HW week 12

w203: Statistics for Data Science

Legg (Ho Man) Yeung

OLS Inference

The file videos.txt contains data scraped from Youtube.com.

Shape of Data

1. There are 9489 observations of 9 variables. 9 observations have NA values across all fields except "video_id". In total there are 9480 observations we can use to build linear models.

```
df = read.table("videos.txt", sep = "\t", header = T)
#summary(df)
str(df)
```

```
## 'data.frame':    9489 obs. of  9 variables:
##  $ video_id: Factor w/ 9489 levels "__zVzDy4M0M",..: 1653 5004 9298 893 3056 8184 4705 9447 4800 542
##  $ uploader: Factor w/ 7166 levels "","000mdc000",..: 755 4393 6196 906 2190 5780 5183 2884 1299 341
##  $ age     : int  1131 1236 1243 1237 1252 1236 1053 1240 1237 1187 ...
##  $ category: Factor w/ 17 levels "","' UNA ","Autos & Vehicles",..: 4 10 6 6 4 6 4 6 10 9 ...
##  $ length  : int  126 243 105 278 26 252 162 37 166 139 ...
##  $ views   : int  204 1652 898 928 392 318 749 10 115 617 ...
##  $ rate    : num  3 3.91 4.48 5 1.5 5 3 0 2 4.67 ...
##  $ ratings : int  2 11 81 24 8 2 6 0 1 24 ...
##  $ comments: int  1 4 36 13 17 3 6 0 0 17 ...
```

```
head(df,10)
```

```
##       video_id            uploader  age      category length views rate
## 1  9QR1tni7Ofo              BHJJYP 1131        Comedy    126   204 3.00
## 2  l1DCSqAJ740          musicalrox 1236         Music    243  1652 3.91
## 3  ZES_o3XYGjM        tessaceleste 1243 Entertainment    105   898 4.48
## 4  4I8b40cViDE booloveswondergirls 1237 Entertainment    278   928 5.00
## 5  Elp6Bf0HJIM  Fizz101Productionz 1252        Comedy     26   392 1.50
## 6  VPuKu7aU9GY         slytherin66 1236 Entertainment    252   318 5.00
## 7  K3SmrkmdCbM     reallycoolgamer 1053        Comedy    162   749 3.00
## 8  ztharMK1gC8      improvcrazy925 1240 Entertainment     37    10 0.00
## 9  KgUXi9EBqiQ           cmmdotcom 1237         Music    166   115 2.00
## 10 mjU2QlfSvGo            kickanus 1187 Howto & Style    139   617 4.67
##    ratings comments
## 1        2        1
## 2       11        4
## 3       81       36
## 4       24       13
## 5        8       17
## 6        2        3
## 7        6        6
## 8        0        0
## 9        1        0
```

```
## 10      24       17
```

```
filter = !is.na(df$views) & !is.na(df$rate) & !is.na(df$length) & !is.na(df$ratings)
df = df[filter,]
```

Question 1.

Fit a linear model predicting the number of views (views), from the length of a video (length) and its average user rating (rate).

Answer 1.

1. Here's a model using the variables as-is.

```
model = lm(views ~ length + rate, data = df)
model
```

```
##
## Call:
## lm(formula = views ~ length + rate, data = df)
##
## Coefficients:
## (Intercept)       length         rate
##     789.683        3.082     2105.454
```

2. Since our variables "views" and "length" are highly skewed (demonstrated below), we build another model using Box-Cox transformation for comparative purposes.

```
# log transform variable views
df$log_views = log(df$views)

# box-cox transform variable length
lenBCMod = caret::BoxCoxTrans(df$length)
print(lenBCMod)
```

```
## Box-Cox Transformation
##
## 9480 data points used to estimate Lambda
##
## Input data summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0    83.0   193.0   226.7   298.2  5289.0
##
## Largest/Smallest: 5290
## Sample Skewness: 7.39
##
## Estimated Lambda: 0.3
```

```
df$bc_length = df$length^0.3

# construct new linear model
model_transformed = lm(log_views ~ bc_length + rate, data = df)
model_transformed
```

```
##
## Call:
```

```
## lm(formula = log_views ~ bc_length + rate, data = df)
##
## Coefficients:
## (Intercept)    bc_length         rate
##     5.09997      0.09409      0.46601
```

Question 2.

Using diagnostic plots, background knowledge, and statistical tests, assess all 6 assumptions of the CLM. When an assumption is violated, state what response you will take.

Answer 2.

The six assumptions of Classical Linear Model:

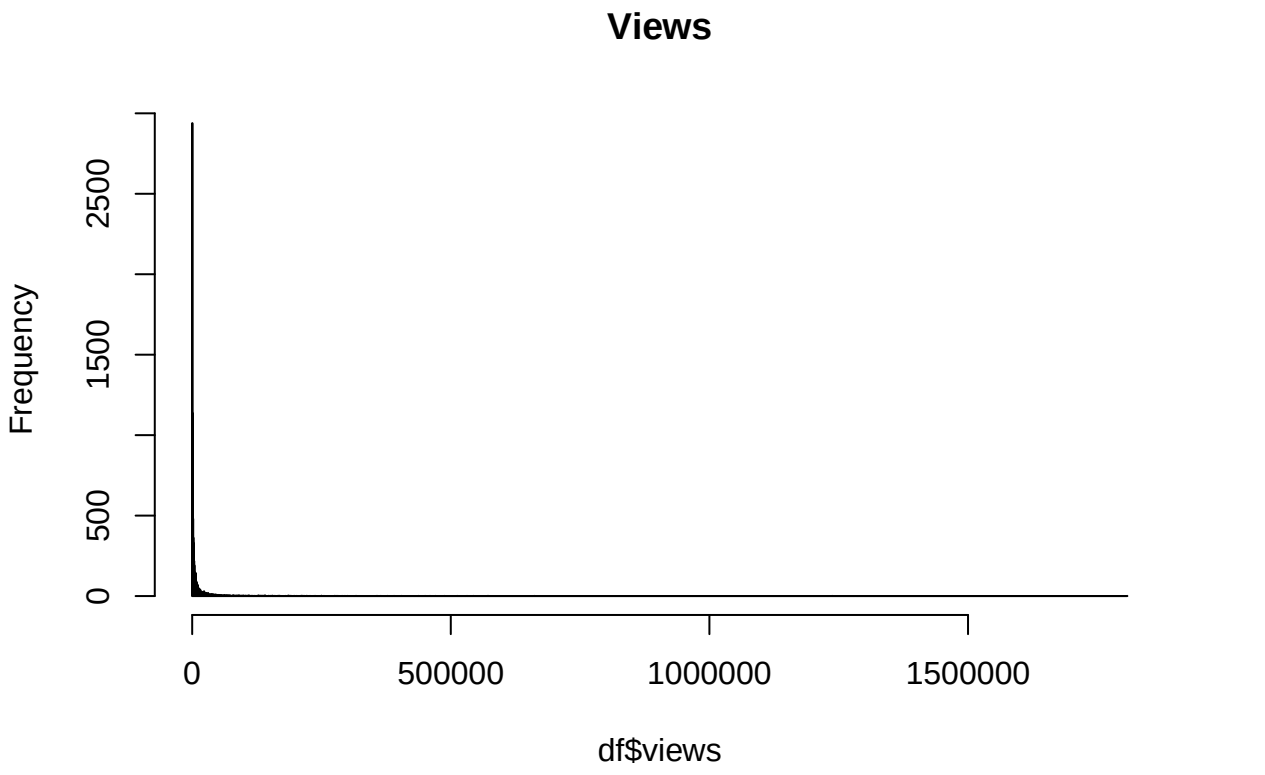
1. Linear Relationship

- We have assumed a linear relationship in the construction of the model. Without further restriction to the error term, this weak assumption holds.

2. Random Sampling:

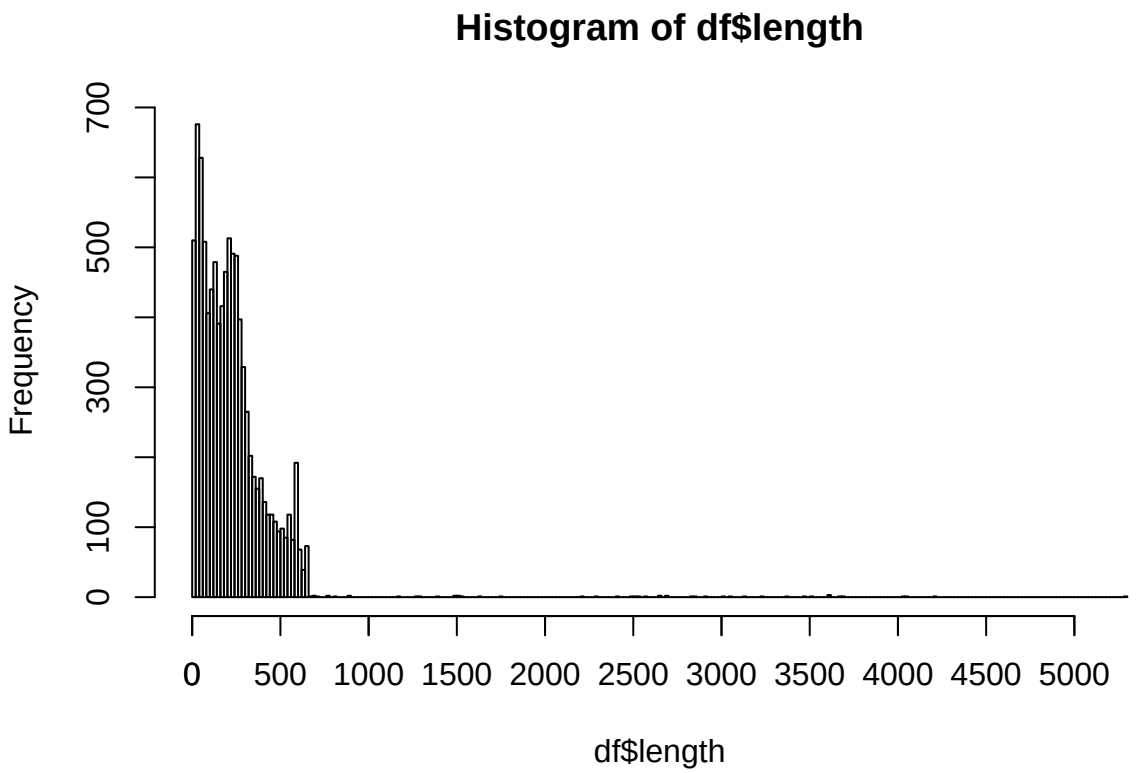
- We don't have good intuitions about the population distribution. Let examine if the sample distributions contradict our intuitions.
- "views" has extreme leptokurtosis(747.5627371) and positive skew (20.5711186). This is not surprising given few videos are popular, the popularity of these videos are further pushed by youtube's recommendation algorithms.

```
hist(df$views, breaks = "FD", main = "Views")
```



