DATA621-FinalProject-SmoothOperators

Rob Hodde, Matt Farris, Jeffrey Burmood, Bin Lin

5/26/2017

Introduction

Abstract

Movies: The quintessential form of story telling that we as humans, have developed thus far. They have become the modern past-time for us, a way to escape the humdrum of everyday life into a fantasy world filled with drama, intrique and delight. Movies have astonding audience for the best part of a century, and with that, have become a vast and lucrative industry. Studios, actors and actresses, directors, and production companies make up just a small part of world of film, and we had hope that looking into some movie data we would be able to find some insight. As avid fans and lovers of all films, we decided that this project would provide us both entertainment, and revelation into a fascinated world.

Problem Description

Our final project will explore, analyze and model a data set containing information on approximately 5,000 movies. The dataset contains movie data extracted from the IMDB website and is available on Kaggle.com.

The project will develop predictive models for three questions:

1) Will the movie make money or lose money?
2) What is the anticipated gross margin (profit) for the movie?
3) Are there any particular genres/keywords that influence profitability?

Data Exploration

Data Exploration

The first part of our project consists of explored our data source. As stated above, it came from Kaggle, a repository/social hub for data analyst like ourselves. The dataset isn't, by any stretch of the imagination

To this point we've removed the data columns for the variables that we will not be using in the analysis. The columns that we will focus on are the following:

```
##  [1] "duration"                "director_facebook_likes"
##  [3] "actor_3_facebook_likes"  "actor_1_facebook_likes"
##  [5] "gross"                   "movie_title"
##  [7] "num_voted_users"         "cast_total_facebook_likes"
##  [9] "facenumber_in_poster"    "content_rating"
## [11] "budget"                  "title_year"
## [13] "actor_2_facebook_likes"  "imdb_score"
```

After exploring the data, we noticed there is a scattering of NAs across the variables. Due to the relatively low number of total NAs, we choose to remove all rows with NAs, leaving 3,828 rows of data.

Furthermore, we noticed aproximately 800 foreign films. Though we would have loved these to be apart of our data source, we realized that the budget and gross variables for these films tended to differ dramatically. We saw that the budget was usually in the currency of the country while the gross tended to be in USA dollars. Because trying to adjust for currency differences across several year, we felt it best to remove this data for simplicity sake. This left us with 3042 movies to analyse, which we felt was more than adequate for the project.

Next we will explore the nature of the data for the variables we will be using in the analysis.

VAR	TYPE
duration	integer
director_facebook_likes	integer
actor_3_facebook_likes	integer
actor_1_facebook_likes	integer
gross	integer
movie_title	character
num_voted_users	integer
cast_total_facebook_likes	integer
facenumber_in_poster	integer
content_rating	character
budget	double
title_year	integer
actor_2_facebook_likes	integer
imdb_score	double

duration	director_facebook_likes	actor_3_facebook_likes	actor_1_facebook_likes	gross
Min. : 37.0	Min. : 0.0	Min. : 0.0	Min. : 0.0	Min. : 703
1st Qu.: 95.0	1st Qu.: 11.0	1st Qu.: 233.0	1st Qu.: 811.2	1st Qu.: 11787482
Median :105.0	Median : 62.0	Median : 467.0	Median : 2000.0	Median : 34264376
Mean :109.5	Mean : 911.3	Mean : 836.2	Mean : 8241.5	Mean : 57651658
3rd Qu.:119.0	3rd Qu.: 235.0	3rd Qu.: 723.0	3rd Qu.: 13000.0	3rd Qu.: 75074326
Max. :330.0	Max. :23000.0	Max. :23000.0	Max. :640000.0	Max. :760505847

movie_title	num_voted_users	cast_total_facebook_likes	facenumber_in_poster	content_rating
Length:3042	Min. : 22	Min. : 0	Min. : 0.000	R :1333
Class :character	1st Qu.: 19117	1st Qu.: 2210	1st Qu.: 0.000	PG-13 :1110
Mode :character	Median : 54462	Median : 4517	Median : 1.000	PG : 472

movie_title	num_voted_users	cast_total_facebook_likes	facenumber_in_poster	content_rating
NA	Mean : 108285	Mean : 12340	Mean : 1.419	G : 70
NA	3rd Qu.: 132124	3rd Qu.: 16904	3rd Qu.: 2.000	Not Rated: 18
NA	Max. :1689764	Max. :656730	Max. :43.000	Unrated : 13
NA	NA	NA	NA	(Other) : 26

budget	title_year	actor_2_facebook_likes	imdb_score
Min. : 218	Min. :1929	Min. : 0.0	Min. :1.600
1st Qu.: 10725000	1st Qu.:1999	1st Qu.: 436.0	1st Qu.:5.800
Median : 25000000	Median :2004	Median : 729.5	Median :6.500
Mean : 40319361	Mean :2003	Mean : 2180.3	Mean :6.383
3rd Qu.: 55000000	3rd Qu.:2010	3rd Qu.: 1000.0	3rd Qu.:7.100
Max. :300000000	Max. :2016	Max. :137000.0	Max. :9.300

We also wanted to investigate the correlations, and we can see that none of the variables have any correlation that we can percieve.

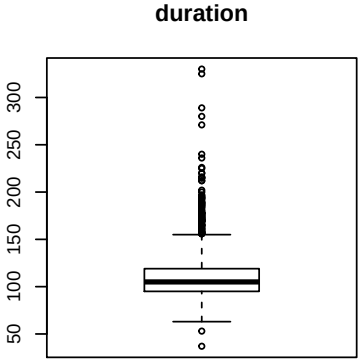
	duration	director_facebook_likes	actor_3_facebook_likes	actor_1_facebook_likes
duration	1.0000000	0.2104197	0.1448777	0.0912903
director_facebook_likes	0.2104197	1.0000000	0.1219467	0.0868426
actor_3_facebook_likes	0.1448777	0.1219467	1.0000000	0.2483043
actor_1_facebook_likes	0.0912903	0.0868426	0.2483043	1.0000000
num_voted_users	0.3705768	0.3190331	0.2818195	0.1741973
cast_total_facebook_likes	0.1349956	0.1172865	0.4830033	0.9459350
facenumber_in_poster	0.0065845	-0.0523321	0.1042739	0.0538466
budget	0.2988689	0.0942904	0.2747815	0.1551897
title_year	-0.1086958	-0.0580504	0.1277213	0.0914452
actor_2_facebook_likes	0.1504159	0.1192872	0.5521997	0.3798140
imdb_score	0.3819342	0.2225461	0.0882029	0.1178984

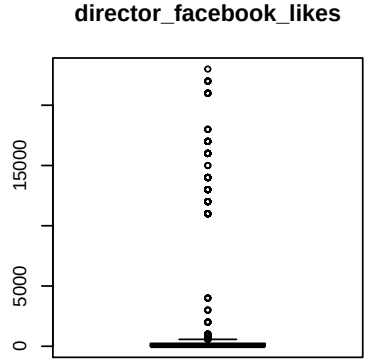
	num_voted_users	cast_total_facebook_likes	facenumber_in_poster	budget
duration	0.3705768	0.1349956	0.0065845	0.2988689
director_facebook_likes	0.3190331	0.1172865	-0.0523321	0.0942904
actor_3_facebook_likes	0.2818195	0.4830033	0.1042739	0.2747815
actor_1_facebook_likes	0.1741973	0.9459350	0.0538466	0.1551897
num_voted_users	1.0000000	0.2486828	-0.0441983	0.4054595

