HW week 12

w203: Statistics for Data Science

w203 teaching team

```
library(tidyverse)
```

```
## Warning in system("timedatectl", intern = TRUE): running command 'timedatectl'
## had status 1
```

```
library(ggplot2)

library(sandwich)
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 `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.

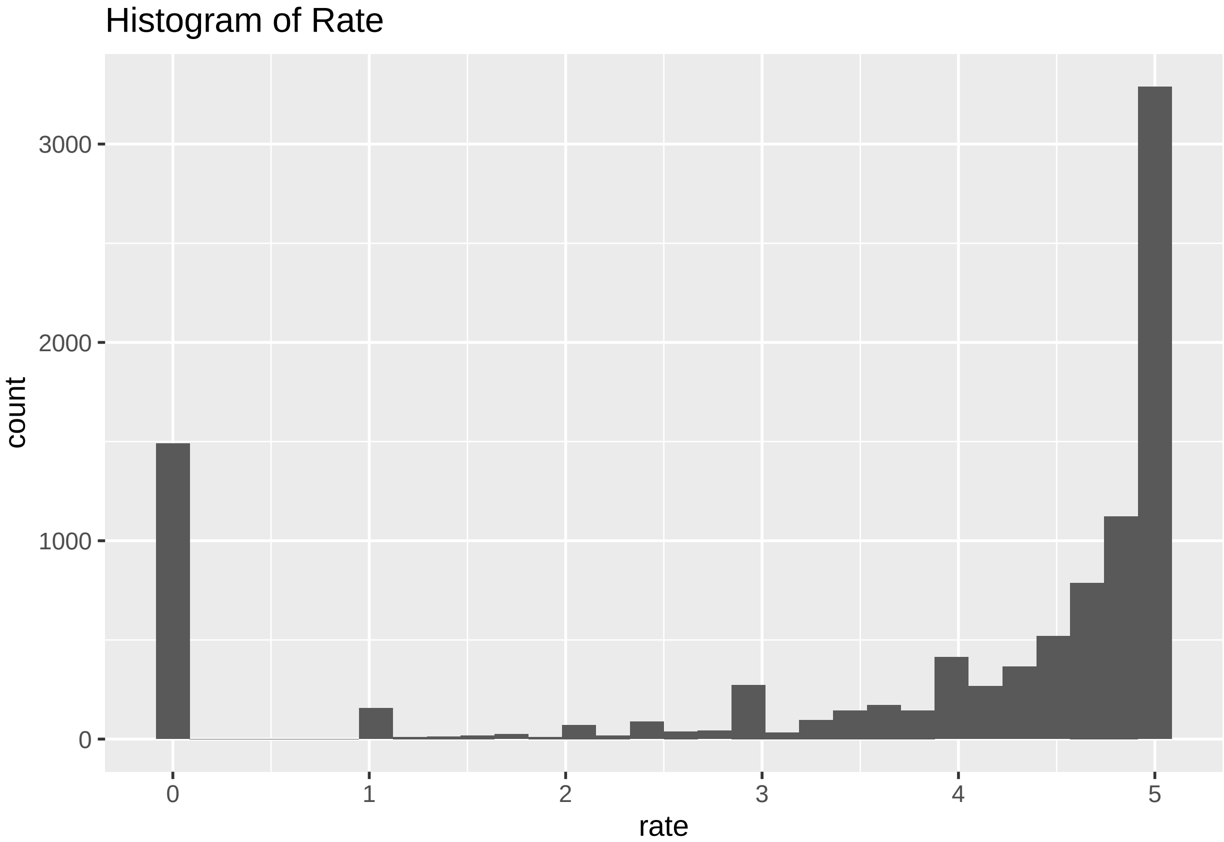
   Next let's take a look at "average_rating" column.

```
sum(is.na(d$average_rating))
```

```
## [1] 9
```

```
d <- d[!is.na(d$average_rating),]
```

```
d_rate <-ggplot(data=d, aes(x=average_rating)) +
  stat_bin() +
  labs(title="Histogram of Rate ", x="rate")
d_rate
```



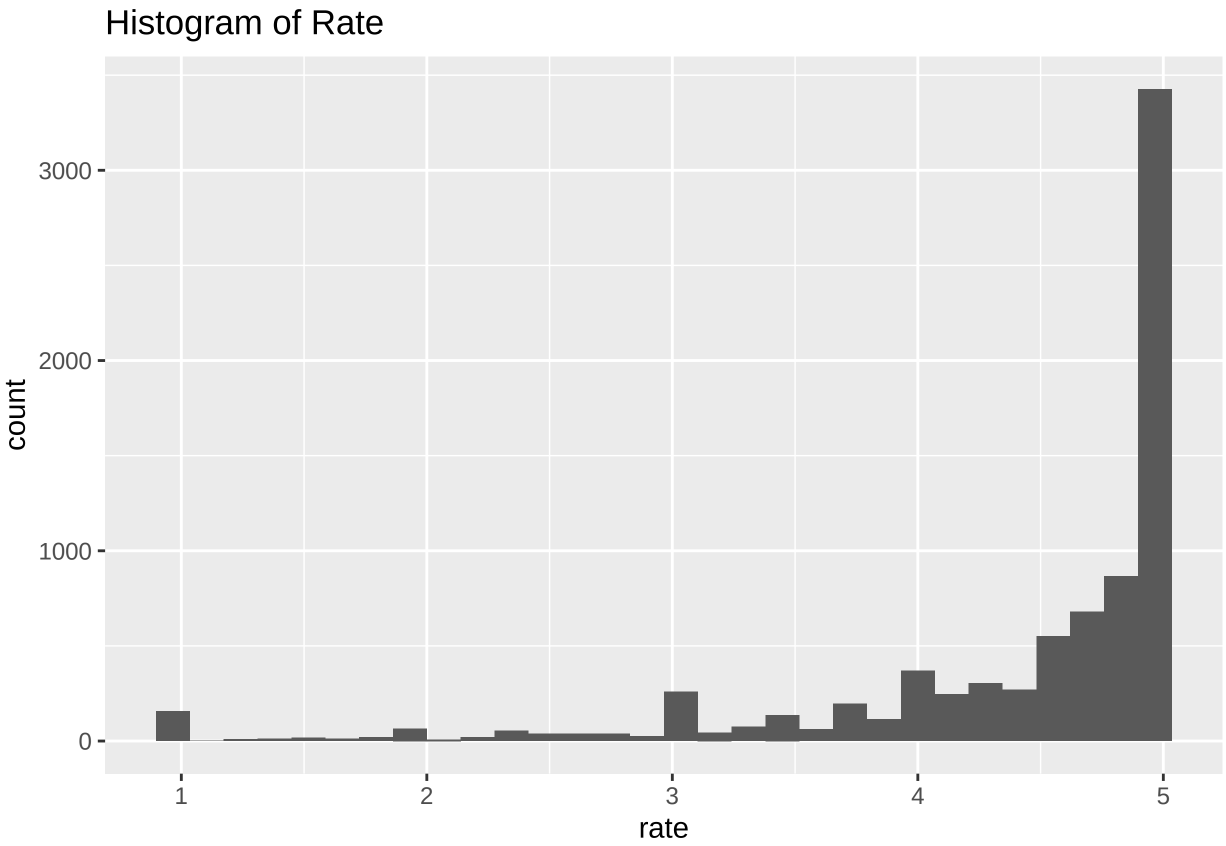
>"Average_rating" column seems to be in likert Scale >Zero rating means that the video has not been rated. >Since having a zero "average_rating" does not tell us anything about the quality. I will go ahead and delete all the "average_ratings" with zero score.