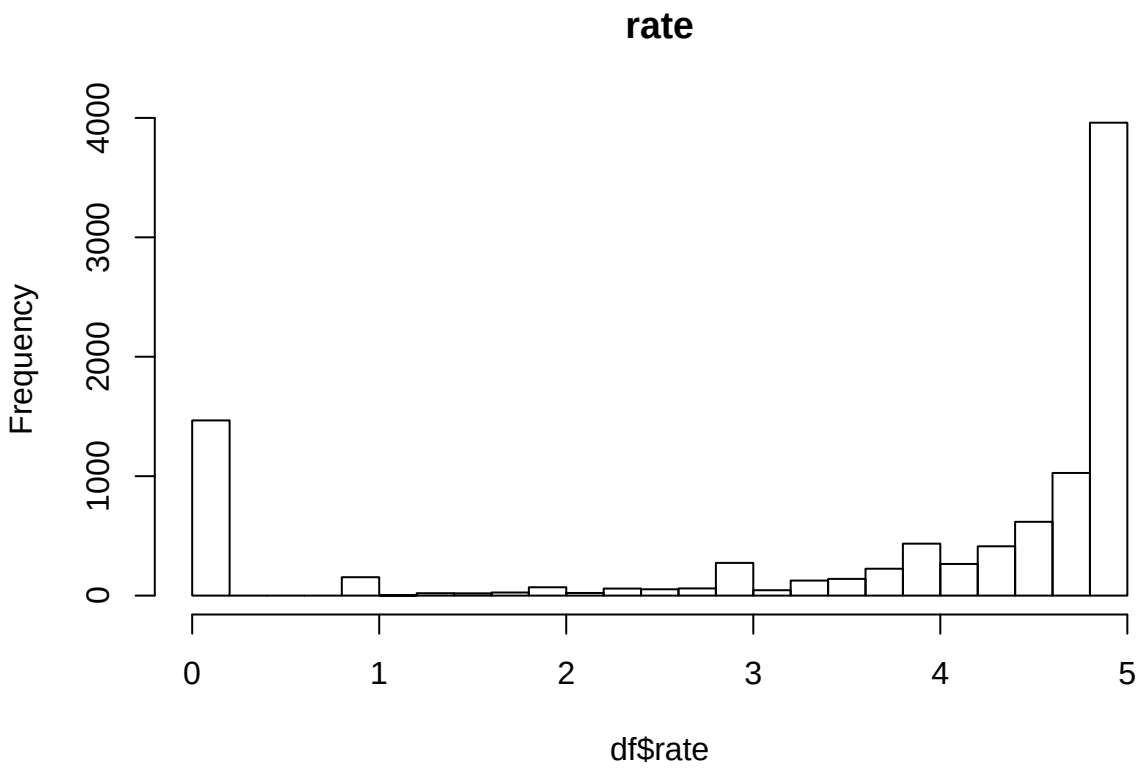
- "length" also has high leptokurtosis(103.6724118) and positive skew (7.3868101). This is not surprising, given the effort to make a long video with substantial content can increase drastically with the length of video. However, notice two spikes of frequency at around 300, 600 and a sharp drop at around 650. These can be signs of a biased sample.
- To verify using external sources, we can test if our sample distribution of "age" and "category" represent the population properly. Specifically, we may use t-test to compare difference of means between our sample and other sources of youtube studies. Should there be a signifiant difference, we should consider to reweight the values of "views", "rate", "length" and "ratings" accordingly.

```
hist(df$length, breaks = "FD")
axis(1, at = seq(0, 5300, 500))
```



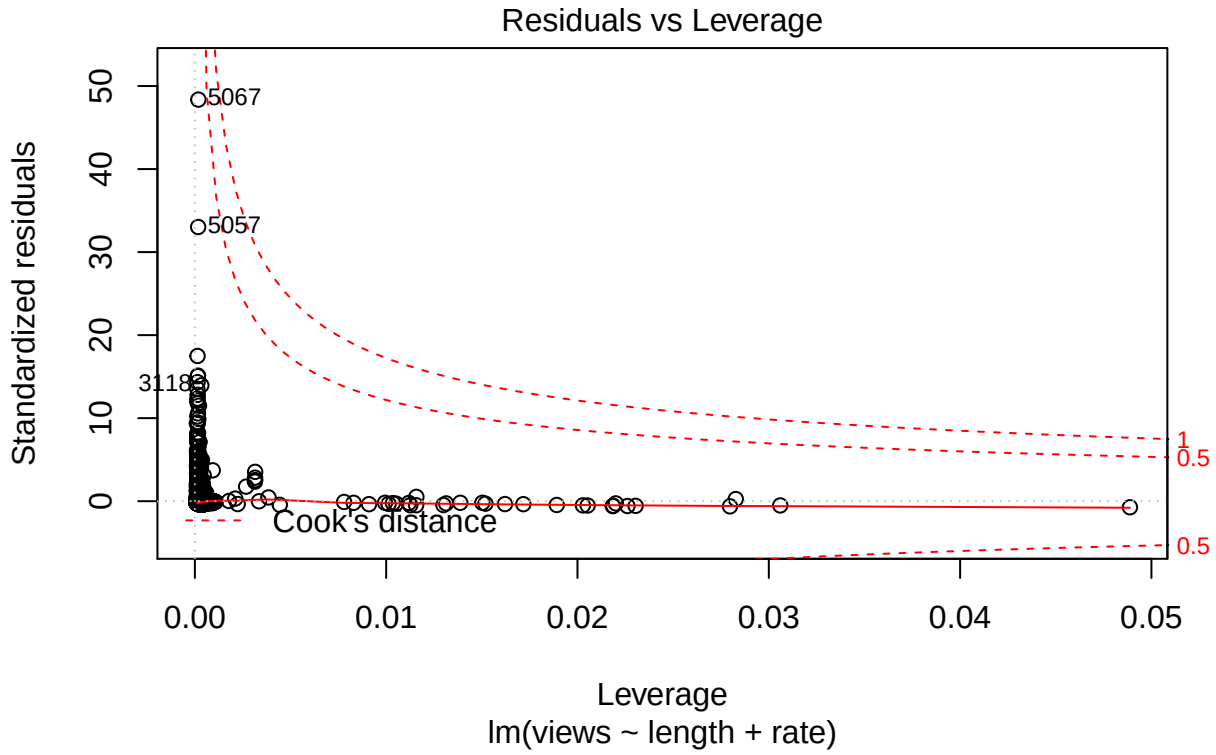
- "rate" has a noticable leptokurtosis(3.2066301) and moderate negative skew(-1.3529164) . Most values are either 5 or 0. This doesn't seem to reflect the true quality of video. Rather, the distribution reflects a self-selection problem in the collection of data: people who have strong feelings for the videos are more likely to give a rating than people who don't. On average, only a fraction(0.007) of viewers leave a rating on a video. Even the most rating provoking video has less than half of viewers leaving a rating. Thus, the values under this variable are highly biased and will lead to biased estimates for the coefficient of "rate".
- One correction is to weight each value under "rate". Each value can be multiplied by the likelihood someone will take action to leave a rating given he/she feels a certain rating after watching a youtube video.

```
hist(df$rate, breaks = "FD", main = "rate")
```

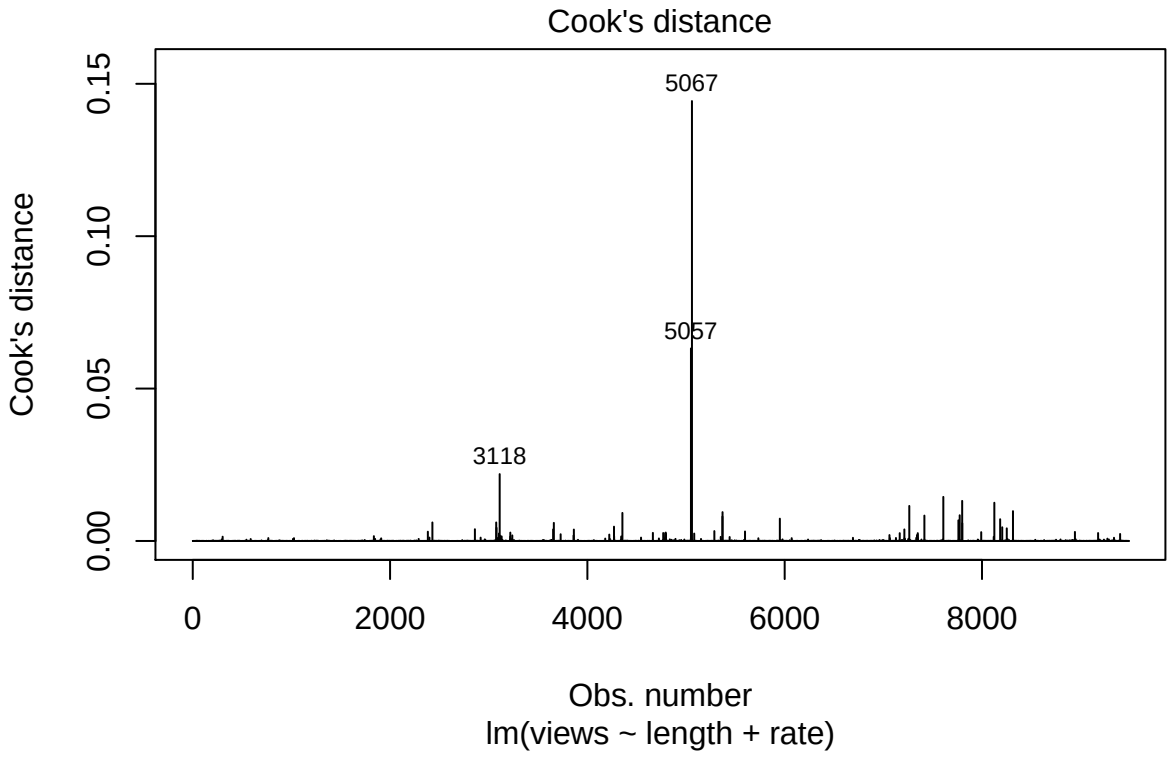


- We should also check if our coefficients are affected severely by extreme outliers. Two points 5057 and 5067, came close to the Cook's distance boundary on the residuals vs plot. Many residuals also have high standardized values, indicating that our model has too much error and represent our data poorly.

```
plot(model, which = 5)
```

