HW Week 12

W203: Statistics for Data Science

Answer Key

OLS Inference

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

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

Apologies

Apologies I have over done this in the interesting of learning all the techniques ahead of Lab 4.

Load Data

First read in the data into a table noting that the data is a text file that is tab separated with a header row with variable names.

```r
#setwd("~/Desktop/W203/HW12/")
setwd("/Users/8ps/Google\ Drive/_UCB_W203/W203\ Update/weekly_materials/week_12/HW12/")
Data = read.table("videos.txt", sep="\t", header=T)
head(Data)
```

```
##      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 Elp6BfOHJIM  Fizz101Productionz 1252        Comedy     26   392 1.50
## 6 VPuKu7aU9GY         slytherin66 1236 Entertainment    252   318 5.00
##   ratings comments
## 1       2        1
## 2      11        4
## 3      81       36
## 4      24       13
## 5       8       17
## 6       2        3
```

The `head()` function indicates that the data has read in successfully.

Exploratory Data Analysis

I first reduce the data table down to the three variables of interest (i.e. `views`, `length` and `rate`) and then filter the data to remove null values and videos that are unrated to ensure that there is an observation for each variable in the regression:

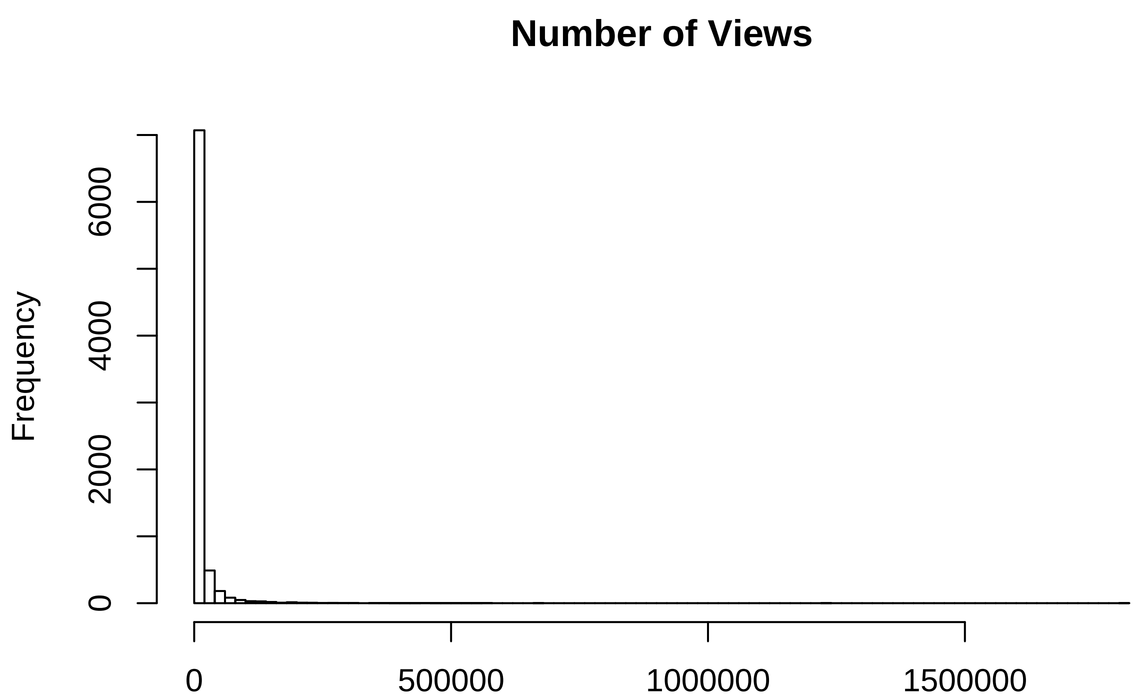
```
# filter the data to remove unrated videos
D = Data[Data$ratings>0,]
D = D[,(5:7)]
# filter the data table to only include rows with 3 non-null observations
filter = !is.na(D$length) | !is.na(D$views) | !is.na(D$rate)
D = D[filter,]
summary(D)
```

```
##      length            views               rate      
##  Min.   :   2.0   Min.   :      9   Min.   :1.000  
##  1st Qu.: 100.0   1st Qu.:    577   1st Qu.:4.220  
##  Median : 204.0   Median :   2149   Median :4.800  
##  Mean   : 239.2   Mean   :  10971   Mean   :4.431  
##  3rd Qu.: 310.0   3rd Qu.:   7764   3rd Qu.:5.000  
##  Max.   :5289.0   Max.   :1807640   Max.   :5.000  
```

