Regression

Nikilas John

September 25th, 2022

Overview of Linear Regression

Linear Regression is a method of prediction that has been around since the 1800s. It is a very simple and powerful tool to learn about the relationship between two variables, a predictor and a target. There are many benefits as well as downsides to using this algorithm. Some of the strengths include: 1. Very simple to learn and use 2. Best results come from data that follows a linear pattern 3. Low variance

Some of the downsides to this algorithm include: 1. High bias due to the algorithm making the assumption that there is a linear relationship between the predictor and the target

Load the data

Import the USvideos.csv data set.This data set has 15424 observations of 16 variables This data set comes from "https://www.kaggle.com/datasets/datasnaek/youtube-new"

```
df <- read.csv("USvideos.csv")
```

Train/Test Split

Split the data into Train and Test

```
i <- sample(1:nrow(df), nrow(df)*0.8, replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

Data Exploration

Prints a summary of the data frame's contents

video_id: the id of the video trending_date: the date that the video started trending title: the title of the video channel_title: the name of the channel category_id: the id number associated with the category the video belongs to publish_time: the time the video was published tags: the tags that the video had views: the number of views the video got likes: the number of likes the video got dislikes: the number of dislikes the video got comment count: the number of comments the video got thumbnail_link: the link to the thumbnail comments_disabled: if the comments were disabled ratings_disabled: if the ratings were disabled video_error_or_removed: if the video had an error or was removed description: the description of the video

```
summary(train)
```

```
##    video_id         trending_date         title           channel_title
##  Length:12339       Length:12339       Length:12339       Length:12339
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##   category_id    publish_time           tags               views
##  Min.   : 1.00   Length:12339       Length:12339       Min.   :      559
##  1st Qu.:17.00   Class :character   Class :character   1st Qu.:    93932
##  Median :24.00   Mode  :character   Mode  :character   Median :   321996
##  Mean   :19.98                                        Mean   :  1248622
##  3rd Qu.:25.00                                        3rd Qu.:  1008586
##  Max.   :43.00                                        Max.   :149376127
##      likes            dislikes       comment_count     thumbnail_link
##  Min.   :      0   Min.   :      0   Min.   :      0   Length:12339
##  1st Qu.:   1970   1st Qu.:     87   1st Qu.:    286   Class :character
##  Median :   9059   Median :    328   Median :   1029   Mode  :character
##  Mean   :  46213   Mean   :   3352   Mean   :   6083
##  3rd Qu.:  29882   3rd Qu.:   1142   3rd Qu.:   3445
##  Max.   :3093544   Max.   :1674420   Max.   :1361580
##  comments_disabled ratings_disabled video_error_or_removed description
##  Mode :logical     Mode :logical    Mode :logical          Length:12339
##  FALSE:12130       FALSE:12271      FALSE:12339            Class
:character
##  TRUE :209         TRUE :68                                Mode
:character
##
##
##

str(train)

## 'data.frame':    12339 obs. of  16 variables:
##  $ video_id              : chr  "s1cpCfX-mgo" "naOKPDCngMI" "qxbS8qbBL0c"
"HUZRjGEwqCg" ...
##  $ trending_date         : chr  "18.17.01" "18.04.01" "17.14.12"
"18.01.02" ...
##  $ title                 : chr  "Incubus - Loneliest"
"巨大なうさぎを癒すねこ。-Maru heals the huge rabbit.-" "Ed Sheeran - Perfect
in the Live Lounge" "State of the Union 2018 live stream: President Donald
Trump delivers first SOTU Address | ABC News" ...
##  $ channel_title         : chr  "IncubusVEVO" "mugumogu" "BBC Radio 1"
"ABC News" ...
##  $ category_id           : int  10 15 10 25 25 23 27 1 23 28 ...
##  $ publish_time          : chr  "2018-01-12T05:00:01.000Z" "2017-12-
26T23:06:30.000Z" "2017-12-12T16:24:25.000Z" "2018-01-31T05:05:25.000Z" ...
##  $ tags                  : chr
"Incubus|\"Loneliest\"|\"Island\"|\"Records\"|\"Alternative\""
"Maru|\"cat\"|\"kitty\"|\"pets\"|\"まる\"|\"猫\"|\"ねこ\"" "Ed
```

```
Sheeran|\"Perfect\"|\"BBC\"|\"Radio 1\"|\"Live Lounge\"" "donald
trump|\"donald trump state of the union\"|\"state of the union 2018\"|\"trump
state of the union\"|\"pre"| __truncated__ ...
##  $ views                 : int  215570 46054 283885 199486 42051 1286392
496031 37594 623894 49633 ...
##  $ likes                 : int  7189 2051 9320 2176 421 58513 18985 2058
32822 2120 ...
##  $ dislikes              : int  293 20 166 1248 93 1174 228 41 561 53 ...
##  $ comment_count         : int  558 213 349 715 97 5366 6004 70 1478 161
...
##  $ thumbnail_link        : chr  "https://i.ytimg.com/vi/s1cpCfX-
mgo/default.jpg" "https://i.ytimg.com/vi/naOKPDCngMI/default.jpg"
"https://i.ytimg.com/vi/qxbS8qbBL0c/default.jpg"
"https://i.ytimg.com/vi/HUZRjGEwqCg/default.jpg" ...
##  $ comments_disabled     : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ ratings_disabled      : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ video_error_or_removed: logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ description           : chr  "Directed by: Julian Schratter\\nProduced
by: Through Whose Productions\\nEdited by: Ryan Adams at Subtractive\\"|
__truncated__ "伝説のマッサージ師まる。次のお客さんは巨大なうさぎさん。Legendary
masseur Maru massages the huge rabbit.\\nBlog"| __truncated__ "Ed Sheeran
performs Perfect in the BBC Radio 1 Live Lounge" "The president discusses
healthcare, immigration and the economy in his address to a joint session of
congress.\"| __truncated__ ...
```

Dimensions Function

Lets use the dim() function to check the dimensions of the rows and columns of the train dataset

