Predicting Film Profitability with Multiple Regression

Demond Love

The Problem Statement

Predicting the profitability of a film is a complex problem, which depends on many factors that can be unforseen at the outset of the filmmaking process. Although many of these factors cannot be accounted for, including competing films or events, cast popularity, economic/political happenings, there are many factors that can not only be accounted for, but also controlled. Therefore, having an accurate gauge on the levers that ultimately control the financial success of a film is highly beneficial to film financiers and filmmakers. As one of the most popular forms of entertainment, filmmaking has always been a highly profitable industry. However, this is also an extremely high-risk industry, built on the back of summer blockbusters that often cost hundreds of millions of dollars. With such large sums of money on the line with every summer blockbuster, it is extremely important that movie studios carefully control the profitability of each and every film.

To this end, I will be researching the relationships between various variables and the revenue that each film generates. Using the TMDb 5000 Movie Dataset, I will use statistical analysis to identify and quantitatively define the these relationships. The ability to predict the financial outlook of planned projects would allow production companies to make smarter decisions, better plan their resources, have an edge in the industry.

The Dataset

I came to this dataset from Kaggle, who received the information from The Movie Database (TMDb) API. TMDb is a community build movie and TV database dating back to 2008, providing movie metadata. This dataset was generated from The Movie Database API. This product uses the TMDb API but is not endorsed or certified by TMDb. Their API also provides access to data on many additional movies, actors and actresses, crew members, and TV shows. The dataset hasn't yet gone through a data quality analysis, wherefore there are multiple missing values throughout the dataset and it is currently unknown if the budgets and revenues are all in US Dollars.

TMDb 5000 Movie Dataset. Retrieved June 24, 2018. https://www.kaggle.com/tmdb/tmdb-movie-metadata

The Approach

The central problem statement that I have chosen to explore is the relationship between film genre, budget, runtime, release date, and financial success. I plan to use correlation analysis to analyze the relationship between the variables given and regression analysis to build a model that will successfully predict the financial success of films. Correlation analysis will help to explore the form, direction, and strength of the relationship between variables. Whereas regression analysis will be used to provide a best fit for film revenue.

Data Cleaning

```
library(readr)
library(dplyr)
library(ggplot2)
library(pastecs)
library(ggm)
library(kableExtra)
library(formattable)
options(scipen=999)
file = read_csv("/Users/Love/Documents/Projects/tmdb-5000-movie-dataset/tmdb_5000_movies.csv")
names(file)
data = data.frame(file$budget, file$genres, file$release_date, file$revenue, file$runtime, file$status,
names(data) = c("budget", "genres", "release_date", "revenue", "runtime", "status", "title", "vote_avera
```

Here I am looking to review the structure of the dataset, ensure all variable have the correct class, explore NAs, and explore the validity of the data within each variable.

```
str(data)
```

```
## 'data.frame':    4803 obs. of  9 variables:
##  $ budget      : int  237000000 300000000 245000000 250000000 260000000 258000000 260000000 28000000
##  $ genres      : Factor w/ 1175 levels "[]","[{\"id\": 10402, \"name\": \"Music\"}, {\"id\": 10749, "
##  $ release_date: Date, format: "2009-12-10" "2007-05-19" ...
##  $ revenue     : num  2787965087 961000000 880674609 1084939099 284139100 ...
##  $ runtime     : int  162 169 148 165 132 139 100 141 153 151 ...
##  $ status      : Factor w/ 3 levels "Post Production",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ title       : Factor w/ 4800 levels "(500) Days of Summer",..: 381 2653 3186 3614 1906 3198 3364 3
##  $ vote_average: num  7.2 6.9 6.3 7.6 6.1 5.9 7.4 7.3 7.4 5.7 ...
##  $ vote_count  : int  11800 4500 4466 9106 2124 3576 3330 6767 5293 7004 ...
```

Immediately I can tell that there are a number of variables with incorrect data types.

```
data$budget = as.numeric(data$budget)
data$genres = as.character(data$genres)
data$runtime = as.numeric(data$runtime)
data$title = as.character(data$title)
data$vote_count = as.numeric(data$vote_count)
```

```
str(data)
```

```
## 'data.frame':    4803 obs. of  9 variables:
##  $ budget      : num  237000000 300000000 245000000 250000000 260000000 258000000 260000000 28000000
##  $ genres      : chr  "[{\"id\": 28, \"name\": \"Action\"}, {\"id\": 12, \"name\": \"Adventure\"}, {"
##  $ release_date: Date, format: "2009-12-10" "2007-05-19" ...
##  $ revenue     : num  2787965087 961000000 880674609 1084939099 284139100 ...
##  $ runtime     : num  162 169 148 165 132 139 100 141 153 151 ...
##  $ status      : Factor w/ 3 levels "Post Production",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ title       : chr  "Avatar" "Pirates of the Caribbean: At World's End" "Spectre" "The Dark Knight
##  $ vote_average: num  7.2 6.9 6.3 7.6 6.1 5.9 7.4 7.3 7.4 5.7 ...
##  $ vote_count  : num  11800 4500 4466 9106 2124 ...
```

```
unique(data$status)
```

```
## [1] Released        Post Production Rumored
## Levels: Post Production Released Rumored
```

```
nrow(data)
```

```
## [1] 4803
```

I can remove Post Production and Rumored data. Only Released files are able to allow us to do the analysis we need.

```
datatemp = subset(data, data$status == 'Released')
nrow(datatemp)
```

```
## [1] 4795
```

This removed 8 erroneous records. Next, in order to see accurate user vote analysis, we only need data with vote_average records. So, I can remove records without at least 30 rows of data.

```
datatemp2 = subset(datatemp, vote_count >= 30)
nrow(datatemp2)
```

```
## [1] 3946
```

Next, in order to see accurate revenue analysis, we only need films with revenue data associated to them. So, I can remove records without revenue data.

```
datatemp3 = subset(datatemp2, revenue >= 1)
nrow(datatemp3)
```

```
## [1] 3183
```

Next, in order to see accurate budget analysis, we only need data with budget records. So, I can remove records without budget data.

```
datatemp4 = subset(datatemp3, budget >= 1)
nrow(datatemp4)
```

```
## [1] 3063
```

Next, I will look for NAs.

```
sapply(datatemp4, function(x) {sum(is.na(x))})
```

```
##       budget       genres release_date      revenue      runtime
##            0            0            0            0           28
##       status        title vote_average   vote_count
##            0            0            0            0
```

There are only runtime NAs left in this dataset. In order to perform any type of runtime analysis, these records will need to be removed also.

```
datatemp5 = subset(datatemp4, is.na(runtime) != TRUE & runtime >= 1)
nrow(datatemp5)
```

```
## [1] 3035
```

```
filmsdata = data.frame(datatemp5$title, datatemp5$release_date, datatemp5$runtime,  datatemp5$budget, da
names(filmsdata) = c("title", "release_date", "runtime", "budget", "revenue", "vote_average", "genres")
```

```
str(filmsdata)
```

```
## 'data.frame':    3035 obs. of  7 variables:
##  $ title       : Factor w/ 3034 levels "(500) Days of Summer",..: 246 1668 2005 2280 1197 2014 2124 1
##  $ release_date: Date, format: "2009-12-10" "2007-05-19" ...
##  $ runtime     : num  162 169 148 165 132 139 100 141 153 151 ...
##  $ budget      : num  237000000 300000000 245000000 250000000 260000000 258000000 260000000 280000000
##  $ revenue     : num  2787965087 961000000 880674609 1084939099 284139100 ...
##  $ vote_average: num  7.2 6.9 6.3 7.6 6.1 5.9 7.4 7.3 7.4 5.7 ...
##  $ genres      : Factor w/ 902 levels "[{\"id\": 10402, \"name\": \"Music\"}, {\"id\": 12, \"name\":
```

Looks like I need to update data types again.

```
filmsdata$genres = as.character(filmsdata$genres)
filmsdata$title = as.character(filmsdata$title)
```