```
d <- d[d$average_rating != "0", ]
```

Let's look at the graph again.

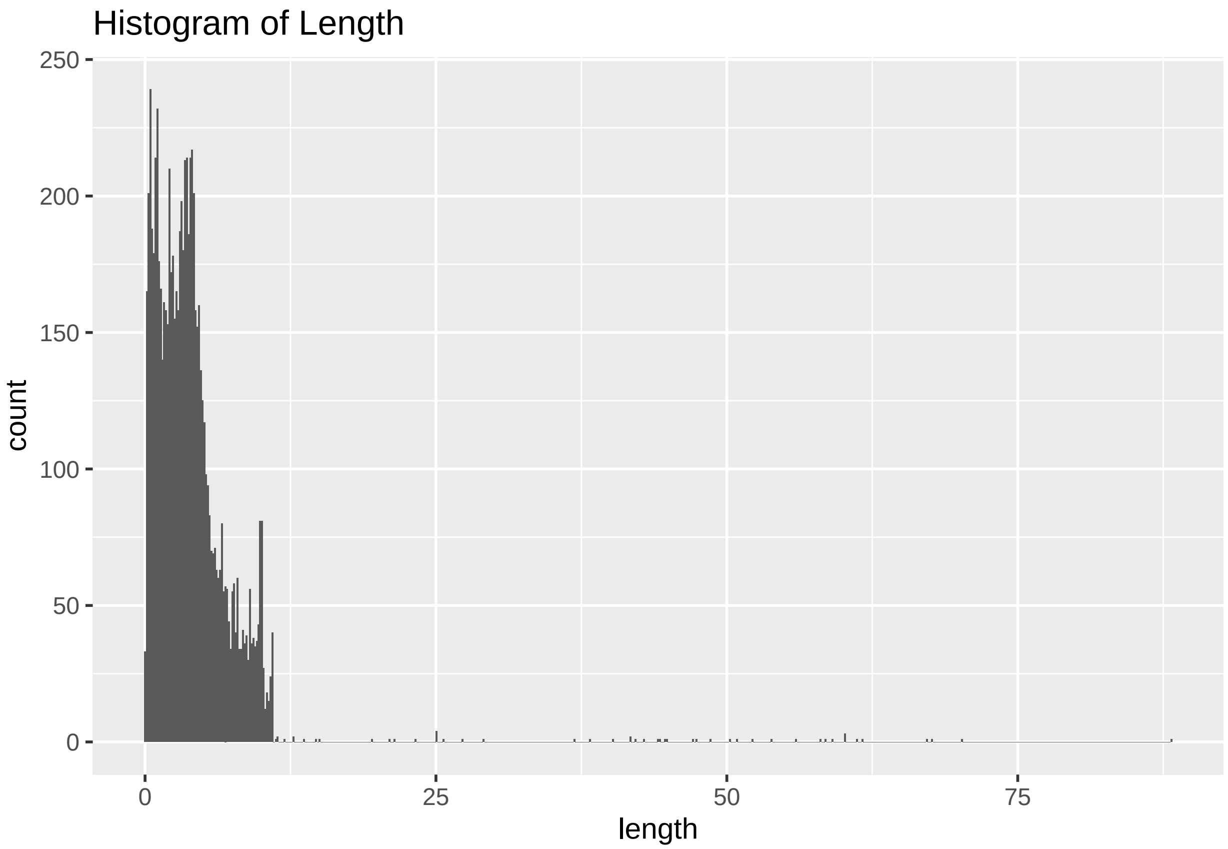
```
d_rate <-ggplot(data=d, aes(x=average_rating)) +
  stat_bin() +
  labs(title="Histogram of Rate ", x="rate")
d_rate
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



Let’s look at the length I converted all the lengths to mins. Looking at the graph, we can see that they are outliers and the bulk of videos have length that is less than 11 mins.

```
d_length <-ggplot(data=d, aes(x=length/60)) +
  geom_histogram(binwidth = 0.15) +
  labs(title="Histogram of Length ", x="length")
d_length
```

