Regression

Code

Dmitrii Obideiko, Matthew McCoy

Hide

```
### Load data
library(readr)
yt_data <- read_csv("yt_data.csv")
```

```
Rows: 26061 Columns: 12── Column specification ────────────────────────────────
──────────────────────────────────────────────────────────────────────────────
──────────
Delimiter: ","
chr  (5): title, channel_title, tags, description, country_code
dbl  (6): video_id, category_id, views, dislikes, comment_count, likes
date (1): publish_date
ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Hide

```
### 2a - Divide the data into 80/20 trian/test - done
set.seed(1234)
i <- sample(1:nrow(yt_data), nrow(yt_data)*0.8, replace=FALSE)
train <- yt_data[i,]
test <- yt_data[-i,]
```

Hide

```
### 2b - 5 R fiunctions for data exploration - done
head(train)
```

video_id <dbl>
83707
31355
31622
63917
76399
12662

6 rows | 1-1 of 12 columns

Hide

```
tail(train)
```

video_id <dbl>
58646
33203
87363
17278
83720
79349

6 rows | 1-1 of 12 columns

Hide

```
str(train)
```

```
tibble [20,848 × 12] (S3: tbl_df/tbl/data.frame)
 $ video_id     : num [1:20848] 83707 31355 31622 63917 76399 ...
 $ title        : chr [1:20848] "CNN Turns to Pot" "Logan Paul: I'm A Good Guy Who Made
A Bad Decision - (Logan Paul Interview)" "Connor McDavid erupts with four-goal game" "ਵੱਡੀ
ਖਬਰ ! Tornado in Punjab | Dist Firozpur | Jeep!! Tractor!! in the Air" ...
 $ channel_title: chr [1:20848] "Mark Dice" "WildSpartanz" "NHL" "AggBani News" ...
 $ category_id  : num [1:20848] 25 24 17 25 26 10 22 24 27 24 ...
 $ publish_date : Date[1:20848], format: "2018-01-02" "2018-02-01" "2018-02-06" "2018-02
-12" ...
 $ tags         : chr [1:20848] "2018|\"CNN New Years Eve\"|\"Funny\"" "logan paul|\"log
an paul interview\"|\"paul\"|\"logan paul vlogs\"|\"logan\"|\"logan paul youtube\"|\"log
an paul"| __truncated__ "NHL|\"National Hockey League\"|\"Hockey\"|\"Hat Trick\"|\"YT Ha
t Trick\"|\"Hatty\"|\"Goal\"|\"Goals\"|\"Highlig"| __truncated__ "Tornado 2018|\"Tornado
Punjab\"|\"Weather News\"|\"Punjab Weather\"|\"Jeep Tractor\"|\"Wind\"|\"Punjabi News
\"|"| __truncated__ ...
 $ views        : num [1:20848] 247344 47881 22939 283383 574250 ...
 $ dislikes     : num [1:20848] 705 133 10 202 534 148 70 59 320 12 ...
 $ comment_count: num [1:20848] 5102 631 125 90 3527 ...
 $ description  : chr [1:20848] "Last year was rough for CNN.  They can't shake the 'fak
e news' label, and so they've found a new strategy hopin"| __truncated__ "Today I talked
about Logan Paul's Interview on Good Morning America and what I thought about it.\\n▶Mer
ch: http"| __truncated__ "Oilers phenom Connor McDavid put on a show in Edmonton en rout
e to his first career four-goal game, including t"| __truncated__ "L I K E | S H A R E |
C O M M E N T | S U B S C R I B E\\nਪੰਜਾਬ ਤੇ ਦੁਨਿਆ ਦੀ ਹਰ ਵੱਡੀ ਖਬਰ ਸਭ ਤੋ ਪਹਿਲਾ ਵੇਖਣ ਲਈ ਜੁੜੇ ਰਹੋ"| __trunca
ted__ ...
 $ country_code : chr [1:20848] "CA" "CA" "CA" "IN" ...
 $ likes        : num [1:20848] 19876 2494 496 1359 19585 ...
```

Hide