Summarize the data

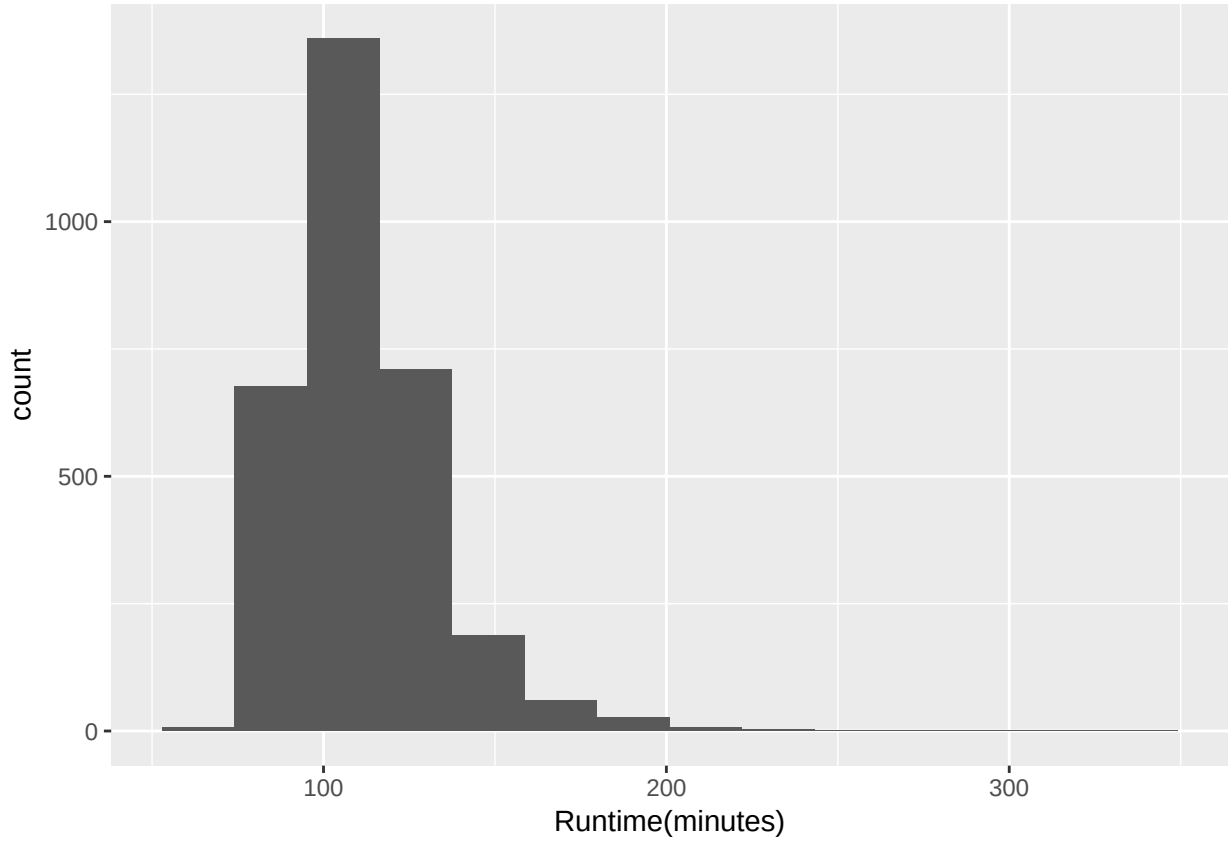
```
summary(filmsdata)
```

```
##     title            release_date          runtime
##  Length:3035        Min.   :1916-09-04   Min.   : 63.0
##  Class :character   1st Qu.:1999-01-25   1st Qu.: 97.0
##  Mode  :character   Median :2005-09-16   Median :107.0
##                     Mean   :2002-09-09   Mean   :111.2
##                     3rd Qu.:2011-01-21   3rd Qu.:121.0
##                     Max.   :2016-09-09   Max.   :338.0
##      budget             revenue            vote_average
##  Min.   :        1   Min.   :         5   Min.   :3.000
##  1st Qu.: 12000000   1st Qu.:  20547760   1st Qu.:5.800
##  Median : 28000000   Median :  61181942   Median :6.400
##  Mean   : 42577396   Mean   : 127773334   Mean   :6.341
##  3rd Qu.: 58000000   3rd Qu.: 152927858   3rd Qu.:6.900
##  Max.   :380000000   Max.   :2787965087   Max.   :8.500
##     genres
##  Length:3035
##  Class :character
##  Mode  :character
##
##
##
```

Here I am looking to review the basic descriptive statistics related to the dataset and explore any outliers. Nothing looks out of line for runtime or vote average. However, both runtime and revenue are are causes for further exploration since the minimum budget is 1 dollar and the minimum revenue is 5 dollars.

First, I will turn my attention to the normaity of the numical variables.

```
ggplot(filmsdata, aes(filmsdata$runtime)) + geom_histogram(bins = 14) + scale_x_continuous(name="Runtim
```




```
round(stat.desc(filmsdata$runtime, basic = FALSE, norm = TRUE), digits = 2)
```

```
##       median         mean      SE.mean CI.mean.0.95          var
##       107.00       111.18         0.38         0.75       442.85
##      std.dev     coef.var     skewness     skew.2SE     kurtosis
##        21.04         0.19         1.75        19.74         7.56
##     kurt.2SE   normtest.W   normtest.p
##        42.54         0.89         0.00
```

This distibution is extremely skewed. Looking at the runtimes over 200 should shed some light on this.

```
runtimeoutlier = subset(filmsdata, runtime > 200)
runtimeoutlier = data.frame(runtimeoutlier$title, runtimeoutlier$budget, runtimeoutlier$revenue)
names(runtimeoutlier) = c("title", "budget", "revenue")
kable(runtimeoutlier, "latex", caption = "Runtime Outliers", booktabs = T) %>%
    kable_styling(latex_options = c("striped", "hold_position"))
```

Table 1: Runtime Outliers

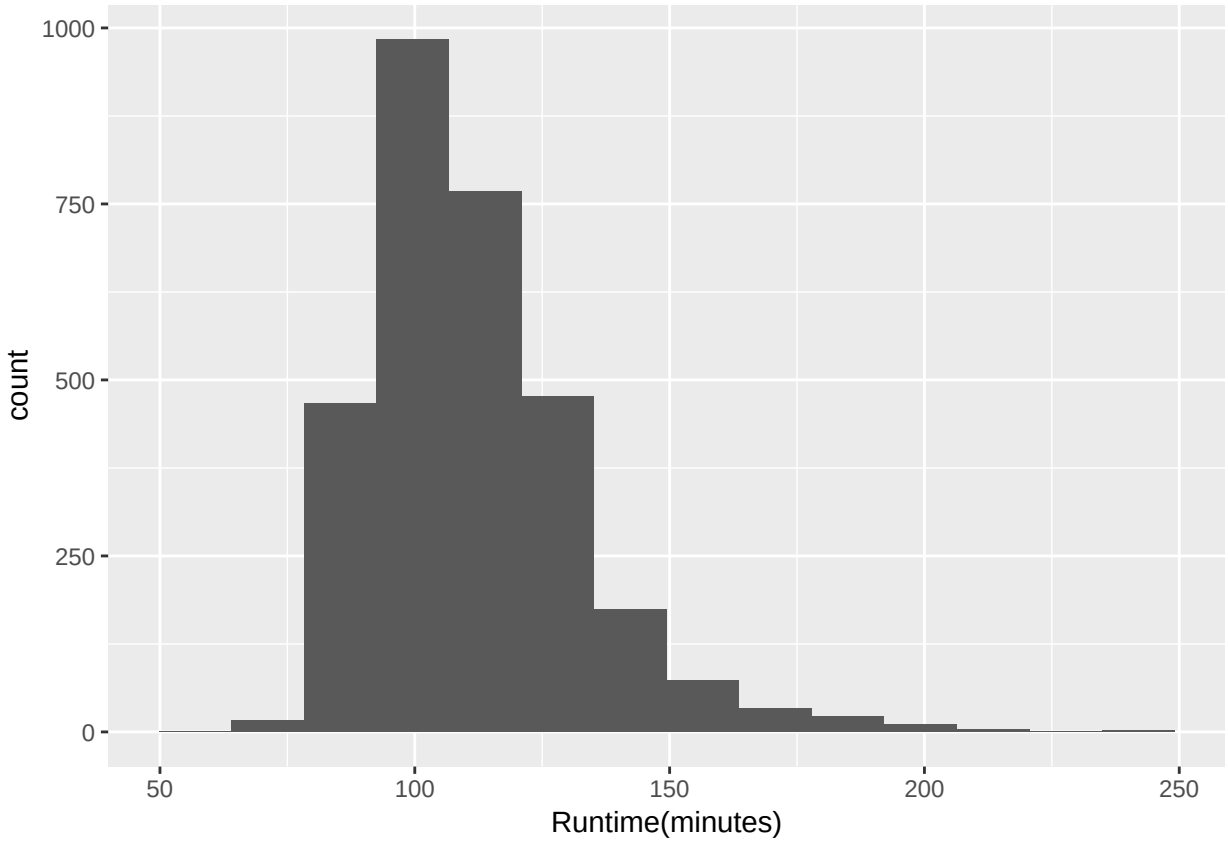
title	budget	revenue
The Lord of the Rings: The Return of the King	94000000	1118888979
Gods and Generals	56000000	12923936
Heaven's Gate	44000000	3484331
Cleopatra	31115000	71000000
Malcolm X	34000000	48169908
Carlos	18000000	871279
Lawrence of Arabia	15000000	69995385
Gone with the Wind	4000000	400176459
Woodstock	600000	34505110
Seven Samurai	2000000	271841