```
dim(train)

## [1] 12339    16
```

Head() Function

Lets use the head() function to see the first twenty values of the views column

```
head(train$views, n=20)

##  [1]  215570   46054  283885  199486   42051 1286392  496031   37594
623894
## [10]   49633  865918  629166 1097679 1107609   93960  125684 1058470
878841
## [19]  854242   32347
```

Check for NA's

Check for NA's in the data to determine if we need to clean them up

```
sapply(train, function(x) sum(is.na(x)))
```

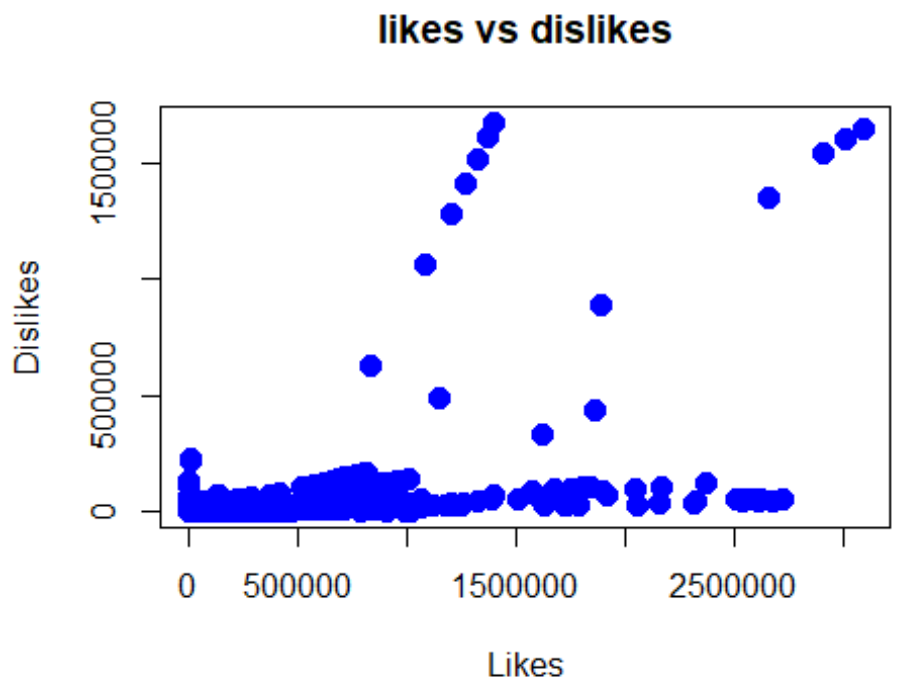
```
##                video_id          trending_date                  title
##                       0                      0                      0
##           channel_title            category_id           publish_time
##                       0                      0                      0
##                    tags                  views                  likes
##                       0                      0                      0
##                dislikes          comment_count         thumbnail_link
##                       0                      0                      0
##       comments_disabled       ratings_disabled video_error_or_removed
##                       0                      0                      0
##             description
##                       0
```

Likes/Dislikes Graph

Plots the Likes vs Dislikes to see trends

Takeaways: 1. Most of the videos that have a like of likes also do not have a lot of dislikes 2. There are very few videos that feature on the trending page that have a lot of dislikes (only 15 fall outside of the grouping made by the majority of the videos)

```
plot(train$likes, train$dislikes, pch=16, col="blue", cex=1.5,
        main="likes vs dislikes", xlab="Likes", ylab="Dislikes")
```

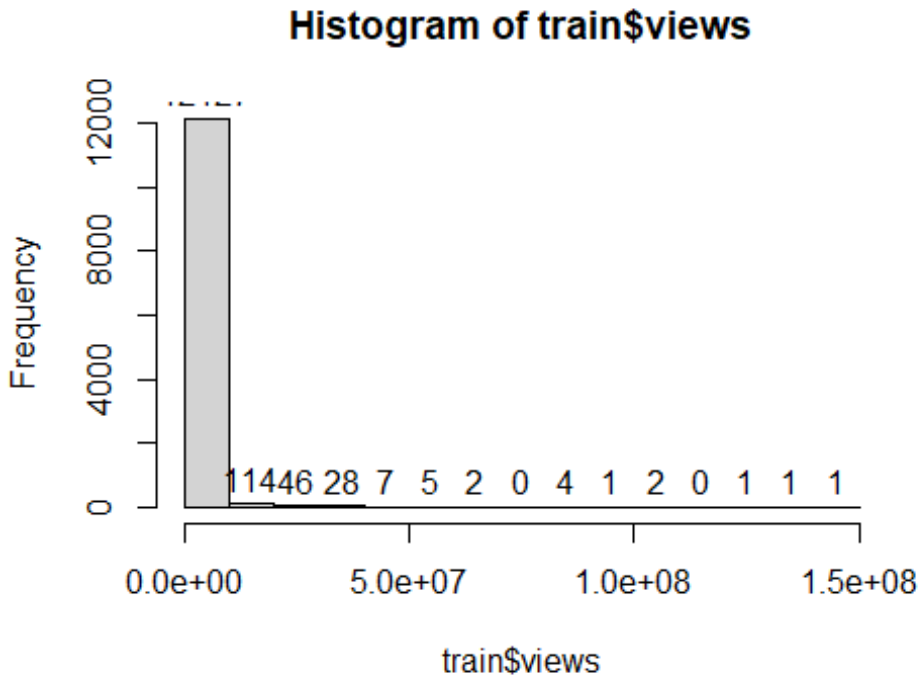


Views Histogram

Here we will create a histogram of view counts to see if YouTube favors videos with a lot of views to put on their trending page

We can see that the extreme majority of the trending videos actually fall under 10 ,000,000 views, so lets limit the X-axis and see a distribution of the ~12,000 videos grouped together

