MLR and YouTube Videos

Xinyi Lu and Daiyan Zhang

Introduction

Response:

Views: number of views of the video

Predictors:

Subscribers: number of subscribers of the channel

CC: manual subtitles (0 means the transcript is auto–generated, 1 means manual subtitles)

Released: time of the video released

Category: Category of the channel

Transcripts: subtitles for the video

Length: duration of the video

Dimation:

Rows: 2098

Columns: 11

Sources: https://www.kaggle.com/datasets/praneshmukhopadhyay/youtubers-saying-things

Data cleaning and manipulation

	URL <chr>	Views <chr>	Subscribers <chr>	Released <chr>	Length <chr>
1	https://www.youtube.com/watch?v=FozCkl1xj-w	7.9M views	6.28M subscribers	2 years ago	13:32
2	https://www.youtube.com/watch?v=RN8yoi-e2yc	2.7M views	1.9M subscribers		24:26
3	https://www.youtube.com/watch?v=IugclAAZJ2M	11M views	4.59M subscribers	2 years ago	7:51
4	https://www.youtube.com/watch?v=JiEO6F8i0eU	2.3M views	282K subscribers	3 years ago	10:06
5	https://www.youtube.com/watch?v=1T4XMNN4bNM	21M views	17.4M subscribers	9 years ago	9:29
6	https://www.youtube.com/watch?v=0ZWGeidvrJw	8.5M views	1.59M subscribers	7 years ago	4:20
7	https://www.youtube.com/watch?v=YiEj9mrqTN0	14M views	7.93M subscribers	2 years ago	20:54
8	https://www.youtube.com/watch?v=PZFLM2DVQHs	502K views	389K subscribers	6 years ago	33:13
9	https://www.youtube.com/watch?v=CoDpjqZpAh0	583K views	216K subscribers	6 months ago	7:17
10	https://www.youtube.com/watch?v=VT128ElBWkM	2.2M views	1.62M subscribers	4 years ago	4:20

	URL <chr>	Views <dbl>	Subscribers <dbl>	Released <dbl>	Length <dbl>
1	https://www.youtube.com/watch?v=FozCkl1xj-w	7900	6280	24	14
2	https://www.youtube.com/watch?v=IugclAAZJ2M	11000	4590	24	8
3	https://www.youtube.com/watch?v=JiEO6F8i0eU	2300	282	36	11
4	https://www.youtube.com/watch?v=1T4XMNN4bNM	21000	17400	108	10
5	https://www.youtube.com/watch?v=0ZWGeidvrJw	8500	1590	84	5
6	https://www.youtube.com/watch?v=YiEj9mrqTN0	14000	7930	24	21
7	https://www.youtube.com/watch?v=PZFLM2DVQHs	502	389	72	34
8	https://www.youtube.com/watch?v=CoDpjqZpAh0	583	216	6	8
9	https://www.youtube.com/watch?v=VT128ElBWkM	2200	1620	48	5
10	https://www.youtube.com/watch?v=AmKX9tCVtUE	5500	5720	36	23

Sentiment Analysis

1. Tokenize the word in Transcript, anti join the stop words, and use **get_sentiments(“afinn”)** to get the afinn value.
   Note: afinn is from -5 to 5.

2. Do tf-idf of words to give a weight of a word’s importance in a viedo transcript.
   I.e. **bind_tf_idf(word, Id, n), outcome is tf_idf and n**
   n: Column containing document-term counts

3. **afinn_score = sum (value * tf_idf)**
   same as Title’s afinn score: **afinn_title_score**

```
> glimpse(df1)
Rows: 2,098
Columns: 13
$ Id                <chr> "FozCkl1xj-w", "IugcIAAZJ2M", "JiEO6F8i0eU", "1T4XMNN4bNM", "0ZWGeidvrJw'
$ Channel           <chr> "JRE Clips", "Munchies", "Parks and Recreation", "Vsauce", "Doctor Who",
$ Subscribers       <dbl> 6280, 4590, 282, 17400, 1590, 7930, 389, 216, 1620, 5720, 4010, 847, 3930
$ Title             <chr> "Former CIA Agent Breaks Down Jeffrey Epstein Case", "The Iconic $1 Pizza
$ CC                <fct> 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0
$ URL               <chr> "https://www.youtube.com/watch?v=FozCkl1xj-w", "https://www.youtube.com/w
$ Released          <dbl> 24, 24, 36, 108, 84, 24, 72, 6, 48, 36, 24, 24, 72, 36, 48, 24, 72, 60, 2
$ Views             <dbl> 7900, 11000, 2300, 21000, 8500, 14000, 502, 583, 2200, 5500, 38000, 3300,
$ Category          <fct> "Blog", "Food", "Entertainment,Comedy", "Science", "Entertainment", "News
$ Transcript        <chr> "the Joe Rogan experience well how about the other so you gotta go to oka
$ Length            <dbl> 14, 8, 11, 10, 5, 21, 34, 8, 5, 23, 24, 11, 11, 16, 7, 3, 13, 14, 37, 14,
$ afinn_score       <dbl> -0.157162956, -0.019053874, 0.027645884, -0.454690380, 0.227942565, -0.12
$ afinn_title_score <dbl> 0.0000000, 0.0000000, 0.0000000, 0.0000000, 3.0189355, 0.0000000, 0.00000
```

stop_words

	word	lexicon
1	a	SMART
2	a's	SMART
3	able	SMART
4	about	SMART
5	above	SMART
6	according	SMART
7	accordingly	SMART
8	across	SMART
9	actually	SMART
10	after	SMART
11	afterwards	SMART
12	again	SMART
13	against	SMART
14	ain't	SMART
15	all	SMART
16	allow	SMART

Clean Data Frame and Variables

Response:
Views: number of views of the video (unit: thousand)

Predictors:
Subscribers: number of subscribers of the channel (unit: thousand)
CC: manual subtitles (0 means the transcript is auto–generated, 1 means manual subtitles)
Released: time of the video released (unit: month)
Category: Category of the channel
afinn_score: afinn value of video subtitles
afinn_title_score: afinn value of video Title
Length: duration of the video (unit: minute)

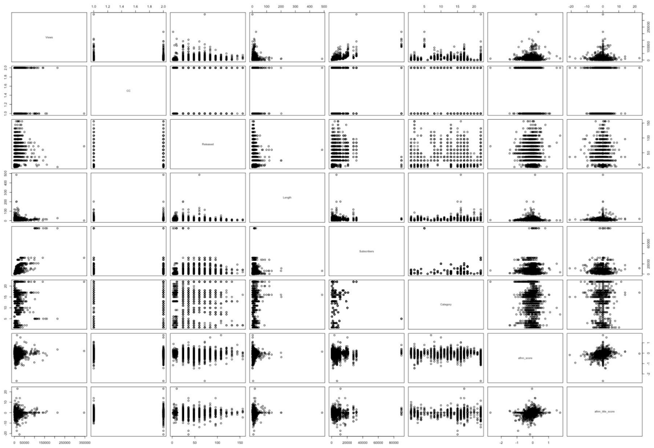
Dimation:
Rows: 2098
Columns: 13

Goal:

Find the best model to predict the number of YouTube video Views using Subscribers, CC, Released, Category, afinn_score, afinn_title_score and Length.