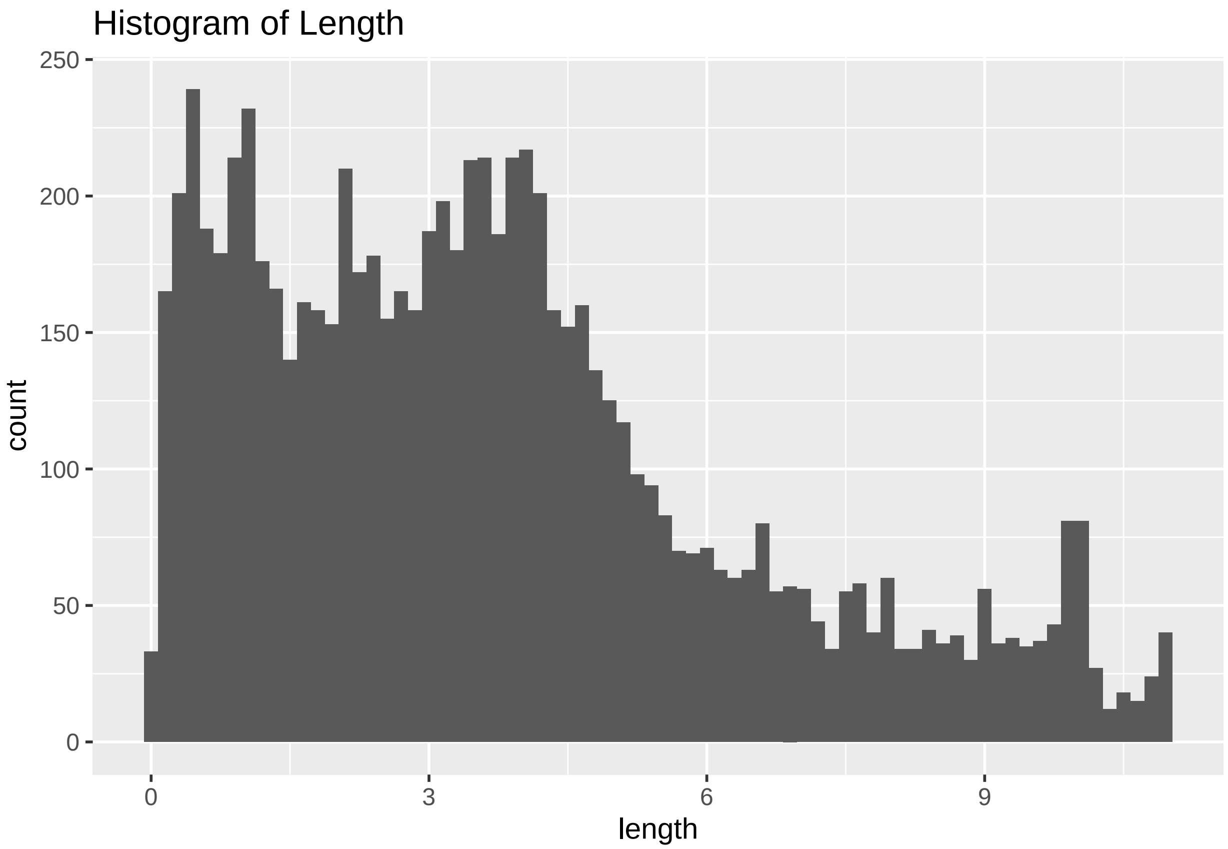


>Removing anything above 11 mins

```
# remove 11+ minute videos
d <- d[d$length <= 660,]
```

let's look at the graph again

```
d_length <-ggplot(data=d, aes(x=length/60)) +
  geom_histogram(binwidth = 0.15) +
  labs(title="Histogram of Length ", x="length")
d_length
```



let's take a look at views.

```
#How many video_ids have less than 11-digit unique string
sum(nchar(as.character(d$video_id)) != 11)
```

```
## [1] 105
```

```
#How many vidoe_ids have "#NAME" instead of 11-digit unique string.
sum(d$video_id == "#NAME?")
```

```
## [1] 105
```

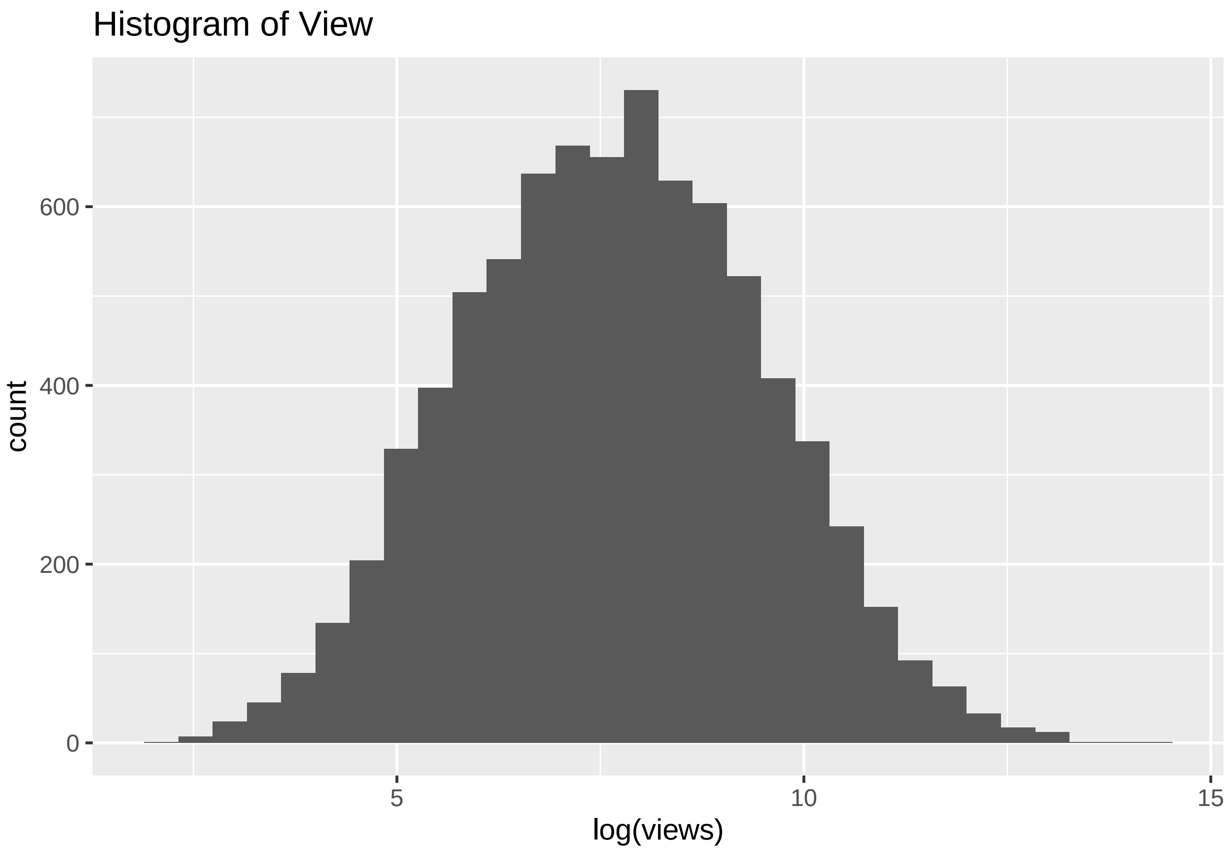
```
#are there any NAs in views column
sum(is.na(d$views))
```

```
## [1] 0
```

```
#removing the 9 NA values
d <- d[!is.na(d$views),]
```