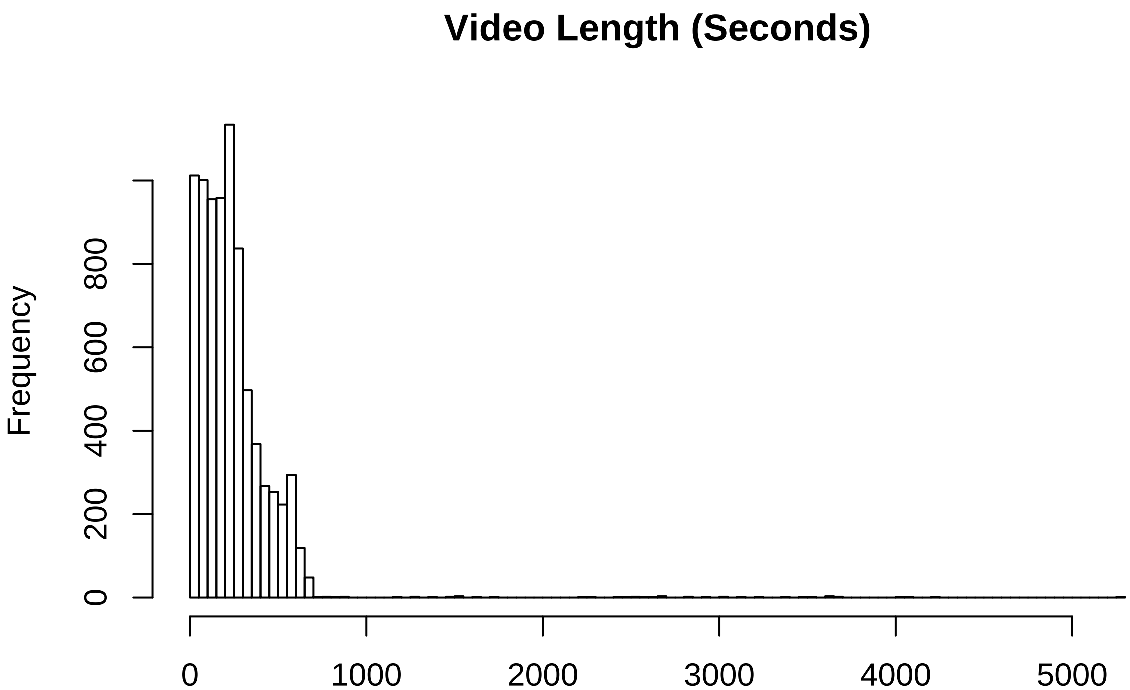
The `summary()` function at a high level indicates that there are no null values as expected, `views` and `length` are non-zero positive variables, while `rate` is a score in the range 1 to 5 and there doesn't appear to be any values of unexpected magnitude.

To further review the variables I next check their histograms:

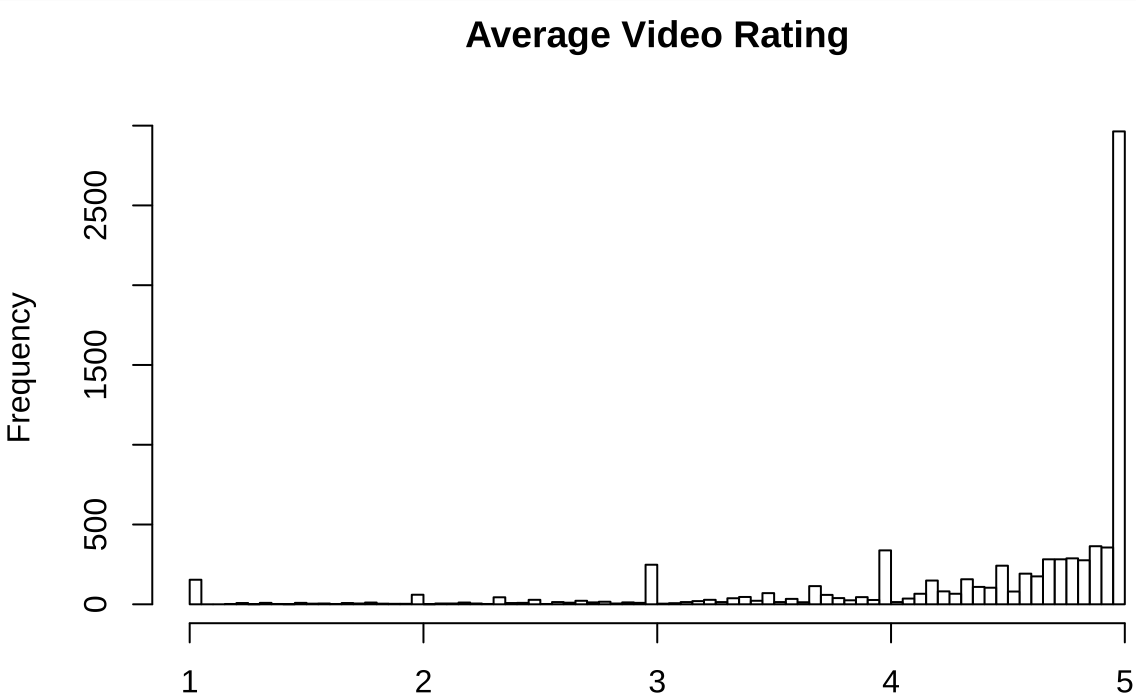
```
hist(D$views, main="Number of Views", xlab=NULL, breaks=100)
```



```r
hist(D$length, main="Video Length (Seconds)", xlab=NULL, breaks=100)
```




```r
hist(D$rate, main="Average Video Rating", xlab=NULL, breaks=100)
```



Observations:

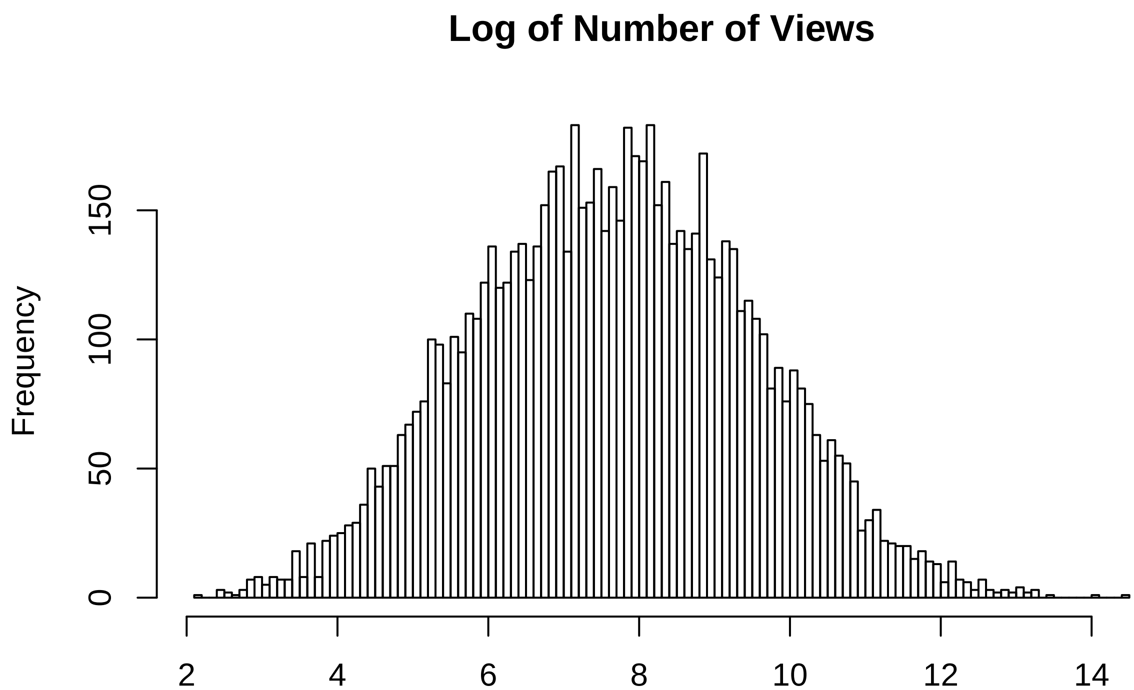
`views`: The histogram indicates that `views` has a significant positive skew. So given that number of views by definition is a non-zero, positive integer a log transformation will likely be beneficial.

