HW week 12

w203: Statistics for Data Science

w203 teaching team

```
library(tidyverse)
library(ggplot2)

library(sandwich)
library(stargazer)

library(grid)
library(gridExtra)

library(corrplot)
```

```
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.

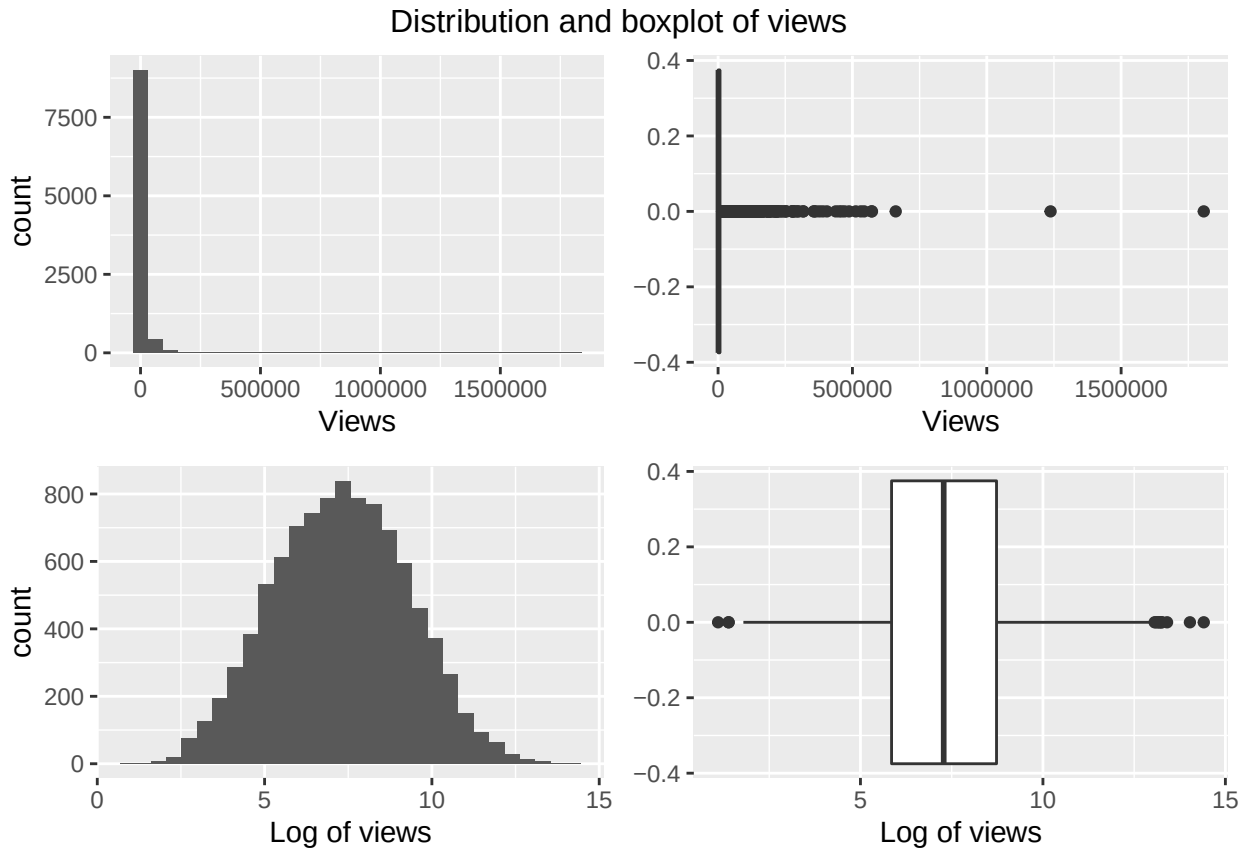
a. Perform a brief exploratory data analysis on the data to discover patterns, outliers, or wrong data entries and summarize your findings.