```
hist(train$views, labels = TRUE)
```



Views Histogram (<10,000,000 Views)

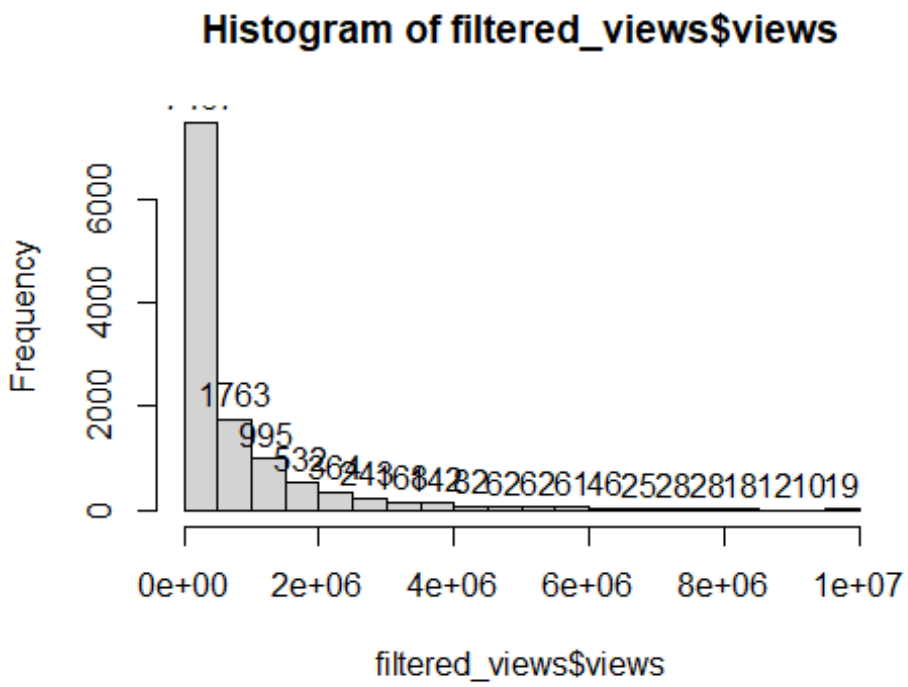
Here we will execute what we discussed above

Still, we see most of the trending videos are under 1 million views

Some takeaways: 1. A video not having above 1 million views is not a lot for a "trending" YouTube video - Is YouTube hand-picking their own trending page to suit content that they like rather than what is really going viral?

Twitter has an excellently designed trending page that will sense keywords in the tweets posted to their platform in the last hour and simply display them back to the user, regardless of content.

```
filtered_views <- subset(train, views < 10000000)
hist(filtered_views$views, labels = TRUE)
```



Names of the columns

I forgot the names of the columns, lets use the names() function to see what they were

```
names(train)

##  [1] "video_id"                "trending_date"           "title"
##  [4] "channel_title"           "category_id"             "publish_time"
##  [7] "tags"                    "views"                   "likes"
## [10] "dislikes"                "comment_count"           "thumbnail_link"
## [13] "comments_disabled"       "ratings_disabled"
"video_error_or_removed"
## [16] "description"
```

Simple Linear Regression Model

Here, we will see the impact views has on likes

After running the code, we see that views is an excellent predictor for the number of likes on a video by seeing that R gave views a three star rating. The R-squared also indicates that the plotted data will fit a trend line really well because the value is closer to one. The p-value is also excellent, being one of the lowest I've seen.

```
lm1 <- lm(likes~views, data=train)
summary(lm1)

