HW week 12

w203: Statistics for Data Science

w203 teaching team

```
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
library(ggplot2)

library(sandwich)
```

```
## Warning: package 'sandwich' was built under R version 4.0.5
```

```
library(stargazer)
```

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

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   video_id = col_character(),
##   uploader = col_character(),
##   age = col_double(),
##   category = col_character(),
##   length = col_double(),
##   views = col_double(),
##   rate = col_double(),
##   ratings = col_double(),
##   comments = col_double()
## )
```

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.

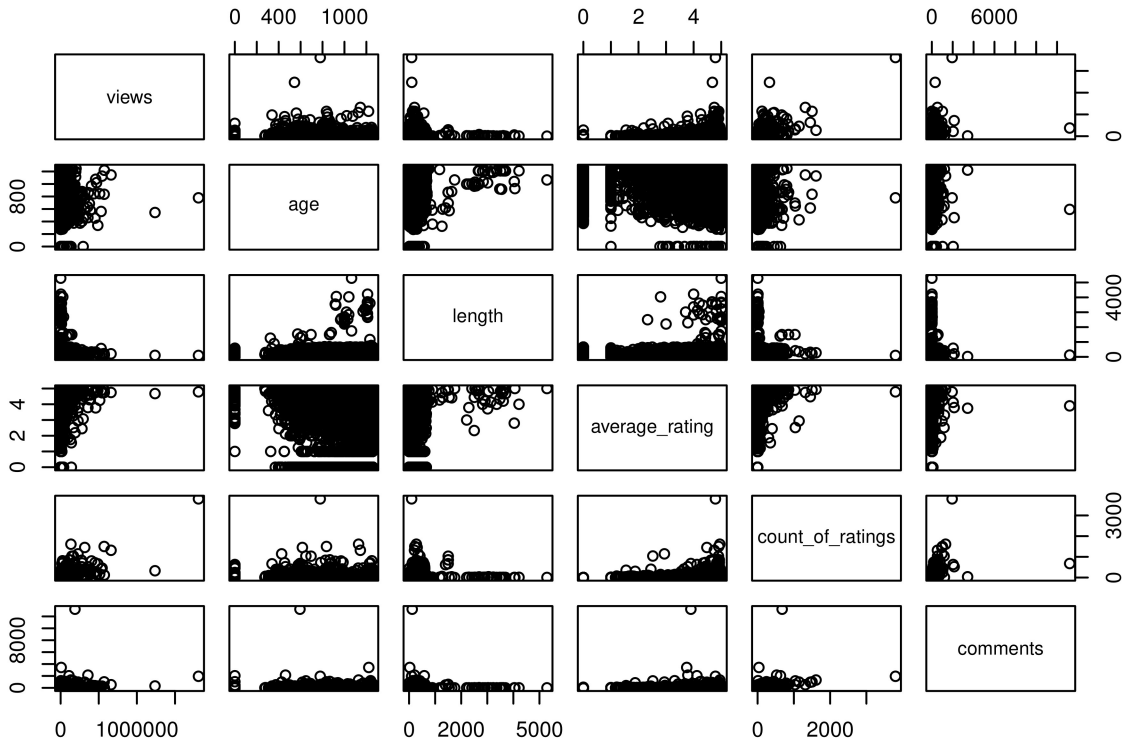
- `average_rating`: This is the average of the ratings that the video received, it is a renamed feature from `rate` that is provided in the original dataset. (Notice that this is different from `count_of_ratings` which is a count of the total number of ratings that a video has received.
- `length`: the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

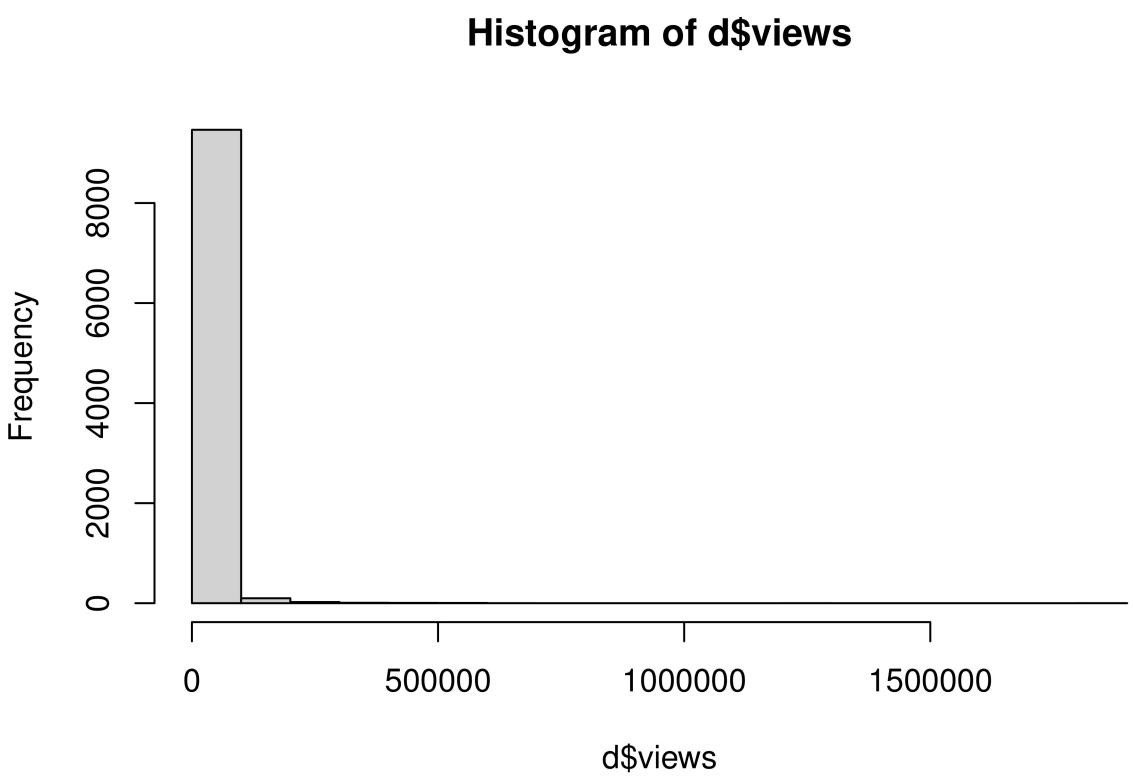
```
#The summary gives us an idea of schema as well as a general distribution
print(summary(d))
```