```
summary(d)
```

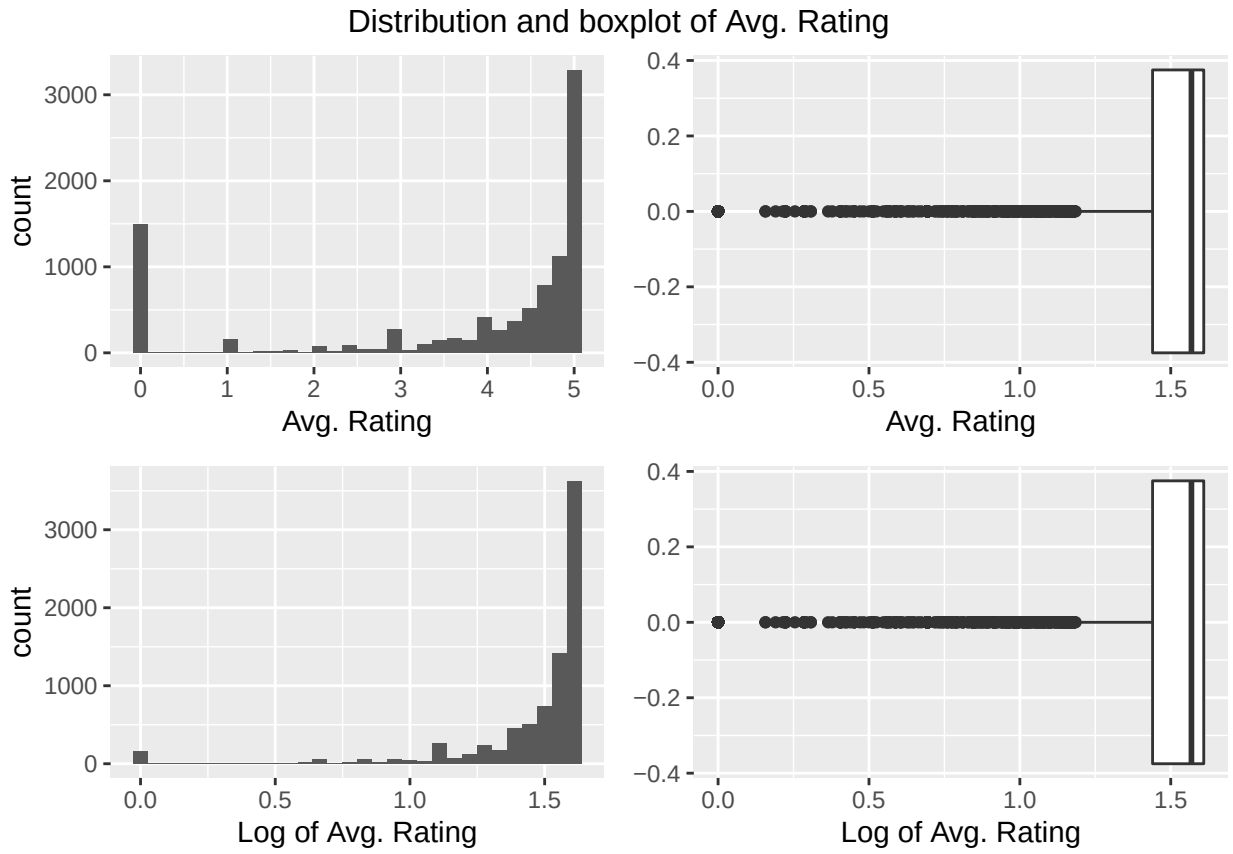
```
##    video_id           uploader              age         category
```