```
summary(train)
```

```
    video_id        title           channel_title       category_id     publish_date
tags               views               dislikes
 Min.   :    2   Length:20848       Length:20848       Min.   : 1.00   Min.   :2006-07-2
3   Length:20848       Min.   :      859   Min.   :     0.0
 1st Qu.:22584   Class :character   Class :character   1st Qu.:22.00   1st Qu.:2017-12-2
6   Class :character   1st Qu.:    67992   1st Qu.:    53.0
 Median :44758   Mode  :character   Mode  :character   Median :24.00   Median :2018-02-1
2   Mode  :character   Median :   153788   Median :   135.0
 Mean   :44612                                         Mean   :21.48   Mean   :2018-02-1
0                      Mean   :   390503   Mean   :   547.3
 3rd Qu.:66808                                         3rd Qu.:25.00   3rd Qu.:2018-04-1
1                      3rd Qu.:   337675   3rd Qu.:   349.0
 Max.   :89260                                         Max.   :43.00   Max.   :2018-06-1
4                      Max.   :143408308   Max.   :217017.0
 comment_count    description         country_code           likes
 Min.   :     0   Length:20848       Length:20848       Min.   :      0
 1st Qu.:    94   Class :character   Class :character   1st Qu.:    691
 Median :   391   Mode  :character   Mode  :character   Median :   2560
 Mean   :  1704                                         Mean   :  12644
 3rd Qu.:  1215                                         3rd Qu.:   8584
 Max.   :692312                                         Max.   :3880088
```

Hide

```
dim(yt_data)
```

```
[1] 26061    12
```

Hide

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

```
Call:
lm(formula = views ~ comment_count, data = train)

Residuals:
      Min        1Q    Median        3Q       Max
-20618784   -233707   -167614    -26019 132745597

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.677e+05  1.092e+04   24.51   <2e-16 ***
comment_count 7.208e+01  1.118e+00   64.49   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1553000 on 20846 degrees of freedom
Multiple R-squared:  0.1663,    Adjusted R-squared:  0.1663
F-statistic:  4160 on 1 and 20846 DF,  p-value: < 2.2e-16
```

2d - Simple Linear Regression

Call - shows us the formula that R used to fit the model.

Residuals - this section shows the difference between the predicted values and actual values. This applies to min, 1q, median, e1, and max. From the given summary, we can tell that the distribution is not symmetrical and is skewed to the left. This means that the model is not very good for predicting views with a high count.

Coefficients - this section shows the estimated coefficients of the linear model. This includes the slope and y intercept. For example, from the summary we can tell that the slop is 24.51 and y-intercept is 2.56e05

Residual standard error - this shows how much uncertainty there's with the calculated coefficient. We can use this information to predict confidence interval for the coefficient. From the calculated residual, we can tell that there's a lot of uncertainty with our coefficient.

Multiple R-Squared - this shows the proportion of the variation in dependent variable that can be explained by dependent variables. The higher the value for multiple R-Squared, the better. Since we got a failry low value, it means that the model is not good for the data.

F statistic is useful for finding evidence against the null hypothesis which is that predictors are bad. F statistic tells us if the results are statistically significant unlike R^2. Since our f value is fairly small, we reject that null hypothesis.

Since there's three asterisks, it means that comment_count was good predictor.

Since the p value is fairly small, we can reject the null hypothesis.

Hide

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

```
Call:
lm(formula = views ~ comment_count, data = train)

Residuals:
      Min        1Q    Median        3Q       Max
-20618784   -233707   -167614    -26019 132745597

