```
---
title: 'Project 1: Explore and Prepare Data'
Name: Padma Prabhavathi Kondaveeti
GTAccountID: pkondaveeti3
subtitle: |-
  CSE6242 - Data and Visual Analytics - Fall 2017
  Due: Sunday, October 15, 2017 at 11:59 PM UTC-12:00 on T-Square
output:
  html_document: default
---

_Note: This project involves getting data ready for analysis and doing some preliminary investigations. Project 2 will involve modeling and predictions on the same dataset, and will be released at a later date. Both projects will have equal weightage towards your grade. You may reuse some of the preprocessing/analysis steps from Project 1 in Project 2._

# Data

In this project, you will explore a dataset that contains information about movies, including ratings, budget, gross revenue and other attributes. It was prepared by Dr. Guy Lebanon, and here is his description of the dataset:

> The file [`movies_merged`](https://s3.amazonaws.com/content.udacity-data.com/courses/gt-cs6242/project/movies_merged) contains a dataframe with the same name that has 40K rows and 39 columns. Each row represents a movie title and each column represents a descriptor such as `Title`, `Actors`, and `Budget`. I collected the data by querying IMDb's API (see [www.omdbapi.com](http://www.omdbapi.com/)) and joining it with a separate dataset of movie budgets and gross earnings (unknown to you). The join key was the movie title. This data is available for personal use, but IMDb's terms of service do not allow it to be used for commercial purposes or for creating a competing repository.

# Objective

Your goal is to investigate the relationship between the movie descriptors and the box office success of movies, as represented by the variable `Gross`. This task is extremely important as it can help a studio decide which titles to fund for production, how much to bid on produced movies, when to release a title, how much to invest in marketing and PR, etc. This information is most useful before a title is released, but it is still very valuable after the movie is already released to the public (for example it can affect additional marketing spend or how much a studio should negotiate with on-demand streaming companies for "second window" streaming rights).
```

````
# Instructions

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. Open this file in RStudio to get started.

When you execute code within the notebook, the results appear beneath the code. Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

```{r}
x = 1:10
print(x^2)
```

Plots appear inline too:
```{r}
plot(x, x^2, 'o')
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*. Enter some R code and run it.
```{r}
numbers <- c(1:10)
print (numbers)
```

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

Please complete all the tasks below by implementing code chunks that have a `TODO` comment in them, running all code chunks so that output and plots are displayed, and typing in answers to each question (**Q:** ...) next to/below the corresponding answer prompt (**A:**). Feel free to add code chunks/show additional output to support any of the answers.

When you are done, you will need to submit the final R markdown file (as **pr1.Rmd**) with all code chunks implemented and executed, and all text responses written in. You also need to submit a PDF export of the markdown file (as **pr1.pdf**), which should show your code, output, plots and written responses--this will be your project report. Compress these two files into a single .zip archive and upload it on T-Square.

# Setup
````

```{r}
#setwd("C:/Users/Ramaiah/Downloads/pr1")
```

Load data

Make sure you've downloaded the [`movies_merged`](https://s3.amazonaws.com/content.udacity-data.com/courses/gt-cs6242/project/movies_merged) file and it is in the current working directory. Now load it into memory:

```{r}
load('movies_merged')
cat("Dataset has", dim(movies_merged)[1], "rows and", dim(movies_merged)[2], "columns", end="\n", file="")
```

This creates an object of the same name (`movies_merged`). For convenience, you can copy it to `df` and start using it:

```{r}
df = movies_merged
cat("Column names:", end="\n", file="")
colnames(df)
```

Load R packages

Load any R packages that you will need to use. You can come back to this chunk, edit it and re-run to load any additional packages later.

