STAT 108: Final Project Analysis

Tim Lanthier

2/8/2022

Youtube Analysis

```
library(tidyverse)
library(knitr)
library(patchwork)
library(broom)
library(leaps)
```

```
youtube_raw <- read.csv('data/youtube_data.csv') %>%
  subset(select = -c(X)) # removed redundant X column
glimpse(youtube_raw)
```

```
## Rows: 982
## Columns: 32
## $ video_id          <chr> "05HuTGeF5AA", "SXrOuIhoslA", "hzwTq8ZZeyM", "Z6dwgW~
## $ title             <chr> "Khabib Nurmagomedov Announces Retirement | UFC 254"~
## $ publishedAt       <chr> "2020-10-24 21:27:37+00:00", "2020-09-26 15:33:12+00~
## $ channelId         <chr> "UCvgfXK4nTYKudb0rFR6noLA", "UCjwmbv6NE4mOh8Z8VhPUx1~
## $ channelTitle      <chr> "UFC - Ultimate Fighting Championship", "Rosanna Pan~
## $ categoryId        <int> 17, 26, 20, 22, 24, 20, 27, 17, 10, 26, 17, 24, 20, ~
## $ trending_date     <chr> "2020-10-27 00:00:00+00:00", "2020-09-30 00:00:00+00~
## $ tags              <chr> "khabib|retires|nurmagomedov|retirement|annouces|ufc~
## $ view_count        <int> 17992021, 710333, 1647002, 812308, 3662591, 801205, ~
## $ likes             <int> 461029, 36136, 49652, 51599, 248601, 30804, 58505, 1~
## $ dislikes          <int> 10048, 619, 1676, 503, 2797, 618, 203, 456, 1492, 20~
## $ comment_count     <int> 50333, 4093, 2179, 2235, 10062, 2418, 1717, 1250, 17~
## $ comments_disabled <chr> "False", "False", "False", "False", "False", "False"~
## $ ratings_disabled  <chr> "False", "False", "False", "False", "False", "False"~
## $ description       <chr> "After defeating Justin Gaethje and improving to 29-~
## $ num_tags          <int> 252, 468, 7, 85, 93, 432, 59, 172, 406, 7, 44, 281, ~
## $ num_caps          <int> 7, 26, 7, 7, 38, 13, 7, 7, 16, 9, 4, 17, 5, 7, 12, 6~
## $ num_exc           <int> 0, 0, 0, 1, 0, 1, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0~
## $ num_qm            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ num_period        <int> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 3, 0, 0, 0, 0, 1, 3~
## $ num_dollar        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ title_length      <int> 50, 38, 37, 43, 64, 76, 42, 39, 83, 51, 54, 95, 38, ~
## $ desc_length       <dbl> 986, 1131, 113, 1835, 64, 753, 866, 647, 648, 739, 4~
## $ weekday_published <int> 5, 5, 6, 1, 1, 1, 0, 1, 3, 0, 4, 6, 3, 1, 0, 2, 1, 0~
## $ day_published     <int> 24, 26, 4, 19, 3, 8, 30, 18, 17, 3, 12, 3, 6, 2, 2, ~
## $ hour_published    <int> 21, 15, 17, 18, 13, 6, 16, 17, 19, 22, 19, 6, 16, 18~
## $ trending_age      <int> 3, 4, 6, 4, 9, 6, 1, 5, 5, 5, 4, 5, 1, 2, 4, 4, 3, 5~
## $ video_length      <dbl> 331, 1226, 524, 902, 351, 613, 287, 920, 233, 690, 1~
```

```
## $ subscriberCount   <int> 13300000, 13200000, 1130000, 7330000, 19300000, 1550~
## $ videoCount        <int> 11340, 917, 124, 1548, 27360, 594, 648, 895, 52, 133~
## $ category          <chr> "Movies", "Horror", "Classics", "Documentary", "Fami~
## $ channel_length    <int> 36, 15, 11, 9, 10, 13, 9, 12, 7, 8, 17, 17, 15, 36, ~
```

We will need to change the types of a few of the variables.

```
youtube <- youtube_raw %>%
  mutate(trending_date = as.Date(trending_date),  # strip time since all trending dates recorded at tim
         publishedAt = as.POSIXct(publishedAt),
         weekday_published = as.factor(weekday_published))
```

We will now drop some of the columns which we will not use in our analysis.

```
youtube <- subset(youtube, select = -c(video_id, channelId, categoryId, tags, description, title, chann
```

Exploratory Data Analysis

Let's start by checking for missing values. When generating some of the features, we automatically dropped videos which no longer exist anymore.

```
colSums(is.na(youtube))
```

```
##       publishedAt     trending_date        view_count             likes
##                 0                 0                 0                 0
##          dislikes     comment_count comments_disabled  ratings_disabled
##                 0                 0                 0                 0
##          num_tags          num_caps           num_exc            num_qm
##                 0                 0                 0                 0
##        num_period        num_dollar      title_length       desc_length
##                 0                 0                 0                32
## weekday_published     day_published    hour_published      trending_age
##                 0                 0                 0                 0
##      video_length   subscriberCount        videoCount          category
##                 0                 8                 0                 0
##    channel_length
##                 0
```

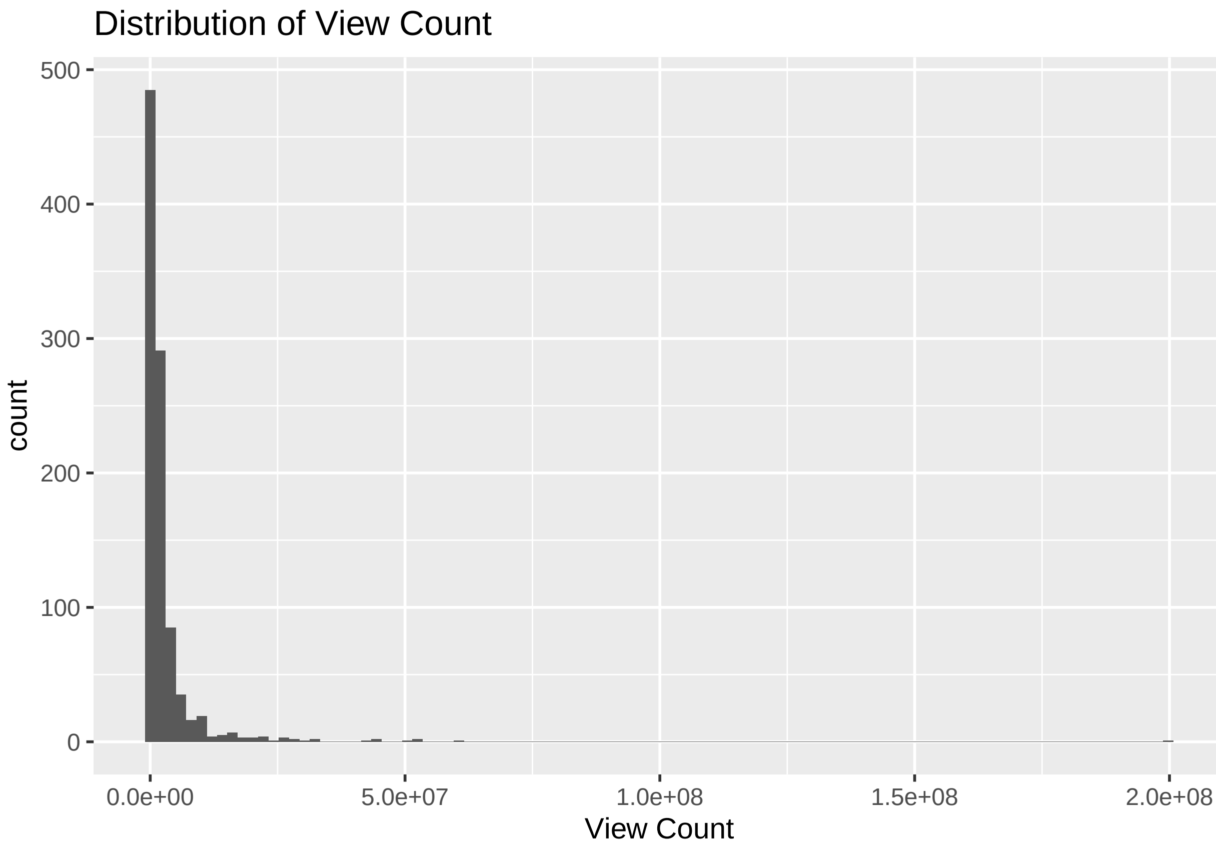
The only column with missing values is `subscriberCount` and `desc_length`. When using the Youtube API to access subscriber counts, for channels with hidden subscriber counts, we decided to input those values of `subscriberCount` as missing. Seeing as we have so few observations, we will drop these from the dataset. For those with missing description length, this is because those videos have no description. So we will replace those with `desc_length` of 0.

```
youtube$desc_length[is.na(youtube$desc_length)] <- 0
youtube <- youtube %>%
  drop_na()
```

Univariate

We will start by looking at the distributions for the response variable and each of the predictor variables.

```
ggplot(data = youtube, aes(view_count)) +
  geom_histogram(bins = 100) +
  labs(x = 'View Count', title = 'Distribution of View Count')
```

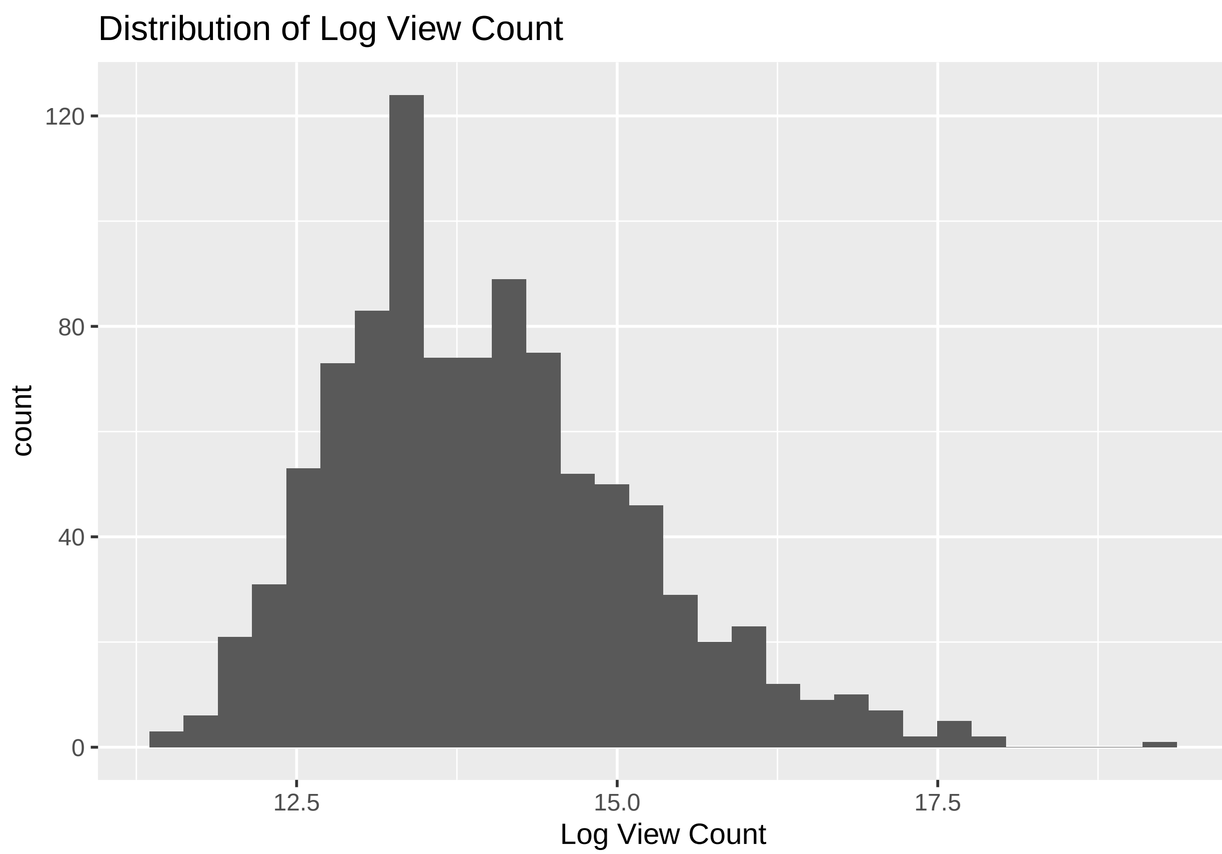



```
summarise(youtube, mean = mean(view_count),
          std_dev = sd(view_count),
          min = min(view_count),
          q1 = quantile(view_count, 0.25),
          median = median(view_count),
          q3 = quantile(view_count, 0.75),
          max = max(view_count),
          IQR = q3-q1)
```

```
##      mean std_dev   min       q1  median      q3       max     IQR
## 1 2908286 8495062 86407 504949.5 1015847 2377521 199860302 1872572
```

Looking at the histogram for view count, we see that the distribution is heavily positively skewed. We also have quite a few outliers. Since we wish to conduct inference, we will be looking at the looking at the log view count.

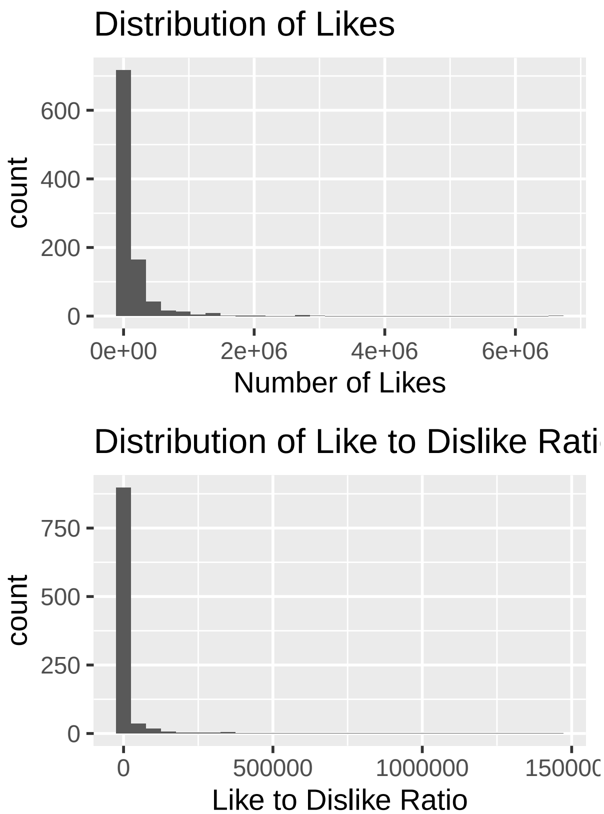
```
youtube <- youtube %>%
  mutate(log_views = log(view_count))

ggplot(data = youtube, aes(log_views))+
  geom_histogram(bins = 30) +
  labs(x = 'Log View Count', title = 'Distribution of Log View Count')
```



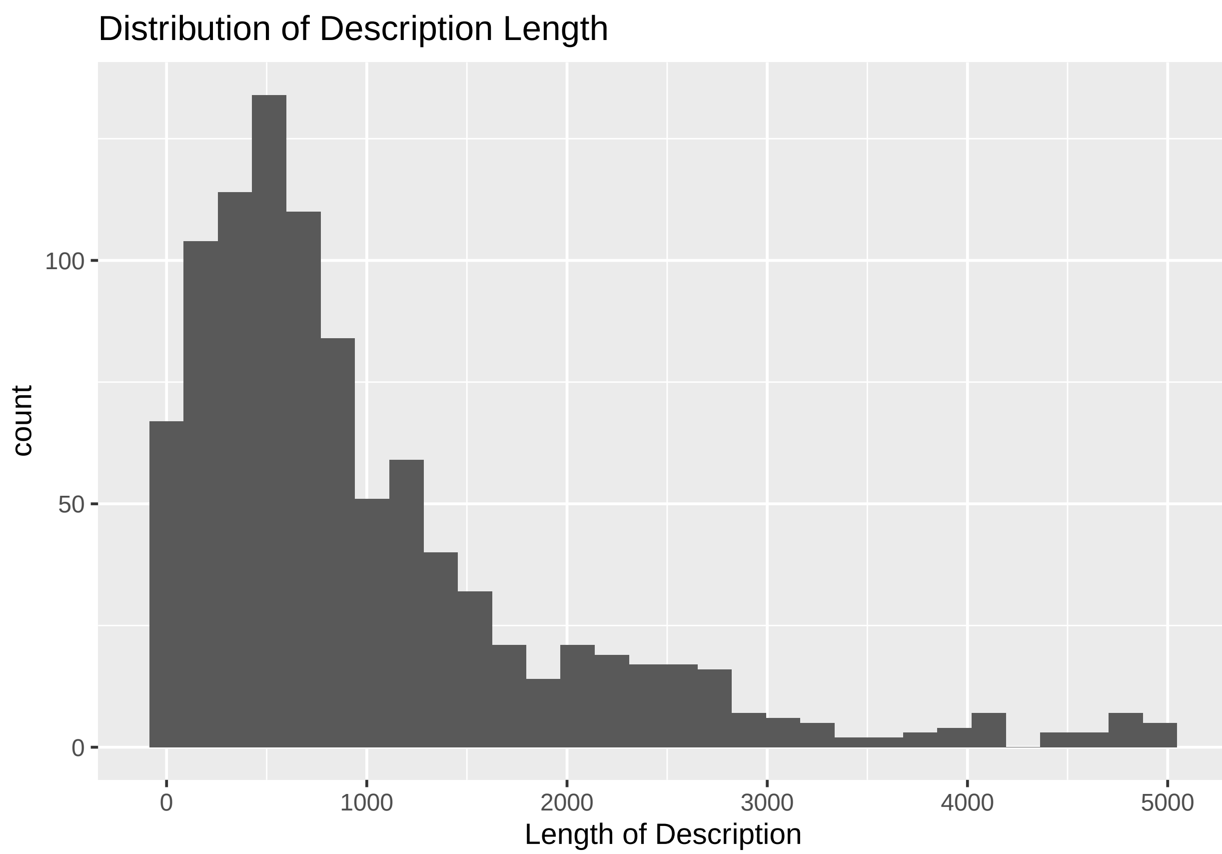
We also will investigate the like to dislike ratio. Note that to avoid division by 0, if we have a video with no dislikes, we will let the like to dislike ratio just be the number of likes.

```
youtube <- youtube %>%
  mutate(like_dislike_ratio = ifelse(dislikes == 0, likes, likes/dislikes)) # avoid division by 0
```

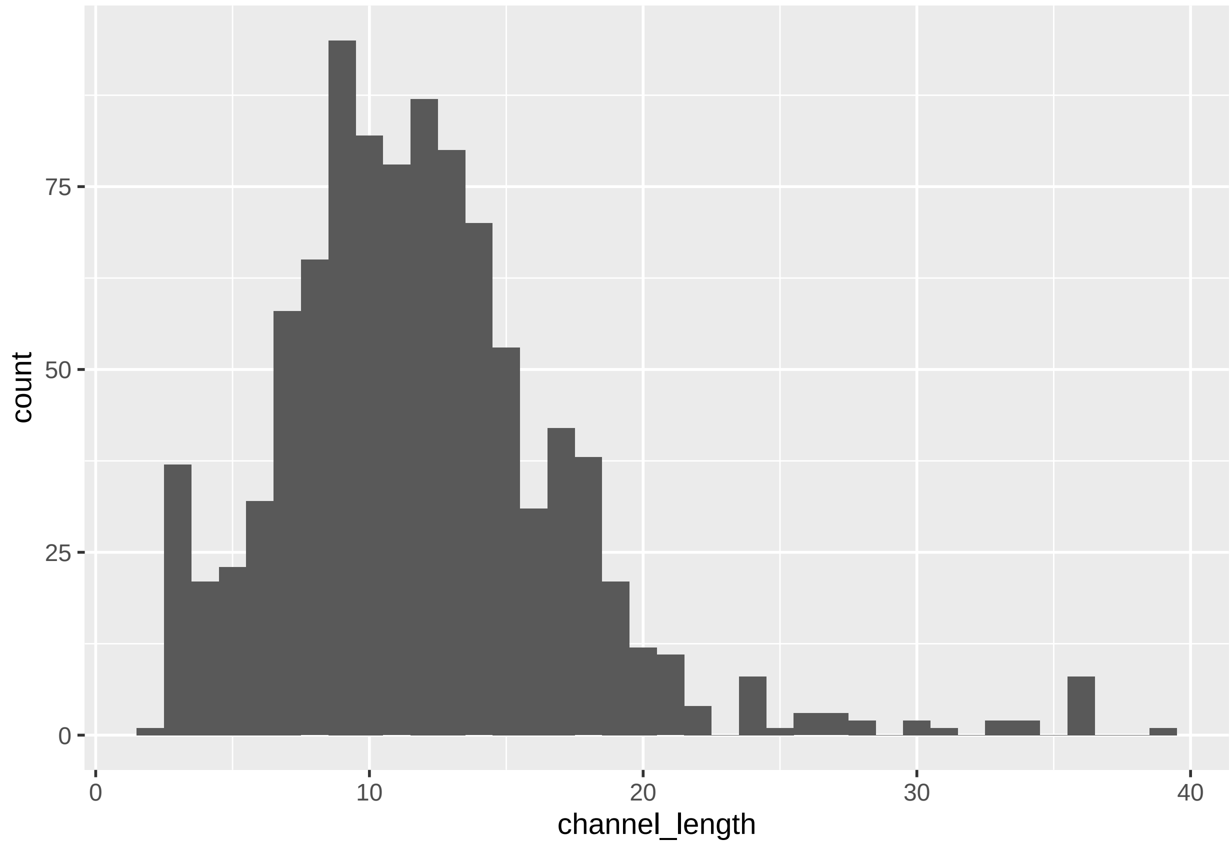
```
p1 <- ggplot(data = youtube, aes(likes))+
  geom_histogram(bins = 30) +
  labs(x = 'Number of Likes', title = 'Distribution of Likes')

p2 <- ggplot(data = youtube, aes(dislikes))+
  geom_histogram(bins = 30) +
  labs(x = 'Number of Dislikes', title = 'Distribution of Dislikes')

p3 <- ggplot(data = youtube, aes(like_dislike_ratio))+
  geom_histogram(bins = 30) +
  labs(x = 'Like to Dislike Ratio', title = 'Distribution of Like to Dislike Ratio')

p4 <- ggplot(data = youtube, aes(comment_count)) +
  geom_histogram(bins = 30) +
  labs(x = 'Comment Count', title = 'Distribution of Comment Count')

p1 + p2 + p3 + p4
```




```
ggplot(data = youtube, aes(desc_length)) +
  geom_histogram(bins = 30) +
  labs(x = 'Length of Description', title = 'Distribution of Description Length')
```