```
##    video_id           uploader              age         category
##  Length:9618        Length:9618        Min.   :   0   Length:9618
##  Class :character   Class :character   1st Qu.: 920   Class :character
##  Mode  :character   Mode  :character   Median :1115   Mode  :character
##                                        Mean   :1045
##                                        3rd Qu.:1226
##                                        Max.   :1258
##                                        NA's   :9
##      length          views          average_rating  count_of_ratings
##  Min.   :   1   Min.   :      3   Min.   :0.000   Min.   :   0.00
##  1st Qu.:  83   1st Qu.:    348   1st Qu.:3.400   1st Qu.:   1.00
##  Median : 193   Median :   1453   Median :4.670   Median :   5.00
##  Mean   : 227   Mean   :   9346   Mean   :3.744   Mean   :  20.66
##  3rd Qu.: 299   3rd Qu.:   6179   3rd Qu.:5.000   3rd Qu.:  15.00
##  Max.   :5289   Max.   :1807640   Max.   :5.000   Max.   :3801.00
##  NA's   :9      NA's   :9         NA's   :9       NA's   :9
##     comments        log_of_average_rating
##  Min.   :   -2.00   Min.   : -Inf
##  1st Qu.:    1.00   1st Qu.:1.224
##  Median :    3.00   Median :1.541
##  Mean   :   19.99   Mean   : -Inf
##  3rd Qu.:   13.00   3rd Qu.:1.609
##  Max.   :13211.00   Max.   :1.609
##  NA's   :9          NA's   :9
```

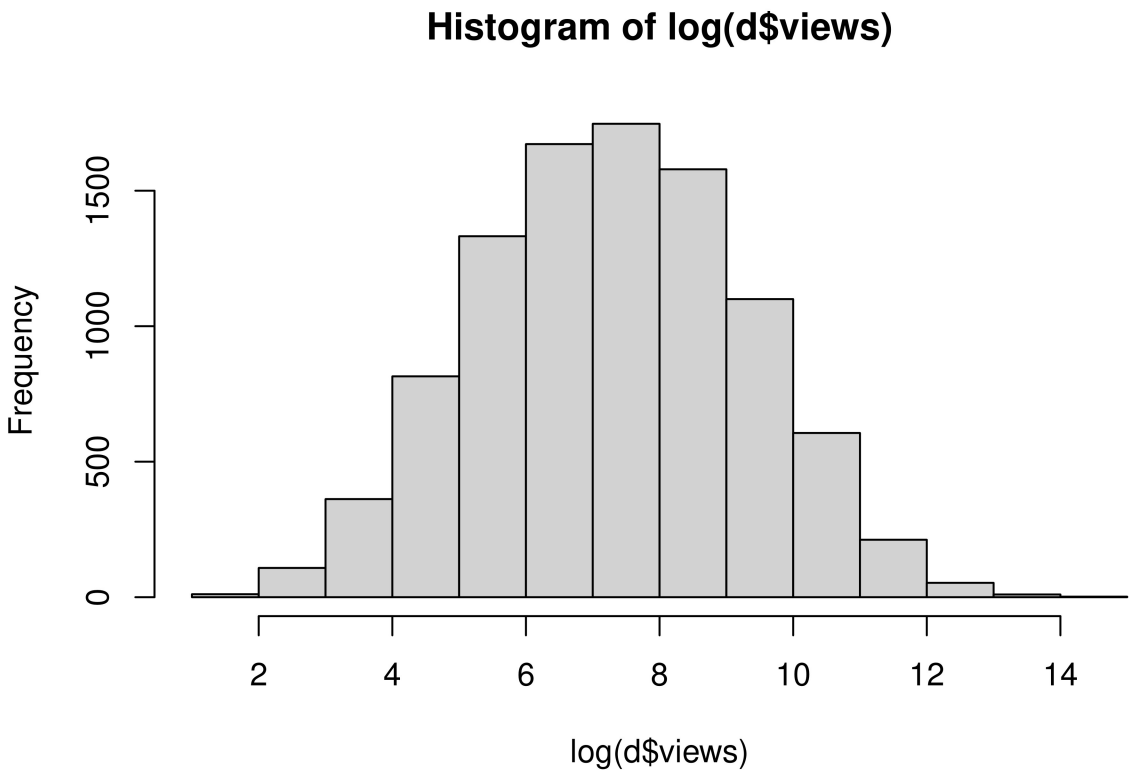
```
#Drop some character columns and columns and pre-transformed columns
#(We will do our own transformations)
d = subset(d, select = c(views, age,length, average_rating, count_of_ratings, comments))
#Exploratory scatterplots to check for relationships, potential issues with colinearity, etc
plot(d)
```