```
##  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
```

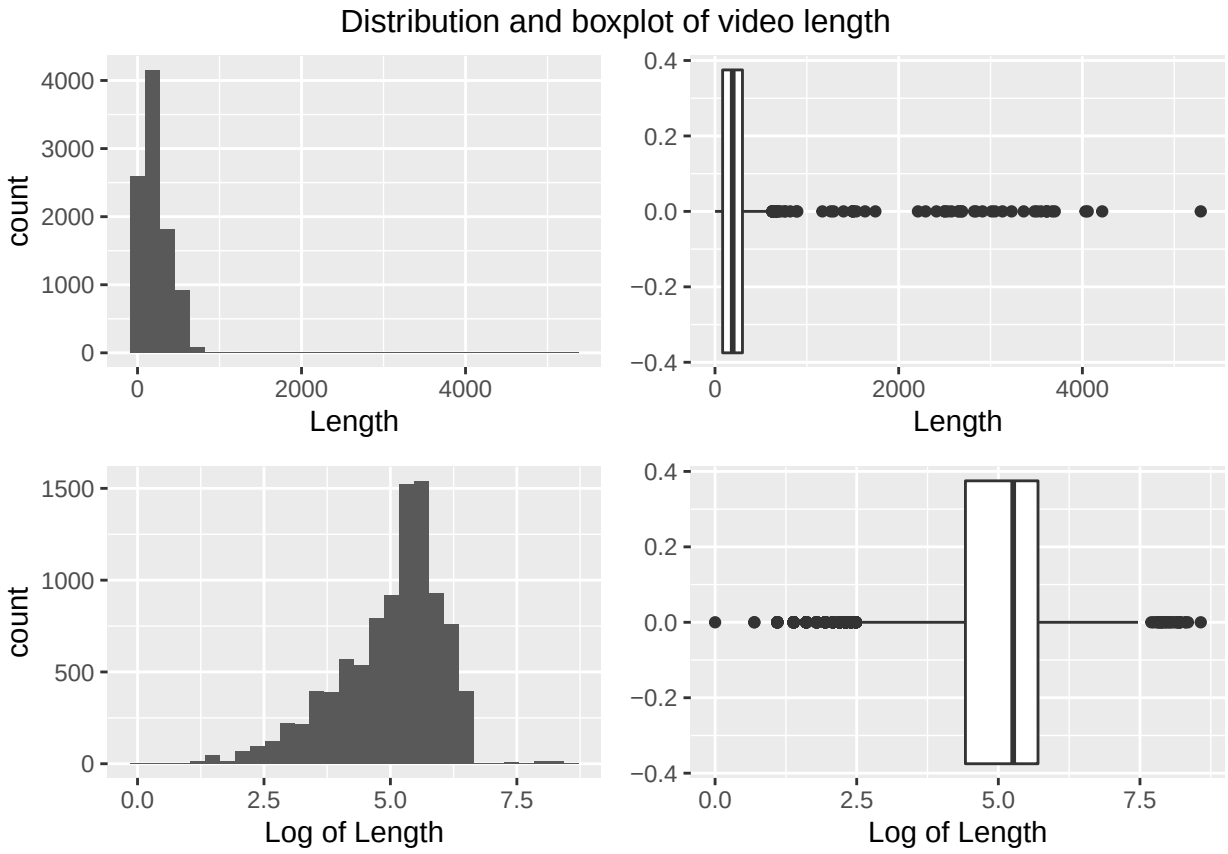
```
views_hist = d %>% ggplot() +
  aes(x=views) +
  geom_histogram() +
  labs(x="Views")

views_bp = d %>% ggplot() +
  aes(x=views) +
  geom_boxplot() +
  labs(x="Views")

logviews_hist = d %>% ggplot() +
  aes(x=log(views)) +
  geom_histogram() +
  labs(x="Log of views")

logviews_bp = d %>% ggplot() +
  aes(x=log(views)) +
  geom_boxplot() +
  labs(x="Log of views")

grid.arrange(views_hist, views_bp,
             logviews_hist, logviews_bp,
             ncol=2, nrow=2,
             top="Distribution and boxplot of views")
```




```
rating_hist = d %>% ggplot() +
  aes(x=average_rating) +
  geom_histogram() +
  labs(x="Avg. Rating")

rating_bp = d %>% ggplot() +
  aes(x=log(average_rating)) +
  geom_boxplot() +
  labs(x="Avg. Rating")

lrating_hist = d %>% ggplot() +
  aes(x=log(average_rating)) +
  geom_histogram() +
  labs(x="Log of Avg. Rating")

lrating_bp = d %>% ggplot() +
  aes(x=log(average_rating)) +
  geom_boxplot() +
  labs(x="Log of Avg. Rating")

grid.arrange(rating_hist, rating_bp,
             lrating_hist, lrating_bp,
             ncol=2, nrow=2,
             top="Distribution and boxplot of Avg. Rating")
```