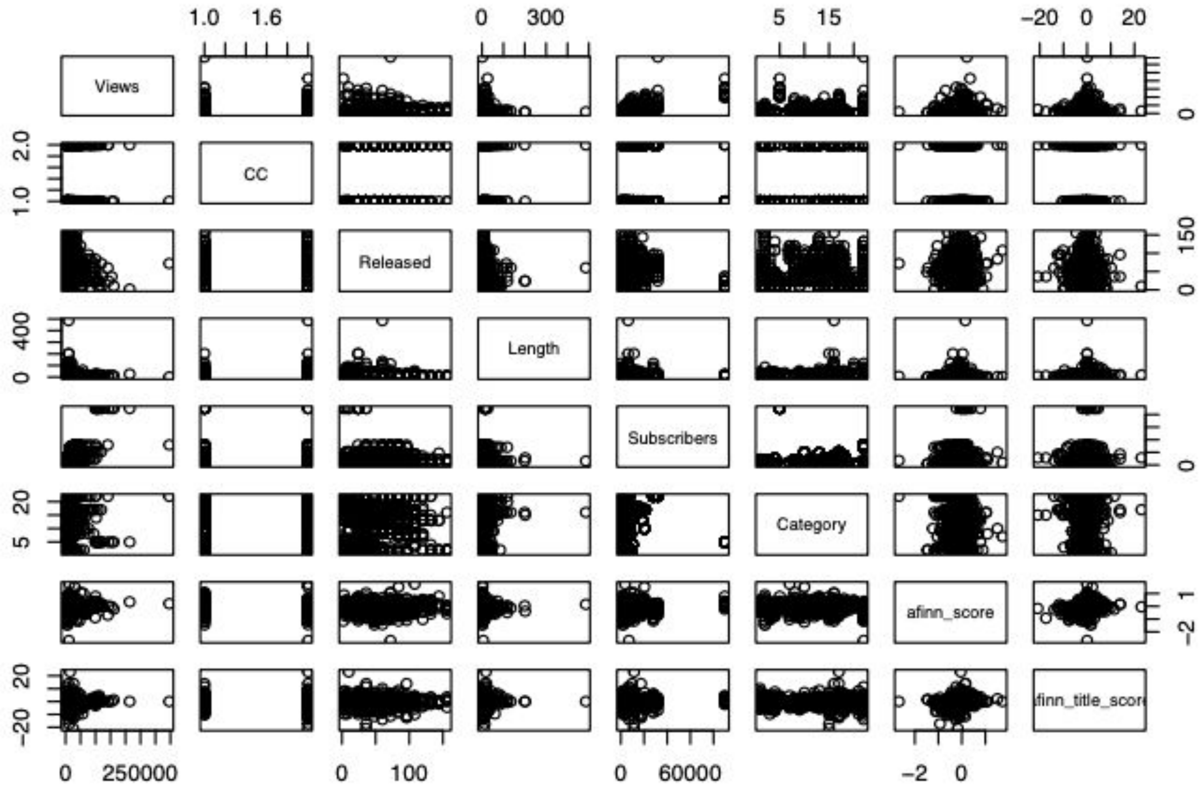
Diagnostics for MLR

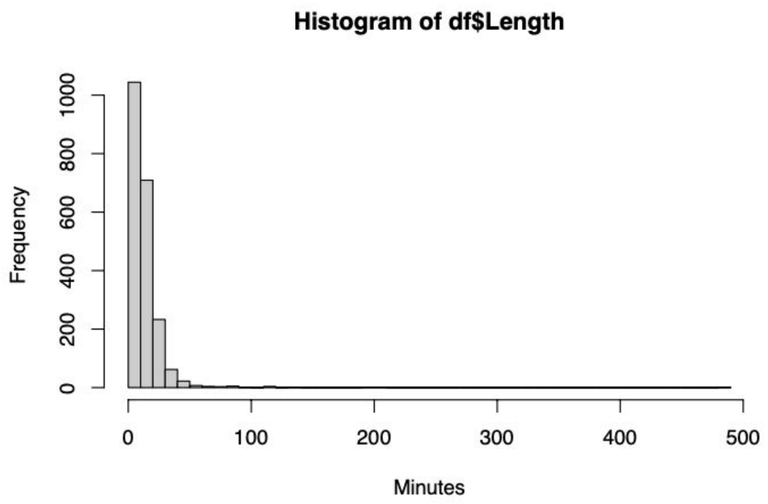
pairs(Views ~ CC + Released + Length + Subscribers + Category + afinn_score + afinn_title_score, data=df1)