Okay, the longest film in the dataset isn't actually a feature film, but is instead a television miniseries. Therefore, it is not actually from the population in which we are looking to pull from and should be excluded. However, the others do appear to be legitimate feature films with theatrical releases.

```
filmsdata = subset(filmsdata, filmsdata$runtime != 338)
```

Time to rerun the analysis on this column.

```
ggplot(filmsdata, aes(filmsdata$runtime)) + geom_histogram(bins = 14) + scale_x_continuous(name="Runtim
```

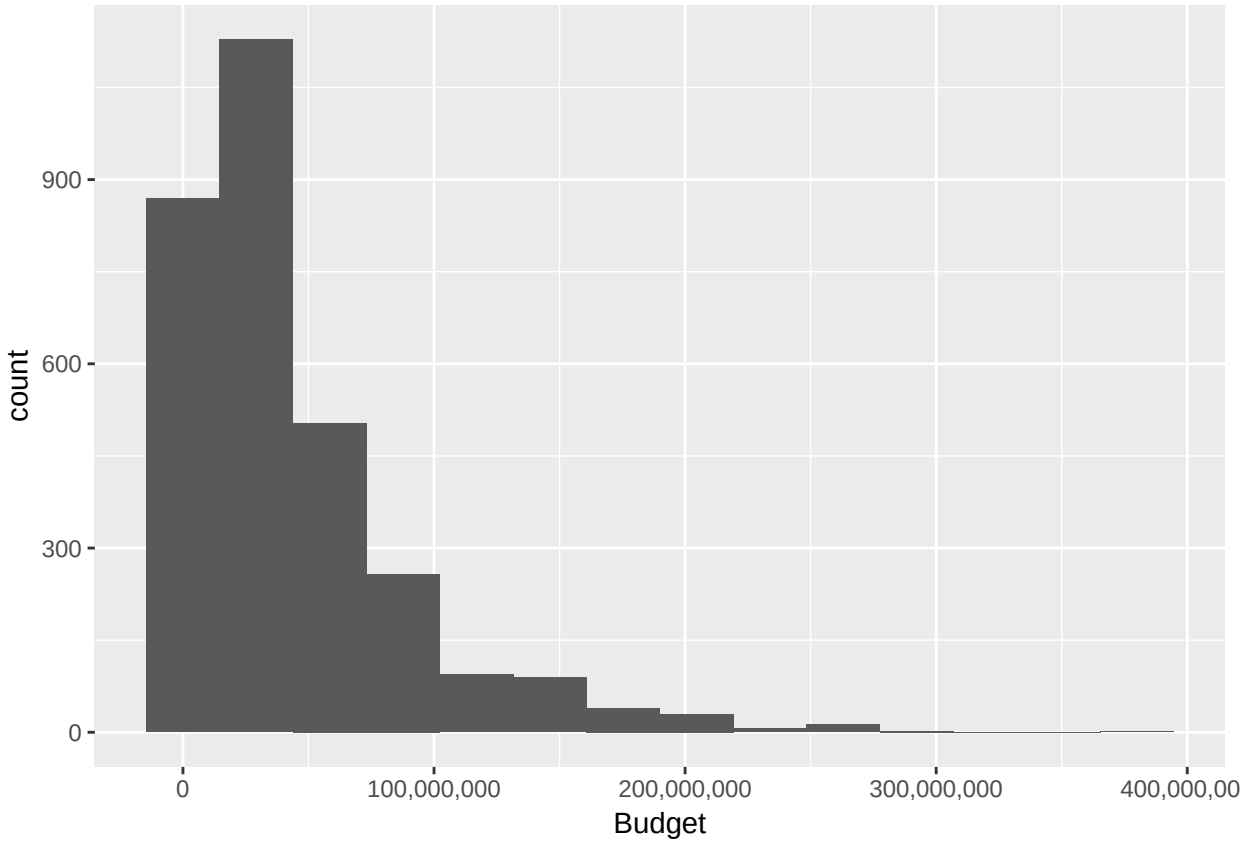


```
round(stat.desc(filmsdata$runtime, basic = FALSE, norm = TRUE), digits = 2)
```

```
##       median         mean      SE.mean CI.mean.0.95          var
##       107.00       111.10         0.37         0.73       426.03
##      std.dev     coef.var     skewness     skew.2SE     kurtosis
##        20.64         0.19         1.43        16.12         3.63
##     kurt.2SE   normtest.W   normtest.p
##        20.41         0.91         0.00
```

This distribution still has statistically significant skewness, but it is a much cleaner dataset that I can definitely work with.

```
ggplot(filmsdata, aes(filmsdata$budget)) + geom_histogram(bins = 14) + scale_x_continuous(name="Budget"
```




```
options(scipen=0)
filmsdata$budget = round(filmsdata$budget, digits = 2)
stat.desc(filmsdata$budget, basic = FALSE, norm = TRUE)
```

```
##       median         mean      SE.mean CI.mean.0.95          var
## 2.800000e+07 4.258550e+07 8.169145e+05 1.601762e+06 2.024738e+15
##      std.dev     coef.var     skewness     skew.2SE     kurtosis
## 4.499709e+07 1.056629e+00 2.030956e+00 2.284637e+01 5.203158e+00
##     kurt.2SE   normtest.W   normtest.p
## 2.927498e+01 7.926284e-01 2.156549e-52
```

```
options(scipen=999)
```

Here, it is a surprising amount of micro-budget movies and a movie that cost almost 400 million dollars. I've heard of Transformers and Avatar films costing 200 million, but this high of a number will need to be investigated.

```
budget_outlier = subset(filmsdata, budget > 200000000)
budget_outlier = data.frame(budget_outlier$title, budget_outlier$budget, budget_outlier$revenue)
names(runtimeoutlier) = c("title", "budget", "revenue")
kable(budget_outlier, "latex", caption = "Budget Outliers", booktabs = T) %>%
    kable_styling(latex_options = c("striped", "hold_position"))
```

Table 2: Budget Outliers

budget_outlier.title	budget_outlier.budget	budget_outlier.revenue
Avatar	237000000	2787965087
Pirates of the Caribbean: At World's End	300000000	961000000
Spectre	245000000	880674609
The Dark Knight Rises	250000000	1084939099
John Carter	260000000	284139100
Spider-Man 3	258000000	890871626
Tangled	260000000	591794936
Avengers: Age of Ultron	280000000	1405403694
Harry Potter and the Half-Blood Prince	250000000	933959197
Batman v Superman: Dawn of Justice	250000000	873260194
Superman Returns	270000000	391081192
The Lone Ranger	255000000	89289910
Man of Steel	225000000	662845518
The Chronicles of Narnia: Prince Caspian	225000000	419651413
The Avengers	220000000	1519557910
Pirates of the Caribbean: On Stranger Tides	380000000	1045713802
Men in Black 3	225000000	624026776
The Hobbit: The Battle of the Five Armies	250000000	956019788
The Amazing Spider-Man	215000000	752215857
The Hobbit: The Desolation of Smaug	250000000	958400000
King Kong	207000000	550000000
Captain America: Civil War	250000000	1153304495
Battleship	209000000	303025485
X-Men: The Last Stand	210000000	459359555
Transformers: Age of Extinction	210000000	1091405097
X-Men: Days of Future Past	250000000	747862775
The Hobbit: An Unexpected Journey	250000000	1021103568

Only one record in the dataset that looks out of place is Pirates of the Caribbean: On Stranger Tides. It is recorded as 380 million dollars. However, other sources, do validate this amount. According to Forbes, it is actually 410 million dollars.