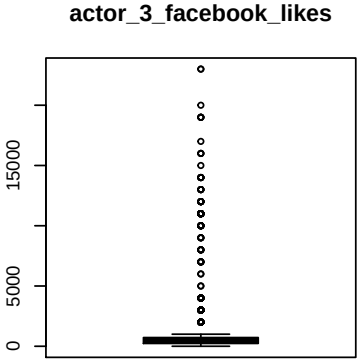
	num_voted_users	cast_total_facebook_likes	facenumber_in_poster	budget
cast_total_facebook_likes	0.2486828	1.0000000	0.0750811	0.2362870
facenumber_in_poster	-0.0441983	0.0750811	1.0000000	-0.0267742
budget	0.4054595	0.2362870	-0.0267742	1.0000000
title_year	0.0241674	0.1256809	0.0873375	0.2412454
actor_2_facebook_likes	0.2524944	0.6319688	0.0625703	0.2526741
imdb_score	0.5089320	0.1377072	-0.0694804	0.0713682

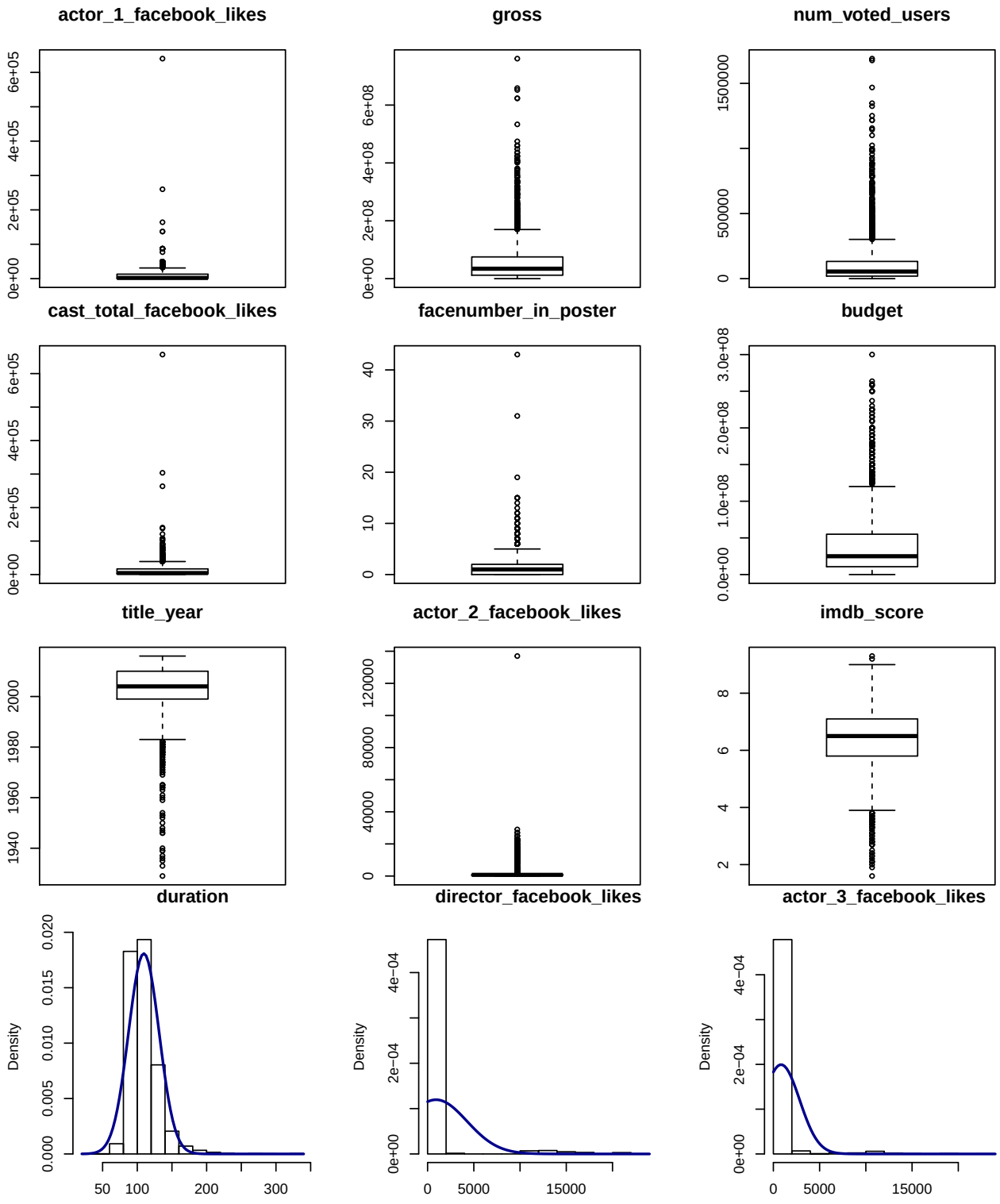
	title_year	actor_2_facebook_likes
duration	-0.1086958	0.1504159
director_facebook_likes	-0.0580504	0.1192872
actor_3_facebook_likes	0.1277213	0.5521997
actor_1_facebook_likes	0.0914452	0.3798140
num_voted_users	0.0241674	0.2524944
cast_total_facebook_likes	0.1256809	0.6319688
facenumber_in_poster	0.0873375	0.0625703
budget	0.2412454	0.2526741
title_year	1.0000000	0.1253783
actor_2_facebook_likes	0.1253783	1.0000000
imdb_score	-0.1504498	0.1274387

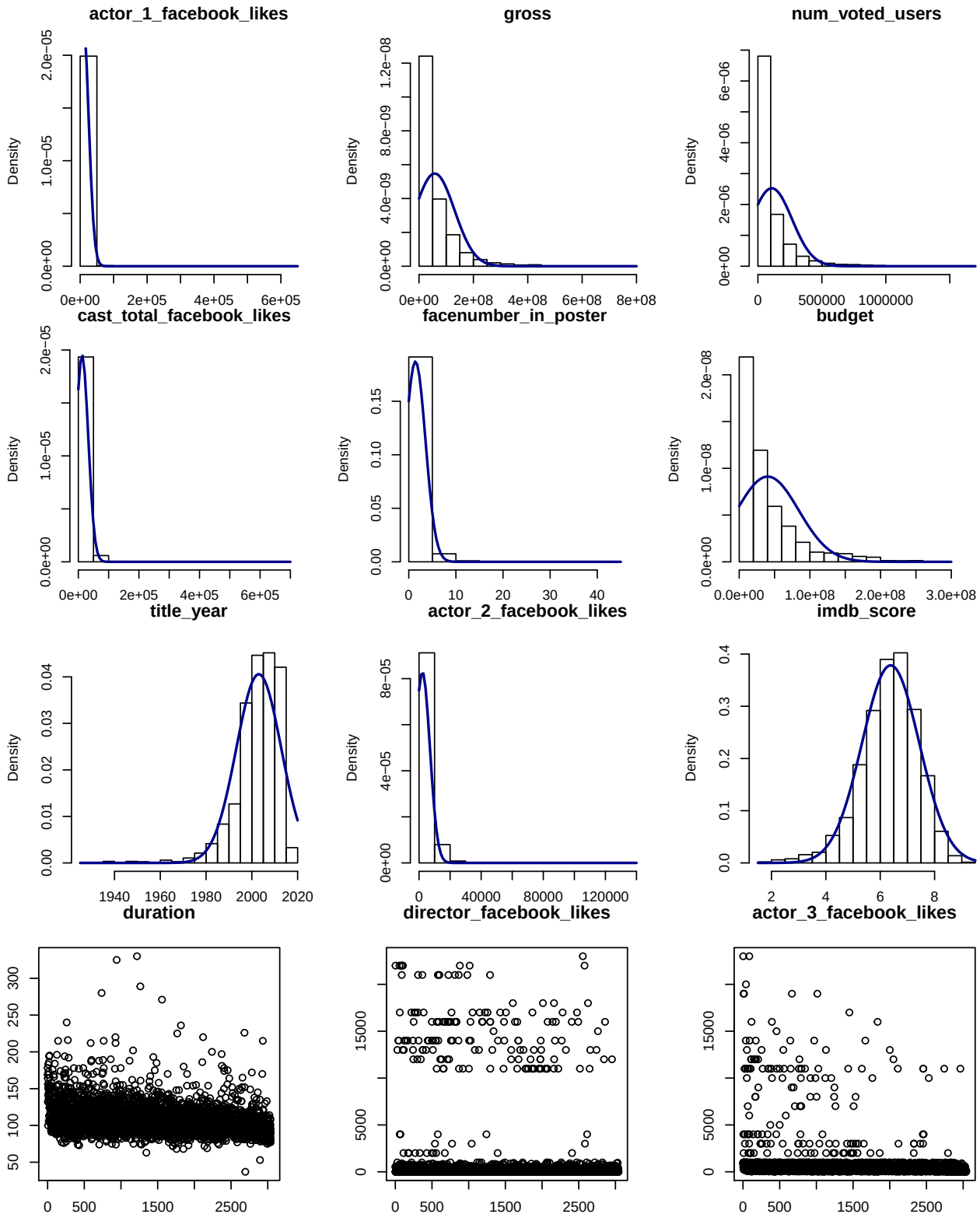
Lastly for exploration, we looked at all our variables through boxplots, histograms, and scatter plots.