```
#Exploratory histogram of the target variable
hist(d$views)
```




```
#Exploratory histogram of transformed target variable
hist(log(d$views))
```



We can see that there are 9 observations with NAs in most columns; although this is a relatively small proportion of the overall observations and so may be dropped with the data with negligible impact on the analysis. We can also see that the scale for `views`, `length`, and `age` are substantially larger than those of other features, in addition to being highly left-skewed. The `uploader` and `category` features are both character features, and will need to be encoded if we want to use them as variables in modeling.

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.

```
#Fit model
model <- lm(log(views + 1) ~ log(average_rating + 1) + log(length+1), data = d)

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

```
##
```

```
## ================================================
##                            Dependent variable:
##                        ------------------------
##                              log(views + 1)
## ------------------------------------------------
## log(average_rating + 1)          1.377***
##                                   (0.026)
##
## log(length + 1)                  0.110***
##                                   (0.018)
##
## Constant                         4.791***
##                                   (0.088)
##
## ------------------------------------------------
## Observations                       9,609
## R2                                 0.208
## Adjusted R2                        0.208
## Residual Std. Error          1.773 (df = 9606)
## F Statistic              1,260.232*** (df = 2; 9606)
## ================================================
## Note:                    *p<0.1; **p<0.05; ***p<0.01
```

Both the `views` and `length` features should be log-transformed to improve normality of distribution, as identified in EDA.