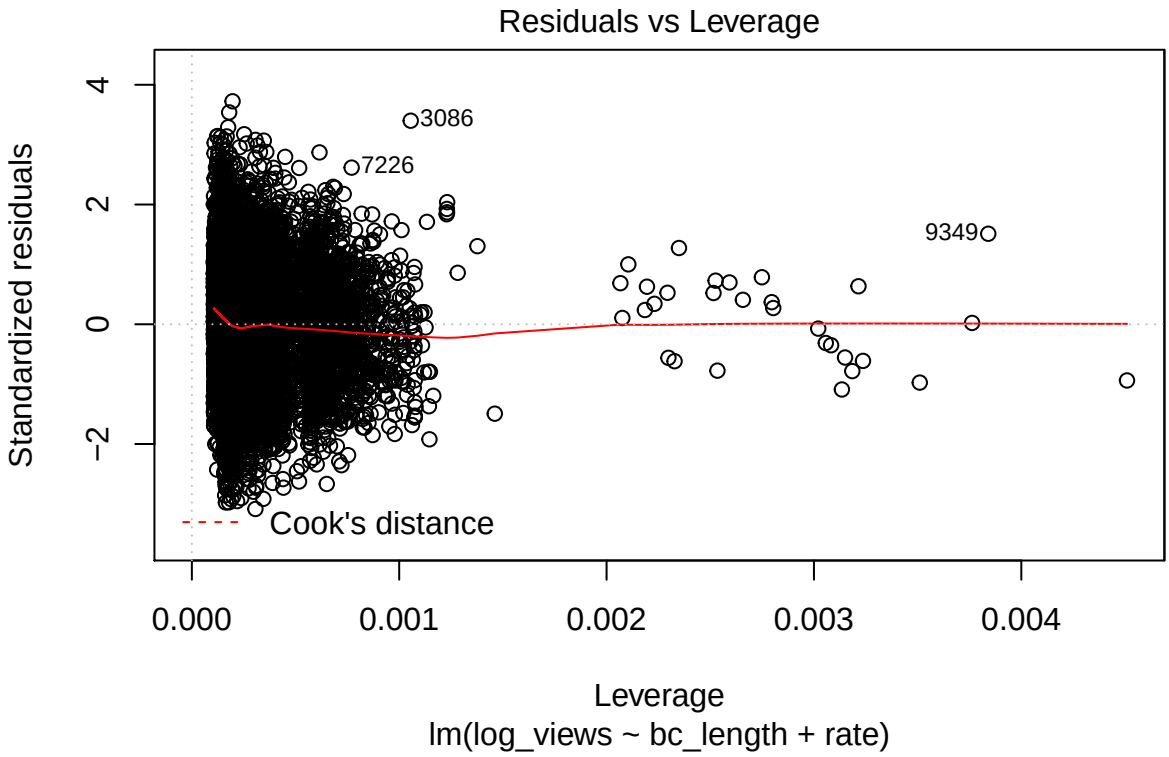


```r
plot(model, which = 4)
```

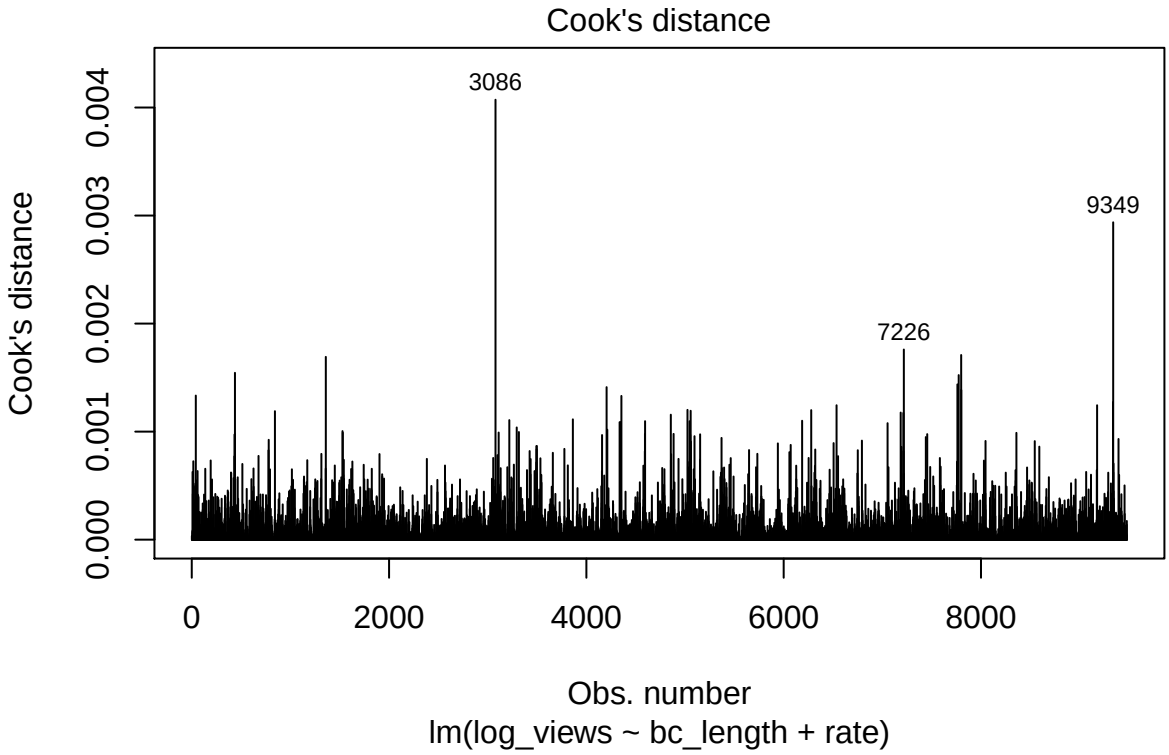


- Our transformed model is doing much better. Although a good proportion of cases still have standardized residual values greater than 2 and 3. We should recommend a revision of the researcher's sampling technique.

```r
plot(model_transformed, which = 5)
```




```
plot(model_transformed, which = 4)
```



3. No perfect Multi-collinearity

- We can use the correlation, variance inflation factor and scatter plot to examine if the two predictor variables are highly correlated.

- The two predictors seem weakly correlated, it seems like most outliers of "length" has an above average "rating" (3.7455591).
- The variance inflation factor also indicates that variance of estimated coefficients are not highly inflated compared to when the predictor variables are not linearly correlated.

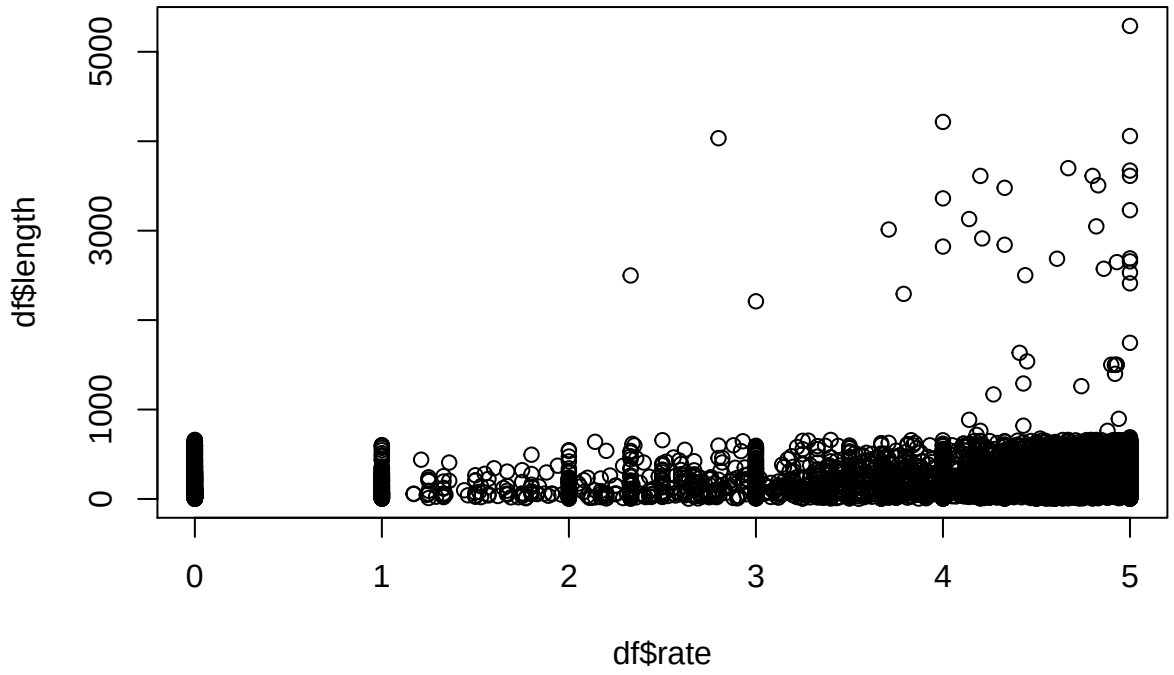
```
cor(df$length, df$rate, use = "complete.obs")
```

```
## [1] 0.1583326
```

```
car::vif(model)
```

```
##   length     rate
## 1.025714 1.025714
```

```
plot(df$length ~ df$rate)
```



- Evaluations for our transformed model is similar. The predictor variables become slightly more correlated, but not enough to worry about.

