. have no missing values. Hence there is no need for any additional pre-processing to deal with them.

Let us take a look at the different categorical/factor variables in our data to ensure that there are no mistakes in encoding. We can do this by taking a look at their levels using the `levels()` function.

```
levels(genre)
```

```
##  [1] "Action & Adventure"        "Animation"
##  [3] "Art House & International" "Comedy"
##  [5] "Documentary"               "Drama"
##  [7] "Horror"                    "Musical & Performing Arts"
##  [9] "Mystery & Suspense"        "Other"
## [11] "Science Fiction & Fantasy"
```

```
levels(critics_rating)
```

```
## [1] "Certified Fresh" "Fresh"           "Rotten"
```

```
levels(mpaa_rating)
```

```
## [1] "G"       "NC-17"   "PG"      "PG-13"   "R"       "Unrated"
```

```
levels(title_type)
```

```
## [1] "Documentary"  "Feature Film" "TV Movie"
```

We can observe that all the categories appear to be unique and no changes is encoding appear to be necessary. Note that we have not observed the Oscar categories because they have only 2 levels `yes` and `no`, which were clearly visible when we used **str()** to observe the data.

Hence, in the case of the given data set no major transformation or cleaning appears to be necessary.

However, to simplify our exploratory analysis we will create a subset of the movies data called **oscar.data** which contains the number of award winning movies, actors, directors and Top 200 movies by box office present in the data.

```
oscar.data =apply(movies[c("best_pic_nom", "best_pic_win","best_actor_win",
"best_actress_win", "best_dir_win","top200_box")],2,table)
oscar.data = melt(oscar.data)
names(oscar.data) = c("Decision","Category","Counts")
oscar.data$Decision = tools::toTitleCase(as.character(oscar.data$Decision))
oscar.data$Category = tools::toTitleCase(as.character(gsub("_", " ", oscar.data$Category, fixed = TRUE)))
head(oscar.data,4)
```

```
##   Decision     Category Counts
## 1       No Best Pic Nom    629
## 2      Yes Best Pic Nom     22
## 3       No Best Pic Win    644
## 4      Yes Best Pic Win      7
```

The format of this subset will be suitable for plotting all categories in a single chart using `ggplot`.

Exploratory Data Analysis

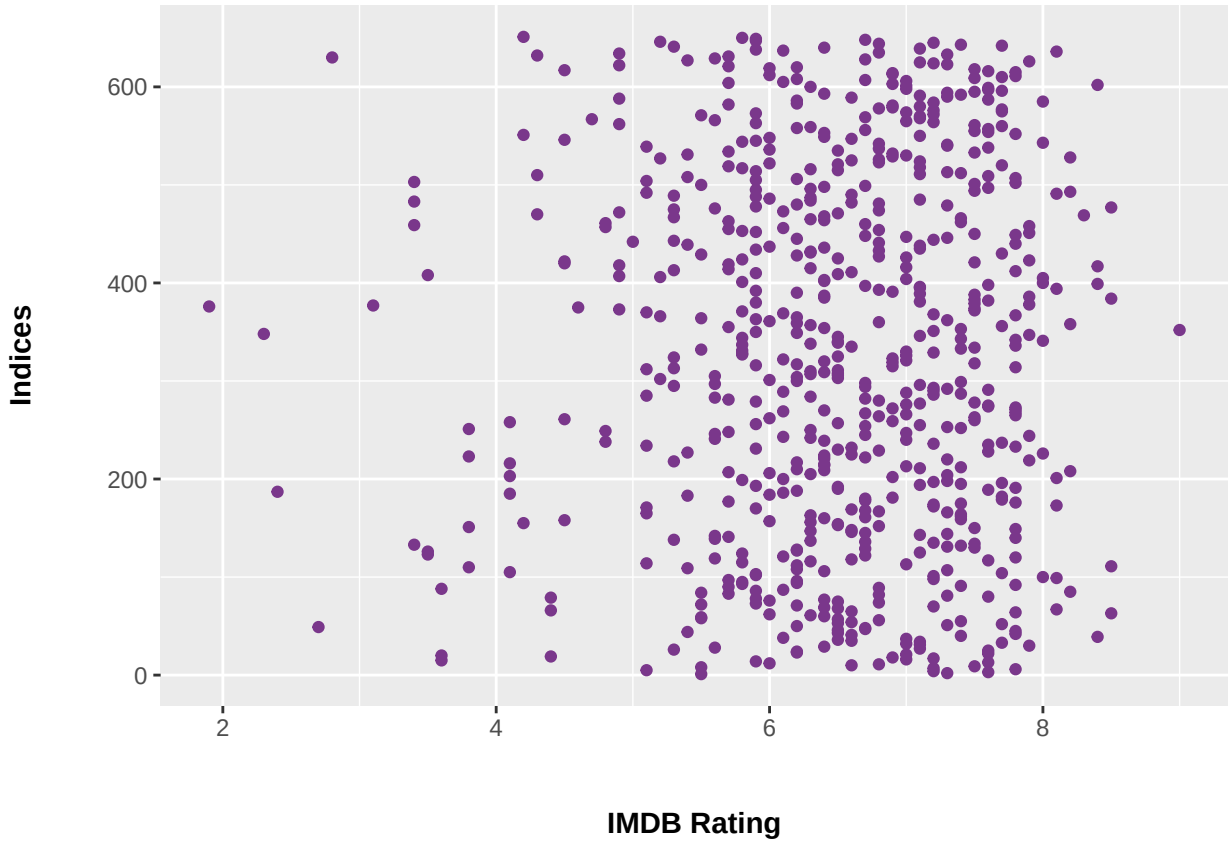
In this section, we will plot the dependent variable against each attribute and check if any correlations exist and assess their strength if they do.

Numerical

We have only 3 numerical variables in our cleaned data - **imdb_ratings**`(dependent variable)`, **critics_score** and **runtime**. Scatter plots are used to explore these variables.

Observing the dependent variable Let's first look at the distribution of our dependent variable ***imdb_rating***.

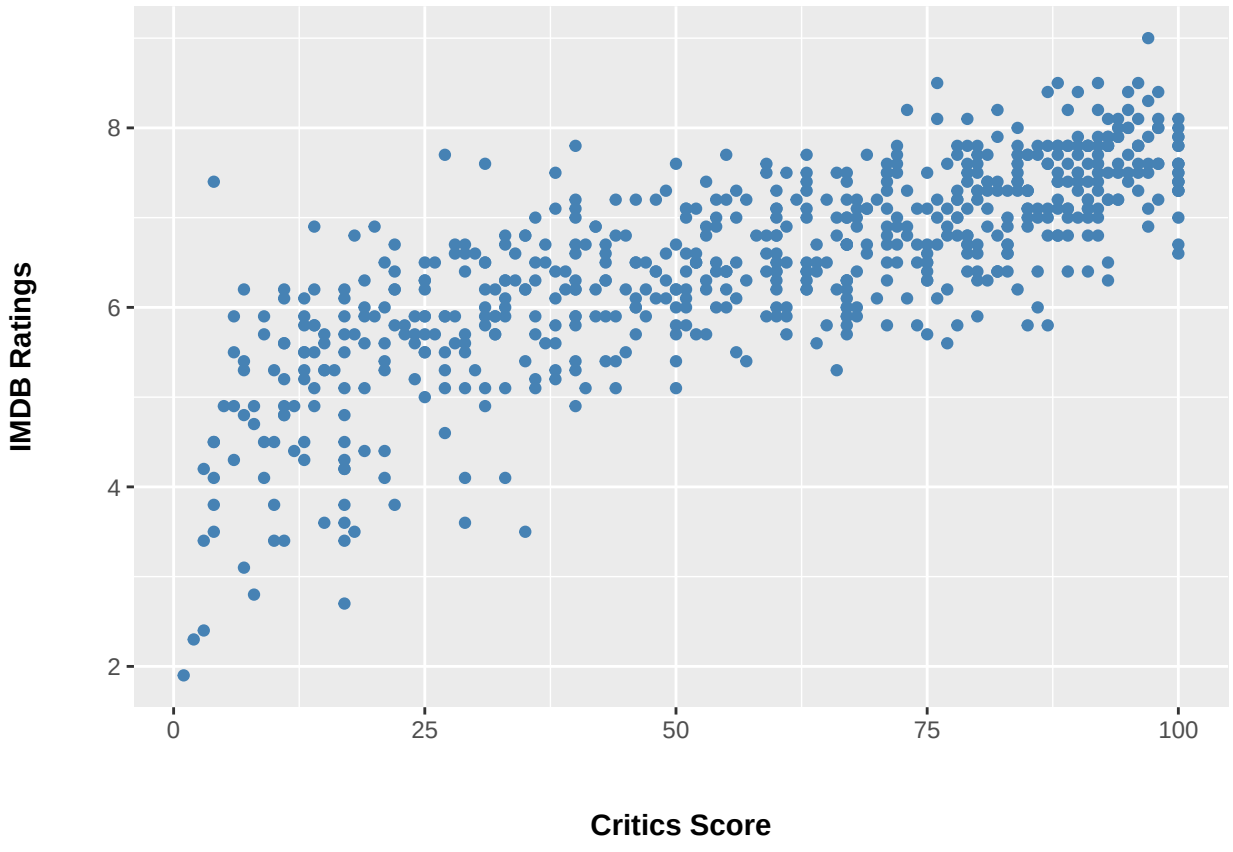
```
ggplot(movies,aes(x=imdb_rating,y = seq_along(imdb_rating))) +
geom_point(color = "mediumorchid4") +
xlab("\n\nIMDB Rating") + ylab("Indices\n\n") +
theme(axis.title = element_text(face = "bold"))
```



We can see that the distribution of ratings is not very even ; **most of the observations have ratings between 5-8**. Ratings below `5` and above `8` are poorly represented which leads to an inherent bias in any model we build. This must be kept in mind when making predictions.

Critics Score Now, we will look at the relationship between **critics_score** and our dependent variable

```
ggplot(movies, aes(x=critics_score,y=imdb_rating)) +
geom_point(color="steelblue") +
ylab("IMDB Ratings\n\n") + xlab("\n\nCritics Score")+
theme(axis.title = element_text(face = "bold"))
```



We can see that the relationship appears to be quite linear, with exceptions here and there, where the same critics score produces varying IMDB ratings.

Let us observe the degree of association between the two by calculating the `correlation coefficient` using `cor()`

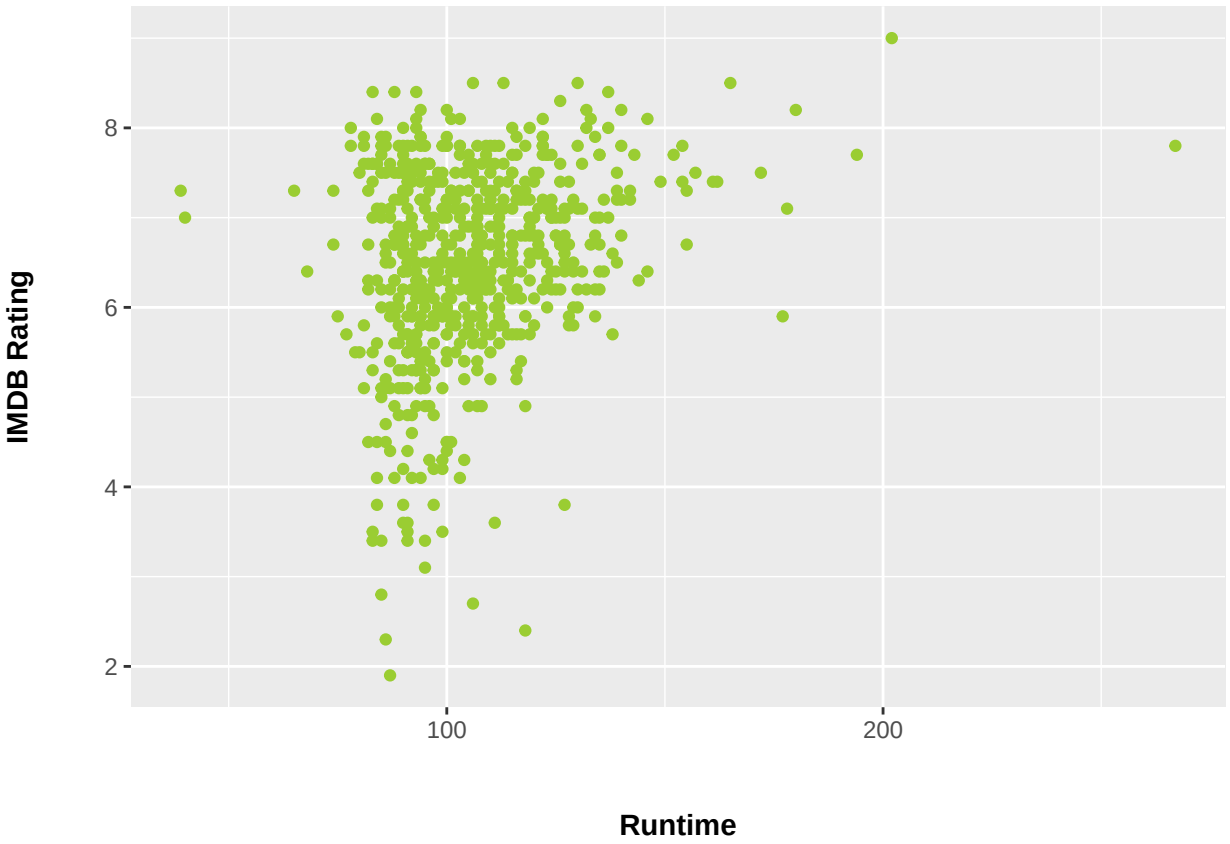
```
cor(imdb_rating,critics_score)
```

```
## [1] 0.7650355
```

Our coefficient is quite **high** and tells us that the two variables are **strongly associated**. It is **positive**, which indicates that they **increase together**.