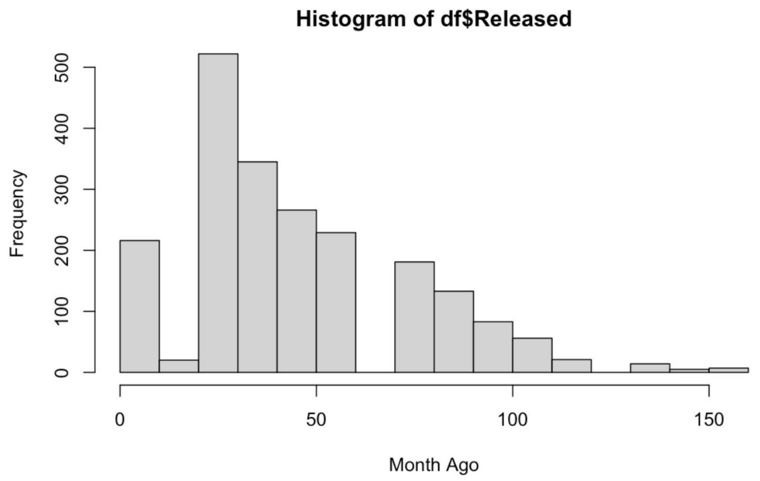


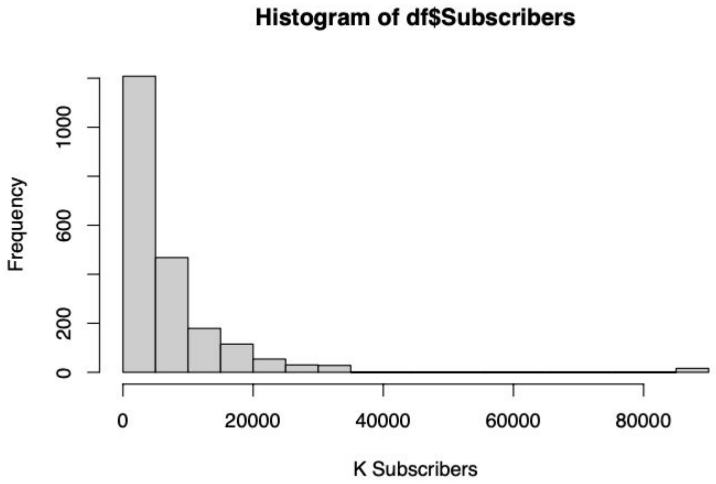
Diagnostics for MLR

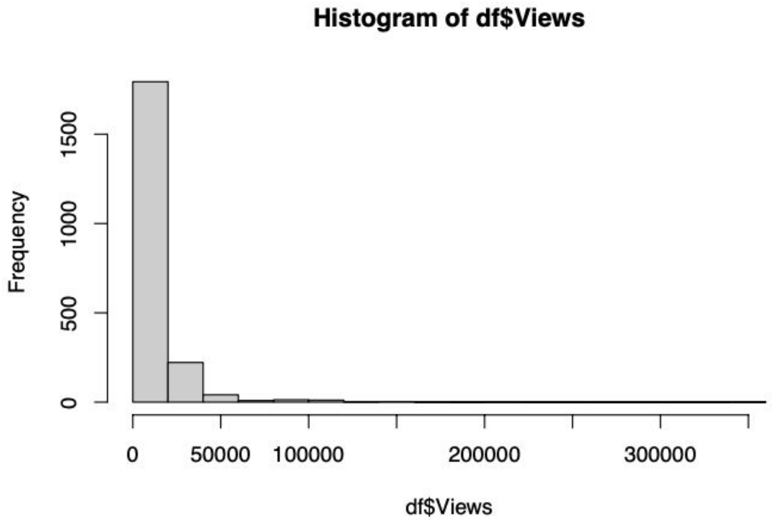
pairs(Views ~ CC + Released + Length + Subscribers + Category + afinn_score + afinn_title_score, data=df1)



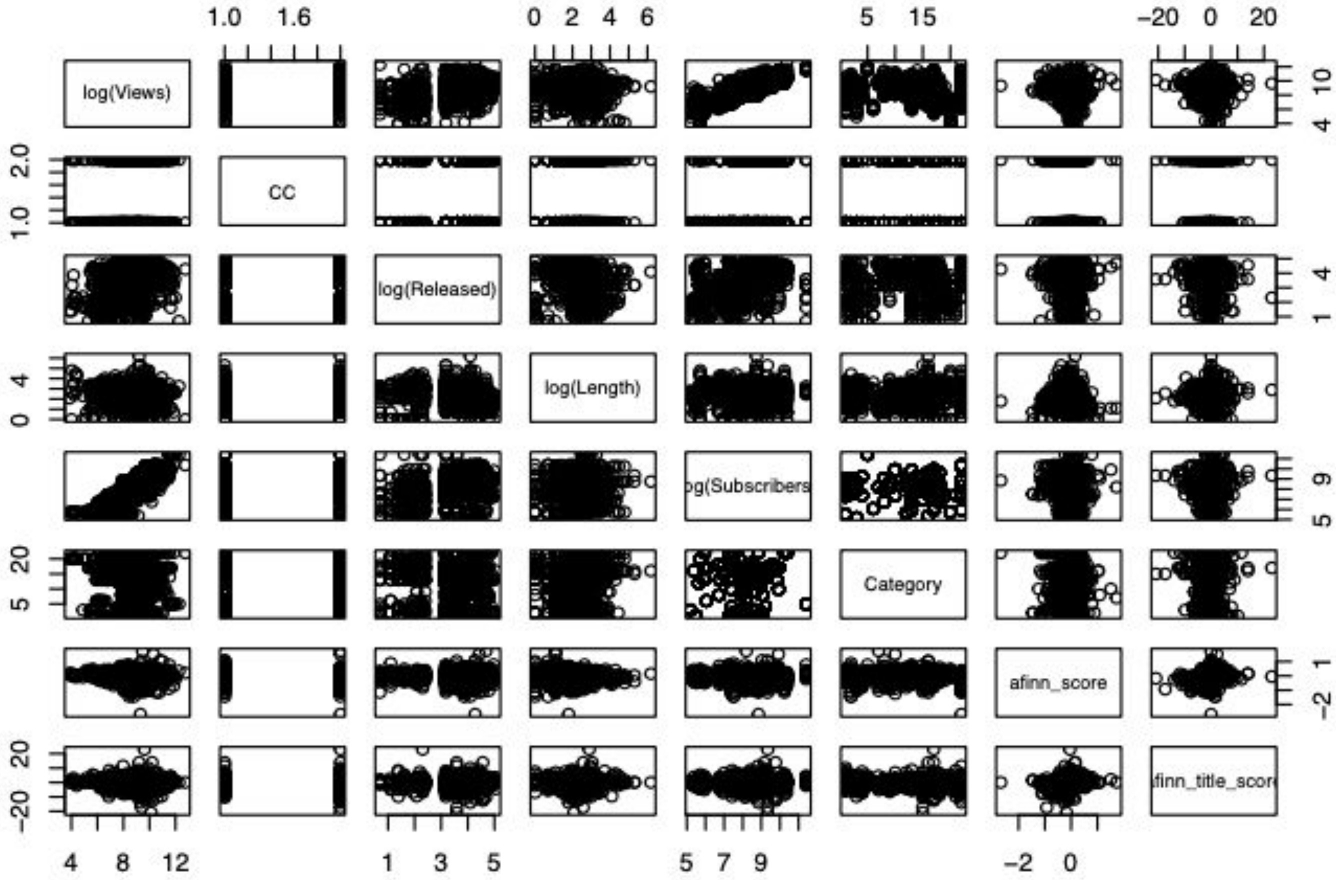

Histogram of df$Length
Frequency
0
200
400
600
800
1000
0
100
200
300
400
500
Minutes


Histogram of df$Released
Frequency
0
100
200
300
400
500
0
50
100
150
Month Ago


Histogram of df$Subscribers
Frequency
0
200
600
1000
0
20000
40000
60000
80000
K Subscribers


Histogram of df$Views
Frequency
0
500
1000
1500
0
50000
100000
200000
300000
df$Views

```
pairs(log(Views) ~
      CC + log(Released) + log(Length) + log(Subscribers) + Category + afinn_score + afinn_title_score, data=df1)
```



Multicollinearity?

##	Subscribers	Released	Length	afinn_score	afinn_title_score
## Subscribers	1.00	0.09	0.02	-0.02	0.01
## Released	0.09	1.00	-0.18	-0.05	0.03
## Length	0.02	-0.18	1.00	0.03	0.03
## afinn_score	-0.02	-0.05	0.03	1.00	0.22
## afinn_title_score	0.01	0.03	0.03	0.22	1.00

From the matrix of scatterplots and correlation table, we didn't see any obvious strong correlation between the predictors.