##
## Call:
```

```
## lm(formula = likes ~ views, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1514199   -11346    -7815    -2032  1683280
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.762e+03  7.140e+02   10.87   <2e-16 ***
## views       3.079e-02  1.531e-04  201.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 76410 on 12337 degrees of freedom
## Multiple R-squared:  0.7662, Adjusted R-squared:  0.7662
## F-statistic: 4.044e+04 on 1 and 12337 DF,  p-value: < 2.2e-16
```

Plotting the Residuals

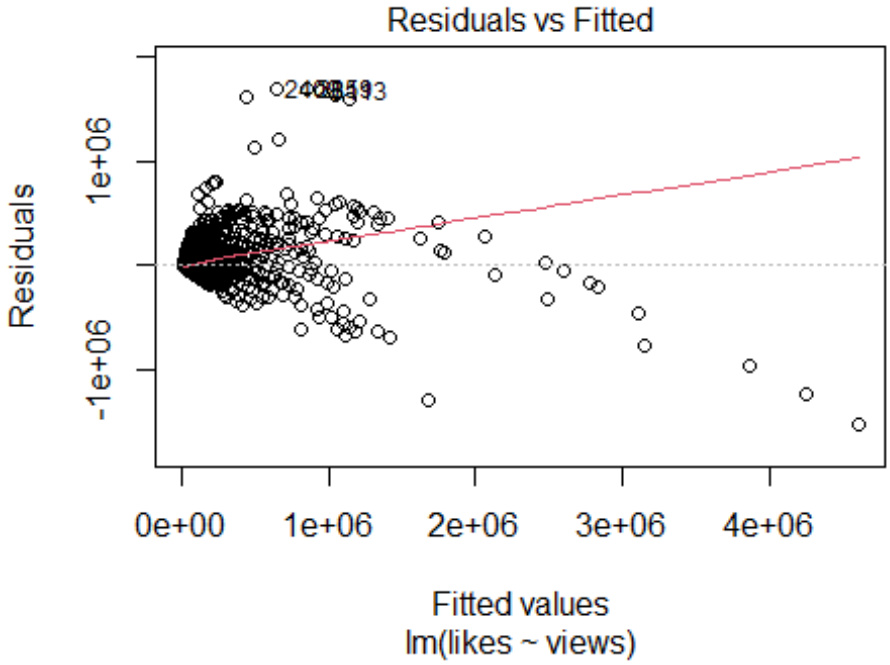
Using the Residuals vs Fitted Plot, you can see that all of the residuals displayed in the graph have an equal spread between the lines with no distinctive pattern. This is how I am able to conclude that the residuals have a linear pattern.

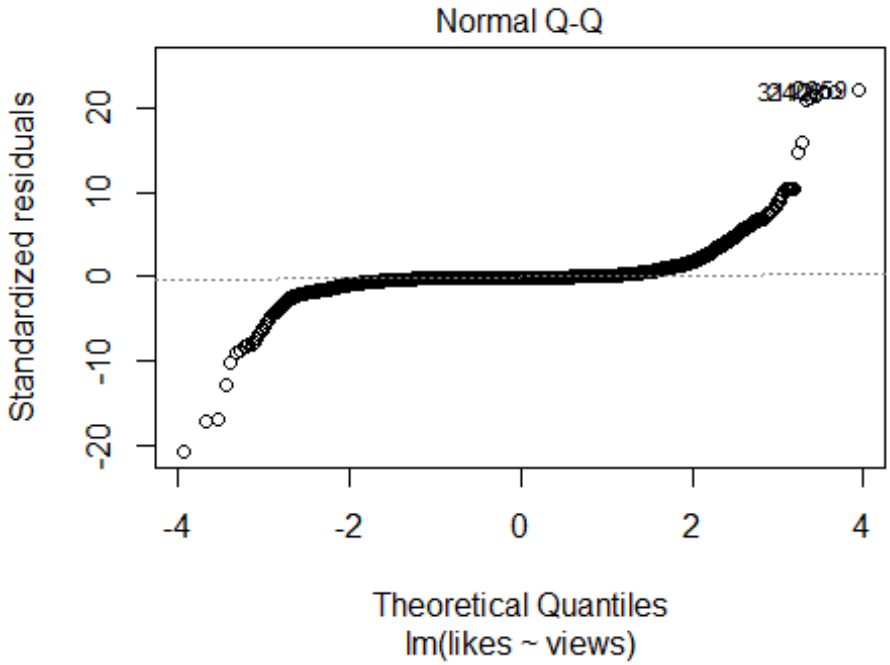
The Normal Q-Q plot shows me that the majority of the residuals fall on one line which is a very good thing. This means that the residuals are normally distributed and there are very few outlier cases.

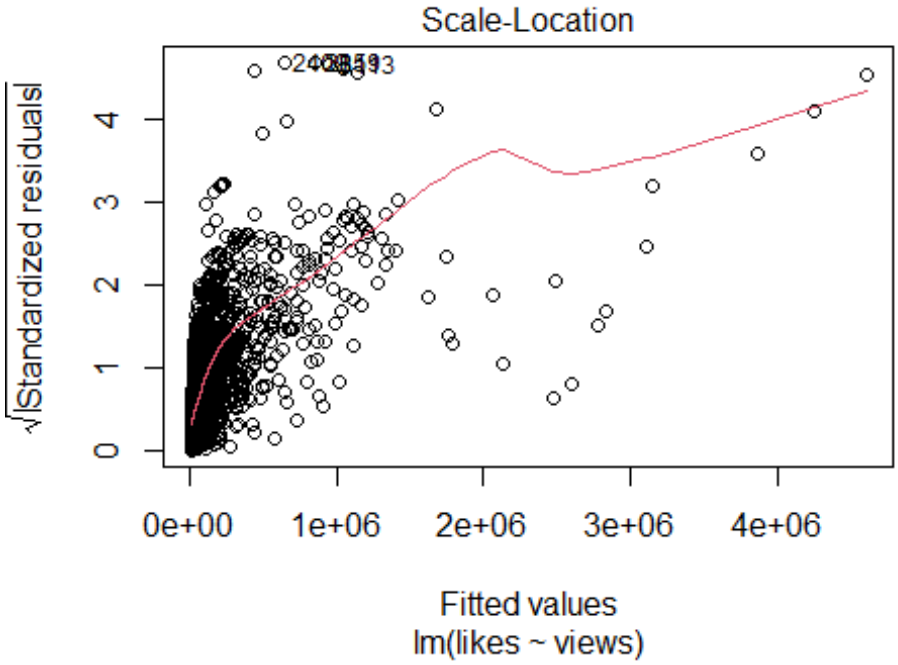
The Scale-Location plot shows that the majority of the residuals are not spread equally among the range of the predictors. It seems to be leaning heavily to the left. This plot indicates that the linear regression using views is not diversified and spread equally on the best fit line.

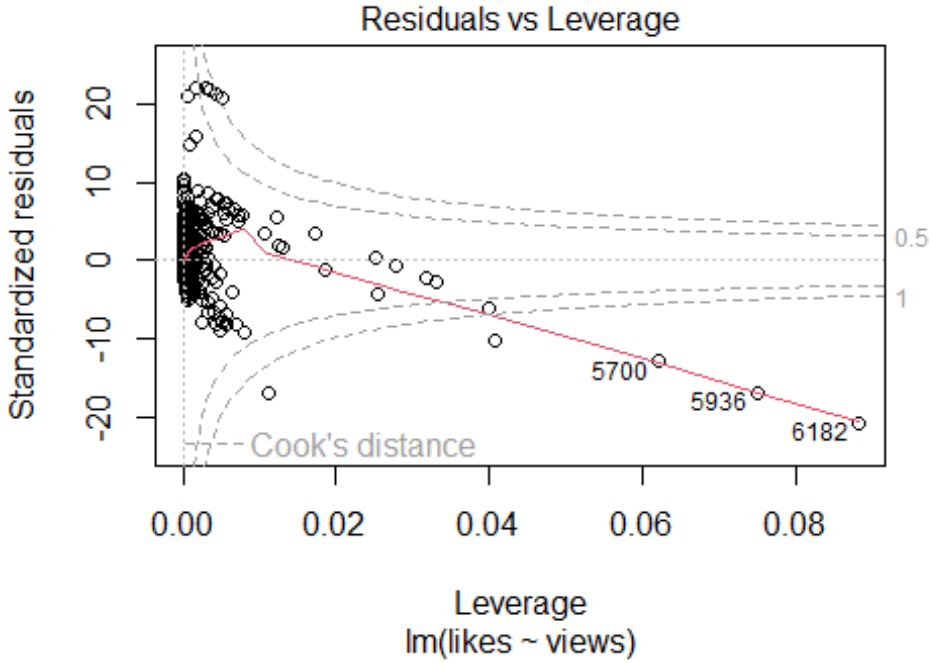
The Residuals vs Leverage plot was, in my opinion, was the plot I could gather the most information from. Using this plot, I was able to discover multiple residuals that fall outside of the Cook's distance. This means that these few residuals, if removed, would affect the regression results, and since there are multiple of them, they would most likely affect them drastically.