Source: https://www.forbes.com/sites/csylt/2014/07/22/fourth-pirates-of-the-caribbean-is-most-expensive-movie-ever-with-co
#70a84bd8364f

So, let's look at the other side of the ledger.

```
budget_outlier = subset(filmsdata, budget < 1000)
budget_outlier = data.frame(budget_outlier$title, budget_outlier$budget, budget_outlier$revenue)
names(budget_outlier) = c("title", "budget", "revenue")
kable(budget_outlier, "latex", caption = "Budget Outliers", booktabs = T) %>%
    kable_styling(latex_options = c("striped", "hold_position"))
```

Table 3: Budget Outliers

title	budget	revenue
Rugrats in Paris: The Movie	30	103
The 51st State	28	14
Angela's Ashes	25	13
Nurse 3-D	10	10000000
Split Second	7	5
Modern Times	1	8500000
The Prophecy	8	16

After doing more research, none of these movies are legitimate. However, they are conversions not to the dollar, but instead 10 means 10 million for Nurse 3-D, according to The Guardian. Same with Rugrats, according to BoxOfficeMojo. According to the same source, the film actually only made 5k on limited release. The same million unit conversion also holds true for Rugrats revenue. Same for The 51st State, Split Second, and The Prophecy. However, according to BoxOfficeMojo, this isn't true for Angela's Ashes. In fact, it cost 50 million dollars to make and made $13,042,112. However, Modern times is completely off and can be removed.

https://www.theguardian.com/film/2015/jul/21/paz-de-la-huerta-sues-director-of-sex-horror-nurse-3d-for-ruining-her-career
https://web.archive.org/web/20140401011944/http://boxofficemojo.com/movies/?id=rugratsinparis.htm
https://www.the-numbers.com/movie/Angelas-Ashes#tab=summary https://www.the-numbers.com/movie/Split-Second#tab=summary http://www.boxofficemojo.com/movies/?id=prophecy95.htm
http://www.boxofficemojo.com/movies/?id=angelasashes.htm

```
budget_update = subset(filmsdata, title == 'Nurse 3-D')
budget_update = data.frame(budget_update$title, budget_update$budget, budget_update$revenue)
names(budget_update) = c("title", "budget", "revenue")
kable(budget_update, "latex", caption = "Budget Outlier", booktabs = T) %>%
    kable_styling(latex_options = c("striped", "hold_position"))
```

Table 4: Budget Outlier

title	budget	revenue
Nurse 3-D	10	10000000

```
filmsdata[1402,4] = 30000000
filmsdata[1402,5] = 103000000
filmsdata[1568,4] = 28000000
filmsdata[1568,5] = 14000000
filmsdata[2438,4] = 10000000
filmsdata[2438,5] = 5000
filmsdata[3005,4] = 8000000
filmsdata[3005,5] = 16000000
# which(filmsdata$title == 'The Prophecy')
```

```
filmsdata = subset(filmsdata, budget > 1000)
```

```
revenue_outliers = filter(filmsdata, revenue < 1000)
revenue_outliers = data.frame(revenue_outliers$title, revenue_outliers$budget, revenue_outliers$revenue)
names(revenue_outliers) = c("title", "budget", "revenue")
kable(revenue_outliers, "latex", caption = "Revenue Outliers", booktabs = T) %>%
    kable_styling(latex_options = c("striped", "hold_position"))
```

Table 5: Revenue Outliers

title	budget	revenue
Chasing Liberty	23000000	12
Death at a Funeral	9000000	46
In the Cut	12000000	23

These appear to be the same as the budgets earlier. The revenues should be multiplied by 1,000,000. In the Cut should be 4.6 million, per BoxOfficeMojo. Same with Chasing Liberty. Same with Death at a Funeral.

http://www.boxofficemojo.com/movies/?id=inthecut.htm http://www.boxofficemojo.com/movies/?id=chasingliberty.htm http://www.boxofficemojo.com/movies/?id=deathatafuneral.htm

```
filmsdata[2300,5] = 4600000
filmsdata[1476,5] = 12000000
filmsdata[1790,5] = 46000000
# which(filmsdata$title == "Death at a Funeral")
filmsdata$period = format(as.Date(filmsdata$release_date, format="%Y/%m/%d"),"%m")
```

```
summary(filmsdata)
```

```
##     title            release_date          runtime
##  Length:3031        Min.   :1916-09-04   Min.   : 63.0
##  Class :character   1st Qu.:1999-01-29   1st Qu.: 97.0
##  Mode  :character   Median :2005-09-16   Median :107.0
##                     Mean   :2002-09-18   Mean   :111.1
##                     3rd Qu.:2011-01-21   3rd Qu.:121.0
##                     Max.   :2016-09-09   Max.   :248.0
##      budget            revenue            vote_average
##  Min.   :     7000   Min.   :      5000   Min.   :3.00
##  1st Qu.: 12000000   1st Qu.:  20738564   1st Qu.:5.80
##  Median : 28000000   Median :  61399552   Median :6.40
##  Mean   : 42652721   Mean   : 128000099   Mean   :6.34
##  3rd Qu.: 58400000   3rd Qu.: 152937642   3rd Qu.:6.90
##  Max.   :380000000   Max.   :2787965087   Max.   :8.50
##     genres             period
##  Length:3031        Length:3031
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

Now, these minimums and maximums look better for runtime, budget, and revenue.

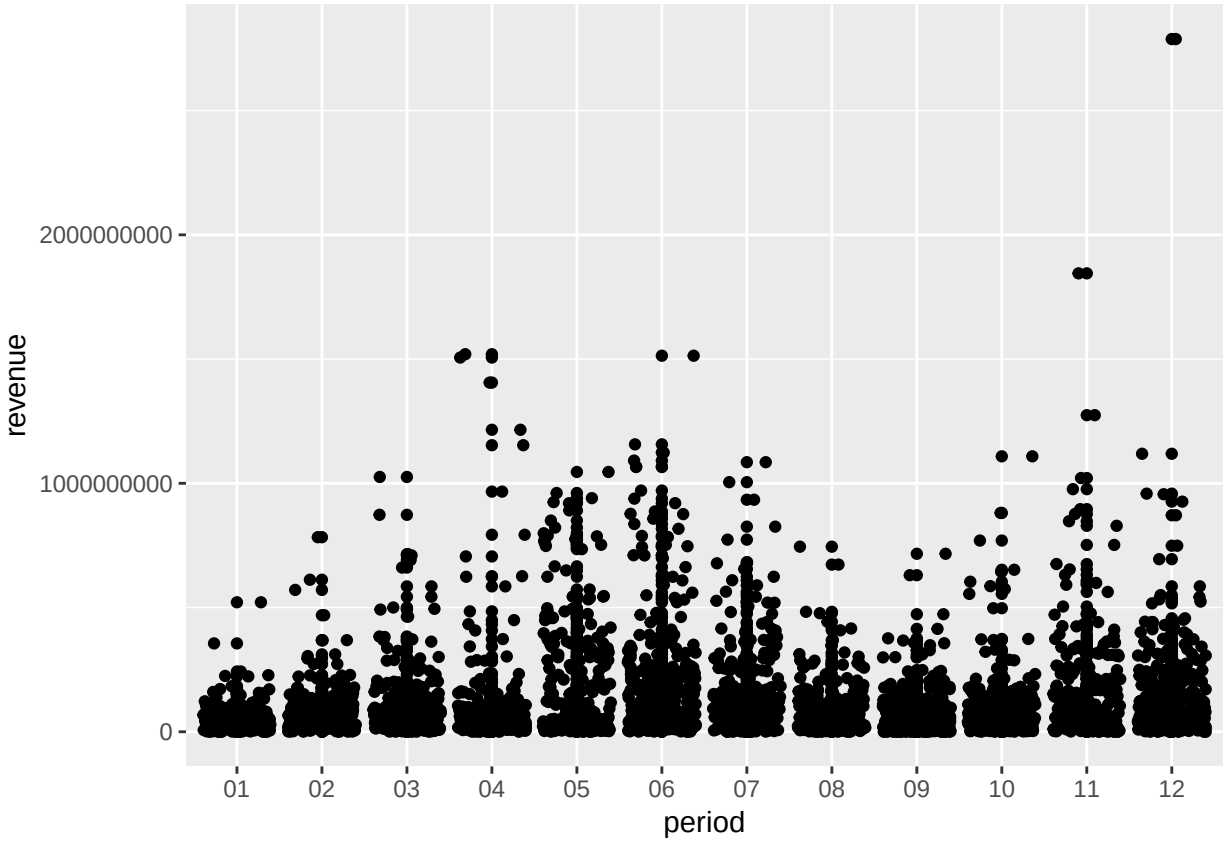
Data Analysis

Time to begin looking for the relationships between the variables and revenue.

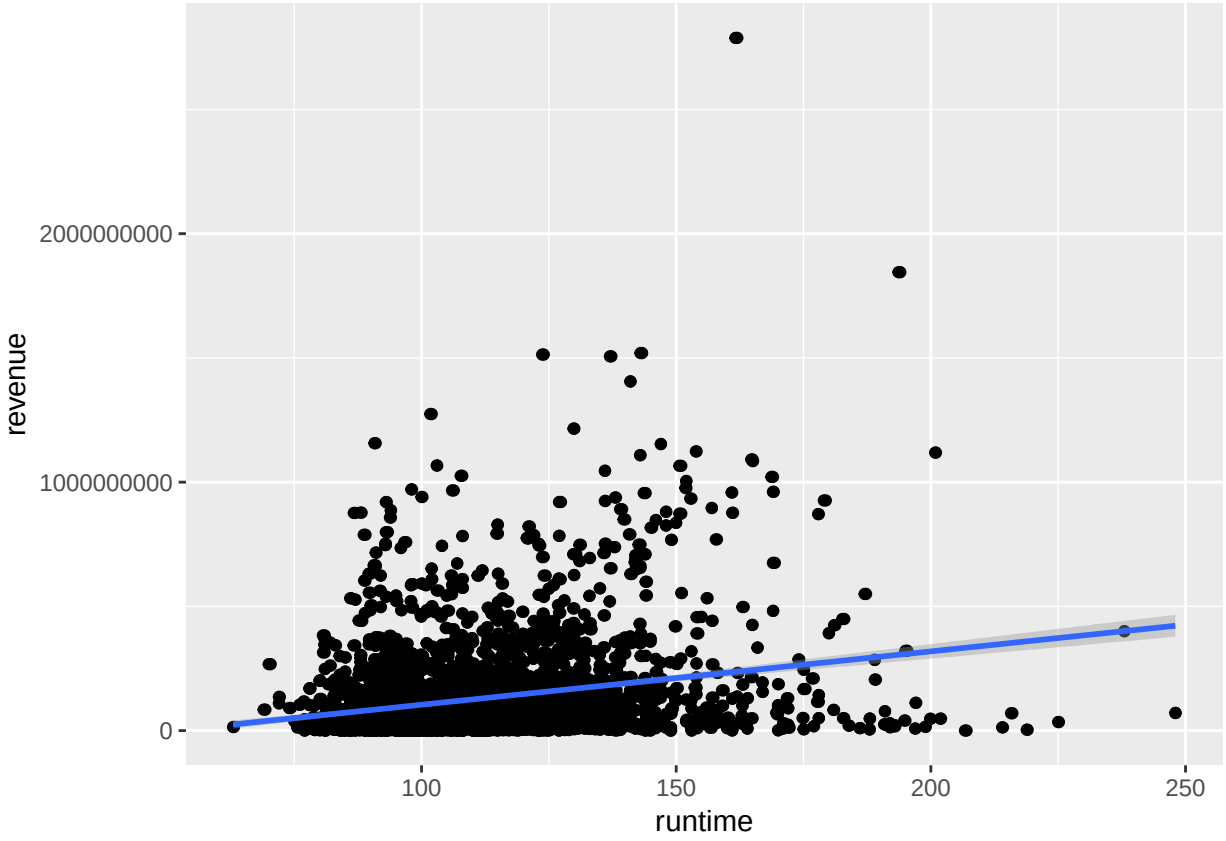
```
numdata = data.frame(filmsdata$revenue, filmsdata$budget, filmsdata$runtime, filmsdata$vote_average)
cor(numdata)
```