MLR modeling

lm_full <- lm(Views ~ CC + Released + Length + Subscribers + Category + afinn_score+ afinn_title_score, data=df1)
summary(lm_full)

```
Call:
lm(formula = Views ~ CC + Released + Length + Subscribers + Category +
    afinn_score + afinn_title_score, data = df1)

Residuals:
   Min     1Q Median     3Q    Max
-25663  -4088   -883   1907 299360

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  5.013e+02  1.034e+03   0.485 0.627801
CC1                          5.904e+02  5.813e+02   1.016 0.309909
Released                     1.656e+00  9.779e+00   0.169 0.865512
Length                      -2.443e+01  1.587e+01  -1.539 0.123935
Subscribers                  1.489e+00  4.445e-02  33.487  < 2e-16 ***
CategoryAutomobile,Comedy    5.772e+03  1.647e+03   3.504 0.000468 ***
CategoryBlog                -8.700e+02  1.362e+03  -0.639 0.522927
CategoryBlog,Comedy         -2.030e+03  1.975e+03  -1.028 0.304224
CategoryBlog,Entertainment  -1.060e+04  4.895e+03  -2.165 0.030486 *
CategoryBlog,Science        -3.780e+02  3.591e+03  -0.105 0.916190
CategoryComedy               1.086e+04  2.341e+03   4.637 3.75e-06 ***
CategoryComedy,Entertainment 9.045e+03  1.555e+03   5.815 7.02e-09 ***
CategoryComedy,Informative   1.114e+04  1.946e+03   5.725 1.19e-08 ***
CategoryEntertainment        3.926e+03  1.848e+03   2.124 0.033754 *
CategoryEntertainment,Blog   2.765e+03  2.393e+03   1.156 0.247943
CategoryEntertainment,Comedy 6.283e+03  1.564e+03   4.017 6.12e-05 ***
CategoryFood                 2.560e+03  1.180e+03   2.170 0.030109 *
CategoryFood,Entertainment   8.604e+03  2.548e+03   3.377 0.000745 ***
CategoryInformative          2.401e+02  1.251e+03   0.192 0.847864
CategoryNews                 1.351e+03  1.348e+03   1.002 0.316437
CategoryScience              1.968e+03  1.181e+03   1.667 0.095759 .
CategoryTech                -3.942e+03  1.274e+03  -3.094 0.002001 **
CategoryTech,Comedy         -6.025e+02  2.617e+03  -0.230 0.817916
CategoryTech,Informative    -4.183e+02  2.813e+03  -0.149 0.881818
CategoryTech,News           -1.998e+03  2.942e+03  -0.679 0.497212
CategoryVideoGames          -1.694e+03  1.239e+03  -1.367 0.171766
afinn_score                  1.187e+03  9.982e+02   1.189 0.234713
afinn_title_score           -1.251e+02  1.167e+02  -1.073 0.283554
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11470 on 2070 degrees of freedom
Multiple R-squared:  0.5998,    Adjusted R-squared:  0.5946
F-statistic: 114.9 on 27 and 2070 DF,  p-value: < 2.2e-16
```

R squared: 0.5998
Adjusted R squared: 0.5946

MLR modeling

lm1 <- lm(log(Views) ~ CC + log(Released) + log(Length) + log(Subscribers) + Category + afinn_score+ afinn_title_score, data=df1)
summary(lm1)

```
Call:
lm(formula = log(Views) ~ CC + log(Released) + log(Length) +
    log(Subscribers) + Category + afinn_score + afinn_title_score,
    data = df1)

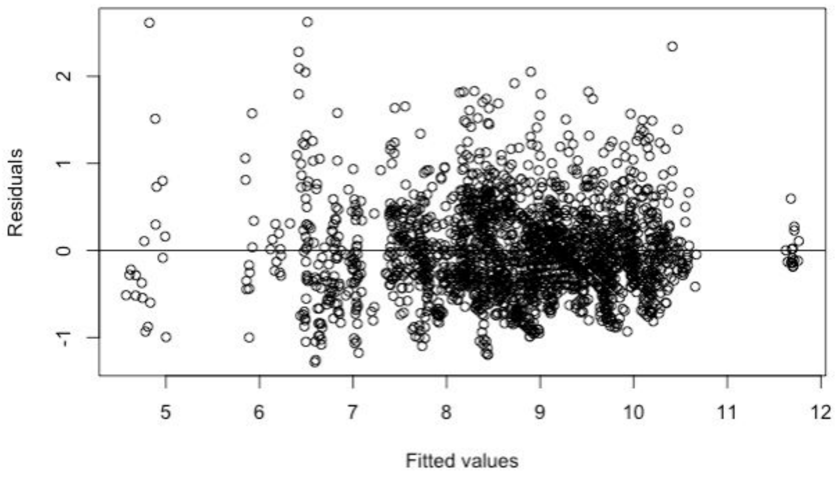
Residuals:
     Min       1Q   Median       3Q      Max
-1.28291 -0.34836 -0.06359  0.29517  2.62433

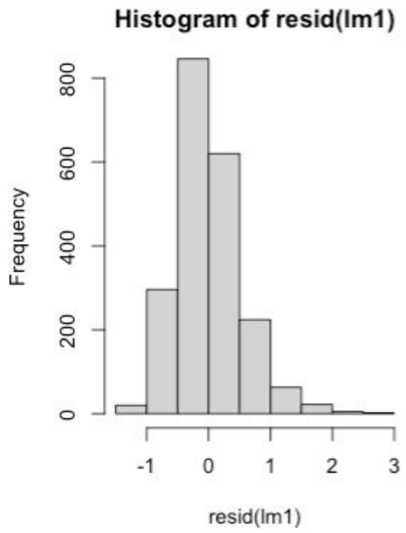
Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                   1.994258   0.115062  17.332  < 2e-16 ***
CC1                           0.061293   0.027270   2.248 0.024702 *
log(Released)                 0.010325   0.016986   0.608 0.543333
log(Length)                  -0.081625   0.016780  -4.864 1.23e-06 ***
log(Subscribers)              0.828092   0.012171  68.038  < 2e-16 ***
CategoryAutomobile,Comedy     0.404303   0.076396   5.292 1.34e-07 ***
CategoryBlog                 -0.218388   0.063018  -3.465 0.000540 ***
CategoryBlog,Comedy          -0.133908   0.091557  -1.463 0.143737
CategoryBlog,Entertainment    0.458470   0.147371   3.111 0.001890 **
CategoryBlog,Science         -0.626816   0.167315  -3.746 0.000184 ***
CategoryComedy                0.950982   0.108783   8.742  < 2e-16 ***
CategoryComedy,Entertainment  0.832074   0.072040  11.550  < 2e-16 ***
CategoryComedy,Informative    0.785949   0.089692   8.763  < 2e-16 ***
CategoryEntertainment         0.393832   0.085010   4.633 3.83e-06 ***
CategoryEntertainment,Blog    0.321985   0.110675   2.909 0.003661 **
CategoryEntertainment,Comedy  0.904802   0.072662  12.452  < 2e-16 ***
CategoryFood                  0.295938   0.054581   5.422 6.58e-08 ***
CategoryFood,Entertainment    0.511487   0.118622   4.312 1.69e-05 ***
CategoryInformative           0.040345   0.057819   0.698 0.485392
CategoryNews                  0.199804   0.062303   3.207 0.001362 **
CategoryScience              -0.049413   0.054384  -0.909 0.363669
CategoryTech                 -0.583077   0.058816  -9.914  < 2e-16 ***
CategoryTech,Comedy          -0.381650   0.120859  -3.158 0.001612 **
CategoryTech,Informative     -1.785302   0.132530 -13.471  < 2e-16 ***
CategoryTech,News            -1.015799   0.136331  -7.451 1.35e-13 ***
CategoryVideoGames           -0.202371   0.056352  -3.591 0.000337 ***
afinn_score                  -0.057197   0.046152  -1.239 0.215366
afinn_title_score            -0.006662   0.005383  -1.238 0.215996
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

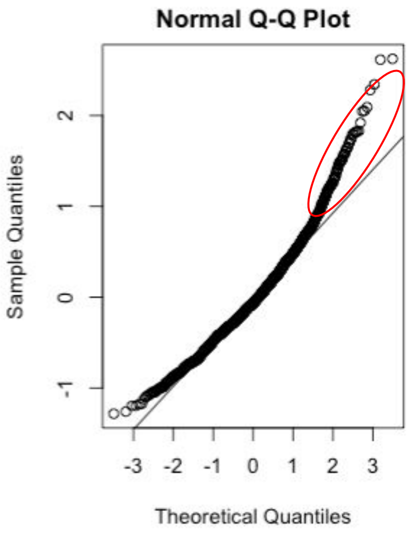
Residual standard error: 0.5293 on 2070 degrees of freedom
Multiple R-squared:  0.8154,    Adjusted R-squared:  0.813
F-statistic: 338.7 on 27 and 2070 DF,  p-value: < 2.2e-16
```