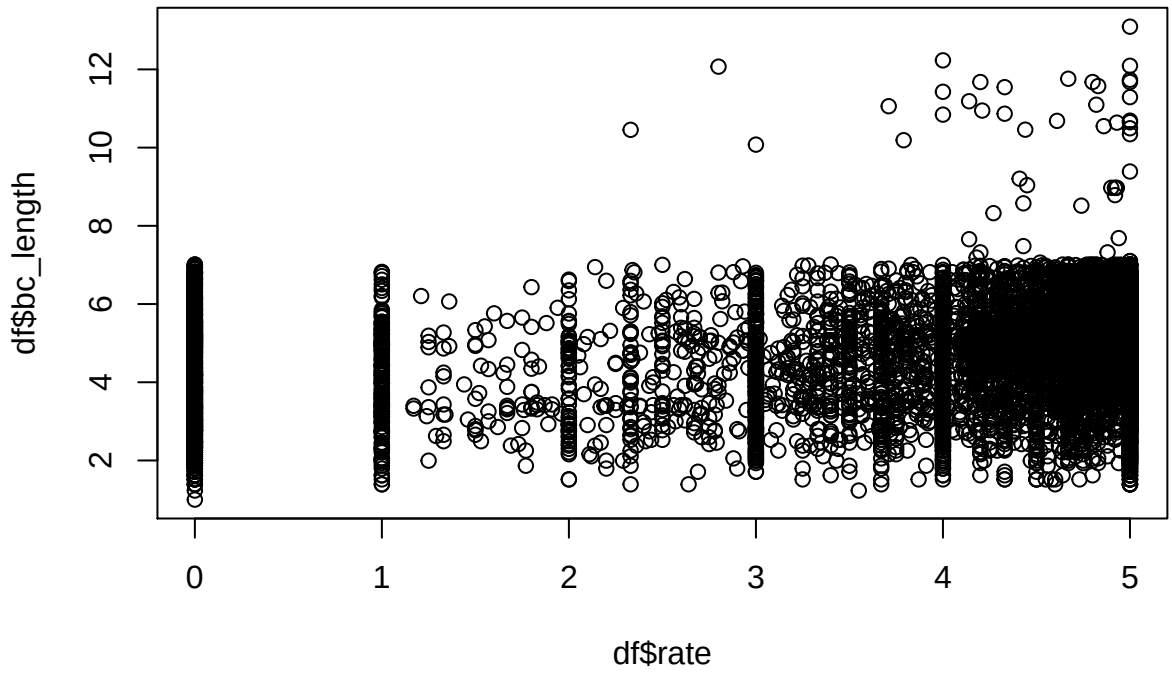
```
cor(df$bc_length, df$rate, use = "complete.obs")
```

```
## [1] 0.2413944
```

```
car::vif(model_transformed)
```

```
## bc_length      rate
##  1.061877  1.061877
```

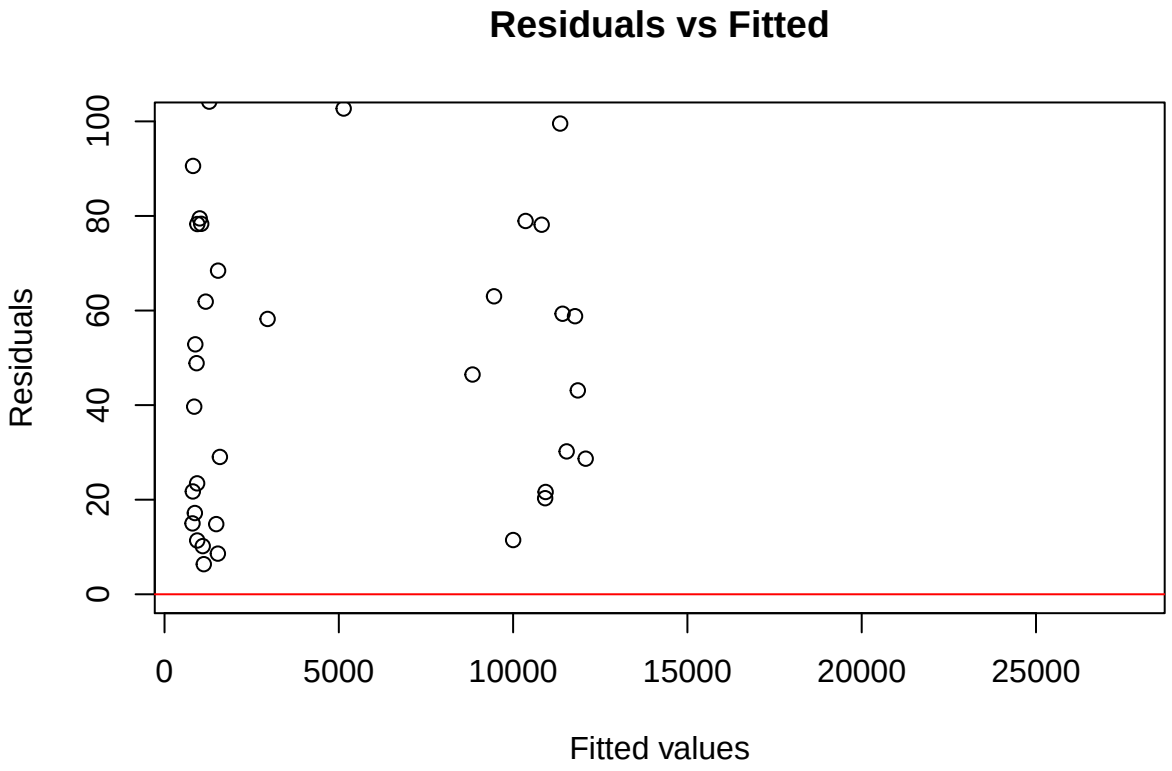
```
plot(df$bc_length ~ df$rate)
```



4. Zero-Conditional Mean

- All residuals deviate noticably above zero, there's a clear violation of zero-condition mean. Our estimated coefficients are biased. We should resort to exogeneity for a consistent estimator instead.

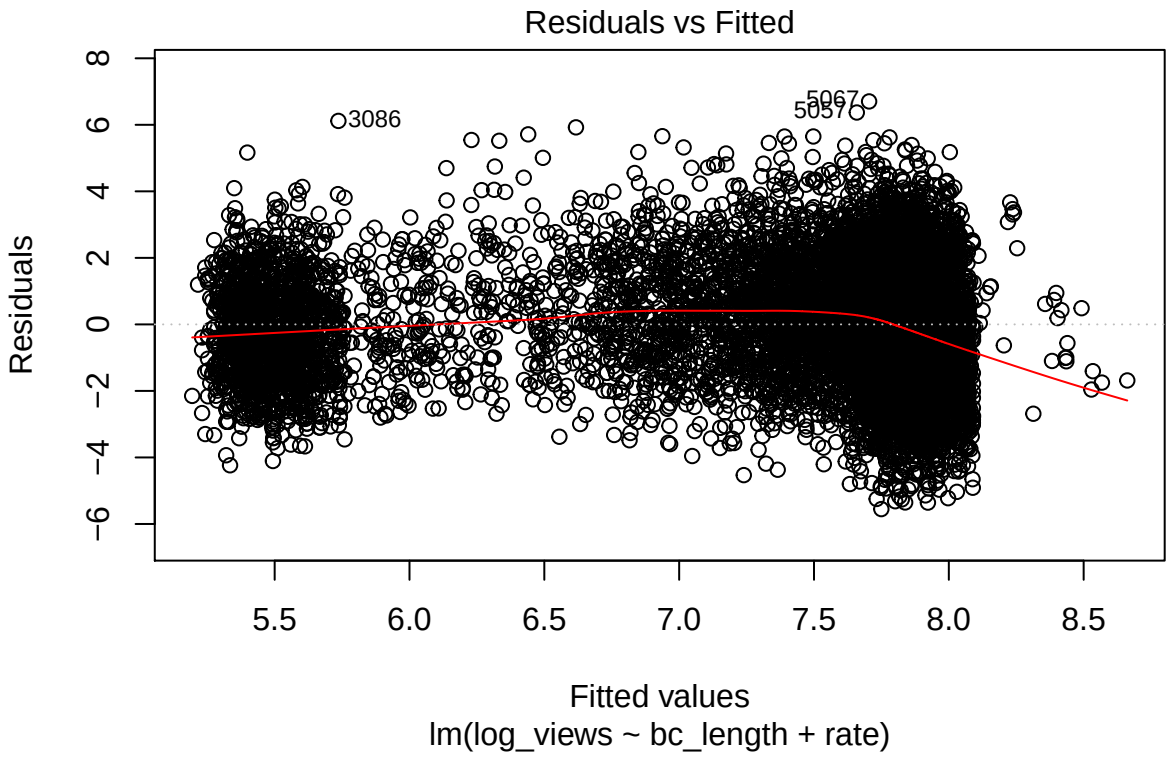
```
plot(model$residuals~model$fitted.values, ylim = c(0,100), main = "Residuals vs Fitted", ylab = "Residua
abline(a = 0, b = 0, col = "red")
```



- The transformed model performed much better, although the mean of residuals given fitted values still

deviate above and below the zero level where we have lots of data points. We should also resort to exogeneity for this model.

```
plot(model_transformed, which = 1)
```



4'. Exogeneity

- If we are only interested in an associative model to find the line which best fit our dataset, we have exogeneity. Because OLS estimators, by minimizing squared residuals, ensures $cov(\hat{u}, x) = 0$ for a given dataset. In a nut shell, data and OLS estimator ensures exogeneity here.
- However, if we are interested in causal inference, we will need to rebuild the model with a casual structure informed by designed experiment, which will give us very different residual patterns. In such models, exogeneity is rare. To demonstrate that our current predictors will not yield exogeneity in a causal model, we can run the regression with an additional variable, "ratings". Notice that the coefficient for "rate" dropped from 2105.45 to 370.70, meaning "rating" can be a much closer cause for "views" than "rate". Thus our causal model will be endogenous for sure.

```
model_ratings = lm(views ~ length + rate + ratings, data = df)
model_ratings
```

```
##
## Call:
## lm(formula = views ~ length + rate + ratings, data = df)
##
## Coefficients:
## (Intercept)       length         rate      ratings
##    1683.400       -4.672      301.366      370.699
```

- Our log transformed model also gave lower coefficients for power-transformed "length" and "rate", indicating that the inclusion of "rating" explained away some variance in log of "views". "length" and "rate" have a lower effect on our estimates for "views" now.