```
##                          filmsdata.revenue filmsdata.budget
## filmsdata.revenue                1.0000000       0.70130981
## filmsdata.budget                 0.7013098       1.00000000
## filmsdata.runtime                0.2334138       0.22969252
## filmsdata.vote_average           0.1803765      -0.05473176
##                          filmsdata.runtime filmsdata.vote_average
## filmsdata.revenue                0.2334138             0.18037646
## filmsdata.budget                 0.2296925            -0.05473176
## filmsdata.runtime                1.0000000             0.40467763
## filmsdata.vote_average           0.4046776             1.00000000
```

```
ggplot(data = filmsdata, aes(y = revenue, x = period)) +
    geom_point() +
    geom_jitter()
```



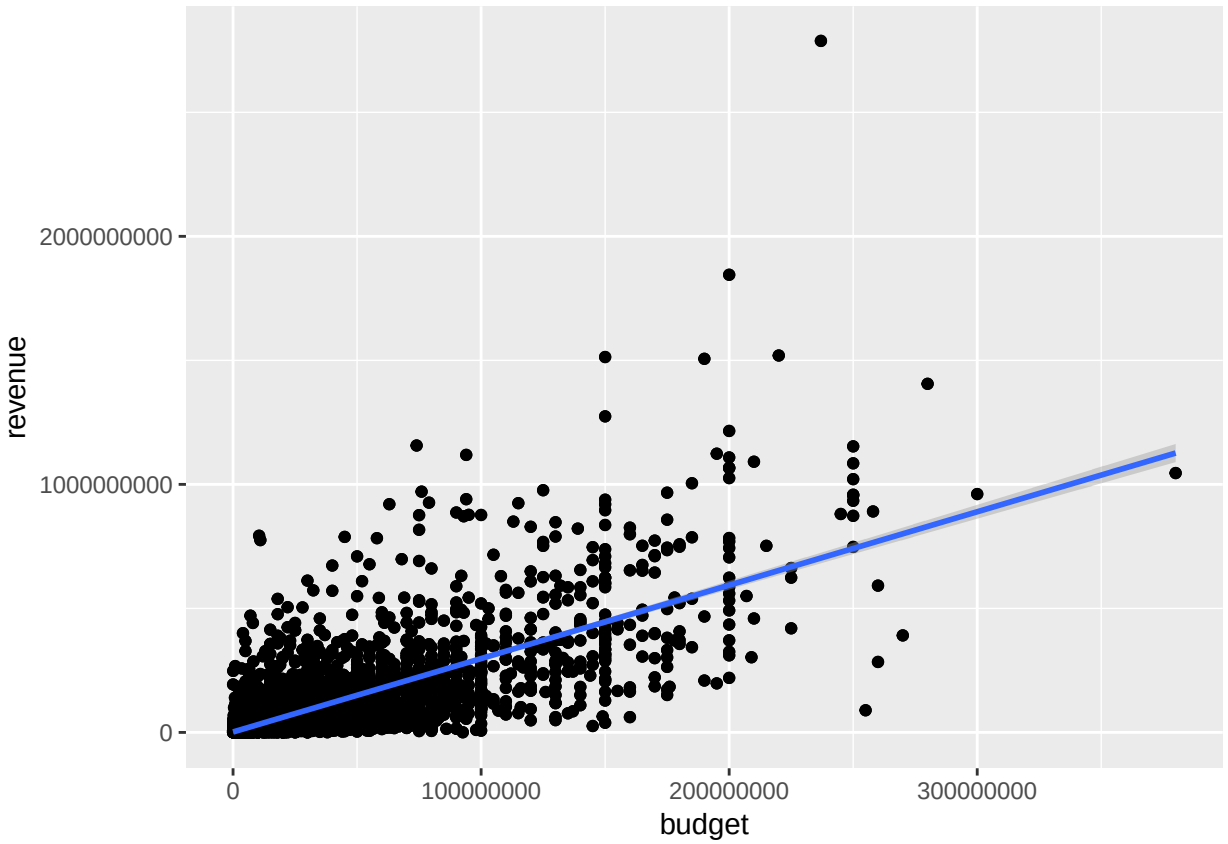

```
ggplot(data = filmsdata, aes(y = revenue, x = runtime)) +
    geom_point() +
    geom_jitter() +
    geom_smooth(method = "lm")
```




```
cor.test(filmsdata$revenue, filmsdata$runtime, method = 'pearson', conf.level = 0.99)
```