Let's look at the graph.

```
d_view <-ggplot(data=d, aes(x=log(views))) +
  stat_bin() +
  labs(title="Histogram of View ", x="log(views)")
d_view
```



>

I transformed the “view” column into log format to better visualize it, and we can see that it is normally distributed.

> ‘What did you learn from your EDA? Cut this quoted text and describe your analysis in the quote block.’

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,

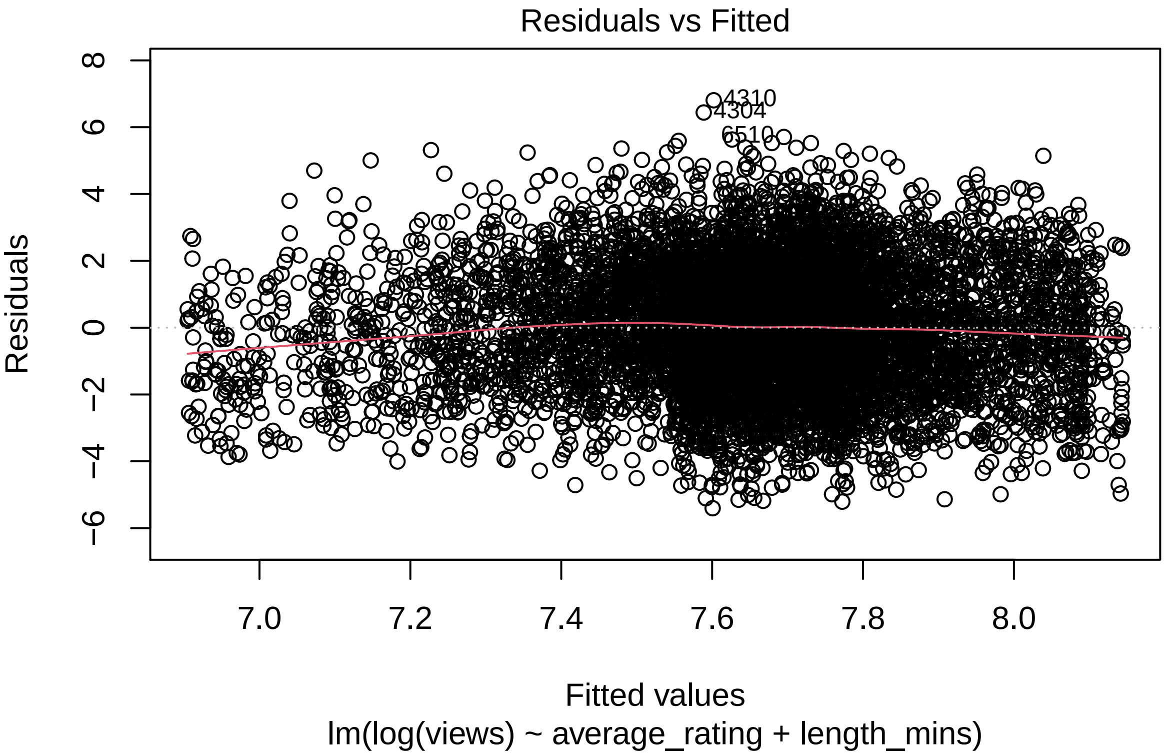
$$f(\text{views}) = \beta_0 + \beta_1 g(\text{rate}) + \beta_3 h(\text{length})$$