```{r}
library(ggplot2)
install.packages('GGally')
install.packages("NLP",dependencies = TRUE)
install.packages("robustHD",dependencies = TRUE)
install.packages("lubridate",dependencies = TRUE)
remove.packages(tm)
install.packages("http://cran.r-project.org/bin/windows/contrib/3.0/tm_0.5-10.zip",repos=NULL)


install.packages('reshape2')
install.packages('splitstackshape')
library('splitstackshape')
```

```
library(SnowballC)
library(robustHD)
library(lubridate)
library(NLP)
library(tm)
library(GGally)
library(stringr)
library(plyr)
library(reshape2)
```

If you are loading any non-standard packages (ones that have not been discussed in class or explicitly allowed for this project), please mention them below. Include any special instructions if they cannot be installed using the regular `install.packages('<pkg name>')` command.

Non-standard packages used: None

Tasks

Each task below is worth **10** points, and is meant to be performed sequentially, i.e. do step 2 after you have processed the data as described in step 1. Total points: **100**

Complete each task by implementing code chunks as described by `TODO` comments, and by responding to questions ("**Q**:") with written answers ("**A**:"). If you are unable to find a meaningful or strong relationship in any of the cases when requested, explain why not by referring to appropriate plots/statistics.

It is okay to handle missing values below by omission, but please omit as little as possible. It is worthwhile to invest in reusable and clear code as you may need to use it or modify it in project 2.

1. Remove non-movie rows

The variable `Type` captures whether the row is a movie, a TV series, or a game. Remove all rows from `df` that do not correspond to movies.

```{r}
# TODO: Remove all rows from df that do not correspond to movies
df2 <- df[df$Type == "movie",]
dim(df2)
```

Q: How many rows are left after removal? _Enter your response below._

A:40000

2. Process `Runtime` column

The variable `Runtime` represents the length of the title as a string. Write R code to convert it to a numeric value (in minutes) and replace `df$Runtime` with the new numeric column.

```{r}
# TODO: Replace df$Runtime with a numeric column containing the runtime in minutes
df<-df2
temp<-df$Runtime
temp[temp=='N/A']<-'0 min'
temp2 <- sapply(str_extract_all(temp, "\\d+"), function(t) {
             t1 <- as.numeric(t)
          if(length(t1)>1) t1[1]*60 + t1[2] else t1 })
df$Runtime<-temp2

```

Now investigate the distribution of `Runtime` values and how it changes over years (variable `Year`, which you can bucket into decades) and in relation to the budget (variable `Budget`). Include any plots that illustrate.

```{r}
# TODO: Investigate the distribution of Runtime values and how it varies by Year and Budget
summary(df$Runtime)
ggplot(df, aes(x = Runtime)) + geom_density()
df_nayear<-subset(df,!(is.na(df["Year"])))
df_nayear$decade=cut(df_nayear$Year, c(1880, seq(1890, 2020, by=10)))
df_nayear$decade = factor(df_nayear$decade)
ggplot(data = df_nayear)  +facet_wrap(~decade,scales="free")+
geom_density(aes(x=Runtime))   +ggtitle("Runtime VS Year")
finalData<-subset(df,!(is.na(df["Budget"])))
ggplot(data = finalData, aes(y =Runtime , x = Budget ) )  + geom_point(size=1)
+ggtitle("Runtime VS Budget") + ylab("Runtime") + xlab("Budget")+
scale_y_continuous(breaks = seq(0, 800, by = 50)) +
  scale_x_continuous(breaks = seq(0, 425000000, by = 50000000))

```

Feel free to insert additional code chunks as necessary.

Q: Comment on the distribution as well as relationships. Are there any patterns or trends that you can observe?

A:
Mean of runtime is 80.23, median at 90 minimum is 0 maximum at 873 and a standard deviation of 39.52582. Density curve of runtime has two peaks.Divided the Year into Decade buckets and then plotted movies Runtime density.Runtime values are very short till 1910 the runtime values started to increase as time increase. The number of short feature films started to increase till 1960s from 1960 feature films started to increase.Budget over runtime has a linear relationship which makes sense because the more budget you have we can make long duration movie.

3. Encode `Genre` column

The column `Genre` represents a list of genres associated with the movie in a string format. Write code to parse each text string into a binary vector with 1s representing the presence of a genre and 0s the absence, and add it to the dataframe as additional columns. Then remove the original `Genre` column.

For example, if there are a total of 3 genres: Drama, Comedy, and Action, a movie that is both Action and Comedy should be represented by a binary vector <0, 1, 1>. Note that you need to first compile a dictionary of all possible genres and then figure out which movie has which genres (you can use the R `tm` package to create the dictionary).