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:** In general, because the original data is a directed graph, it can be assumed that observations may be dependent on one another (because observations are derived from graph relationships with other observations.) Additionally, the original data was collected based on lists of "most viewed", "top rated", etc. which influences potential dependent relationships between individual observations. The utilization of a breadth-first methodology was likely an effort to address this concern, although this approach is still problematic due to the influence of past observations on future observations. With this in mind, there is not enough information in the given dataset to conclude with certainty IID.

2. **No Perfect Colinearity:** Through visual observation of plots generated in EDA, it is clear that there is no perfect colinearity between features. Although `log_of_average_rating` and `average_rating` are colinear, only one of the two (`log_of_average_rating`) is used as a feature in the model. Because there are no colinear features used as modeling variables, the data meets the assumption of no perfect colinearity for the purposes of modeling.

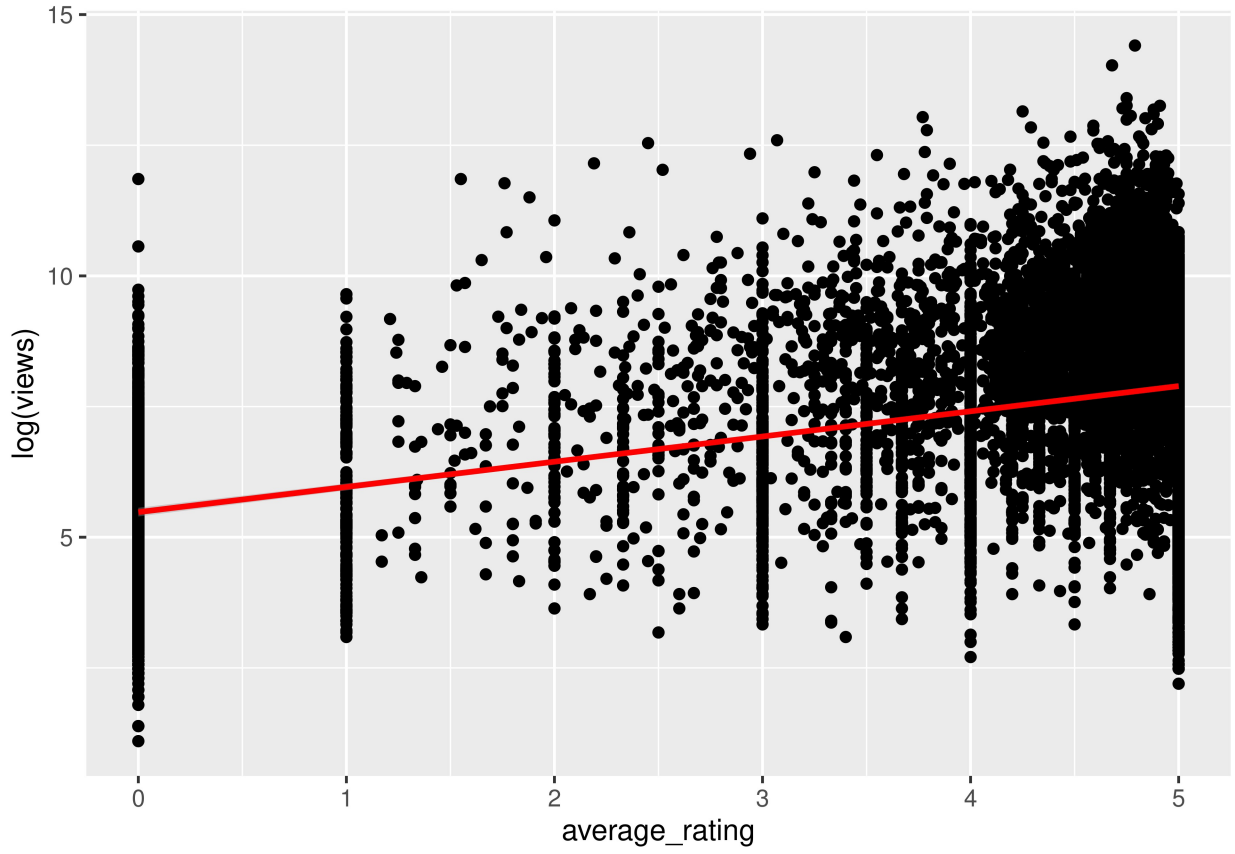
3. **Linear Conditional Expectation:** Visual observation of scatterplots deciting the relationship between target variable `views` and features `average_rating` and `log(length)` demonstrate an apparent linear correlation between the features and target variables, so the data meets the assumption of linear conditional expectation.

```r
#Basic scatterplot with smoothing line to check for possibility of linear
#conditional expectation with regards to target variable and feature
#`average_rating`
ggplot(d, aes(y = log(views), x = average_rating)) +
     geom_point() +
     stat_smooth(method = "lm", col = "red")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

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