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.677e+05  1.092e+04   24.51   <2e-16 ***
comment_count 7.208e+01  1.118e+00   64.49   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1553000 on 20846 degrees of freedom
Multiple R-squared:  0.1663,    Adjusted R-squared:  0.1663
F-statistic:  4160 on 1 and 20846 DF,  p-value: < 2.2e-16
```

2e - Plotting the Residuals

Since residuals are not equally spread out in residuals vs fitted graphs, it means that there’s no indication of a linear relationship.

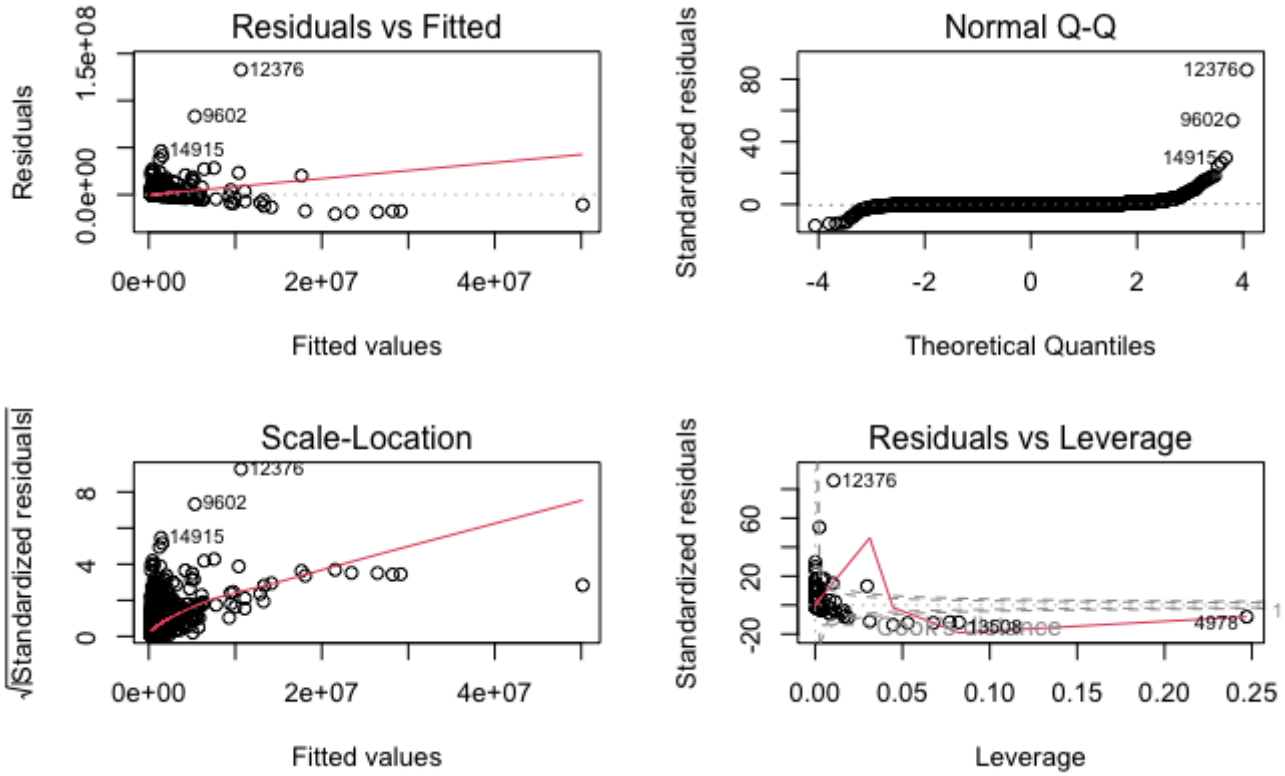
Since Normal Q-Q graph is normally distributed, it means that residuals are normally distributed.

Since you can’t see a horizontal like with equally spread points, it means that residuals are not spread equally.

Since there’s a dot in the right button corner, it means that there’s a value that is influential again a regression line.

Hide