```{r}
# TODO: Replace Genre with a collection of binary columns
df_nagenre<-df[df$Genre != "N/A",]
df_nagenre<-df_nagenre[df_nagenre$Genre != "NA",]
nagenre<-df_nagenre$Genre



temp_crpus = Corpus(VectorSource(df_nagenre$Genre))



temp_crpus <- tm_map(temp_crpus, stripWhitespace)
temp_crpus <- tm_map(temp_crpus, tolower)
temp_crpus <- tm_map(temp_crpus, removePunctuation)
temp_crpus <- tm_map(temp_crpus, PlainTextDocument)
```

```
tm_map(temp_crpus, stemDocument, language="english")


genre_dict <- DocumentTermMatrix(temp_crpus)
 title <- df_nagenre$Title
  Genre <- df_nagenre$Genre
  Runtime <- df_nagenre$Runtime
  Gross <- df_nagenre$Gross

  res<- data.frame(title, Genre,as.matrix(genre_dict),
Runtime,Gross,stringsAsFactors=FALSE)


uniq_genre=sort(unique(unlist(strsplit(as.character(tolower(df_nagenre$Genre)),",
"))))


   genrecnt1 = list()
   for (column in uniq_genre){
       rep_colname=str_replace_all(column,"[[:punct:]]","")
       genrecnt1[[column]] <- sum(res[,eval(noquote(rep_colname))])
   }

genrecnt1<-(genrecnt1[order(-unlist(genrecnt1))])
top10genre_melt = melt(genrecnt1[1:10])



```

Plot the relative proportions of movies having the top 10 most common genres.

```{r}
# TODO: Select movies from top 10 most common genres and plot their relative
proportions
ggplot(top10genre_melt, aes(L1,value)) +
  geom_bar(fill="darkseagreen",stat="identity") +
  labs(x="Top 10 Genre",y="Count") +
  ggtitle("Top 10 Genre Vs Count")
```

Examine how the distribution of `Runtime` changes across genres for the top 10 most common genres.

```{r}
```

```
# TODO: Plot Runtime distribution for top 10 most common genres
temp<-res$Runtime
temp[temp=='N/A']<-'0 min'
temp2 <- sapply(str_extract_all(temp, "\\d+"), function(t) {
              t1 <- as.numeric(t)
          if(length(t1)>1) t1[1]*60 + t1[2] else t1 })
res$Runtime<-temp2
top10_runtime = list()

for (column in names(genrecnt1[1:10]))
{
    rep_colname=str_replace_all(column,"[[:punct:]]","")
    top10_runtime[[column]] <- res$Runtime[res[,eval(noquote(rep_colname))]==1]
}

ggplot(melt(top10_runtime), aes(L1,value)) + geom_boxplot() + coord_flip()+
scale_y_continuous(labels = scales::comma)+ggtitle("Box plot of Genre and
Runtime")



```

Q: Describe the interesting relationship(s) you observe. Are there any expected or unexpected trends that are evident?

A: Drama and comedy movies have highest nuumber of count wheareas animation, adventure, and documentary genred movies have lowest count.Median of runtime for top 10 genre movies are close to 100 with animation and short film having the lowest median

4. Eliminate mismatched rows

The dataframe was put together by merging two different sources of data and it is possible that the merging process was inaccurate in some cases (the merge was done based on movie title, but there are cases of different movies with the same title). There are 3 columns that contain date information: `Year` (numeric year), `Date` (numeric year), and `Released` (string representation of the release date).

Find and remove all rows where you suspect a merge error occurred based on a mismatch between these variables. To make sure subsequent analysis and modeling work well, avoid removing more than 10% of the rows that have a `Gross` value present.

Note: Do not remove the rows with `Gross == NA` at this point, just use this a guideline.