```
## Warning: Removed 9 rows containing missing values (geom_point).
```

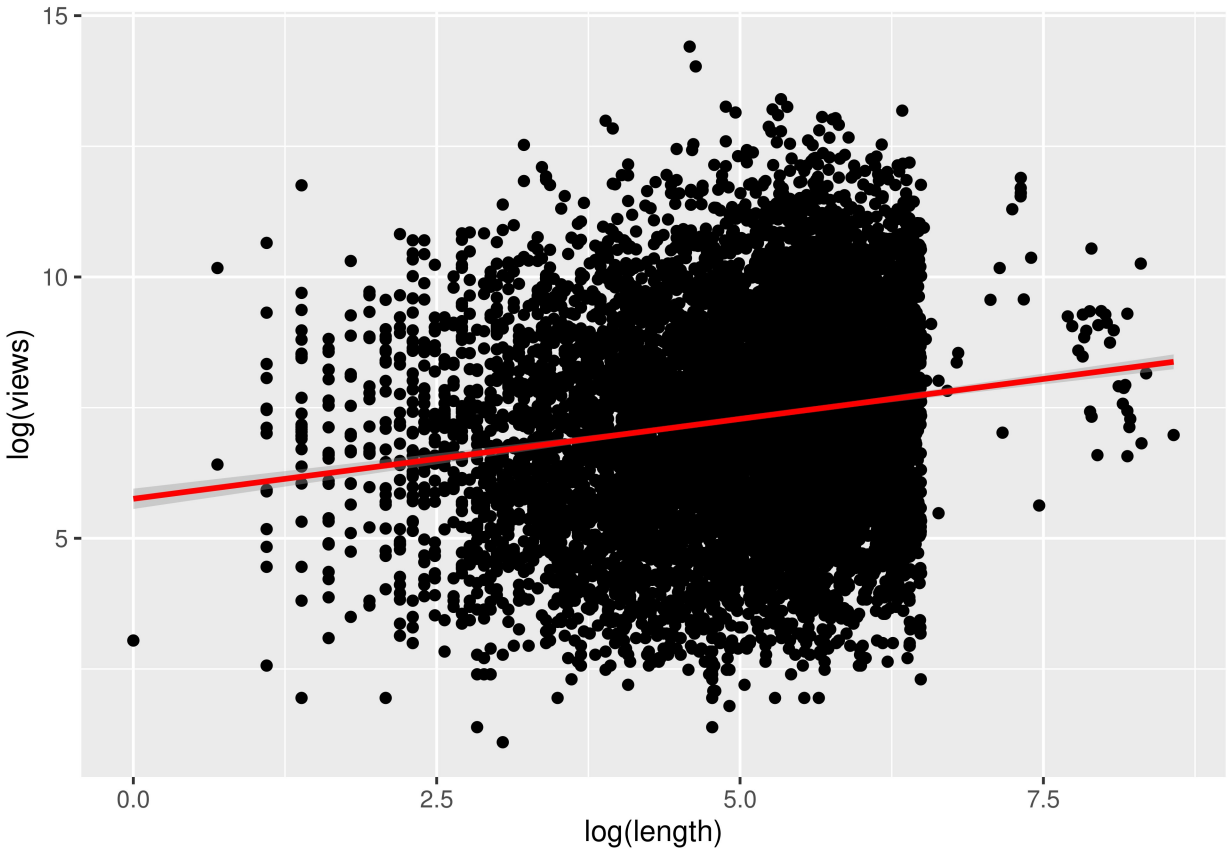