Runtime Let us observe the relationship between a movie's runtime and audience ratings.

```
ggplot(movies, aes(x=runtime,y=imdb_rating)) +
geom_point(color="olivedrab3")  +
ylab("IMDB Rating\n\n") + xlab("\n\nRuntime")+
theme(axis.title = element_text(face = "bold"))
```



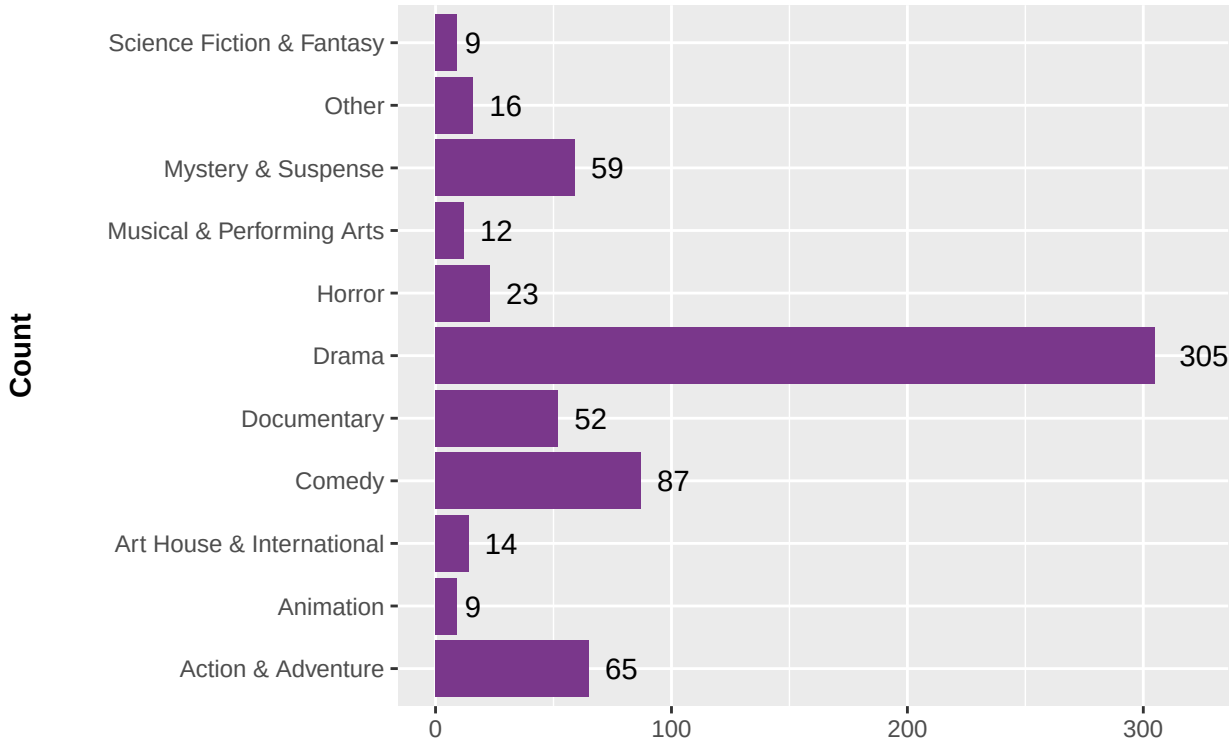
We can clearly see that there appears to **no linear relationship** here - which also agrees well with our intuition that runtime could not possibly contribute to a movie's popularity.

Categorical

The categorical variables we will observe are **genre**, **critics_rating**, **title_type**, **mpaa_ratings***, **top200_box** and all the variables related to Oscar wins. We will use bar charts to observe categorical data. The charts will show the number of movies in the data belonging to each category.

```
# Create a bar plot of the 'genre' variable from the 'movies' dataset
ggplot(data = movies, aes(x = genre)) +

  # Add bars to the plot, filling them with the color 'mediumorchid4'
  geom_bar(fill = "mediumorchid4") +

  # Flip the coordinates to make a horizontal bar plot
  coord_flip() +

  # Add text labels showing the count of each category
  geom_text(stat = "count", aes(label = ..count..), position = "stack", hjust = -0.5) +

  # Set the y-axis limits to make room for the labels
  ylim(0, 320) +
```

```
  # Label the y-axis with 'Genre', and the x-axis with 'Count'
  ylab("\n\nGenre") +
  xlab("Count\n\n") +

  # Customize the appearance of axis titles to be bold
  theme(axis.title = element_text(face = "bold"))
```

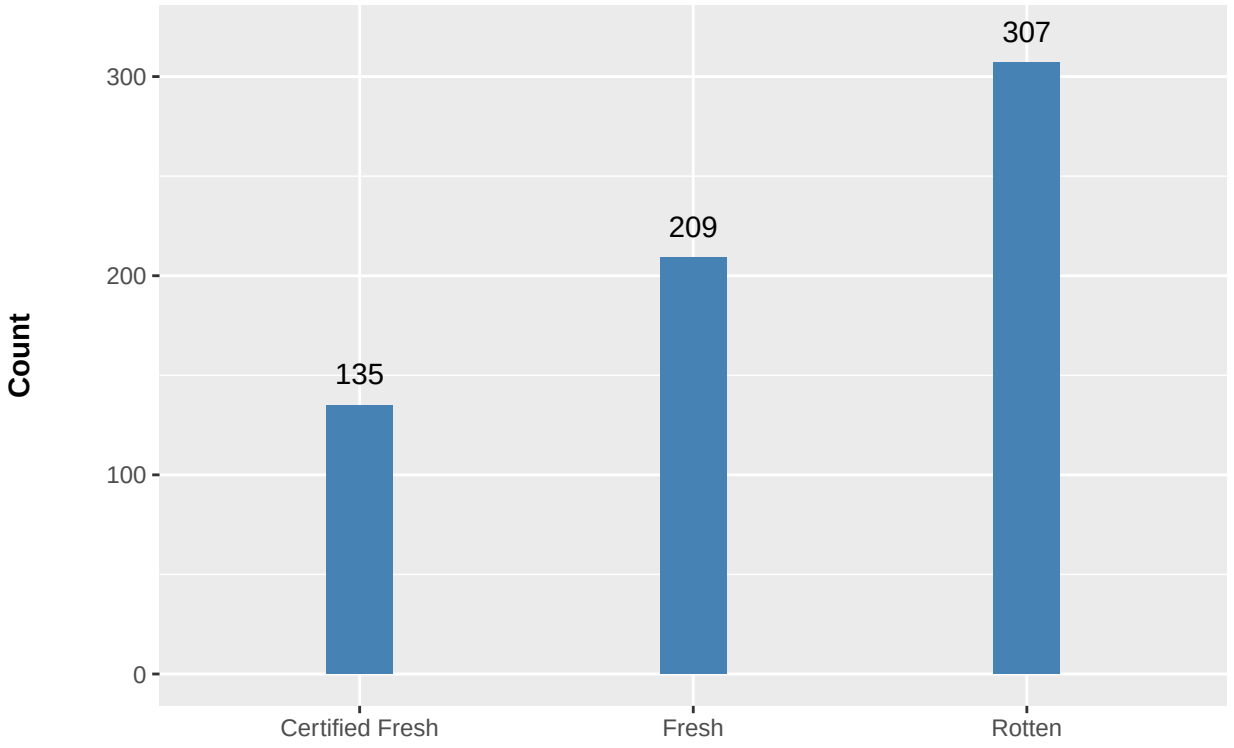


Genre

It can be observed that almost half the movies in our sample come from the genre `Drama`. Few other major categories are `Action`, `Mystery` and `Comedy`. The remaining genres are poorly represented - again, this could lead to some bias in the model.

```
# Create a bar plot of the 'critics_rating' variable from the 'movies' dataset
ggplot(data = movies, aes(x = critics_rating)) +

  # Add bars to the plot, filling them with the color 'steelblue' and setting width to 0.2
  geom_bar(fill = "steelblue", width = 0.2) +

  # Add text labels showing the count of each category, adjust the position with vjust
  geom_text(stat = "count", aes(label = ..count..), vjust = -1) +

  # Set the y-axis limits to make room for the labels
  ylim(0, 320) +

  # Label the y-axis with 'Count', and the x-axis with 'Critics Rating'
```

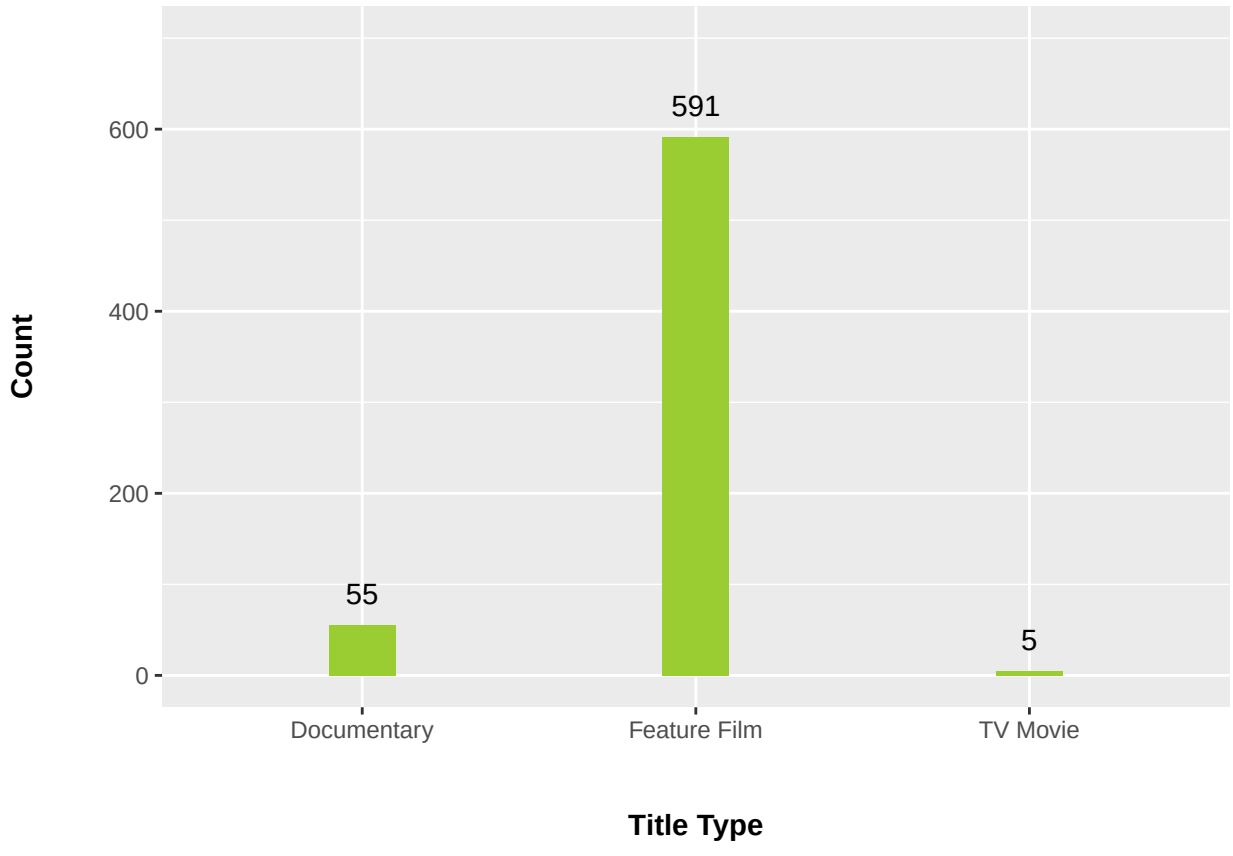
```
  ylab("Count\n\n") +
  xlab("\n\nCritics Rating") +

  # Customize the appearance of axis titles to be bold
  theme(axis.title = element_text(face = "bold"))
```