```
##
##  Pearson's product-moment correlation
##
## data:  filmsdata$revenue and filmsdata$runtime
## t = 13.211, df = 3029, p-value < 0.00000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 99 percent confidence interval:
##  0.1886981 0.2771635
## sample estimates:
##       cor
## 0.2334138
```

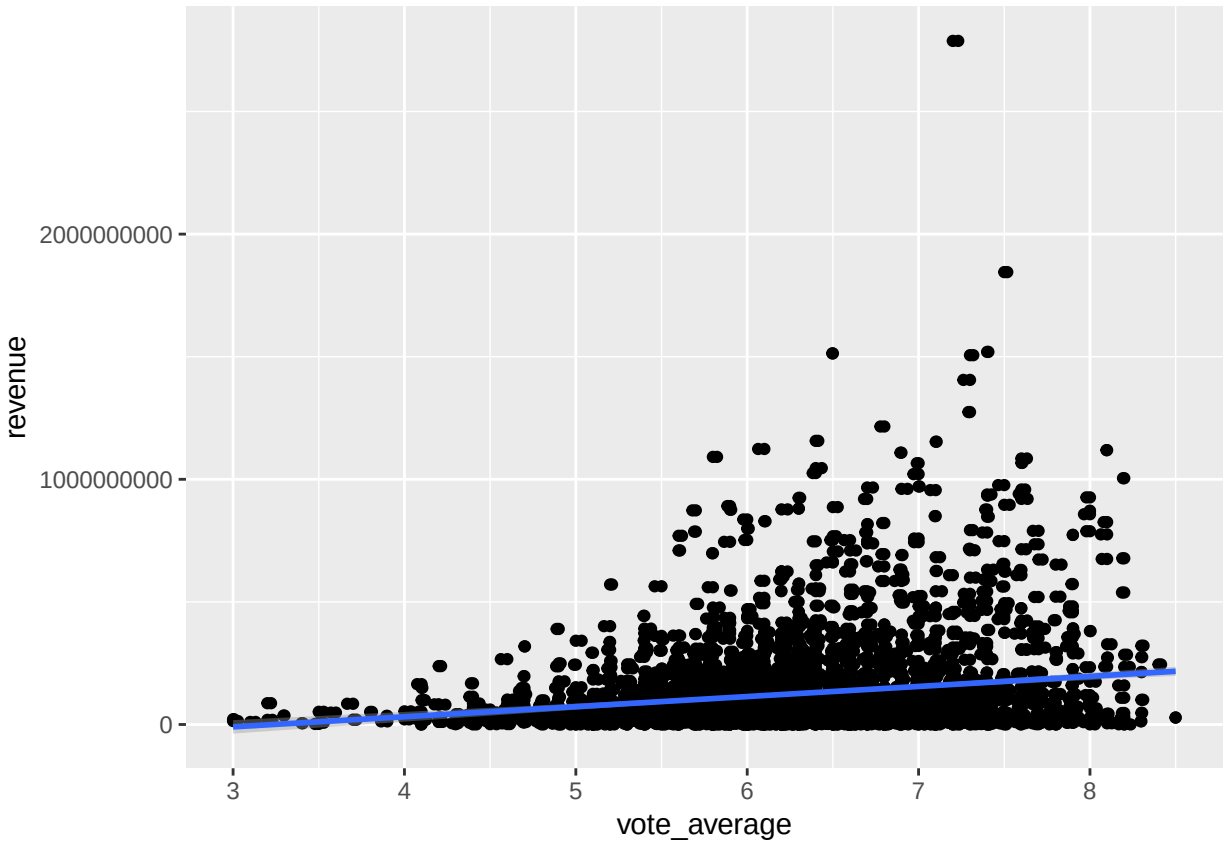
```
ggplot(data = filmsdata, aes(y = revenue, x = budget)) +
    geom_point() +
    geom_jitter() +
    geom_smooth(method = "lm")
```




```
cor.test(filmsdata$revenue, filmsdata$budget, method = 'pearson', conf.level = 0.99)
```

```
##
##  Pearson's product-moment correlation
##
## data:  filmsdata$revenue and filmsdata$budget
## t = 54.145, df = 3029, p-value < 0.00000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 99 percent confidence interval:
##  0.6767338 0.7243247
## sample estimates:
##       cor
## 0.7013098
```

```
ggplot(data = filmsdata, aes(y = revenue, x = vote_average)) +
    geom_point() +
    geom_jitter() +
    geom_smooth(method = "lm")
```




```
cor.test(filmsdata$revenue, filmsdata$vote_average, method = 'pearson', conf.level = 0.99)
```

```
##
##  Pearson's product-moment correlation
##
## data:  filmsdata$revenue and filmsdata$vote_average
## t = 10.093, df = 3029, p-value < 0.00000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 99 percent confidence interval:
##  0.1347374 0.2252518
## sample estimates:
##       cor
## 0.1803765
```

```
revenueregressionmodel = lm(revenue ~ budget + runtime + vote_average, filmsdata)
summary(revenueregressionmodel)
```

```
##
## Call:
## lm(formula = revenue ~ budget + runtime + vote_average, data = filmsdata)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -651735943  -59876966  -15390652   36271639 2036319807
##
## Coefficients:
##                       Estimate       Std. Error t value            Pr(>|t|)
## (Intercept)  -308944122.36549   19005984.82658 -16.255 <0.0000000000000002
```

```
## budget               3.03652          0.05418  56.045 <0.0000000000000002
## runtime         -228890.59649     128973.20253  -1.775                0.076
## vote_average   52501874.45220    3121692.97825  16.818 <0.0000000000000002
##
## (Intercept)  ***
## budget       ***
## runtime      .
## vote_average ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 128800000 on 3027 degrees of freedom
## Multiple R-squared:  0.5403, Adjusted R-squared:  0.5399
## F-statistic:  1186 on 3 and 3027 DF,  p-value: < 0.00000000000000022
```

```
revenueregressionmodel = lm(revenue ~ budget + vote_average, filmsdata)
summary(revenueregressionmodel)
```

```
##
## Call:
## lm(formula = revenue ~ budget + vote_average, data = filmsdata)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -655824333  -59830426  -15052942   36638028 2031870747
##
## Coefficients:
##                      Estimate      Std. Error t value            Pr(>|t|)
## (Intercept)  -318163503.5207   18288746.2089  -17.40 <0.0000000000000002
## budget                3.0100          0.0521   57.78 <0.0000000000000002
## vote_average   50123198.9411    2820313.2246   17.77 <0.0000000000000002
##
## (Intercept)  ***
## budget       ***
## vote_average ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 128800000 on 3028 degrees of freedom
## Multiple R-squared:  0.5398, Adjusted R-squared:  0.5395
## F-statistic:  1776 on 2 and 3028 DF,  p-value: < 0.00000000000000022
```

What I really care about is profit (revenue - budget). Measuring these variables against a film's profit will be the most enlightening, since this i ultimately what matters here.

```
filmsdata$profit = filmsdata$revenue-filmsdata$budget
```

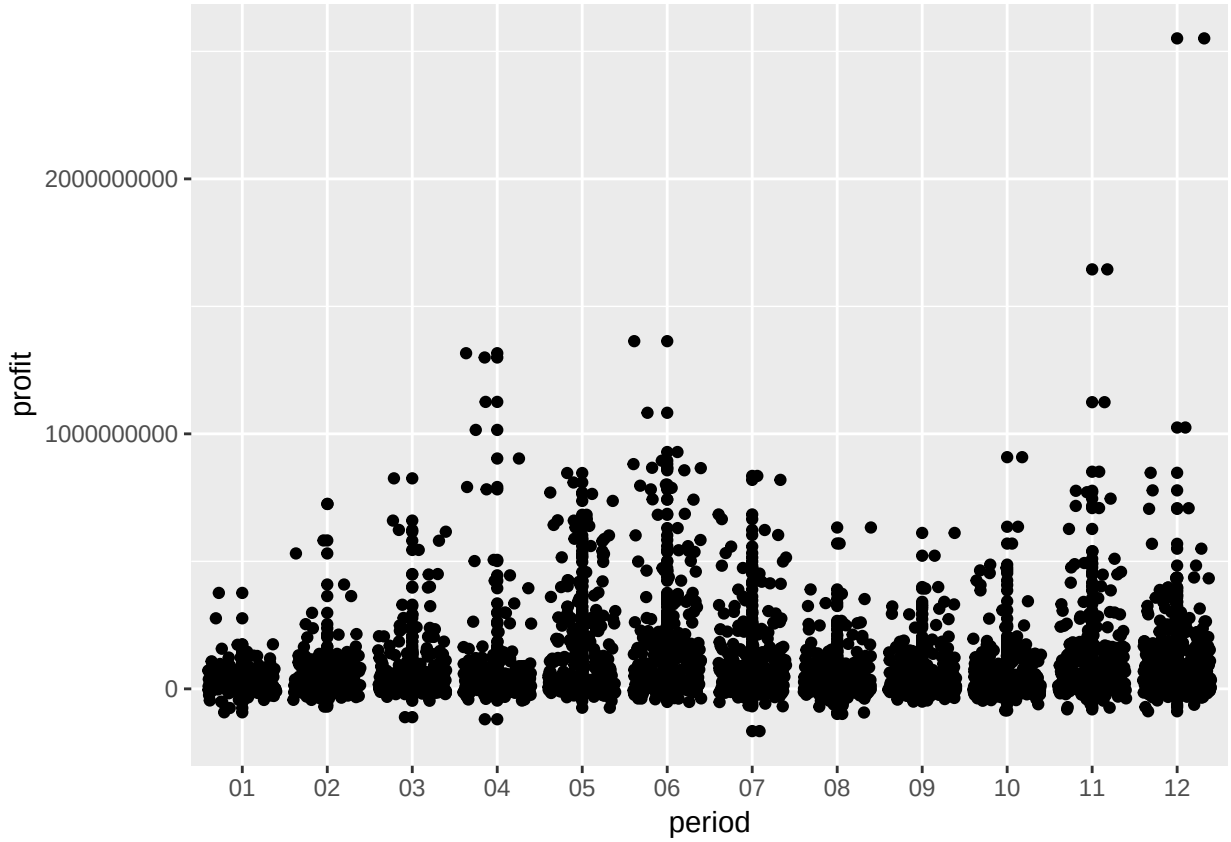
```
numdata = data.frame(filmsdata$profit, filmsdata$budget, filmsdata$runtime, filmsdata$vote_average)
cor(numdata)
```