`length`: The histogram of `length` also has a significant positive skew, but it is seems to have a somewhat truncated distribution at a `length` of around 750. Without a code book describing the variables it appears that the `length` variable is measured in seconds, so to make it more intuitive to the reader it is probably worth transforming the data into minutes by dividing by 60 for further observation. It may also be appropriate to use a log transformation given the positive skew, but with consideration of the apparent truncation of the distribution of video `length`, which is possibly reflective of clustering.

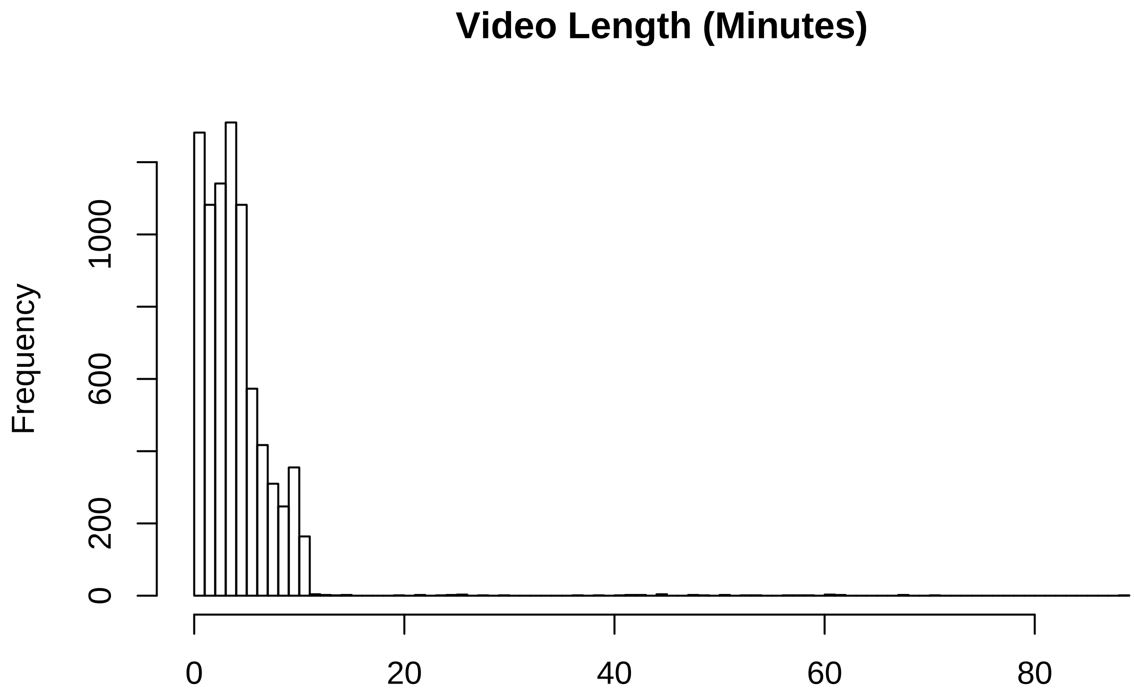
`rate`: The histogram of `rate` has a very unusual profile. Among other things: (i) it has a negative skew, (ii) there are a lot of videos that have an exact integer rating compared to smaller numbers of ratings between two adjacent integer ratings, and (iii) the maximum rating of 5 is by far the most frequent or modal rating. The `head()` function output on `rate` above shows that the variable is a score between 1 and 5 to two decimal places indicating that it is some sort of average rating. This will be discussed further when considering whether or not the 6 assumptions of the CLM hold.

Following the first stage of data analysis I now perform a log transformation of `views` and `length` and plot `length` in minutes.