Critics Rating

We can see that while the proportion of movies with a `Rotten` rating is more, the distribution is not terribly biased and depicts reasonable amounts of variation.

```
ggplot(movies, aes(x=title_type)) +
geom_bar(fill = "olivedrab3",width = 0.2) +
geom_text(stat = "count", aes(label=..count..), vjust=-1) +
ylim(0,700) +
xlab("\n\nTitle Type") + ylab("Count\n\n")+
theme(axis.title = element_text(face = "bold"))
```

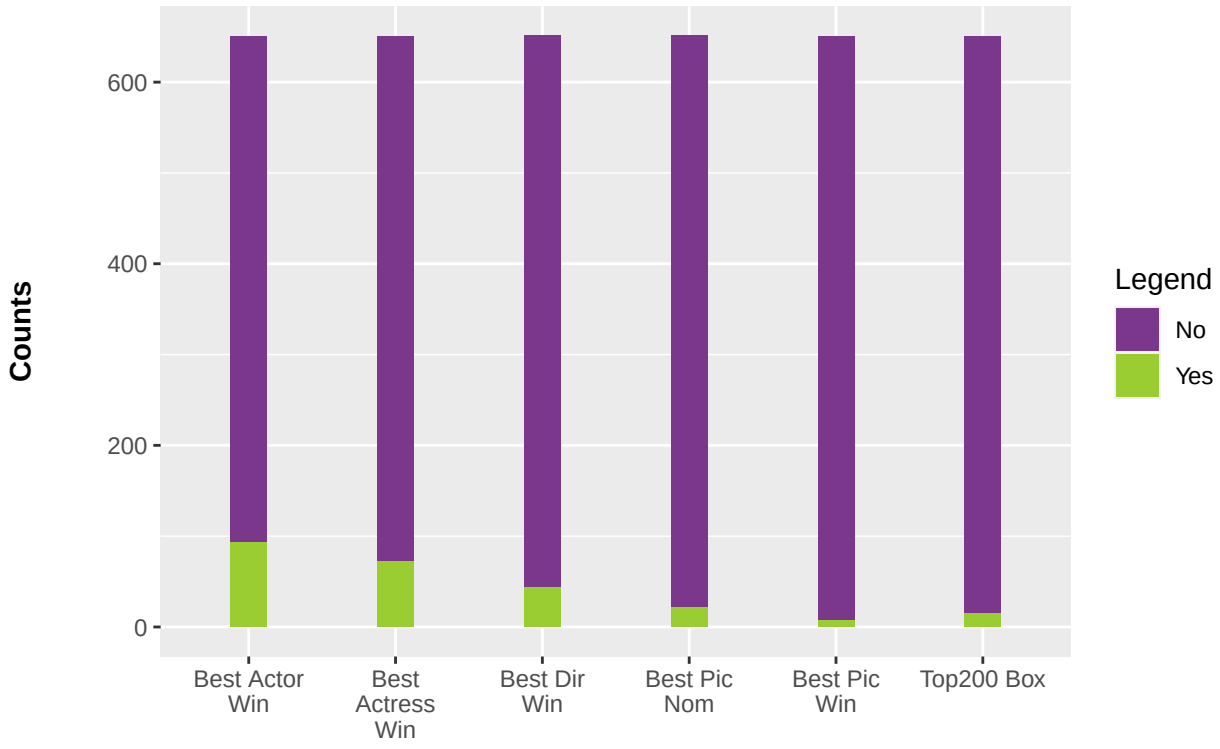


Title type

We can see that a majority of the movies in the data are feature films, with hardly any documentaries and TV Movies. While this would actually bias results, it can be overlooked in the current context.

Academy Awards and Box Office We use the `oscar.data`data frame we created earlier to plot the number of movies which have won in each category as well as the number that were on Box Office Mojo's Top 200 box office list.

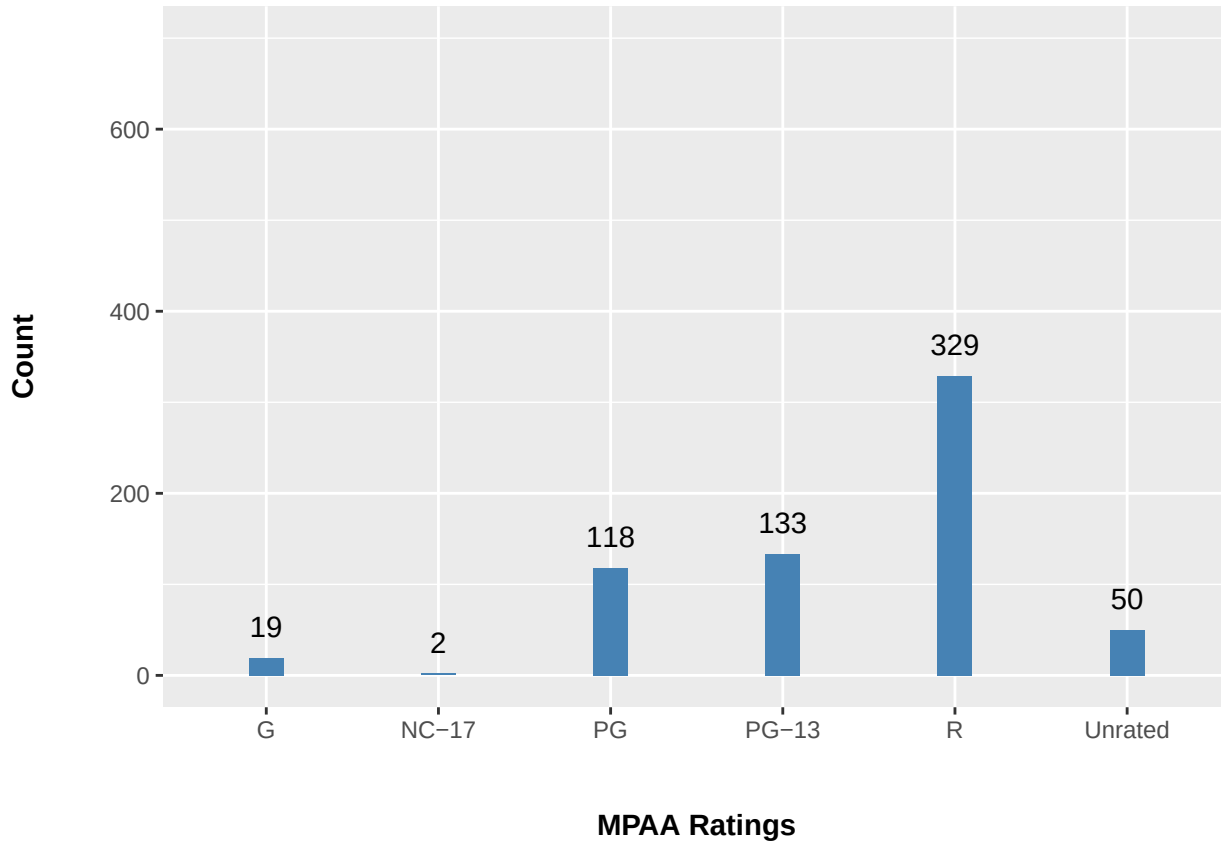
```
ggplot(oscar.data, aes(x = Category, y = Counts, fill = factor(Decision),width=0.25)) +
geom_bar(stat = "identity") +
scale_fill_manual(name = "Legend", values = c("No" = "mediumorchid4","Yes"="olivedrab3")) +
labs(x = "\n\nCategory", y="Counts\n\n")  +
scale_x_discrete(labels = function(x) stringr::str_wrap(x, width = 10))+
theme(axis.title = element_text(face = "bold"))
```



There are hardly any movies in our data which are best picture nominees and the count is worse for wins. The distribution for best director and box office success is similar. There would be no value to adding such variables in our model as they do not showcase much of the dependent variable's variability.

For the best actor and actress, the distribution is arguably better than for the best picture category, but the variation is still rather minimum. Yet, it is not insignificant enough to completely ignore and merits some consideration.

```
ggplot(movies, aes(x=mpaa_rating)) +
geom_bar(fill = "steelblue",width = 0.2)+
geom_text(stat = "count", aes(label=..count..), vjust=-1)+
ylim(0,700)+
xlab("\n\nMPAA Ratings") + ylab("Count\n\n")+
theme(axis.title = element_text(face = "bold"))
```



MPAA Ratings

Most of the movies in this data set fall between PG and R. While R rated movies are maximum, the distribution of the other 2 is reasonable enough to be considered.

Modeling

In this section we will build our linear regression model. We will first decide which variables to add to our model, explore interaction effects and then keep refining our model until the desired selection criterion is fulfilled. Then, we will diagnose our model and ensures it satisfies the assumptions of linear regression and explain the coefficients used.

Selecting our independent variables

We will select our variables based on strength of linear association `(for numerical variables)` or by their ability to explain variation in the dependent variable `(for categorical variables)`.

- From the exploratory analysis, we can see that **critics_score** has a strong positive correlation with our dependent variables. Hence, we will keep it.
- The variable **runtime** has no linear relationship with our response variable and hence can be omitted.
- We keep **genre**, **critics_rating**, **mpaa_ratings** and **title_type** since they explain reasonable amounts of variation in our data.

- Among the academy award/box office variables, we keep only **best_actor_win** and **best_actress_win**. The rest of them are too biased to be of any real value. Even the actor variables do not appear to explain much variability - however, their distribution appears to be slightly more significant and hence we retain them.

Hence, the varaibles we will use while building our model are : ***critics_score, genre, critics_rating, mpaa_ratings, title_type, best_actor_win and best_actress_win***

Note that this is just a basic selection where only the most obviously un-related independent variables are ruled out, such as ones that have no linear relationship with the response variable. We must keep any variable that seems even remotely worth observing, for it could significantly improve our model. We can assess their significance during model selection and decide to leave them out if they do not contribute towards explaining our response variable in any way.

Building our Linear Regression Model

Before starting model building we need to fix our **model selection criterion** and whether we are going to use **backward/forward** elimination to build the model.

The model selection criterion's value determines whether or not we keep a predictor variable in our model. We will use **adjusted R-squared** as the model selection criterion because it gives more reliable predictions when compared to using the `p-value`.

If we are building a model to test for a hypotheses or isolate the relationship between each predictor and response variable, then it makes more sense to use `p-values`. Since our ultimate goal is prediction, we will stick to **adjusted R-squared**, which tells us how much of the variability in our response variable is explained by our model/predictor variables.

We will use **backward selection** to build our model. Let us start by looking at the **adjusted R-squared** value for a model built using all our predictor variables.

```
summary(lm(imdb_rating ~ critics_score + critics_rating + genre +
best_actor_win + best_actress_win + title_type + mpaa_rating))$adj.r.squared
```

```
## [1] 0.619696
```

Now, we improve our model by dropping one variable at a time and observing the impact on **adjusted R-squared**. If it increases, it means we leave out the variable for good. If **adjusted R-squared** decreases we keep the variable since it explains some variability in the model.