```{r}
# TODO: Remove rows with Year/Date/Released mismatch
elim_rows_df<-df2
elim_rows_df$Released = as.Date(elim_rows_df$Released, "%Y-%m-%d")
elim_rows_df$ReleasedYear = as.numeric(format(elim_rows_df$Released, "%Y"))
mismatch_df <- elim_rows_df[(elim_rows_df$Year == elim_rows_df$ReleasedYear)|
((elim_rows_df$Year+1) == elim_rows_df$ReleasedYear),]
orig_len = length(elim_rows_df$Gross[!is.na(elim_rows_df$Gross)])
print(orig_len)
new_len = length(mismatch_df$Gross[!is.na(mismatch_df$Gross)])
print(new_len)
print((orig_len-new_len)/orig_len)
```

Q: What is your precise removal logic, and how many rows remain in the resulting dataset?

A:
 In order to find Released-Year mismatch, parse the Released column and extract the year part of it and assign it to a new column ReleasedYear.remove the rows that ReleasedYear don't equal Year OR Year is one year greater than ReleasedYear.38169 rows remain in resulting data set. 4422 rows remains with gross not equal to NA. 2.9% of rows with gross not equal to na are removed with this logic.

5. Explore `Gross` revenue

For the commercial success of a movie, production houses want to maximize Gross revenue. Investigate if Gross revenue is related to Budget, Runtime or Genre in any way.

Note: To get a meaningful relationship, you may have to partition the movies into subsets such as short vs. long duration, or by genre, etc.

```{r}
# TODO: Investigate if Gross Revenue is related to Budget, Runtime or Genre
movies_gross_na = subset(df2, !is.na(df2$Gross))
movies_gross_na <- movies_gross_na[ movies_gross_na$Gross != 0 ,]
temp<-movies_gross_na$Runtime
temp[temp=='N/A']<-'0 min'
temp2 <- sapply(str_extract_all(temp, "\\d+"), function(t) {
             t1 <- as.numeric(t)
```

```
          if(length(t1)>1) t1[1]*60 + t1[2] else t1 })
movies_gross_na$Runtime<-temp2
short_duration_df = subset(movies_gross_na,movies_gross_na$Runtime < 50)
long_duration_df = subset(movies_gross_na,movies_gross_na$Runtime > 90)
ggplot(movies_gross_na, aes(log(Budget), log(Gross))) + geom_point() +
ggtitle("Movies Budget vs. Gross")

ggplot(short_duration_df, aes(Runtime,Gross)) + geom_point() + ggtitle("Runtime
vs. Gross")
ggplot(long_duration_df, aes(Runtime,Gross)) + geom_point()+ ggtitle("Long
Runtime vs. Gross")
df_genre_rev<-data.frame(Genre=character(), Gross=numeric(), stringsAsFactors =
FALSE)
for (column in names(genrecnt1[1:15]))
{
    rep_colname=str_replace_all(column,"[[:punct:]]","")
       genrecnt1[[column]] <- sum(res[,eval(noquote(rep_colname))])
  df_genre_rev <- rbind(df_genre_rev, data.frame(Genre = column,Gross=
res$Gross[res[,eval(noquote(rep_colname))] == 1]))
}
df_genre_rev = subset(df_genre_rev, !is.na(df_genre_rev$Gross) &
df_genre_rev$Gross != 0)
zeroToFifthQ = boxplot.stats(df_genre_rev$Gross)$stats[c(1, 5)]
ggplot(df_genre_rev, aes(reorder(Genre, -Gross,median), Gross)) +
  geom_boxplot() +
  theme(axis.text.x=element_text(angle=90,hjust=1,vjust=0.5)) +
  xlab("Genre") +
  coord_flip() +
  coord_cartesian(ylim = zeroToFifthQ*2) +
  scale_y_continuous(labels = scales::comma) +
  ggtitle("Movies Genre vs. Gross")