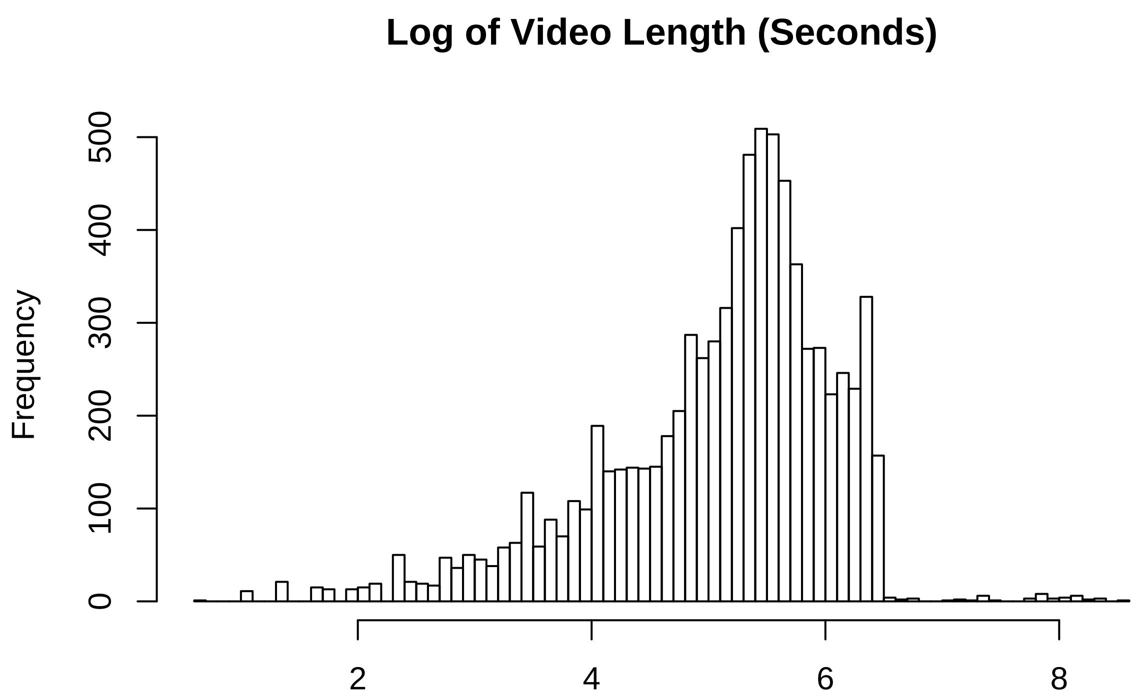
```
hist(log(D$views), main="Log of Number of Views", xlab=NULL, breaks=100)
```




```
hist(D$length/60, main="Video Length (Minutes)", xlab=NULL, breaks=100)
```




```r
hist(log(D$length), main="Log of Video Length (Seconds)", xlab=NULL, breaks=100)
```



Observations:

`log(views)`: The log transformation has made the distribution of `log(views)` quite normal. This will help to ensure that the errors of the model are normal (i.e. CLM 6)

`length/60`: Converting `length` to minutes indicates that the distribution is truncated around a length of 12 minutes.

`log(length)`: The log transformation of `length` more clearly shows the truncation in the distribution. The non-truncated portion of the distribution is somewhat normal. This will be discussed further when considering whether or not the 6 assumptions of the CLM hold.

Proposed Model

Based on the exploratory data analysis and underlying intuition about viewing behavior an initial proposed model specification and coefficient expectations are:

`log(length)`: It is not intuitively obvious how the `length` of a video may relate to the number of `views`, in particular it seems unlikely that there would be a simple linear relationship between `views` and `length` in that you would not expect the longer the videos the more (or less) `views` or conversely for shorter videos. There could however be an argument that there may be an optimal length in that very short videos may have little content of interest, while very long videos may be too time consuming to watch, in which case a parabolic model may be suited to `length`. Accordingly, the expectation would be that the coefficient on the squared term would be negative, such that their is optimal `length` to maximize `views`. If only fitting `length` as a linear function then I would expect the coefficient to be negative in that short videos are probably more consumable and catchy to attract high numbers of `views` compared to long videos.

`rate`: Intuitively, you would expect that as the average rating rises (`rate`) the number of `views` would also likely rise, so a simple linear relationship between `views` and `rate` is expected with a positive coefficient.

The model:

$$log(views) = \beta_0 + \beta_1 log(length) + \beta_2 log(length)^2 + \beta_3 rate + u$$

Running the model to test the validity of the 6 assumptions of the CLM:

```
m1 = lm(log(views) ~ poly(log(length), 2) + rate, data = D)
```

Testing the validity of the 6 assumptions of the CLM

CLM 1 - A linear model

The model is specified such that the dependent variable is a linear function of the explanatory variables.