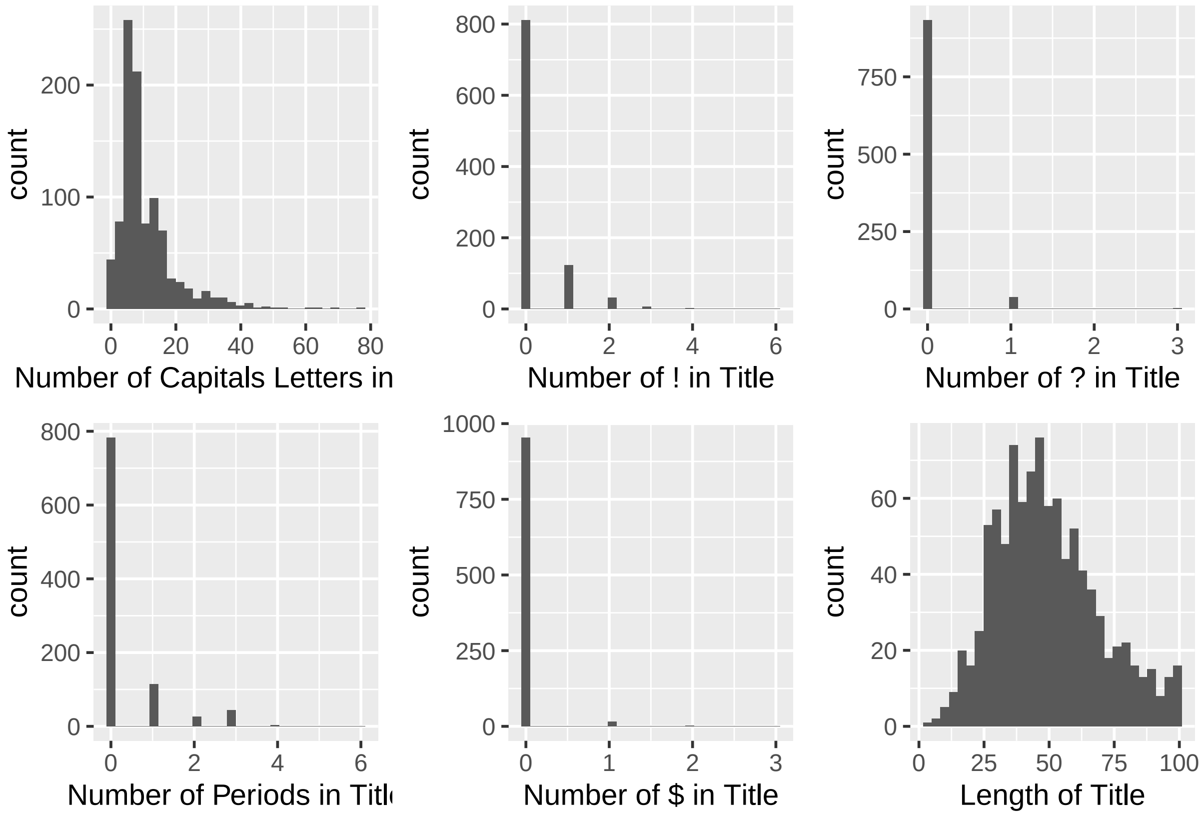



```
ggplot(data = youtube, aes(channel_length)) +
  geom_histogram(binwidth = 1)
```

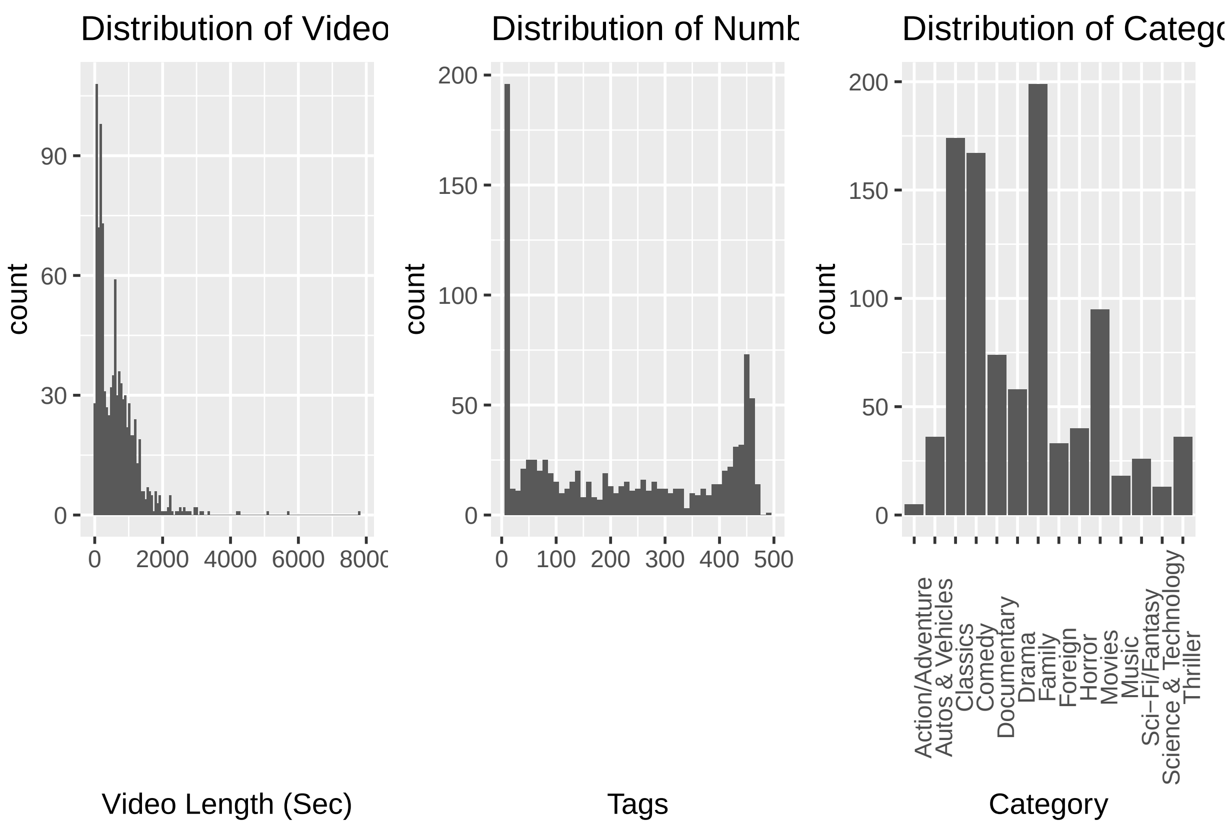


Considering that view count is positively skewed, it is no surprise that the likes, dislikes, and comment count is also positively skewed. Now we will investigate some of the predictors based on the title.

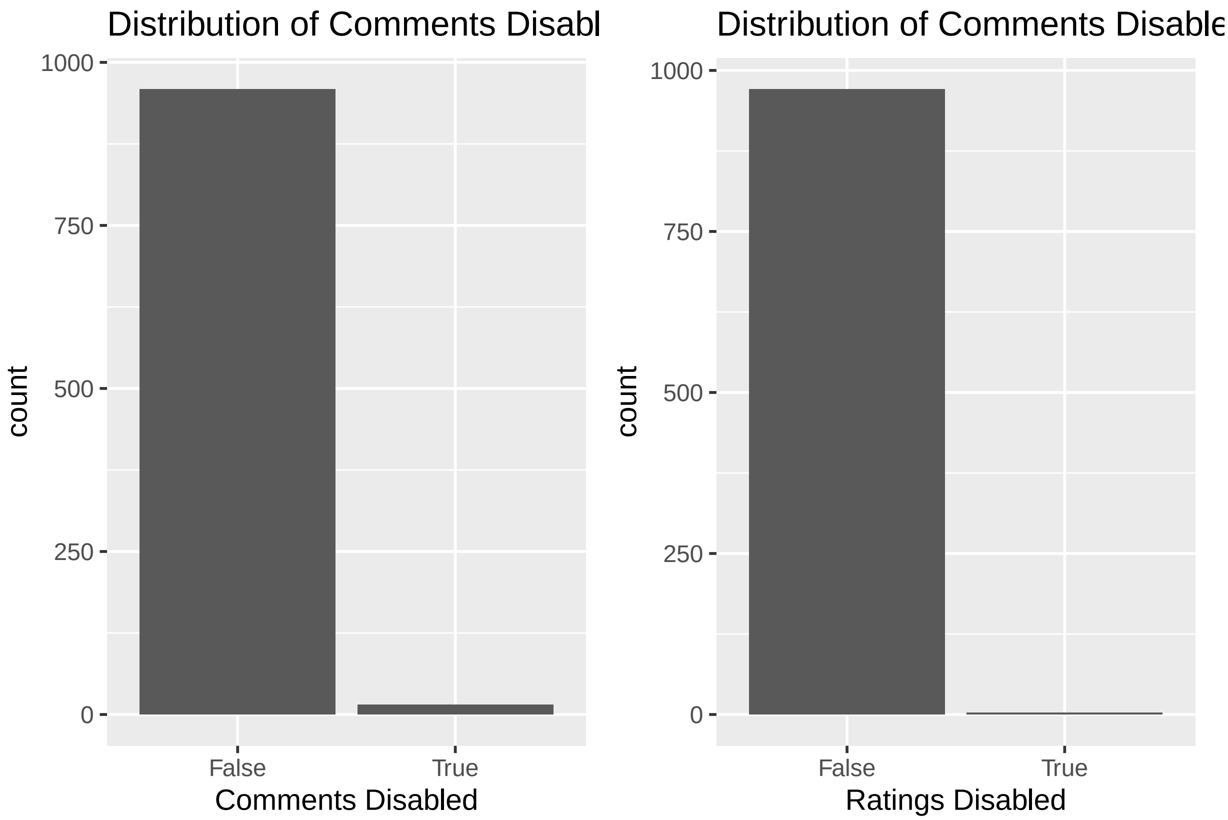
```
p5 <- ggplot(data = youtube, aes(num_caps))+
  geom_histogram(bins = 30)+
  labs(x = 'Number of Capitals Letters in Title')
p6 <- ggplot(data = youtube, aes(num_exc))+
  geom_histogram(bins = 30) +
  labs(x = 'Number of ! in Title')
p7 <- ggplot(data = youtube, aes(num_qm))+
  geom_histogram(bins = 30) +
  labs(x = 'Number of ? in Title')
p8 <- ggplot(data = youtube, aes(num_period))+
  geom_histogram(bins = 30) +
  labs(x = 'Number of Periods in Title')
p9 <- ggplot(data = youtube, aes(num_dollar))+
  geom_histogram(bins = 30) +
  labs(x = 'Number of $ in Title')
p10 <- ggplot(data = youtube, aes(title_length))+
  geom_histogram(bins = 30)+
  labs(x = 'Length of Title')
p5+p6+p7+p8+p9+p10
```



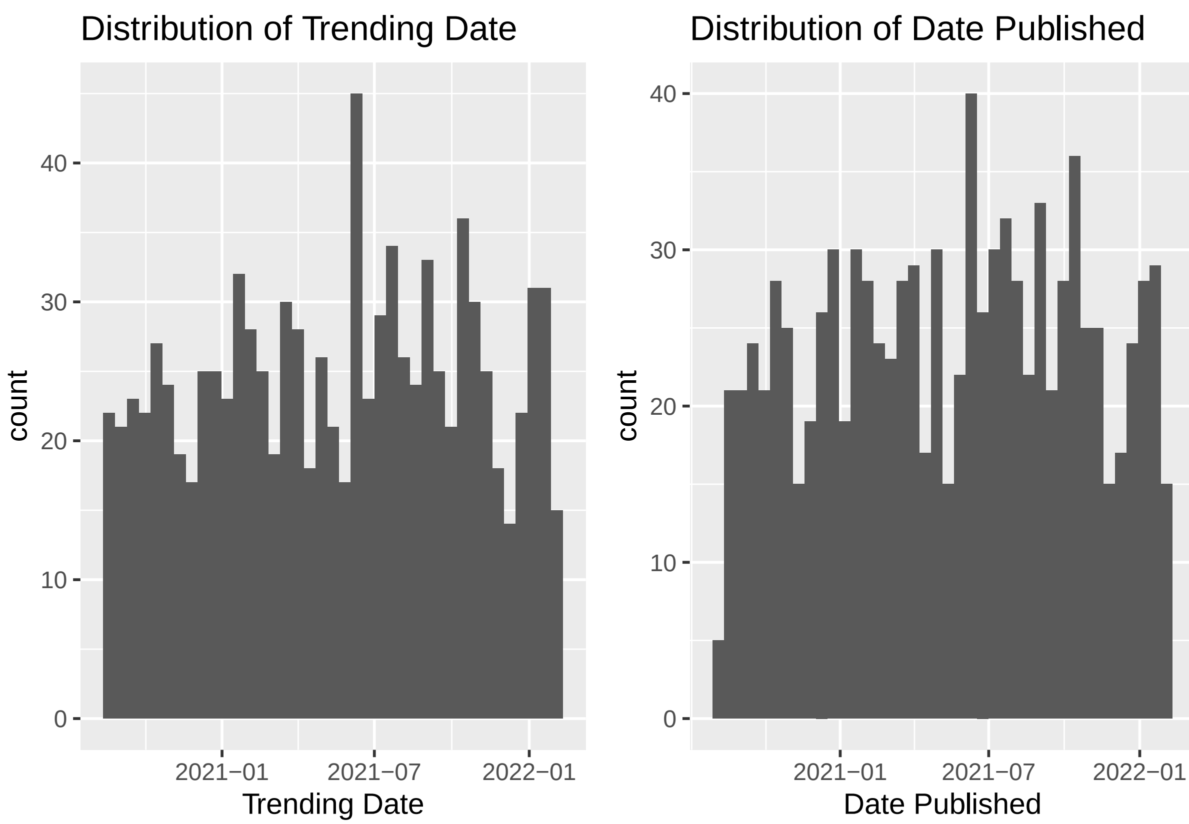

```r
p11 <- ggplot(data = youtube, aes(video_length)) +
  geom_histogram(binwidth = 60) +
  labs(x = 'Video Length (Sec)', title = 'Distribution of Video Length')

p12 <- ggplot(data = youtube, aes(num_tags))+
  geom_histogram(binwidth = 10)+
  labs(x = 'Tags', title = 'Distribution of Number of Tags')

p13 <- ggplot(data = youtube, aes(category)) +
  geom_bar() +
  labs(x = 'Category', title = 'Distribution of Category') +
  theme(axis.text.x = element_text(angle = 90))

p11 + p12 + p13
```




```
p14 <- ggplot(data = youtube, aes(comments_disabled))+
  geom_bar()+
  labs(x = 'Comments Disabled', title = 'Distribution of Comments Disabled')

p15 <- ggplot(data = youtube, aes(ratings_disabled))+
  geom_bar()+
  labs(x = 'Ratings Disabled', title = 'Distribution of Comments Disabled')

p14 + p15
```




```
p16 <- ggplot(data = youtube, aes(trending_date))+
  geom_histogram(binwidth = 14) +
  labs(title = 'Distribution of Trending Date', x = 'Trending Date')

p17<- ggplot(data = youtube, aes(publishedAt))+
  geom_histogram(binwidth = 14*60*60*24)+ # 2 week bin width
  labs(x = 'Date Published', title = 'Distribution of Date Published')

p16 + p17
```




```
levels(youtube$weekday_published) = c('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday
```

```
p18 <- ggplot(data = youtube, aes(weekday_published))+
  geom_bar() +
  labs(x = 'Weekday Published', title = 'Distribution of Weekday Published')

p19 <-ggplot(data = youtube, aes(day_published)) +
  geom_histogram(binwidth = 1) +
  labs(x = 'Day Published', title = 'Distribution of Day Published')

p20 <- ggplot(data = youtube, aes(hour_published))+
  geom_bar(binwidth = 1) +
  labs(x = 'Hour Published', title = 'Distribution of Hour Published')
```

```
## Warning: Ignoring unknown parameters: binwidth
```

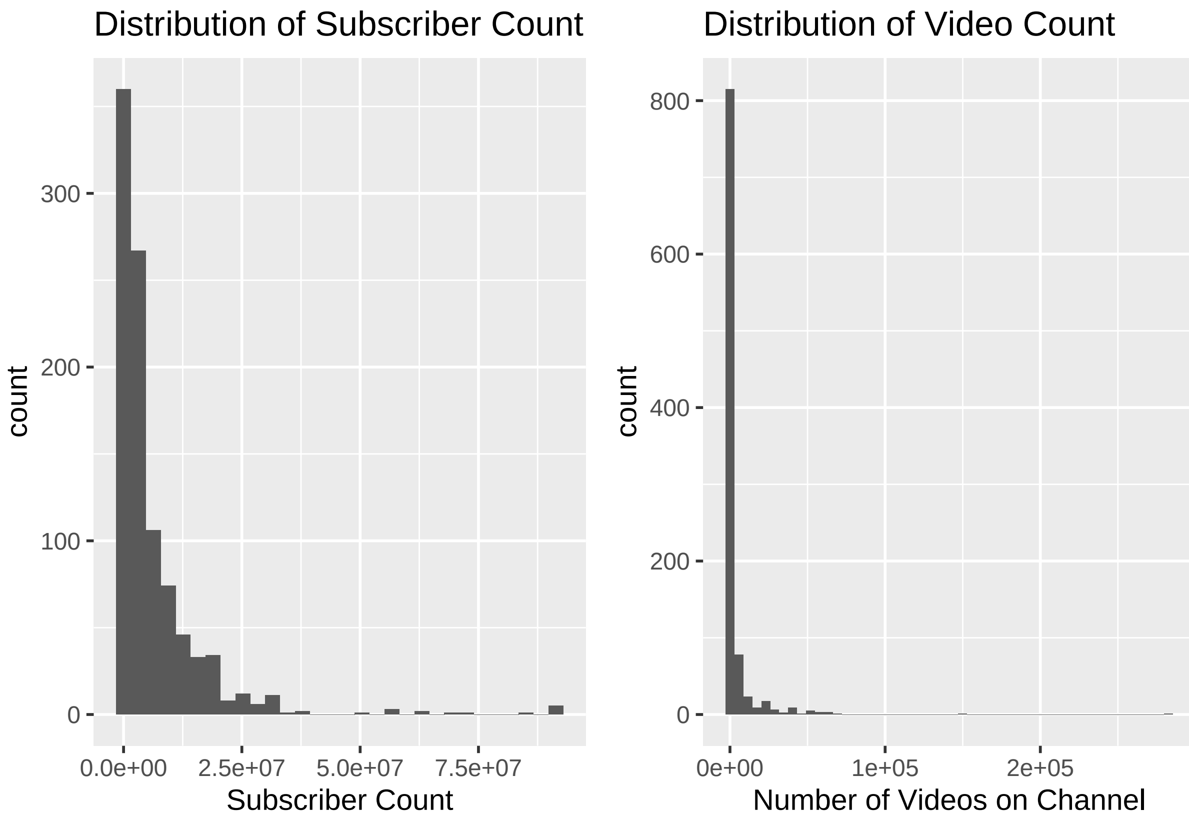
```
p21 <- ggplot(data = youtube, aes(trending_age))+
  geom_bar(binwidth = 1) +
  labs(x = 'TrendingAge', title = 'Distribution of Trending Age')
```

```
## Warning: Ignoring unknown parameters: binwidth
```

```
p18 + p19 + p20 + p21
```

Note that according to Youtube API, time and dates are in UTC time zone.

```
p22 <- ggplot(data = youtube, aes(subscriberCount))+
  geom_histogram(bins = 30) +
  labs(x = 'Subscriber Count', title = 'Distribution of Subscriber Count')
p23 <- ggplot(data = youtube, aes(videoCount)) +
  geom_histogram(bins = 50) +
  labs(x = 'Number of Videos on Channel', title = 'Distribution of Video Count')

p22 + p23
```