```
plot(lm1)
```


Residuals vs Fitted
Residuals
1e+06
-1e+06
0e+00
1e+06
2e+06
3e+06
4e+06
Fitted values
lm(likes ~ views)


Normal Q-Q
Standardized residuals
20
10
0
-10
-20
-4
-2
0
2
4
Theoretical Quantiles
lm(likes ~ views)


Scale-Location
√|Standardized residuals|
0
1
2
3
4
0e+00
1e+06
2e+06
3e+06
4e+06
Fitted values
lm(likes ~ views)


Residuals vs Leverage
Standardized residuals
20
10
0
-10
-20
0.5
1
5700
5936
6182
Cook's distance
0.00
0.02
0.04
0.06
0.08
Leverage
lm(likes ~ views)

Multiple Linear Regression Model

This is where we will use multiple linear regression to see if we can get an even better model. The predictor that we will be adding to the mix is dislikes. We will see if views and dislikes have an impact on likes.

The dislikes predictor received a three star rating from R meaning that we made a good decision choosing it as a predictor. Most of the conclusions that were made in the last linear regression model are the same with this one other than a few exceptions. The first exception is the R-squared which is .0002 higher than the previous one.

```
lm2 <- lm(likes~views+dislikes, data=train)
summary(lm2)

##
## Call:
## lm(formula = likes ~ views + dislikes, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1441824   -11219    -7512    -1916  1673800
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.452e+03  7.164e+02  10.402  < 2e-16 ***