Is the assumption valid? **Yes**

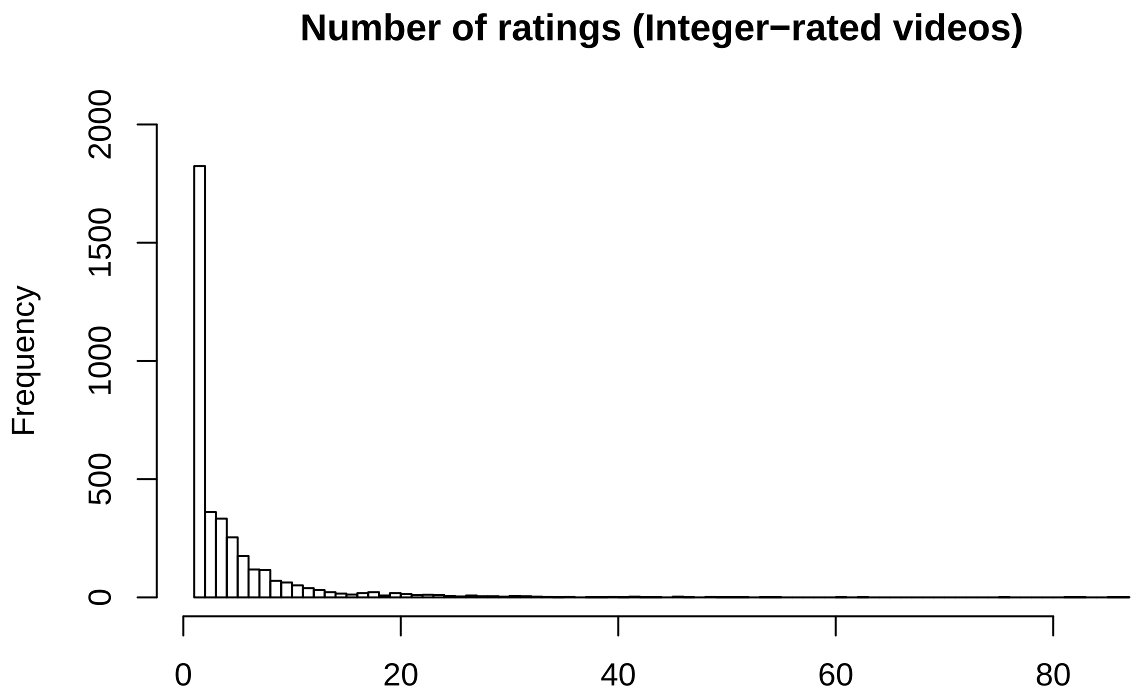
Response: No response required.

CLM 2 - Random Sampling

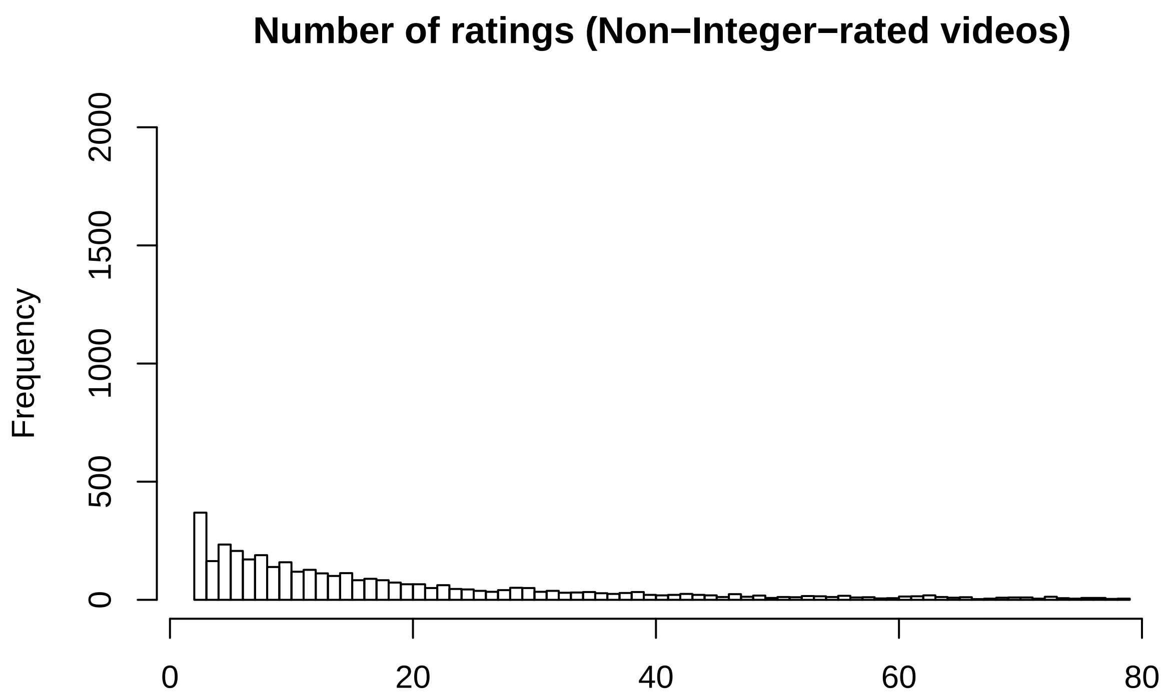
The distribution of `length` has been shown to be truncated. This is caused by YouTube's upload conditions that don't allow you to upload videos longer than 15 minutes if you don't verify your account. I suspect many contributors to YouTube may only do one video to try it out and don't verify their account. This is fairly consistent with the observed truncation in the distribution above video length of ~12 minutes. There is also a 20GB upload limit as well, so dependent on the content and quality this will also cause a truncation in the distribution. The shape of the log distribution is more normal indicating that if there were no upload constraints then contributors may contribute longer videos to fill out the likely true distribution, i.e. there are probably some missing observations as a result of people wanting to upload a longer video, but not being able to. The constraint likely causes `length` to not meet the random sampling condition, however, a constraint which effectively causes clustering is not the worst violation of this assumption because the variable is likely random within the cluster.