```
length_hist = d %>% ggplot() +
  aes(x=length) +
  geom_histogram() +
  labs(x="Length")

length_bp = d %>% ggplot() +
  aes(x=length) +
  geom_boxplot() +
  labs(x="Length")

llength_hist = d %>% ggplot() +
  aes(x=log(length)) +
  geom_histogram() +
  labs(x="Log of Length")

llength_bp = d %>% ggplot() +
  aes(x=log(length)) +
  geom_boxplot() +
  labs(x="Log of Length")

grid.arrange(length_hist, length_bp,
             llength_hist, llength_bp,
             ncol=2, nrow=2,
             top="Distribution and boxplot of video length")
```



As we can see,, all the variables are fairly skewed. The `views` variable is highly skewed to the right, meaning that most videos got close to zero views. This is not entirely desirable, so we apply a logarithmic transformation and we see that the transformed variable is quite similar to a normal distribution and the numbers of outliers also reduces. We will then keep this transformation for the model.

Average rating has a somewhat bimodal feature as we see more density in the cutoff values 0 and 5. In this case, a logarithmic transformation doesn't help. Standardizing the variable wouldn't be appropriate either as it is based on a likert scale, for what the mean may not have any particular meaning and would complicate the interpretation of the coefficient. Another possible transformation would be binning the variables and include them as a factor with values {1, 2, 3, 4, 5}. But we are not going to follow this approach as we will be losing some information in the process.

Last, the `length` of the videos has also quite a right skewed distribution, with many videos being just over 1 minute (~83 seconds). Applying a logarithm to the variable also helps to center the distribution, but the transformation has a clear cutoff which is not ideal. We will still use this transformation for the model.

We also see some NA's that may generate some errors down the line. Given the large amount of data and that there are only a few NA's, we will remove observations with missing data.

```
d_clean = d %>% filter(!is.na(length))
```

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(I(log(views)) ~ 1 + average_rating + I(log(length)), data=d_clean)
```

```
stargazer(
  model,
  title="Linear regression for YouTube views",
  se = list(get_robust_se(model)),
  dep.var.labels=c("Log of views"),
  covariate.labels=c("Avg. Rating", "Log of Length"),
  type="latex",
  header=FALSE
  )
```

Table 1: Linear regression for YouTube views

	Dependent variable:
	Log of views
Avg. Rating	0.467***
	(0.010)
Log of Length	0.105***
	(0.018)
Constant	5.010***
	(0.088)
Observations	9,609
R^2	0.189
Adjusted R^2	0.189
Residual Std. Error	1.799 (df = 9606)
F Statistic	1,121.899*** (df = 2; 9606)
Note:	*p<0.1; **p<0.05; ***p<0.01