## views        3.128e-02  1.842e-04 169.843  < 2e-16 ***
## dislikes    -8.792e-02  1.860e-02  -4.728 2.29e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 76350 on 12336 degrees of freedom
## Multiple R-squared:  0.7667, Adjusted R-squared:  0.7666
## F-statistic: 2.027e+04 on 2 and 12336 DF,  p-value: < 2.2e-16
```

Plotting the Residuals

The Residuals vs Fitted plot now has an even straighter line of best fit and shows us, even more obviously than the last linear regression model, that the residuals have a linear pattern.

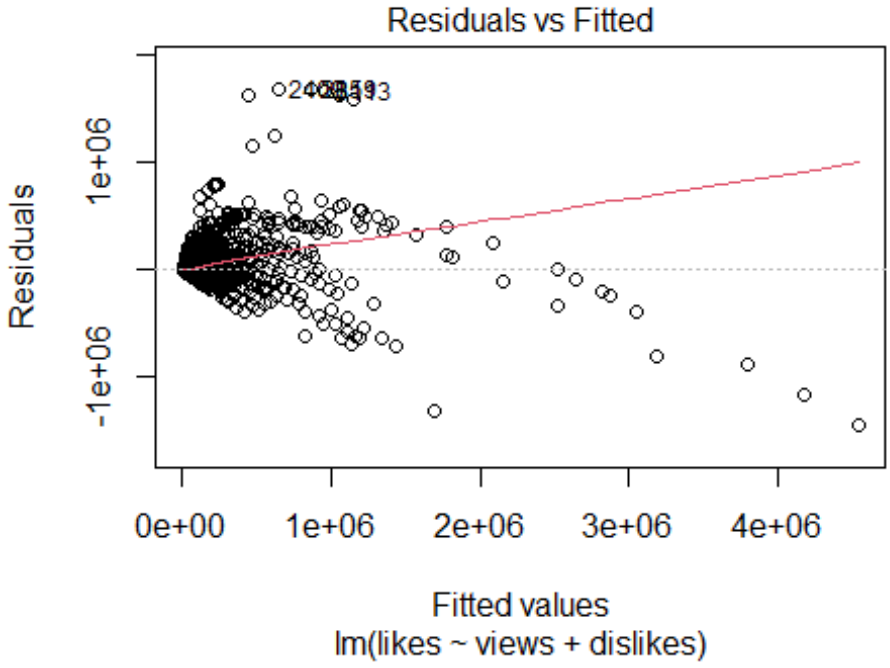
The Normal Q-Q plot looks the same as the last one, meaning that the residuals are normally distributed.

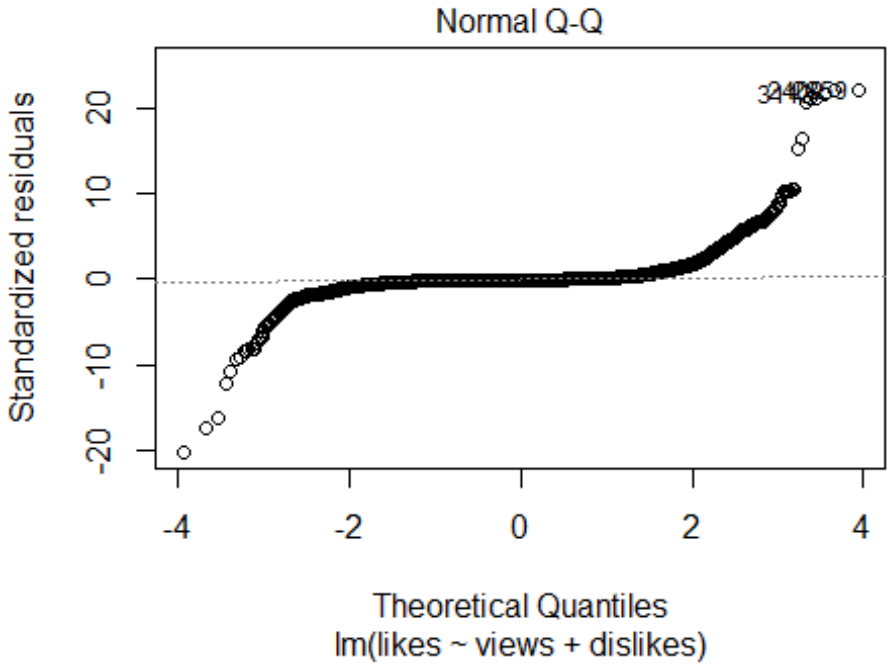
The Scale-Location Plot has similar structure to the last one as well, meaning that the linear regression using views and dislikes are not diversified and spread equally on the best fit line and are heavily left-sided

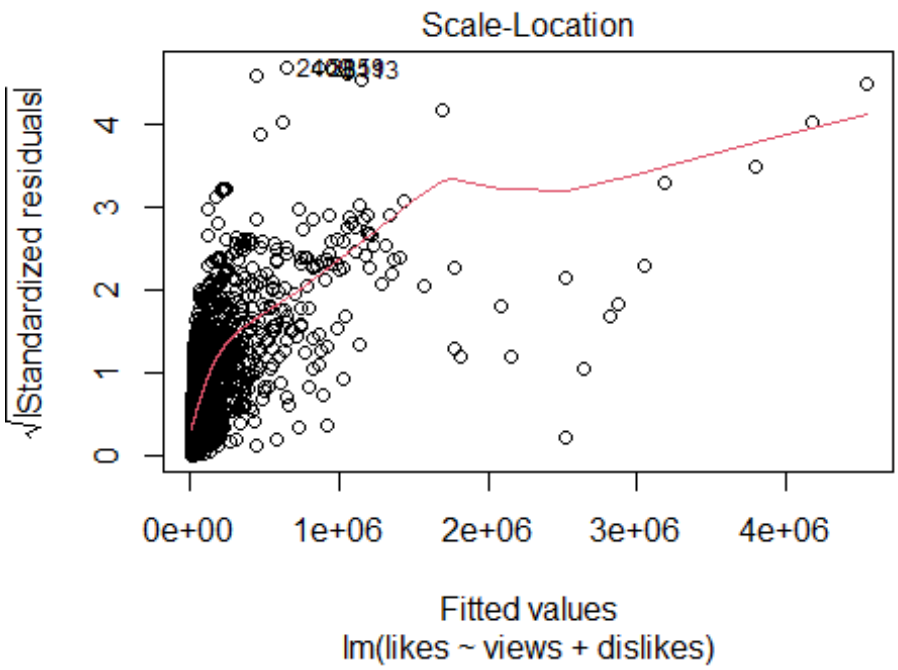
The Residuals vs Leverage plot shows us a lot more differences compared to the last one. There are more residuals on the right hand side and more variables that fall in or outside of

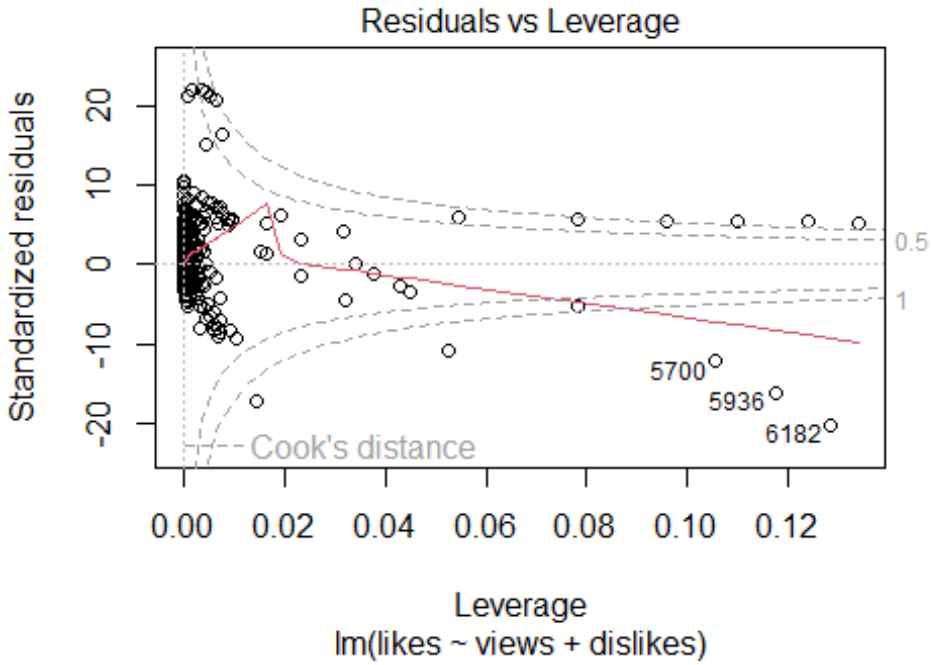
the Cook's distance. This means there are a lot more influential residuals to this linear regression model.