Bivariate

Now we will explore the relationship between our predictor variables and our response variable. As we discussed earlier, we will be using `log_views` instead of `view_count` as our response since we wish to conduct inference. We will start by observing the correlations for some of the numerical data.

```
library(Hmisc)
```

```
## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
youtube_num <- youtube %>%
  subset(select = -c(publishedAt, trending_date, comments_disabled, ratings_disabled, weekday_published

cor(youtube_num)
```

```
##                      view_count        likes      dislikes comment_count
## view_count          1.0000000000  0.872865928  0.8788954385   0.307851384
## likes               0.8728659278  1.000000000  0.8338809379   0.521898270
## dislikes            0.8788954385  0.833880938  1.0000000000   0.299306683
## comment_count       0.3078513845  0.521898270  0.2993066828   1.000000000
## num_tags           -0.0458425316 -0.069974898 -0.0374015958  -0.014041795
## num_caps           -0.0508995632 -0.059118998 -0.0425567443   0.017747779
## num_exc            -0.0499645567 -0.063765402 -0.0437459943  -0.044233308
## num_qm             -0.0118469424 -0.007696133 -0.0087973666  -0.033779080
## num_period         -0.0393117102 -0.024384378 -0.0090681248  -0.007896145
## num_dollar          0.0234788072  0.037913686  0.0002469487   0.019937566
## title_length       -0.0692386946 -0.111962718 -0.0558722700  -0.062303957
## desc_length        -0.0005891424  0.011044968  0.0008517721   0.044398062
## day_published      -0.0182071450  0.001700713  0.0020587045   0.012126508
## hour_published     -0.0768285716 -0.062706669 -0.1214434861  -0.022231024
## trending_age        0.5009803222  0.434955363  0.4611275000   0.131391831
## video_length       -0.0817083689 -0.083095989 -0.0506311551  -0.013075584
## subscriberCount     0.2614496422  0.403858831  0.1920597452   0.446491764
## videoCount         -0.0285374923 -0.057484812 -0.0262261289  -0.032059520
## channel_length     -0.0170774584 -0.028720798 -0.0108705645  -0.029331897
## log_views           0.5833814071  0.617382072  0.4748307703   0.398837427
## like_dislike_ratio  0.0945812291  0.159037116 -0.0448482614  -0.016000982
##                        num_tags     num_caps       num_exc       num_qm
## view_count         -0.045842532 -0.050899563 -0.0499645567 -0.011846942
## likes              -0.069974898 -0.059118998 -0.0637654020 -0.007696133
## dislikes           -0.037401596 -0.042556744 -0.0437459943 -0.008797367
## comment_count      -0.014041795  0.017747779 -0.0442333083 -0.033779080
## num_tags            1.000000000  0.149769522  0.0020065826  0.001655671
## num_caps            0.149769522  1.000000000  0.2508693253  0.043895999
## num_exc             0.002006583  0.250869325  1.0000000000  0.130361826
## num_qm              0.001655671  0.043895999  0.1303618263  1.000000000
## num_period          0.087201705 -0.031230752 -0.0881118263 -0.064900664
## num_dollar          0.000443382 -0.051868814  0.0502914775 -0.025749555
## title_length        0.232756665  0.404452660  0.0620674627  0.037724679
## desc_length         0.258649333  0.083722031  0.0007045872  0.042820881
## day_published       0.026638228 -0.030288479 -0.0186109827 -0.058146980
## hour_published     -0.014474825 -0.118627435  0.0481512167  0.044725001
## trending_age       -0.016961327 -0.039431297 -0.0126081210  0.010807121
## video_length        0.052218126  0.111571219  0.0794666555  0.048964323
## subscriberCount    -0.007911243 -0.027756366 -0.0378116840 -0.050221419
## videoCount         -0.012278348  0.050452542 -0.0611177922 -0.036238454
## channel_length      0.022110372  0.005148084  0.0058393639  0.043868248
## log_views          -0.004778297 -0.092851993 -0.1388607675 -0.037941641
## like_dislike_ratio -0.057429053 -0.015500992 -0.0136695510 -0.015729864
##                      num_period    num_dollar title_length   desc_length
## view_count         -0.039311710  0.0234788072 -0.069238695 -0.0005891424
## likes              -0.024384378  0.0379136863 -0.111962718  0.0110449684
## dislikes           -0.009068125  0.0002469487 -0.055872270  0.0008517721
## comment_count      -0.007896145  0.0199375665 -0.062303957  0.0443980625
## num_tags            0.087201705  0.0004433820  0.232756665  0.2586493329
## num_caps           -0.031230752 -0.0518688139  0.404452660  0.0837220306
## num_exc            -0.088111826  0.0502914775  0.062067463  0.0007045872
## num_qm             -0.064900664 -0.0257495555  0.037724679  0.0428208810
## num_period          1.000000000  0.0239839759  0.121111215 -0.0097051057
```

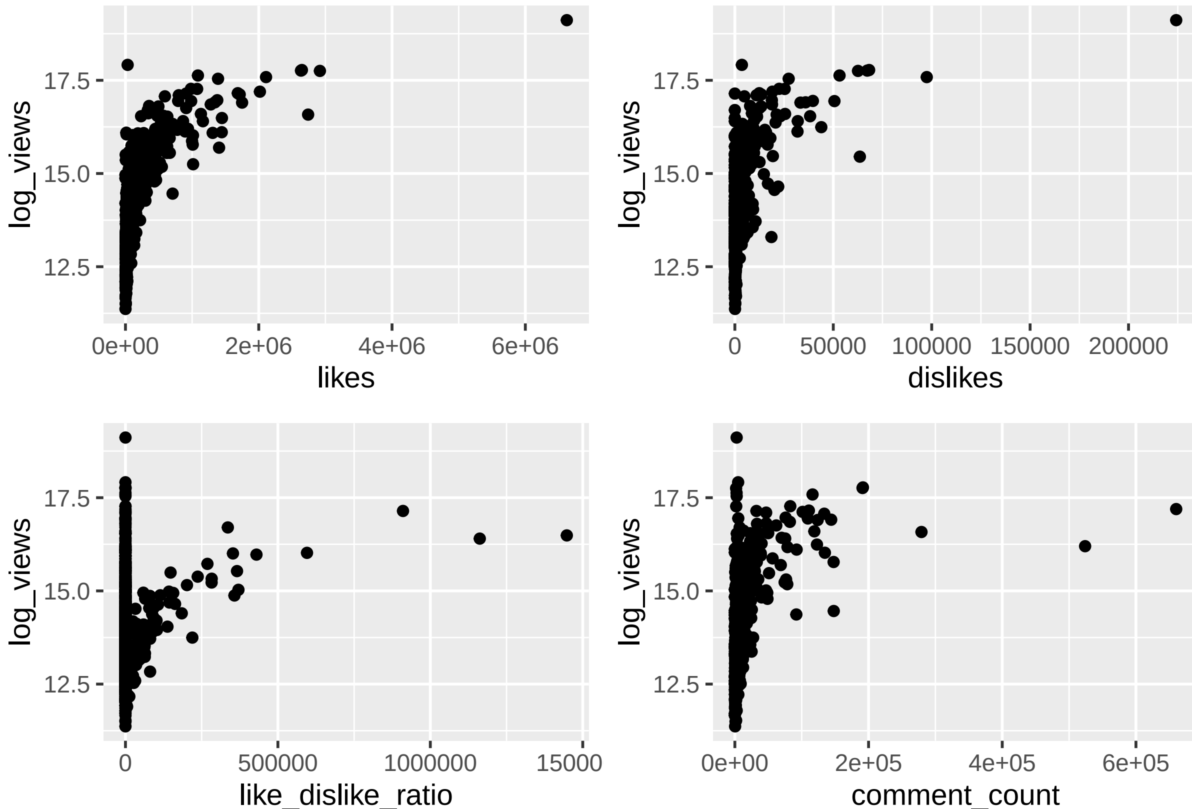
```
## num_dollar          0.023983976  1.0000000000 -0.023348075  0.0056257035
## title_length        0.121111215 -0.0233480752  1.000000000  0.1759318524
## desc_length        -0.009705106  0.0056257035  0.175931852  1.0000000000
## day_published       0.005129004  0.0224966857 -0.015409999  0.0430072239
## hour_published      0.023669898  0.0033922089 -0.130745538 -0.0346772955
## trending_age       -0.011197392  0.0205332179  0.006316315 -0.0302849606
## video_length        0.103315069  0.0251557982  0.031904721  0.1331435863
## subscriberCount    -0.049774735  0.0918141525 -0.099107456  0.0458835698
## videoCount          0.039557508 -0.0134512393  0.188035417 -0.0602952626
## channel_length     -0.102395591 -0.0159156692  0.017167599  0.0785635596
## log_views          -0.014487082  0.0440810089 -0.089705236  0.0706481913
## like_dislike_ratio -0.044342524 -0.0167616767 -0.043822655 -0.0403716055
##                    day_published hour_published trending_age video_length
## view_count          -0.018207145   -0.076828572  0.500980322  -0.08170837
## likes                0.001700713   -0.062706669  0.434955363  -0.08309599
## dislikes             0.002058705   -0.121443486  0.461127500  -0.05063116
## comment_count        0.012126508   -0.022231024  0.131391831  -0.01307558
## num_tags             0.026638228   -0.014474825 -0.016961327   0.05221813
## num_caps            -0.030288479   -0.118627435 -0.039431297   0.11157122
## num_exc             -0.018610983    0.048151217 -0.012608121   0.07946666
## num_qm              -0.058146980    0.044725001  0.010807121   0.04896432
## num_period           0.005129004    0.023669898 -0.011197392   0.10331507
## num_dollar           0.022496686    0.003392209  0.020533218   0.02515580
## title_length        -0.015409999   -0.130745538  0.006316315   0.03190472
## desc_length          0.043007224   -0.034677295 -0.030284961   0.13314359
## day_published        1.000000000    0.019038332 -0.027331760   0.04250078
## hour_published       0.019038332    1.000000000  0.076503251   0.03063420
## trending_age        -0.027331760    0.076503251  1.000000000  -0.01538800
## video_length         0.042500777    0.030634202 -0.015387996   1.00000000
## subscriberCount      0.056126296    0.025852179  0.067077340   0.04082792
## videoCount           0.031557289   -0.109263272 -0.067552387  -0.04122981
## channel_length       0.050576343    0.015543291  0.017929979  -0.07240164
## log_views            0.048693925   -0.024849701  0.376875913  -0.07385917
## like_dislike_ratio   0.037911636    0.044948015  0.027467910  -0.07390559
##                    subscriberCount  videoCount channel_length    log_views
## view_count             0.261449642 -0.02853749   -0.017077458  0.583381407
## likes                  0.403858831 -0.05748481   -0.028720798  0.617382072
## dislikes               0.192059745 -0.02622613   -0.010870564  0.474830770
## comment_count          0.446491764 -0.03205952   -0.029331897  0.398837427
## num_tags              -0.007911243 -0.01227835    0.022110372 -0.004778297
## num_caps              -0.027756366  0.05045254    0.005148084 -0.092851993
## num_exc               -0.037811684 -0.06111779    0.005839364 -0.138860768
## num_qm                -0.050221419 -0.03623845    0.043868248 -0.037941641
## num_period            -0.049774735  0.03955751   -0.102395591 -0.014487082
## num_dollar             0.091814152 -0.01345124   -0.015915669  0.044081009
## title_length          -0.099107456  0.18803542    0.017167599 -0.089705236
## desc_length            0.045883570 -0.06029526    0.078563560  0.070648191
## day_published          0.056126296  0.03155729    0.050576343  0.048693925
## hour_published         0.025852179 -0.10926327    0.015543291 -0.024849701
## trending_age           0.067077340 -0.06755239    0.017929979  0.376875913
## video_length           0.040827919 -0.04122981   -0.072401637 -0.073859175
## subscriberCount        1.000000000  0.07090522   -0.066478908  0.462068940
## videoCount             0.070905224  1.00000000   -0.138082167 -0.029270485
## channel_length        -0.066478908 -0.13808217    1.000000000 -0.012408926
```

```
## log_views              0.462068940 -0.02927049   -0.012408926  1.000000000
## like_dislike_ratio     0.006709642 -0.03389018    0.010800693  0.180504961
##                    like_dislike_ratio
## view_count                0.094581229
## likes                     0.159037116
## dislikes                 -0.044848261
## comment_count            -0.016000982
## num_tags                 -0.057429053
## num_caps                 -0.015500992
## num_exc                  -0.013669551
## num_qm                   -0.015729864
## num_period               -0.044342524
## num_dollar               -0.016761677
## title_length             -0.043822655
## desc_length              -0.040371606
## day_published             0.037911636
## hour_published            0.044948015
## trending_age              0.027467910
## video_length             -0.073905586
## subscriberCount           0.006709642
## videoCount               -0.033890177
## channel_length            0.010800693
## log_views                 0.180504961
## like_dislike_ratio        1.000000000
```

Looking at the correlations, we unsurprisingly see some moderate correlation between `log_views` and `likes`, `dislikes`, and `comment_count`. This is unsurprising as naturally videos which have more viewers will have more people rating and commenting on the video. But much like with view count, the likes, dislikes, and comment count on a video will not be observed until after it has been uploading. Since we want to focus on predictor variables that are known at or before the video is uploaded, we will not use these.

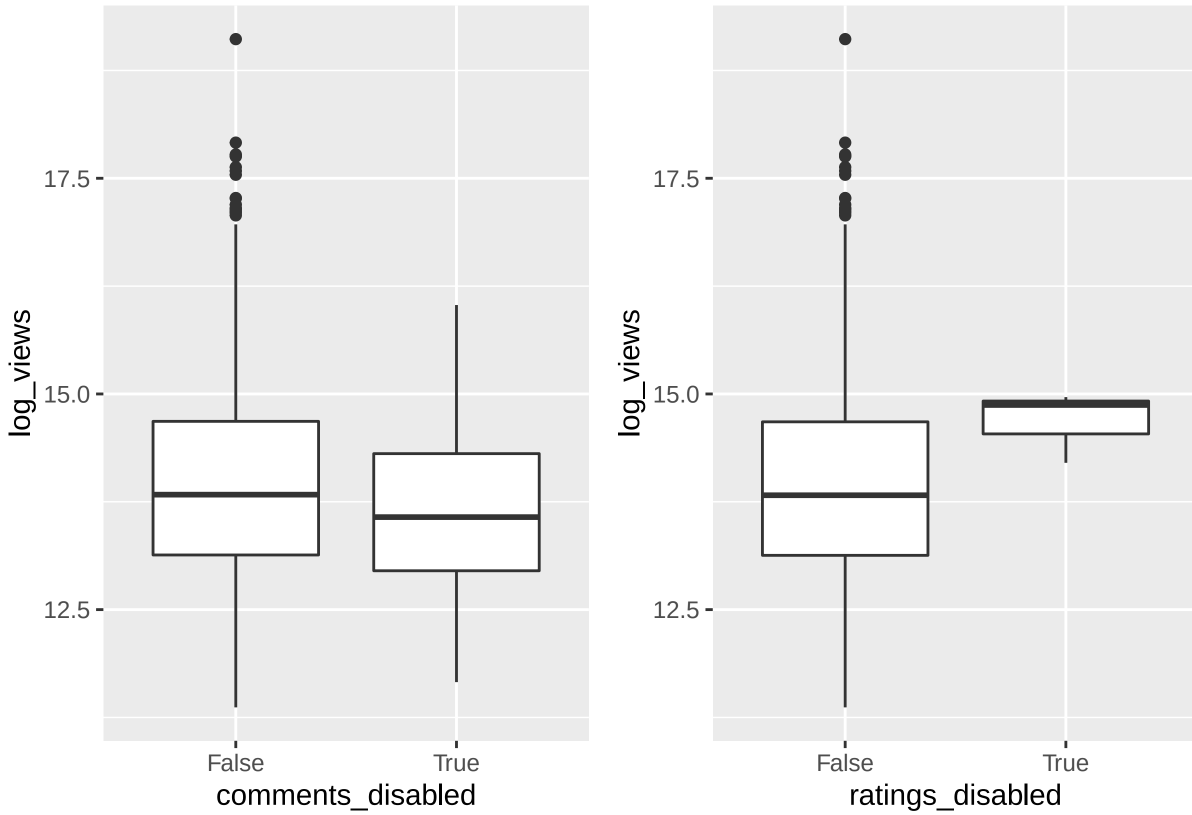
Of the variables which are useful in this sense, we see the channel's subscriber count has a moderate postive correlation of 0.484 with `log_views`. We also see that the number of exclamation points has a weak negative correlation as well as the length of the description.

```r
b1 <- ggplot(data = youtube, aes(x = likes, y = log_views)) +
  geom_point()

b2 <- ggplot(data = youtube, aes(x = dislikes, y = log_views)) +
  geom_point()

b3 <- ggplot(data = youtube, aes(x = like_dislike_ratio, y = log_views)) +
  geom_point()

b4 <- ggplot(data = youtube, aes(x = comment_count, y = log_views)) +
  geom_point()

b1+b2+b3+b4
```



We might expect a linear relationship between likes/dislikes and the number of views. Since we took the log of views, then if we were to include likes and dislikes in our model, we might want to also take the log of likes and dislikes so we can preserved this relationship. This will likely be true for other variables as well such as subscriber count.

```
b5 <- ggplot(data = youtube, aes(x = comments_disabled, y = log_views)) +
  geom_boxplot()

b6 <- ggplot(data = youtube, aes(x = ratings_disabled, y = log_views)) +
  geom_boxplot()

b5 + b6
```



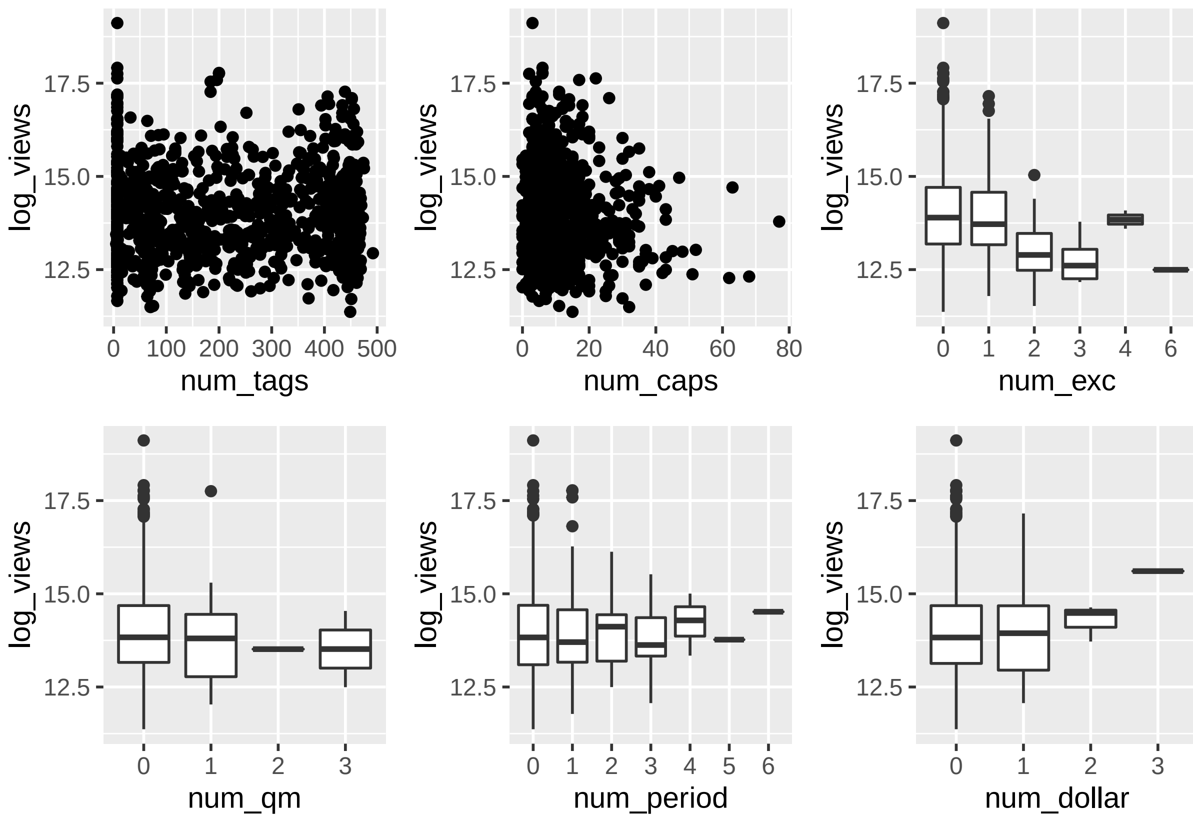
It looks like disabling ratings might have an effect on `log_views` shown by the different means across videos with disabled and enabled ratings. There is also a small difference in means for videos with comments disabled.