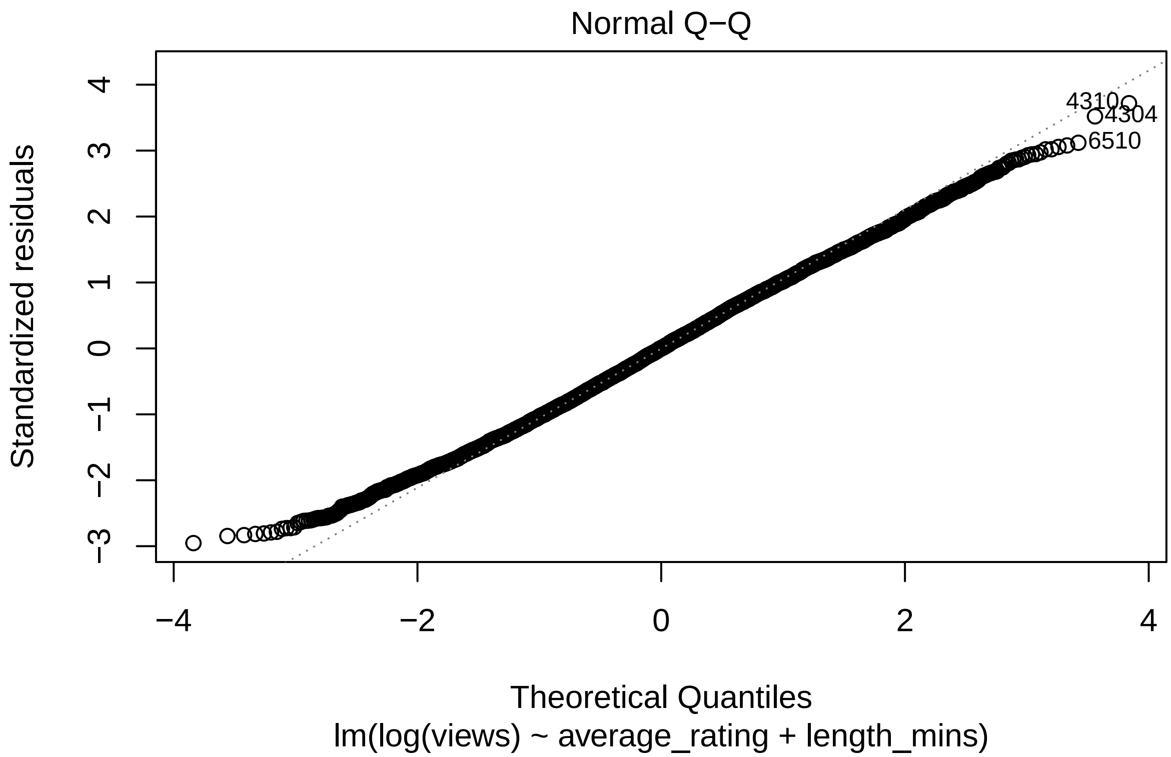
Where f, g and h are sensible transformations, which might include making *no* transformation.

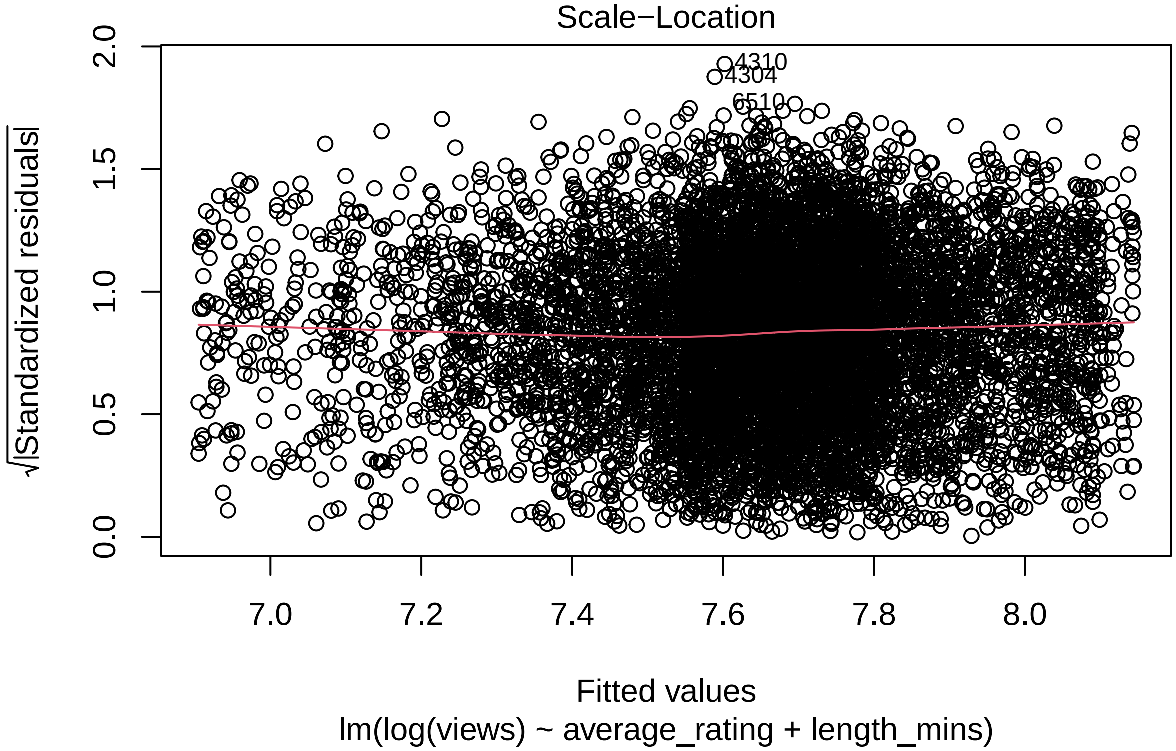
```
#transforming the length column to mins

d$length_mins <- d$length/60
model <- lm(log(views) ~ average_rating + length_mins, data=d)

summary(model)
```

