Statistical Learning in Movies

Max Sellers, Claire Martino, Ari Augustine, and Rim Nassiri

Abstract

To be a producer of film, understanding the factors influencing movie success is extremely important. This project uses statistical learning techniques to predict how successful a film will be critically and popularity-wise. Using a dataset compiled from the Internet Movie Database (IMDB), we delve into the relationships between these variables and movie ratings. Our goal in this project is to use variables like the movie's budget, revenue, runtime, rating, and popularity we can determine what the film's critical and commercial success could be by using models like simple and multiple linear regression.

I. Introduction

II. Methods

a. Cleaning Data

```
movies_data = movies_data %>%
  group_by(title) %>%
  filter(budget >= 1000000 & revenue >= 1000000 & vote_count > 100) %>%
  arrange(desc(vote_average))
```

b. Choosing Important Values on Dataset

```
#number of categories and variables in the dataset
categories = ncol(movies_data)
var = nrow(movies_data)

#means of variables that will be using in analysis
mean_budget = format(round(mean(movies_data$budget),2),scientific=F)
mean_popularity = format(round(mean(movies_data$popularity),2), scientific = F)
mean_revenue = format(round(mean(movies_data$revenue),2), scientific = F)
mean_vote_average = format(round(mean(movies_data$vote_average),2), scientific = F)
mean_runtime = format(round(mean(movies_data$runtime),2), scientific = F)
```

There are 12 categories in this movies dataset that we will be using. There are 3566 rows in this dataset that we will be using.

The mean budget of the dataset is 42149142. The mean popularity of the movies is 12.34. The mean revenue of the movies is 130185255. The mean voter average of the movies 6.39. The mean runtime of the movies is 110.84.

c. Training/Testing Data [USE SOMEWHERE]

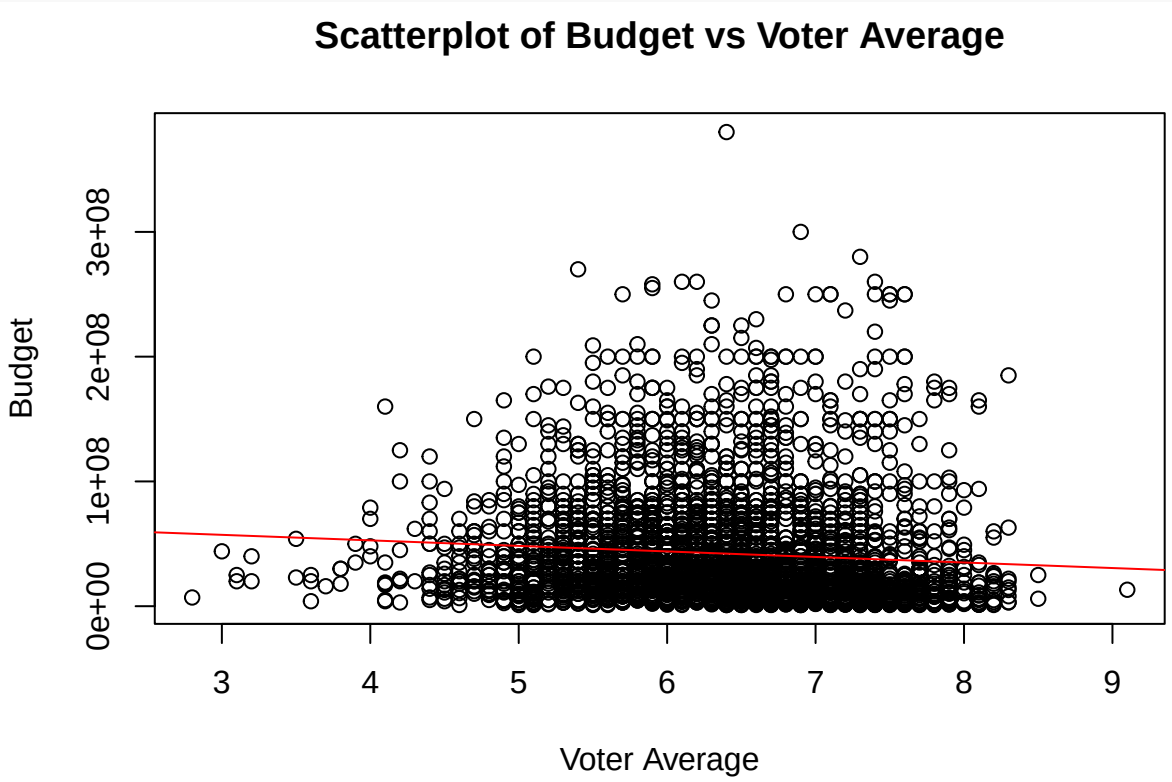
```
num_obs = nrow(movies_data)  #extracting the total rows from the data set

movies_indx = sample(num_obs, size = trunc(0.5*num_obs))
```

```r
#training data set:
movies_train = movies_data[movies_indx,]

#test data set:
movies_test = movies_data[-movies_indx,]
```