For the sake of plotting some of these variables, we will convert the following variables to factors.

```
youtube <- youtube %>%
  mutate(num_exc = as.factor(num_exc),
         num_qm = as.factor(num_qm),
         num_period = as.factor(num_period),
         num_dollar = as.factor(num_dollar),
         hour_published = as.factor(hour_published), # convert back after eda
         trending_age = as.factor(trending_age),
         day_published = as.factor(day_published))
```

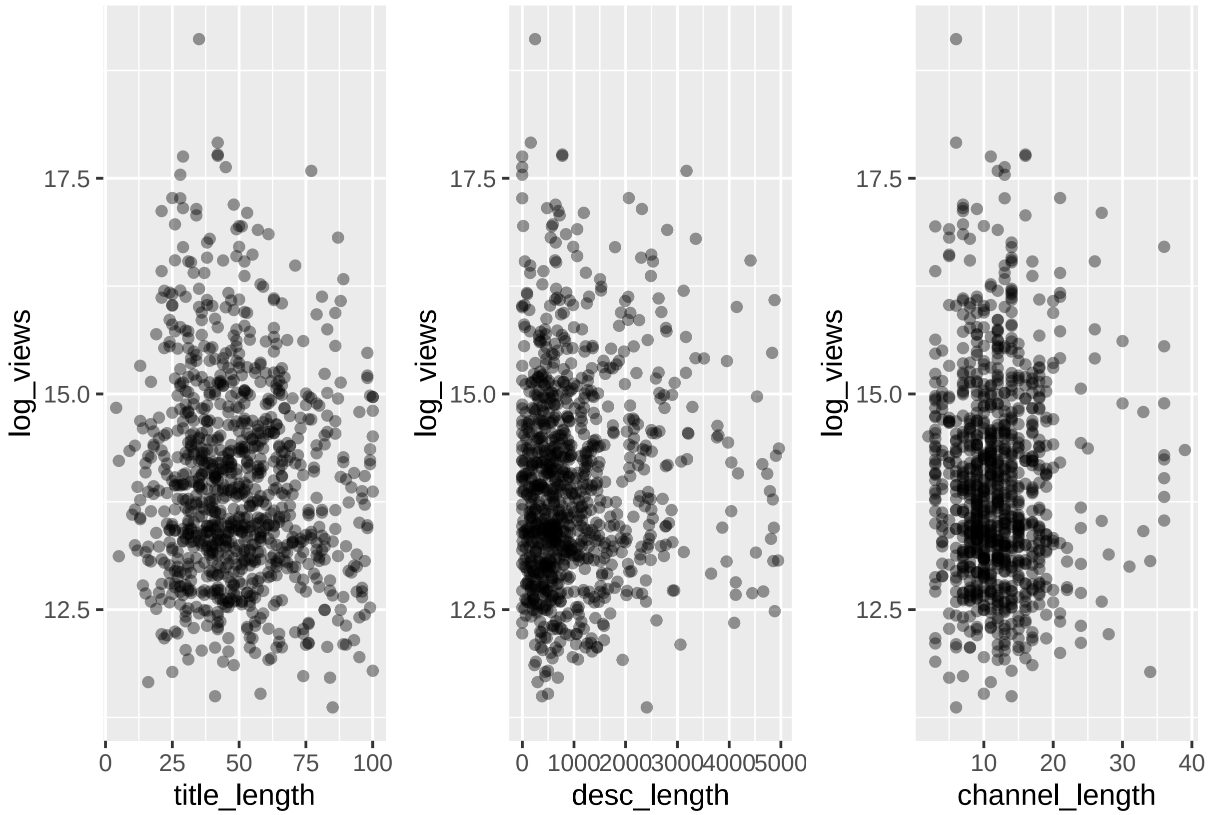
```
b7 <- ggplot(data = youtube, aes(x = num_tags, y = log_views)) +
  geom_point()

b8 <- ggplot(data = youtube, aes(x = num_caps, y = log_views)) +
  geom_point()

b9 <- ggplot(data = youtube, aes(x = num_exc, y = log_views)) +
  geom_boxplot()

b10 <- ggplot(data = youtube, aes(x = num_qm, y = log_views)) +
  geom_boxplot()
```

```r
b11 <- ggplot(data = youtube, aes(x = num_period, y = log_views)) +
  geom_boxplot()

b12 <- ggplot(data = youtube, aes(x = num_dollar, y = log_views)) +
  geom_boxplot()

b7+b8+b9+b10+b11+b12
```



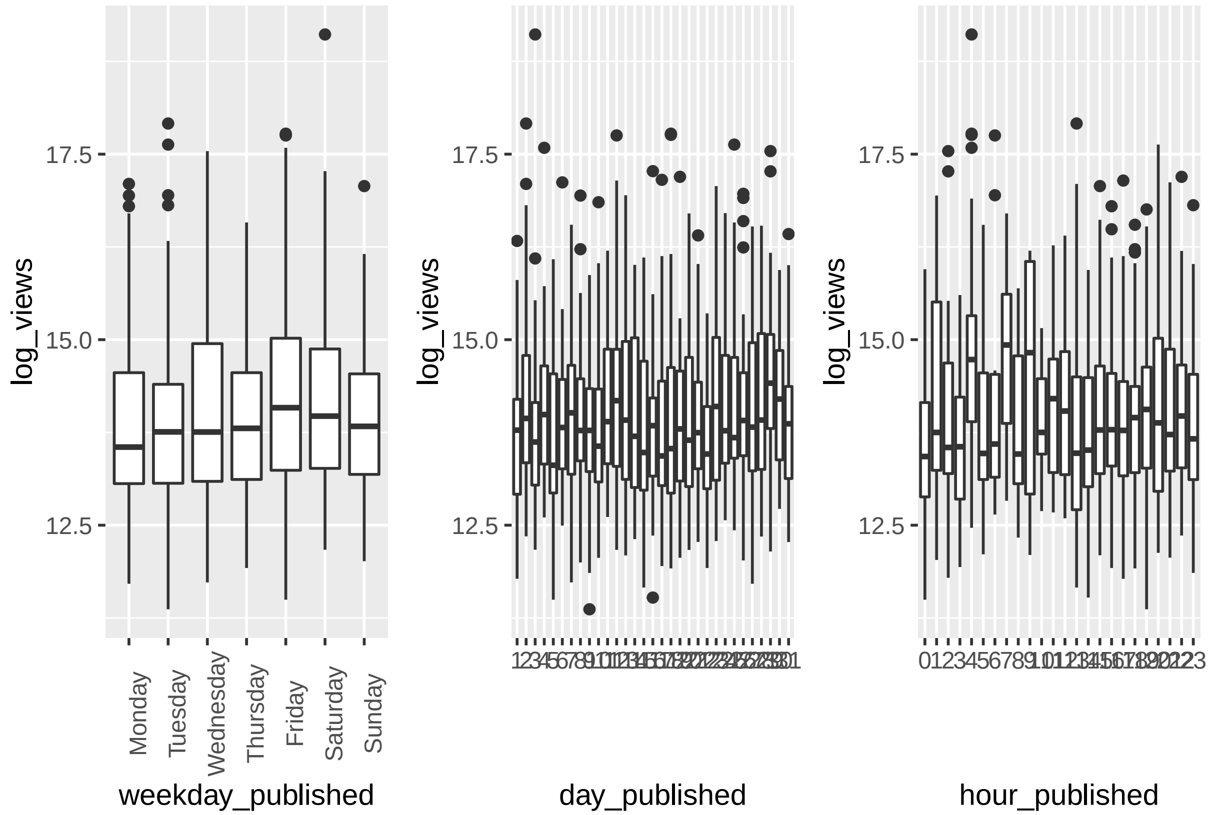
Here it looks like the `num_qm` has little to no predictive power. But we do see there is quite a bit of variability for `log_views` across the groups for `num_dollar`, `num_period`, and `num_exc`. Also there might be a weak downward trend for `num_caps` as well. Meanwhile `num_tags` looks completely random. So the number of tags doesn't appear to help us.

```r
b13 <- ggplot(data = youtube, aes(x = title_length, y = log_views)) +
  geom_point(alpha = 0.4)

b14 <- ggplot(data = youtube, aes(x = desc_length, y = log_views)) +
  geom_point(alpha = 0.4)

b15 <- ggplot(data = youtube, aes(x = channel_length, y = log_views)) +
  geom_point(alpha = 0.4)

b13+b14+b15
```

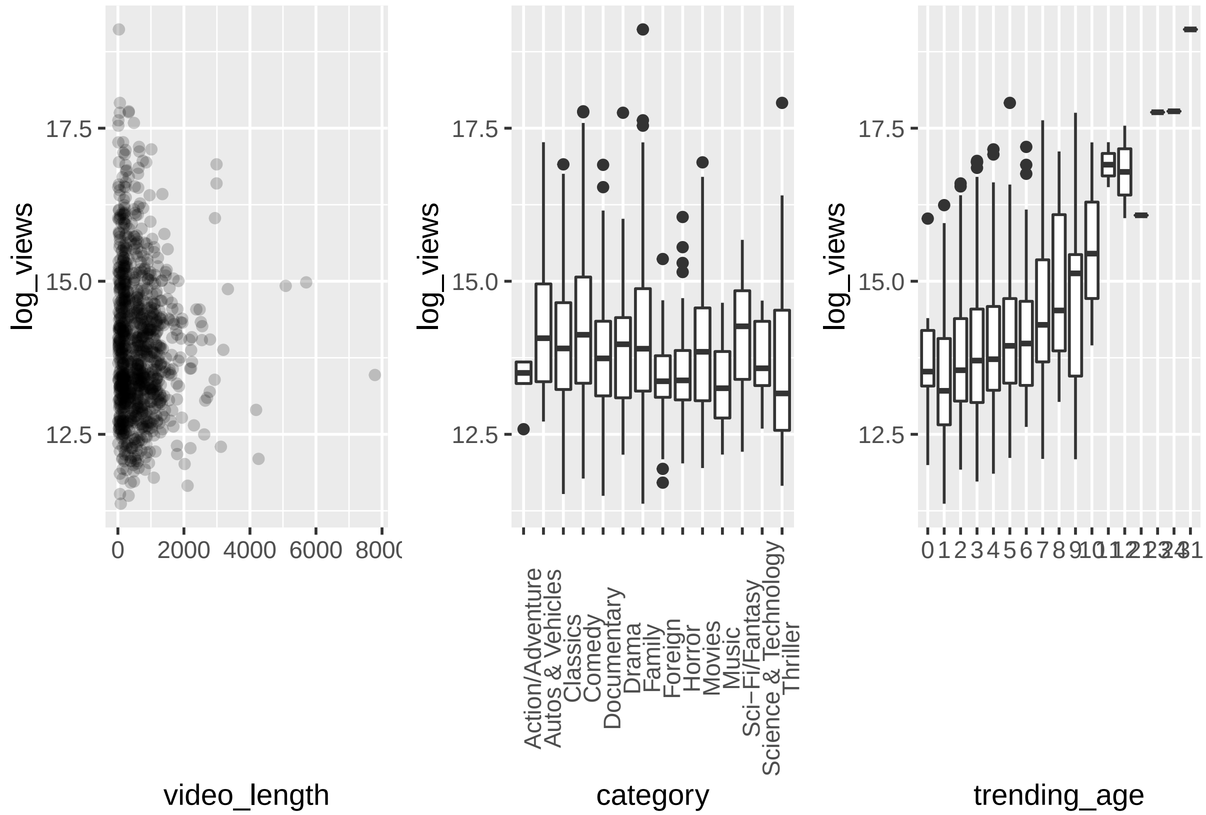


Here, the scatterplot for `title_length` looks quite random, as does the scatterplot for `channel_length`. Meanwhile there looks like there might be a weak positive trend for `desc_length`.

```
b15 <- ggplot(data = youtube, aes(x = weekday_published, y = log_views)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 90))

b16 <- ggplot(data = youtube, aes(x = day_published, y = log_views)) +
  geom_boxplot()

b17 <- ggplot(data = youtube, aes(x = hour_published, y = log_views)) +
  geom_boxplot()

b18 <- ggplot(data = youtube, aes(x = video_length, y = log_views)) +
  geom_point(alpha = 0.2)

b19 <- ggplot(data = youtube, aes(x = category, y = log_views)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 90))

b20 <- ggplot(data = youtube, aes(x = trending_age, y = log_views)) +
  geom_boxplot()

b15+b16+b17
```


log_views
17.5
15.0
12.5
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
weekday_published
day_published
hour_published

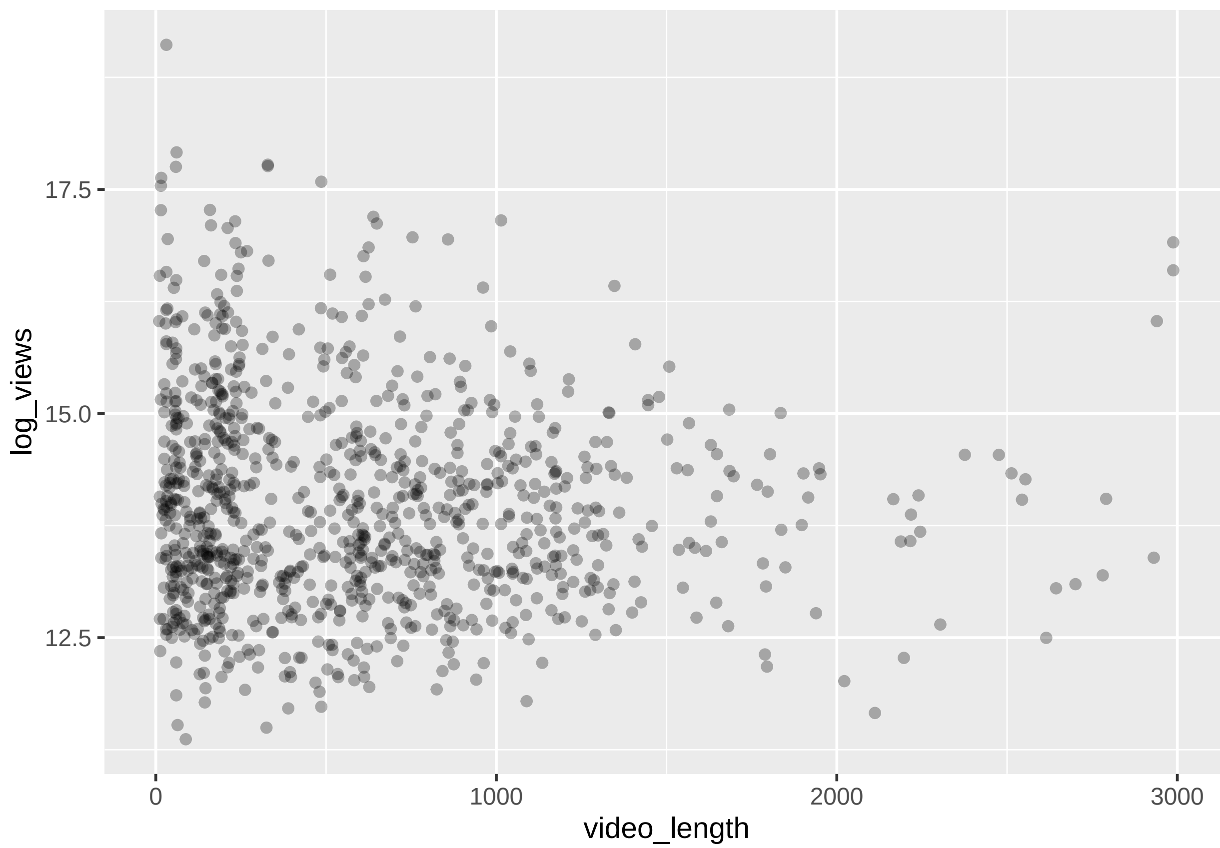
```
b18+b19+b20
```



It looks like `trending_age` and `log_views` have an association. This is expected since older videos will have had more time to get more views. That being said, the uploader will not know whether a video will trend or how long it takes to get to trending so it would not make sense to include it in our model. As for the category it looks like `log_views` varies quite a bit between different categories. Hence it might be useful to consider it in our model. For `video_length` it is hard to tell whether there is a relationship. We will construct another plot with adjusted limits

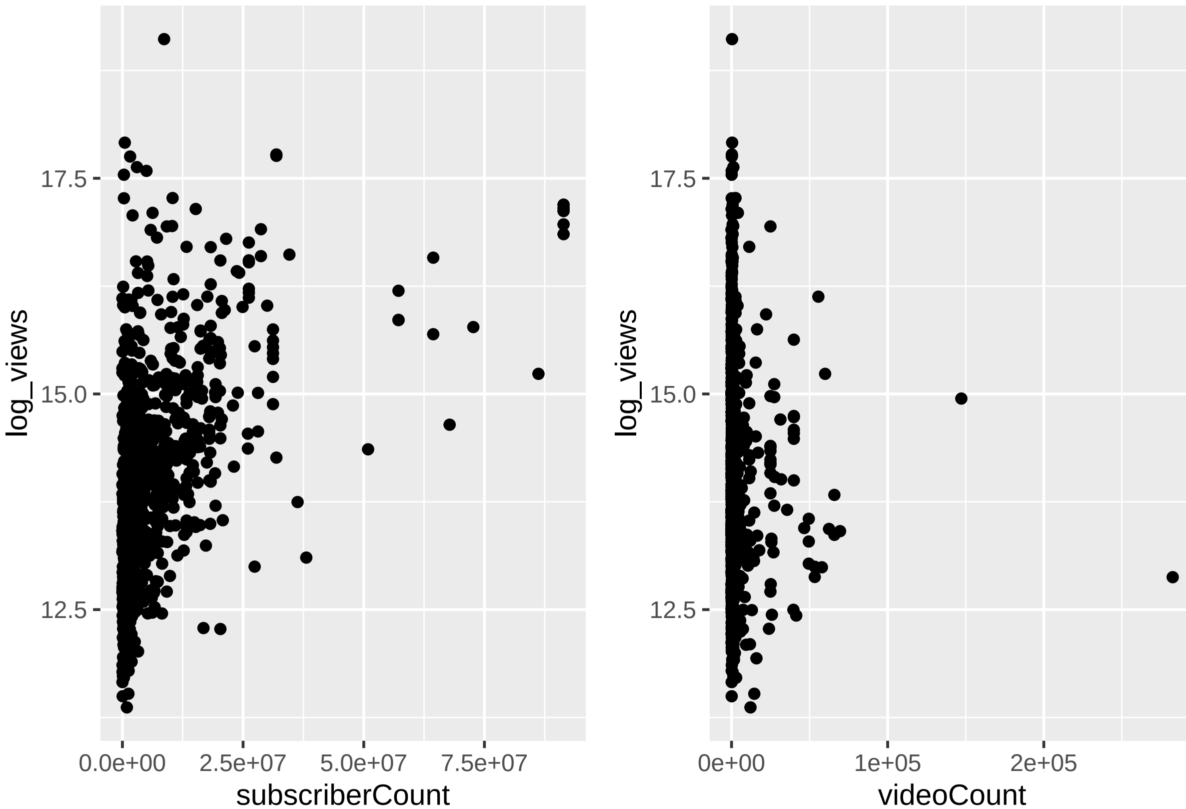
```
ggplot(data = youtube, aes(x = video_length, y = log_views)) +
  geom_point(alpha = 0.3) +
  xlim(0,3000)
```

```
## Warning: Removed 8 rows containing missing values (geom_point).
```



It is hard to tell whether there is a relationship between `log_views` and `video_length`. If anything, there is a very weak negative relationship between `video_length` and `log_views`, but it is hard to tell due to the difference in variability between short and long videos.

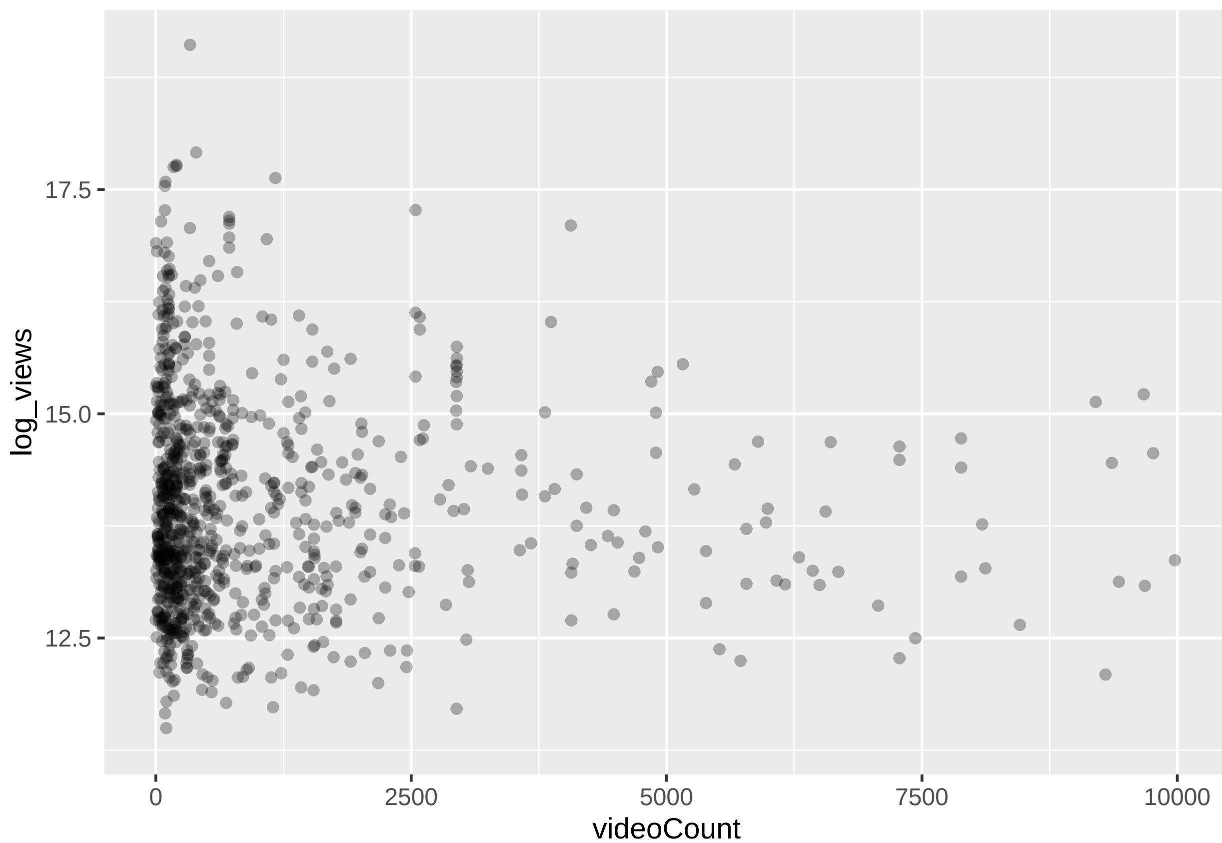
```
b21 <- ggplot(data = youtube, aes(x = subscriberCount, y = log_views)) +
  geom_point()

b22 <- ggplot(data = youtube, aes(x = videoCount, y = log_views)) +
  geom_point()

b21+b22
```



As we noted earlier, it looks like there may be a postive relationship between `subscriberCount` and `log_views`. Meanwhile `videoCount` doesn't seem to be all that useful. Just to check, we will adjust the limits of our plot to get a clearer picture.