R squared: 0.8154
Adjusted R squared: 0.813

Assumption?







Variances are not constant
Right skewed. Normality isn't okay too.

Stepwise Selection

lm1 <- lm(log(Views) ~ CC + log(Released) + log(Length) + log(Subscribers) + Category + afinn_score+ afinn_title_score)

lm4 <- step(lm1)

log(Views) ~ CC + log(Length) + log(Subscribers) + Category + afinn_score

```
Call:
lm(formula = log(Views) ~ CC + log(Length) + log(Subscribers) +
    Category + afinn_score, data = df1)

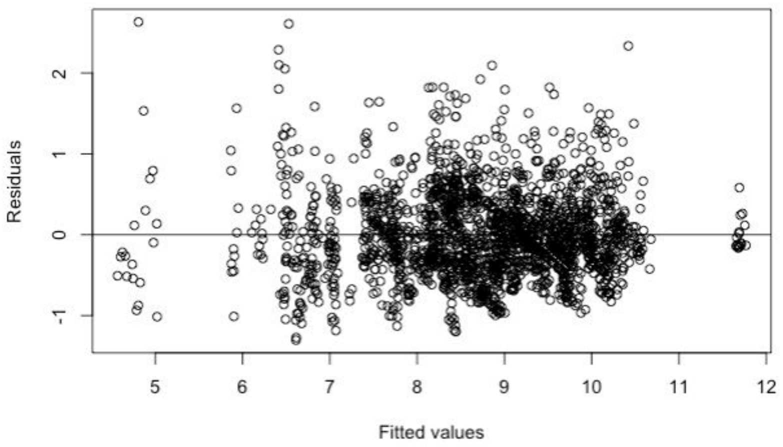
Residuals:
     Min       1Q   Median       3Q      Max
-1.30364 -0.34696 -0.06379  0.29275  2.63096

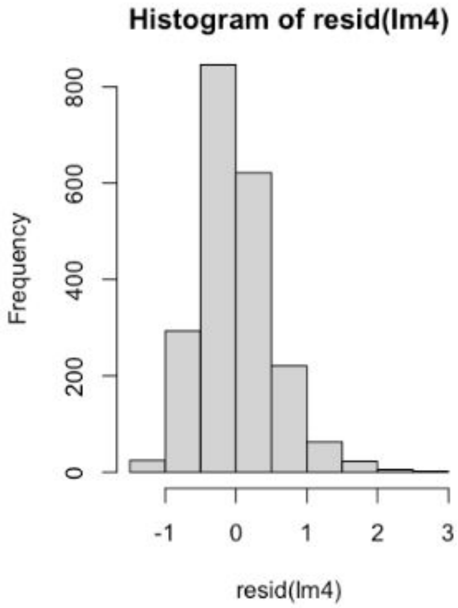
Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                   2.01453    0.10738  18.761  < 2e-16 ***
CC1                           0.06063    0.02711   2.237 0.025411 *
log(Length)                  -0.08380    0.01647  -5.087 3.97e-07 ***
log(Subscribers)              0.83081    0.01169  71.084  < 2e-16 ***
CategoryAutomobile,Comedy     0.39759    0.07622   5.216 2.01e-07 ***
CategoryBlog                 -0.21831    0.06273  -3.480 0.000512 ***
CategoryBlog,Comedy          -0.13598    0.09120  -1.491 0.136110
CategoryBlog,Entertainment    0.43821    0.14538   3.014 0.002608 **
CategoryBlog,Science         -0.63493    0.16621  -3.820 0.000137 ***
CategoryComedy                0.95713    0.10866   8.808  < 2e-16 ***
CategoryComedy,Entertainment  0.83664    0.07187  11.641  < 2e-16 ***
CategoryComedy,Informative    0.79422    0.08926   8.898  < 2e-16 ***
CategoryEntertainment         0.39775    0.08487   4.686 2.96e-06 ***
CategoryEntertainment,Blog    0.32099    0.11049   2.905 0.003710 **
CategoryEntertainment,Comedy  0.90616    0.07233  12.528  < 2e-16 ***
CategoryFood                  0.29563    0.05435   5.439 5.98e-08 ***
CategoryFood,Entertainment    0.51036    0.11860   4.303 1.76e-05 ***
CategoryInformative           0.04349    0.05775   0.753 0.451476
CategoryNews                  0.20090    0.06208   3.236 0.001230 **
CategoryScience              -0.04915    0.05430  -0.905 0.365534
CategoryTech                 -0.58233    0.05878  -9.908  < 2e-16 ***
CategoryTech,Comedy          -0.38563    0.12048  -3.201 0.001391 **
CategoryTech,Informative     -1.79659    0.13119 -13.695  < 2e-16 ***
CategoryTech,News            -1.01754    0.13520  -7.526 7.74e-14 ***
CategoryVideoGames           -0.19716    0.05606  -3.517 0.000446 ***
afinn_score                  -0.06743    0.04527  -1.489 0.136523
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

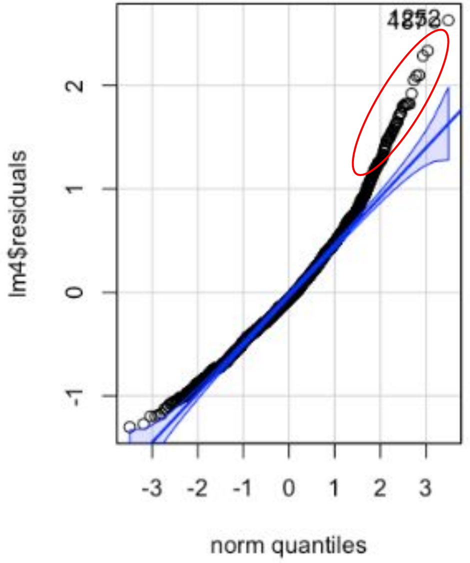
Residual standard error: 0.5293 on 2072 degrees of freedom
Multiple R-squared:  0.8153,	Adjusted R-squared:  0.813
F-statistic: 365.8 on 25 and 2072 DF,  p-value: < 2.2e-16
```

R squared: 0.8154
Adjusted R squared: 0.813

Assumption?