III. Results

a. Linear Regression Summary and Line

```r
#linear regression model of budget vs vote_average
linear_1 = lm(budget ~ vote_average, data = movies_data)
summary_linear1 = summary(linear_1)

r_2_lin_1 = summary_linear1$r.squared


#graph of above data
plot(budget~ vote_average, data= movies_data, xlab = "Voter Average", ylab = "Budget", main = "Scatterpl
abline(linear_1, col = "red")
```



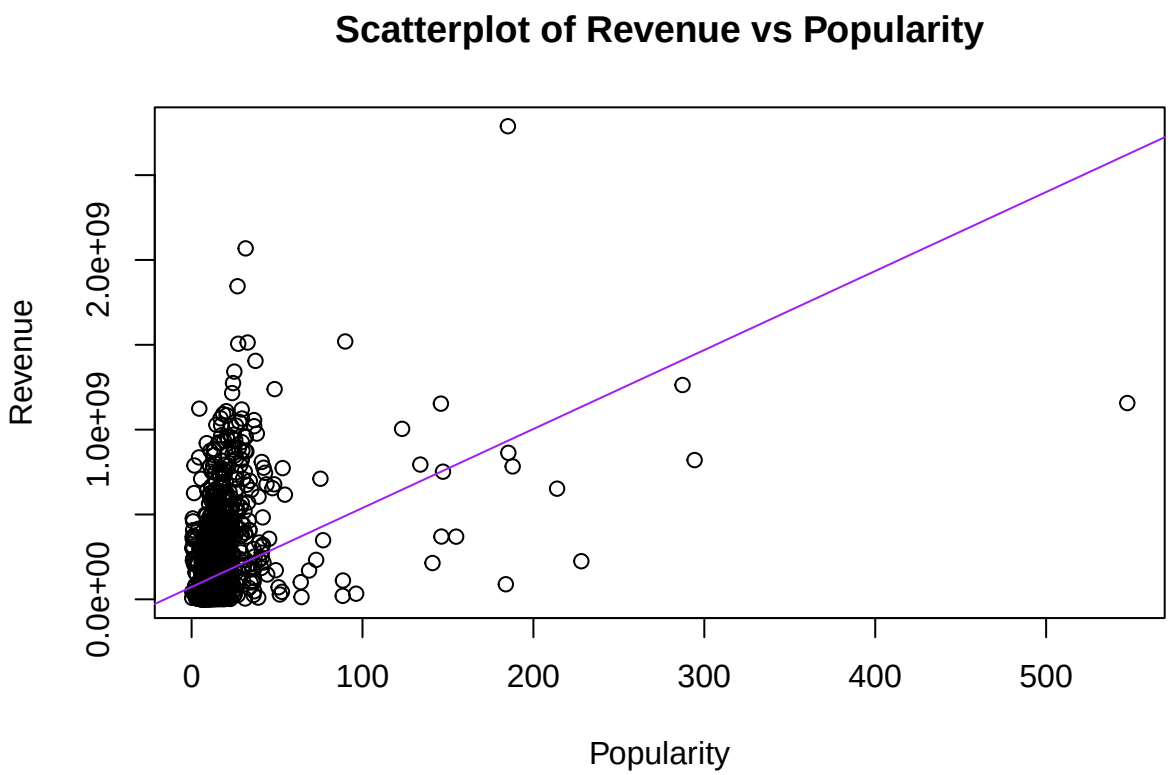
Since the Pr(>|t|) value of voter average is 8.77e_07, and this value is less than the standard level of significance of 0.05, this shows that there is a statistically significant relationship between voter average and budget.

In order to assess the relationship between the predictor and the response variable, you must look at the R^2 value. In this case, $R^2 = 0.006763$. Since this value is closer to 0 than it is 1, this indicates a weak relationship between voter average and budget.

