HW week 12

w203: Statistics for Data Science

w203 teaching team

```
library(tidyverse)
library(ggplot2)

library(lmtest)
library(sandwich)
library(stargazer)
```

```
d <- load_and_clean(input = 'videos.txt')
```

```
## Rows: 9618 Columns: 9
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (3): video_id, uploader, category
## dbl (6): age, length, views, rate, ratings, comments
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Regression analysis of YouTube dataset

You want to explain how much the quality of a video affects the number of views it receives on social media. In a world where people can now buy followers and likes, would such an investment increase the number of views that their content receives? **This is a causal question.**

You will use a dataset created by Cheng, Dale and Liu at Simon Fraser University. It includes observations about 9618 videos shared on YouTube. Please see this link for details about how the data was collected.

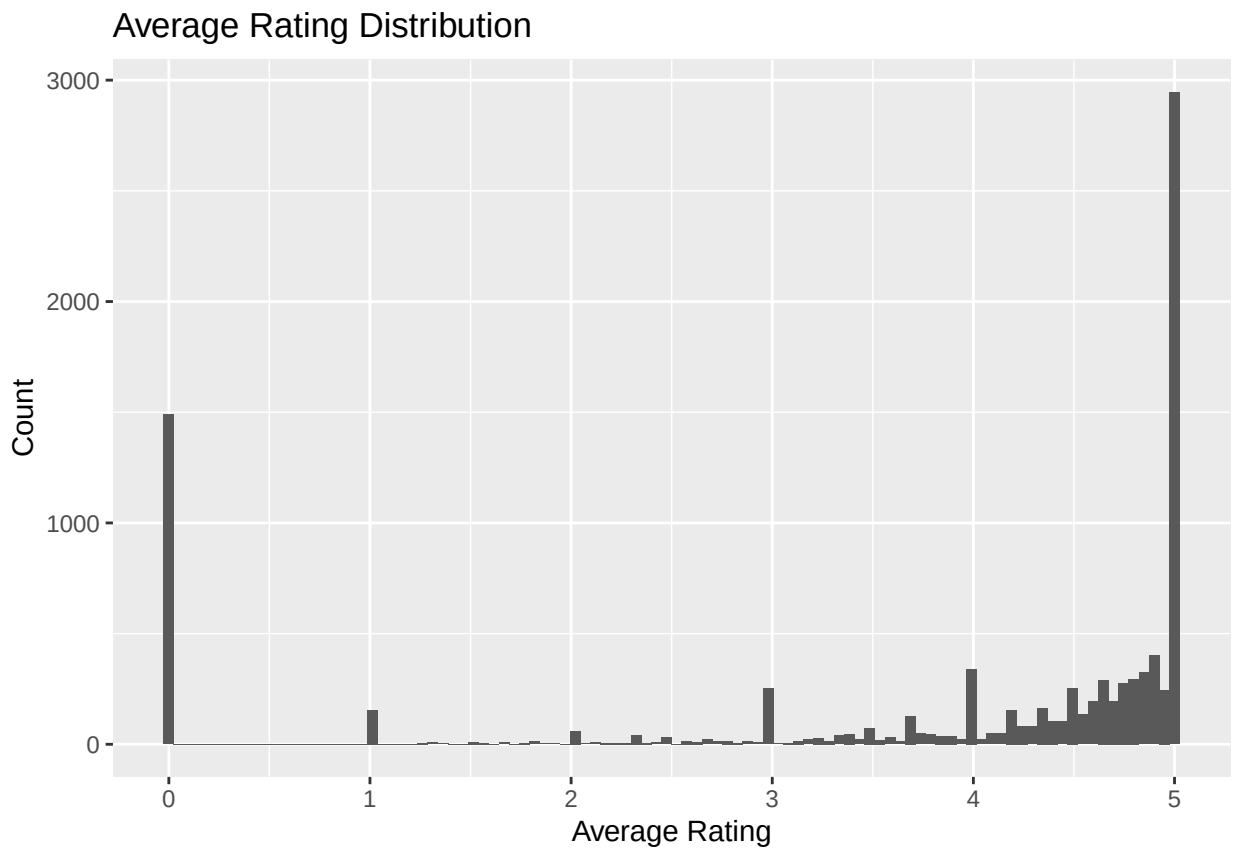
You will use the following variables:

- `views`: the number of views by YouTube users.
- `average_rating`: This is the average of the ratings that the video received, it is a renamed feature from `rate` that is provided in the original dataset. (Notice that this is different from `cout_of_ratings` which is a count of the total number of ratings that a video has received.
- `length`: the duration of the video in seconds.

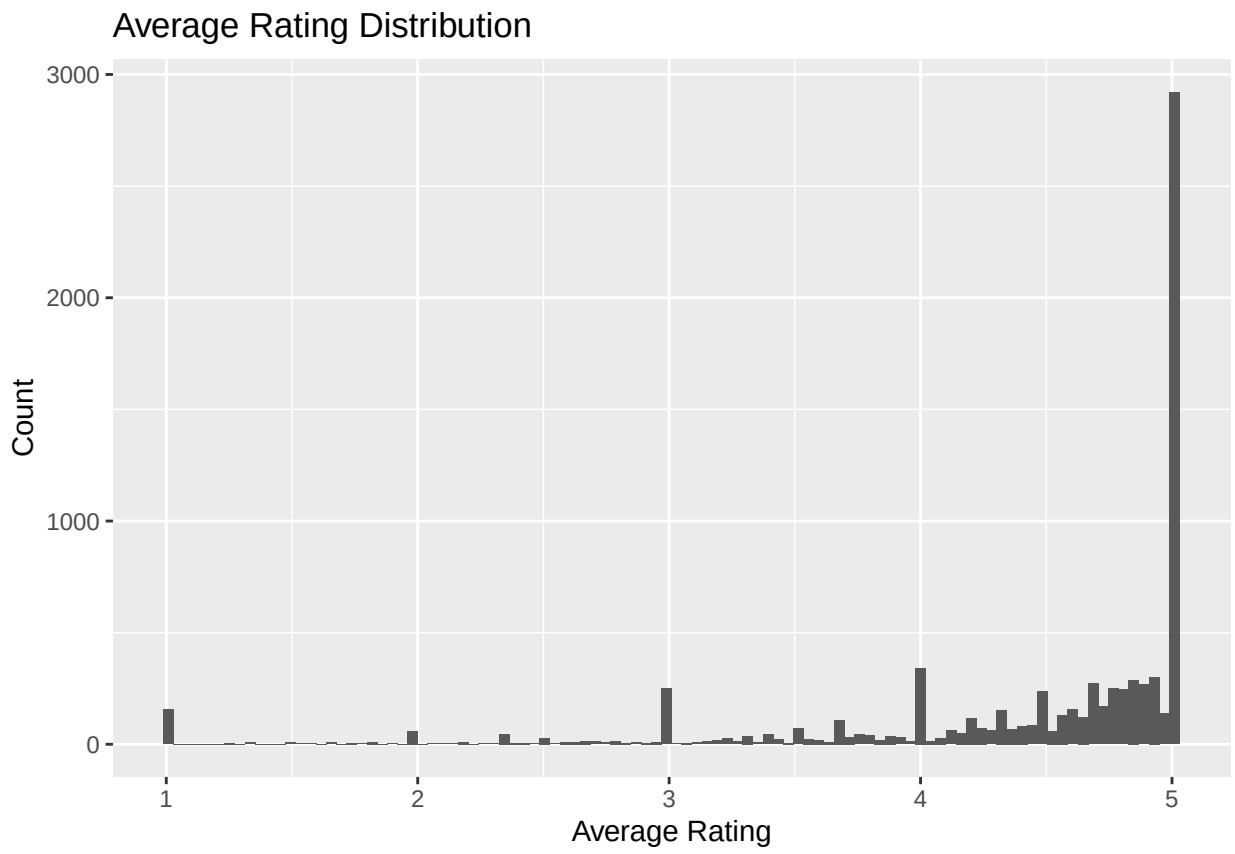
a. Perform a brief exploratory data analysis on the data to discover patterns, outliers, or wrong data entries and summarize your findings.