Variances not constant
Right skewed. Normality isn't okay too.

Cross-Validation

Randomly split the data set in a 70% training and 30% test set. Use set.seed() so that your results are reproducible

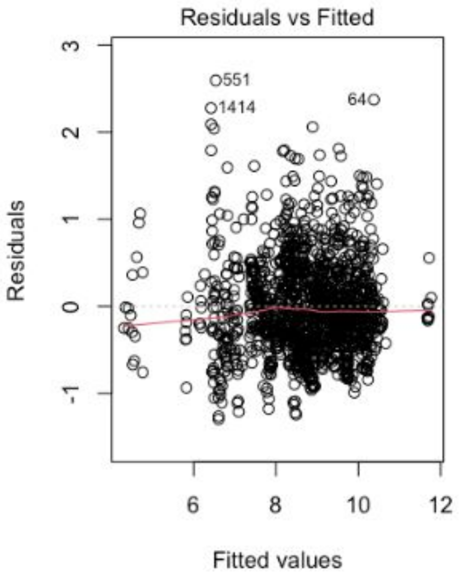
fit lm4 (the final model in MLR part) to df_train:

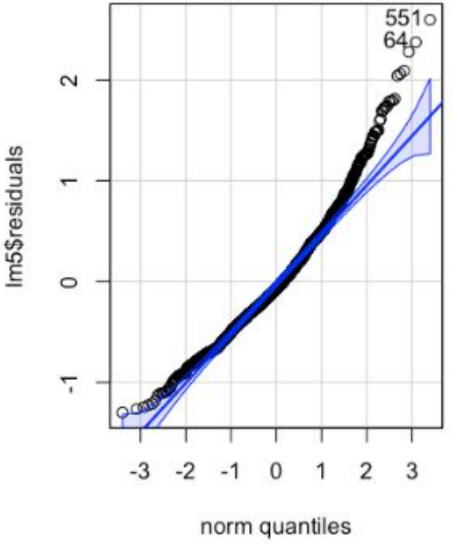
```
lm5 <- lm(log(Views) ~ CC + log(Length) + log(Subscribers) + Category + afinn_score,
             data = df_train)
```

```
# make prediction
pred1 <- predict(lm5, newdata = df_test)
pred_lm5 <- exp(pred1); length(pred_lm5)
```

```
# Compute the RMSE
lm_RMSE <- RMSE(df_test$Views, pred_lm5);
```

lm_RMSE = 7618.171
Multiple R-squared: 0.8162
Adjusted R-squared: 0.813





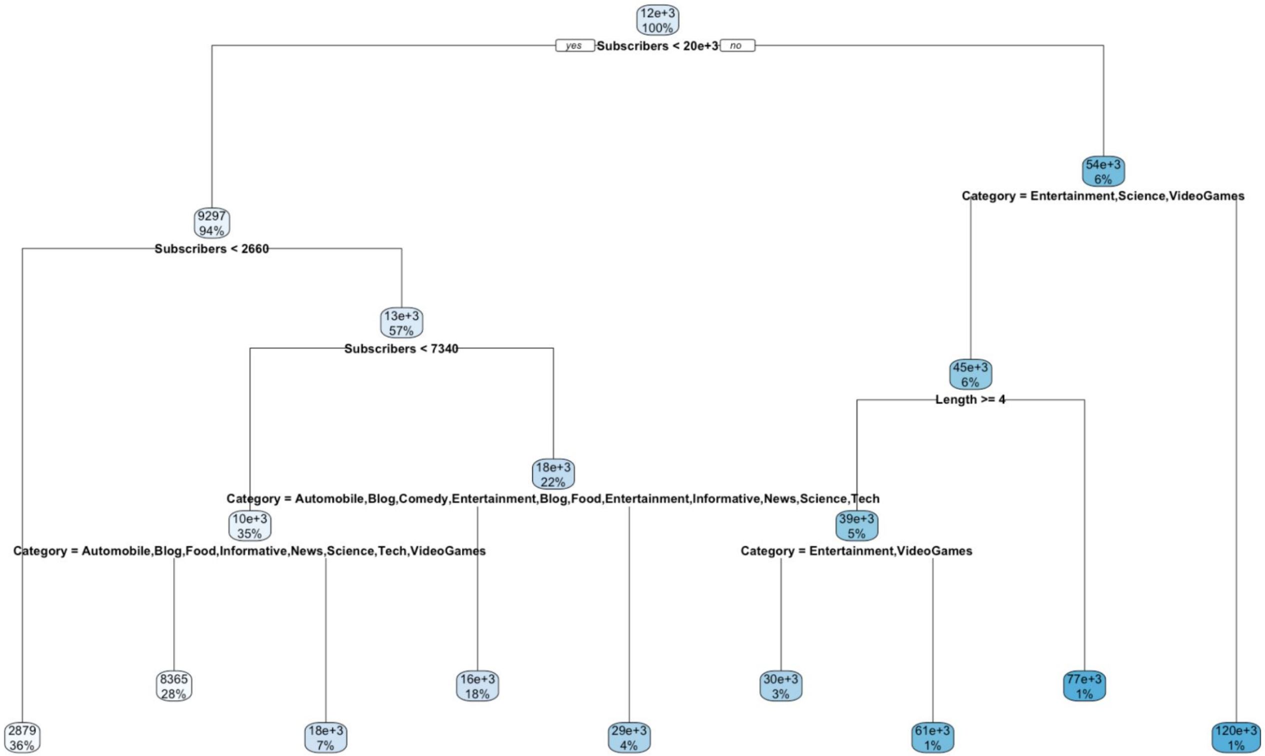
Decision Tree

```
# Fit tree model
t1 <- rpart(Views ~ CC + Released + Category + Length + Subscribers + afinn_score + afinn_title_score,
            data = df_train,
            method = "anova")
summary(t1)

# Plot the desicion tree
rpart.plot(t1)

# Plot R-square vs Splits and the Relative Error vs Splits.
rsq.rpart(t1)
```