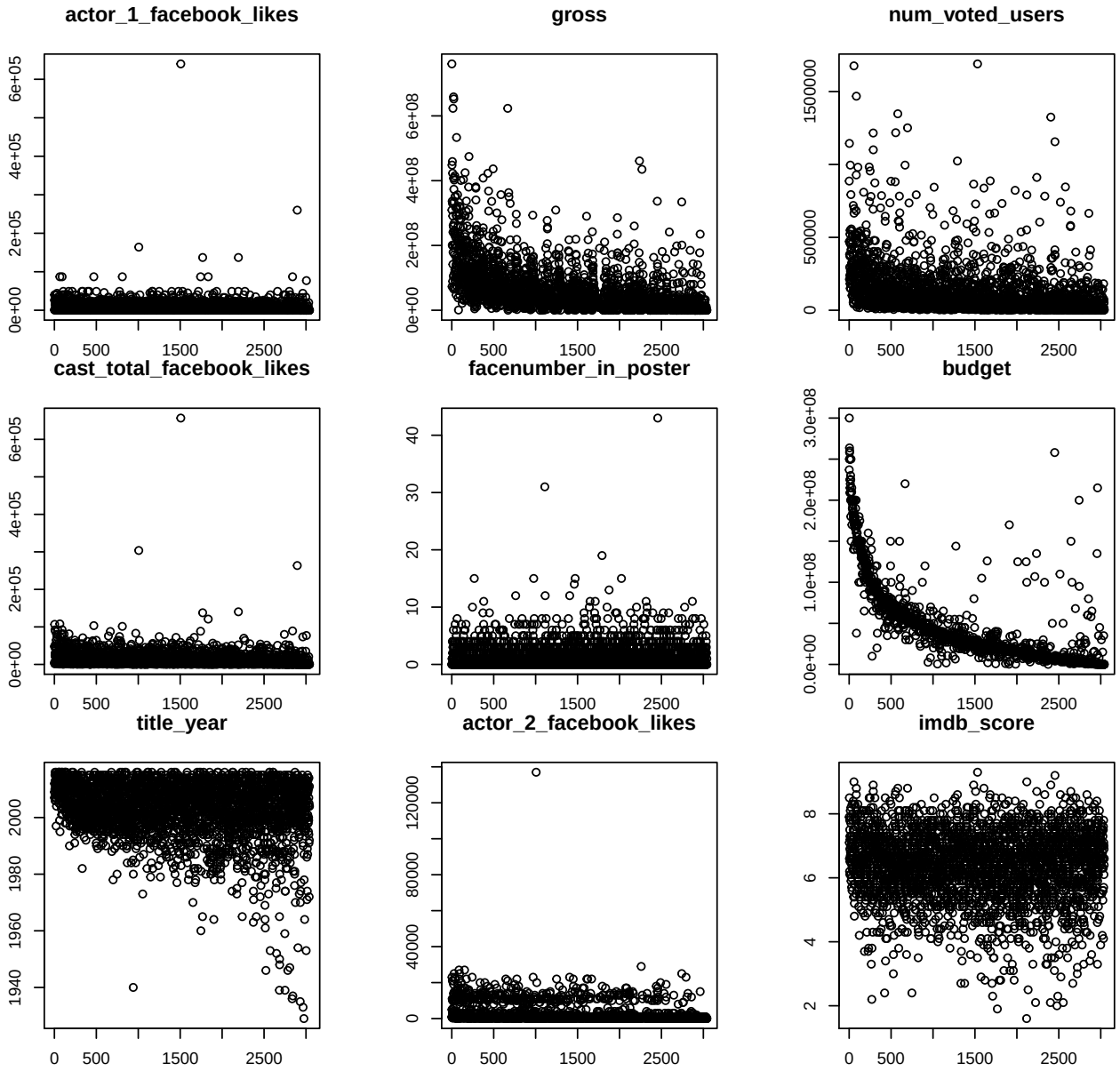







actor_1_facebook_likes
gross
num_voted_users
cast_total_facebook_likes
facenumber_in_poster
budget
title_year
actor_2_facebook_likes
imdb_score
duration
director_facebook_likes
actor_3_facebook_likes
Density
Density
Density


actor_1_facebook_likes
gross
num_voted_users
cast_total_facebook_likes
facenumber_in_poster
budget
title_year
actor_2_facebook_likes
imdb_score
duration
director_facebook_likes
actor_3_facebook_likes
Density



As we can see from the plots and statistical summary, most of the variables have a reasonable distribution except those variable associated with the Facebook likes. There are five variables related to Facebook likes that are highly skewed due to a large number of zeros. While examining the dataset source, they revealed that along of the "zero" values from the facebook likes were caused by simple errors in the scraping. At this point we assume these zeros represent NAs in the Facebook data.