```
par(mfrow=c(2,2))
plot(lm1)
```



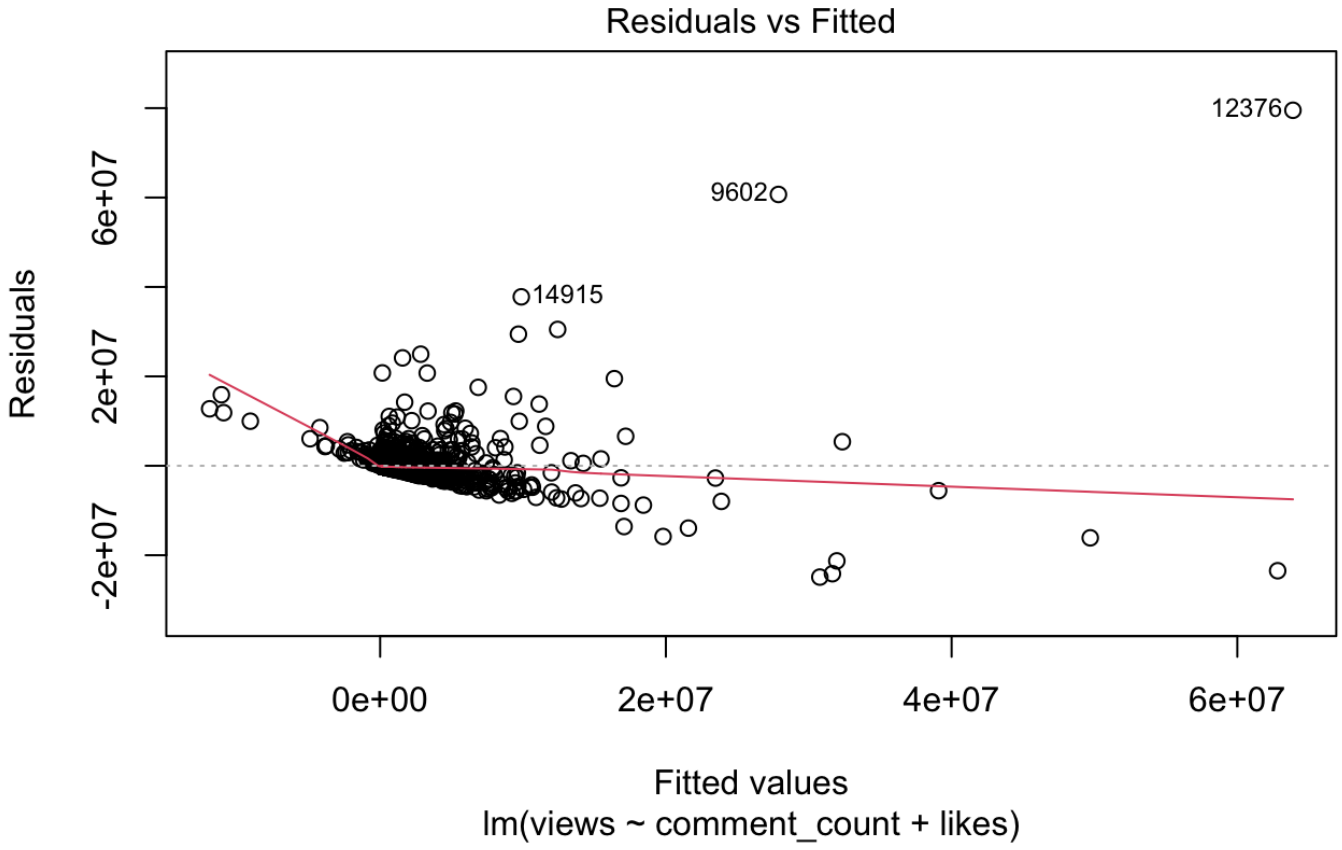
Hide

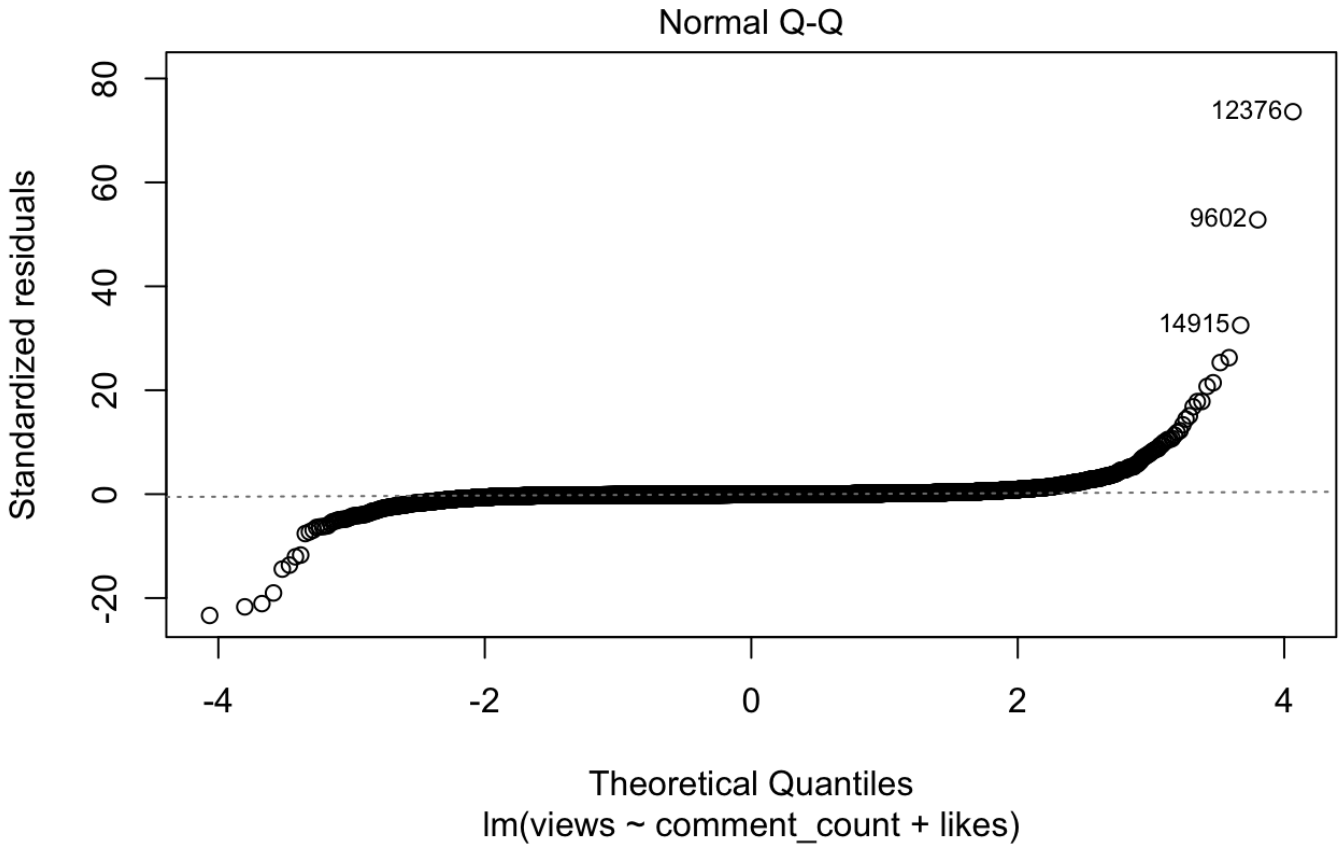
```