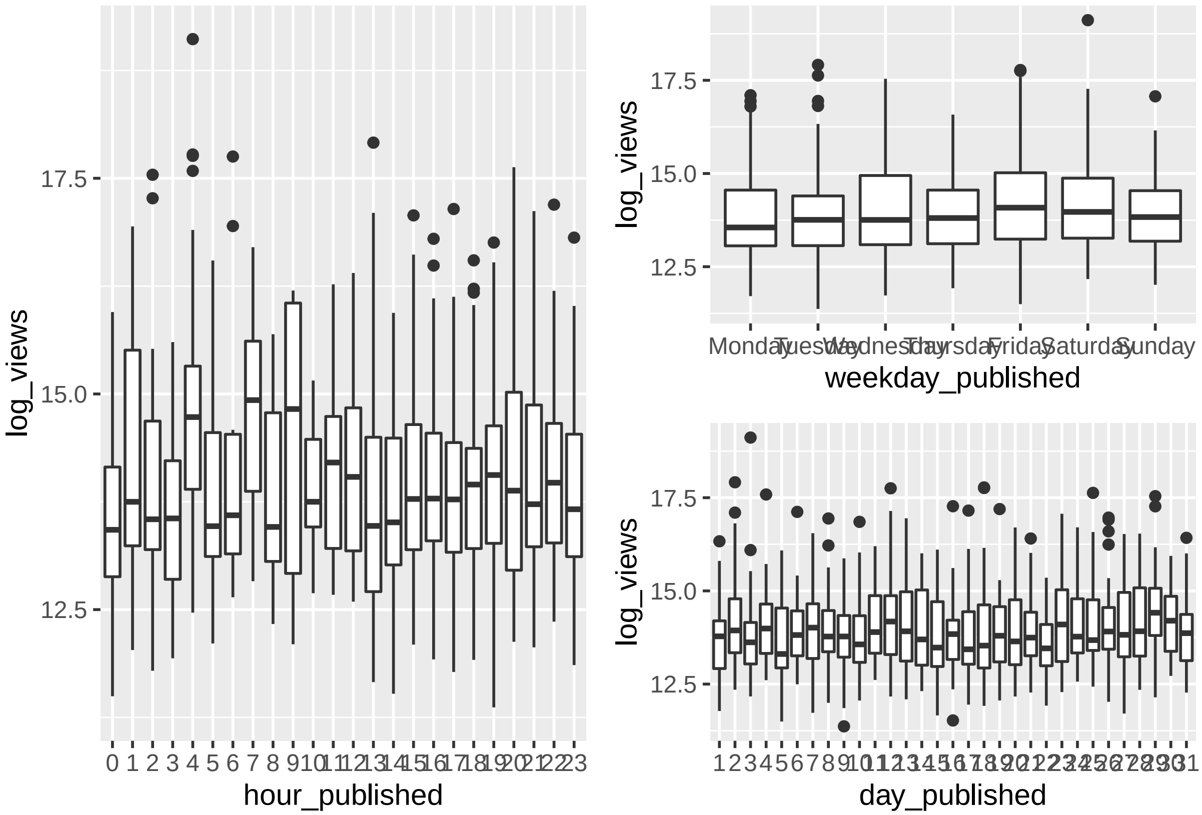
```
ggplot(data = youtube, aes(x = videoCount, y = log_views)) +
  geom_point(alpha = 0.3)+
  xlim(0,10000)
```

```
## Warning: Removed 73 rows containing missing values (geom_point).
```



Based off of this second plot, `videoCount` still does not appear to be all that useful. It doesn't look like the number of video's a channel has affects the number of views a trending video might have.

```
b23 <- ggplot(data = youtube, aes(x = weekday_published, y =log_views)) +
  geom_boxplot()

b24 <- ggplot(data = youtube, aes(x = hour_published, y =log_views)) +
  geom_boxplot()

b25 <- ggplot(data = youtube, aes(x = day_published, y =log_views)) +
  geom_boxplot()

b24+b23/b25
```




```
youtube %>% group_by(weekday_published) %>%
  summarise(across(log_views, mean))
```

```
## # A tibble: 7 x 2
##   weekday_published log_views
##   <fct>                 <dbl>
## 1 Monday                 13.9
## 2 Tuesday                13.8
## 3 Wednesday              14.0
## 4 Thursday               13.9
## 5 Friday                 14.2
## 6 Saturday               14.2
## 7 Sunday                 13.9
```

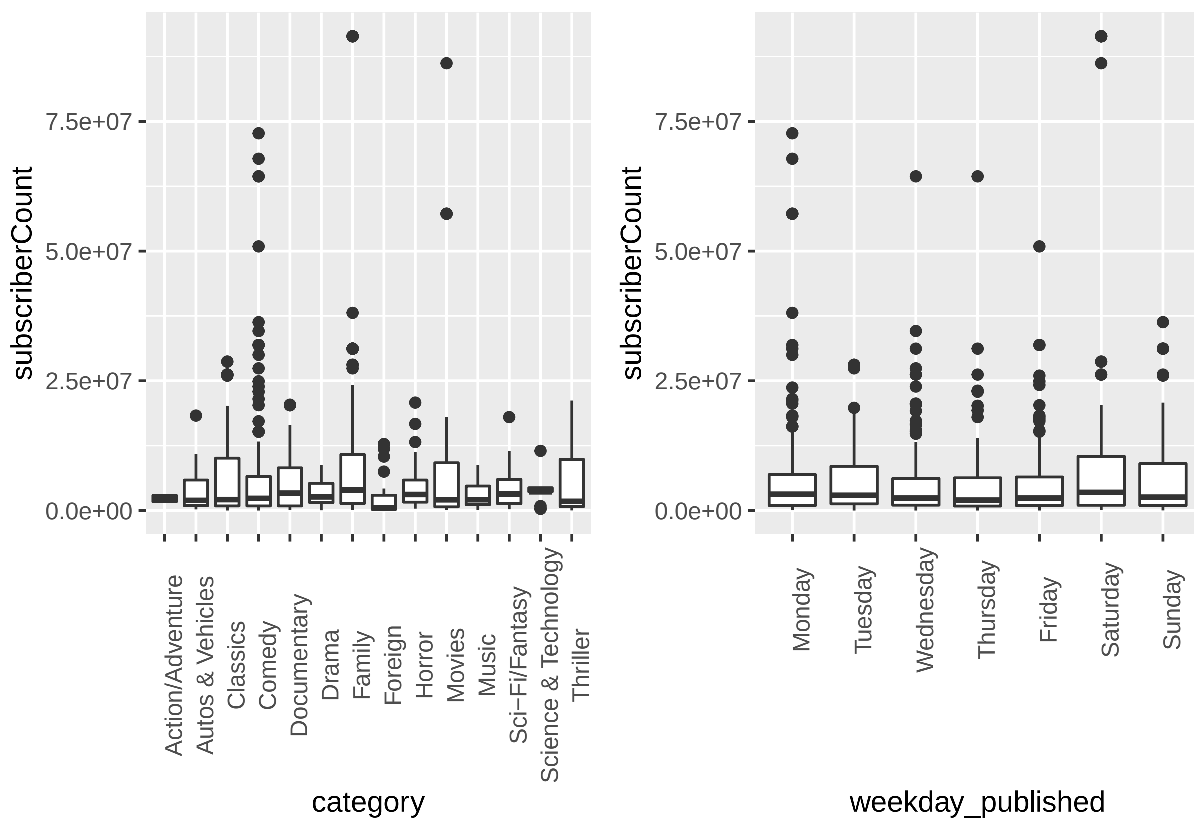
Here it looks like the day of the week might be useful in predicting `log_views`. While the boxplot doesn't look all that helpful, looking at the means across the weekdays, `log_views` for Friday and Saturday are closer to around 14.2 while that mean is closer to 13.8 for Mondays and Tuesdays. While this seems like a very small difference, this is in fact a $e^{14.2} - e^{13.8} = 484255$ view difference which is quite substantial. Looking at the box plots, there is quite a bit of variability in `log_views` across `hour_published`. So `hour_published` might be useful in our model. The same could be said about `day_published` as well.

Multivariate

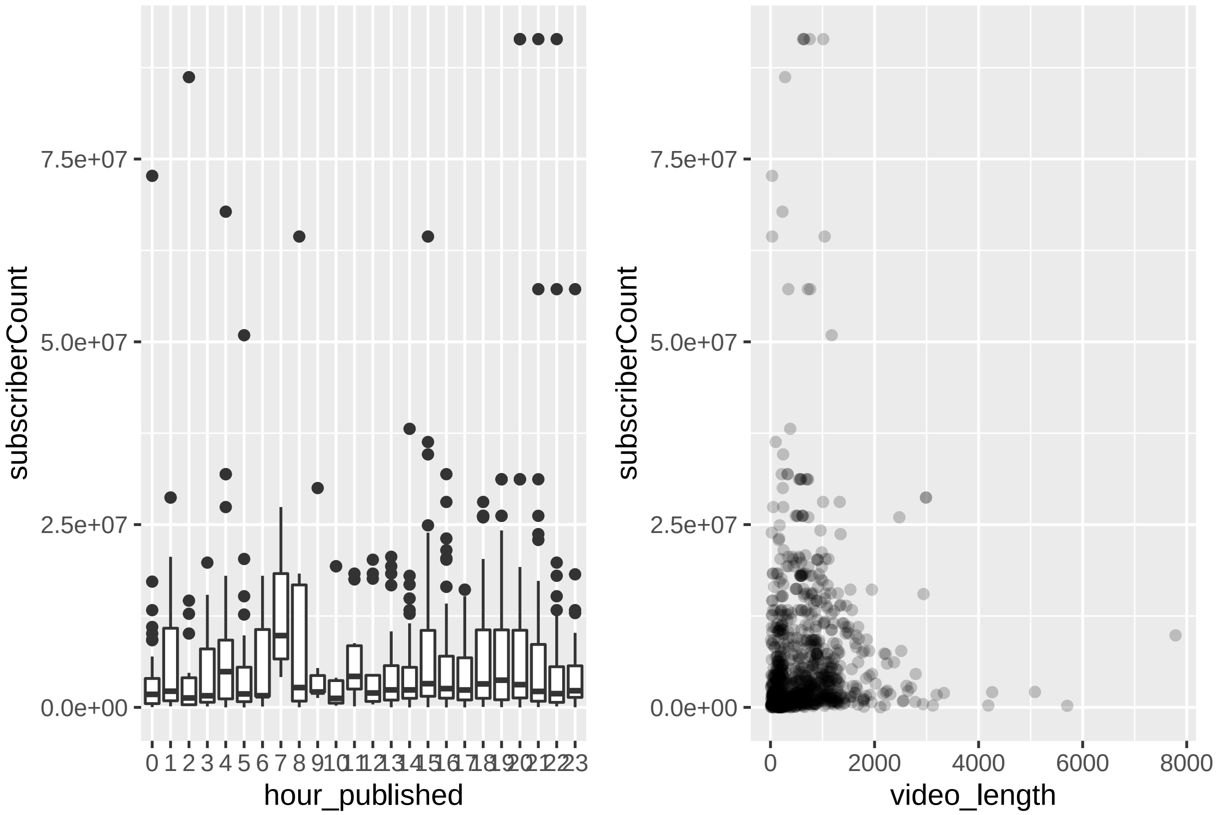
Now that we have identified some potentially useful predictor variables that may be useful, we will explore some of the relationships between them. As we noted in the previous section of our EDA, `subscriberCount`, `category`, `weekday_published`, `hour_published` and a few other variables. We will explore the relationships between some of these variables here.

We will start with subscriber count. It is possible that more popular channels tend to have videos of a certain length or tend to upload at specific times or days. It is also possible that subscriber counts vary across video categories.

```r
m1 <- ggplot(data = youtube, aes(x = category, y = subscriberCount)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 90))

m2 <- ggplot(data = youtube, aes(x = weekday_published, y = subscriberCount)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 90))

m3 <- ggplot(data = youtube, aes(x = hour_published, y = subscriberCount)) +
  geom_boxplot()

m4 <- ggplot(data = youtube, aes(x = video_length, y = subscriberCount)) +
  geom_point(alpha = 0.2)

m1+m2
```




```r
m3+m4
```

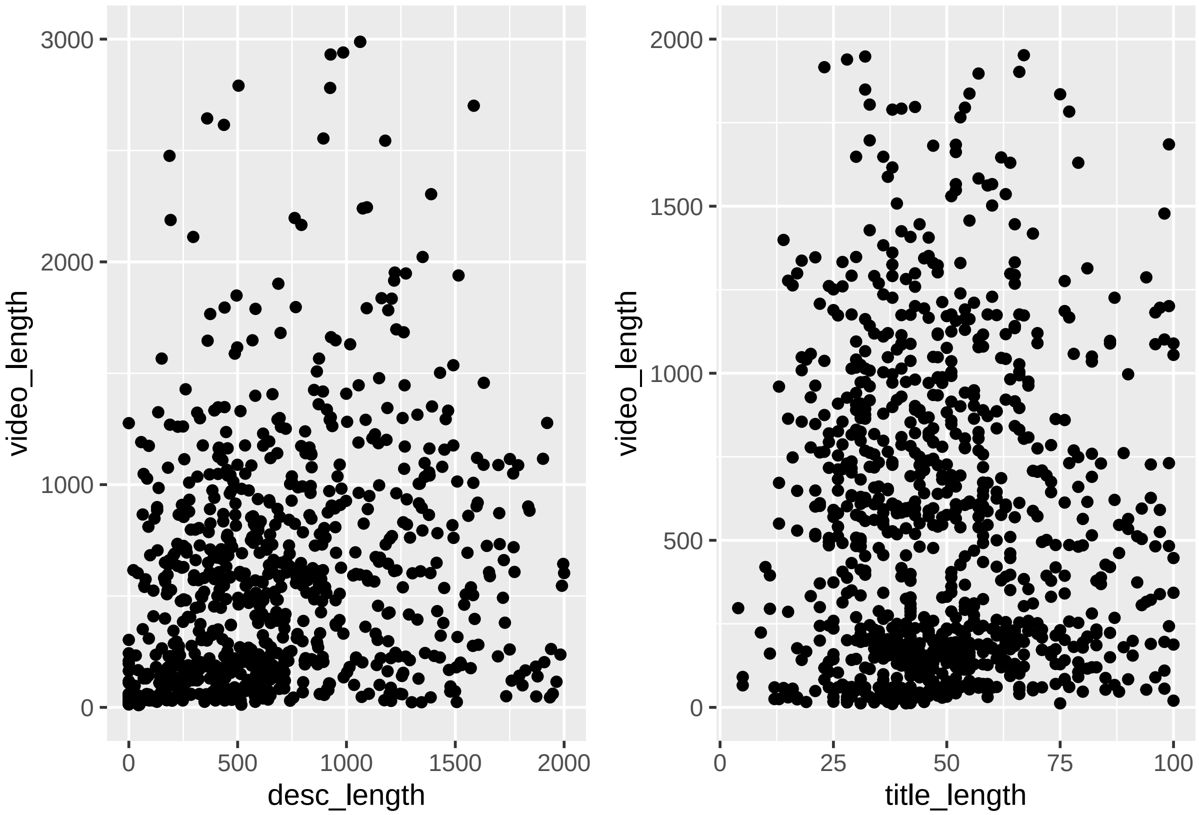


Looking at the above plots, it looks like the `hour_published` and `weekday_published` are potential covariates. Meanwhile `video_length` doesn't appear to be associated with `subscriberCount`. This isn't super surprising as the in the correlation matrix, we found the correlation between `subscriberCount` and `video_length` to be quite small.

Now we will look at `video_length`. A few possible relationships may be with respect to `desc_length` and `title_length`. Potentially longer videos might have more characters in the description or title since there might be more content to summarize in these videos.

```r
m5 <- ggplot(data = youtube, aes(x = desc_length, y = video_length)) +
  geom_point() + xlim(0,2000) + ylim(0, 3000)# note: limits adjusted to zoom in

m6 <- ggplot(data = youtube, aes(x = title_length, y = video_length)) +
  geom_point() + ylim(0,2000)

m5+m6
```

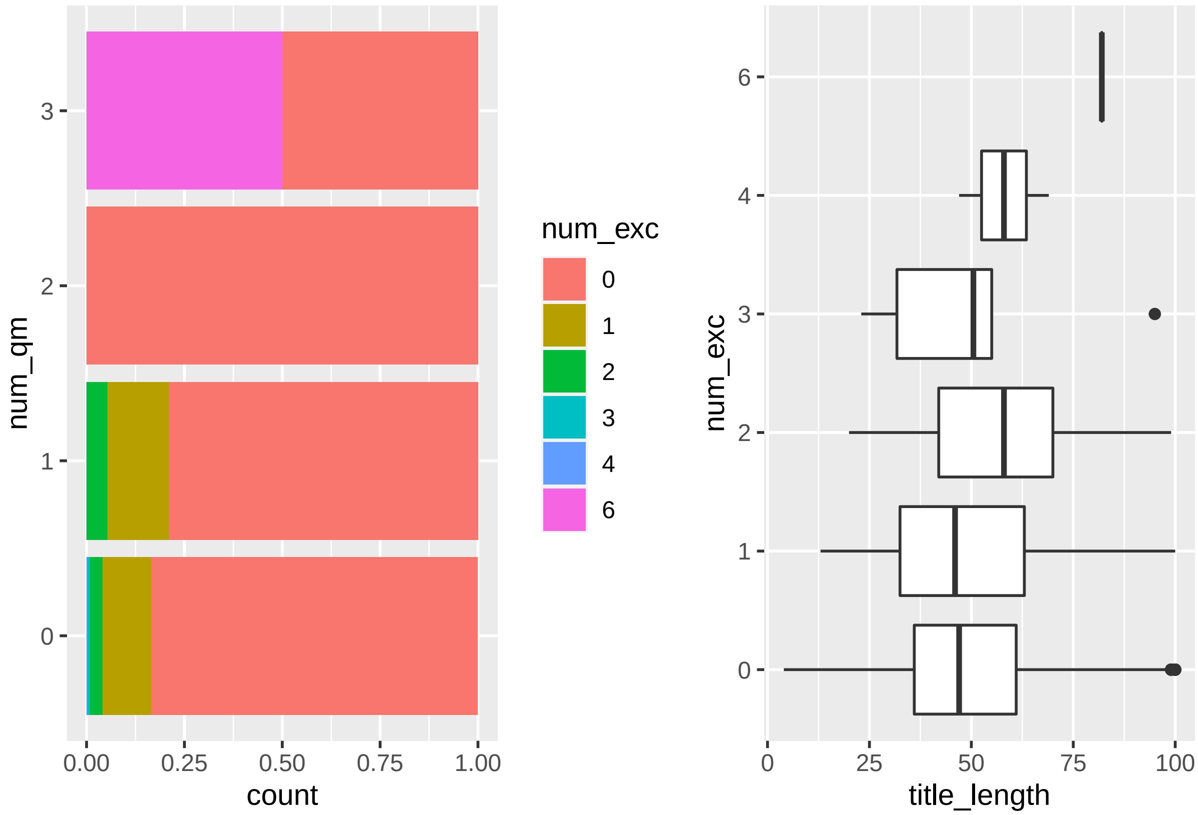
```
## Warning: Removed 148 rows containing missing values (geom_point).

## Warning: Removed 32 rows containing missing values (geom_point).
```



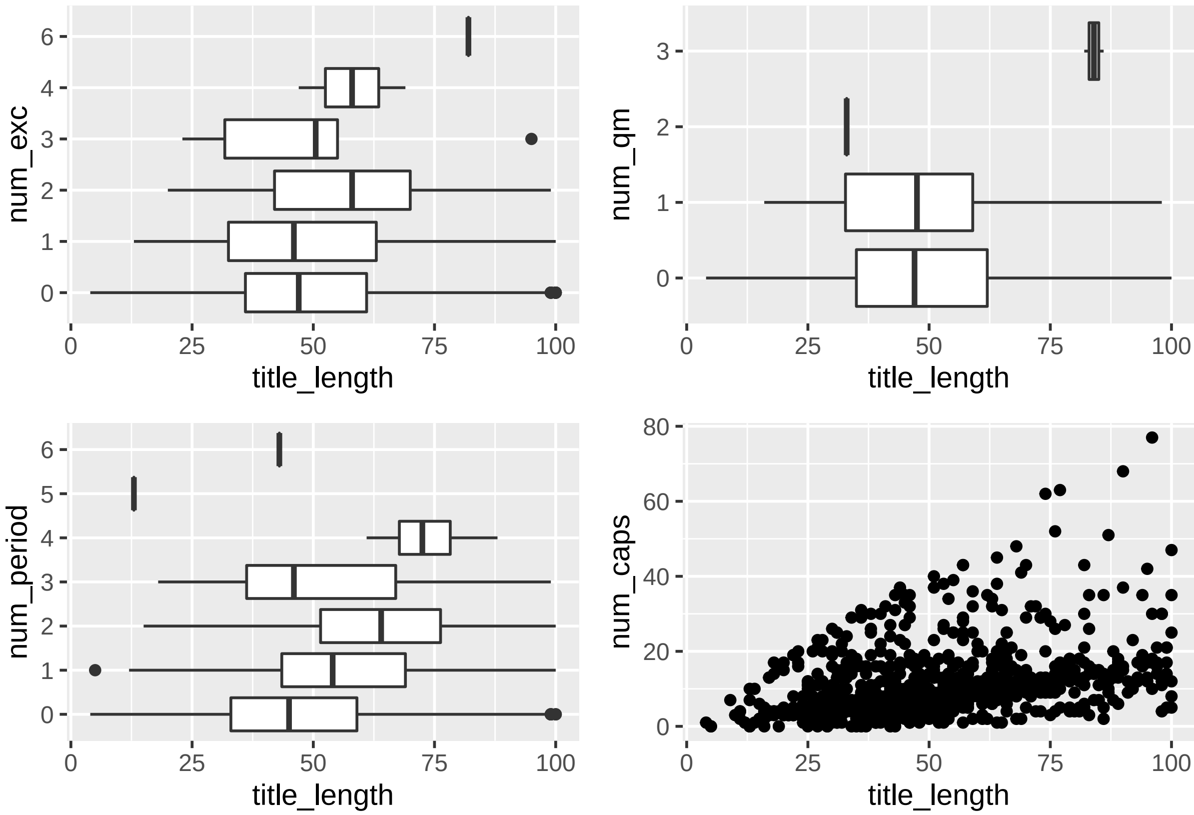
There doesn't seem to be much of a relationship in these cases. If anything there might be a very weak positive relationship between `title_length` and `video_length`.

One thing we would expect is that there is a relationship between many of the variables related to title. We would expect variables like `title_length`, `num_exc`, `num_qm` and so on to be related.

```
m7 <- ggplot(data = youtube, aes(fill = num_exc, y = num_qm))+
  geom_bar(position = 'fill')

m8 <- ggplot(data = youtube, aes(x = title_length, y = num_exc))+
  geom_boxplot()

m9 <- ggplot(data = youtube, aes(x = title_length, y = num_qm))+
  geom_boxplot()

m10 <- ggplot(data = youtube, aes(x = title_length, y = num_period))+
  geom_boxplot()

m11 <- ggplot(data = youtube, aes(x = title_length, y = num_caps))+
  geom_point()

m7 +m8
```


num_qm
count
num_exc
0
1
2
3
4
6
0.00
0.25
0.50
0.75
1.00
title_length
25
50
75
100

```
m8+m9+m10+m11
```



As we can see, many of these variables having to do with the title are associated. `title_length` and `num_caps` seem most obvious with a fairly clear positive linear association.

Once we decide on what variables to use in our model, we can further investigate whether any other predictors are associated with one another.