```
ggplot(d, aes(x=average_rating)) + geom_histogram(bins = 100) + labs(
  title = "Average Rating Distribution",
  x = "Average Rating",
  y = "Count")
```

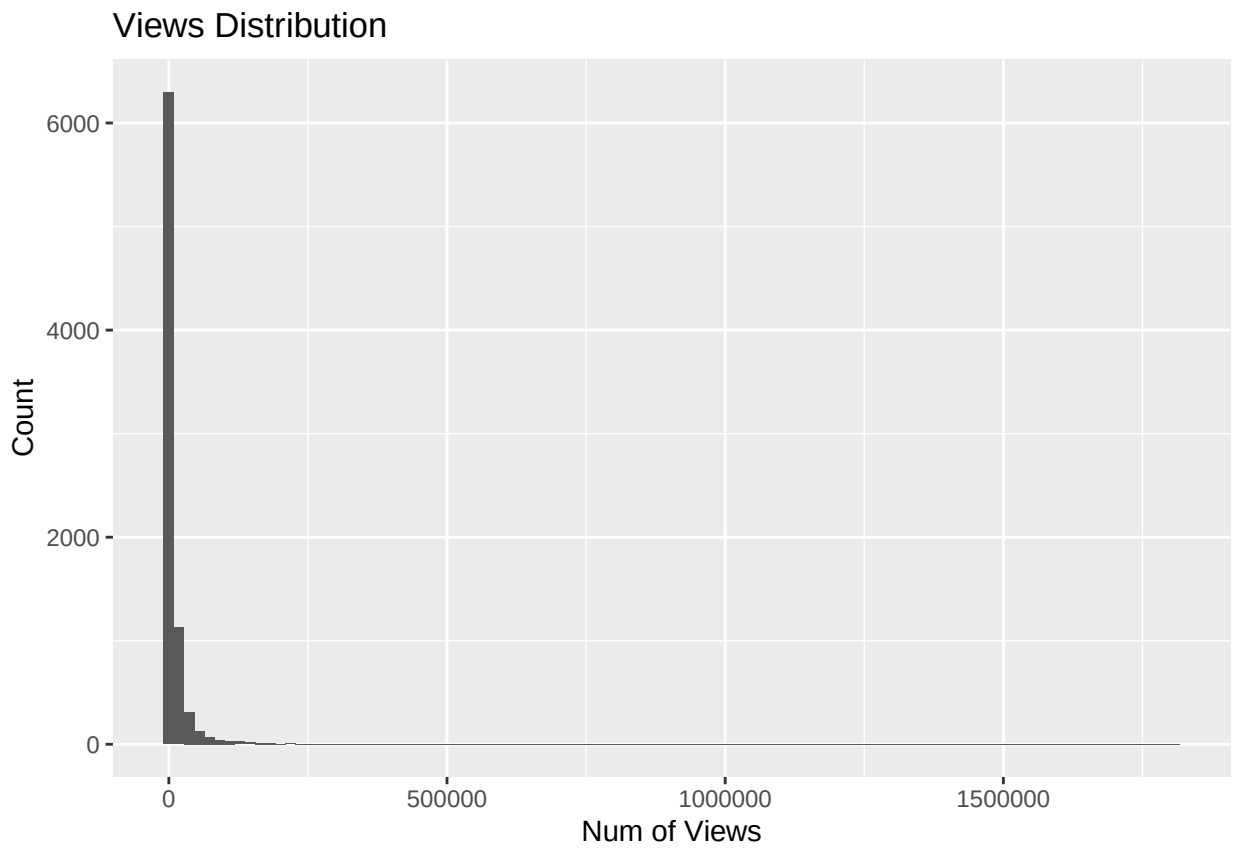
```
## Warning: Removed 9 rows containing non-finite values (stat_bin).
```



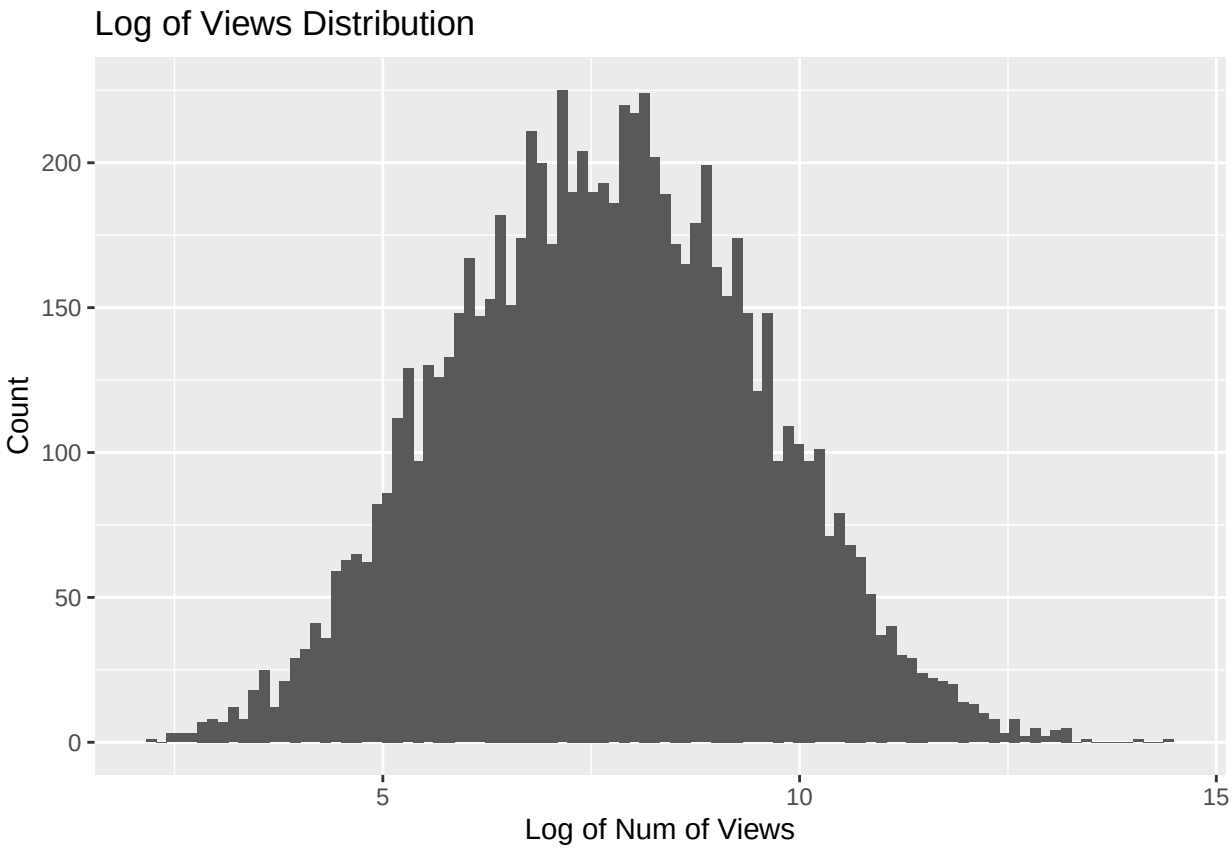