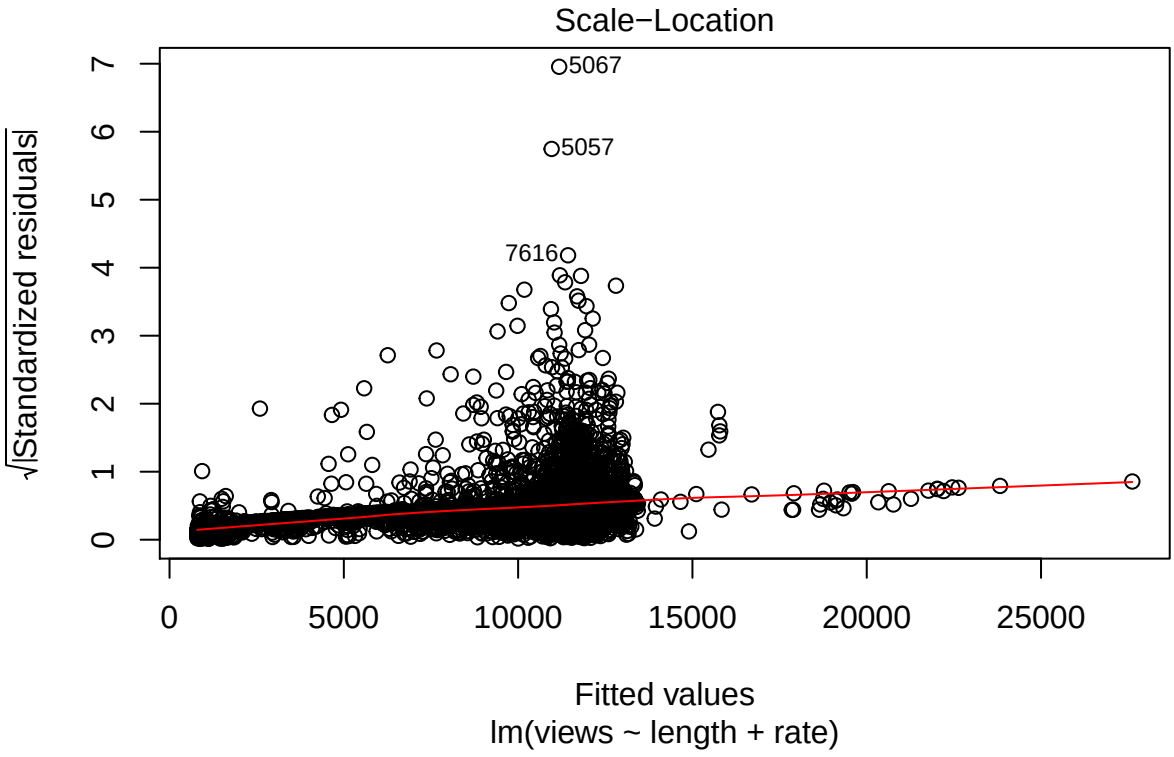
```
model_ratings_transformed = lm(df$log_views ~ df$bc_length + df$rate + df$ratings)
model_ratings_transformed
```

```
##
## Call:
## lm(formula = df$log_views ~ df$bc_length + df$rate + df$ratings)
##
## Coefficients:
##  (Intercept)  df$bc_length       df$rate    df$ratings
##     5.269963      0.050343      0.425971      0.009005
```

5. Constant Variance of Error Term: Homoskedasticity

- From the Scale-Location plot, the standardized residuals noticably picks up as fitted values increases until fitted value reaches 14000. The sign of heteroskedasticity is strong.

```
plot(model, which = 3)
```



- Notice that the Breusch-Pagan test failed to reject the null hypothesis of homoskedasticity regardless our large sample size. One explanation is that the Breusch Pagan test regress the model's residuals on the predictor variables, thus evaluate the linear relationship between the two. Since our residuals drop sharply around the fitted value of 14000, the overall linear relationship between the residuals and predictor variables is not strong. Therefore, Breusch Pagan test failed to pick up heteroskedasticity.

```
lmtest::bptest(model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 3.4552, df = 2, p-value = 0.1777
```

- Notice that our heteroskedasticity robust standard errors are also dramatically smaller than regular standard errors. One explanation is that robust standard errors depend on the observations at the far ends of fitted values. Since our residuals have thin and roughly even tails, robust standard errors will miss the larger variance in the center thus produce smaller estimates of coefficient variance. We should depend on the more conservative, regular standard error instead of robust standard errors.

```
summary(model)
```

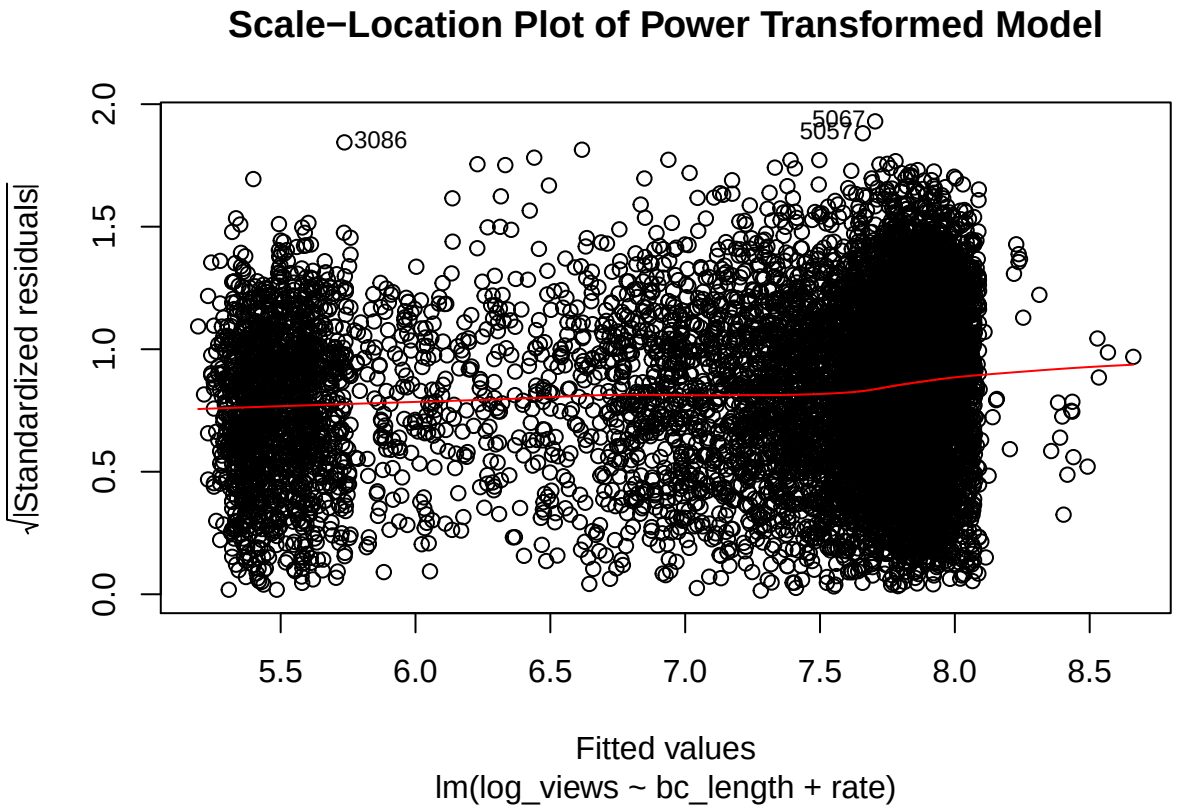
```
##
## Call:
## lm(formula = views ~ length + rate, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -26547  -10277   -6556    -833 1796463
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  789.683    917.714   0.860   0.3895
## length         3.082      1.628   1.893   0.0584 .
## rate        2105.454    216.082   9.744   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37150 on 9477 degrees of freedom
## Multiple R-squared:  0.01117,    Adjusted R-squared:  0.01096
## F-statistic: 53.53 on 2 and 9477 DF,  p-value: < 2.2e-16
```

```
vcovHC_matrix = sandwich::vcovHC(model)
lmtest::coeftest(model, vcov = vcovHC_matrix )
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  789.6825   281.1757  2.8085  0.004987 **
## length         3.0822     1.2515  2.4628  0.013804 *
## rate        2105.4545   128.1371 16.4313 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Our power transformed model gives residuals of much more stable variance here. There's a slight uptake towards the positive end of fitted values, where we have a lot of data.