This scatterplot with the linear regression line shows a weak and negative relationship between between

budget and voter average.

```
#linear regression model of revenue vs popularity
linear_2 = lm(revenue ~ popularity, data = movies_data)
summary_linear2 = summary(linear_2)

r_2_lin_2 = summary_linear2$r.squared


plot(revenue ~ popularity, data= movies_data, xlab = "Popularity", ylab = "Revenue", main = "Scatterplo
abline(linear_2, col = "purple")
```

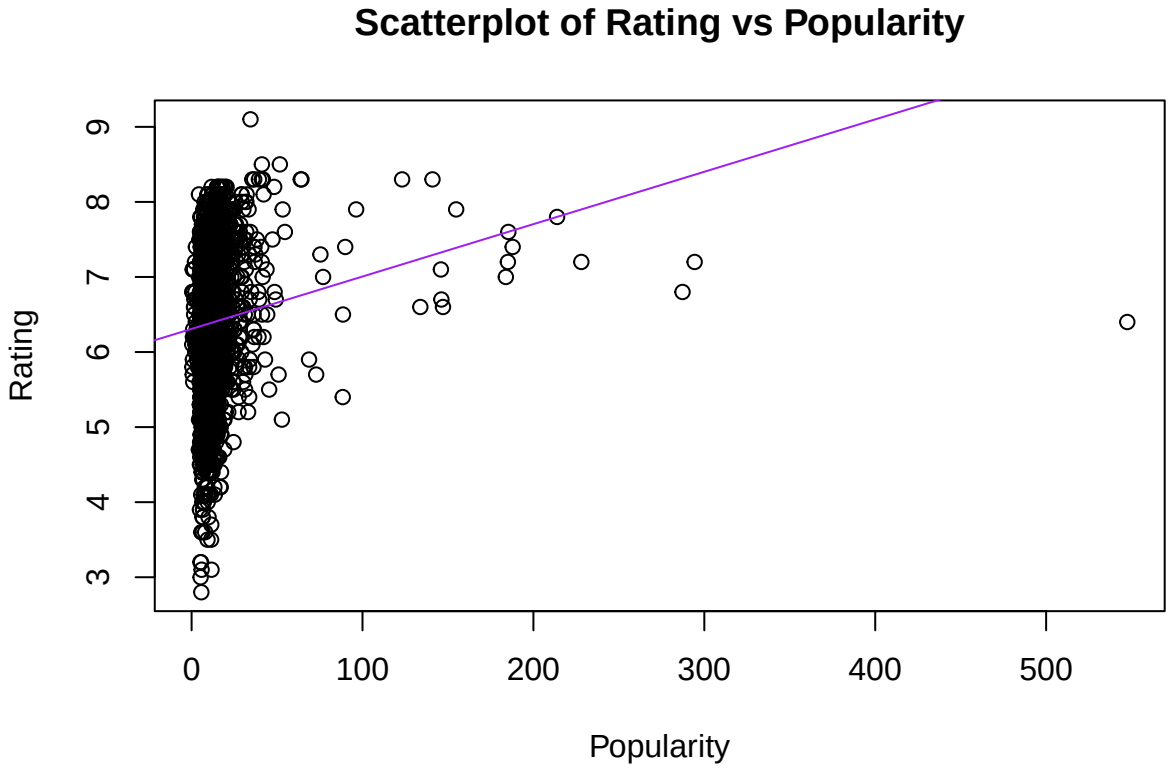


Since the Pr(>|t|) value of popularity is <2e-16, and this value is less than the standard level of significance of 0.05, this shows that there is a statistically significant relationship between popularity and revenue.

In order to assess the relationship between the predictor and the response variable, you must look at the R^2 value. In this case, $R^2 = 0.1547861$. Since this value is closer to 0 than it is 1, this indicates a mildly weak relationship between revenue and popularity.

The scatter plot with the linear regression line shows this mildy weak positive relationship between revenue and popularity.

```
linear_3 = lm(vote_average ~ popularity, data = movies_data)
summary_linear3 = summary(linear_3)

r_2_lin_3 = summary_linear3$r.squared

plot(vote_average ~ popularity, data= movies_data, xlab = "Popularity", ylab = "Rating", main = "Scatte
abline(linear_3, col = "purple")
```



Since the Pr(>|t|) value of popularity is 8.77e-07, and this value is less than the standard level of significance of 0.05, this shows that there is a statistically significant relationship between popularity and rating.