Next, we'll use the mice package to impute the Facebook likes data for the zeros/NAs.

```
##      actor_1_facebook_likes cast_total_facebook_likes
## 2502                      1                         1
##  520                      1                         1
##   10                      1                         1
##    1                      1                         1
##    6                      1                         1
##    2                      1                         1
##    1                      0                         0
```

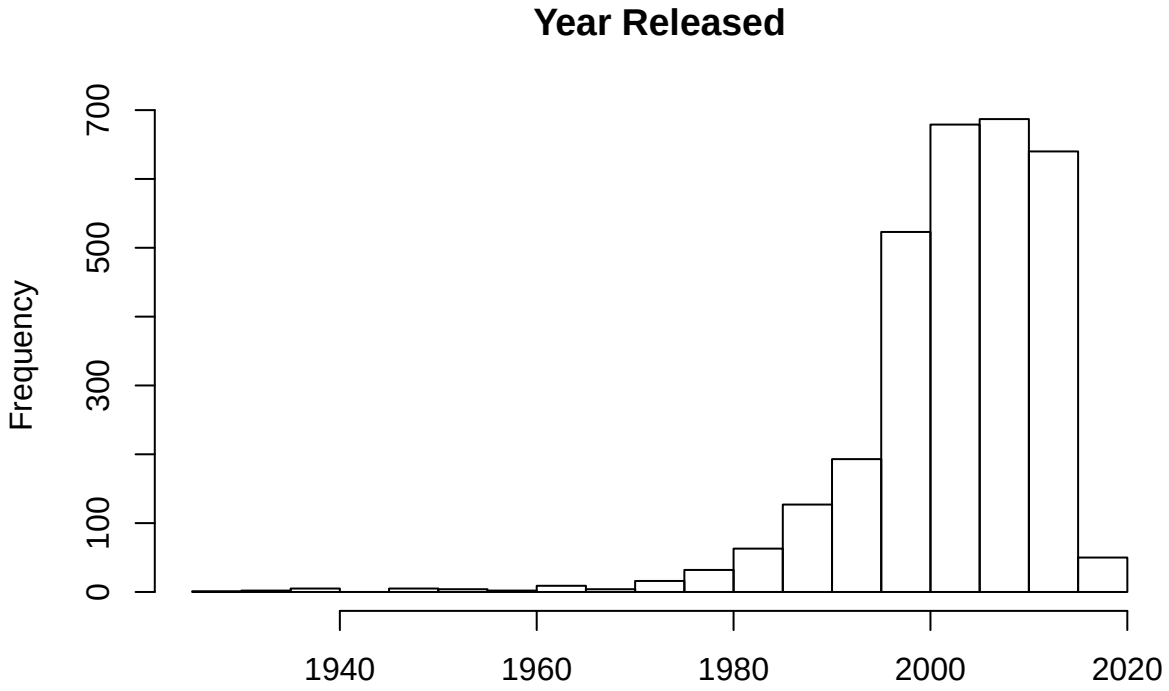
```
##                           1                         1
##      actor_2_facebook_likes actor_3_facebook_likes director_facebook_likes
## 2502                      1                       1                       1
##  520                      1                       1                       0
##   10                      1                       0                       1
##    1                      1                       0                       0
##    6                      0                       0                       1
##    2                      0                       0                       0
##    1                      0                       0                       1
##                           9                      20                     523
##
## 2502   0
##  520   1
##   10   1
##    1   2
##    6   2
##    2   3
##    1   4
##      554


##     duration     director_facebook_likes actor_3_facebook_likes
##  Min.   : 37.0   Min.   :    2           Min.   :    2.0
##  1st Qu.: 95.0   1st Qu.:   32           1st Qu.:  233.0
##  Median :105.0   Median :   99           Median :  467.5
##  Mean   :109.5   Mean   : 1186           Mean   :  836.4
##  3rd Qu.:119.0   3rd Qu.:  304           3rd Qu.:  723.0
##  Max.   :330.0   Max.   :23000           Max.   :23000.0
##
##  actor_1_facebook_likes     gross           movie_title
##  Min.   :     2         Min.   :      703   Length:3042
##  1st Qu.:   812         1st Qu.: 11787482   Class :character
##  Median :  2000         Median : 34264376   Mode  :character
##  Mean   :  8244         Mean   : 57651658
##  3rd Qu.: 13000         3rd Qu.: 75074326
##  Max.   :640000         Max.   :760505847
##
##  num_voted_users   cast_total_facebook_likes facenumber_in_poster
##  Min.   :     22   Min.   :     2            Min.   : 0.000
##  1st Qu.:  19117   1st Qu.:  2212            1st Qu.: 0.000
##  Median :  54462   Median :  4523            Median : 1.000
##  Mean   : 108285   Mean   : 12343            Mean   : 1.419
##  3rd Qu.: 132124   3rd Qu.: 16904            3rd Qu.: 2.000
##  Max.   :1689764   Max.   :656730            Max.   :43.000
##
##    content_rating     budget            title_year
##  R        :1333   Min.   :      218   Min.   :1929
##  PG-13    :1110   1st Qu.: 10725000   1st Qu.:1999
##  PG       : 472   Median : 25000000   Median :2004
##  G        :  70   Mean   : 40319361   Mean   :2003
##  Not Rated:  18   3rd Qu.: 55000000   3rd Qu.:2010
##  Unrated  :  13   Max.   :300000000   Max.   :2016
##  (Other)  :  26
##  actor_2_facebook_likes   imdb_score
##  Min.   :     2.0       Min.   :1.600
```

```
##  1st Qu.:   436.2       1st Qu.:5.800
##  Median :   729.5       Median :6.500
##  Mean   :  2180.4       Mean   :6.383
##  3rd Qu.:  1000.0       3rd Qu.:7.100
##  Max.   :137000.0       Max.   :9.300
##
```

Data Preparation

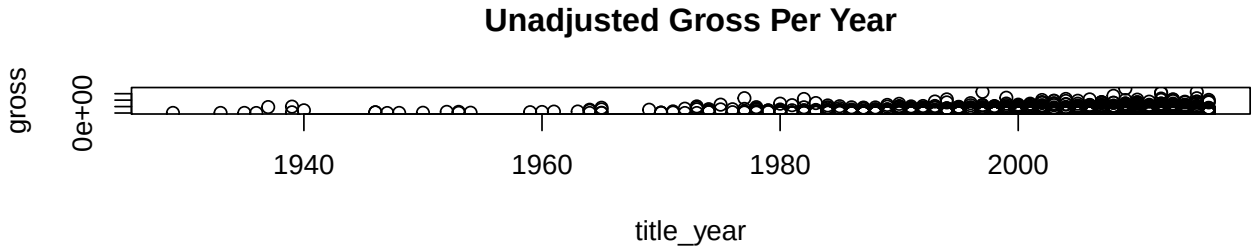
Data Preparation

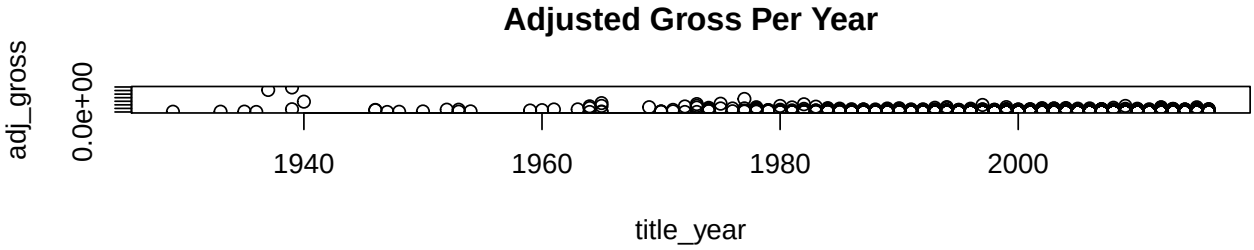
One of the big issues faced when using this dataset is the time frame. These movies were collected over the past 80+ years, and the following shows our distribution over time:



As you can see, the vast majority came from 1990s and above, but we can't discredit the movies from previous year. In order to accurately portray elements from the past, we have instituted a rate of inflation calculation. Using the consumer price index (for our part here we are making a crucial assumption, that all dollars are calculated based on US currency, and we are ignoring even more complex foreign exchange rates of the time), we can calculate the gross value per year. As a basis of comparison, we are using the CPI index from 2016, as the last movie was made in 2016.

```
## The following object is masked _by_ .GlobalEnv:
##
##     cpi
```