```
##                        filmsdata.profit filmsdata.budget filmsdata.runtime
## filmsdata.profit             1.0000000       0.54578984         0.2103814
## filmsdata.budget             0.5457898       1.00000000         0.2296925
## filmsdata.runtime            0.2103814       0.22969252         1.0000000
## filmsdata.vote_average       0.2272683      -0.05473176         0.4046776
##                        filmsdata.vote_average
```

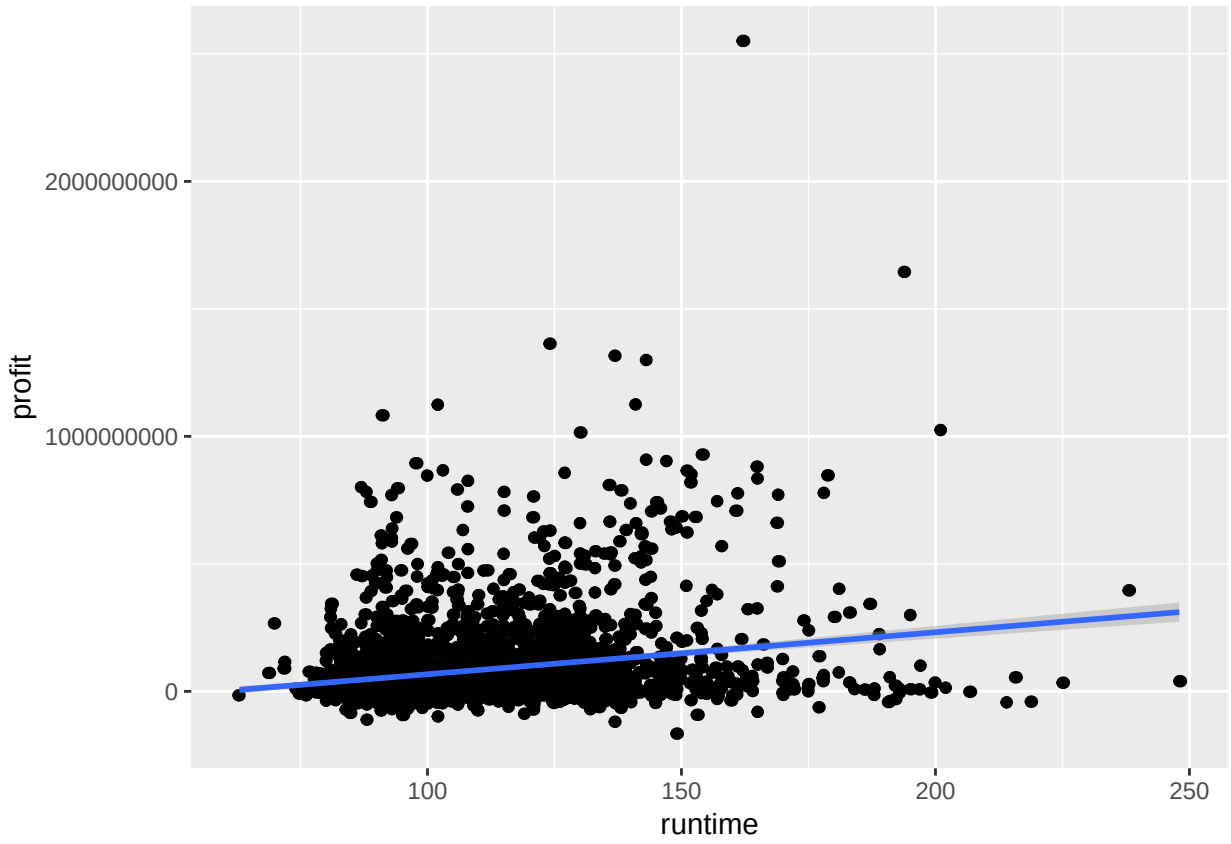
```
## filmsdata.profit                  0.22726826
## filmsdata.budget                 -0.05473176
## filmsdata.runtime                 0.40467763
## filmsdata.vote_average            1.00000000
```

‘‘‘

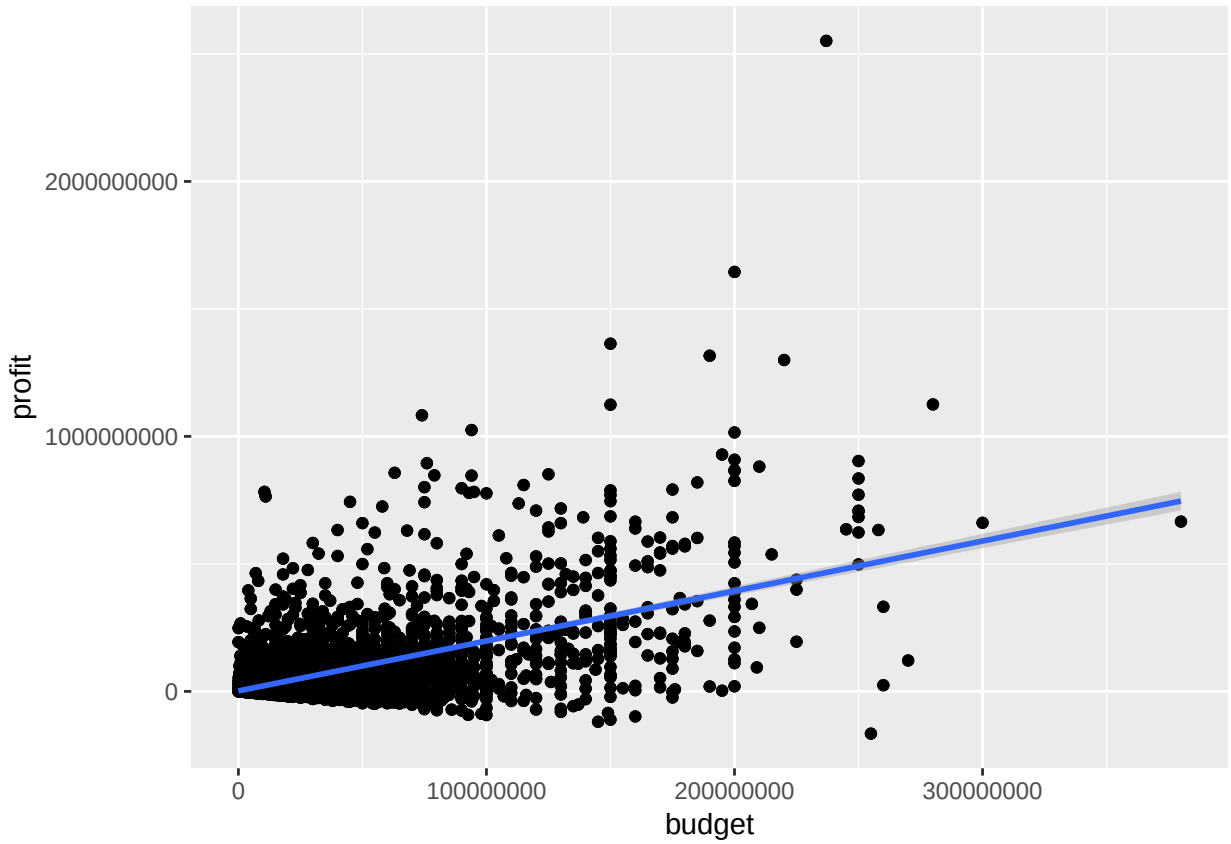
```
ggplot(data = filmsdata, aes(y = profit, x = period)) +
    geom_point() +
    geom_jitter()
```



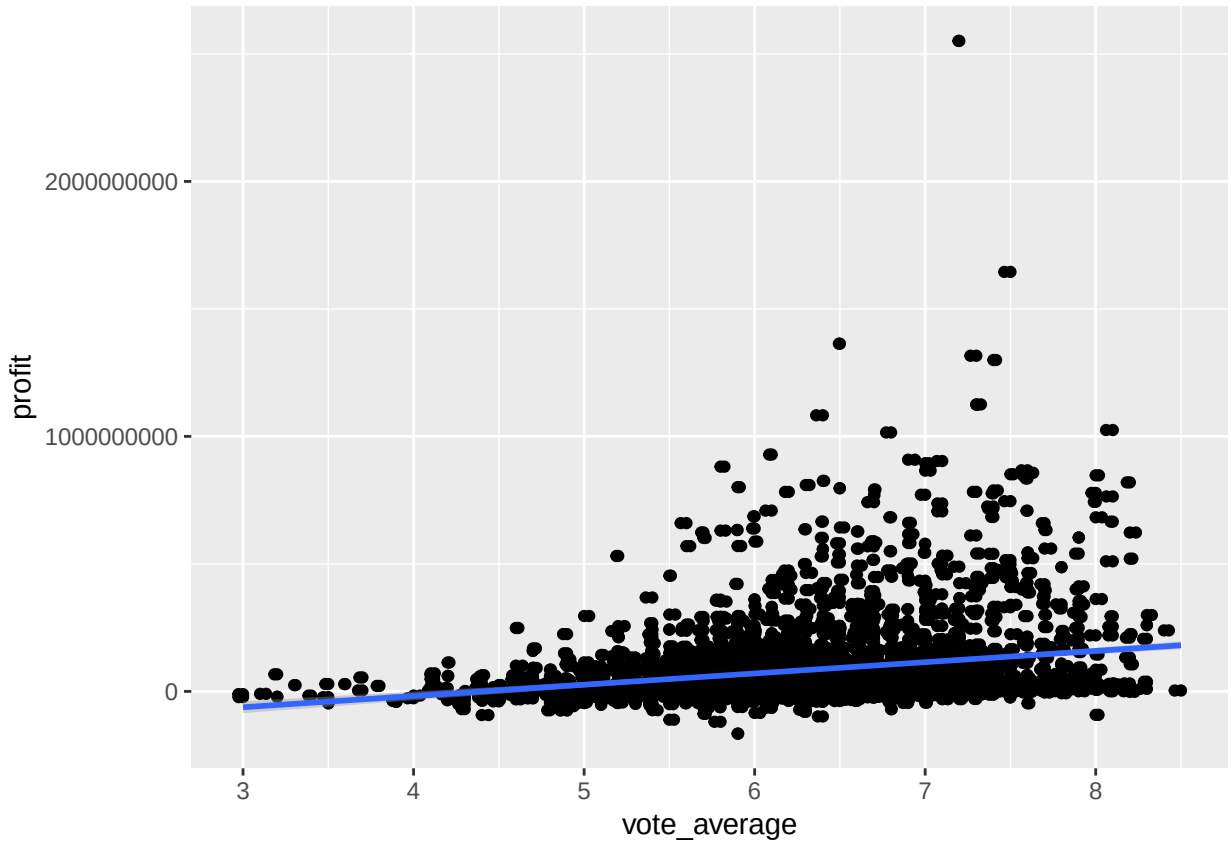