### 2f - Multiple Linear Regression Model - Done
lm2 <- lm(views~comment_count+likes, data=train)
summary(lm2)
```

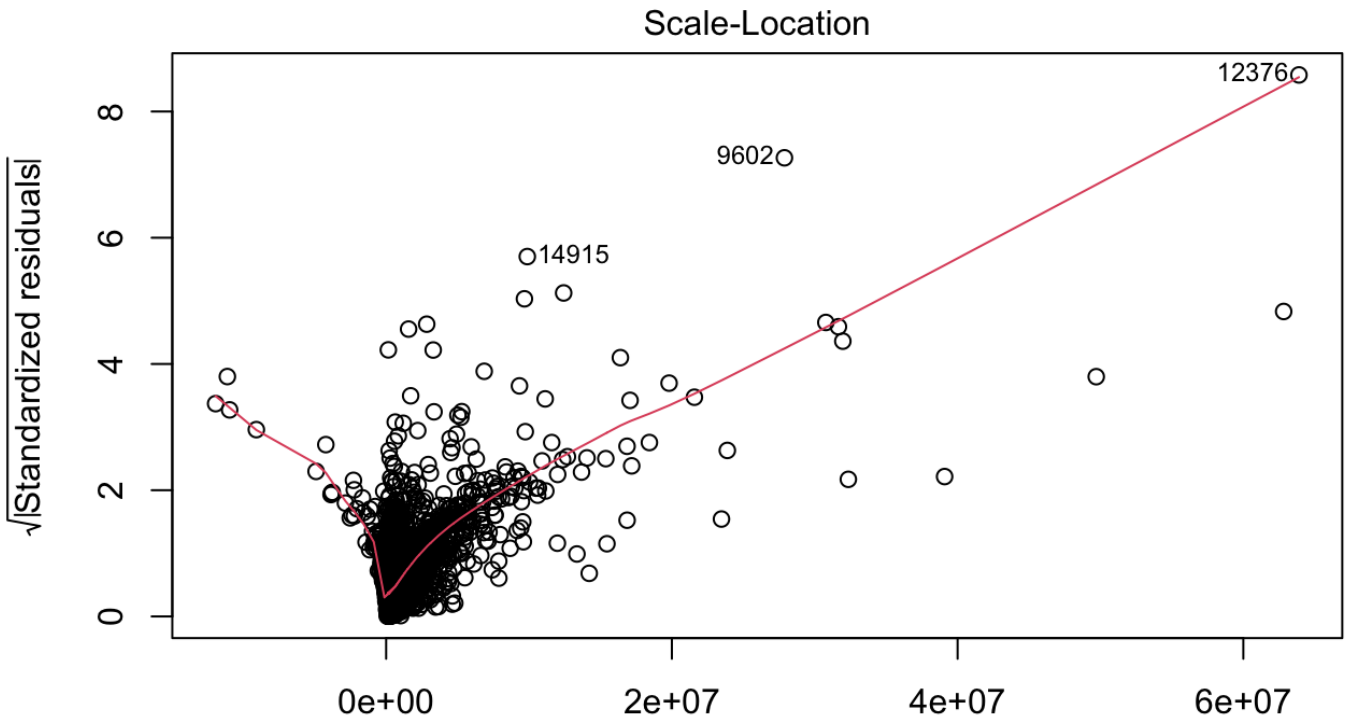
```