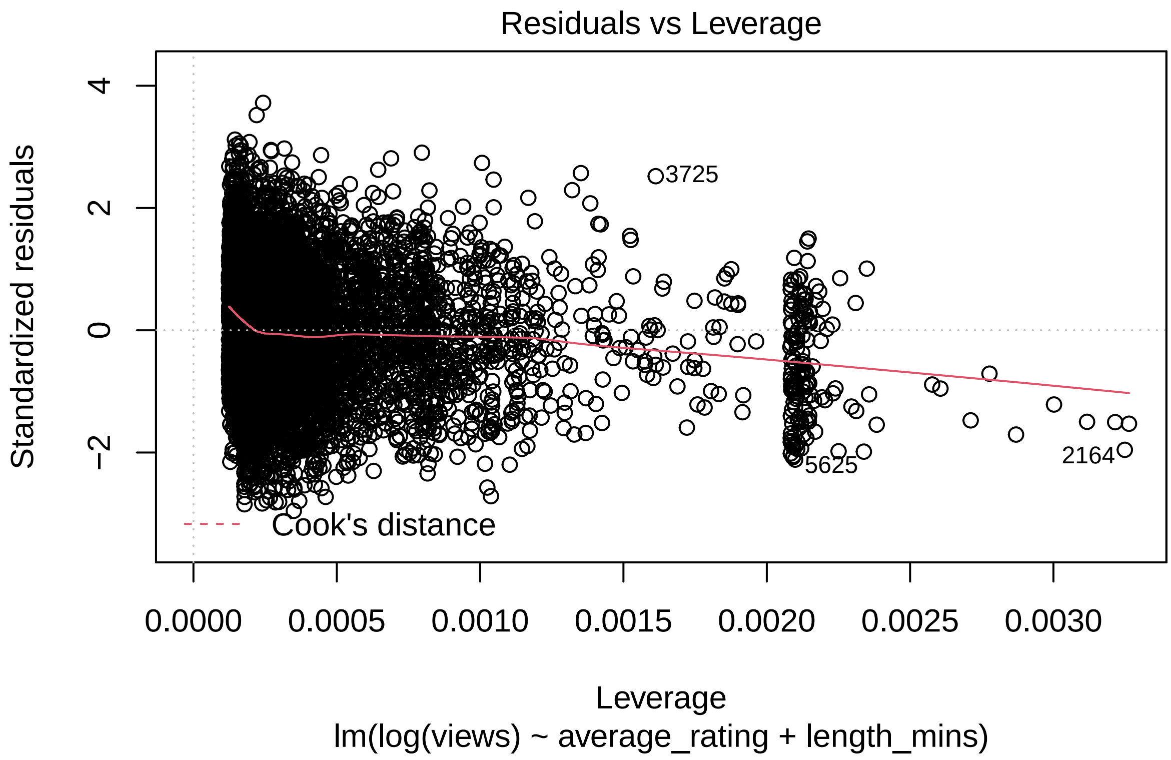
```
##
## Call:
## lm(formula = log(views) ~ average_rating + length_mins, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4034 -1.3066  0.0054  1.2960  6.8055
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.740836   0.107175  62.896  < 2e-16 ***
## average_rating 0.161292   0.023963   6.731 1.80e-11 ***
## length_mins    0.054273   0.007793   6.965 3.55e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.83 on 8065 degrees of freedom
## Multiple R-squared:  0.01383,    Adjusted R-squared:  0.01358
## F-statistic: 56.54 on 2 and 8065 DF,  p-value: < 2.2e-16
```

```r
plot(model)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(log(views) ~ average_rating + length_mins)
−4
−2
0
2
4
−3
−2
−1
0
1
2
3
4
4310
4304
6510


Scale−Location
√|Standardized residuals|
Fitted values
lm(log(views) ~ average_rating + length_mins)
7.0
7.2
7.4
7.6
7.8
8.0
0.0
0.5
1.0
1.5
2.0
4310
4304
6510



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:**

Given the explanation of the data collection, the researchers looked at all youtube videos and used a breadth-first search method. This fullfills the IID.

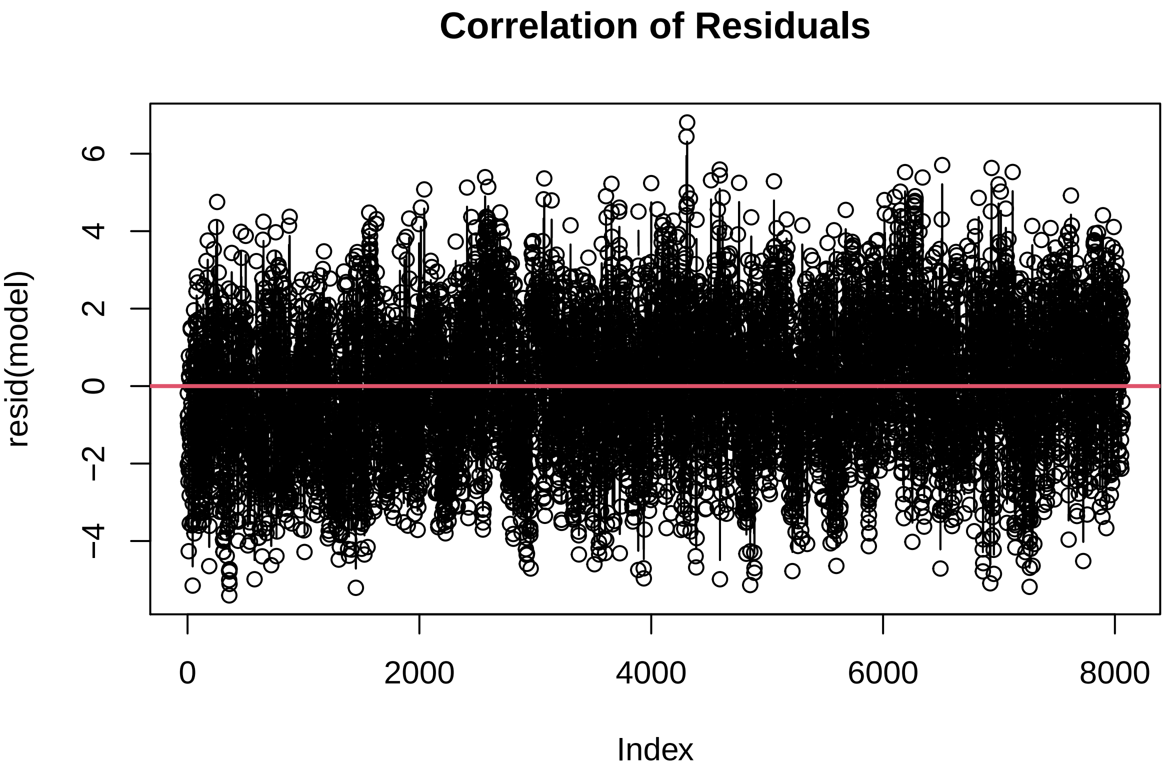
2. **No Perfect Colinearity:**

I run a correlation test below between variables, and we can see that correlation is pretty weak and not at all perfect colinear.