```
ggplot(data = filmsdata, aes(y = profit, x = runtime)) +
    geom_point() +
    geom_jitter() +
    geom_smooth(method = "lm")
```




```
ggplot(data = filmsdata, aes(y = profit, x = budget)) +
    geom_point() +
    geom_jitter() +
    geom_smooth(method = "lm")
```




```
ggplot(data = filmsdata, aes(y = profit, x = vote_average)) +
    geom_point() +
    geom_jitter() +
    geom_smooth(method = "lm")
```




```
profitregressionmodel = lm(profit ~ budget + vote_average + runtime, filmsdata)
summary(profitregressionmodel)
```

```
##
## Call:
## lm(formula = profit ~ budget + vote_average + runtime, data = filmsdata)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -651735943  -59876966  -15390652   36271639 2036319807
##
## Coefficients:
##                      Estimate       Std. Error t value            Pr(>|t|)
## (Intercept)  -308944122.36549   19005984.82658 -16.255 <0.0000000000000002
## budget               2.03652          0.05418  37.588 <0.0000000000000002
## vote_average  52501874.45220    3121692.97825  16.818 <0.0000000000000002
## runtime        -228890.59649     128973.20253  -1.775               0.076
##
## (Intercept)  ***
## budget       ***
## vote_average ***
## runtime      .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 128800000 on 3027 degrees of freedom
## Multiple R-squared:  0.3649, Adjusted R-squared:  0.3642
```

```
## F-statistic: 579.6 on 3 and 3027 DF,  p-value: < 0.00000000000000022
```

```
profitregressionmodel = lm(profit ~ budget + vote_average, filmsdata)
summary(profitregressionmodel)
```

```
##
## Call:
## lm(formula = profit ~ budget + vote_average, data = filmsdata)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -655824333  -59830426  -15052942   36638028 2031870747
##
## Coefficients:
##                     Estimate      Std. Error t value            Pr(>|t|)
## (Intercept)  -318163503.5207   18288746.2089  -17.40 <0.0000000000000002
## budget                2.0100          0.0521   38.58 <0.0000000000000002
## vote_average   50123198.9411    2820313.2246   17.77 <0.0000000000000002
##
## (Intercept)  ***
## budget       ***
## vote_average ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 128800000 on 3028 degrees of freedom
## Multiple R-squared:  0.3642, Adjusted R-squared:  0.3638
## F-statistic: 867.3 on 2 and 3028 DF,  p-value: < 0.00000000000000022
```

Summarization

This analysis above, beginning on page 11, shows the relationships among the core indicating factors of budget, average rating, and runtime on the financial success of a film. This analysis is done on both revenue and profit (revenue-budget).

Beginning with the correlation matrix, we can see the strengths of the relationships among the variables. Each correlation coefficient in the matrix can range between -1 and 1, with higher positive numbers meaning a closer relationship between the two variables, lower negative numbers meaning an inverse relationship and numbers near zero meaning no relationship. There are really 3 figures that are most interesting in the matrix. The relationships of revenue and budget, the relationship of revenue and vote_average, then lastly the relationship of revenue and runtime.

Revenue and Budget have a high positive value of .70, which implies a strong positive relationship. Meaning that as the budget of a film increases, then so does the revenue the film generates on average. This finding is statistically significant at well-over the < 0.05 confidence level.

Revenue and Average Rating have a low positive value of .23, which implies a weak positive relationship. Meaning that as the average film rating of a film increases, then there is also a small increase in the revenue the film generates on average. This finding is statistically significant at well-over the < 0.05 confidence level.

Revenue and Runtime have an even lower positive value of .18, which implies a weak positive relationship. Meaning that as the runtime of a film increases, then there is not a significant increase in the revenue that the film generates. This finding is statistically significant at well-over the < 0.05 confidence level.

Next I looked at a visualization of the relationship between period and revenue. The scatterplot confirmed the hypothesis that films released in the summer and around Christmas genereate a larger ROI.

Afterwards, I visually represent each of these relationships with a linear you model. This allows me to fact-check these findings against commonsense and each of them checkout. You can see that the gradient of each cooresponds with the strength of its relationship with revenue. Same goes for profit, and while the relationship between revenue and average rating remain consistent, the relationship with budget is significantly less strong with profit than it's relationship with revenue.

Lastly, I plotted a multiple linear regression model for these factors. After plotting both revenue and profit, I found that runtime didn't have statistically regression coefficient for either dependent variable. The coefficient of determination, denoted by the multiple R-squared value, provides a measure of how well the model as a whole explains the values of the dependent variable. The model explains nearly 54% of the variation in revenue, but only 36% of the variation in profit. Since models with more features always explain more variation, the adjusted R-squared value corrects R-squared by penalizing models with a large number of independent variables. For both of our models have similar R-squared and adjusted R-squared values.

Therefore, with this analysis, we can confirm that runtime and quality, denoted by it's average rating, don't have a significant impact on the film's financial success, but the film's budget doesn't have a significant impact. Therefore, it can be said that to maximize ones revenue or profitability, one must make a significant investment in terms of budget. However, if you are to make a strong investment, then more must be known regarding the film in question, since both models only predict at most a mediocre amount of the variability in the film's financial success.