```
yt_df<-d[!(d$average_rating==0),]
yt_df<-yt_df[!(is.na(yt_df$length)),]

ggplot(yt_df, aes(x=average_rating)) + geom_histogram(bins = 100) + labs(
  title = "Average Rating Distribution",
  x = "Average Rating",
  y = "Count")
```



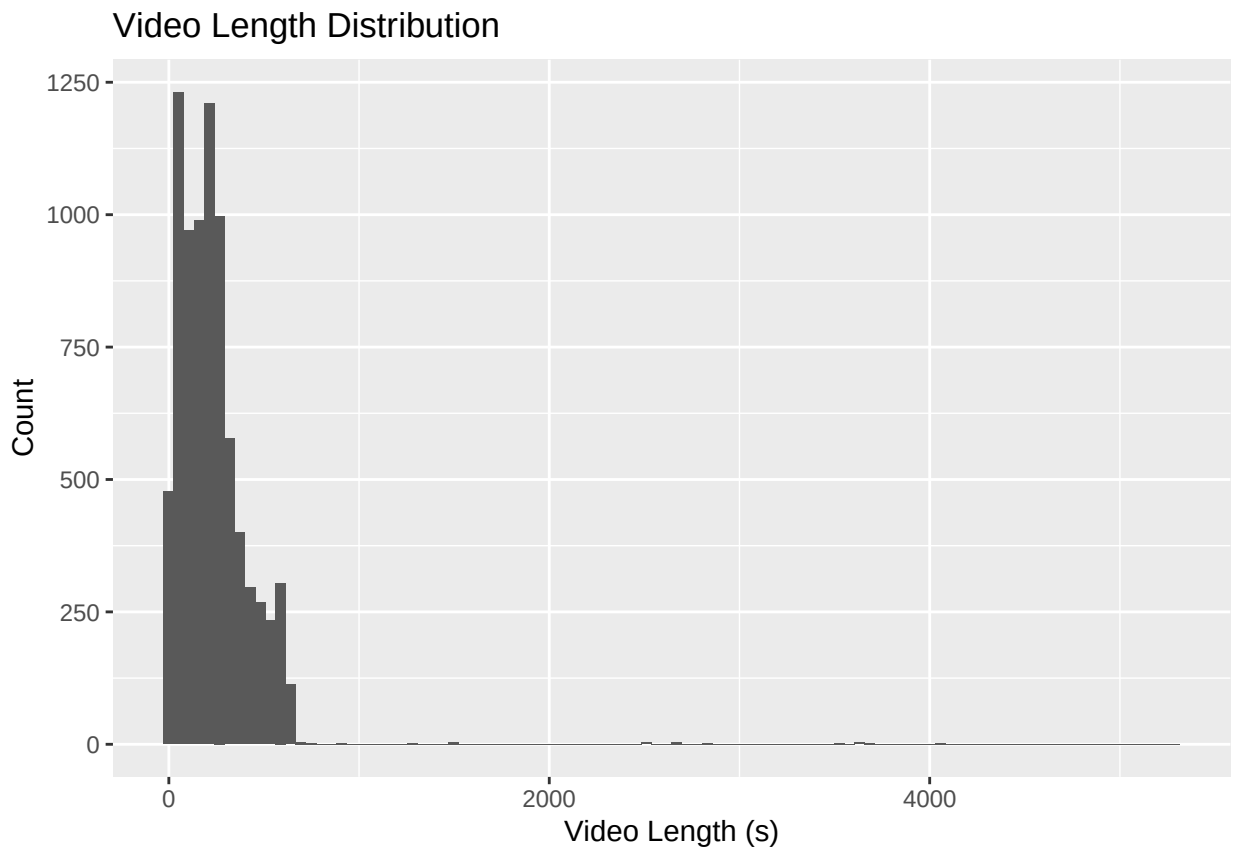