Let us first drop **mpaa_rating** and observe the results.

```
summary(lm(imdb_rating ~ critics_score + critics_rating + genre +
best_actor_win + best_actress_win + title_type))$adj.r.squared
```

```
## [1] 0.6198487
```

We can see an increase in **adjusted R-squared** and hence we leave out **mpaa_rating**. Now, let us drop **title_type** and see the results.

```
summary(lm(imdb_rating ~ critics_score + critics_rating + genre +
best_actor_win + best_actress_win))$adj.r.squared
```

```
## [1] 0.6192591
```

We can see a minute decrease in the criterion's value. Hence, we will retain **title_type**. Next, let us try and drop **best_actress_win**.

```
summary(lm(imdb_rating ~ critics_score + critics_rating + genre +
best_actor_win + title_type))$adj.r.squared
```

```
## [1] 0.6202308
```

The criterion value increases, meaning the variable did not contribute much towards explaining variability in the data and hence can be omitted. Let us drop **best_actor_win** and observe the value.

```
summary(lm(imdb_rating ~ critics_score + critics_rating + genre +
title_type))$adj.r.squared
```

```
## [1] 0.6202591
```

There is a very small increase in **adjusted R-squared**, but an increase nevertheless. This reiterates the previous conclusion. Now, let us drop **genre** and check the results.

```
summary(lm(imdb_rating ~ critics_score + critics_rating + title_type))$adj.r.squared
```

```
## [1] 0.6008399
```

There is a clear decrease in **adjusted R-squared**, meaning we will keep the variable. Next, we will drop **critics_rating** and observe the result.

```
summary(lm(imdb_rating ~ critics_score + genre + title_type))$adj.r.squared
```

```
## [1] 0.610349
```

Again, the criterion value decreases which means we will keep **critics_rating**. Finally, the only variable left to test is **critics_score**, as shown below.

```
summary(lm(imdb_rating ~ critics_rating + genre + title_type))$adj.r.squared
```

```
## [1] 0.4759659
```

The drastic drop in **adjusted R-squared** tells us that this variable explains a major portion of the variability in the response and naturally, we retain it.

Our final regression model, along with the associated summary values, is given below.

```
model = lm(imdb_rating ~ critics_score + critics_rating + genre + title_type)
summary(model)
```

```
##
## Call:
## lm(formula = imdb_rating ~ critics_score + critics_rating + genre +
##     title_type)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.75262 -0.37556  0.03053  0.44537  2.12552
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     4.374665   0.327457  13.360   <2e-16 ***
## critics_score                   0.031890   0.002047  15.578   <2e-16 ***
## critics_ratingFresh            -0.154265   0.076979  -2.004   0.0455 *
## critics_ratingRotten            0.270915   0.129357   2.094   0.0366 *
## genreAnimation                 -0.305798   0.238200  -1.284   0.1997
## genreArt House & International  0.405170   0.197428   2.052   0.0406 *
## genreComedy                    -0.200083   0.109777  -1.823   0.0688 .
## genreDocumentary                0.593569   0.262696   2.260   0.0242 *
## genreDrama                      0.168739   0.094000   1.795   0.0731 .
## genreHorror                    -0.266673   0.162259  -1.643   0.1008
## genreMusical & Performing Arts  0.412855   0.226718   1.821   0.0691 .
## genreMystery & Suspense         0.184004   0.121416   1.515   0.1301
## genreOther                      0.107047   0.189261   0.566   0.5719
## genreScience Fiction & Fantasy -0.378773   0.238549  -1.588   0.1128
## title_typeFeature Film          0.096167   0.247907   0.388   0.6982
## title_typeTV Movie             -0.426664   0.391532  -1.090   0.2762
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6685 on 635 degrees of freedom
## Multiple R-squared:  0.629,  Adjusted R-squared:  0.6203
## F-statistic: 71.78 on 15 and 635 DF,  p-value: < 2.2e-16
```

The final value of **adjusted R-squared** for our parsimonious model is **0.6203**. It is worth mentioning that although the variable **title_type** enables our model to capture some amount of variability, it is not at all significant. But we leave it in since our goal is to maximize the chance of reliable predictions.

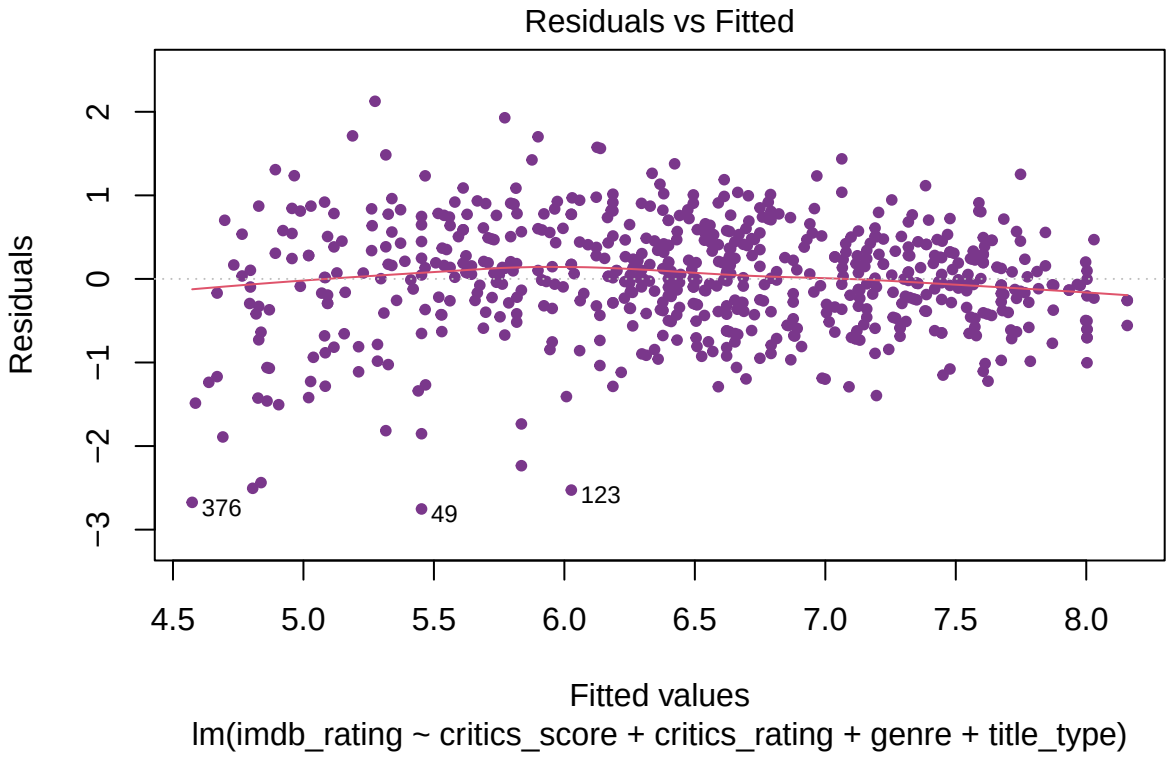
Hence the the factors associated with a change in *imdb_rating* are *critics_score*, *critics_rating*, *genre* and *title_type*

Model Diagnostics

Model diagnostics refers to the process of validating our model against the assumptions of linear regression, to ensure that they are not violated. The **plot()** function provides us with the plots we require for such validation.

Assumption 1 - Linear relationship between independent and dependent variables The linear relationship has meaning only when the independent variable is numerical (it does not make sense to define `linearity` for a categorical variable). This can be verified using the **Residuals vs. Fitted Values** plot, in which a random scatter about zero denotes a linear association.

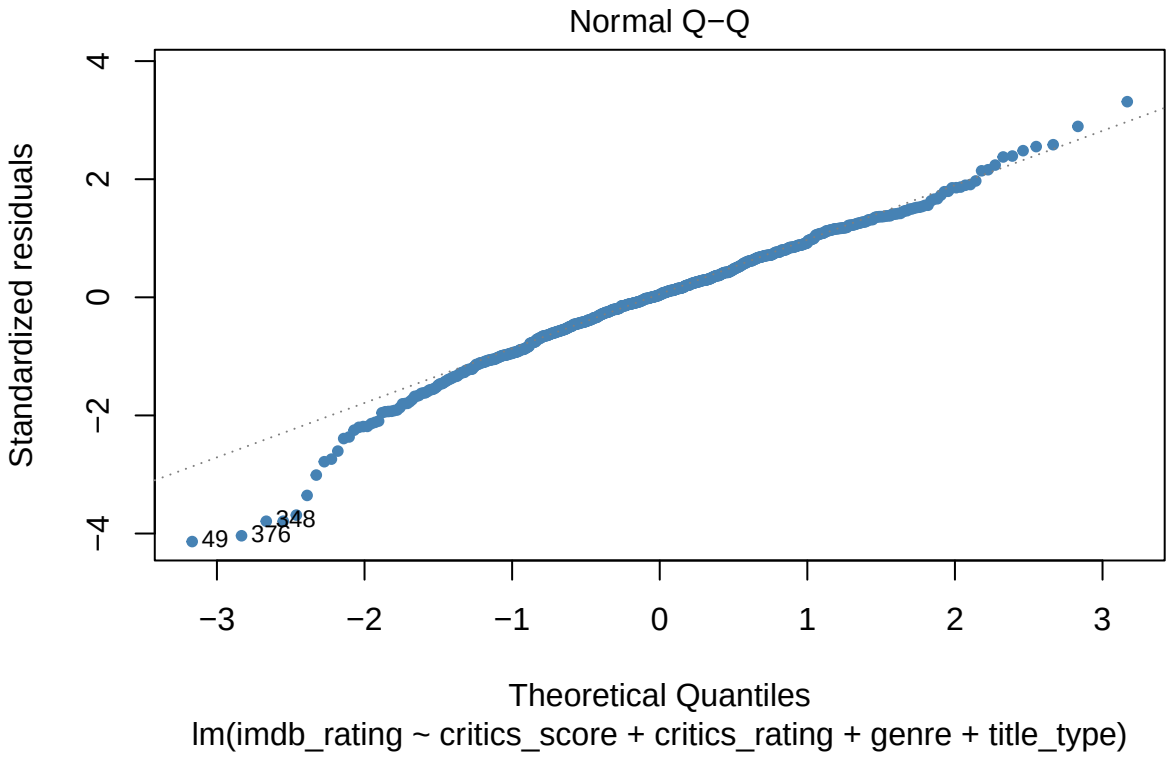
```
# Can also be obtained using plot(resid(model)~fitted(model))
plot(model, which=1, pch = 20, col = "mediumorchid4")
```



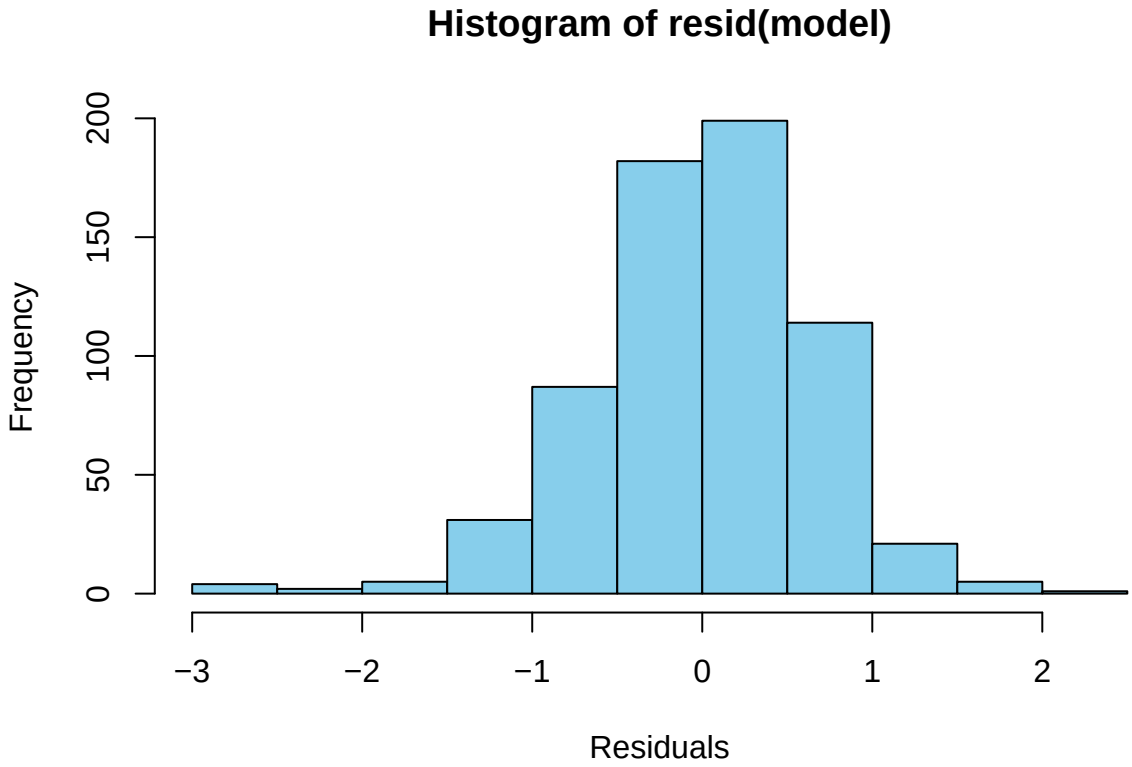
We can see from the plot that the scatter appears fairly random around 0. There appear to be a greater number of outliers towards the lower end of fitted values. However, when prediction is the goal, these factors are not very significant since the quality of prediction remains unaffected by them.