In order to assess the relationship between the predictor and the response variable, you must look at the R^2 value. In this case, $R^2 = 0.0185173$. Since this value is closer to 0 than it is 1, this indicates a weak relationship between rating and popularity.

The scatter plot with the linear regression line shows this weak positive relationship between rating and popularity.

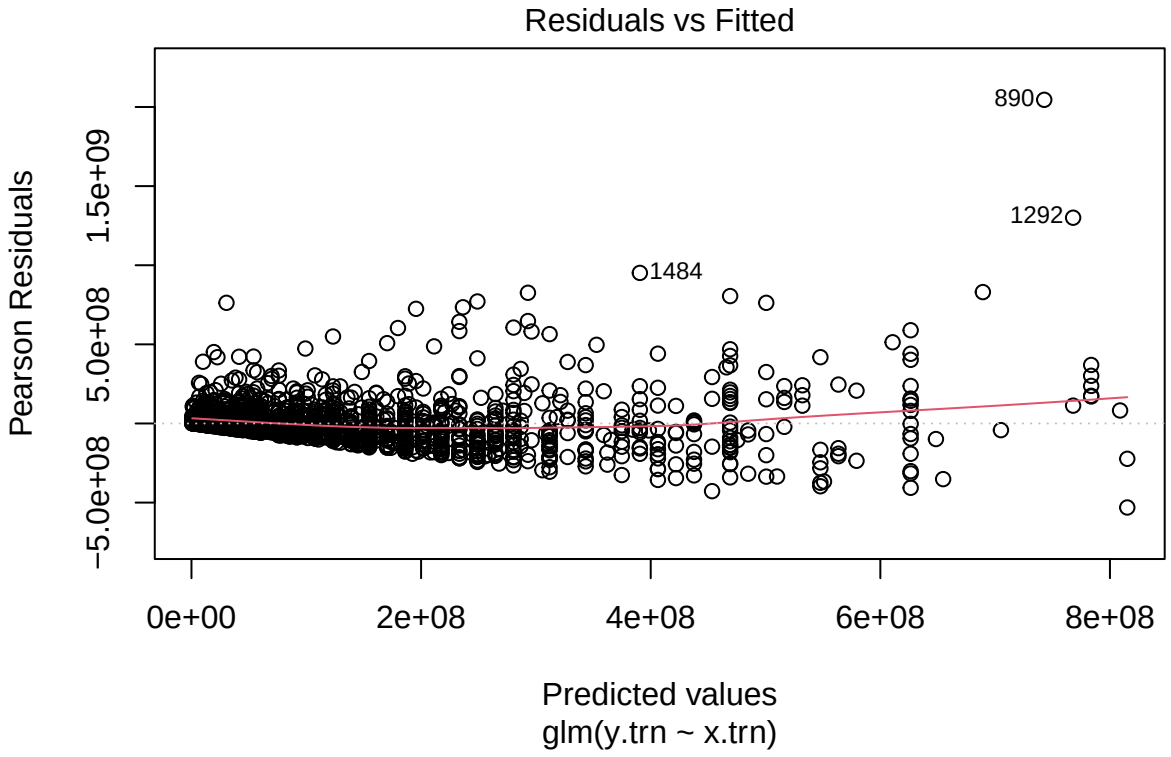
b. Trained Simple Linear Regression Summary