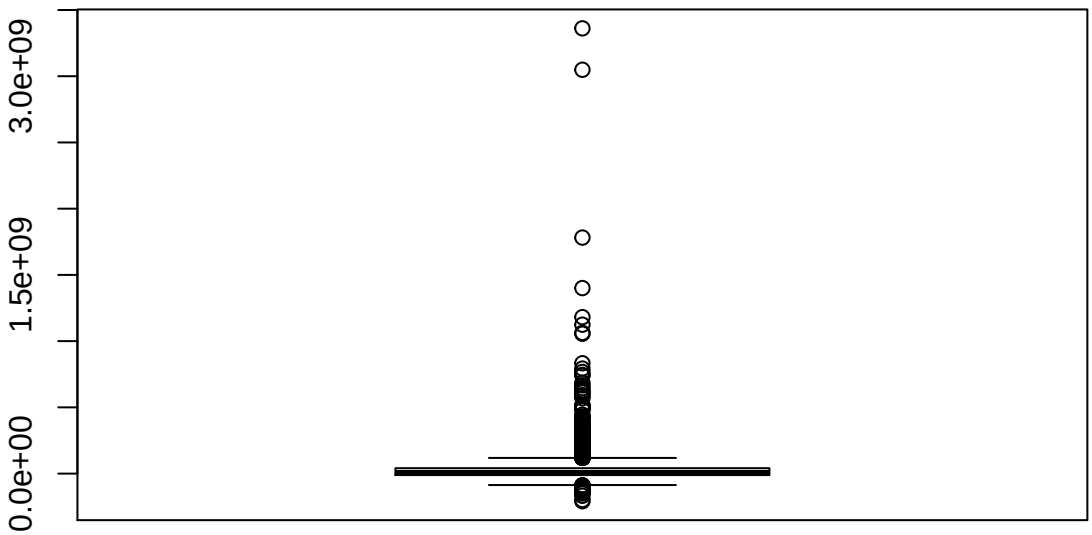




From the above graphs, we can see that the adjustment for the gross did indeed create a more uniformed dataset (where as before we saw movies increasing over the years). As a point of interest, the movies that made over a billion dollars are shown below:

```
##                            movie_title     gross  adj_gross
## 5     Snow White and the Seven Dwarfs 184925485 3082091417
## 7                   Gone with the Wind 198655278 3430019188
## 8                             Pinocchio  84300000 1445142857
## 26                 The Sound of Music 163214286 1243537417
## 39                      The Exorcist 204565000 1105756757
## 48                              Jaws 260000000 1159851301
## 53 Star Wars: Episode IV - A New Hope 460935665 1825487782
## 90         E.T. the Extra-Terrestrial 434949459 1081739587
```

A quick Google search indicates that the above movies are consistently listed as the top grossing movies of all time. Furthermore, our "estimated adjusted gross" mimics the findings that we see with adjusted gross (for the most part, there are two schools of thought on how to adjust gross, using ticket prices or our method adjusting based on CPI). Though our dollar amount vary slightly from other sources, any variance is consistent across our dataset, and would not negatively impact on the overall results.



Build Models

Build Models

Binomial Regression

Our first model we want to investigate is whether or not we can predict if film will make money given the cast and direction. To do this, we decided to create a binary regression model, transforming our adjusted margin into a simple binary: 0 equals a loss of money, 1 equals a proft.

Below we utilized a binomial model with the logistic regression function in R.

```
##
## Call:
## glm(formula = money ~ ., family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.5244  -1.1118   0.5147   1.0642   1.8648
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                8.049e+01  1.067e+01   7.546 4.47e-14 ***
## title_year                -3.972e-02  5.307e-03  -7.484 7.22e-14 ***
## duration                  -1.321e-02  2.469e-03  -5.352 8.68e-08 ***
## director_facebook_likes   -3.137e-05  1.405e-05  -2.232   0.0256 *
## actor_3_facebook_likes    -1.278e-04  7.455e-05  -1.715   0.0864 .
## actor_1_facebook_likes    -1.220e-04  5.009e-05  -2.436   0.0148 *
## num_voted_users            8.666e-06  6.509e-07  13.315  < 2e-16 ***
## cast_total_facebook_likes  1.179e-04  5.004e-05   2.356   0.0185 *
## facenumber_in_poster       3.979e-02  2.221e-02   1.792   0.0732 .
## actor_2_facebook_likes    -1.241e-04  5.237e-05  -2.370   0.0178 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3307.1  on 2432  degrees of freedom
## Residual deviance: 2957.9  on 2423  degrees of freedom
## AIC: 2977.9
##
## Number of Fisher Scoring iterations: 5
```