Modeling

As discussed earlier, the distribution for `view_count` is heavily skewed. Since the purpose of our analysis is to conduct inference on our model, we would like our response variable to be approximately normally distributed. Hence we will be using `log_views` instead of `view_count` for our predictions. As for the predictor variables, we only want to include predictor variables that the uploader might have knowledge of before uploading. So we will not be using variables such as `likes`, `dislikes`, and `trending_age` since the uploader will not know the values of these variables until the video is uploaded. This narrows down the variables we can use considerably. Following from the EDA we will build a model using the following predictor variables:

`subscriberCount`, `category`, `comments_disabled`, `ratings_disabled`, `num_caps`, `num_exc`, `num_period`, `num_dollar`, `desc_length`, `hour_published`, `weekday_published`, `video_length`

We can quickly build a model to predict `log_views` with these predictor variables with no interaction terms. But as we noted in the Exploratory data analysis, there seems to be a weak linear relationship between `num_exc`, `num_period`, `num_dollar` and `log_views`. So we will consider 2 separate models: one with those variables as factors, the other with those variables as integers. For the model with those variables treated as integers, we will use the data `youtube_int`.

```
youtube_int <- youtube %>%
  mutate(num_exc = as.integer(num_exc),
         num_period = as.integer(num_period),
         num_dollar = as.integer(num_dollar))
```

Now we will create 2 models both using all of the variables we mentioned above. In `full_model_factor`, we will treat `num_exc`, `num_period`, and `num_dollar` as factors. In `full_model_int` they will be treated as integers.

```
full_model_factor <- lm(log_views ~ subscriberCount + category + comments_disabled + ratings_disabled + 

full_model_int <- lm(log_views ~ subscriberCount + category + comments_disabled + ratings_disabled + nu

glance(full_model_factor)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.335         0.266  1.01      4.83 3.46e-36    92 -1344. 2876. 3335.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
glance(full_model_int)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.322         0.261  1.01      5.24 3.92e-37    81 -1353. 2872. 3278.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Looking at the AIC and BIC of our 2 models, it looks like our model with `full_model_int` is considered better. This isn't super surprising as they have fairly close values of R^2 but `full_model_factor` has 92 degrees of freedom compared to `full_model_int` with only 81. Seeing as AIC and BIC penalizes more complex models, this is expected. Since the difference in R^2 is still fairly small, we will be building off of `full_model_int` Now we will try running backwards selection on our model using AIC.

```
model_back_selection <- step(full_model_int, direction = 'backward')
```

```
## Start:  AIC=106.28
## log_views ~ subscriberCount + category + comments_disabled +
##     ratings_disabled + num_caps + num_exc + num_period + num_dollar +
##     desc_length + hour_published + weekday_published + day_published +
##     video_length
##
##                     Df Sum of Sq     RSS    AIC
## - day_published     30    28.405  946.36  75.96
## - hour_published    23    38.623  956.58 100.42
## - weekday_published  6     7.671  925.63 102.38
## - num_period         1     0.047  918.00 104.33
## - num_dollar         1     0.109  918.07 104.39
## - comments_disabled  1     0.204  918.16 104.49
## - ratings_disabled   1     0.404  918.36 104.71
## <none>                            917.96 106.28
## - desc_length        1     3.159  921.11 107.62
## - num_caps           1     3.312  921.27 107.79
## - video_length       1     3.772  921.73 108.27
## - category          13    29.119  947.08 110.70
## - num_exc            1     6.812  924.77 111.48
## - subscriberCount    1   237.735 1155.69 328.60
##
## Step:  AIC=75.96
## log_views ~ subscriberCount + category + comments_disabled +
##     ratings_disabled + num_caps + num_exc + num_period + num_dollar +
```

```
##     desc_length + hour_published + weekday_published + video_length
##
##                     Df Sum of Sq     RSS     AIC
## - hour_published    23    40.388  986.75  70.666
## - weekday_published  6     9.631  955.99  73.823
## - num_period         1     0.019  946.38  73.980
## - num_dollar         1     0.033  946.39  73.995
## - comments_disabled  1     0.175  946.54  74.141
## - ratings_disabled   1     0.620  946.98  74.599
## - category          13    25.533  971.89  75.891
## <none>                            946.36  75.961
## - desc_length        1     3.437  949.80  77.491
## - video_length       1     4.246  950.61  78.321
## - num_caps           1     4.542  950.90  78.624
## - num_exc            1     8.328  954.69  82.494
## - subscriberCount    1   239.872 1186.23 294.001
##
## Step:  AIC=70.67
## log_views ~ subscriberCount + category + comments_disabled +
##     ratings_disabled + num_caps + num_exc + num_period + num_dollar +
##     desc_length + weekday_published + video_length
##
##                     Df Sum of Sq     RSS     AIC
## - num_period         1     0.030  986.78  68.696
## - comments_disabled  1     0.076  986.82  68.741
## - num_dollar         1     0.093  986.84  68.758
## - weekday_published  6    10.685  997.43  69.156
## - ratings_disabled   1     1.734  988.48  70.376
## - category          13    26.450 1013.20  70.431
## <none>                            986.75  70.666
## - num_caps           1     3.908  990.66  72.516
## - desc_length        1     4.141  990.89  72.746
## - video_length       1     4.809  991.56  73.401
## - num_exc            1     9.926  996.68  78.415
## - subscriberCount    1   251.002 1237.75 289.409
##
## Step:  AIC=68.7
## log_views ~ subscriberCount + category + comments_disabled +
##     ratings_disabled + num_caps + num_exc + num_dollar + desc_length +
##     weekday_published + video_length
##
##                     Df Sum of Sq     RSS     AIC
## - comments_disabled  1     0.073  986.85  66.768
## - num_dollar         1     0.098  986.88  66.792
## - weekday_published  6    10.669  997.45  67.170
## - ratings_disabled   1     1.741  988.52  68.413
## - category          13    26.486 1013.26  68.494
## <none>                            986.78  68.696
## - num_caps           1     3.933  990.71  70.570
## - desc_length        1     4.138  990.92  70.771
## - video_length       1     4.787  991.57  71.409
## - num_exc            1    10.118  996.90  76.632
## - subscriberCount    1   252.004 1238.78 288.221
##
```

```
## Step:  AIC=66.77
## log_views ~ subscriberCount + category + ratings_disabled + num_caps +
##     num_exc + num_dollar + desc_length + weekday_published +
##     video_length
##
##                     Df Sum of Sq     RSS     AIC
## - num_dollar         1     0.094  986.95  64.861
## - weekday_published  6    10.781  997.63  65.352
## - ratings_disabled   1     1.724  988.58  66.468
## - category          13    26.548 1013.40  66.625
## <none>                            986.85  66.768
## - num_caps           1     3.892  990.74  68.602
## - desc_length        1     4.082  990.93  68.788
## - video_length       1     4.748  991.60  69.443
## - num_exc            1    10.128  996.98  74.713
## - subscriberCount    1   252.010 1238.86 286.283
##
## Step:  AIC=64.86
## log_views ~ subscriberCount + category + ratings_disabled + num_caps +
##     num_exc + desc_length + weekday_published + video_length
##
##                     Df Sum of Sq     RSS     AIC
## - weekday_published  6    10.856  997.80  63.516
## - ratings_disabled   1     1.719  988.67  64.556
## - category          13    26.483 1013.43  64.652
## <none>                            986.95  64.861
## - num_caps           1     3.987  990.93  66.787
## - desc_length        1     4.075  991.02  66.874
## - video_length       1     4.736  991.68  67.524
## - num_exc            1    10.042  996.99  72.721
## - subscriberCount    1   254.579 1241.53 286.375
##
## Step:  AIC=63.52
## log_views ~ subscriberCount + category + ratings_disabled + num_caps +
##     num_exc + desc_length + video_length
##
##                    Df Sum of Sq     RSS     AIC
## - ratings_disabled  1     1.645  999.45  63.121
## <none>                           997.80  63.516
## - category         13    28.080 1025.88  64.548
## - num_caps          1     4.275 1002.08  65.680
## - desc_length       1     4.292 1002.09  65.697
## - video_length      1     4.918 1002.72  66.305
## - num_exc           1    10.557 1008.36  71.767
## - subscriberCount   1   257.889 1255.69 285.425
##
## Step:  AIC=63.12
## log_views ~ subscriberCount + category + num_caps + num_exc +
##     desc_length + video_length
##
##                   Df Sum of Sq     RSS     AIC
## <none>                          999.45  63.121
## - category        13    28.227 1027.67  64.247
## - num_caps         1     3.819 1003.27  64.835
```

```
## - desc_length      1     4.055 1003.50  65.065
## - video_length     1     4.507 1003.95  65.503
## - num_exc          1    11.044 1010.49  71.824
## - subscriberCount  1   258.300 1257.75 285.019
```

```
glance(model_back_selection)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.262         0.248  1.02      18.9 1.49e-51    18 -1395. 2829. 2927.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Thus, according to the backwards selection our best model has the predictors `subscriberCount`, `num_exc`, `video_length`, `desc_length`, `num_caps`, `category`. But while we have a small AIC, our R^2 has dropped significantly from its level of about 0.32. So we are going to consider adding back some of our variables which had quite a bit of predictive power.

```
aic_model_weekday <- lm(log_views ~ subscriberCount + category + num_caps + num_exc + desc_length + vide
aic_model_hr <- lm(log_views ~ subscriberCount + category + num_caps + num_exc + desc_length + video_ler
aic_model_day <- lm(log_views ~ subscriberCount + category + num_caps + num_exc + desc_length + video_le
aic_model_weekday_hr <- lm(log_views ~ subscriberCount + category + num_caps + num_exc + desc_length + v


glance(aic_model_weekday)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.270         0.252  1.02      14.6 6.01e-50    24 -1389. 2831. 2958.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
glance(aic_model_hr)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.294         0.263  1.01      9.45 6.35e-47    41 -1373. 2833. 3043.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
glance(aic_model_day)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.286         0.249  1.02      7.73 8.98e-42    48 -1378. 2857. 3101.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
glance(aic_model_weekday_hr)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.301         0.265  1.01      8.47 7.28e-46    47 -1368. 2835. 3074.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

As shown above, it looks like adding back `weekday_published` and `hour_published` has increased our R^2 close to 0.3 shown in our model `aic_model_weekday_hr`. While our AIC has increased, R^2 has increased a fairly significant amount. Now we will consider adding in some interaction terms. So in our model we have `subscriberCount`,`category`,`num_caps`,`num_exc`,`desc_length`,`video_length`,`weekday_published`, and `hour_published`. As we noted in the EDA, the variables relating to the title seem to be associated. So we might want to include the interaction between `num_caps` and `num_exc` in our model. Also from the EDA, we might want to include the interaction between `subscriberCount` and `hour_published`. Additionally, we will include the interaction between `category` and `subscriberCount`.

```
model <- lm(log_views ~ subscriberCount * hour_published + category + num_caps * num_exc + desc_length +
            data = youtube_int)
tidy(model, conf.int = TRUE) %>%
  kable(digits = 3)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	12.286	1.947	6.311	0.000	8.465	16.106
subscriberCount	0.000	0.000	0.712	0.476	0.000	0.000
hour_published1	0.060	0.284	0.211	0.833	-0.498	0.618
hour_published2	0.503	0.281	1.793	0.073	-0.048	1.054
hour_published3	-0.462	0.287	-1.609	0.108	-1.026	0.102
hour_published4	0.959	0.244	3.927	0.000	0.480	1.438
hour_published5	-0.006	0.291	-0.019	0.985	-0.577	0.566
hour_published6	0.329	0.441	0.747	0.455	-0.536	1.195
hour_published7	-0.178	0.705	-0.252	0.801	-1.562	1.206
hour_published8	-0.292	0.484	-0.603	0.547	-1.241	0.658
hour_published9	0.408	0.485	0.842	0.400	-0.543	1.360
hour_published10	0.497	0.401	1.239	0.216	-0.290	1.285
hour_published11	0.391	0.492	0.795	0.427	-0.574	1.357
hour_published12	0.459	0.322	1.423	0.155	-0.174	1.091
hour_published13	-0.193	0.263	-0.733	0.464	-0.708	0.323
hour_published14	0.129	0.243	0.530	0.596	-0.347	0.605
hour_published15	0.036	0.220	0.163	0.870	-0.396	0.468
hour_published16	0.062	0.215	0.286	0.775	-0.361	0.484
hour_published17	0.140	0.225	0.620	0.536	-0.302	0.581
hour_published18	0.062	0.226	0.275	0.784	-0.382	0.506
hour_published19	-0.112	0.229	-0.489	0.625	-0.562	0.338
hour_published20	0.120	0.225	0.532	0.595	-0.323	0.562
hour_published21	0.078	0.227	0.345	0.730	-0.368	0.524
hour_published22	0.278	0.230	1.209	0.227	-0.173	0.729
hour_published23	0.122	0.245	0.499	0.618	-0.358	0.602
categoryAutos & Vehicles	1.600	1.953	0.819	0.413	-2.233	5.433
categoryClassics	1.156	1.942	0.595	0.552	-2.656	4.968
categoryComedy	1.489	1.943	0.766	0.444	-2.325	5.303
categoryDocumentary	1.283	1.944	0.660	0.509	-2.532	5.097
categoryDrama	1.259	1.953	0.645	0.519	-2.573	5.091
categoryFamily	1.580	1.940	0.814	0.416	-2.228	5.388
categoryForeign	0.780	1.949	0.400	0.689	-3.044	4.605
categoryHorror	1.164	1.953	0.596	0.551	-2.668	4.997
categoryMovies	1.326	1.941	0.683	0.495	-2.484	5.136
categoryMusic	0.542	1.976	0.274	0.784	-3.336	4.421
categorySci-Fi/Fantasy	1.537	1.965	0.782	0.434	-2.320	5.394
categoryScience & Technology	1.013	1.996	0.507	0.612	-2.904	4.930
categoryThriller	1.024	1.951	0.525	0.600	-2.806	4.854
num_caps	-0.004	0.008	-0.496	0.620	-0.019	0.011
num_exc	-0.160	0.101	-1.583	0.114	-0.359	0.038
desc_length	0.000	0.000	1.430	0.153	0.000	0.000
video_length	0.000	0.000	-2.800	0.005	0.000	0.000

term	estimate	std.error	statistic	p.value	conf.low	conf.high
weekday_publishedTuesday	0.035	0.120	0.293	0.770	-0.201	0.271
weekday_publishedWednesday	0.201	0.125	1.610	0.108	-0.044	0.446
weekday_publishedThursday	0.133	0.126	1.051	0.293	-0.115	0.381
weekday_publishedFriday	0.264	0.126	2.091	0.037	0.016	0.511
weekday_publishedSaturday	0.188	0.132	1.433	0.152	-0.070	0.447
weekday_publishedSunday	0.154	0.124	1.238	0.216	-0.090	0.397
subscriberCount:hour_published1	0.000	0.000	1.836	0.067	0.000	0.000
subscriberCount:hour_published2	0.000	0.000	-1.058	0.290	0.000	0.000
subscriberCount:hour_published3	0.000	0.000	1.684	0.092	0.000	0.000
subscriberCount:hour_published4	0.000	0.000	-0.762	0.446	0.000	0.000
subscriberCount:hour_published5	0.000	0.000	-0.364	0.716	0.000	0.000
subscriberCount:hour_published6	0.000	0.000	0.752	0.452	0.000	0.000
subscriberCount:hour_published7	0.000	0.000	1.020	0.308	0.000	0.000
subscriberCount:hour_published8	0.000	0.000	-0.120	0.905	0.000	0.000
subscriberCount:hour_published9	0.000	0.000	0.803	0.422	0.000	0.000
subscriberCount:hour_published10	0.000	0.000	-0.474	0.636	0.000	0.000
subscriberCount:hour_published11	0.000	0.000	-0.243	0.808	0.000	0.000
subscriberCount:hour_published12	0.000	0.000	-0.197	0.844	0.000	0.000
subscriberCount:hour_published13	0.000	0.000	1.754	0.080	0.000	0.000
subscriberCount:hour_published14	0.000	0.000	-1.429	0.153	0.000	0.000
subscriberCount:hour_published15	0.000	0.000	0.225	0.822	0.000	0.000
subscriberCount:hour_published16	0.000	0.000	0.802	0.423	0.000	0.000
subscriberCount:hour_published17	0.000	0.000	0.383	0.702	0.000	0.000
subscriberCount:hour_published18	0.000	0.000	-0.066	0.948	0.000	0.000
subscriberCount:hour_published19	0.000	0.000	1.761	0.079	0.000	0.000
subscriberCount:hour_published20	0.000	0.000	0.519	0.604	0.000	0.000
subscriberCount:hour_published21	0.000	0.000	0.660	0.509	0.000	0.000
subscriberCount:hour_published22	0.000	0.000	0.257	0.797	0.000	0.000
subscriberCount:hour_published23	0.000	0.000	0.330	0.742	0.000	0.000
num_caps:num_exc	-0.003	0.004	-0.603	0.547	-0.011	0.006
subscriberCount:categoryAutos & Vehicles	0.000	0.000	-0.596	0.551	0.000	0.000
subscriberCount:categoryClassics	0.000	0.000	-0.601	0.548	0.000	0.000
subscriberCount:categoryComedy	0.000	0.000	-0.662	0.508	0.000	0.000
subscriberCount:categoryDocumentary	0.000	0.000	-0.642	0.521	0.000	0.000
subscriberCount:categoryDrama	0.000	0.000	-0.576	0.564	0.000	0.000
subscriberCount:categoryFamily	0.000	0.000	-0.676	0.499	0.000	0.000
subscriberCount:categoryForeign	0.000	0.000	-0.586	0.558	0.000	0.000
subscriberCount:categoryHorror	0.000	0.000	-0.655	0.513	0.000	0.000
subscriberCount:categoryMovies	0.000	0.000	-0.669	0.504	0.000	0.000
subscriberCount:categoryMusic	0.000	0.000	-0.494	0.622	0.000	0.000
subscriberCount:categorySci-Fi/Fantasy	0.000	0.000	-0.592	0.554	0.000	0.000
subscriberCount:categoryScience & Technology	0.000	0.000	-0.612	0.541	0.000	0.000
subscriberCount:categoryThriller	0.000	0.000	-0.600	0.548	0.000	0.000

Looking at the model output, we see that the interactions between `subscriberCount` and `hour_published` mostly seem to have fairly large p-values. While there are a few with small p-values, such as `subscriberCount:hour_published1` with a p-value of 0.008, looking at the confidence interval, it is extremely close to 0. So while the p-value suggests we include the interaction in our model, since the confidence interval is so close to 0, we will not include it. Also, the interaction between `num_caps` and `num_exc` has a large p-value of 0.404 and the confidence interval captures 0. So that interaction will not be included in the final model as well. We reach the same conclusions for the interactions between `subscriberCount` and `category`. So our model will not include any interaction terms. But looking at the coefficient for `subscriberCount`, it is essentially 0. Looking at the confidence interval, it is also extremely close to 0. So while the p-value suggests that `subscriberCount` is important, we will remove it from our model.

```
model <- lm(log_views ~ category + num_caps + num_exc + desc_length + video_length + weekday_published +
```

But one thing we have not addressed is outliers. Just looking at the EDA, we have quite a few outliers in both the response variable and many of the predictors. Since this is the case it is quite likely there are a few leverage points.

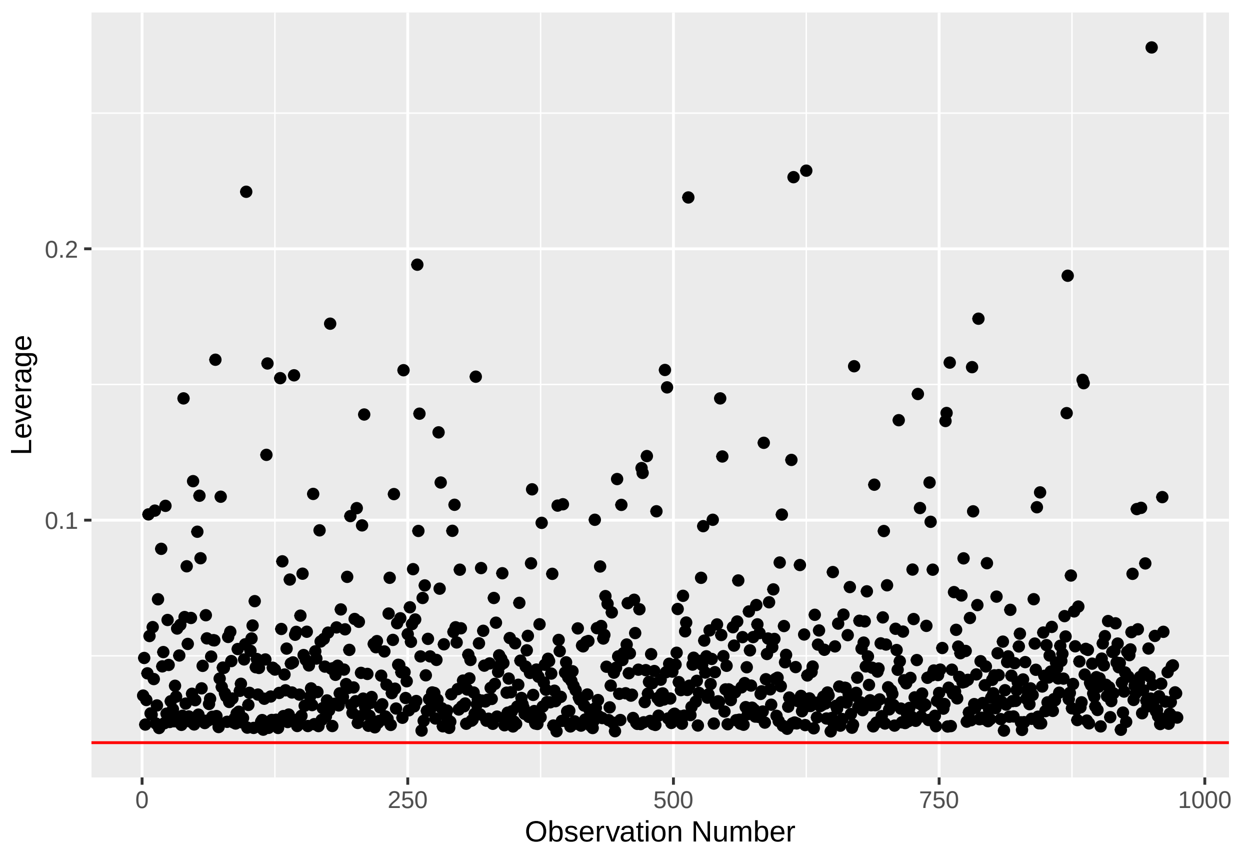
```
model_aug <- augment(model) %>%
  mutate(obs_num = 1:n())
```

Now we will investigate the leverage for each of the observations in our dataset. We will set the threshold for leverage at

$$\frac{2(p+1)}{n} = \frac{2(8+1)}{974} = 0.018$$

We can plot the leverage for each observation.