```r
#Basic scatterplot with smoothing line to check for possibility of linear
#conditional expectation with regards to target variable and feature
#`length`
ggplot(d, aes(y = log(views), x = log(length))) +
     geom_point() +
     stat_smooth(method = "lm", col = "red")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

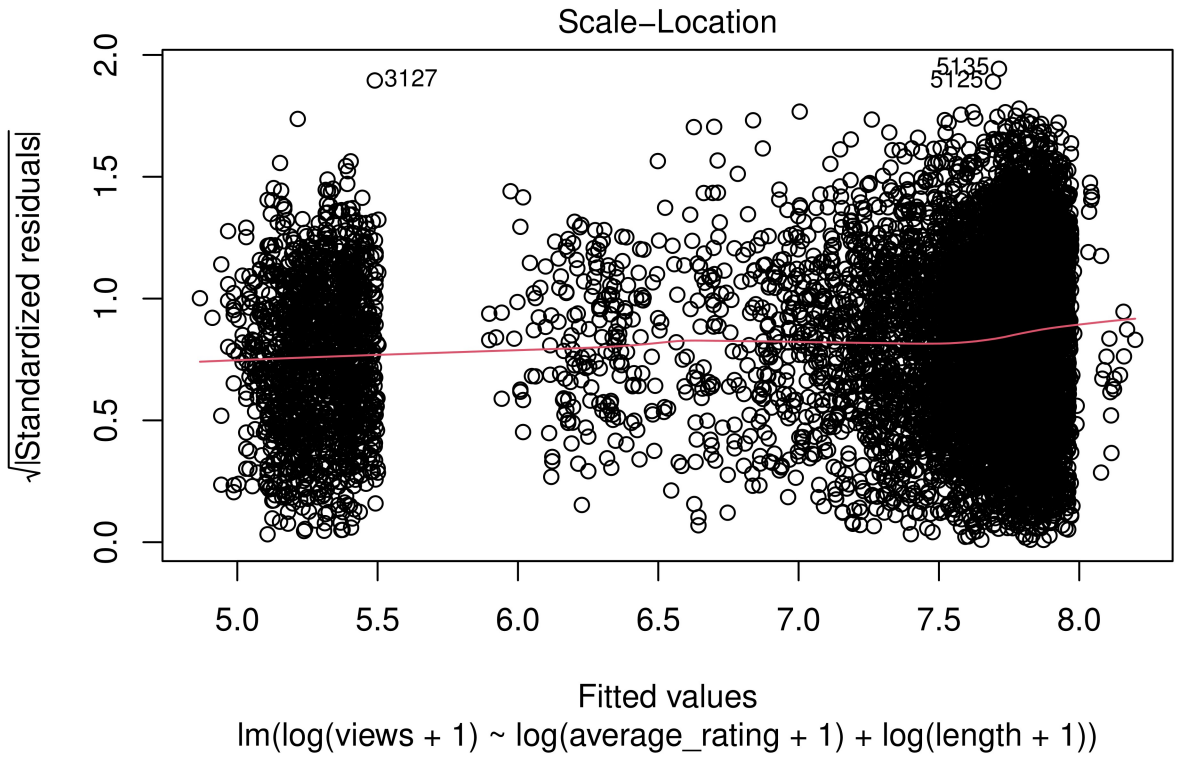
```
## Warning: Removed 9 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 9 rows containing missing values (geom_point).
```