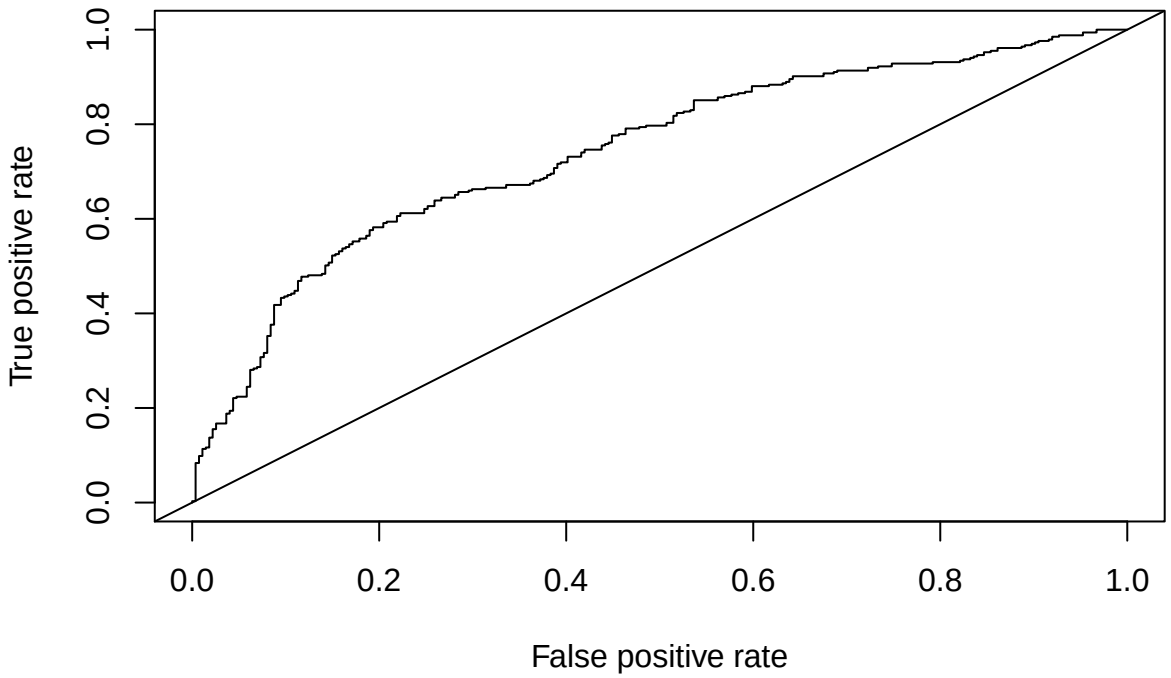
```
## [1] 0.7416058
```

Using all the prediction variables at hand, the model accurately predicts 74% of the time. Using backward stepwise regression, we attempted to remove some variables that may not have had significance in our model.

```
## Start:  AIC=2977.92
## money ~ title_year + duration + director_facebook_likes + actor_3_facebook_likes +
##     actor_1_facebook_likes + num_voted_users + cast_total_facebook_likes +
##     facenumber_in_poster + actor_2_facebook_likes
##
##                              Df Deviance    AIC
## <none>                            2957.9 2977.9
## - actor_3_facebook_likes      1   2960.9 2978.9
## - facenumber_in_poster        1   2961.2 2979.2
## - director_facebook_likes     1   2962.8 2980.8
## - cast_total_facebook_likes   1   2963.8 2981.8
## - actor_2_facebook_likes      1   2963.8 2981.8
## - actor_1_facebook_likes      1   2964.2 2982.2
## - duration                    1   2987.6 3005.6
## - title_year                  1   3020.9 3038.9
## - num_voted_users             1   3241.7 3259.7
```

```
##
## Call:
## glm(formula = money ~ title_year + duration + director_facebook_likes +
##     actor_3_facebook_likes + actor_1_facebook_likes + num_voted_users +
##     cast_total_facebook_likes + facenumber_in_poster + actor_2_facebook_likes,
##     family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.5244  -1.1118   0.5147   1.0642   1.8648
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 8.049e+01  1.067e+01   7.546 4.47e-14 ***
## title_year                 -3.972e-02  5.307e-03  -7.484 7.22e-14 ***
## duration                   -1.321e-02  2.469e-03  -5.352 8.68e-08 ***
## director_facebook_likes    -3.137e-05  1.405e-05  -2.232   0.0256 *
## actor_3_facebook_likes     -1.278e-04  7.455e-05  -1.715   0.0864 .
## actor_1_facebook_likes     -1.220e-04  5.009e-05  -2.436   0.0148 *
## num_voted_users             8.666e-06  6.509e-07  13.315  < 2e-16 ***
```

```
## cast_total_facebook_likes  1.179e-04  5.004e-05   2.356   0.0185 *
## facenumber_in_poster       3.979e-02  2.221e-02   1.792   0.0732 .
## actor_2_facebook_likes    -1.241e-04  5.237e-05  -2.370   0.0178 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3307.1  on 2432  degrees of freedom
## Residual deviance: 2957.9  on 2423  degrees of freedom
## AIC: 2977.9
##
## Number of Fisher Scoring iterations: 5
```




```
## [1] 0.7416058
```

As you can see the using backward stepwise regression produced slightly better AIC scores, however, the AUC decreased, but minimially. Another revelation, was that the Director Facebook score was not a signficant factor in our model, and was thus removed by the backward stepwise regression. It appears that for our purposes here, the actors facebook likes were better indicators of profitability that directors, which goes to show how the industry has unfolded. A few directors may have become prominent in our culture, but the recognizability of actors and actresses have a greater pull on whether or not a movie will make money.

As a final step, we used a confusion matrix to show the relative strength of our model.

```
#Creating confusion matrix
bin_prediction <- ifelse(p > 0.5, 1, 0)
confusion_bin <- confusionMatrix(data = bin_prediction, reference = test[,10])
confusion_bin$table
```

```
##           Reference
```

```
## Prediction   0   1
##          0 154  84
##          1 120 251
```

As you can see we tend to have more false negatives than false positives, and the break down of accuracy, specificity, precision and F1-score can be seen below:

Parameters	Model1
Accuracy	0.6650246
Classification Error Rate	0.3349754
Precision	0.6470588
Sensitivity	0.5620438
Specificity	0.7492537
F1 Score	0.6015625

Profit Margin Model

We now have a model to give us an insight if a particular movie is going to make money or loss money. However, from a movie investor's perspective, that information is not sufficient to make up his or her mind to go into that business. Therefore, another focus of our final project to build a model that can predict how much money each movie makes (gross margin) or otherwise.

Below we built multivariable linear regression model without any data transformation.

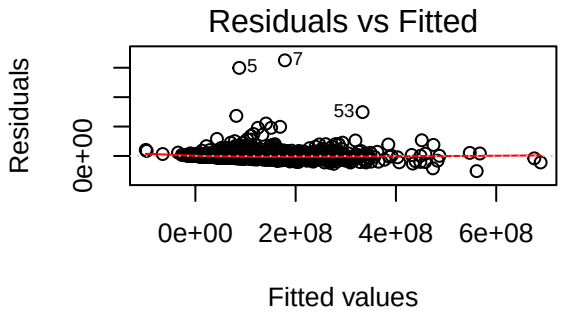
```
##               title_year                   duration
##                        0                          0
##   director_facebook_likes    actor_3_facebook_likes
##                        0                          0
##    actor_1_facebook_likes                      gross
##                        0                          0
##           num_voted_users cast_total_facebook_likes
##                        0                          0
##      facenumber_in_poster                     budget
##                        0                          0
##    actor_2_facebook_likes                imdb_score
##                        0                          0
##                       cpi                  adj_gross
##                        0                          0
##               adj_budget                 adj_margin
##                        0                          0

##               title_year                   duration
##                        0                          0
##   director_facebook_likes    actor_3_facebook_likes
##                        0                          0
##    actor_1_facebook_likes                      gross
##                        0                          0
##           num_voted_users cast_total_facebook_likes
##                        0                          0
```

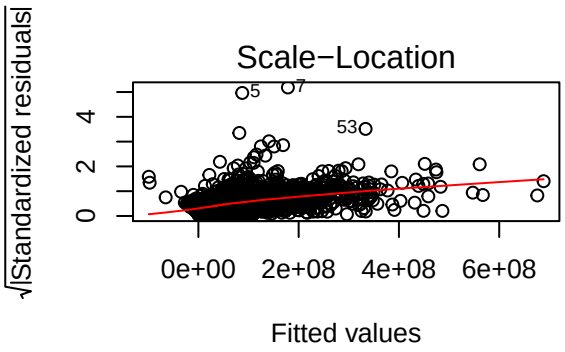
```
##      facenumber_in_poster                    budget
##                      1259                         0
##    actor_2_facebook_likes                 imdb_score
##                         0                         0
##                       cpi                  adj_gross
##                         0                         0
##                adj_budget                 adj_margin
##                         0                         0
```

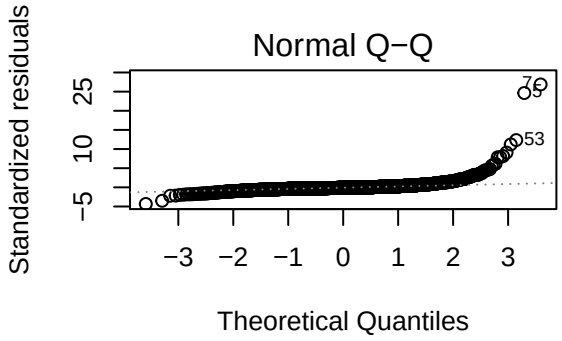
The first model we created does not show good performance. The AIC is 121904.1, BIC is 121970.4 and logLik is -60941.07. After we take a look at the histogram of the residuals plot, we realized it is highly skewed to the right. The skewness is as high as 13.65786