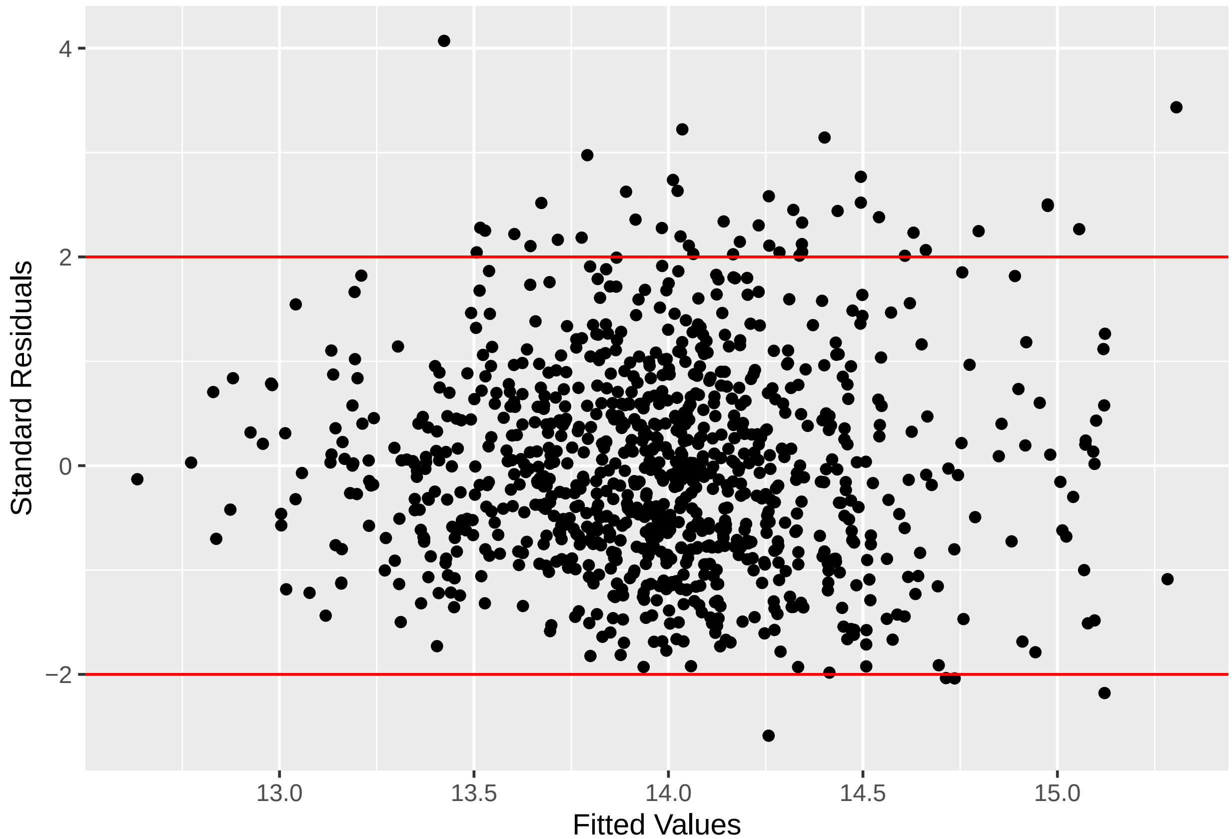
```
ggplot(data = model_aug, aes(x = obs_num, y = .hat))+
  geom_point() +
  geom_hline(yintercept = 0.018, color = 'red') +
  labs(x = 'Observation Number', y = 'Leverage')
```



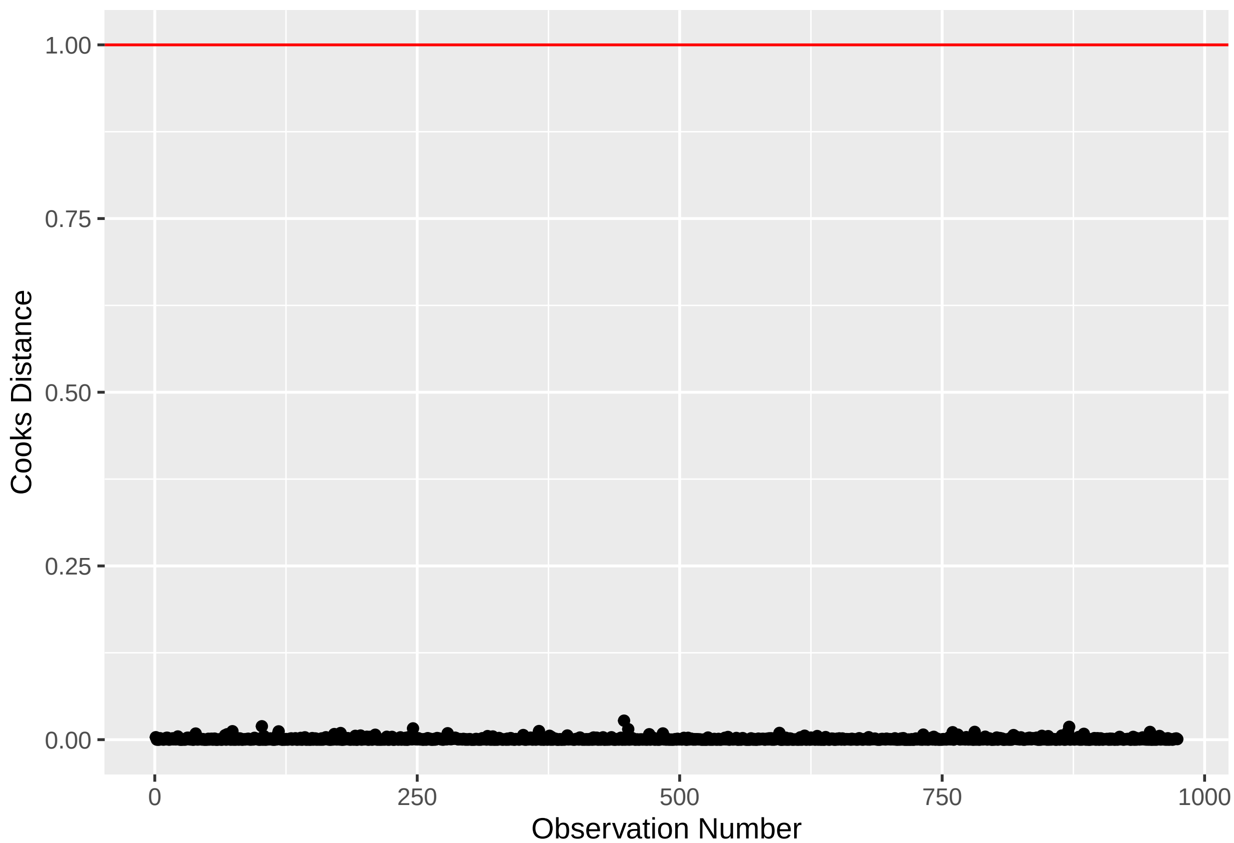
Weirdly it looks like all observations lie above our threshold for

Now we will look at the standardized residuals as well as the cooks distance.

```r
ggplot(data = model_aug, aes(x = .fitted, y = .std.resid))+
  geom_point() +
  geom_hline(yintercept = -2, color = 'red')+
  geom_hline(yintercept = 2, color = 'red') +
  labs(x = 'Fitted Values', y='Standard Residuals')
```




```r
ggplot(data = model_aug, aes(x = obs_num, y = .cooksd))+
  geom_point() +
  geom_hline(yintercept = 1, color = 'red')+
  labs(x = 'Observation Number', y='Cooks Distance')
```




```
model_aug[which(abs(model_aug$.cooksd)>1),]
```

```
## # A tibble: 0 x 15
## # ... with 15 variables: log_views <dbl>, category <chr>, num_caps <int>,
## #   num_exc <int>, desc_length <dbl>, video_length <dbl>,
## #   weekday_published <fct>, hour_published <fct>, .fitted <dbl>, .resid <dbl>,
## #   .hat <dbl>, .sigma <dbl>, .cooksd <dbl>, .std.resid <dbl>, obs_num <int>
```

```
model_aug[which(abs(model_aug$.std.resid)>2),]
```

```
## # A tibble: 48 x 15
##    log_views category num_caps num_exc desc_length video_length weekday_publish~
##        <dbl> <chr>       <int>   <int>       <dbl>        <dbl> <fct>
##  1      16.7 Movies          7       1         986          331 Saturday
##  2      12.7 Comedy          7       1        2319          143 Friday
##  3      17.3 Autos &~       11       1        2060          159 Saturday
##  4      16.0 Family         30       2          45           10 Monday
##  5      16.6 Comedy          9       1        2495          243 Wednesday
##  6      17.9 Thriller        6       1         164           61 Tuesday
##  7      17.3 Family          4       1           0           15 Wednesday
##  8      16.5 Comedy          7       1        4413          192 Friday
##  9      11.4 Family         15       1        2407           88 Tuesday
## 10      17.6 Family         22       1           0           16 Tuesday
## # ... with 38 more rows, and 8 more variables: hour_published <fct>,
## #   .fitted <dbl>, .resid <dbl>, .hat <dbl>, .sigma <dbl>, .cooksd <dbl>,
## #   .std.resid <dbl>, obs_num <int>
```

It looks like we have 48 observations which lie outside our standard residual threshold. Meanwhile we have a no observations lying above the threshold for Cooks Distance. Since this is the case and these observations are perfectly valid videos, we will not remove them from the dataset.

Hence we can get our final model

```
final_model <- lm(log_views ~  category + num_caps + num_exc + desc_length + video_length + weekday_publ

tidy(final_model, conf.int = TRUE) %>%
  kable(digits = 3)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	13.639	0.572	23.846	0.000	12.516	14.761
categoryAutos & Vehicles	0.532	0.563	0.945	0.345	-0.573	1.636
categoryClassics	0.314	0.534	0.587	0.557	-0.734	1.361
categoryComedy	0.373	0.540	0.692	0.489	-0.686	1.432
categoryDocumentary	0.190	0.545	0.348	0.728	-0.881	1.260
categoryDrama	0.148	0.549	0.269	0.788	-0.931	1.226
categoryFamily	0.500	0.531	0.942	0.347	-0.542	1.543
categoryForeign	-0.350	0.566	-0.619	0.536	-1.461	0.760
categoryHorror	-0.016	0.554	-0.029	0.977	-1.104	1.071
categoryMovies	0.214	0.538	0.398	0.690	-0.842	1.271
categoryMusic	-0.314	0.587	-0.535	0.593	-1.466	0.838
categorySci-Fi/Fantasy	0.533	0.568	0.937	0.349	-0.583	1.648
categoryScience & Technology	-0.024	0.621	-0.038	0.969	-1.242	1.194
categoryThriller	0.007	0.561	0.012	0.990	-1.095	1.109
num_caps	-0.012	0.004	-2.769	0.006	-0.021	-0.004
num_exc	-0.224	0.070	-3.196	0.001	-0.361	-0.086
desc_length	0.000	0.000	2.372	0.018	0.000	0.000
video_length	0.000	0.000	-1.094	0.274	0.000	0.000
weekday_publishedTuesday	-0.032	0.134	-0.236	0.814	-0.295	0.232
weekday_publishedWednesday	0.157	0.141	1.112	0.267	-0.120	0.433
weekday_publishedThursday	0.048	0.142	0.339	0.734	-0.231	0.327
weekday_publishedFriday	0.181	0.142	1.278	0.201	-0.097	0.459
weekday_publishedSaturday	0.378	0.145	2.607	0.009	0.093	0.662
weekday_publishedSunday	0.136	0.139	0.982	0.326	-0.136	0.408
hour_published1	0.558	0.266	2.093	0.037	0.035	1.080
hour_published2	0.473	0.292	1.619	0.106	-0.100	1.047
hour_published3	-0.083	0.269	-0.308	0.758	-0.611	0.446
hour_published4	1.029	0.244	4.219	0.000	0.550	1.507
hour_published5	0.050	0.295	0.168	0.867	-0.529	0.628
hour_published6	0.687	0.386	1.781	0.075	-0.070	1.443
hour_published7	0.990	0.438	2.261	0.024	0.131	1.849
hour_published8	0.071	0.465	0.152	0.879	-0.841	0.983
hour_published9	0.699	0.466	1.502	0.133	-0.214	1.613
hour_published10	0.289	0.386	0.748	0.454	-0.468	1.046
hour_published11	0.391	0.392	0.998	0.319	-0.378	1.160
hour_published12	0.413	0.309	1.338	0.181	-0.193	1.019
hour_published13	0.096	0.244	0.391	0.696	-0.384	0.575
hour_published14	-0.038	0.236	-0.160	0.873	-0.500	0.425
hour_published15	0.206	0.219	0.942	0.347	-0.223	0.635
hour_published16	0.217	0.212	1.024	0.306	-0.199	0.634
hour_published17	0.184	0.212	0.871	0.384	-0.231	0.599
hour_published18	0.265	0.216	1.225	0.221	-0.160	0.690
hour_published19	0.338	0.219	1.541	0.124	-0.092	0.767

term	estimate	std.error	statistic	p.value	conf.low	conf.high
hour_published20	0.425	0.229	1.857	0.064	-0.024	0.874
hour_published21	0.351	0.230	1.530	0.126	-0.099	0.802
hour_published22	0.366	0.237	1.543	0.123	-0.099	0.831
hour_published23	0.249	0.244	1.017	0.309	-0.231	0.728

Since we took the log of view count, all of our interpretations for the coefficients need to be adjusted. We will not interprate all of the coefficients, but we will interpret a few. For `num_exc`, we have a coefficient of -0.224. Interpreting this coefficient, this means that holding all else constant, if we were to increase the number of exclamation points in the title by 1, we would expect the number of views to be multiplied by a factor of $e^{-0.224} = 0.799$. Interpreting `weekday_publishedSaturday` with a coefficient of 0.378, this means that holding all else constant, if we were to change the publish date of the same video to Saturday from Monday, we would expect the number of views to be multiplied by a factor of $e^{0.378} = 1.459$.

The coefficients for the different categories of `weekday_published` are quite interesting. It seems that holding all else constant, changing the day published from Monday to any other day except Tuesday is associated with an increase in viewership. Looking at `hour_published4`, it looks like 4 and 7 am UTC lead to the largest increases in viewership from midnight UTC holding all else constant. These times correspond to 8 and 11 pm PST. Since we are looking at videos in the US, this might suggest that youtube users are most active during those times.

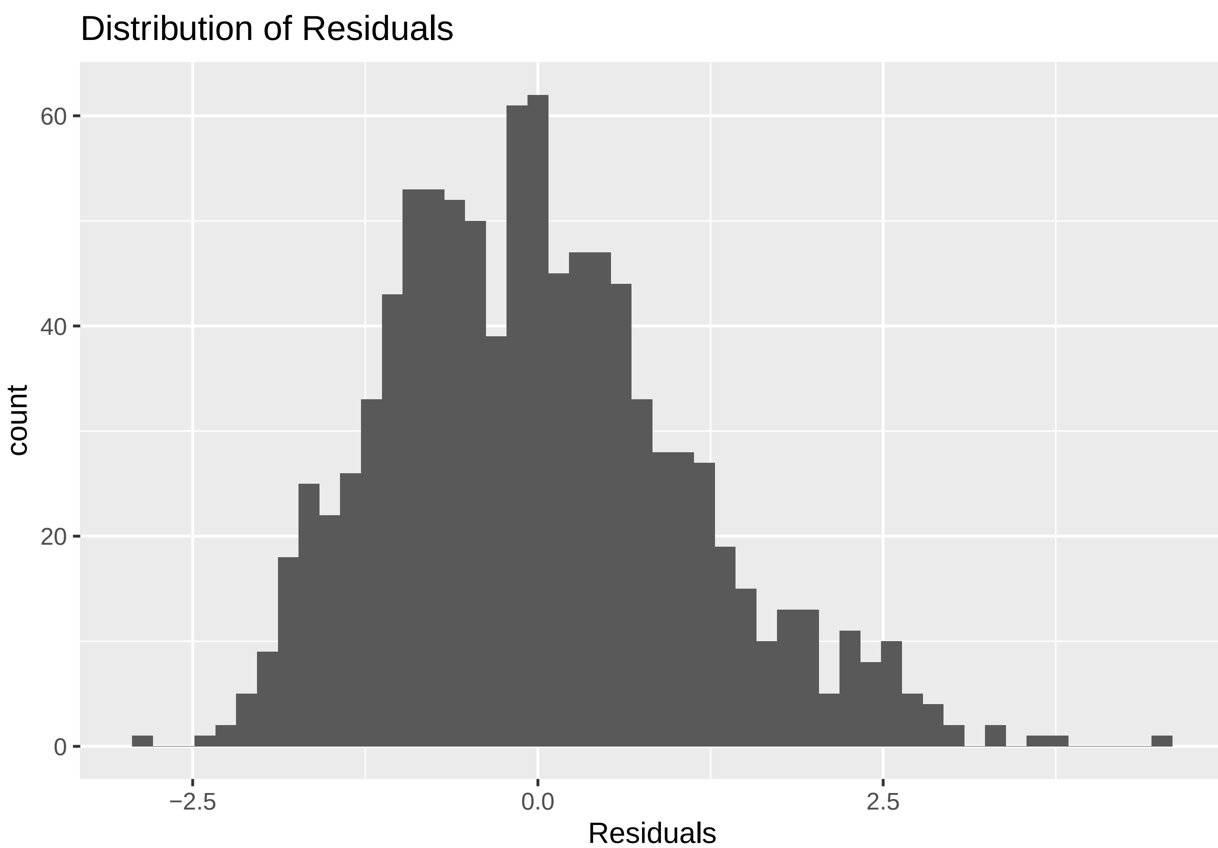
Checking Assumptions

Using our full model, we will check the assumptions for our model.

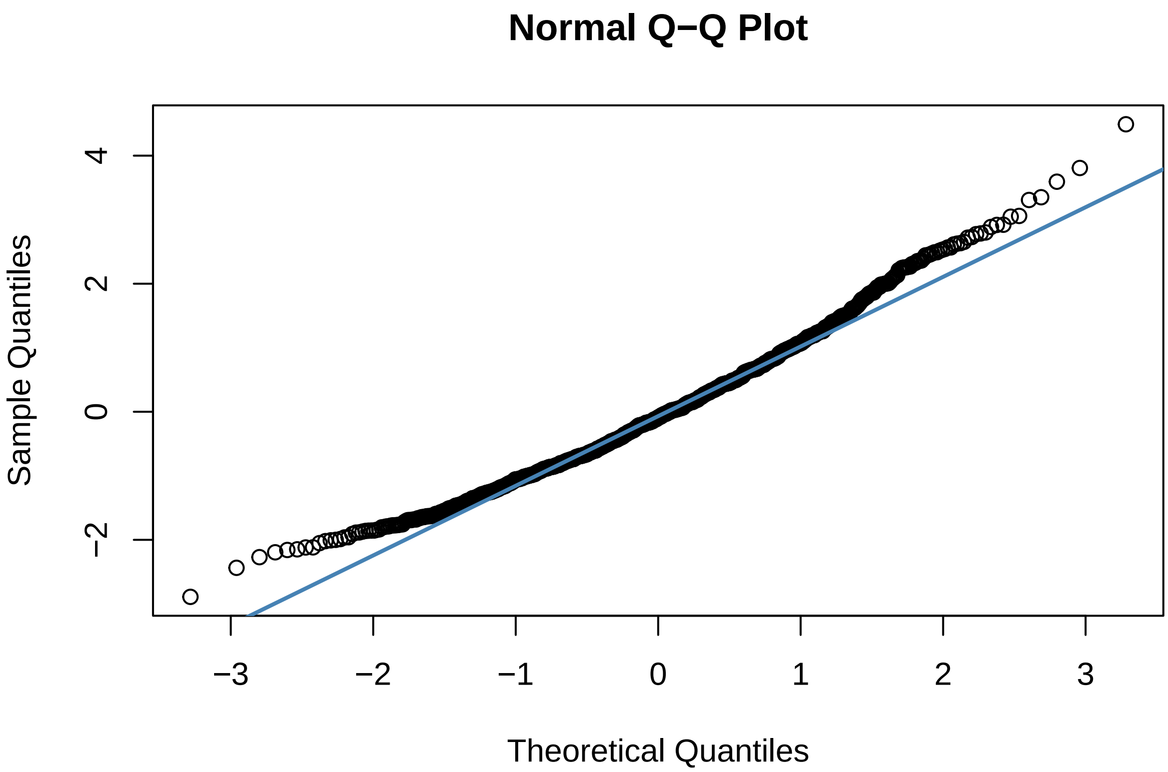
```
final_model_aug <- augment(final_model) %>%
  mutate(obs_num = 1:n())
```

We will start by looking at Normality.