```
ggplot(yt_df, aes(x=views)) + geom_histogram(bins = 100) + labs(
  title = "Views Distribution",
  x = "Num of Views",
  y = "Count")
```



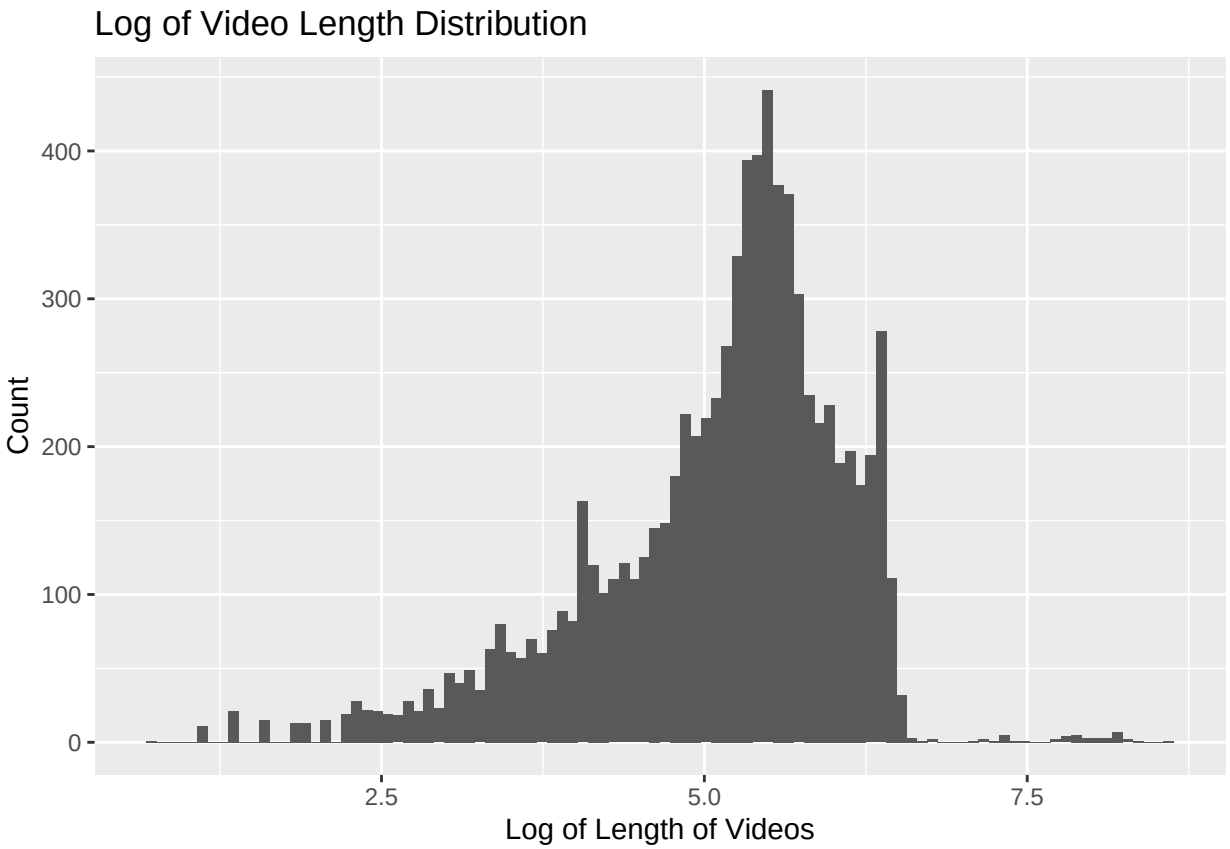

```
ggplot(yt_df, aes(x=log(views))) + geom_histogram(bins = 100) + labs(
  title = "Log of Views Distribution",
  x = "Log of Num of Views",
  y = "Count")
```




```
ggplot(yt_df, aes(x=length)) + geom_histogram(bins = 100) + labs(
  title = "Video Length Distribution",
  x = "Video Length (s)",
  y = "Count")
```




```
ggplot(yt_df, aes(x=log(length))) + geom_histogram(bins = 100) + labs(
  title = "Log of Video Length Distribution",
  x = "Log of Length of Videos",
  y = "Count")
```



A lot of entries have a 0 for average rating, this seems unusual since Youtube did not have the option to leave such a rating. If we look at the table, we'll notice that this is because it received no ratings.

It would be a good idea to drop entries without ratings, as 0 will be interpreted as being of low quality, when in reality we don't have information about its quality.

Both views and lengths have some outliers with some very high values. The logarithm of views appears to be more normally distributed than the views themselves. Also, the logarithm of the length of the videos appears to be more normally distributed than the raw length of videos, though the distribution is not entirely symmetrical.

b. Based on your EDA, select an appropriate variable transformation (if any) to apply to each of your three variables. You will fit a model of the type,

$$f(\text{views}) = \beta_0 + \beta_1 g(\text{rate}) + \beta_3 h(\text{length})$$

Where f, g and h are sensible transformations, which might include making *no* transformation.

```
model <- lm(log(views) ~ average_rating + log(length), data=yt_df)

stargazer(
  model,
  type = 'text',
  se = list(get_robust_se(model))
)
```

```
##
## ===============================================
```

```
##                         Dependent variable:
##                     ---------------------------
##                              log(views)
## ------------------------------------------------
## average_rating                0.156***
##                                (0.024)
##
## log(length)                   0.150***
##                                (0.020)
##
## Constant                      6.212***
##                                (0.136)
##
## ------------------------------------------------
## Observations                    8,119
## R2                              0.014
## Adjusted R2                     0.014
## Residual Std. Error       1.830 (df = 8116)
## F Statistic           58.190*** (df = 2; 8116)
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

c. Using diagnostic plots, background knowledge, and statistical tests, assess all five assumptions of the CLM. When an assumption is violated, state what response you will take. As part of this process, you should decide what transformation (if any) to apply to each variable. Iterate against your model until your satisfied that at least four of the five assumption have been reasonably addressed.

1. **IID Data:** "IID assumes that the data was gathered in a truly random sampling method, with chosen videos in the sample being truly independent of one another, and an even distribution of the videos across all of the population, Youtube. However, in the publication notes that describe how the dataset was gathered, the researchers describe that the data was initiated from these curated lists on Youtube: "Recently Featured", "Most Viewed", "Top Rated" and "Most Discussed", for "Today", "This Week", "This Month" and "All Time". This in itself means that videos that are more popular are often used as the "seeds" in the crawler algorithm. On top of this, the crawler uses the "related videos" of the starting set to determine where to branch out to, and compounds on this process 3-5 times to gather the entire dataset. Moreover, number of views on a video in a specific genre or interest area can influence views on other videos since they may get recommended more frequently. This whole process is conducted in multiple drawings from Feb 2007 to Sep 2008. Given that the videos are "related" in content, that directly means they are not independent from one another. Moreover, the starting point being the curated lists means that they may not be an accurately dispersed representation of all the videos on youtube. Our personal assessment of the data is thus is that it is not truly IID, however given the large sample size of the data, and the incorporation of multiple dates of data pulls, the net effect of this may not be consequential. Our ideal response would be to re-design a separate web crawler and re-run the program, but it may not be possible to effectively guarantee IID of the dataset through another pull."