 ggpairs(data=df_genre_rev)
```

Q: Did you find any observable relationships or combinations of Budget/Runtime/Genre that result in high Gross revenue? If you divided the movies into different subsets, you may get different answers for them - point out interesting ones.

A:
A graph is plotted on a log scale of gross revenue and log scale of budget. from the graph we can infer that gross revenue is proportional to budget i.e as the budget of the movie increases revenue increases.

Movies are divided into two subsets short duration and long duration based on the runtime. relationship of Short duration movies(runtime < 50) with gross revenue cannot be determined. The relationship of long duration movies(runtime >90)with gross revenue is linear.

From the boxplot of gross revenue vs top 15 genre we can infer that median of gross revenue for animation movie is high and documentary movies has lowest median. The median of gross revenue for the most of the genre movies is less than 100000000$.

From the graph of release month vs gross revenue we can infer that the best month to release a movie is in the month of Jun, Dec, July and the worst month is sep,
.

```{r}
# TODO: Investigate if Gross Revenue is related to Release Month
RelMonth <- month(as.POSIXlt(mismatch_df$Released, format="%y-%m-%d"))
mismatch_df$RelMonth <- month.abb[RelMonth]
ggplot(mismatch_df, aes(mismatch_df$RelMonth, mismatch_df$Gross)) +
  geom_bar(stat="identity") + labs(x="Released month",y="Gross Revenue") +
  ggtitle("Gross Revenue by released Month")

```

6. Process `Awards` column

The variable `Awards` describes nominations and awards in text format. Convert it to 2 numeric columns, the first capturing the number of wins, and the second capturing nominations. Replace the `Awards` column with these new columns, and then study the relationship of `Gross` revenue with respect to them.

Note: The format of the `Awards` column is not standard; you may have to use regular expressions to find the relevant values. Try your best to process them, and you may leave the ones that don't have enough information as NAs or set them to 0s.

```{r}
# TODO: Convert Awards to 2 numeric columns: wins and nominations
df2$Wins = 0
df2$Nominations = 0
InvalidAwrds=0
for(i in 1:nrow(df2)){
  numCount = as.numeric( unlist(
```

```
                        regmatches(df2[i,'Awards'], gregexpr("+[0-9]+",
df2[i,'Awards'])))
  if(length(numCount)[1] == 1)
  {
      if((length(grep("win",df2[i,]$Awards))>0) ||
(length(grep("won",df2[i,]$Awards))>0)|| (length(grep("Won",df2[i,]$Awards))>0)){

      df2[i,]$Wins = as.numeric(numCount[1])
    }
    if(length(grep("nomination",df2[i,]$Awards))>0){
         df2[i,]$Nominations = as.numeric(numCount[1])
    }

  }
  if(length(numCount)[1] == 2)
  {
          if((length(grep("win",df2[i,]$Awards))>0) ||
(length(grep("won",df2[i,]$Awards))>0)|| (length(grep("Won",df2[i,]$Awards))>0)){

      df2[i,]$Wins = as.numeric(numCount[1])
    }
    if(length(grep("nomination",df2[i,]$Awards))>0){

      df2[i,]$Nominations = as.numeric(numCount[2])
    }

  }
  if(length(numCount)[1] == 3)
  {
          if((length(grep("win",df2[i,]$Awards))>0) ||
(length(grep("won",df2[i,]$Awards))>0)|| (length(grep("Won",df2[i,]$Awards))>0)){

      df2[i,]$Wins = as.numeric(numCount[2])
    }
    if(length(grep("nomination",df2[i,]$Awards))>0){

      df2[i,]$Nominations = as.numeric(numCount[3])
    }
  }
  if(length(numCount)[1] == 0){

    df2[i,]$Wins = 0
    df2[i,]$Nominations = 0
    InvalidAwrds = InvalidAwrds + 1
```