```
ggplot(data = final_model_aug, aes(x =.resid)) +
  geom_histogram(bins = 50)+
  labs(x = 'Residuals', title = 'Distribution of Residuals')
```



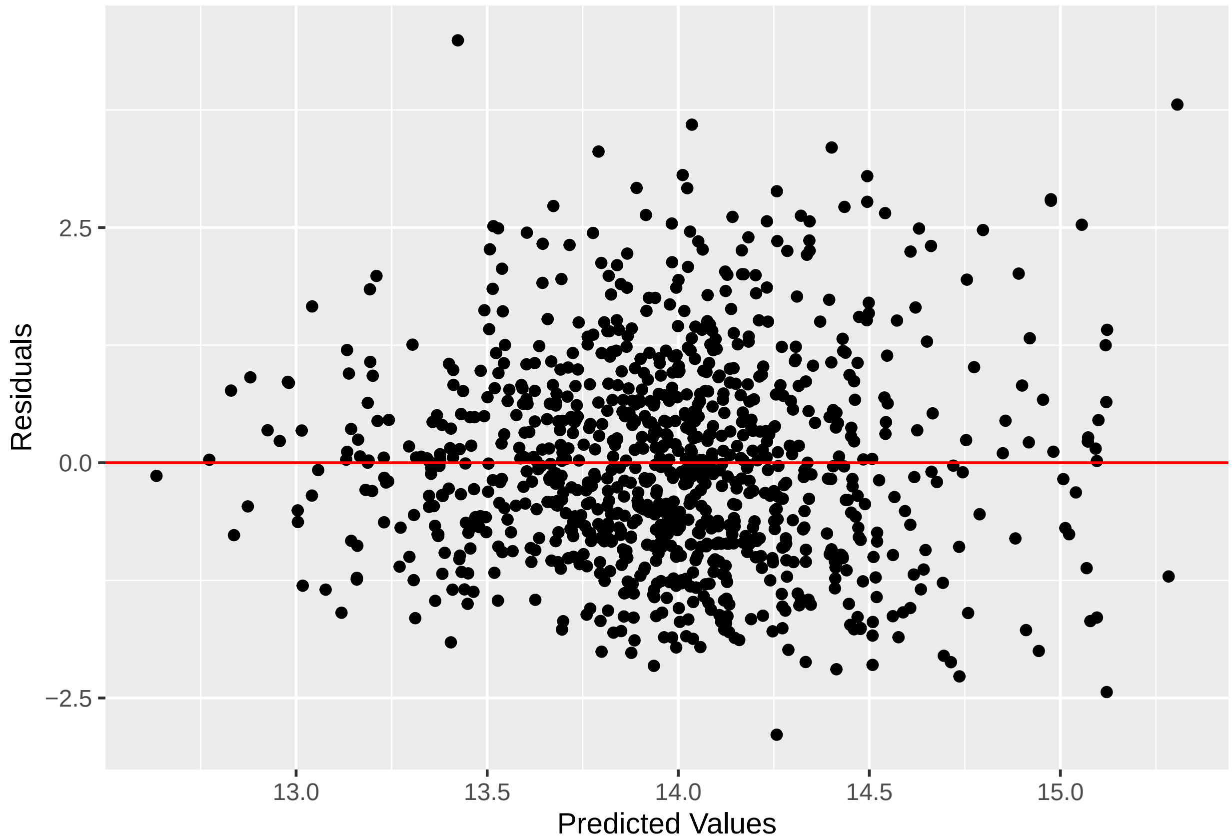

```
qqnorm(final_model_aug$.resid)
qqline(final_model_aug$.resid, col = "steelblue", lwd = 2)
```



Looking at the histogram, it looks like the residuals follow a fairly normal shape. There is symmetry but it appears to be somewhat bimodal. Also the tails look heavier. Looking at the QQ Plot, it looks like our sample quantiles match the theoretical quantiles around near the center of the data. But at the tails the quantiles do not match. Due to the bimodality and the lack of normality in the tails, we would claim that the normality condition has not been satisfied.

Next we will investigate the constant variance assumption.

```
ggplot(data = final_model_aug, aes(x =.fitted, y=.resid)) +
  geom_point()+
  geom_hline(yintercept = 0, color = 'red') +
  labs(x = 'Predicted Values', y = 'Residuals')
```

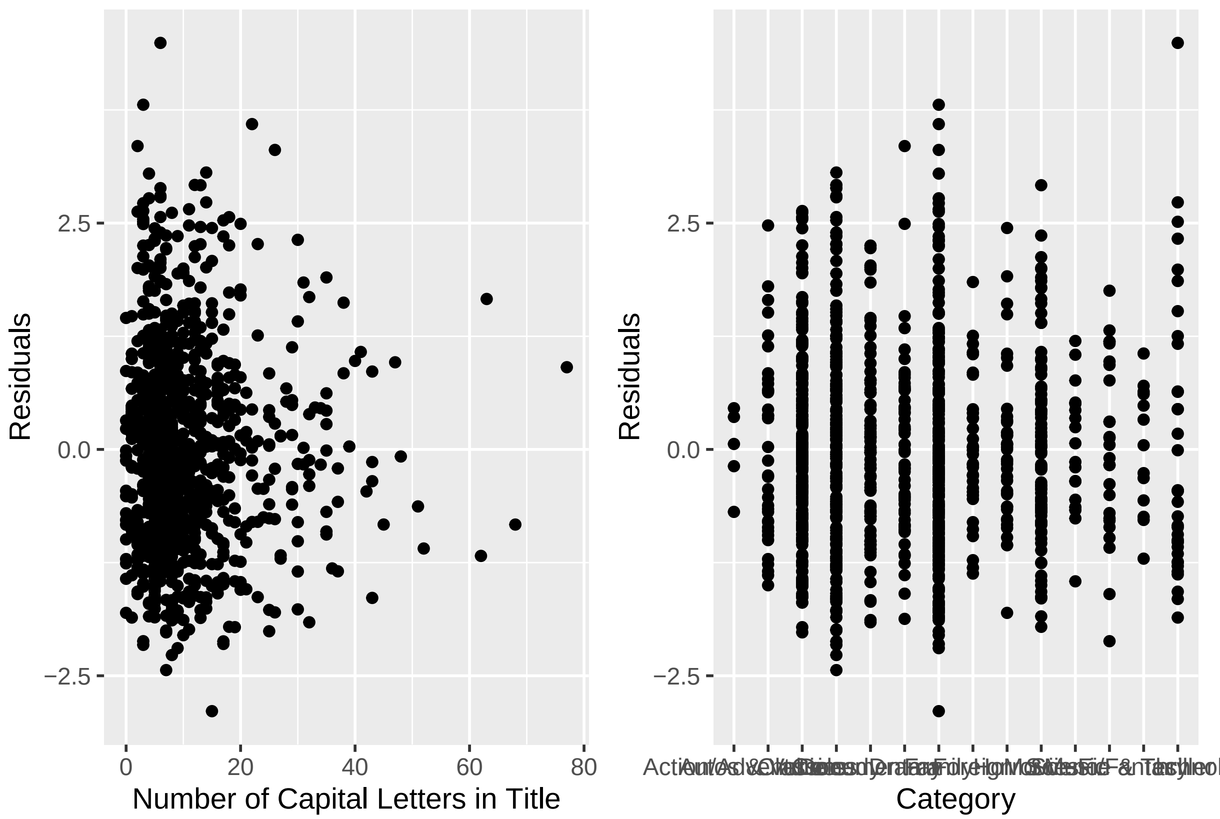


Looking at the residuals versus the predicted, it does not look like a random scatter about 0. Instead there is very little variability for low predicted values. Meanwhile the variability with our middle to high predicted values is quite high. Hence the constant variance assumption is not satisfied.

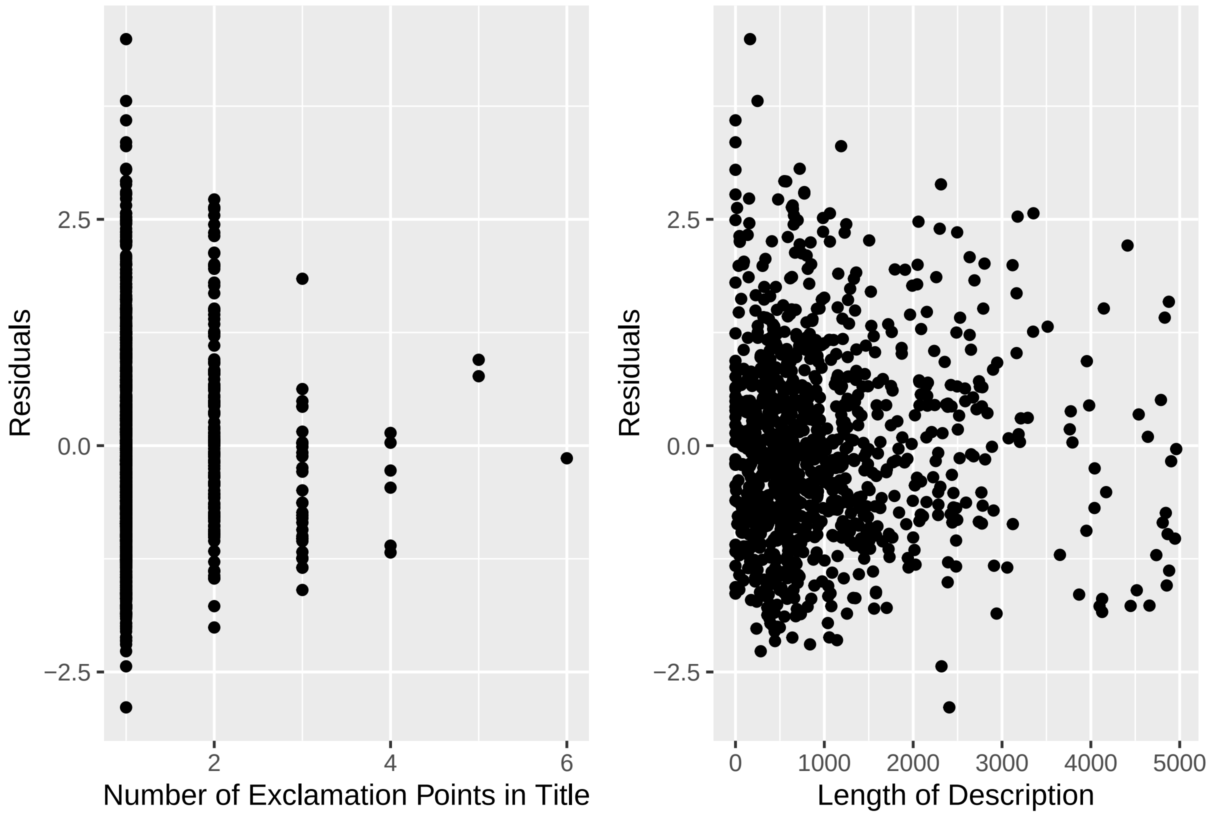
Next we will investigate linearity. We can look at same plot as above to see that there isn't much of a pattern (besides lack of constant variance).

```
a1 <- ggplot(data = final_model_aug, aes(x = num_caps, y=.resid)) +
  geom_point()+
  labs(x = 'Number of Capital Letters in Title', y = 'Residuals')

a2 <- ggplot(data = final_model_aug, aes(x = category, y=.resid)) +
  geom_point()+
  labs(x = 'Category', y = 'Residuals')

a3 <- ggplot(data = final_model_aug, aes(x = num_exc, y=.resid)) +
  geom_point()+
  labs(x = 'Number of Exclamation Points in Title', y = 'Residuals')

a4 <- ggplot(data = final_model_aug, aes(x = desc_length, y=.resid)) +
  geom_point()+
  labs(x = 'Length of Description', y = 'Residuals')

a5 <- ggplot(data = final_model_aug, aes(x = video_length, y=.resid)) +
  geom_point()+
  labs(x = 'Video Length', y = 'Residuals')

a6 <- ggplot(data = final_model_aug, aes(x = weekday_published, y=.resid)) +
```

```
  geom_boxplot()+
  labs(x = 'Weekday Published', y = 'Residuals')

a7 <- ggplot(data = final_model_aug, aes(x = hour_published, y=.resid)) +
  geom_boxplot()+
  labs(x = 'Hour Published', y = 'Residuals')

a1+a2
```

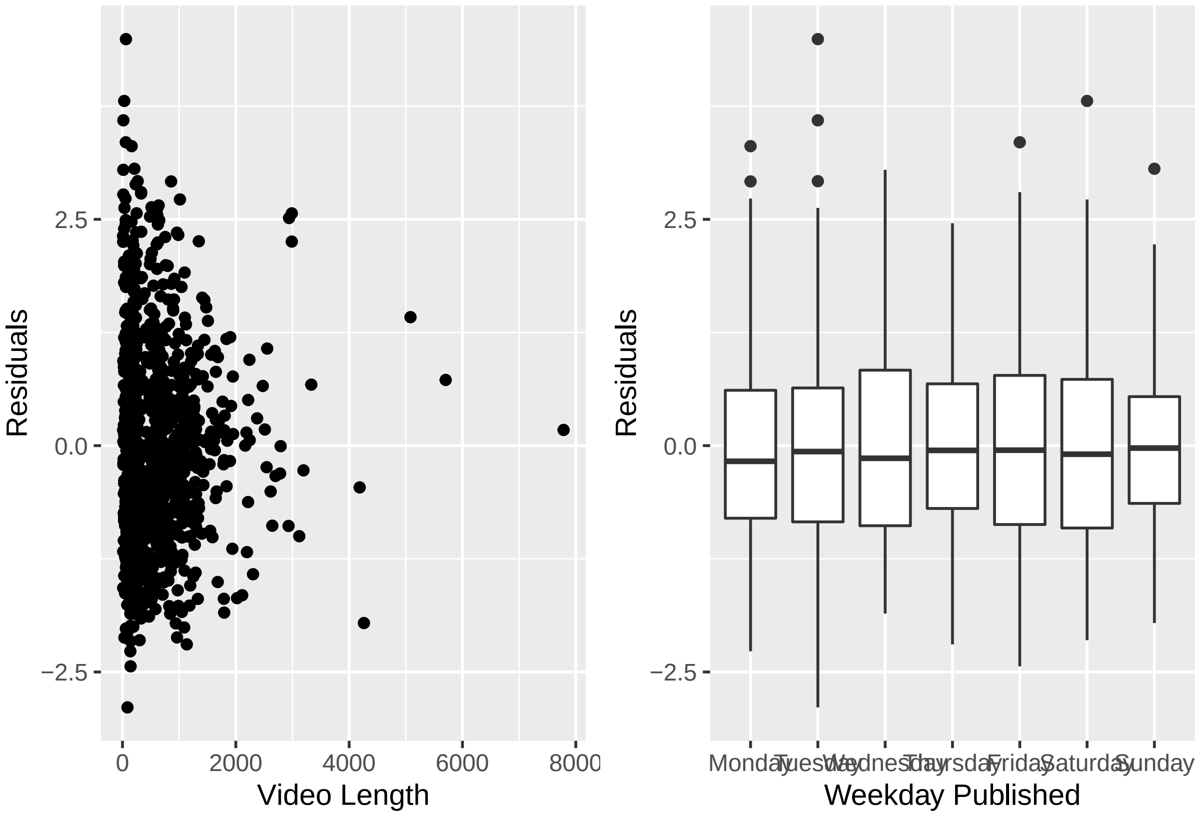



```
a3+a4
```

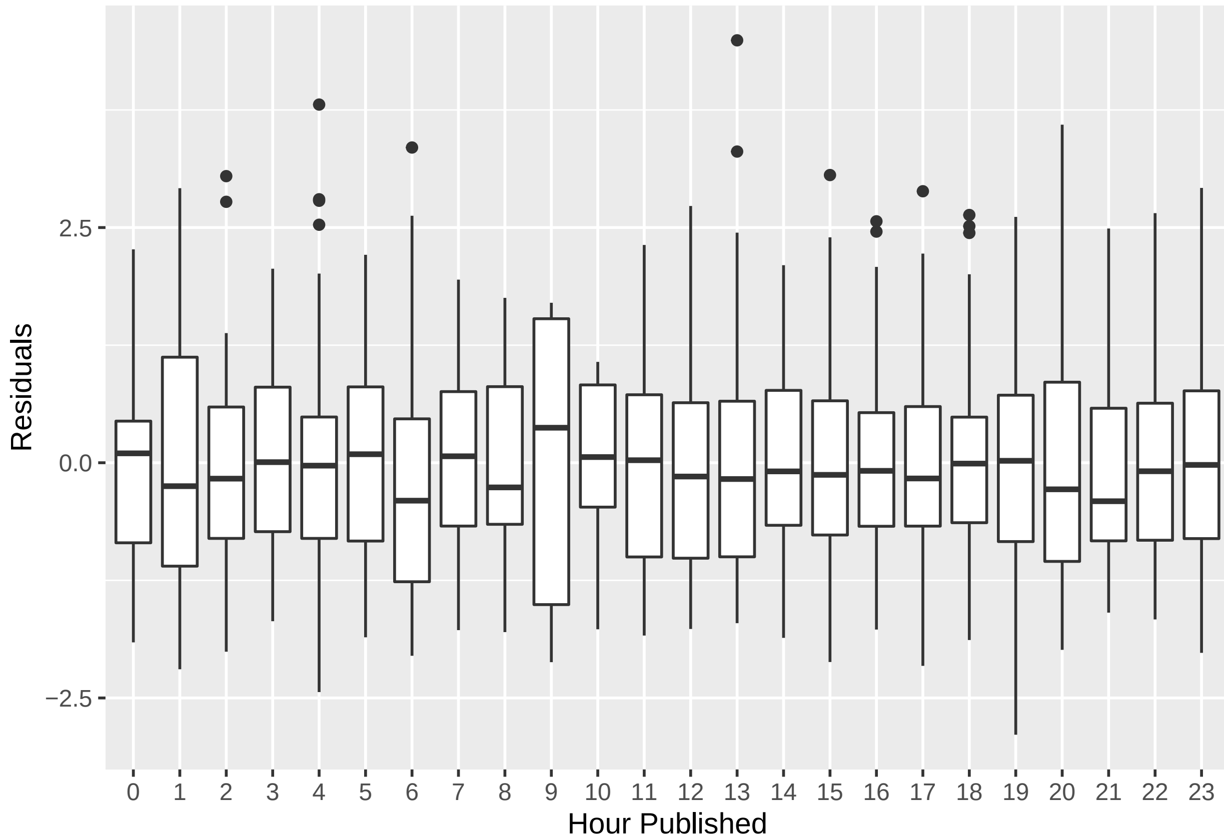


Residuals
2.5
0.0
−2.5
2
4
6
Number of Exclamation Points in Title
Residuals
2.5
0.0
−2.5
0
1000
2000
3000
4000
5000
Length of Description


```
a5+a6
```


Residuals
2.5
0.0
−2.5
0
2000
4000
6000
8000
Video Length
Residuals
2.5
0.0
−2.5
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Weekday Published

```
a7
```

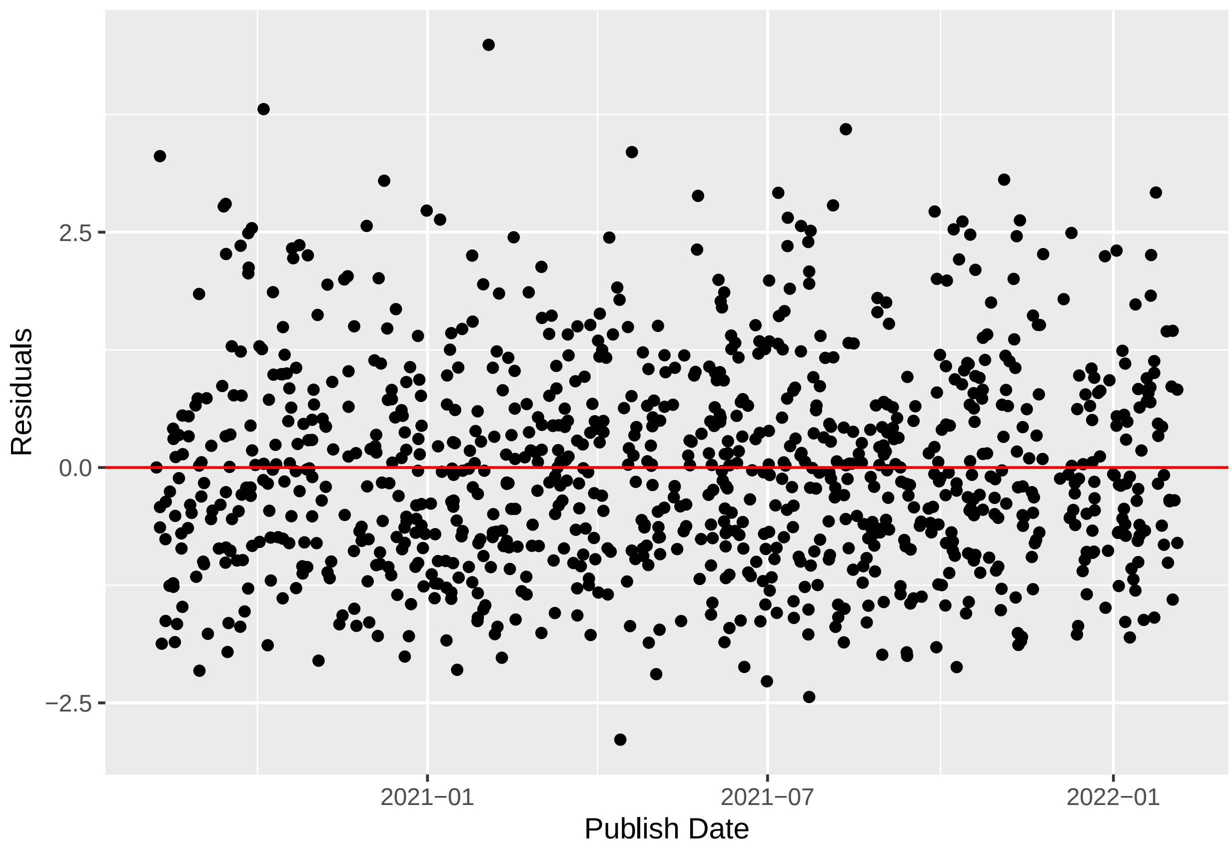


Looking at the residuals versus our predictor variables, there doesn't appear to be a discernible linear relationship between our predictor variables and the residuals. But there is a fan-like shape across many of these plots. The only concerning predictor is `hour_published` as mean residuals between groups varies somewhat. But overall they seem to stay within the same range. So it doesn't look like the linearity condition has been violated.

Now we will check independence. Since we have taken a random sample from the population dataset of Trending Youtube Videos from August 2020 to February of 2022, we would assume that the independence condition has been satisfied. To check, we will plot the residuals over time.

```
youtube_raw <- read.csv('data/youtube_data.csv') %>%
  subset(select = publishedAt) %>%
  mutate(publishedAt = as.POSIXct(publishedAt), obs_num = 1:n())


final_model_aug_ind <- merge(final_model_aug, youtube_raw, by = 'obs_num')
```

```
ggplot(data = final_model_aug_ind, aes(x = publishedAt, y = .resid))+
  geom_point() +
  geom_hline(yintercept = 0, color = 'red') +
  labs(x = 'Publish Date', y = 'Residuals')
```



Seeing as the the residuals are a random scatter about 0, then it looks like our observations are independent of one another. With this in addition to knowing the methods for gaining the data, it is enough to say that the independence assumption has been satisfied.