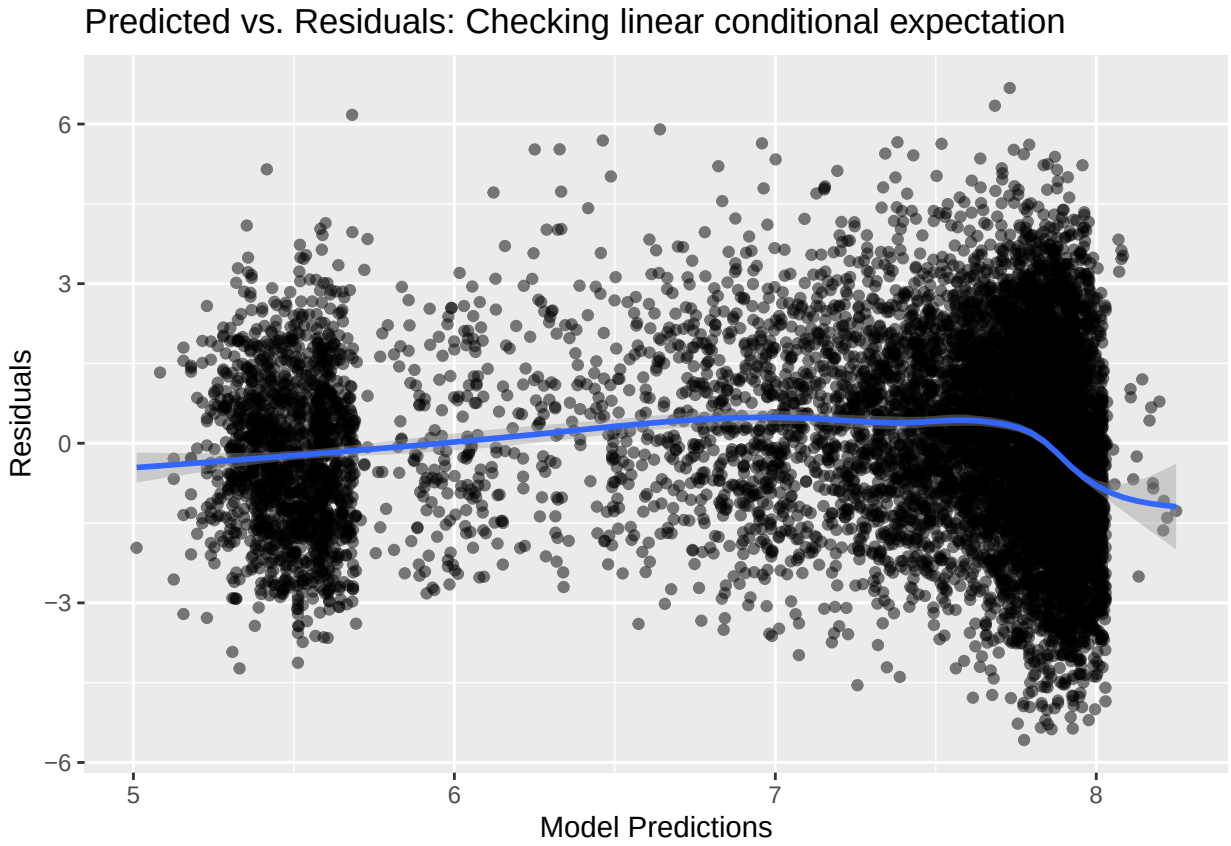
c. Using diagnostic plots, background knowledge, and statistical tests, assess all five assumptions of the CLM. When an assumption is violated, state what response you will take. As part of this process, you should decide what transformation (if any) to apply to each variable. Iterate against your model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

1. **IID Data:** This condition is rarely met perfectly and this is not the exception. First of all, the way in which the data was collected is not completely random. The videos were selected by looking into YouTube's most popular uploads and initiating a breadth depth search to look into the related videos section of those. This means that the data will potentially have more popular than non popular videos as a starting point. Also, given that the video selection is done from the related videos section, there probably other similarities between the videos selected such as being from the same author or YouTube channel, or in the same language which will create some dependencies or clustering among them.
2. **No Perfect Collinearity:** We can see that R hasn't dropped any of our variables in the model or failed to run the regression. This means that the X matrix is invertible, thus no perfect colinearity exists and the BLP exists. We can also see that there is a low correlation between all the explanatory variables:

```
corrplot(cor(d_clean[c("views", "average_rating", "length")]), method="number")
```