```
plot(lm2)
```









Third Linear Regression Model

Going to try out different combinations of predictors to see if we can get a more precise linear regression model

First, lets try doing views and comment_count. This already seems like a better model since the R-squared is significantly closer to one than using views and dislikes.

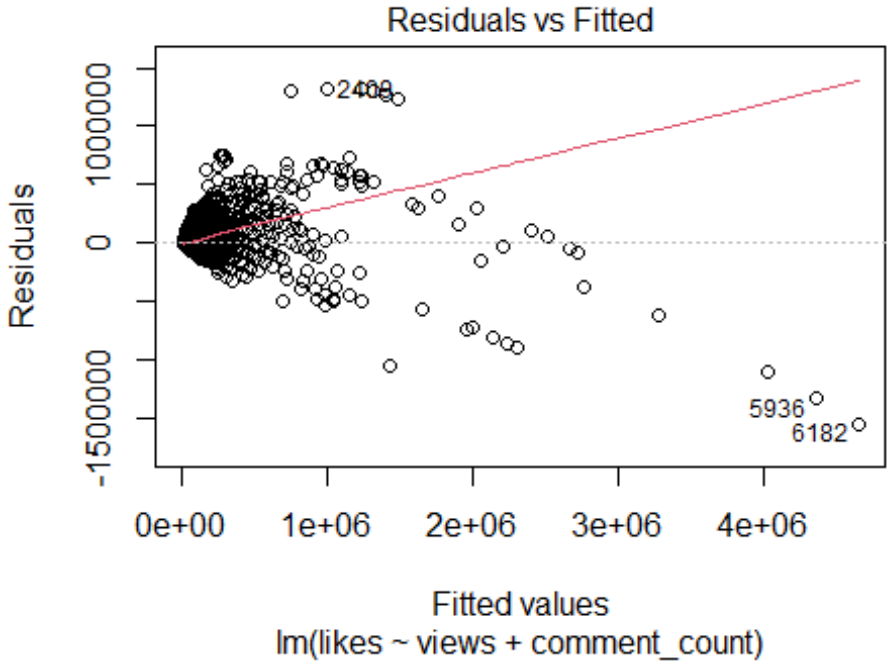
The Residuals vs Fitted plot has an even straighter line than the last model, but other than that the conclusions are the same.

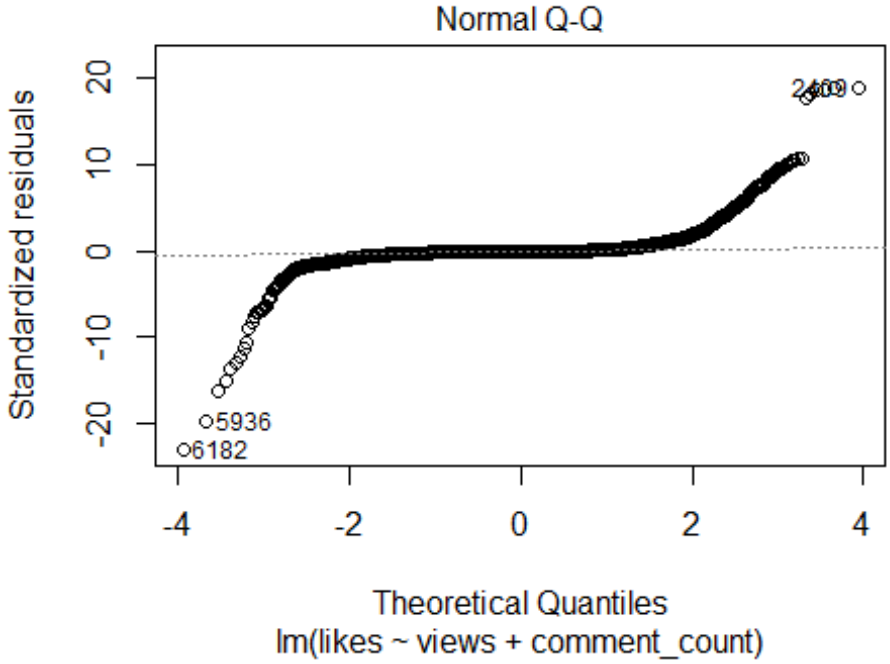
The Normal Q-Q plot is the exact same with the same conclusions able to be drawn

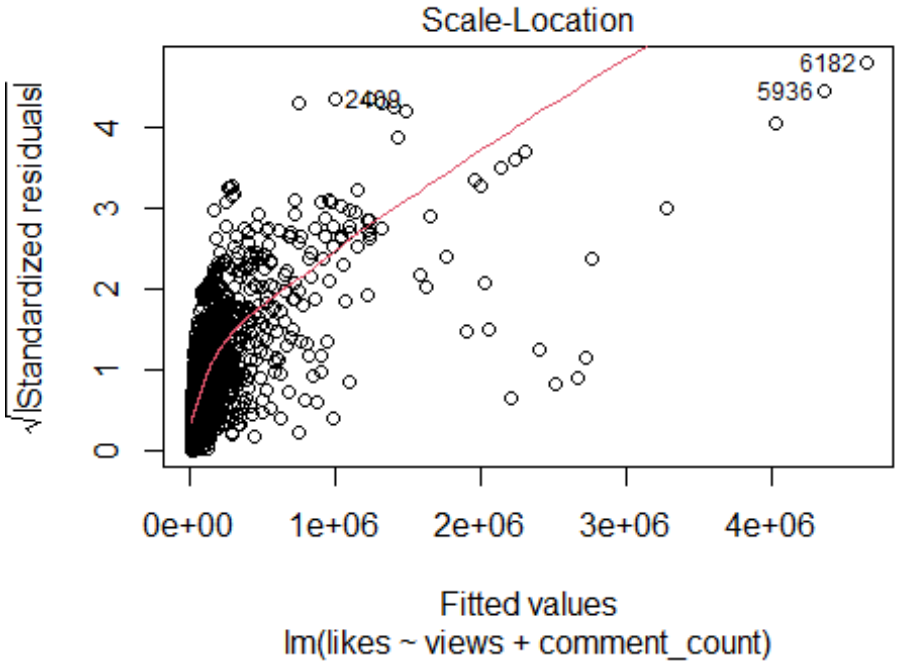
The Scale-Location plot is similar looking with the same conclusions able to be drawn

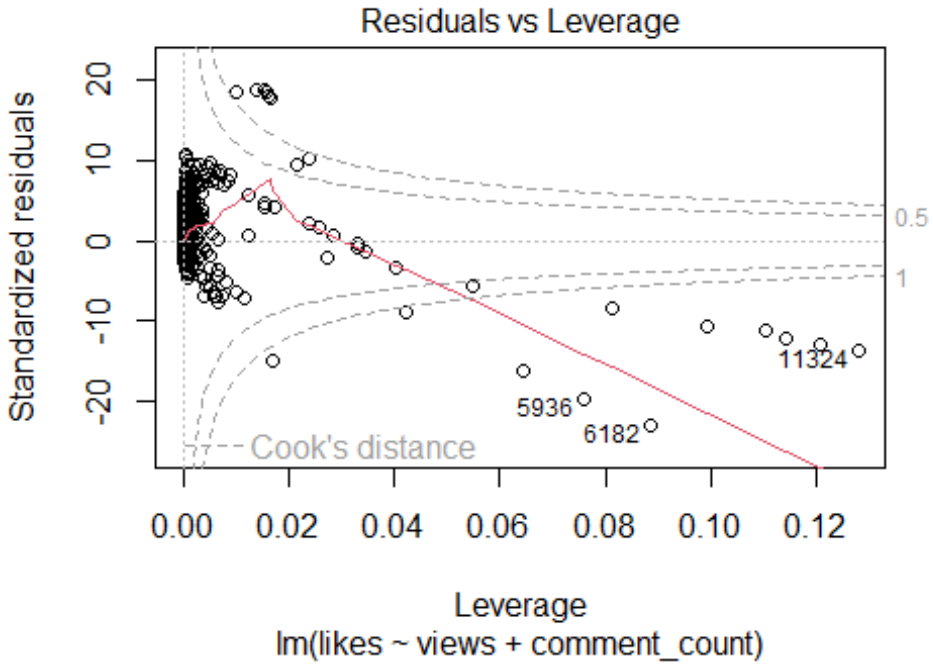
The Residuals vs Leverage plot contains even more influential residuals than the last model