2. **No Perfect Colinearity:** "The No Perfect Colinearity assumption states that no set of independent variables used for the model can be directly "explained" by each other, with a pure/exact linear relationship between them. Of the variables used in the model, rate and length of video, neither of these are colinear. One way to know if this were not the case is that R would discard one of the variables in the regression performed above, because it won't be possible to invert the matrix of regressors. We also don't see any indicators of Near Perfect Colinearity. If we get the correlation for average_rating with both length and log of

length, we see that they have some correlation but nowhere near enough to indicate there is some near perfect colinearity. ”

```
cor(yt_df$average_rating, yt_df$length)
```

```
## [1] 0.1108145
```

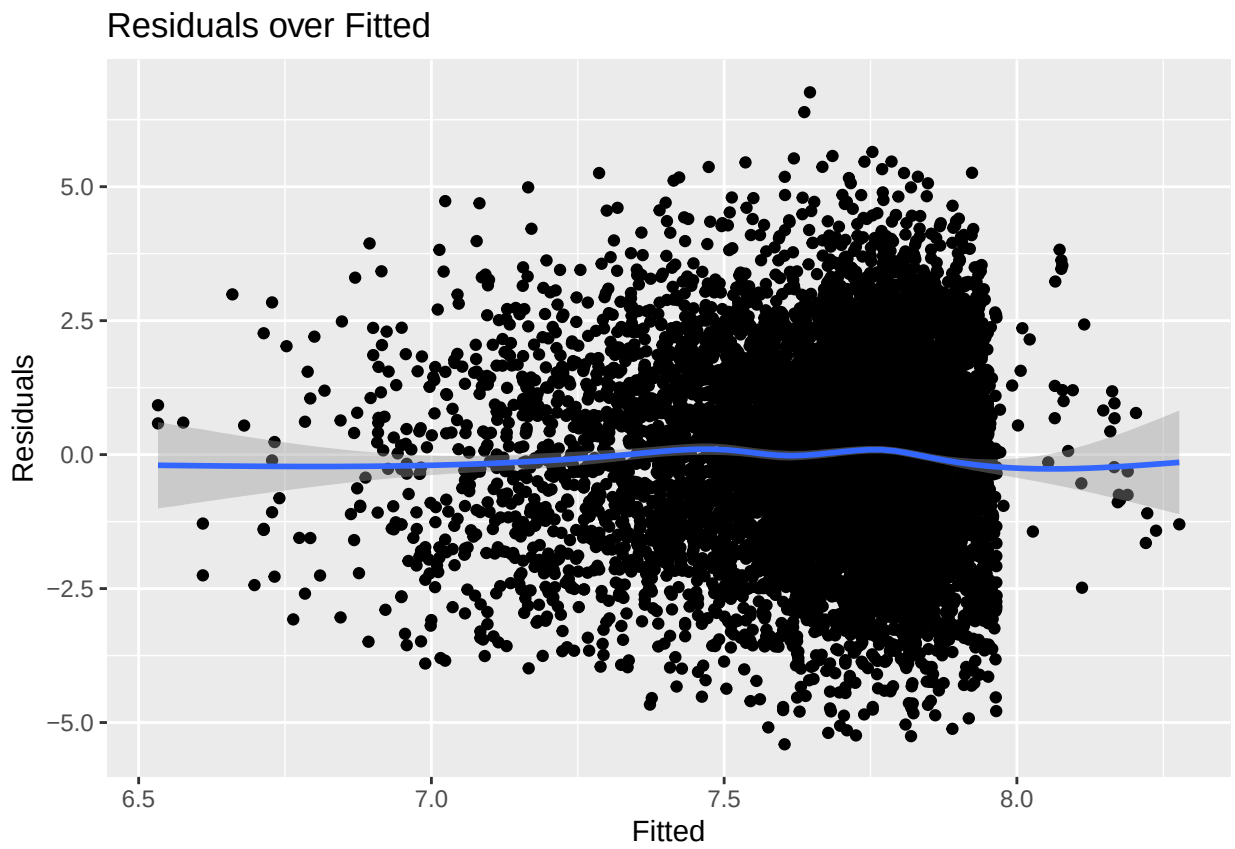
```
cor(yt_df$average_rating, log(yt_df$length))
```

```
## [1] 0.1940562
```

3. **Linear Conditional Expectation:** When we're looking to assess linear conditional expetation we want to look at the range of predicted values and errors or residuals. The model we chose does satisfy this assumption because we see a linear relationship in the plot. There is a fairly straight line near zero. But this same plot gives us some info about the next assumption on Homoskedastic Errors. If we look each of the variables against the residuals, we still see them as mmostly straight lines near zero, with averge rating being slightly less straight, but still surrounding zero. It does not seem necessary to do any transformations.