Assumption 2 - Nearly Normal residuals with mean 0* Since some residuals will be positive and some negative, we will have a random scatter around 0, as established. This translates to nearly normal distribution of residuals which can be checked with a normal probability plot or a histogram, as shown below.

```
## Normal Q-Q Plot, can also be obtained using qqplot(resid(model))
plot(model, which=2, pch = 20, col = "steelblue")
```



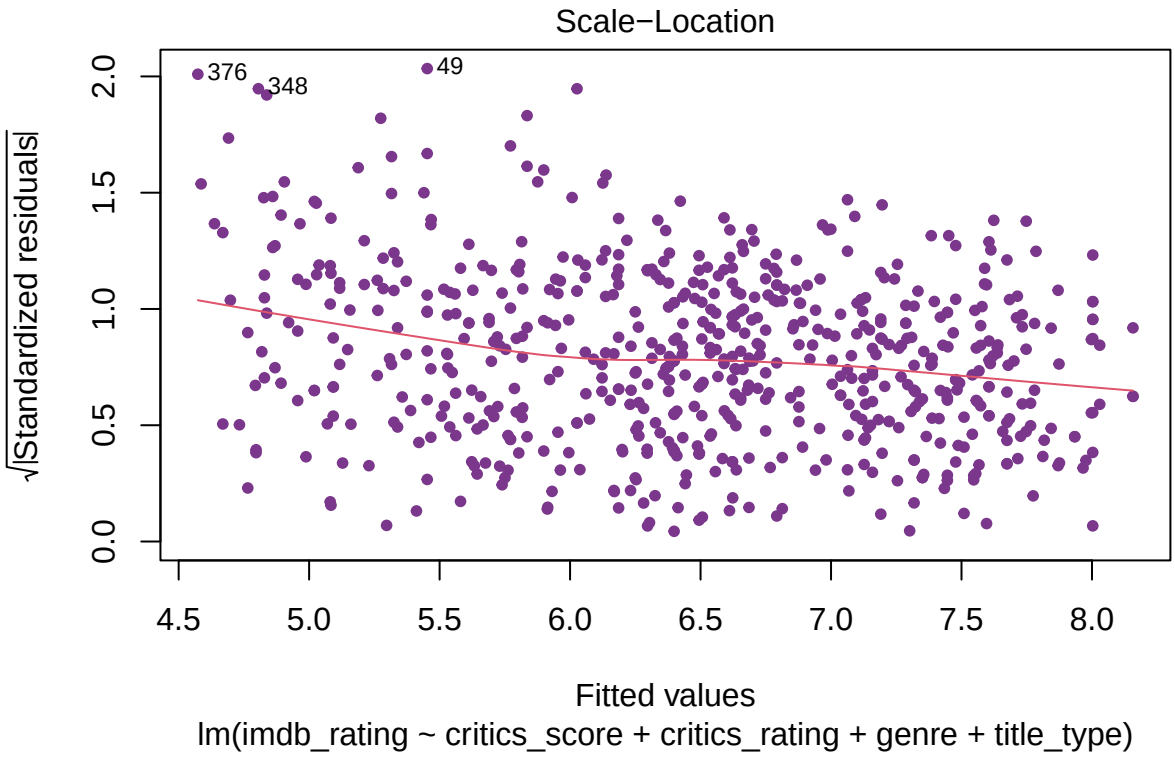

```
## Histogram
hist(resid(model), col= "skyblue", xlab = "Residuals")
```



The distribution of residuals appears fairly normal, except for a slight skew in the tails which can be ignored. Hence, this condition is satisfied.

Assumption 3 - Constant variability of residuals One of the assumptions of linear regression is **Homoscedasticity** - which means that the residuals should be equally variable for low and high values of the predicted response variable. We can check this using a **Scale-Location** plot, as shown below. The line obtained in the plot must be fairly horizontal for this condition to be satisfied.

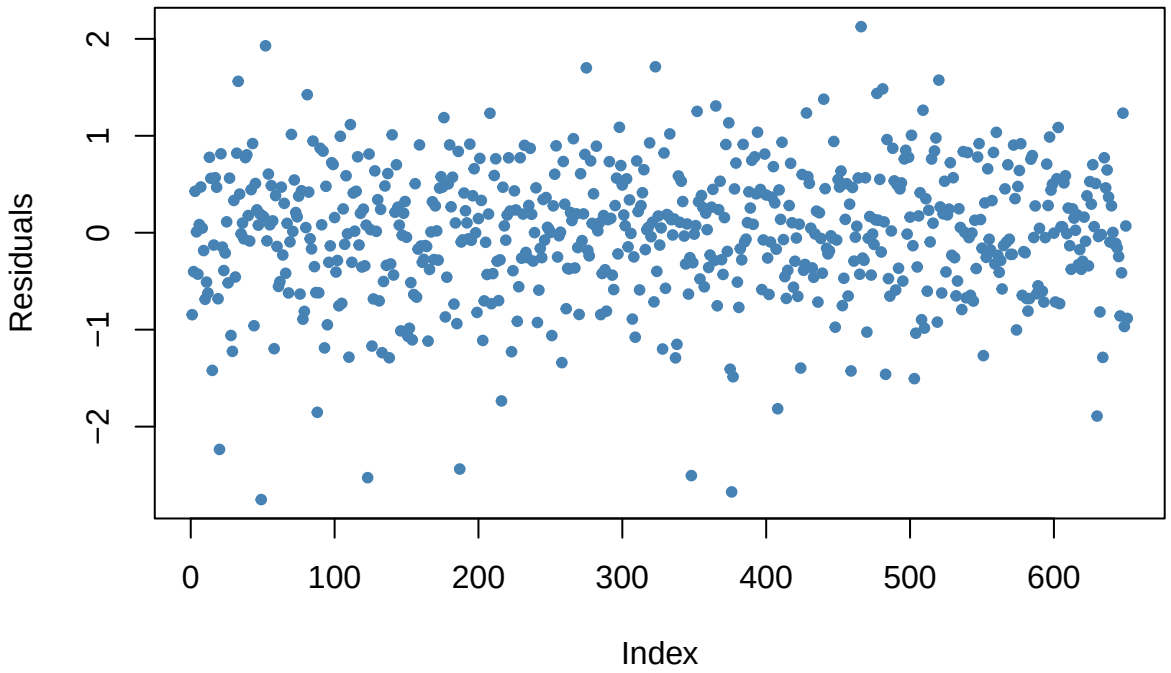
```
## Scale location plot
plot(model, which=3, pch = 20, col = "mediumorchid4")
```



While not perfect, the red line in the plot appears horizontal enough to satisfy the condition.

Assumption 4 - Independent Residuals Independent residuals translates to independent observations. To verify that there are no time series structures and such involved we can plot the residuals of each index of the data i.e. `Residuals vs Order of the data`. If this plot shows no patterns, then it means the observations are independent.

```
plot(resid(model),col = "steelblue", xlab="Index",ylab = "Residuals",pch=20)
```



We do not see any apparent patterns. If there was some non-independent structure we would see the residuals increasing or decreasing but we don't see any such pattern and hence can conclude that the observations are independent.

Interpretation of Model Coefficients

To interpret the coefficients, let us take a look at the model summary again.

```
summary(model)
```

```
##
## Call:
## lm(formula = imdb_rating ~ critics_score + critics_rating + genre +
##     title_type)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.75262 -0.37556  0.03053  0.44537  2.12552
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      4.374665   0.327457  13.360   <2e-16 ***
## critics_score                    0.031890   0.002047  15.578   <2e-16 ***
## critics_ratingFresh             -0.154265   0.076979  -2.004   0.0455 *
## critics_ratingRotten             0.270915   0.129357   2.094   0.0366 *
## genreAnimation                  -0.305798   0.238200  -1.284   0.1997
## genreArt House & International   0.405170   0.197428   2.052   0.0406 *
## genreComedy                     -0.200083   0.109777  -1.823   0.0688 .
## genreDocumentary                 0.593569   0.262696   2.260   0.0242 *
## genreDrama                       0.168739   0.094000   1.795   0.0731 .
## genreHorror                     -0.266673   0.162259  -1.643   0.1008
```

```
## genreMusical & Performing Arts  0.412855   0.226718   1.821   0.0691 .
## genreMystery & Suspense         0.184004   0.121416   1.515   0.1301
## genreOther                      0.107047   0.189261   0.566   0.5719
## genreScience Fiction & Fantasy -0.378773   0.238549  -1.588   0.1128
## title_typeFeature Film          0.096167   0.247907   0.388   0.6982
## title_typeTV Movie             -0.426664   0.391532  -1.090   0.2762
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6685 on 635 degrees of freedom
## Multiple R-squared:  0.629,  Adjusted R-squared:  0.6203
## F-statistic: 71.78 on 15 and 635 DF,  p-value: < 2.2e-16
```

First it is crucial to define the **base level** or **reference level** for our dummy variables. This is the category which is not mentioned in the output (Learn more about dummy encoding **here**)

- For **critics_rating** this is the category `Certified Fresh`.
- For **genre** the reference level is `Action & Adventure`.
- For **title_type** the reference level is `Documentary`.

Interpretation

- Firstly, let us observe our **Intercept**. The coefficient for the intercept tells us that if **critics_score** = 0 , **critics_rating** = `Certified Fresh`, **genre** = `Action & Adventure` and **title_type** = `Documentary`, then the **imdb_rating** will be **4.374**.In our case this is meaningless, since **genre** and **title_type** should match for `Documentary`. A critics rating of `Certified Fresh` is given only when the **critics_score** is 60 or above - it is 0 in this scenario, which is another contradiction. Hence, our intercept has no meaning here and only serves to adjust the height of the line.

- The value of the estimate for **critics_score** is **0.03189**. This means that, all else held constant, for a unit increase in **critics_score**, our response variable **imdb_ratings** will increase by **0.03189** units.

- Let us interpret the variable **critics_rating** which is categorical. The output tells us that, all else held constant, the **imdb_rating** for **critics_rating** = `Fresh` is **0.154** points lower than for **critics_rating** = `Certified Fresh (reference level)`. Similarly, all else held constant, the **imdb_rating** for **critics_rating** = `Rotten` is **0.271** points higher than for **critics_rating** = `Certified Fresh`.

- The model result for **critics_rating** is rather interesting because we would naturally expect a higher critics rating to be associated with greater audience popularity. Yet, the category `Rotten`, which is the lowest ranking one, appears to be correlated with higher IMDB ratings.

- The categorical variable **genre** can be interpreted in a similar fashion. We will interpret only 2 genres here (all follow the same pattern). All else held constant, the **imdb_rating** for the `Comedy` genre is **0.2** points lower than for `Action and Adventure (reference level)`. Similarly, the rating for `Drama` is higher than `Action & Adventure` by **0.168** points.

- For **title_type** the interpretation is as follows - all else held constant, the rating for `Feature Film` is **0.0961** points higher than for `Documentary` and the rating for `TV Movie` is **0.426** points lower than for *Documentary*.

These interpretations are useful in answering our research question. If we put them into context, we can see that some of our factors are associated with a higher **imdb_rating** and by extension a higher popularity among audience while others are associated with lower ratings and popularity.

The most notable is the Critic's Score. A higher score always appears to be correlated with higher IMDB ratings. Similarly, Feature Films appear to have better ratings than Documentaries or TV Movies. The ratings also vary across Genres.

Interaction Effects

In regression, when the influence of an independent variable on a dependent variable keeps varying based on the values of other independent variables, we say that there is an **interaction effect**.

This type of effect makes the model more complex, but if the real world behaves this way, it is critical to incorporate it in your model, as they will explain a lot of variability that might otherwise go unnoticed.

To account for the interaction effect we will simply add the **product** of the independent variables in question to the model. This product is called the **interaction term**. We will assess the significance of each interaction term using **p-values** and keep only the significant/relevant ones in our model.

Interaction Plots

We can identify interaction effects using **interaction plots**. An interaction plot is a line graph that reveals the presence or absence of interactions among independent variables.