```
lm3 <- lm(likes~views+comment_count, data=train)
summary(lm3)

##
## Call:
## lm(formula = likes ~ views + comment_count, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1551297   -11072    -8160    -2158  1316233
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.134e+03  6.591e+02   12.34   <2e-16 ***
## views         2.573e-02  1.787e-04  143.94   <2e-16 ***
## comment_count 9.789e-01  2.114e-02   46.31   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 70530 on 12336 degrees of freedom
## Multiple R-squared:  0.8009, Adjusted R-squared:  0.8008
## F-statistic: 2.481e+04 on 2 and 12336 DF,  p-value: < 2.2e-16

plot(lm3)
```


Residuals vs Fitted
Residuals
1000000
0
-1500000
2408
5936
6182
0e+00
1e+06
2e+06
3e+06
4e+06
Fitted values
lm(likes ~ views + comment_count)


Normal Q-Q
Standardized residuals
20
10
0
-10
-20
5936
6182
-4
-2
0
2
4
Theoretical Quantiles
lm(likes ~ views + comment_count)





Comparison of Results

These three results have mostly the same summaries and residual graphs. The biggest differences between all three of the results are with the R-squared values and the Residuals

vs Leverage plots. The R-squared value is the best with the model that shows the impact of views and comment count on likes. The Residuals vs Leverage plots also progressively show more and more values in or outside the Cook's distance. We can see which one of the models is the best using the anova() function.

Seeing all of the console outputs of the anova() function, we can see that the third model is the best because it lowers the RSS and the Sum of Squares.

```
anova(lm1, lm2)

## Analysis of Variance Table
##
## Model 1: likes ~ views
## Model 2: likes ~ views + dislikes
##   Res.Df        RSS Df Sum of Sq      F    Pr(>F)
## 1  12337 7.2036e+13
## 2  12336 7.1906e+13  1 1.303e+11 22.354 2.293e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(lm2, lm3)

## Analysis of Variance Table
##
## Model 1: likes ~ views + dislikes
## Model 2: likes ~ views + comment_count
##   Res.Df        RSS Df  Sum of Sq F Pr(>F)
## 1  12336 7.1906e+13
## 2  12336 6.1367e+13  0 1.0539e+13
```

Evaluate simple linear regression model on the test set

```
pred <- predict(lm1, newdata=test)
cor <- cor(pred, test$likes)
mse <- sqrt(mean((pred-test$likes)^2))
rmse <- sqrt(mse)

print(paste('correlation:', cor))

## [1] "correlation: 0.856249417349519"

print(paste('mse:', mse))

## [1] "mse: 83104.8130851465"

print(paste('rmse:', rmse))

## [1] "rmse: 288.279054190807"
```

Evaluate multiple linear regression model on the test set

```
pred2 <- predict(lm2, newdata=test)
cor2 <- cor(pred2, test$likes)
mse2 <- sqrt(mean((pred2-test$likes)^2))
```

```
rmse2 <- sqrt(mse2)

print(paste('correlation:', cor2))

## [1] "correlation: 0.854668956615879"

print(paste('mse:', mse2))

## [1] "mse: 83520.7918594811"

print(paste('rmse:', rmse2))

## [1] "rmse: 288.999639895072"
```

Evaluate third linear regression model on the test set

```
pred3 <- predict(lm3, newdata=test)
cor3 <- cor(pred3, test$likes)
mse3 <- sqrt(mean((pred3-test$likes)^2))
rmse3 <- sqrt(mse3)

print(paste('correlation:', cor3))

## [1] "correlation: 0.905730966677266"

print(paste('mse:', mse3))

## [1] "mse: 68405.0261905531"

print(paste('rmse:', rmse3))

## [1] "rmse: 261.543545495875"
```

Evaluation Results

The third regression model is the best model because it has the highest correlation and the lowest rmse