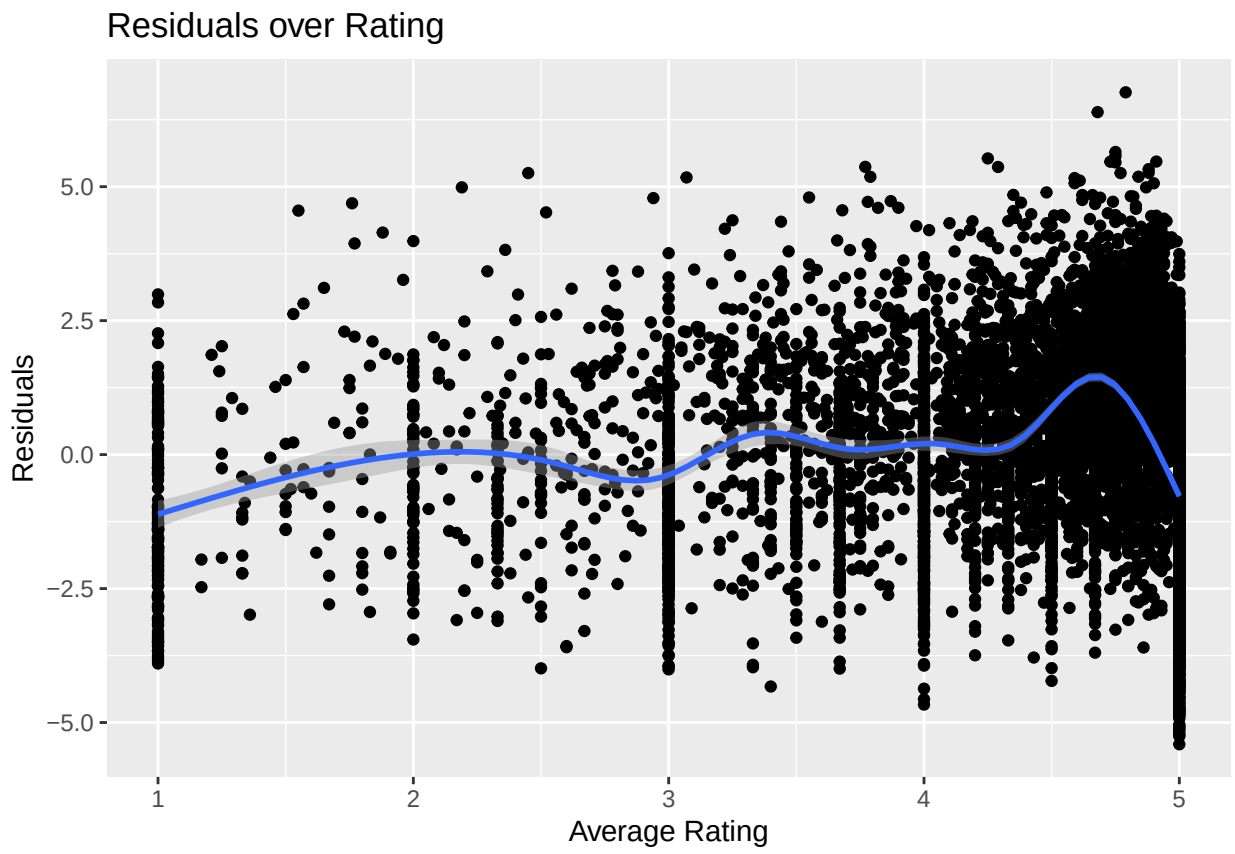
```
yt_df <- yt_df %>%
  mutate(
    model_predictions = predict(model),
    model_residuals   = resid(model)
    )

plot_model_fitted <- yt_df %>%
  ggplot(aes(x = model_predictions, y = model_residuals)) +
  geom_point() + stat_smooth() + labs(
  title = "Residuals over Fitted",
  x = "Fitted",
  y = "Residuals")

plot_model_rating <- yt_df %>%
  ggplot(aes(x = average_rating, y = model_residuals)) +
  geom_point() + stat_smooth() + labs(
  title = "Residuals over Rating",
  x = "Average Rating",
  y = "Residuals")

plot_model_length <- yt_df %>%
  ggplot(aes(x = log(length), y = model_residuals)) +
  geom_point() + stat_smooth() + labs(
  title = "Residuals over Log of Length",
  x = "Log of Video Length",
  y = "Residuals")

plot_model_fitted
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



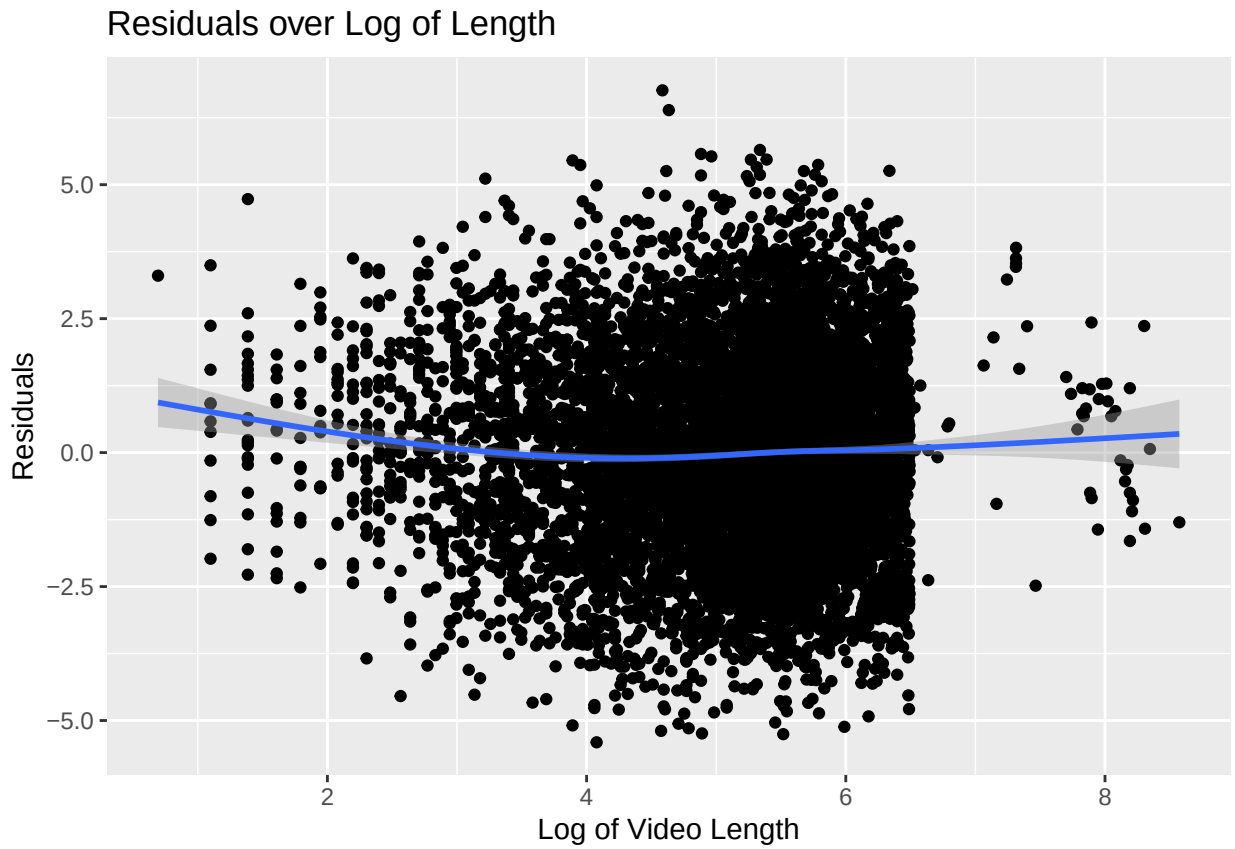

```
plot_model_rating
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```




```
plot_model_length
```