```
plot(model_transformed, which = 3, title("Scale-Location Plot of Power Transformed Model"))
```



- The Breusch Pagan test gives a statistcally significant result for heteroskedasticity, despite a healthy Scale-Location Plot. One explanation is that with our large sample, the test will show statiscal significance even for negligible deviations.

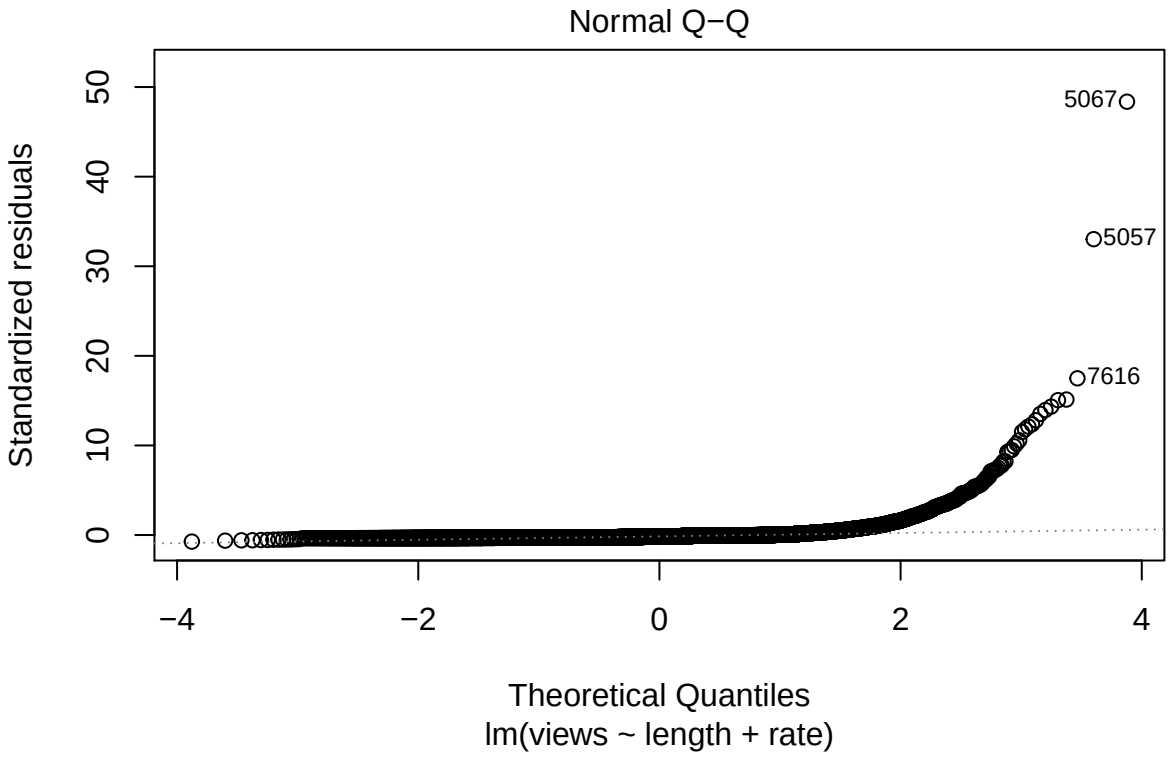
```
lmtest::bptest(model_transformed)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model_transformed
## BP = 122.26, df = 2, p-value < 2.2e-16
```

6. Normal distribution of Error Term

- Our shape measures for residuals indicate extreme positive skew (measures 20.7817382) and leptokurtosis(measures 760.5737714). The QQ plot also show clear take off of residuals from the perfect diagonal.

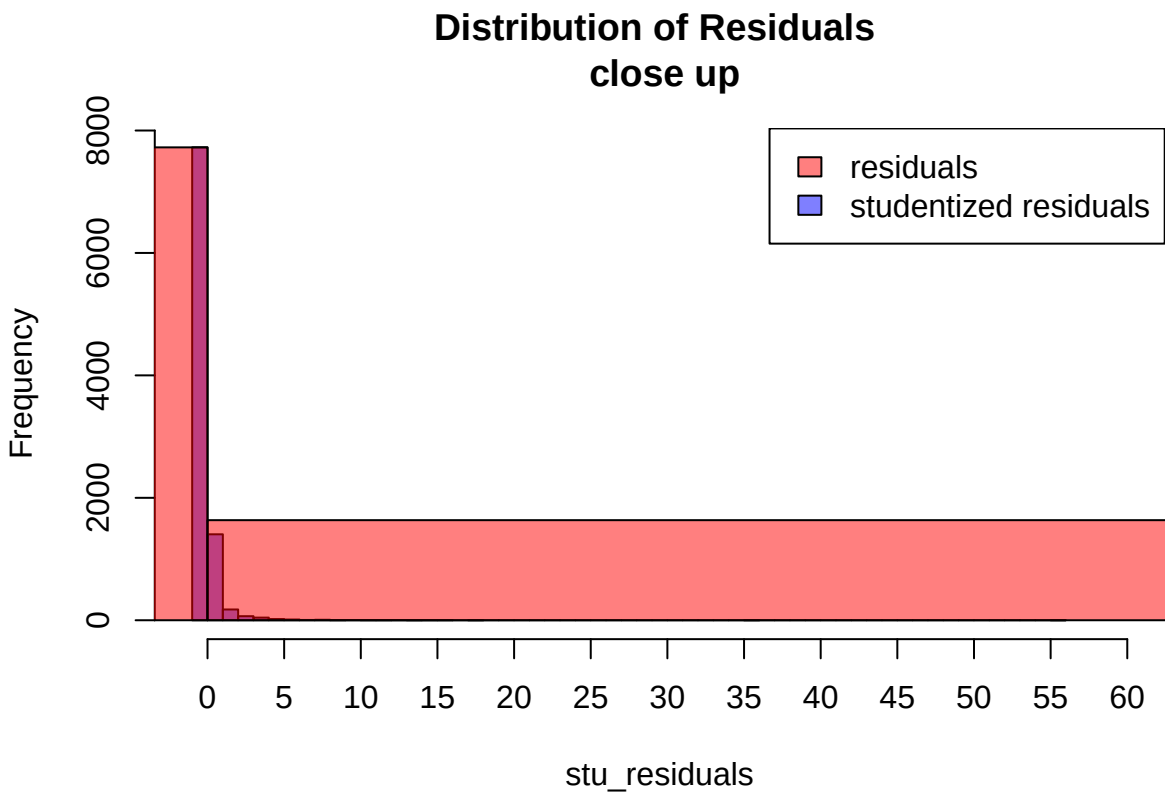
```
plot(model, which = 2)
```



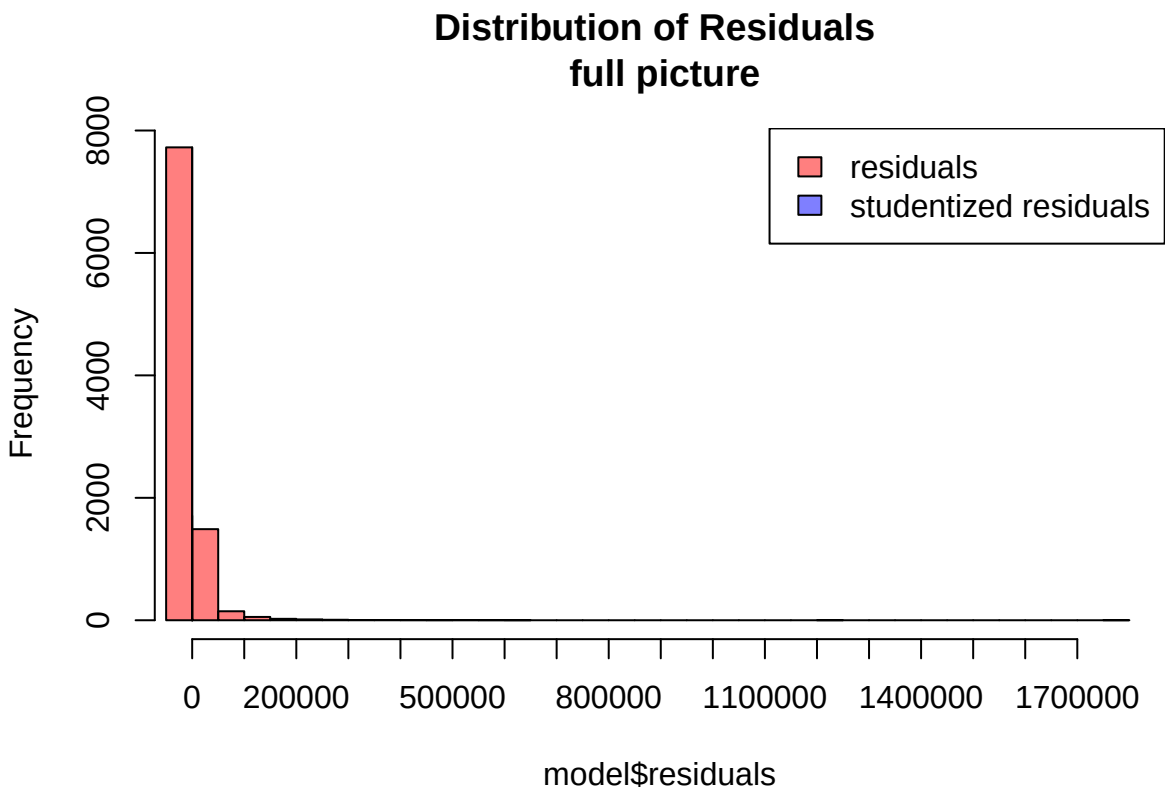
- Both residuals and studentized residual plots show highly skewed distributions. Notice the dramatic difference between the range of residuals as-is and studentized residuals. This is because of the high standard error of our residuals.

```
# studentized residual daa
stu_residuals = MASS::studres(model)

# plot residuals
hist(stu_residuals,  col = rgb(0,0,1,.5), xaxt = "n", xlim = c(-1, 60),
     main = "Distribution of Residuals \n close up", breaks = 50 )
hist(model$residuals,  col=rgb(1, 0, 0, .5), xaxt = "n", add = T)
axis(1, at = seq(0, 60, 5))
legend("topright", legend = c("residuals", "studentized residuals"),
       fill = c(rgb(1, 0, 0, .5), rgb(0,0,1,.5)))
```

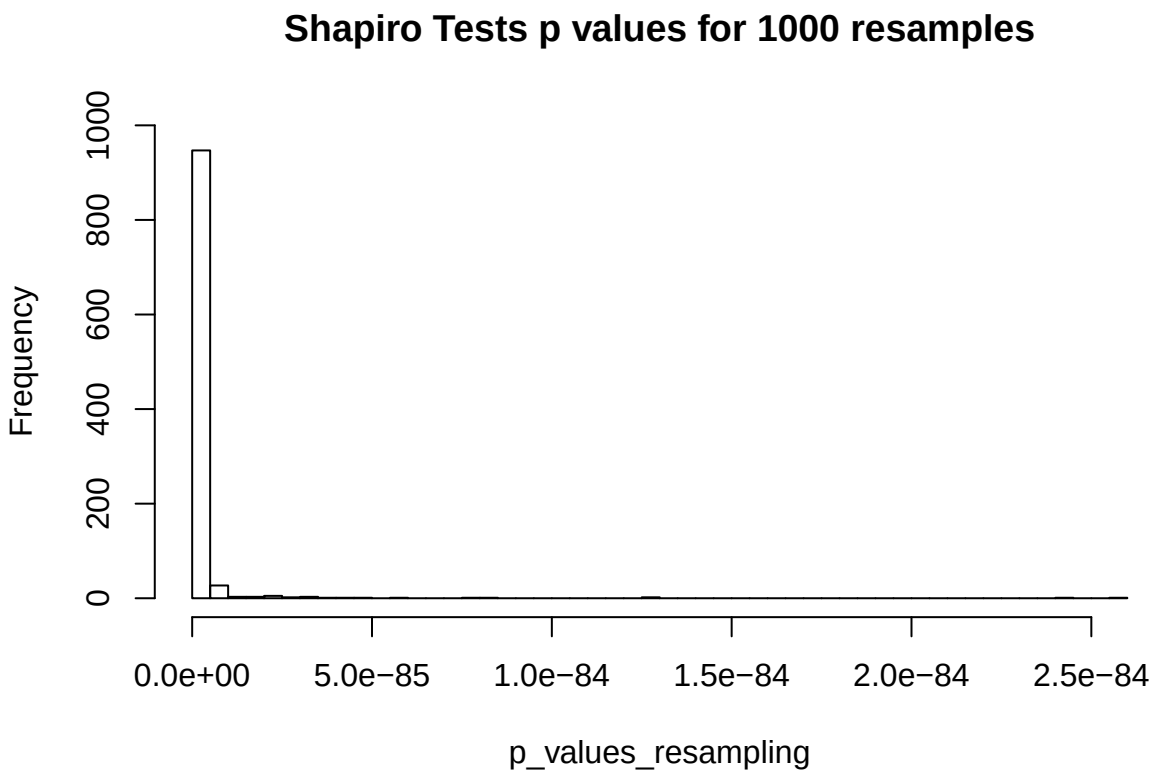