This type of plot is created by displaying the fitted values of the dependent variable on the y-axis while the x-axis shows the values of the first independent variable. Meanwhile, the second independent variable can act as a grouping parameter - we can color different sections of our plot according to the second independent variable and thereby produce group-wise lines of best fit.

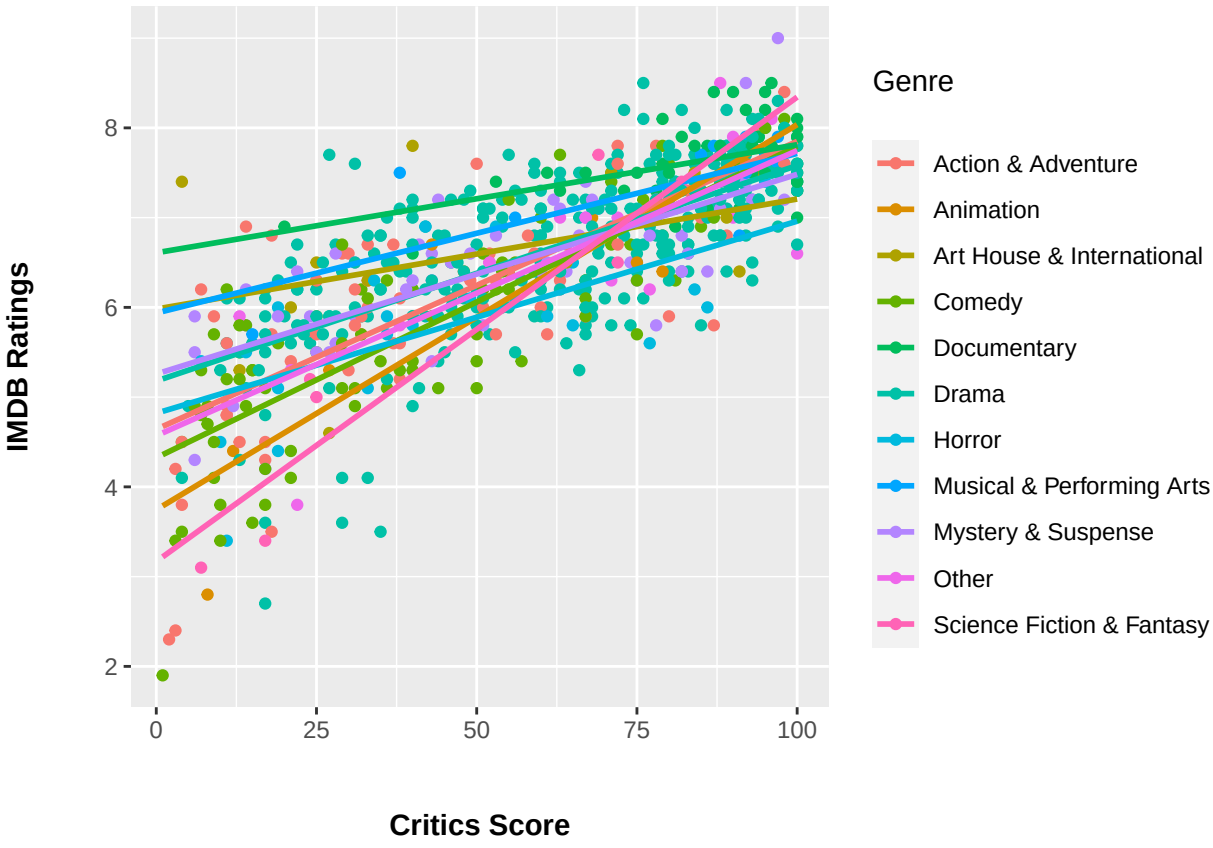
On an interaction plot, parallel lines of best fit indicate that there is no interaction effect while different slopes suggest that it might be present.

Plotting Interaction Effects for our data

For the ease of interpretation and simplicity, we will observe only two-way interactions(between 2 independent variables) and not higher orders. In most cases, 2-way interactions will suffice and going beyond can unnecessarily complicate the model.

Critics Score and Genre Let us observe if the relationship between **imdb_ratings** and **critics_score** changes according to **genre**.

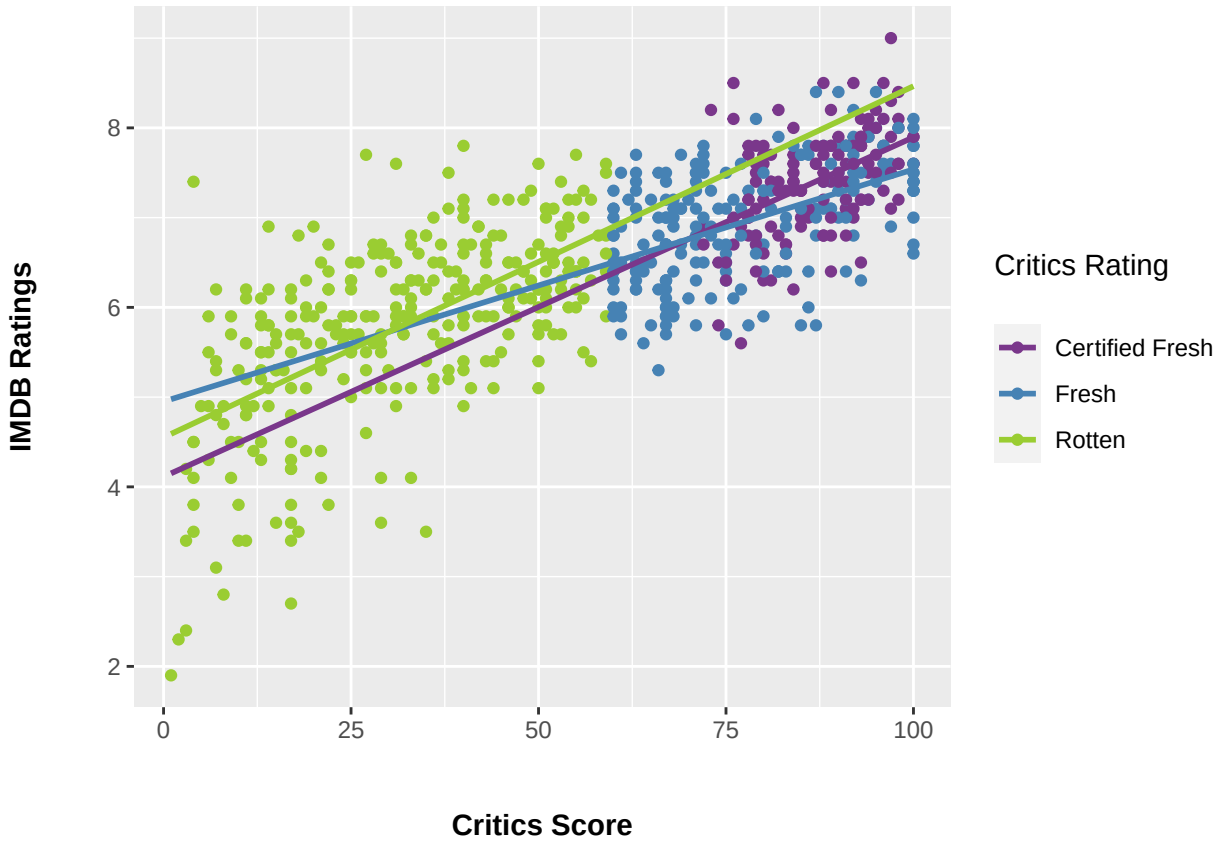
```
ggplot(movies, aes(x=critics_score,y=imdb_rating,color = genre)) +
geom_point() +
stat_smooth(method = "lm", se = FALSE, fullrange = TRUE) +
xlab("\n\nCritics Score") + ylab("IMDB Ratings\n\n") +
labs(color="Genre\n") +
theme(axis.title = element_text(face = "bold"))
```



It can be seen that different genres produce different slopes. This clearly shows that the association between **imdb_ratings** and **critics_scores** is influenced by **genre**. Hence, we can conclude that an **interaction effect exists** here.

Critics Score and Critics Rating Now let us see if the critics ratings influences the relationship between IMDB ratings and critics scores.

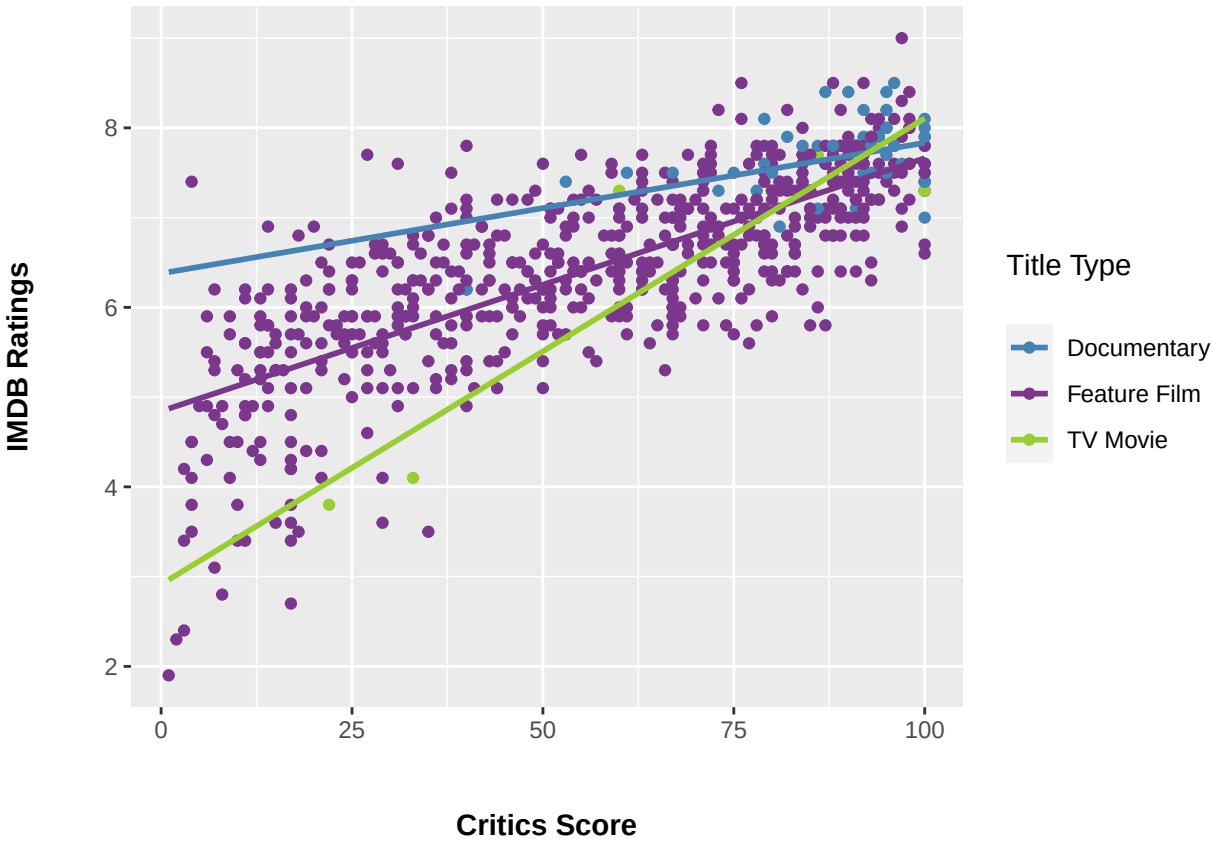
```
ggplot(movies, aes(x=critics_score,y=imdb_rating, color = critics_rating)) +
geom_point() +
stat_smooth(method = "lm", se = FALSE, fullrange = TRUE)+
xlab("\n\nCritics Score") + ylab("IMDB Ratings\n\n")+
labs(color="Critics Rating\n")+
scale_color_manual(values = c("Certified Fresh" = "mediumorchid4","Fresh"="steelblue"
,"Rotten"="olivedrab3")) +
theme(axis.title = element_text(face = "bold"))
```



In this case, even though the slopes intersect, there is **no interaction** per-say. This is because critics rating is simply a categorical representation of critics score values since different ranges of critics scores are grouped into critics ratings. Critics Rating is not influencing the relationship because it is essentially capturing the same relationship. To understand how Rotten Tomatoes groups scores into ratings, see **here**.