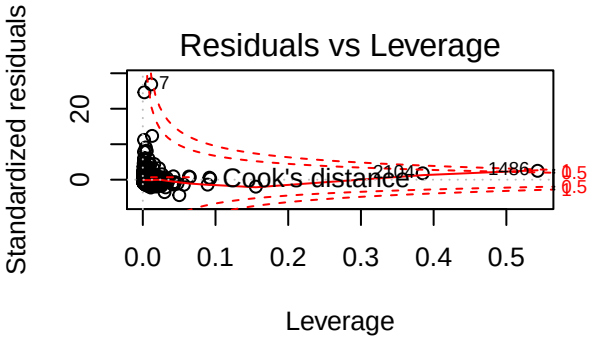
```
##
## Call:
## lm(formula = adj_gross ~ (duration + director_facebook_likes +
##     actor_3_facebook_likes + actor_1_facebook_likes + num_voted_users +
##     cast_total_facebook_likes + actor_2_facebook_likes + imdb_score +
##     adj_budget + adj_margin) - adj_margin, data = movies1)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -515166072  -36276181  -14813576   15366156 3251468994
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -8.729e+07  1.718e+07  -5.081 3.97e-07 ***
## duration                   2.419e+05  1.172e+05   2.064  0.03909 *
## director_facebook_likes   -9.268e+02  6.303e+02  -1.470  0.14157
## actor_3_facebook_likes    -1.027e+04  3.160e+03  -3.251  0.00116 **
## actor_1_facebook_likes    -7.729e+03  1.927e+03  -4.011 6.18e-05 ***
## num_voted_users            2.781e+02  1.885e+01  14.758  < 2e-16 ***
## cast_total_facebook_likes  7.405e+03  1.923e+03   3.851  0.00012 ***
## actor_2_facebook_likes    -8.165e+03  2.036e+03  -4.011 6.18e-05 ***
## imdb_score                 1.232e+07  2.590e+06   4.756 2.06e-06 ***
## adj_budget                 6.729e-01  5.133e-02  13.110  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 121700000 on 3032 degrees of freedom
## Multiple R-squared:  0.2674, Adjusted R-squared:  0.2652
## F-statistic:   123 on 9 and 3032 DF,  p-value: < 2.2e-16
```

```
##   r.squared adj.r.squared     sigma statistic       p.value df    logLik
## 1 0.2674188     0.2652442 121677563  122.9767 1.542982e-197 10 -60943.97
##        AIC      BIC     deviance df.residual
## 1 121909.9 121976.2 4.489006e+19        3032
```

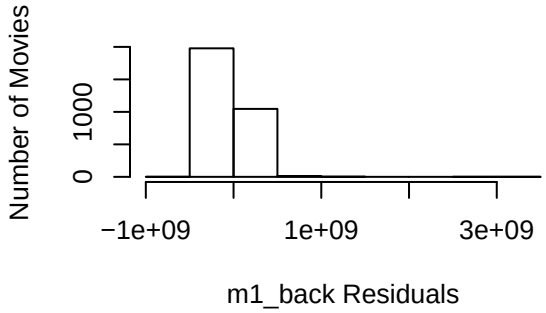






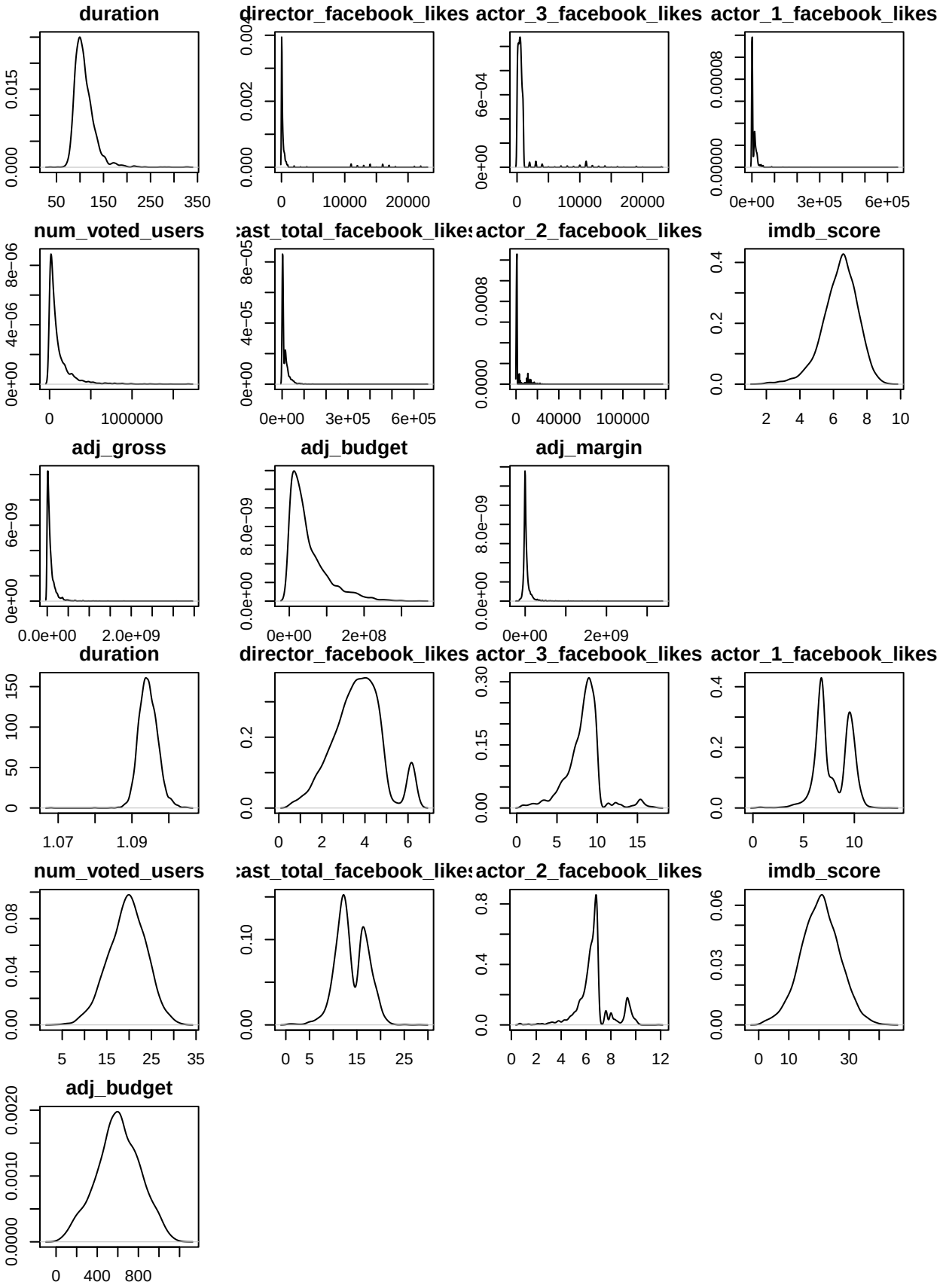



```
## [1] 13.63697
## attr(,"method")
## [1] "moment"
```



The bad performance is most likely due to the abnormal skewness and kurtosis from the original data. Therefore, we adopted Box-Cox transformation based on normality assumption.