```
# plot residuals
hist(model$residuals,  main = "Distribution of Residuals \n full picture", col=rgb(1, 0, 0, .5),
     xaxt = "n", breaks = 50, xlim = c(-1, 1796000))
hist(stu_residuals,  col = rgb(0,0,1,.5), xaxt = "n", add = T)
axis(1, at = seq(0, 1796000, 100000))
legend("topright", legend = c("residuals", "studentized residuals"),
       fill = c(rgb(1, 0, 0, .5), rgb(0,0,1,.5)))
```



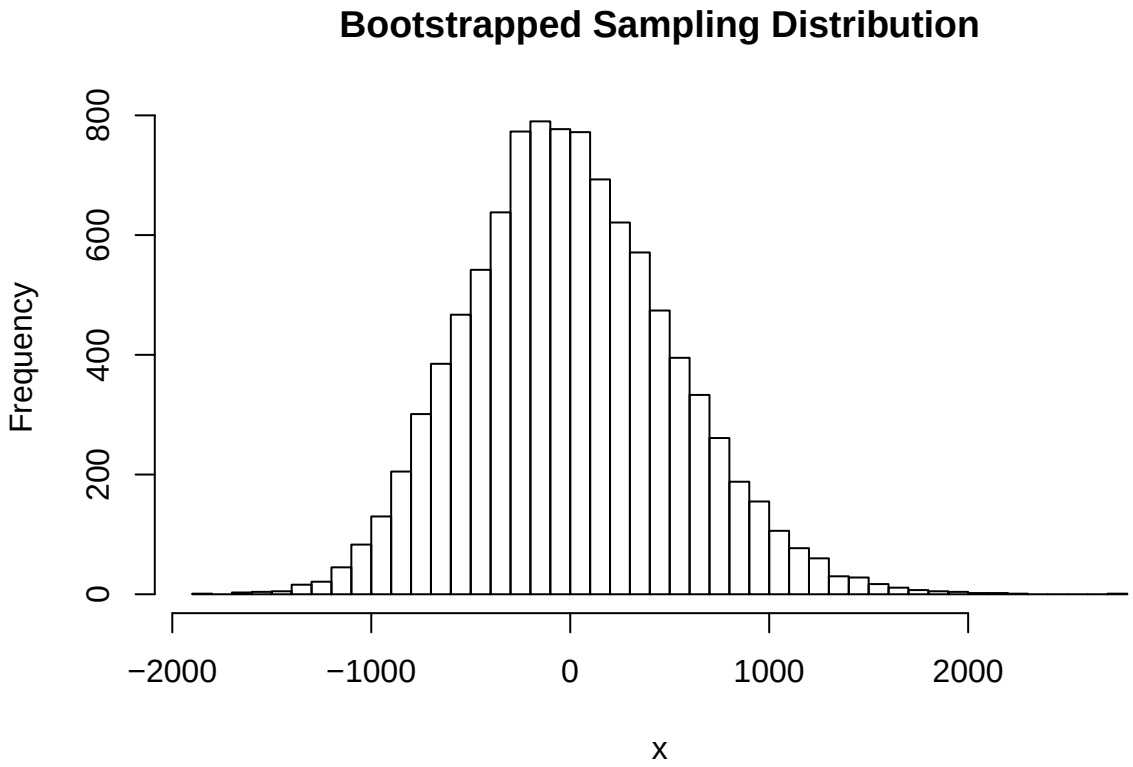
- Since Shapiro-Wilk test is not supported in R for sample size above 5000, we will run it with 1000 random resamples each of 5000 observations. The test results give 1000 p values that are very close to zero, showing that our test result is robust. Normality of error term is clearly violated.

```
# p values of shapiro tests on 1000 resamples
p_values_resampling = replicate(1000, shapiro.test(sample(model$residuals, 5000,
                                                          replace = TRUE, prob = NULL))$p.value)
hist(p_values_resampling, main = "Shapiro Tests p values for 1000 resamples",
     breaks = 50, ylim = c(0,1000))
```



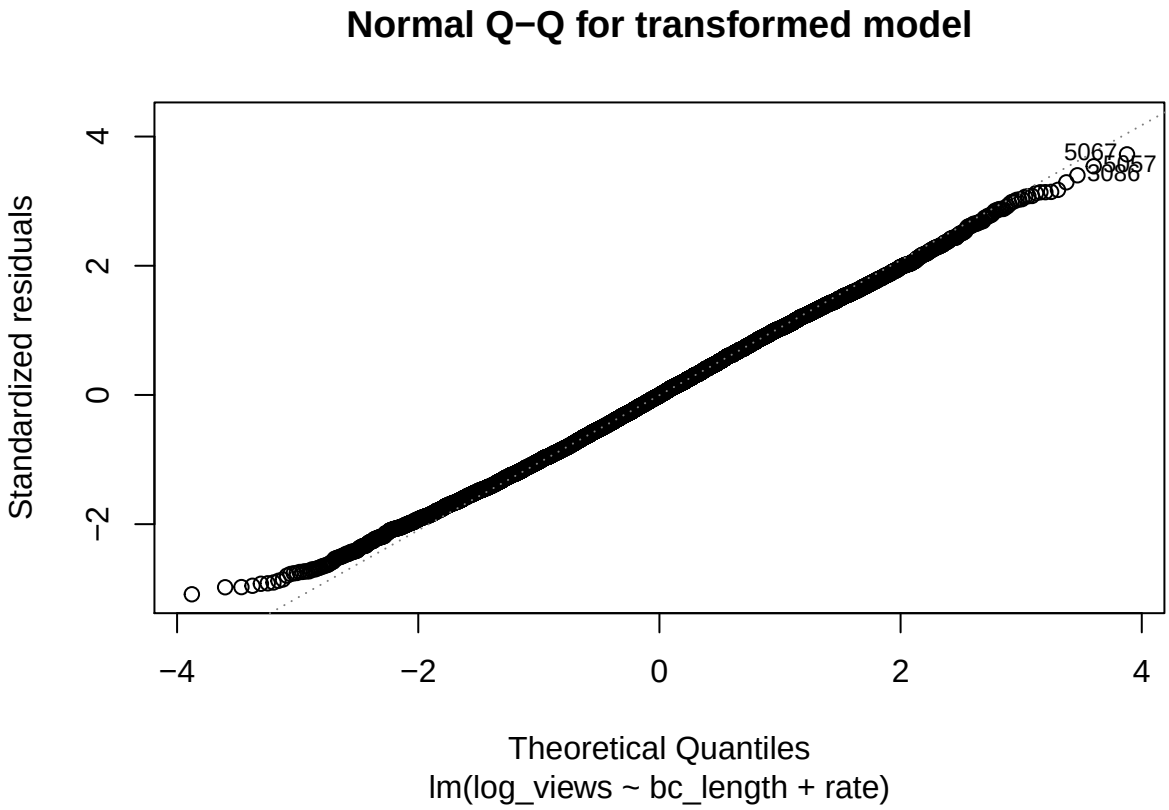
- We may be able to rely on OLS asymptotics, since our sample size is quite large (9480 observations). To simulate sampling distribution of our error term, we can resample from our dataset (bootstrapping). Our pseudo sampling distribution is much more normal, hinting that OLS asymptotics for our sample will work.
- Alternatively, we may consider other generalized linear models which assume other distributions for our residuals, such as zero-inflated negative binomial regression which is more similar to our "views" distribution.