Critics Score and Title Type The final interaction left to explore between a numerical and categorical variable in our model is between critics score and the title type of the movie.

```
ggplot(movies, aes(x=critics_score,y=imdb_rating,color = title_type)) +
geom_point() +
stat_smooth(method = "lm", se = FALSE, fullrange = TRUE) +
xlab("\n\nCritics Score") + ylab("IMDB Ratings\n\n") +
labs(color="Title Type\n")+
scale_color_manual(values = c("Feature Film" = "mediumorchid4","Documentary"="steelblue"
,"TV Movie"="olivedrab3"))+
theme(axis.title = element_text(face = "bold"))
```



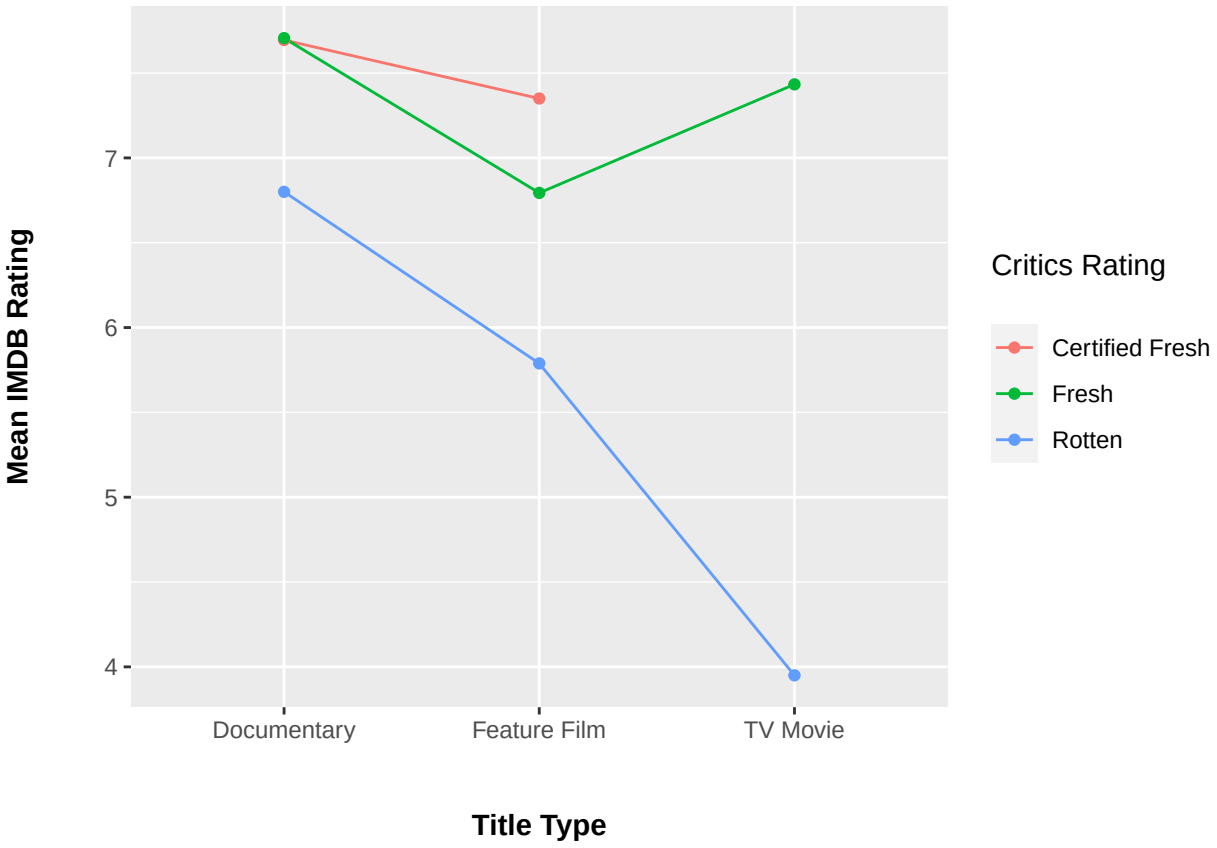
Clearly, an **interaction effect exists** between **critics_score** and **title_type** since none of the slopes are similar.

Title type and Critics Rating Now that we have observed the interaction between the numerical and categorical variables in our model, we can move on to assessing 2 way categorical interactions. These are slightly more complex and require a few extra summaries before plotting.

It is easier to check for interaction effects between categorical variables by adding their product to the model and evaluating the significance of the interaction term in terms of **p-value** and **anova**. If they are significant, then we can conclude that an interaction exists, else we can discard the interaction term and conclude that there is no interaction effect.

However, for the sake of understanding, we will graphically observe this. First, we will group our data by **title_type** and **critics_rating** and obtain the mean **imdb_rating** for each sub-group. We will then plot this data and observe the results.

```
new = movies %>% group_by(title_type,critics_rating) %>% summarise(mir = mean(imdb_rating))

ggplot(new, aes(title_type, mir)) +
geom_line(aes(group = critics_rating, color = critics_rating)) +
geom_point(aes(color = critics_rating)) +
xlab("\n\nTitle Type") + ylab("Mean IMDB Rating\n\n") +
labs(color = "Critics Rating\n")+
theme(axis.title = element_text(face = "bold"))
```



The plot tells us that the average **imdb_rating** for each **title_type** changes based on **critics_rating**. The means that an **interaction effect exists** between the two categories. We will look at the remaining categorical interaction effects by directly incorporating the interaction terms in the model.

Adding Interaction Terms to our Model

In the previous section, we obtained a parsimonious model for our dependent variable. We will now modify that model and add our interaction terms to improve its explanatory power and ensure more variability is accounted for.

The first **interaction term** we will add is **(critics_score* * *genre)**

```
model = lm(imdb_rating ~ critics_score + critics_rating + genre + title_type)
model.with.interaction = lm(imdb_rating ~ critics_score + critics_rating + genre + title_type + critics_
summary(model.with.interaction)$adj.r.squared
```

```
## [1] 0.6423211
```

We can see that the **adjusted R-Squared** has improved, meaning the interaction term is clearly explaining some added variation in the dependent variable, compared to before. We can test the significance of our new model compared to the previous one using the fucntion **anova()**.

```
anova(model, model.with.interaction)
```

```
## Analysis of Variance Table
```

```
##
## Model 1: imdb_rating ~ critics_score + critics_rating + genre + title_type
## Model 2: imdb_rating ~ critics_score + critics_rating + genre + title_type +
##     critics_score * genre
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    635 283.74
## 2    625 263.05 10    20.693 4.9167 7.251e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The extremely low **p-value** tells us that our new model is extremely significant.

We can now add the second interaction term **(critics_score and title_type)** and assess its significance.

```
model = model.with.interaction
model2 = lm(imdb_rating ~ critics_score + critics_rating + genre + title_type + critics_score*genre + c
summary(model2)$adj.r.squared
```

```
## [1] 0.6457855
```

```
anova(model,model2)
```

```
## Analysis of Variance Table
##
## Model 1: imdb_rating ~ critics_score + critics_rating + genre + title_type +
##     critics_score * genre
## Model 2: imdb_rating ~ critics_score + critics_rating + genre + title_type +
##     critics_score * genre + critics_score * title_type
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1    625 263.05
## 2    623 259.66  2    3.3814 4.0564 0.01777 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The **p-value** is below 0.05 and hence the interaction effect is significant. There is also an increase in **adjusted R-Squared**.

However, for the sake of simplicity and ease of explanation, we will retain only those effects which are highly significant and considerably less than 0.05. Hence, we will discard this interaction term.

Now we will explore the interaction between two categorcial variables **critics_rating** and **title_type**.

```
model2  = lm(imdb_rating ~ critics_score + critics_rating + genre + title_type + critics_score*genre + 
summary(model2)$adj.r.squared
```

```
## [1] 0.6497344
```

```
anova(model,model2)
```

```
## Analysis of Variance Table
##
## Model 1: imdb_rating ~ critics_score + critics_rating + genre + title_type +
```

```
##     critics_score * genre
## Model 2: imdb_rating ~ critics_score + critics_rating + genre + title_type +
##     critics_score * genre + critics_rating * title_type
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1    625 263.05
## 2    622 256.36  3    6.6884 5.4094 0.001117 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This interaction effect is observed to be highly significant based on the **p-value** and hence we retain the term. Lets us look at **critics_rating** and **genre**.

```
model = model2
model2  = lm(imdb_rating ~ critics_score + critics_rating + genre + title_type + critics_score*genre + c
summary(model2)$adj.r.squared
```

```
## [1] 0.6523734
```

```
anova(model,model2)
```

```
## Analysis of Variance Table
##
## Model 1: imdb_rating ~ critics_score + critics_rating + genre + title_type +
##     critics_score * genre + critics_rating * title_type
## Model 2: imdb_rating ~ critics_score + critics_rating + genre + title_type +
##     critics_score * genre + critics_rating * title_type + critics_rating *
##     genre
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    622 256.36
## 2    602 246.24 20    10.112 1.2361 0.2177
```

We observe that the **adjusted R-Squared** increases but the interaction term is not at all significant. Hence, we omit the term. Let us now look at the last pair left - **genre** and **title_type**.

```
model2  = lm(imdb_rating ~ critics_score + critics_rating + genre + title_type + critics_score*genre + c
summary(model2)$adj.r.squared
```

```
## [1] 0.6481819
```

```
anova(model,model2)
```

```
## Analysis of Variance Table
##
## Model 1: imdb_rating ~ critics_score + critics_rating + genre + title_type +
##     critics_score * genre + critics_rating * title_type
## Model 2: imdb_rating ~ critics_score + critics_rating + genre + title_type +
##     critics_score * genre + critics_rating * title_type + title_type *
##     genre
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    622 256.36
## 2    619 256.25  3   0.10567 0.0851 0.9682
```

This interaction term is not significant in any way and hence we omit it from our model. Based on the above our final model can be defined as shown below.