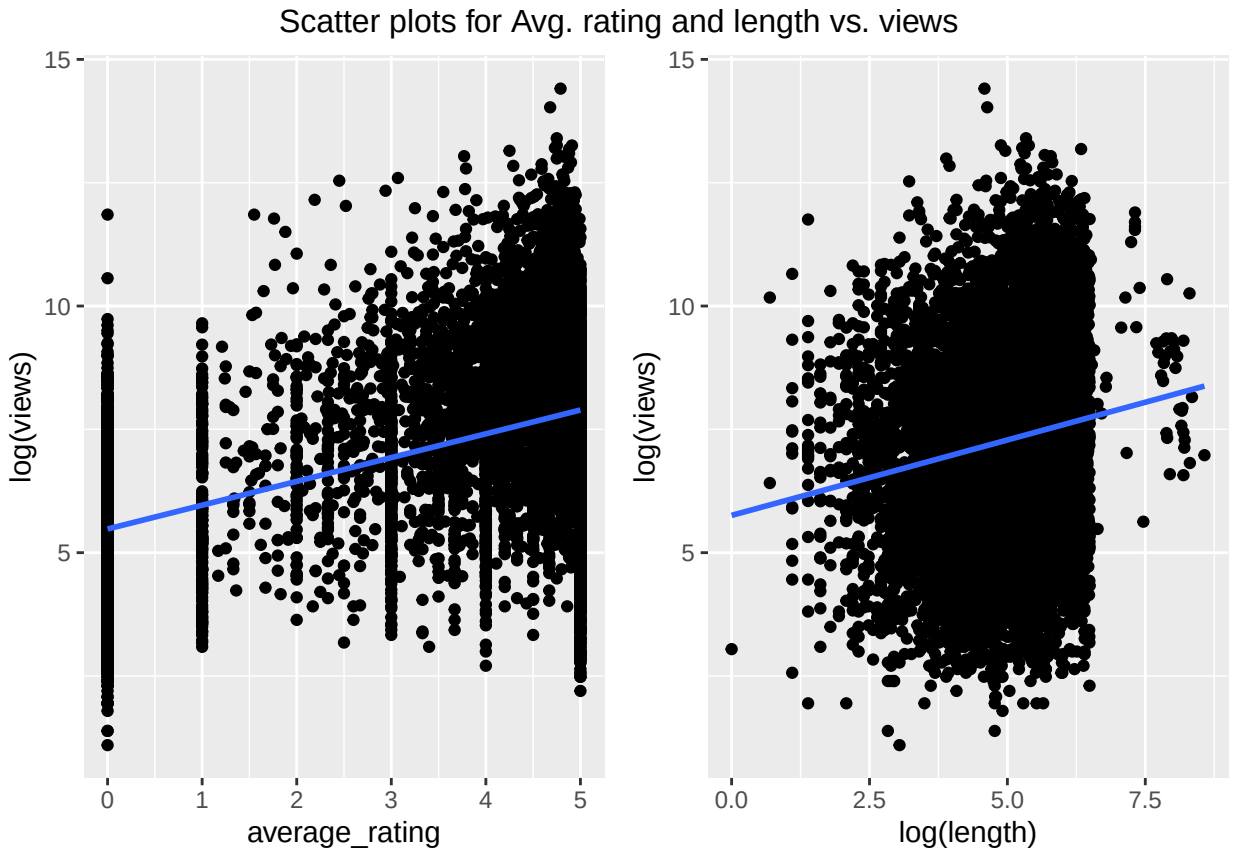
3. **Linear Conditional Expectation:** For this assumption, we can check the predicted vs residuals graph and see if the pattern we see is linear or not. Given the graph below with the smooth curve, we see that for almost the whole range of values, the relationship is linear. Nevertheless, after the cutoff value of 8, the relationship becomes less clear. Given the low density of points in this range of values, we can assume for the relationship to be linear.

```
d_mod = d_clean %>%
  mutate(
    model_preds=predict(model),
    model_resids=resid(model)
  )

d_mod %>%
  ggplot(aes(x=model_preds, y=model_resids)) +
  geom_point(alpha=0.5) +
  geom_smooth() +
  labs(title="Predicted vs. Residuals: Checking linear conditional expectation",
       x="Model Predictions",
       y="Residuals")
```