Decision Tree

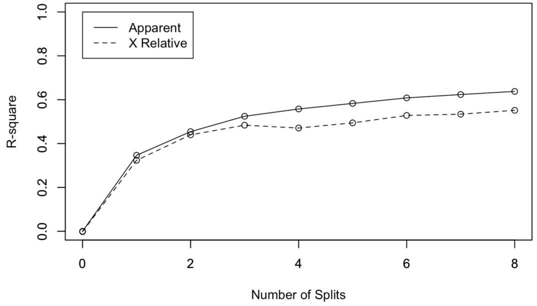



```
Call:
rpart(formula = Views ~ CC + Released + Category + Length + Subscribers +
    afinn_score, data = df_train, method = "anova")
  n= 1465

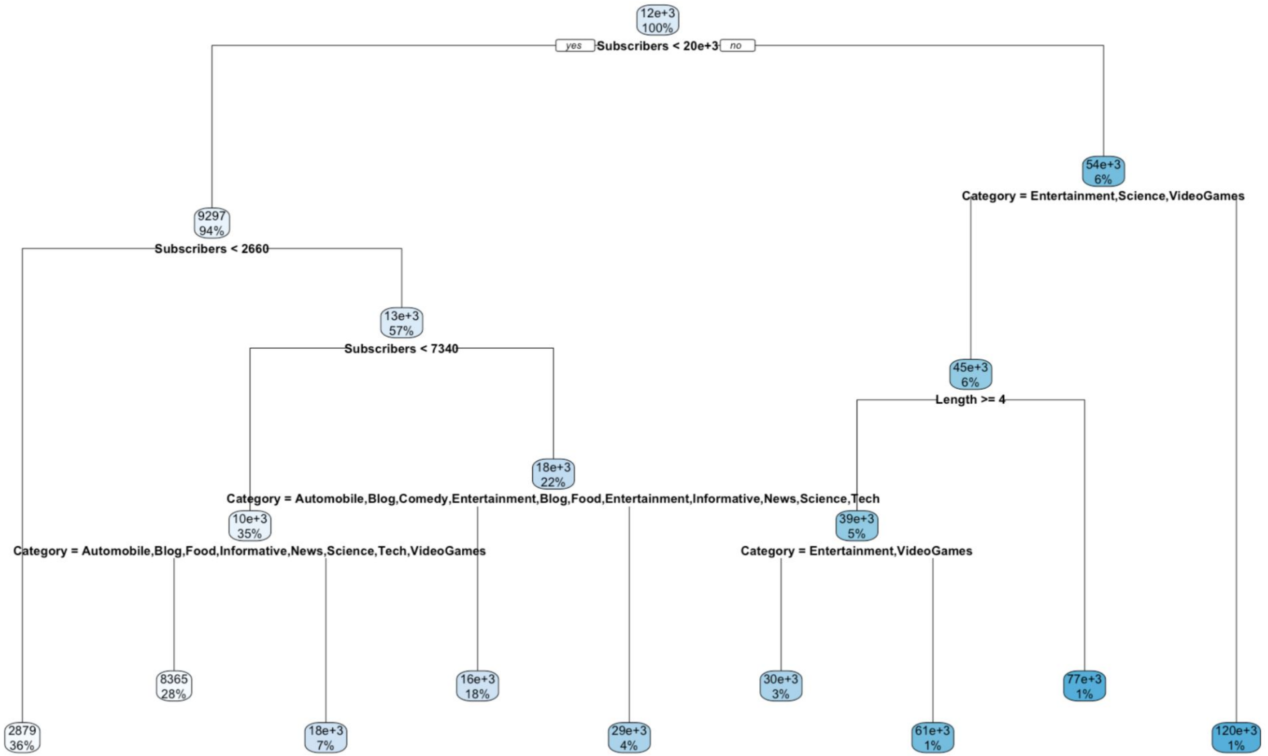
          CP nsplit rel error    xerror      xstd
1 0.34656079      0 1.0000000 1.0023252 0.2420648
2 0.10785793      1 0.6534392 0.6988232 0.1858269
3 0.07015559      2 0.5455813 0.6165450 0.1913220
4 0.03314964      3 0.4754257 0.4946788 0.1889961
5 0.02540655      4 0.4422761 0.4965904 0.1890720
6 0.02514482      5 0.4168695 0.4751621 0.1945831
7 0.01531699      6 0.3917247 0.4456764 0.1941797
8 0.01424796      7 0.3764077 0.4354656 0.1941017
9 0.01000000      8 0.3621597 0.4238468 0.1939854

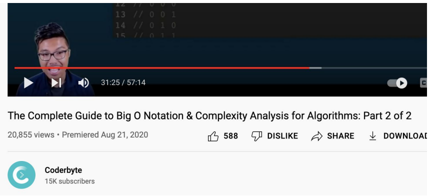
Variable importance
Subscribers    Category      Length    Released afinn_score
         64          27           4           4           1

Variables actually used in tree construction:
[1] Category    Length      Subscribers
```



Decision Tree Interpretation





Subscribers: 15K
Category: Tech
Length: 57min

Views: 20855
Predicted Views: 16000

Decision Tree Performance

````
```{r}
# Make prediction
pred_tree <- predict(t1, newdata = df_test)

# Compute the RMSE
t1_RMSE <- RMSE(df_test$Views, pred_tree); t1_RMSE

# Compute R^2
t1_R2 <- cor(df_test$Views, pred_tree)^2; t1_R2
```
````

```
[1] 7977.73
[1] 0.7801641
```

Random Forest

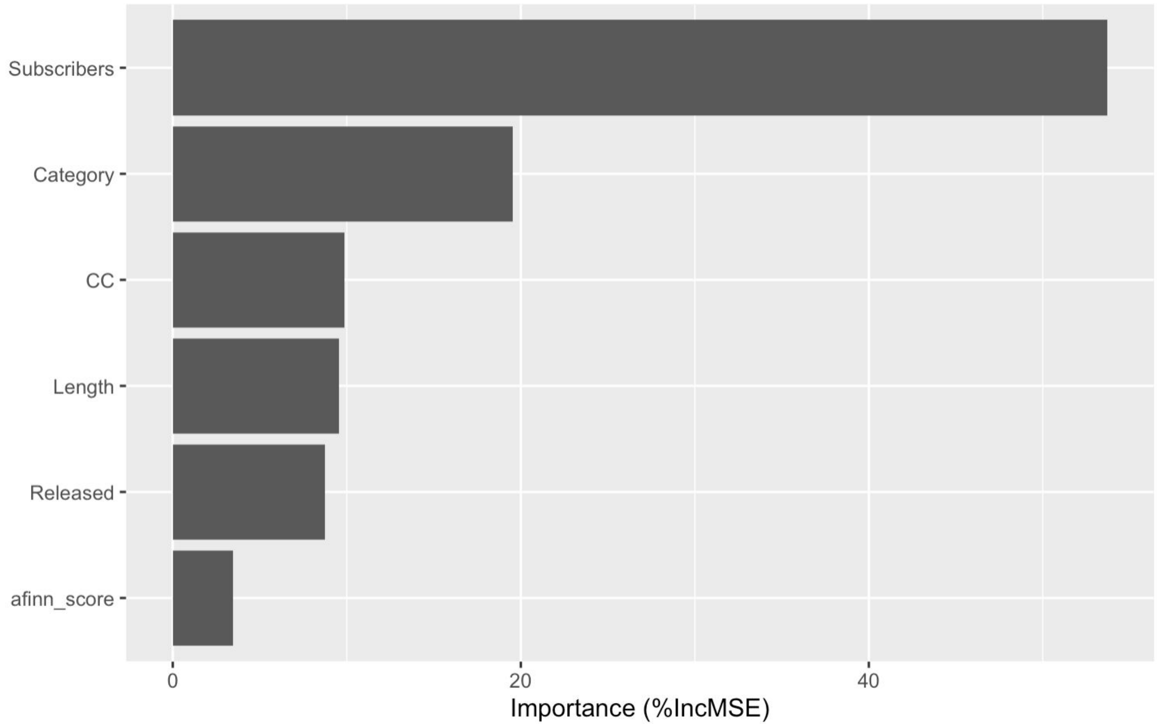
```
# Fit random forest mode using the training set
rf1 <- randomForest(Views ~ CC + Released + Category +
                       Length + Subscribers + afinn_score +
                       afinn_title_score, importance = TRUE,
                     data = df_train)
rf1

# Make a variable importance plot
vip(rf1, num_features = 14,  include_type = TRUE)
```

```
Call:
 randomForest(formula = Views ~ CC + Released + Category + Length +      Subscribers +
afinn_score + afinn_title_score, data = df_train,      importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 154686582
                    % Var explained: 55.62
```