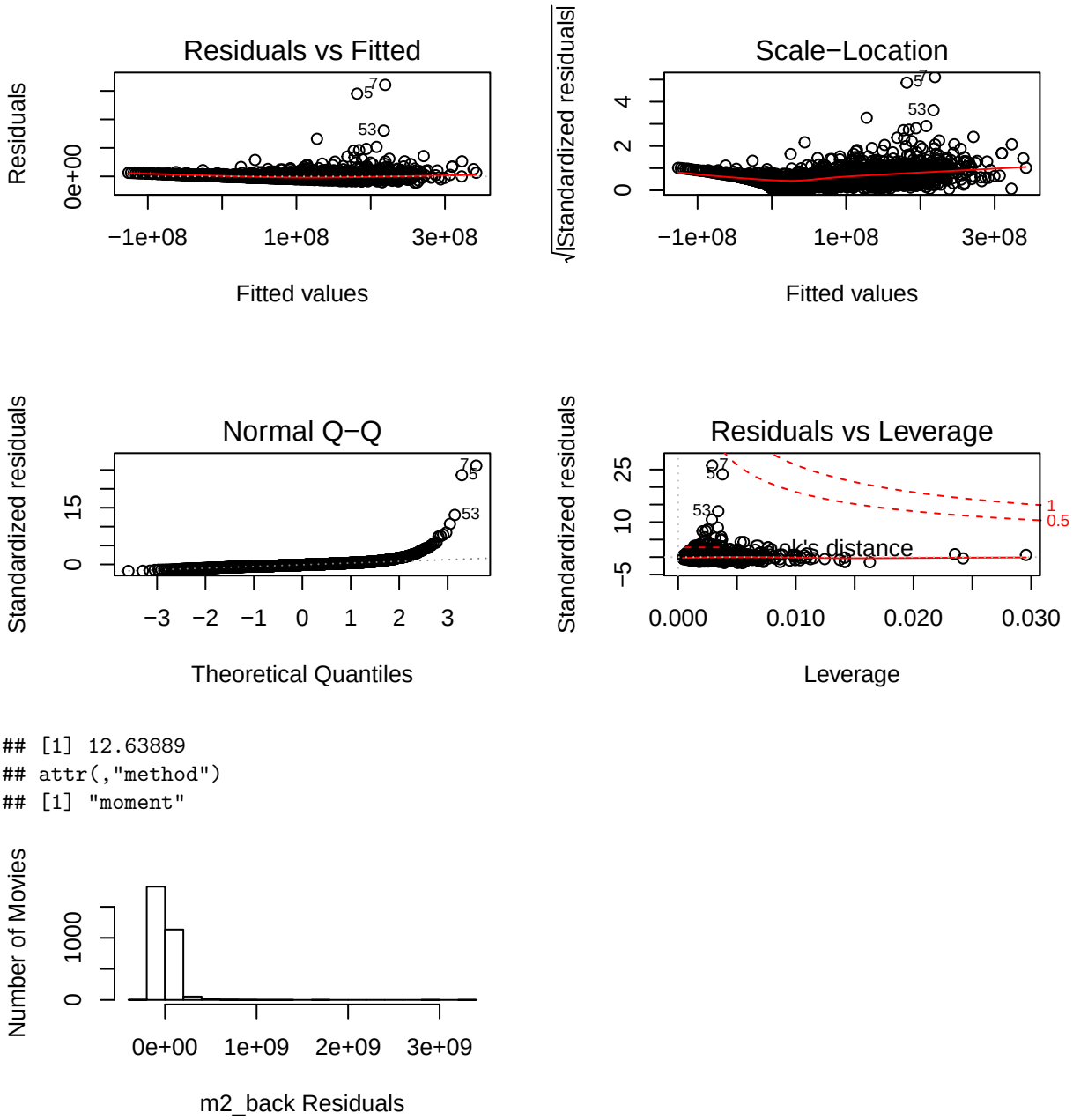
The following code is just comparing distributions of the two datasets. One is dataset before transformation, then the second one is the dataset after transformation. From the comparison, we clearly notice Box-Cox transformation approximately normalize the data.


duration
director_facebook_likes
actor_3_facebook_likes
actor_1_facebook_likes
num_voted_users
cast_total_facebook_likes
actor_2_facebook_likes
imdb_score
adj_gross
adj_budget
adj_margin
duration
director_facebook_likes
actor_3_facebook_likes
actor_1_facebook_likes
num_voted_users
cast_total_facebook_likes
actor_2_facebook_likes
imdb_score
adj_budget

Our second model has very similar AIC, BIC, and loglikelyhood values. However, the skewness of the residual histogram has been reduced. This has indicated the second model is a better model compare to the first one.

```
##
## Call:
## lm(formula = adj_gross ~ actor_1_facebook_likes + num_voted_users +
##     cast_total_facebook_likes + actor_2_facebook_likes + imdb_score +
##     adj_budget, data = movies2)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -213344311  -48993621  -14876796   25965264 3210512676
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -156863255   16641037  -9.426  < 2e-16 ***
## actor_1_facebook_likes     -37869155    6536766  -5.793 7.61e-09 ***
## num_voted_users              8857320     796003  11.127  < 2e-16 ***
## cast_total_facebook_likes   15740940    3983262   3.952 7.93e-05 ***
## actor_2_facebook_likes      -5018195    3302762  -1.519    0.129
## imdb_score                   3756367     429370   8.749  < 2e-16 ***
## adj_budget                    170184      13375  12.724  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.23e+08 on 3035 degrees of freedom
## Multiple R-squared:  0.2508, Adjusted R-squared:  0.2493
## F-statistic: 169.3 on 6 and 3035 DF,  p-value: < 2.2e-16
```

```
##   r.squared adj.r.squared     sigma statistic       p.value df    logLik
## 1  0.250803     0.2493219 122988888   169.334 3.647107e-186  7 -60978.08
##        AIC      BIC     deviance df.residual
## 1 121972.2 122020.3 4.590822e+19        3035
```




```
## [1] 12.63889
## attr(,"method")
## [1] "moment"
```


The following code is just the data preparation step for evaluation before we apply our model. Finally, we create a master dataframe containing our predicted results and actual data.

Final output

```
head(movies_profit)
```

```
##                       Movie Title Actual Adjusted Gross Predicted Gross
## 1             The Broadway Melody               39181395         4284412
## 2                     42nd Street               42790698        38323928
## 3                         Top Hat               52554745        50010725
## 4                     Modern Times                2818619       174324549
## 5 Snow White and the Seven Dwarfs             3082091417       181651092
## 6                 The Wizard of Oz              383354452       210475502
```

```
##   Actual Profit Margin Predicted Profit Margin
## 1            0.8650285              -0.2343287
## 2            0.8091304               0.7868840
## 3            0.7970000               0.7866735
## 4           -8.1886428               0.8514307
## 5            0.9891848               0.8164980
## 6            0.8738887               0.7703043
```

The predicted profit margin variable will serve as a reference for investors to decide if they want to contriubte to the production of the movies and share the profit that is generated.

Since we have the actual profit margin variable available to us, it will also be really interesting to investigate if the quality of the movies will have any imapcts on the profitability of the movies. We will like to use IMDB rating as a standard representing movie quality.

This line of code just proves that there exists very weak positive relationship between quality and profitability of movies. The p-value is 0.007905, which is less than the significance level of 0.05. In addition, the 95% confidence interval is (0.01263244, 0.08354498), which does not cross zero. It also shows the result is statistically significant. However, the correlation coefficient is only 0.04814938, which is a very weak association between the two variable.

Therefore, if investors care more about profitability, it is recommended not to care too much about the quality of the movie. Spending huge amount of money on improving the movie quality will lead the expected return that is minuscule.

```
##
##  Pearson's product-moment correlation
##
## data:  movies$imdb_score and movies$profit_margin
## t = 2.6579, df = 3040, p-value = 0.007905
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.01263244 0.08354498
## sample estimates:
##        cor
## 0.04814938
```

Smooth Operators - All Done!