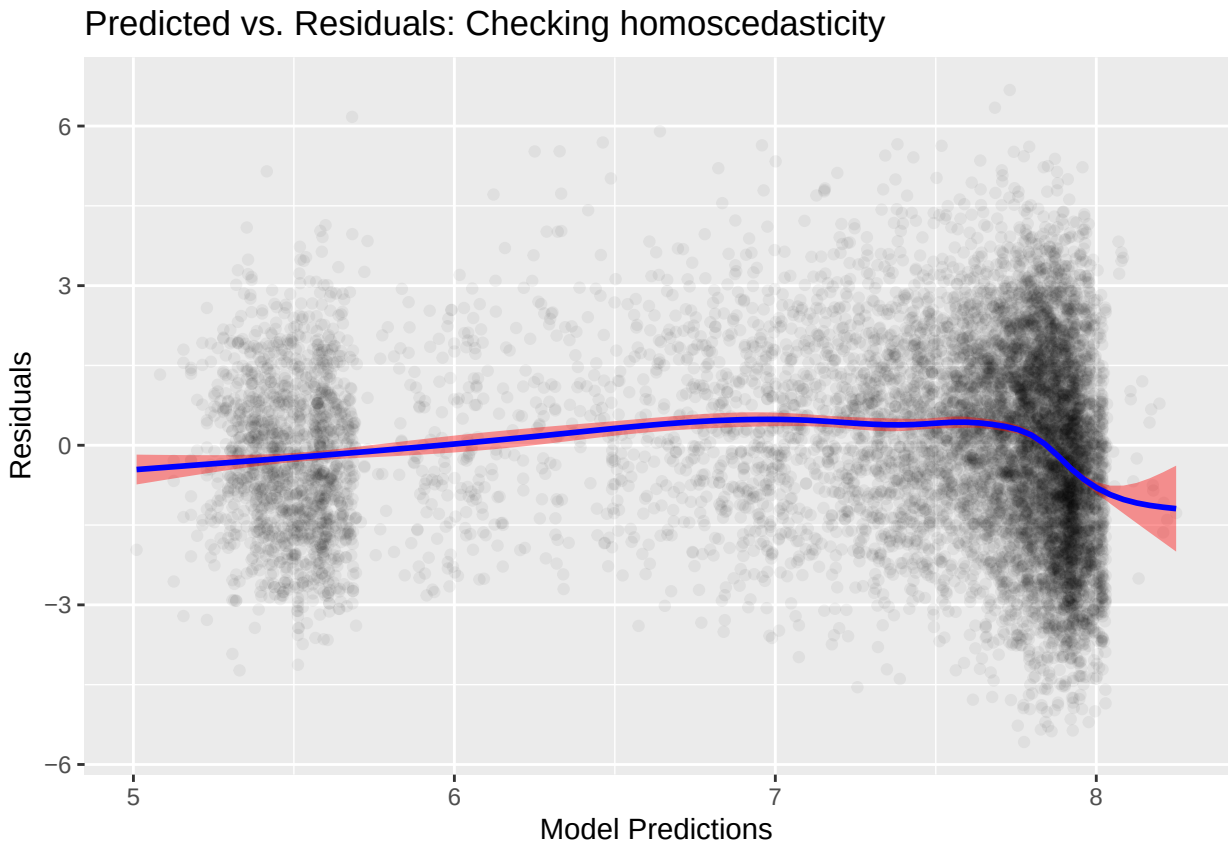
Given the low number of variables in the model, we could also check the scatter plots to see if the pattern across the outcome and indepdent variables appears linear as well. We can see that in both cases a line could potentially capture well the relationship for both variables, strengthening the theory that this condition is satisfied:

```
rate_scat = d_mod %>%
  ggplot() +
  aes(x=average_rating, y=log(views)) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE)

len_scat = d_mod %>%
  ggplot() +
  aes(x=log(length), y=log(views)) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE)

grid.arrange(rate_scat, len_scat, ncol=2, top="Scatter plots for Avg. rating and length vs. views")
```



4. **Homoskedastic Errors:** For this condition to be met we need to check if the conditional variance is constant, meaning that the spread of the residuals is the same across all range of predicted values. In the graph below, we see that this is the case for the ranges of values with the highest density, but towards the extreme values, the standard errors become bigger. Given the low density on the range of values we see this isn't met, we can attribute this to the sampling variation, and assume this condition to be satisfied.

```
d_mod %>%
  ggplot(aes(x=model_preds, y=model_resids)) +
  geom_point(alpha=0.05) +
  geom_smooth(color="blue", fill="red") +
  labs(title="Predicted vs. Residuals: Checking homoscedasticity",
       x="Model Predictions",
       y="Residuals")
```



R also makes it easy to run a Breusch-Pagan homoscedasticity test. Given the small p-value of the test, we fail to reject H_0, meaning that we can't find enough evidence to say that homoscedasticity is not present. This strengthens our visual analysis from above and we conclude that the condition is met:

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

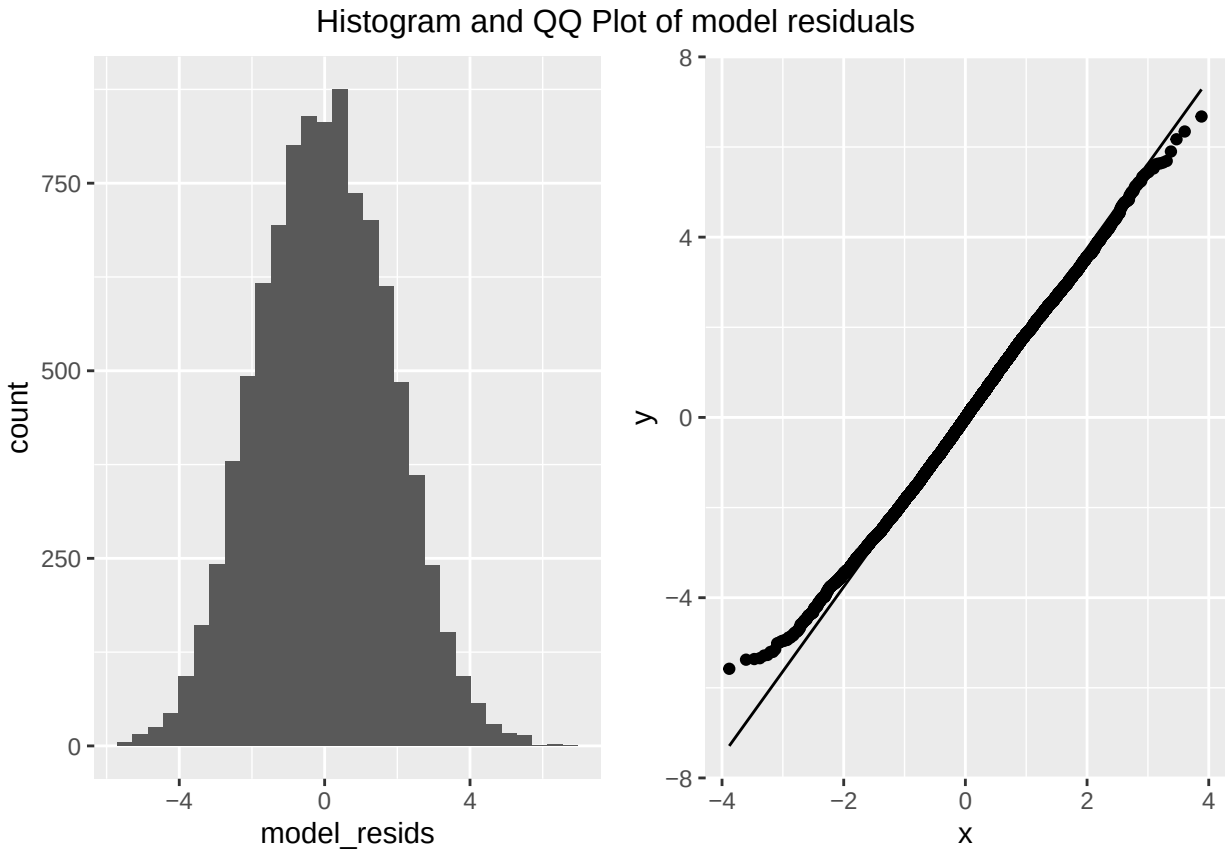
```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
bptest(model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 123.66, df = 2, p-value < 2.2e-16
```

5. **Normally Distributed Errors:** Here we are looking for the residuals of the model to be Normally distributed with $\mu = 0$. Given the distribution and QQPlot of the residuals from the below, we can say that besides a small deviation on the smaller quantiles of the distribution, this is most likely the case and thus the model complies with this requirement.

```
resid_hist = d_mod %>% ggplot(aes(x=model_resids)) +
  geom_histogram()

resid_qq = d_mod %>% ggplot(aes(sample=model_resids)) +
  stat_qq() + stat_qq_line()

grid.arrange(resid_hist, resid_qq, ncol=2, top="Histogram and QQ Plot of model residuals")
```



Based on our analysis, all CLM conditions are met, for what we can say that this OLS regression produces unbiased estimates for both the coefficients and their associated uncertainty (i.e. Standard Errors). Given the large sample, we can also say that the estimates are also consistent.