Random Forest Interpretation



Random Forest Performance

```
```{r}
# Make prediction
pred_rf <- predict(rf1, newdata = df_test);

# Compute the RMSE
rf1_RMSE <- RMSE(df_test$Views, pred_rf); rf1_RMSE

# Compute R^2
rf1_R2 <- cor(df_test$Views, pred_rf)^2; rf1_R2
```
```

```
[1] 6660.282
[1] 0.836752
```

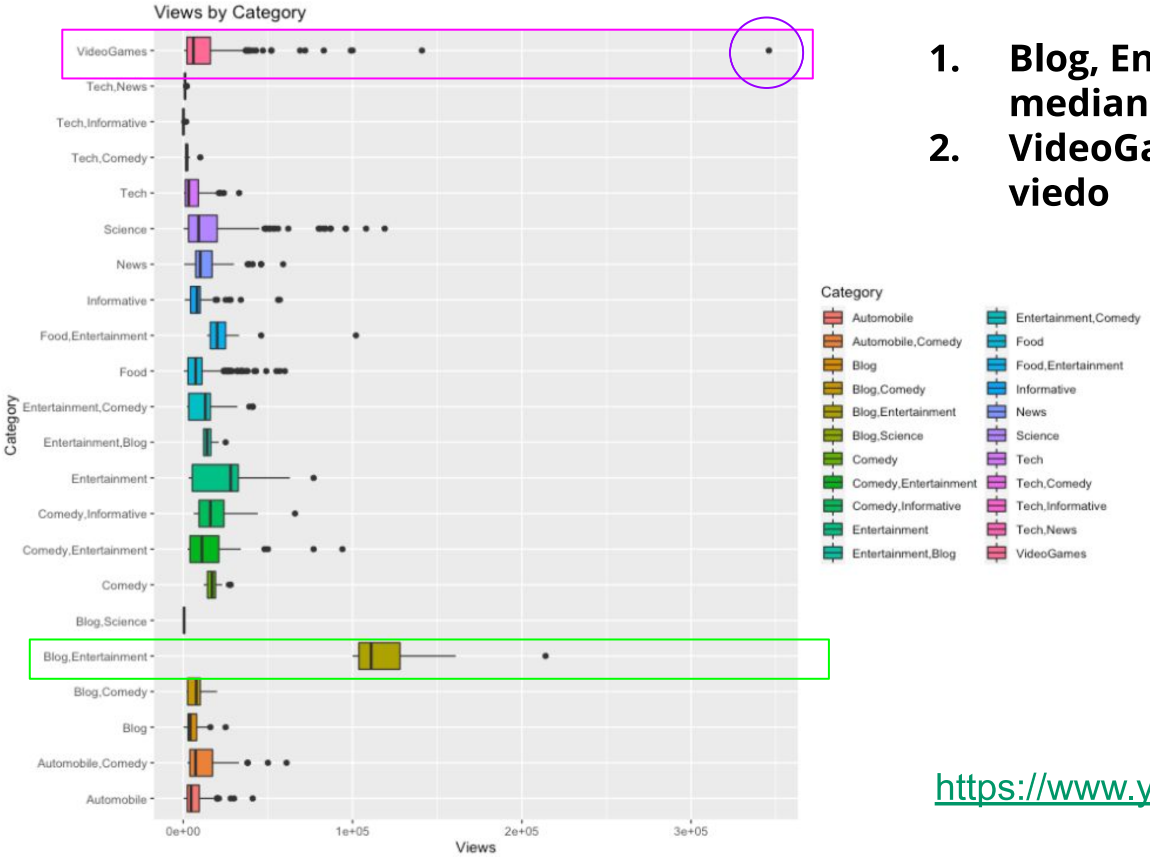
Comparing models and conclusions

Model <chr>	R_squared <dbl>	adj_R_squared <dbl>	formula <chr>
full model without transformation	0.5998046	0.5945846	lm(Views ~ CC + Released + Length + Subscribers + Category + afinn_score+ afinn_title_score, data=df1)
lm1	0.8154339	0.8130265	lm(log(Views) ~ CC + log(Released) + log(Length) + log(Subscribers) + Category + afinn_score+ afinn_title_score, data=df1)
lm2	0.8154009	0.8130834	lm(log(Views) ~ CC + log(Length) + log(Subscribers) + Category + afinn_score + afinn_title_score, data=df1)
lm3	0.8150697	0.8129286	lm(log(Views) ~ CC + log(Length) + log(Subscribers) + Category, data=df1)
lm4	0.8152674	0.8130385	lm(log(Views) ~ CC + log(Length) + log(Subscribers) + Category + afinn_score, data = df1)
lm5	0.8162162	0.8130322	lm(log(Views) ~ CC + log(Length) + log(Subscribers) + Category + afinn_score, data = df_train)

82% of the variability observed in the Views is explained by the regression model

Model <chr>	R_squared <dbl>	RMSE <dbl>
linear regression	0.8162162	7618.171
regression tree	0.7801641	7977.730
random forest	0.8367520	6660.282

Something Interesting!



1. **Blog, Entertainment has the highest median of Views**
2. **VideoGames has a extreme high Views viedo**

https://www.youtube.com/watch?v=ndsaoMFz9J4

Difficulty in research

1. Data wrangling: convert unit of string variables to number
2. Sentiment analysis: being creative and find the best way to represent the score/ value of video transcript sentiment
3. MLR: scatter plot matrix, transformation, variable selection (i.e. insignificant predictors, but high R square)

Question?

Thank You !