Call:
lm(formula = views ~ comment_count + likes, data = train)

Residuals:
      Min        1Q    Median        3Q       Max 
-24892986   -140636    -81510     37987  79515970 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)    1.528e+05  8.239e+03   18.54   <2e-16 ***
comment_count -6.074e+01  1.337e+00  -45.42   <2e-16 ***
likes          2.699e+01  2.118e-01  127.43   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1164000 on 20845 degrees of freedom
Multiple R-squared:  0.5314,    Adjusted R-squared:  0.5313 
F-statistic: 1.182e+04 on 2 and 20845 DF,  p-value: < 2.2e-16
```

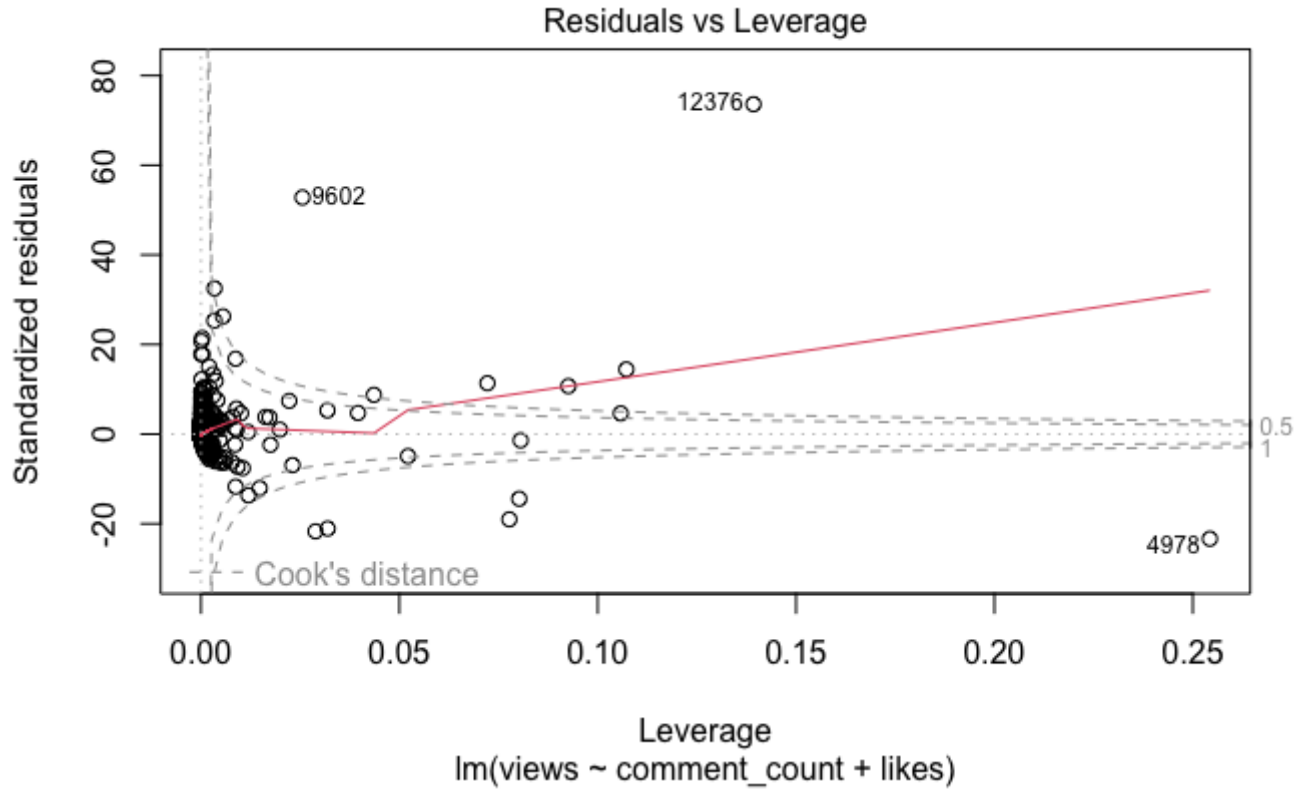
Hide