```
cor.test(d$average_rating, d$length, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  d$average_rating and d$length
## t = 15.537, df = 8066, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1492035 0.1915781
## sample estimates:
##       cor
## 0.1704696
```

```
plot(resid(model),type="b", main="Correlation of Residuals")+
abline(h=0,lwd=2, type="dashed", col=2)
```



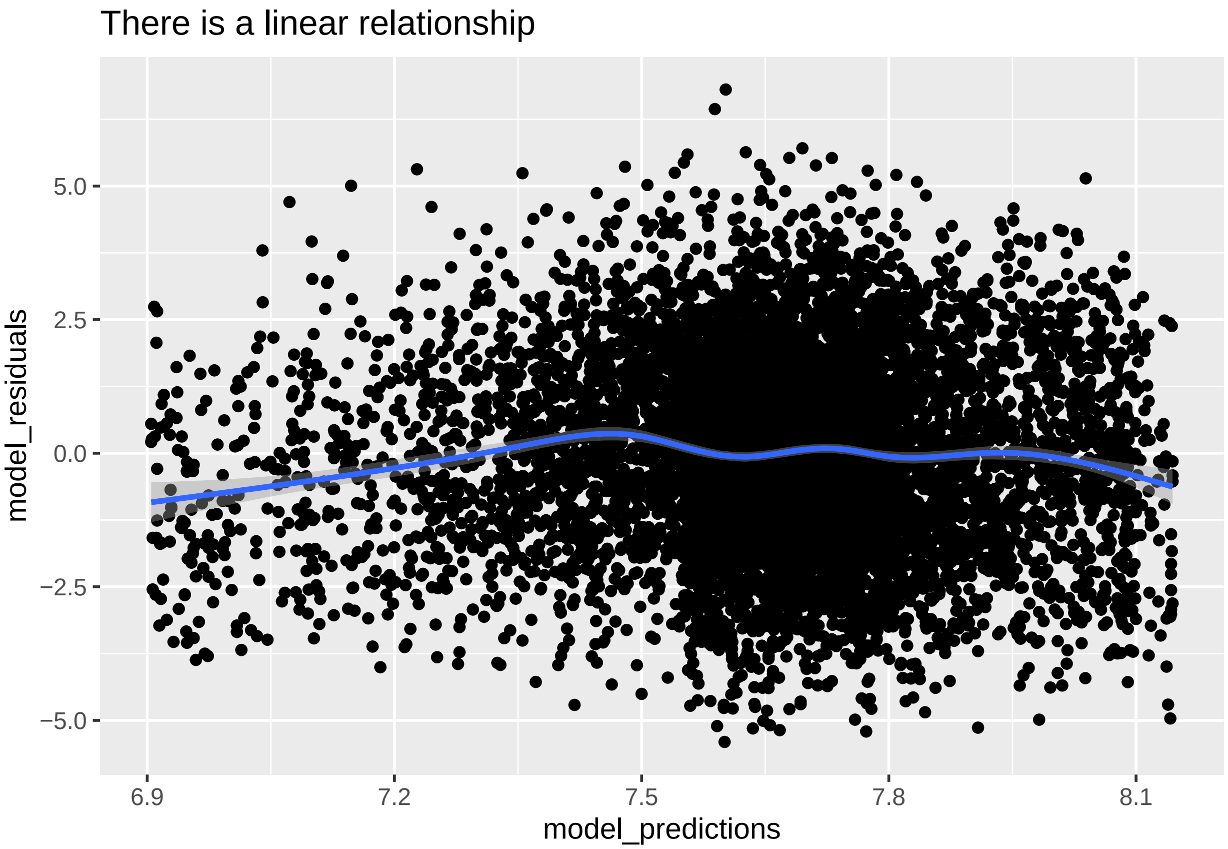
```
## integer(0)
```

3. **Linear Conditional Expectation:**

Looking at plot_1 below, we can clearly see that there is a linear relation, thus this assumption is fullfiled.

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

```
plot_1 <- d %>%
  ggplot(aes(x = model_predictions, y = model_residuals)) +
  geom_point() + stat_smooth() +
  labs(title = 'There is a linear relationship')