```
  }

}
```

Q: How did you construct your conversion mechanism? How many rows had valid/non-zero wins or nominations?

A:
For each row in the dataset check how many numbers exist in the Awards column using regex . Based on the number count decide the value for Wins and Nominations column. If there is one number: use regular expression to decide if this number is for Wins or Nominations. If there are two numbers: use regular expressions to make sure the first one goes to Wins if any of the following exist("Win", "win", "won"), use regular expression to make sure the second one goes to Nominations if ("nomination") exists. If there are three numbers: use regular expressions to make sure the second one goes to Wins if any of the following exist("Win", "win", "won"), use regular expression to make sure the third one goes to Nominations if ("nomination") exists. If there are no numbers: count this row as Invalid row. I found rows with valid non-zero wins or nominations.

```{r}
# TODO: Plot Gross revenue against wins and nominations
ggplot(df2) +
  geom_point(data=df2, aes(x=Wins, y=Gross)) +
  ggtitle("Wins vs. Gross")+ xlab("Wins")+ ylab("Gross")
ggplot(df2) +
  geom_point(data=df2, aes(x=Nominations, y=Gross)) +
  ggtitle("Nominations vs. Gross")+ xlab("Nominations")+ ylab("Gross")
```

Q: How does the gross revenue vary by number of awards won and nominations received?

A: From the graphs we can infer that wins or nominations have linear relationship with Gross.

7. Movie ratings from IMDb and Rotten Tomatoes

There are several variables that describe ratings, including IMDb ratings (`imdbRating` represents average user ratings and `imdbVotes` represents the number of user ratings), and multiple Rotten Tomatoes ratings (represented by several variables pre-fixed by `tomato`). Read up on such ratings on the web (for example [rottentomatoes.com/about](https://www.rottentomatoes.com/about) and [

www.imdb.com/help/show_leaf?votestopfaq](http://www.imdb.com/help/show_leaf?votestopfaq)).

Investigate the pairwise relationships between these different descriptors using graphs.

```{r}
# TODO: Illustrate how ratings from IMDb and Rotten Tomatoes are related
ggpairs(df2[, c("imdbVotes","imdbRating","tomatoMeter","tomatoRating",
"tomatoUserRating")],
                title = " Pairwise relationships between IMDB and tomato
ratings.")
df_ratings <- data.frame(df2$imdbRating,df2$tomatoRating)
df_ratings <-df_ratings[complete.cases(df_ratings),]

ggplot(melt(df_ratings),aes(x=value, fill=variable)) + geom_density(alpha=0.25) +
  ggtitle("Density plot: Imdb Rating Vs Tomato Rating")
```

Q: Comment on the similarities and differences between the user ratings of IMDb and the critics ratings of Rotten Tomatoes.

A: Pairwise relationships between IMDB and tomato ratings. that most rating columns have a bell shaped distribution with some sort skewness except imdbVotes. We can also notice that tomatoRating and tomatoMeter have high correlation 94 tomatoRating and imdbRating follows that with 79.4 correlation. Lowest correlation happend to be between tomatoMeter and imdbVotes.

8. Ratings and awards

These ratings typically reflect the general appeal of the movie to the public or gather opinions from a larger body of critics. Whereas awards are given by professional societies that may evaluate a movie on specific attributes, such as artistic performance, screenplay, sound design, etc.

Study the relationship between ratings and awards using graphs (awards here refers to wins and/or nominations).