```
plot(lm2)
```


Residuals vs Fitted
12376
9602
14915
Residuals
6e+07
2e+07
-2e+07
0e+00
2e+07
4e+07
6e+07
Fitted values
lm(views ~ comment_count + likes)


Normal Q-Q
12376
9602
14915
Standardized residuals
80
60
40
20
0
-20
-4
-2
0
2
4
Theoretical Quantiles
lm(views ~ comment_count + likes)





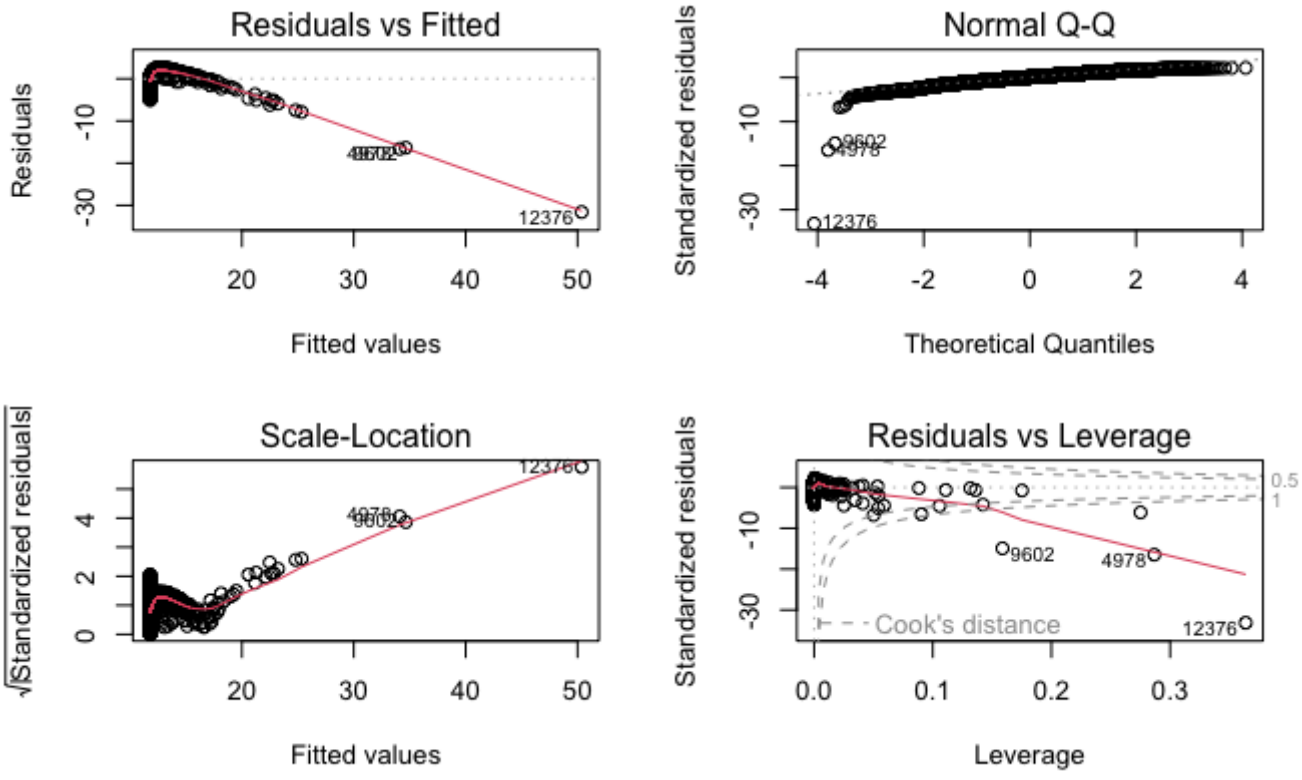
Hide