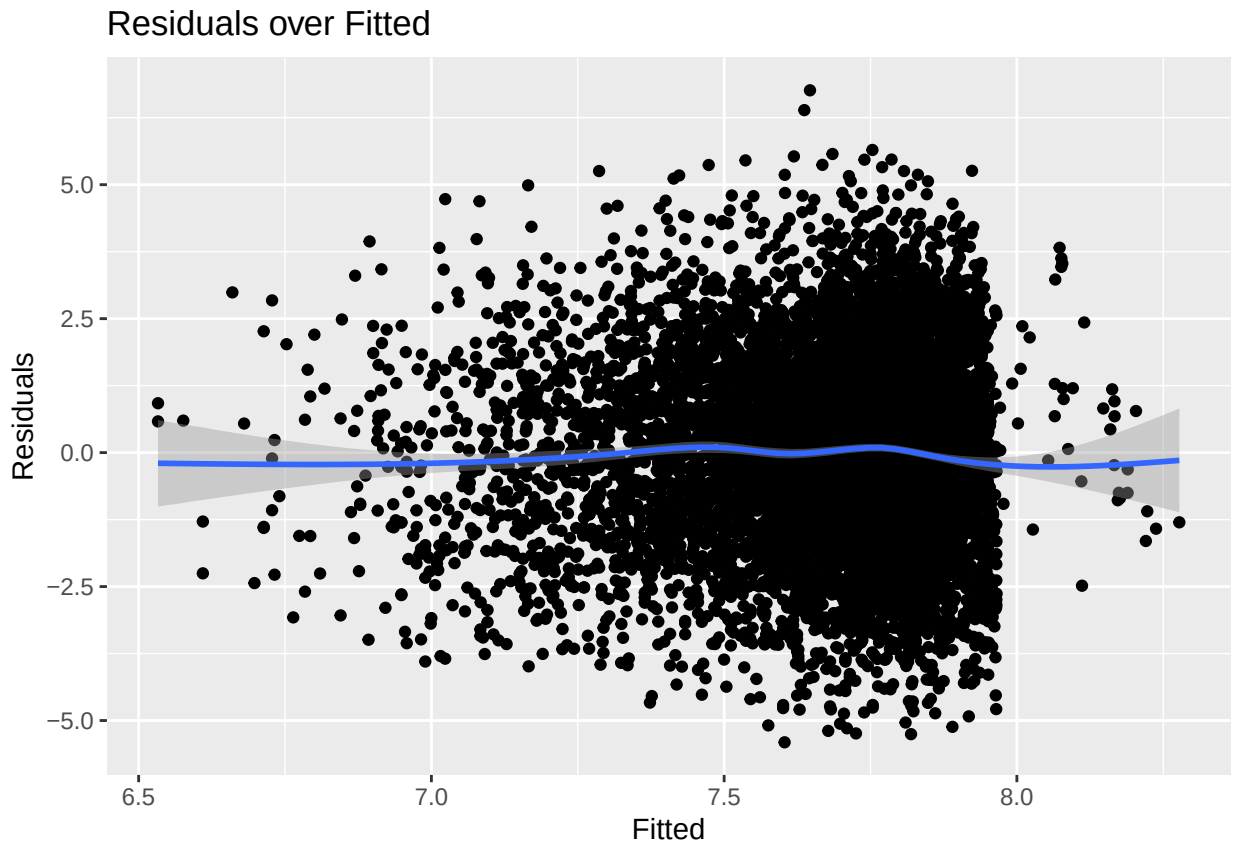
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



4. **Homoskedastic Errors:** We're using the same fitted vs. residual values plot which shows that errors are not evenly distrbuted from left to right and there is an increase in variation of error terms as we move right on the x-axis. This means there is presence of heteroskedasticity. As for the Breusch-Pagan test, since the p-value is less than 0.05, we can reject the null hypothesis. This evidence suggests that heteroscedasticity is present in the regression model. Since we do have a large sample of well above 100 we are not as concerned with this issue since we'll be using robust standard errors that will produce correct or nominal standard errors in p-values. This may have to do with skewness that we saw in the data earlier when fitting the model.

```
modf <- fortify(model)
ggplot(modf, aes(x = .fitted, y = .resid)) + geom_point() + geom_smooth() + labs(
  title = "Residuals over Fitted",
  x = "Fitted",
  y = "Residuals")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```




```
bptest(model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 10.094, df = 2, p-value = 0.006429
```

We could also consider omitted variables as the reason for the heteroskedasticity. Adding the log of count of ratings significantly increases the R2 (and Adjusted R2). However it does not seem help with the heteroskedasticity. It seems to be shifting it to the low values. One could interpret this as count_of_ratings being a better predictor for videos with high views, while videos with low views have more variance. Breusch-Pagan test also rejects the null hypothesis, which reinforces that we still seem to have a heteroskedasticity. problem.

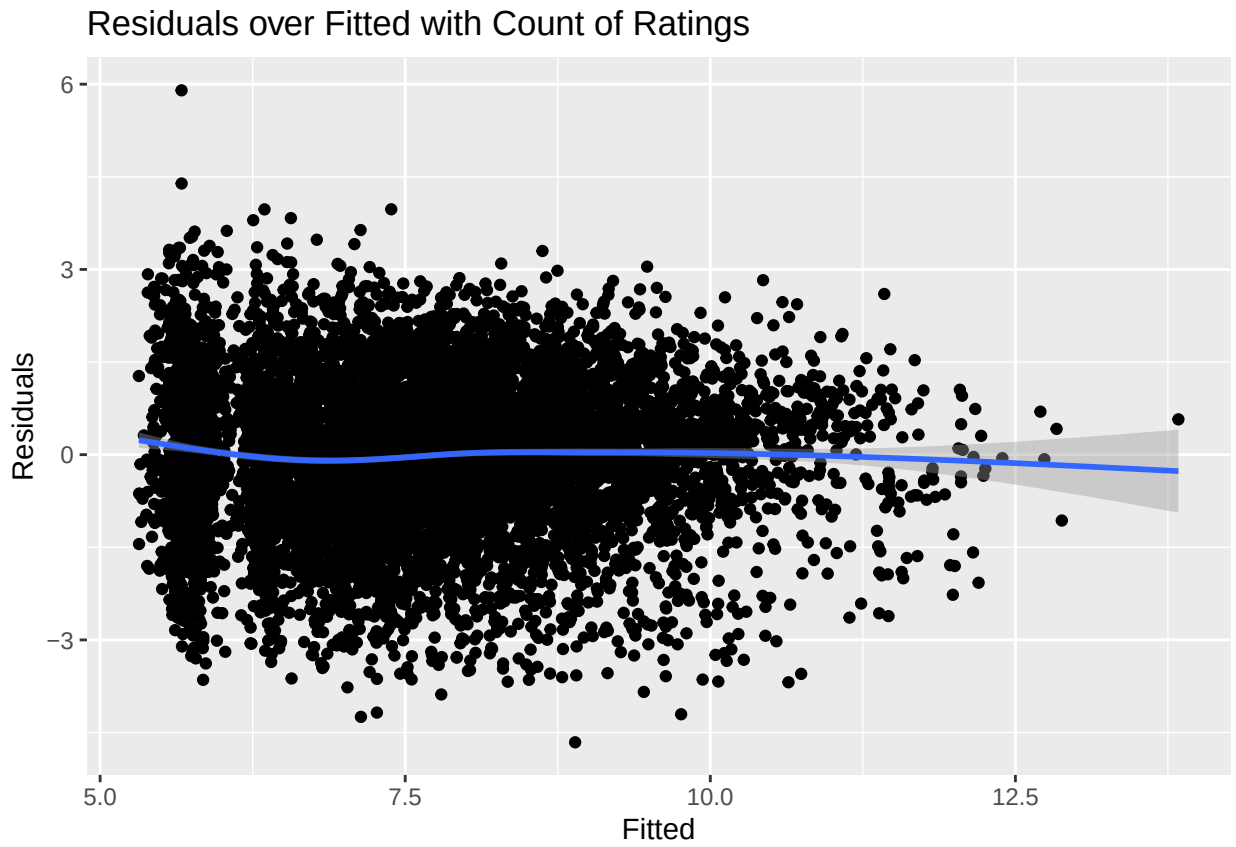
```
model_two <- lm(log(views) ~ average_rating + log(length) + log(count_of_ratings), data=yt_df)