```
model  = lm(imdb_rating ~ critics_score + critics_rating + genre + title_type + critics_score*genre + c
summary(model)$adj.r.squared
```

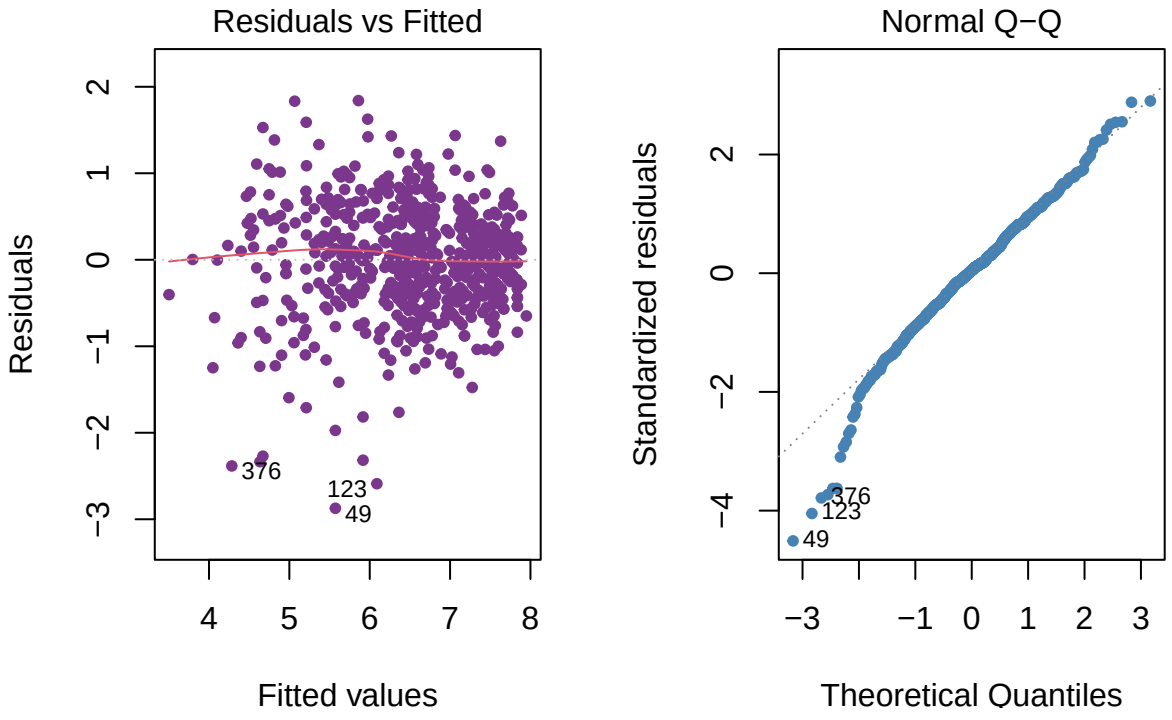
```
## [1] 0.6497344
```

As we can see, have improved our **adjusted R-Squared** value from our previous modeling section, where no interaction terms were included.

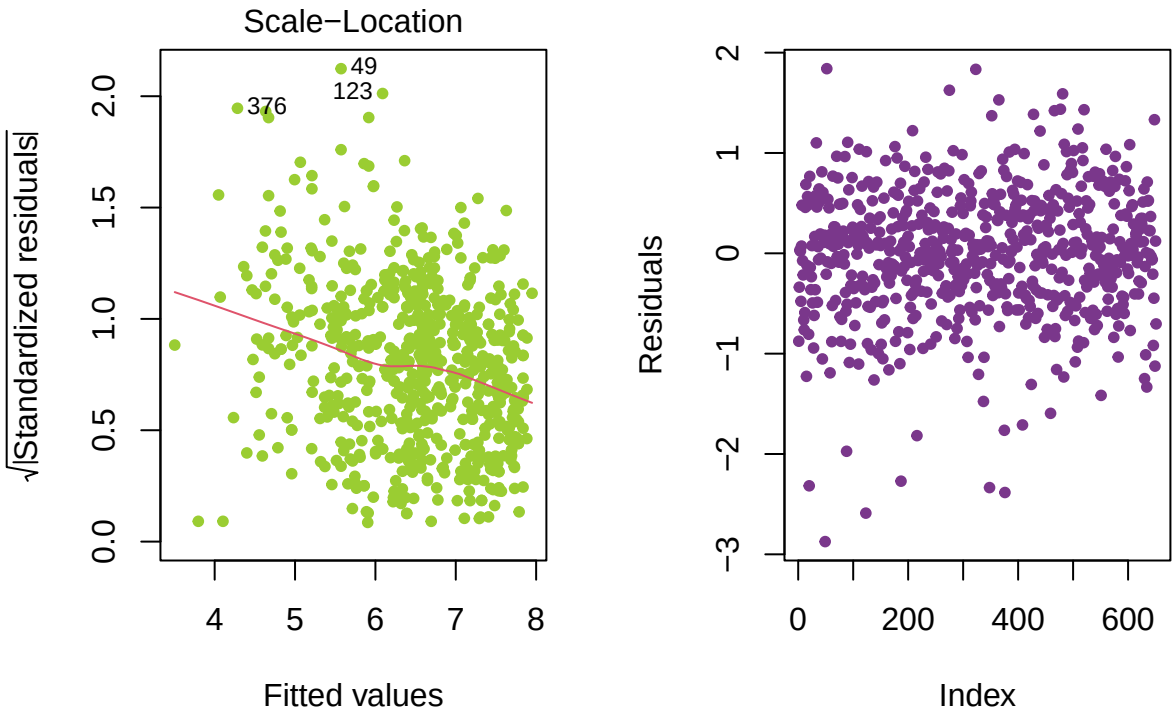
Model Diagnostics

Now that we've constructed our new model, we must ensure that it satisfies the conditions and assumptions of linear regression. To that end, we will plot the diagnostic plots, in the same way as before. For a detailed overview go to the previous section.

```
par(mfrow=c(1,2))
plot(model, which=1, pch = 20, col = "mediumorchid4")
plot(model, which=2, pch = 20, col = "steelblue")
```




```
par(mfrow=c(1,2))
plot(model, which=3, pch = 20, col = "olivedrab3")
plot(resid(model),col = "mediumorchid4", xlab="Index",ylab = "Residuals",pch=20)
```



While not perfect, most of the conditions are reasonably satisfied and hence our diagnostics validate our assumptions.

Interpreting Interaction Coefficients

Let us take a look at the model summary first.

```
summary(model)
```

```
##
## Call:
## lm(formula = imdb_rating ~ critics_score + critics_rating + genre +
##     title_type + critics_score * genre + critics_rating * title_type)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.87301 -0.36324  0.03167  0.42117  1.84017
##
## Coefficients: (1 not defined because of singularities)
##                                              Estimate Std. Error t value
## (Intercept)                                  4.522605   0.383498  11.793
## critics_score                                0.035972   0.003286  10.946
## critics_ratingFresh                          0.057883   0.184971   0.313
## critics_ratingRotten                         0.319884   0.536904   0.596
## genreAnimation                              -0.886433   0.456586  -1.941
## genreArt House & International               1.348770   0.366968   3.675
## genreComedy                                 -0.317842   0.190951  -1.665
## genreDocumentary                             2.036807   0.527131   3.864
## genreDrama                                   0.522531   0.176771   2.956
## genreHorror                                  0.140013   0.302099   0.463
```

```
## genreMusical & Performing Arts                1.376659   0.616648   2.232
## genreMystery & Suspense                        0.582327   0.238935   2.437
## genreOther                                     0.515741   0.463818   1.112
## genreScience Fiction & Fantasy                -1.456340   0.446633  -3.261
## title_typeFeature Film                        -0.161320   0.333094  -0.484
## title_typeTV Movie                            -2.208016   0.677793  -3.258
## critics_score:genreAnimation                   0.010561   0.008056   1.311
## critics_score:genreArt House & International -0.019226   0.006338  -3.034
## critics_score:genreComedy                      0.002760   0.003870   0.713
## critics_score:genreDocumentary                -0.023769   0.007439  -3.195
## critics_score:genreDrama                      -0.007290   0.003301  -2.209
## critics_score:genreHorror                     -0.009445   0.005973  -1.581
## critics_score:genreMusical & Performing Arts -0.016678   0.008329  -2.002
## critics_score:genreMystery & Suspense         -0.008289   0.004275  -1.939
## critics_score:genreOther                      -0.006597   0.006835  -0.965
## critics_score:genreScience Fiction & Fantasy  0.020680   0.007856   2.632
## critics_ratingFresh:title_typeFeature Film   -0.272738   0.201717  -1.352
## critics_ratingRotten:title_typeFeature Film  -0.118289   0.540015  -0.219
## critics_ratingFresh:title_typeTV Movie         2.186382   0.803769   2.720
## critics_ratingRotten:title_typeTV Movie              NA         NA      NA
##                                              Pr(>|t|)
## (Intercept)                                   < 2e-16 ***
## critics_score                                 < 2e-16 ***
## critics_ratingFresh                          0.754439
## critics_ratingRotten                         0.551529
## genreAnimation                               0.052657 .
## genreArt House & International               0.000258 ***
## genreComedy                                  0.096512 .
## genreDocumentary                             0.000123 ***
## genreDrama                                   0.003235 **
## genreHorror                                  0.643191
## genreMusical & Performing Arts               0.025938 *
## genreMystery & Suspense                      0.015082 *
## genreOther                                   0.266591
## genreScience Fiction & Fantasy               0.001172 **
## title_typeFeature Film                       0.628338
## title_typeTV Movie                           0.001184 **
## critics_score:genreAnimation                 0.190333
## critics_score:genreArt House & International 0.002517 **
## critics_score:genreComedy                    0.476014
## critics_score:genreDocumentary               0.001467 **
## critics_score:genreDrama                     0.027569 *
## critics_score:genreHorror                    0.114305
## critics_score:genreMusical & Performing Arts 0.045667 *
## critics_score:genreMystery & Suspense        0.052941 .
## critics_score:genreOther                     0.334831
## critics_score:genreScience Fiction & Fantasy 0.008692 **
## critics_ratingFresh:title_typeFeature Film   0.176840
## critics_ratingRotten:title_typeFeature Film  0.826685
## critics_ratingFresh:title_typeTV Movie       0.006707 **
## critics_ratingRotten:title_typeTV Movie            NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.642 on 622 degrees of freedom
## Multiple R-squared:  0.6648, Adjusted R-squared:  0.6497
## F-statistic: 44.06 on 28 and 622 DF,  p-value: < 2.2e-16
```

It can be noticed that the coefficients differ from our previous model, for almost all the predictors. This is on account of including the interaction terms.

As an example, we will interpret the the effect of **critics_score** on the dependent variable. In our old model, the effect of **critics_score** on the dependent variable was based only on the value of its coefficient.

But here, we have to take into account all the interaction effects **critics_score** has. This means that we basically have to look at any variable that contains ***critics_score*** in it. Here, the only interaction term with critics score is **(critics_score** * ***genre)** and hence we consider its effect in conjunction with the **critics_score**'s coefficient.

For **genre** the reference level is `Action & Adventure`. This means that for a movie of this genre the effect of **critics_score** is simply the value of its coefficient i.e. **0.035972**. All else held constant, a unit increase in **critics_score**, when the genre is `Action & Adventure`, causes the **imdb_rating** to increase by **0.035972**.

Since **genre** has many levels, we will look at `Comedy` as an example. From the summary output we can gather that, all else held constant, a unit increase in the **critics_score** of a `Comedy` movie will cause the **imdb_rating** to increase by the sum of coefficients of ***critics_score*** and the interaction term for `Comedy`, which is **critics_score:genreComedy** (see output).

```
Effect of critics_score for a Comedy movie  = critics_score + critics_score:genreComedy (from output)
                                            = 0.035972 +  0.002760
                                            = 0.038732
```

Hence, a unit increase in the **critics_score** of a `Comedy` movie will cause the **imdb_rating** to increase by **0.038732**.

Another way to phrase it would be to say, the effect of **critics_score** on **imdb_rating** is higher by **0.002760** when the **genre** is `Comedy` as opposed to `Action & Adventure`. Here **0.002760** is the coefficient of the interaction term **critics_score:genreComedy**.

Let us take an example between two categorical interactions - ***critics_rating*** and ***title_type***. For a movie rated `Fresh`, the **imdb_rating** is **0.272738** points lower for a `Feature Film` compared to a `Documentary`.

Significance

Earlier, we applied our interpretations to our research question. We can see that interaction effects will significantly affect the conclusions we came to. For instance, the effect of Critic's scores change in accordance with the genre. Some genres cause an increase in the effect (when compared to the reference level) while others cause a decrease. The same applies to the other attributes as well. Hence, it is always useful to keep interaction effects in mind while building models.

Prediction

In this section, we will pick a movie from 2016 that is not present in the data set and predict its IMDB Rating using our **Multiple Linear Regression Model**.

Let us pick the movie **Sully** which was released in September 2016. First, we will ensure the movie is not already present in the data.

```r
grep("Sully",title,ignore.case = T)
```

```
## integer(0)
```

Since `grep()` returned no matches, this movies does not exist in our data. Now we will use `predict()` to predict the IMDB Rating of the movie. The input data was obtained from Rotten Tomatoes and IMDB.

```r
input = list(critics_score=85, genre = "Drama",critics_rating = "Certified Fresh",title_type = "Feature
predict(model,input)
```

```
##        1
## 7.321812
```

```r
predict(model,input, interval = "prediction",level = 0.95)
```

```
##        fit      lwr      upr
## 1 7.321812 6.054359 8.589264
```

Interpretation

- The predicted **IMDB Rating** is **7.33**. This indicates good popularity among audiences. The actual IMDB rating, as given on the website, is **7.5**. Hence, our model has made a reasonably accurate prediction.
- Based on the prediction interval, we can infer that our model predicts with **95%** confidence that a ***Feature Film*** of genre ***Drama*** having a Critic's Score of ***85*** on Rotten Tomatoes and a Critic's Rating of ***Certified Fresh*** is likely to have an IMDB Rating in the range ***6.05 - 8.6***.

Conclusion

The aim of this work was to identify and understand the attributes that made a movie popular among audiences. To that end, a Multiple Linear Regression model was implemented using R and it was found that the variables associated with a change in movie popularity are **critics_score**, **critics_rating**, **genre** and **title_type**. Our model could explain around **65%** of the variability in our response variable from the predictor variables.

We also explored the effects of **interaction** among said variables and discovered that some of the variables did interact with one another and understanding these not only helped build a better model but also contributed to our overall perception of the solution.

One major **drawback** is that a massive number of movies have been released during the time period this data covers and this data only captures a very **small percentage** of them. Further, the data is quite heavily **biased** towards certain categories and this reduces the reliability of the predictions, since our model does not have enough observations to learn from.

Another **drawback** is that our diagnostic plots were not as random as they should be to call it a perfectly linear model. Some amount of **heteroscedasticity** appears to be present and this could mean that there are better models to solve the problem.

Resources

The links below are very useful in understanding interaction effects, how to plot them and how to interpret them.

Understanding Interaction Effects - 1

Understanding Interaction Effects - 2

Interpreting Interactions in Regression

Interpreting Interaction Coefficients in R

Visualize Interactions using ggplot - 1

Visualize Interactions using ggplot - 2