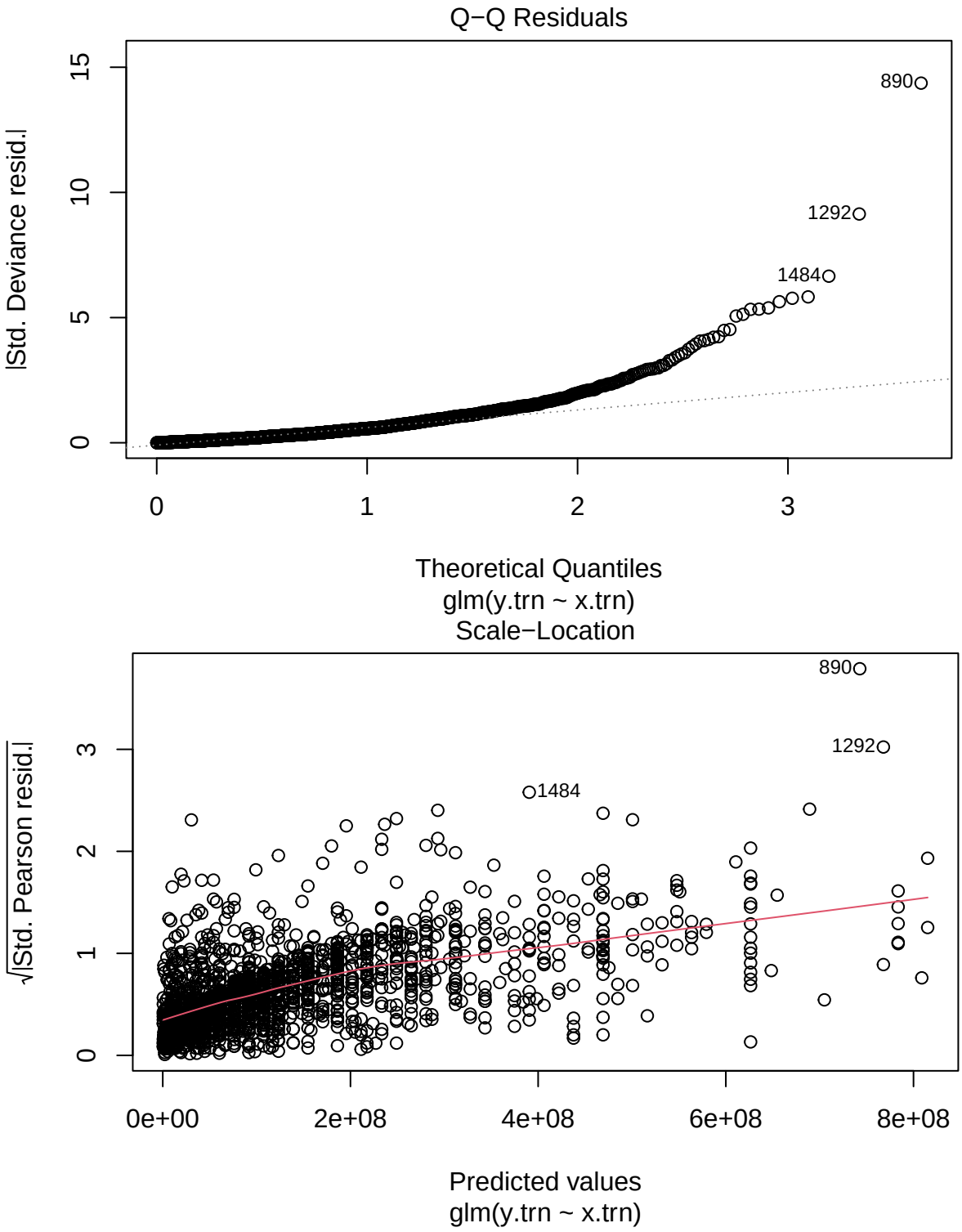
```
y.trn = movies_train$revenue
x.trn = movies_train$budget

train.data_frame = data.frame(y.trn, x.trn)

model_train = glm(formula = y.trn ~ x.trn, data = train.data_frame)
summary(model_train)
```