plot_1
```

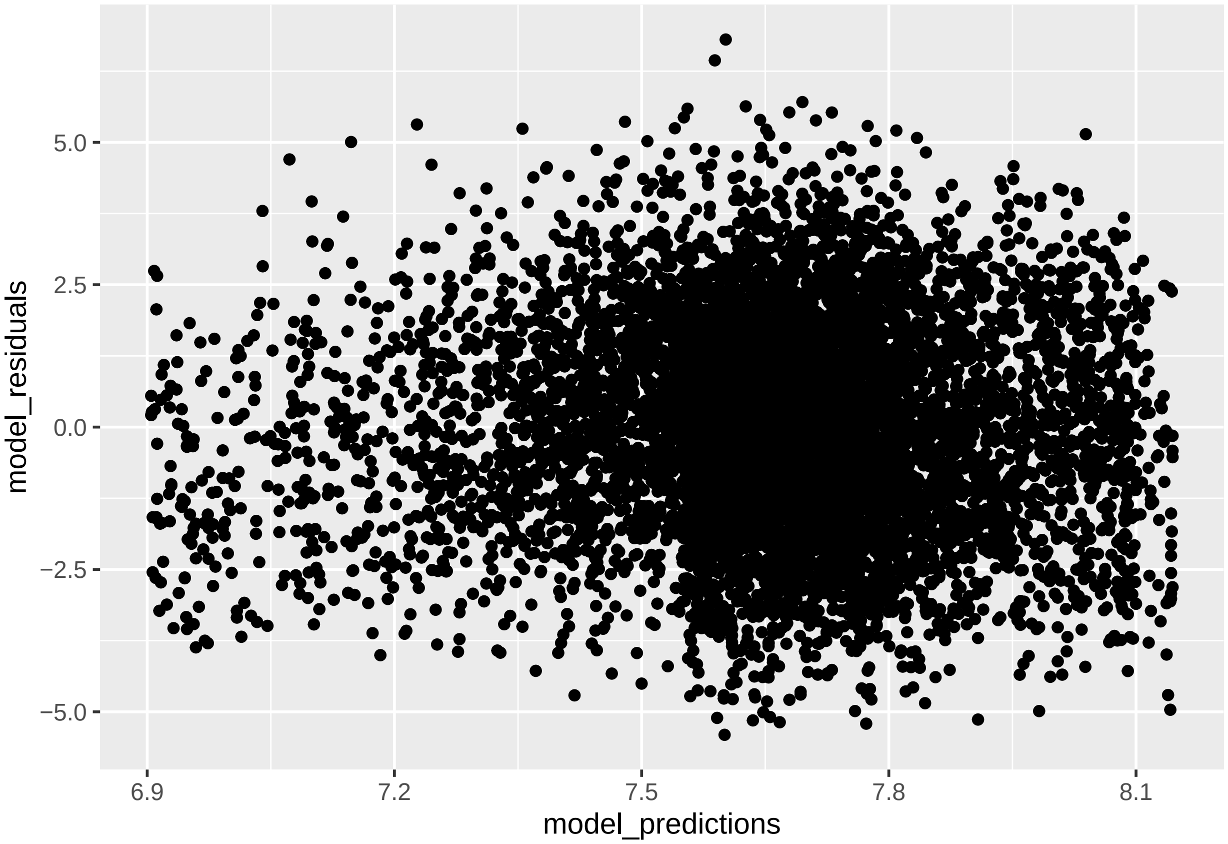


4. **Homoskedastic Errors:**

Looking at plot_2, we can see that there seems to be some heterskesadity. In other words, as we move to the right side of x axis, we can see that the residulas are concentrating more.

I run the bptest and we can can reject the null hypothesis. Null hypothesis being there is no evidence for heteroskedastic error. Looking at the p-value from our test, we can reject the null hypothesis.

```
plot_2 <- d %>%
  ggplot(aes(x = model_predictions, y = model_residuals)) +
  geom_point()
plot_2
```



```
#install.packages("lmtest")
lmtest::bptest(model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 11.044, df = 2, p-value = 0.003997
```

5. **Normally Distributed Errors:**

Looking at the qqPlot, we can see that the shape is almost normal, except that it has thin tails. However, since we have a much larger sample than the 30 that we need for CLM, we can go ahead and apply CLM with no issues.

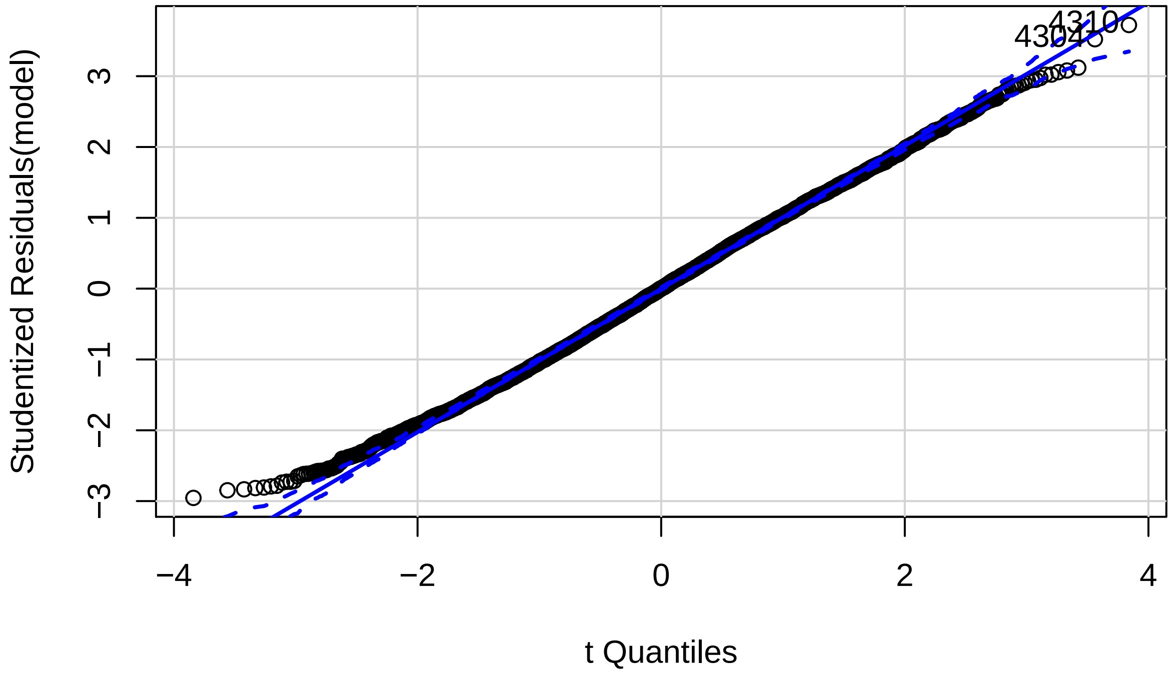
```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
## The following object is masked from 'package:purrr':
##
##     some
```

```
qqPlot(model)
```



```
## [1] 4304 4310
```