```
### 2g - Multiple Models are not always straight lines - done
lm3 <- lm(log(views)~views+likes+dislikes+comment_count, data=train)
summary(lm3)
```

```
Call:
lm(formula = log(views) ~ views + likes + dislikes + comment_count,
    data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-31.5886  -0.6923   0.0948   0.8166   2.5893

Coefficients:
               Estimate Std. Error  t value Pr(>|t|)
(Intercept)   1.179e+01  8.523e-03 1383.606  < 2e-16 ***
views         2.063e-07  7.633e-09   27.033  < 2e-16 ***
likes         2.240e-06  2.907e-07    7.707 1.34e-14 ***
dislikes      1.484e-05  3.212e-06    4.619 3.88e-06 ***
comment_count 6.382e-06  1.523e-06    4.192 2.78e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.195 on 20843 degrees of freedom
Multiple R-squared:  0.1612,    Adjusted R-squared:  0.161
F-statistic:  1001 on 4 and 20843 DF,  p-value: < 2.2e-16
```

Hide

```
par(mfrow=c(2,2))
plot(lm3)
```



2h - Comparing the results

It looks like the second model performed the best as it has the highest multiple R-squared variable, which means that the second model is the best out of all three for predicting views. In addition, to support that, the second model also has the highest F-statistic which indicates that our results are significant and we can reject the null hypothesis that our predictors are bad. Since every model had a low p-value, we can reject the null hypotheses for other models as well. Model 3 has the smallest residual standard error, which means that the predictions are more accurate for this model. I think the second model gave the best results because videos with more likes do tend to get more views on average.

2i - Comparing the results using Metrics Correlation and MSE

From what it looks like, the second model performed the best as it has the highest correlation. The only difference between the first model and the second model is that we added the number of likes into our model, which means that it's a good predictor when it comes to views. The only difference between the second model and the third model is that we added the number of dislikes, which indicates to us the number of dislikes is a bad predictor when it comes to the number of views. This is perhaps because the more likes a video has, the more likely a person would click on the video to watch it. If the person sees that that a video has more likes than dislikes, perhaps that makes them think that the video is not good, which makes them want to not click on it and thus view it.

Hide

```
### 2i - Predicict, evaluate, compare the results

# First model
print("First model")
```

```
[1] "First model"
```

Hide

```
pred1 <- predict(lm1, newdata=test)
cor1 <- cor(pred1, test$views)
mse1 <- mean((pred1-test$views)^1)
rmse1 <- sqrt(mse1)
print(paste('correlation:', cor1))
```

```
[1] "correlation: 0.437396830208616"
```

Hide

```
print(paste('mse:', mse1))
```

```
[1] "mse: 10628.8343433396"
```

Hide

```
print(paste('rmse:', rmse1))
```

```
[1] "rmse: 103.096238259888"
```

Hide

```
# Second Model
print("Second model")
```

```
[1] "Second model"
```

Hide

```
pred2 <- predict(lm2, newdata=test)
cor2 <- cor(pred2, test$views)
mse2 <- mean((pred2-test$views)^2)
rmse2 <- sqrt(mse2)
print(paste('correlation:', cor2))
```

```
[1] "correlation: 0.74080515428932"
```

Hide

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

```
[1] "mse: 582062994668.313"
```

Hide

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

```
[1] "rmse: 762930.530696152"
```

Hide

```
# Third Model
print("Third model")
```

```
[1] "Third model"
```

Hide

```
pred3 <- predict(lm3, newdata=test)
cor2 <- cor(pred3, test$views)
mse2 <- mean((pred3-test$views)^2)
rmse2 <- sqrt(mse3)
print(paste('correlation:', cor3))
```

```
[1] "correlation: 0.567299678589398"
```

Hide

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

```
[1] "mse: 52194222003720"
```

Hide

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

```
[1] "rmse: 7224556.87248153"
```