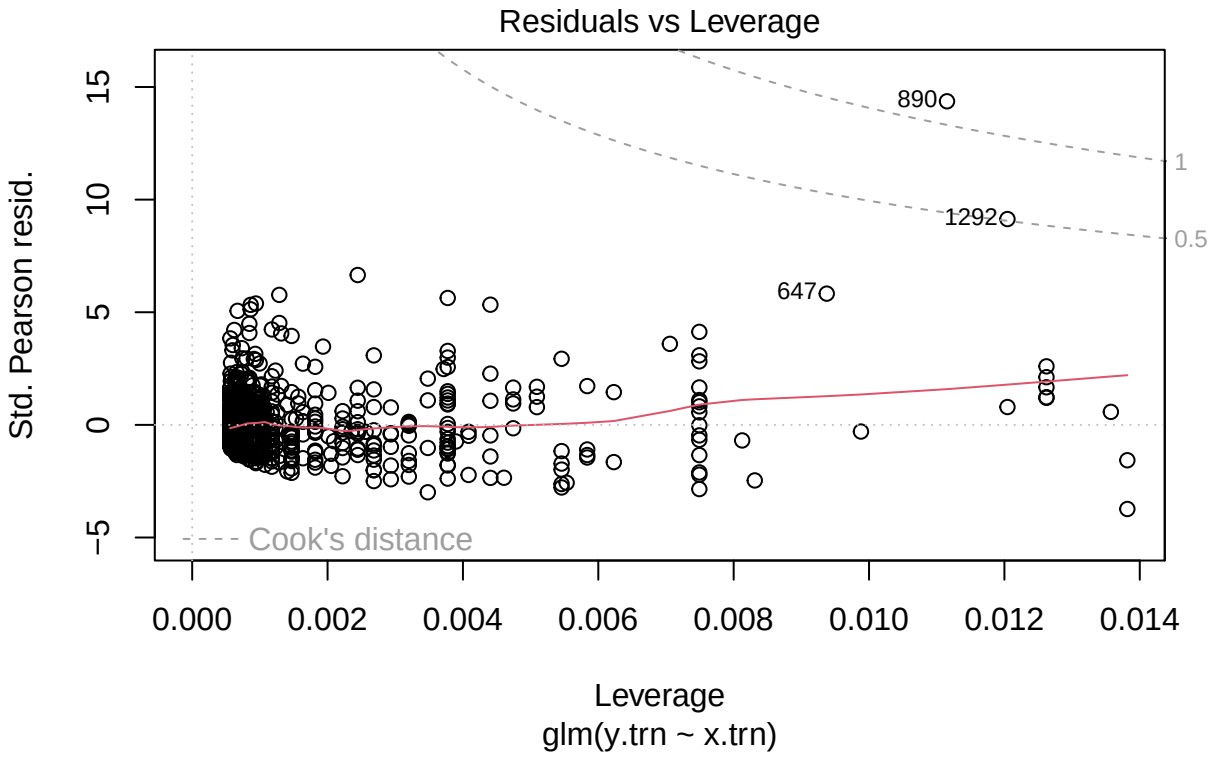
```
##
## Call:
## glm(formula = y.trn ~ x.trn, data = train.data_frame)
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.527e+06  4.725e+06  -0.535    0.593
## x.trn        3.145e+00  7.606e-02  41.346   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for gaussian family taken to be 2.049805e+16)
##
##     Null deviance: 7.1548e+19  on 1782  degrees of freedom
## Residual deviance: 3.6507e+19  on 1781  degrees of freedom
## AIC: 72032
##
## Number of Fisher Scoring iterations: 2
```

```
plot(model_train)
```




Q−Q Residuals
|Std. Deviance resid.|
890
1292
1484
0
5
10
15
0
1
2
3
Theoretical Quantiles
glm(y.trn ~ x.trn)
Scale−Location
√|Std. Pearson resid.|
890
1292
1484
0
1
2
3
0e+00
2e+08
4e+08
6e+08
8e+08
Predicted values
glm(y.trn ~ x.trn)




```
actual = ifelse(movies_test$revenue> 100000000, "Yes", "No")
head(actual)
```

```
## [1] "No"  "Yes" "Yes" "Yes" "Yes" "Yes"
```

```
predicted = ifelse(predict(model_train) > 100000000, "Yes", "No")
head(predicted)
```

```
##     1     2     3     4     5     6
##  "No" "Yes"  "No"  "No"  "No" "Yes"
```

```
error_function = function(actual, predicted){
  mean(actual != predicted)
}

error_function(actual = actual, predicted = predicted)
```

```
## [1] 0.4806506
```

```
1 - error_function(actual = actual, predicted = predicted)
```

```
## [1] 0.5193494
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
train_table = table(predicted = predicted, Actual = actual)

train_confusion_matrix = confusionMatrix(train_table, positive = "Yes")

train_confusion_matrix
```

```
## Confusion Matrix and Statistics
```

```
##
##          Actual
## predicted  No Yes
##       No  629 368
##       Yes 489 297
##
##                Accuracy : 0.5193
##                  95% CI : (0.4959, 0.5428)
##     No Information Rate : 0.627
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0089
##
##  Mcnemar's Test P-Value : 4.147e-05
##
##             Sensitivity : 0.4466
##             Specificity : 0.5626
##          Pos Pred Value : 0.3779
##          Neg Pred Value : 0.6309
##              Prevalence : 0.3730
##          Detection Rate : 0.1666
##    Detection Prevalence : 0.4408
##       Balanced Accuracy : 0.5046
##
##        'Positive' Class : Yes
##
```

IV. Discussion