```{r}
# TODO: Show how ratings and awards are related
ggplot(df2) +
  geom_point(aes(x=imdbVotes,Wins+Nominations)) +
  geom_smooth(aes(x=imdbVotes, Wins+Nominations), method="lm") +
  ggtitle(" Wins + Nominations vs. IMDB Votes")+xlab("IMDB Votes")+ylab("Wins +
Nominations")
```

```
ggplot(df2) +
  geom_point(aes(x=imdbRating,Wins+Nominations)) +
  geom_smooth(aes(x=imdbRating, Wins+Nominations), method="lm") +
  ggtitle(" Wins + Nominations vs. IMDB Rating")+xlab("IMDB Rating")+ylab("Wins +
Nominations")
ggplot(df2) +
  geom_point(aes(x=tomatoMeter,Wins+Nominations)) +
  geom_smooth(aes(x=tomatoMeter, Wins+Nominations), method="lm") +
  ggtitle(" Wins + Nominations vs. Tomato Meter Rating")+xlab("Tomato Meter
Rating")+ylab("Wins + Nominations")
ggplot(df2) +
  geom_point(aes(x=tomatoRating,Wins+Nominations)) +
  geom_smooth(aes(x=tomatoRating, Wins+Nominations), method="lm") +
  ggtitle(" Wins + Nominations vs. Tomato  Rating")+xlab("Tomato
Rating")+ylab("Wins + Nominations")
ggplot(df2) +
  geom_point(aes(x=tomatoUserRating,Wins+Nominations)) +
  geom_smooth(aes(x=tomatoUserRating, Wins+Nominations), method="lm") +
  ggtitle(" Wins + Nominations vs. Tomato User Rating")+xlab("Tomato User
Rating")+ylab("Wins + Nominations")

```

Q: How good are these ratings in terms of predicting the success of a movie in winning awards or nominations? Is there a high correlation between two variables?

A:
All ratings have linear correlation with movies' Wins + Nominations; so they can be good predictors for movie award model. The highest correlation is between IMDB votes and total award wins/nomination.

9. Expected insights

Come up with two new insights (backed up by data and graphs) that is expected. Here “new” means insights that are not an immediate consequence of one of the above tasks. You may use any of the columns already explored above or a different one in the dataset, such as `Title`, `Actors`, etc.

```{r}
# TODO: Find and illustrate two expected insights
ggplot(df2) +
  geom_point(aes(x=Gross,Wins+Nominations)) +
  geom_smooth(aes(x=Gross, Wins+Nominations), method="lm") +
```

```
  ggtitle(" Wins + Nominations vs. Gross Revune")+xlab("Gross
Revenue")+ylab("Wins + Nominations")
ggplot(df2) +
  geom_point(aes(x=Domestic_Gross,Gross)) +
  geom_smooth(aes(x=Domestic_Gross, Gross), method="lm") +
  ggtitle(" Domestic Grosss vs. Gross Revune")+xlab("Domestic Gross")+ylab("Gross
Revenue")

```

Q: Expected insight #1.

A: A graph is drawn between Awards and Gross Revenue. From the graph we can see that Gross Revenue has a linear relationship between Awards which makes sense because if a movie has high revenue then the chance of getting awards are high.

Q: Expected insight #2.

A: A graph is drawn between Total Gross and Gross revenue. From the graph we can infer that total Gross has a linear relationship with Gross revenue

10. Unexpected insight

Come up with one new insight (backed up by data and graphs) that is unexpected at first glance and do your best to motivate it. Same instructions apply as the previous task.

```{r}
# TODO: Find and illustrate one unexpected insight
ggplot(df2) +
  geom_point(aes(x=Budget,Gross + Domestic_Gross)) +
  geom_smooth(aes(x=Budget, Gross + Domestic_Gross),method="lm") +
  ggtitle(" Budget vs. Total Gross")+xlab("Budget")+ylab("Total Gross")
 ggplot(df2) +
  geom_point(aes(x=Budget, Wins+Nominations)) +
  geom_smooth(aes(x=Budget, Wins+Nominations),method="lm") +
  labs(title="Budget vs. Wins + Nominations",
       x="Budget",
       y="Wins + Nominations")


```

Q: Unexpected insight.

A: Two Graphs are drawn Budget Vs Total Gross and Budget Vs Awards . From the graphs we can infer that Budget is highly correlated with both Total Gross and Awards.