```
x = replicate(10000, mean(sample(model$residuals, 5000, replace = TRUE, prob = NULL)))
hist(x, breaks = 50, main = "Bootstrapped Sampling Distribution")
```



- In our power transformed model, our shape measures for residuals indicate much more tamed skewness(measures 0.0544891) and lepokurtosis(measures 2.7794072). The QQ plot also show better conformation to the perfect diagonal.

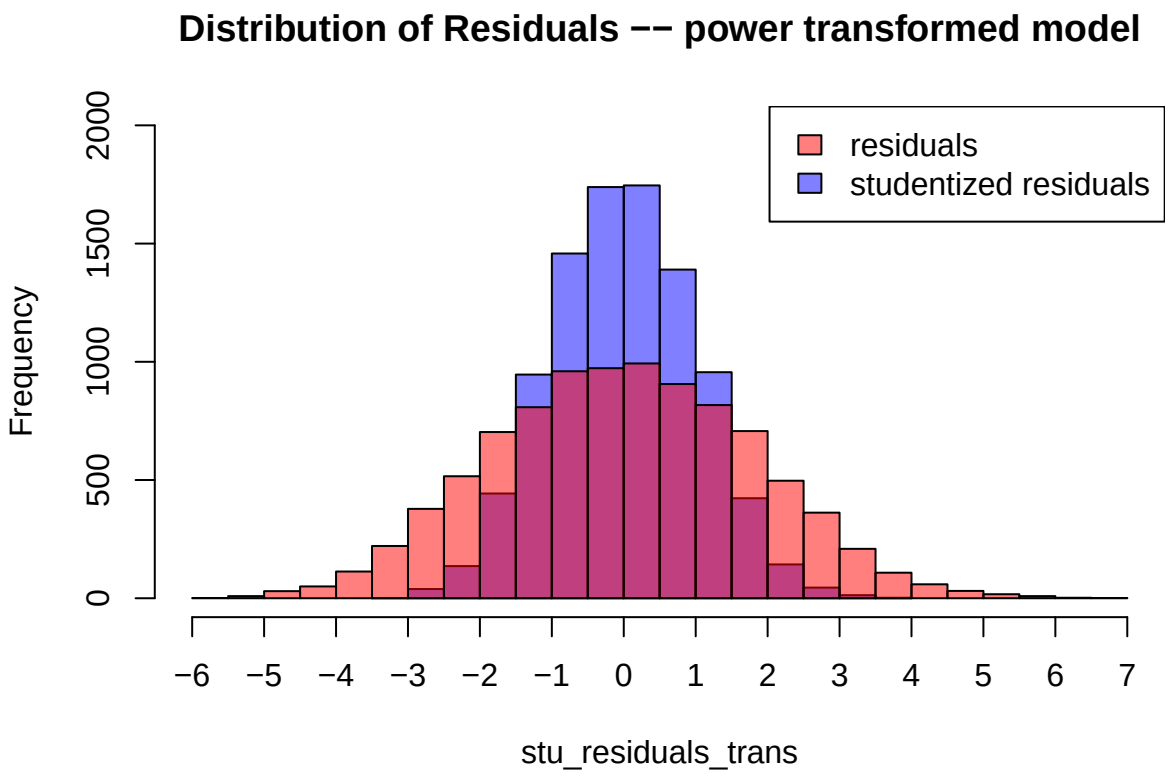
```
plot(model_transformed, which = 2, title("Normal Q-Q for transformed model"))
```



- Both normalized residuals and studentized residual plots show close to normal distributions. Our power transformed model is complying much better with this assumption.

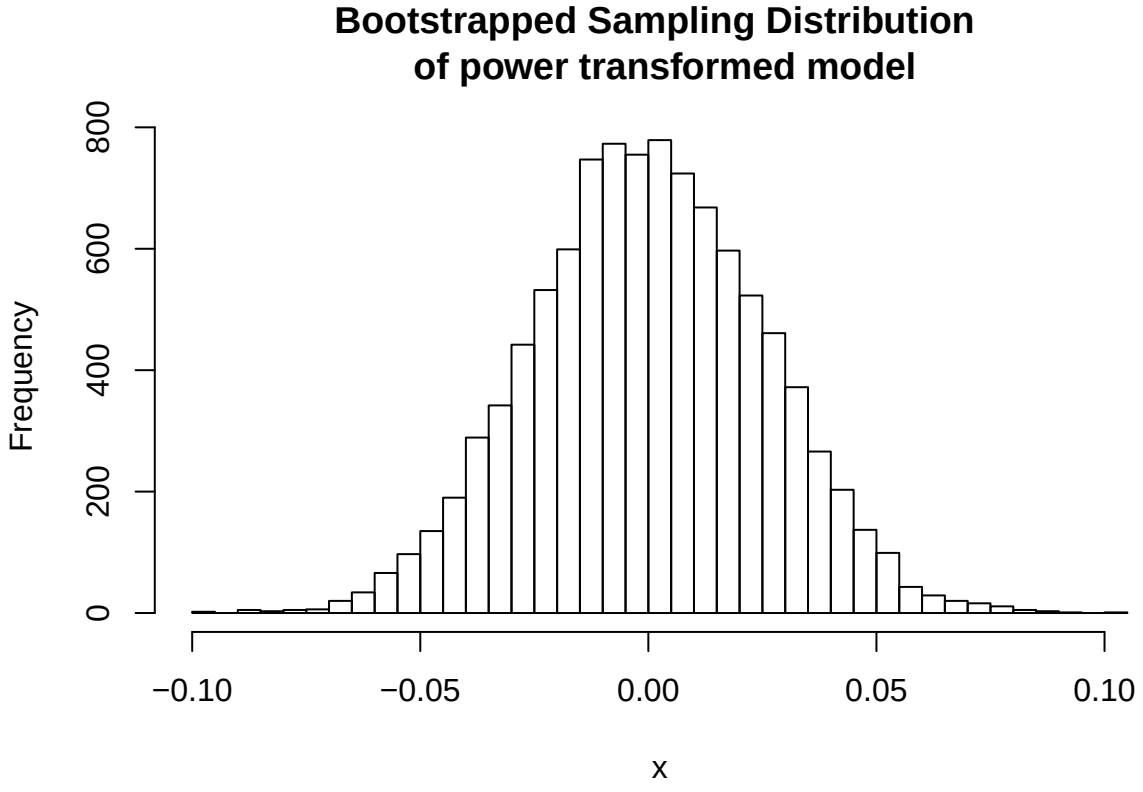
```
# studentized residual daa
stu_residuals_trans = MASS::studres(model_transformed)

# plot residuals
hist(stu_residuals_trans,  col = rgb(0,0,1,.5), xaxt = "n", xlim = c(-6, 7),  main = "Distribution of R
hist(model_transformed$residuals,  col=rgb(1, 0, 0, .5), xaxt = "n", add = T , breaks = 20 )
axis(1, at = seq(-6,7, 1))
legend("topright", legend = c("residuals", "studentized residuals"), fill = c(rgb(1, 0, 0, .5), rgb(0,0
```



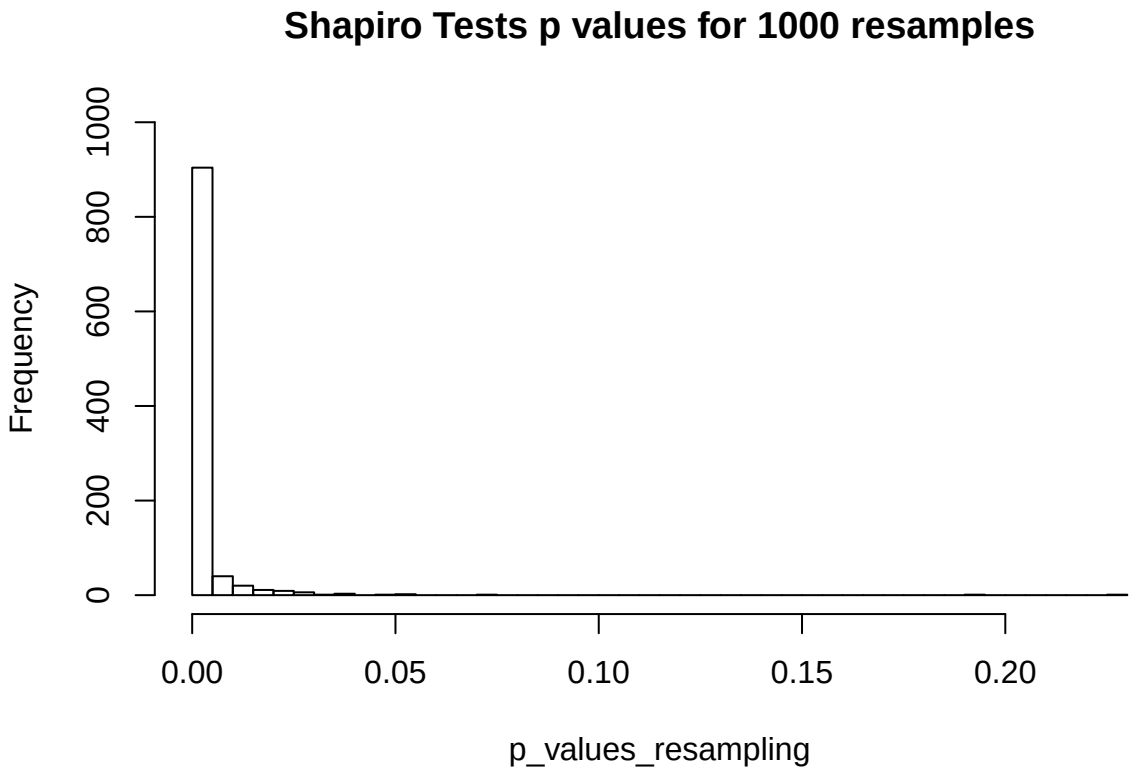
- The Bootstrapped sampling distribution of our power transformed model also confirms that we can rely on OLS asymptotics.

```
x = replicate(10000, mean(sample(model_transformed$residuals, 5000, replace = TRUE, prob = NULL)))
hist(x, main = "Bootstrapped Sampling Distribution \n of power transformed model", breaks = 50 )
```



- Using similar procedures as above, the Shapiro-wilk test output most p values under 0.05, which is contrary to our intuition. Note that because our sample size is large, even simulated data drawn from normal distribution can produce significant results with the Shapiro-wilk test. We should thus take the results with a grain of salt. The fact that the range of p values here is much less extreme than that of the model without power transformation, it is already a good sign.

```
# p values of shapiro tests on 1000 resamples
p_values_resampling = replicate(1000, shapiro.test(sample(model_transformed$residuals, 5000,
                                                          replace = TRUE, prob = NULL))$p.value)
hist(p_values_resampling, main = "Shapiro Tests p values for 1000 resamples",
     breaks = 50, ylim = c(0,1000))
```



Question 3)

Generate a printout of your model coefficients, complete with standard errors that are valid given your diagnostics. Comment on both the practical and statistical significance of your coefficients.

Answer 3)

- For our model without power transformation, coefficient of "length" is not statistically significant, given its small coefficient and usual magnitude of "views", its coefficient is not practically significant either. Coefficient of "rate" is statistically significant. Its interpretation – 2105 more views for every unit increase in rating, is also practically significant.
- For our model with power transformation, all coefficients are statistically significant. The interpretation of the coefficients – 9.4% increase in views for every unit increase in cubic root of video length and 59.52% increase in views for every unit increase of average rating – are also practically significant.

```r
stargazer::stargazer(model, model_transformed, type = "text", omit.stat = "f",
                     title = "Linear Models Predicting Views",
                     add.lines = list(c("AIC", round(AIC(model),0) , round(AIC(model_transformed),0))),
                     star.cutoffs = c(0.05, 0.01, 0.001),
                     column.labels = c("no transformation", "power transformed"),
                     model.names = F)
```

```
##
## Linear Models Predicting Views
## ================================================================
##                                        Dependent variable:
##                                    -----------------------------
##                                          views          log_views
##                                    no transformation power transformed
##                                           (1)               (2)
## ----------------------------------------------------------------
## length                                   3.082
##                                         (1.628)
##
## bc_length                                                0.094***
##                                                           (0.015)
##
## rate                                   2,105.454***       0.466***
##                                          (216.082)         (0.011)
##
## Constant                                 789.683          5.100***
##                                         (917.714)          (0.073)
##
## ----------------------------------------------------------------
## AIC                                      226416             38052
## Observations                              9,480             9,480
## R2                                        0.011             0.190
## Adjusted R2                               0.011             0.190
## Residual Std. Error (df = 9477)       37,145.040            1.800
## ================================================================
## Note:                                  *p<0.05; **p<0.01; ***p<0.001
```