stargazer(
  model_two,
  type = 'text',
  se = list(get_robust_se(model_two))
)
```

```
##
## ================================================
##                            Dependent variable:
##                        ---------------------------
##                                 log(views)
```

```
## ------------------------------------------------
## average_rating                  0.062***
##                                  (0.019)
##
## log(length)                     -0.120***
##                                  (0.015)
##
## log(count_of_ratings)           0.978***
##                                  (0.010)
##
## Constant                        6.025***
##                                  (0.105)
##
## ------------------------------------------------
## Observations                      8,119
## R2                                0.544
## Adjusted R2                       0.543
## Residual Std. Error         1.245 (df = 8115)
## F Statistic           3,221.096*** (df = 3; 8115)
## ================================================
## Note:                 *p<0.1; **p<0.05; ***p<0.01
```

```
modf_2 <- fortify(model_two)
ggplot(modf_2, aes(x = .fitted, y = .resid)) + geom_point() + geom_smooth() + labs(
  title = "Residuals over Fitted with Count of Ratings",
  x = "Fitted",
  y = "Residuals")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

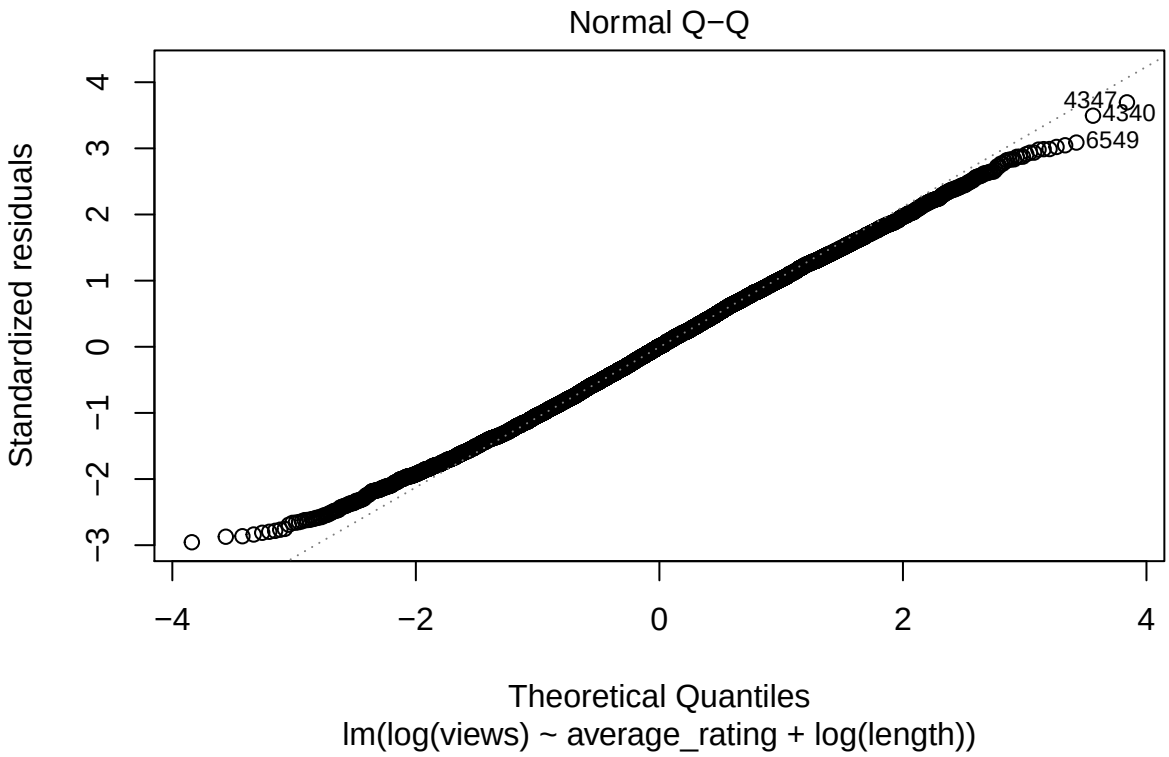



```
bptest(model_two)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model_two
## BP = 92.753, df = 3, p-value < 2.2e-16
```

5. **Normally Distributed Errors:** To check for normal distribution in errors we look at the Normal Q-Q plot, which compares the quantiles of our results against each other, in order to see if the distributions are similar. When errors are normally distrbuted we'll see a straight line, which we do see in the plot below. There is small deviation in the tails but overall we don't see serious deviations from normality. This assumption is satisfied.

```
plot(model, which=2)
```



We've gone over all 5 assumptions. The only one that could pose a problem is the lack of IID data, as explained above. We also do see that there is evidence of heteroskedasticity, but as mentioned above we do have a large sample of well above 100 so we are not as concerned with this issue since we'll be using robust standard errors that will produce correct or nominal standard errors in p-values. Having said that, I do think we should add the log(count_of_ratings) as a variable in our model, even if it doesn't fix the heteroskedasticity as it does significantly increase the adjusted R2, and is found to be a significant coefficient.