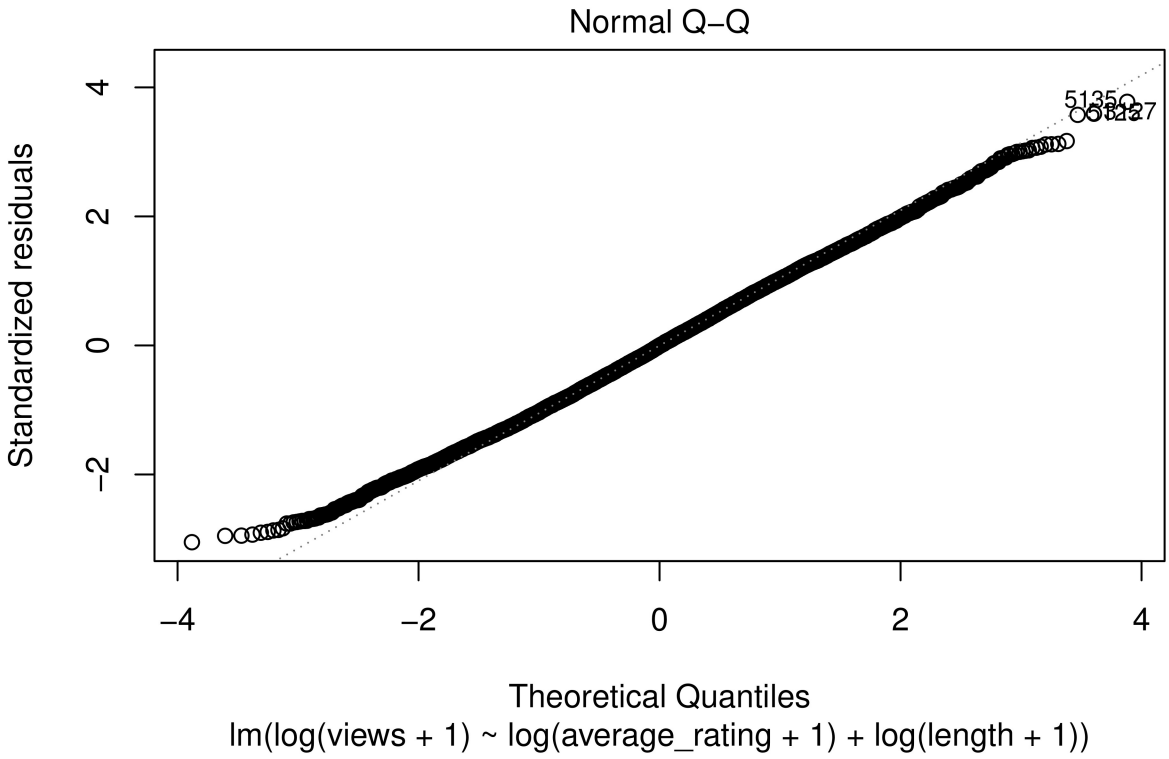
4. **Homoskedastic Errors:** To test this assumption, we conducted a scale-location plot analysis. The scale-plot analysis, when the assumption is met, should produce a roughly horizontal line midway through the scatterplot of fitted vlues and residuals (standardized) wherein the distribution of points are evenly distributed above and below the line. Initially, we tested the model `lm(log(views+1) ~ log(length+1) + average_rating))` and found that the plot did not meet the assumption. However, upon revision of the model to `lm(log(views+1) ~ log(length+1) + log(average_rating+1))`, the plotted line was approximately horizontal with an even distribution of points, indicating that the revised model does meet the assumption for homoskedastic errors.

```
#Plot scale-location plot analysis chart
plot(model, 3)
```



5. **Normally Distributed Errors:** To test the normal distribution of errors, we plotted a quantile-quantile plot (Q-Q plot) to visualize the quantiles of residuals versus theoretical quantiles of the fitted model. To meet the assumption, we are looking for a positive linear relationship between `x` and `y` with minimal outliers. The model achieves this, and therefore meets the assumption for normally distributed errors.

```
#Plot QQ plot to check for normally distributed errors
plot(model, 2)
```


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(log(views + 1) ~ log(average_rating + 1) + log(length + 1))