As discussed above the distribution of `rate` is certainly unusual. There is a very large number of ratings that only have a low single digit number of raters (i.e. number of `ratings`), so you have very low confidence that it is a fair rating for the video. This is what causes the high frequency of average ratings that are integers, i.e. they only represent the views of a small number of people, where as videos with an average rating to two-decimal places tend to have more raters.

```
hist(Data$ratings[Data$rate %in% c(1,2,3,4,5) & Data$ratings < 90], main="Number of ratings (Integer-ra
```




```
hist(Data$ratings[!(Data$rate %in% c(0,1,2,3,4,5)) & Data$ratings < 80], main="Number of ratings (Non-I
```




```
(tot5 = length(Data$ratings[Data$rate==5]))
```

```
## [1] 2893
```

```
(less3_tot5 = length(Data$ratings[Data$rate==5 & Data$ratings<=3]))
```

```
## [1] 1653
```

The histograms of integer-rated and non-integer-rated videos show that integer-rated videos typically have a small number of raters (thinner right tail) and the many only have one rating. For example, 57.14% of 5-rated videos have 3 or less raters. It is quite possible that 5-rated videos are rated by the person who uploaded the video and possibly a mate or two to try to get it to go **viral**.

Accordingly, `rate` does not appear to be random because many videos with a low number of `ratings` are potentially self-`rated` videos and hence possibly may not be a reliable estimate of the true average rating for the video. Even if this behavior preposition is unfounded there are still many videos with very low numbers of `ratings`, which would introduce increased error into the model because you still cannot be confident in the reliability of the `rate` variable (albeit this is less indicative of a random sampling breach), overall `rate` likely breaches CLM 2.

Is the assumption valid? **No**

Response: You possibly don't need to respond to the `length` concern because it relates to clustering, while the `rate` concern appears more problematic. To ensure that the ratings are fair and hence reflect random sampling you could require the rating to be an average of at least a certain number of people, say 15 (this is just randomly chosen to get at least a reasonable distribution).

CLM 3 - Multicollinearity

As a quick test of the multicollinearity condition I check the correlation of the two explanatory variables and their Variance Inflation Factors (VIF):

```
D$loglength = log(D$length)
D$loglength2 = D$loglength^2
```

```r
X = data.matrix(subset(D, select=c("length", "loglength", "loglength2", "rate")))
(Cor = cor(X))
```

```
##               length loglength loglength2      rate
## length     1.0000000 0.6979152  0.7790734 0.1111822
## loglength  0.6979152 1.0000000  0.9864857 0.1944981
## loglength2 0.7790734 0.9864857  1.0000000 0.1913050
## rate       0.1111822 0.1944981  0.1913050 1.0000000
```

```r
vif(m1)
```

```
##                         GVIF Df GVIF^(1/(2*Df))
## poly(log(length), 2) 1.03933  2        1.009691
## rate                 1.03933  1        1.019475
```

The three explanatory variables (`loglength`, `loglength2` and `rate`) are not perfectly correlated and the VIFs are low (i.e. less than 10), so there is no perfect multicollinearity of the independent variables.

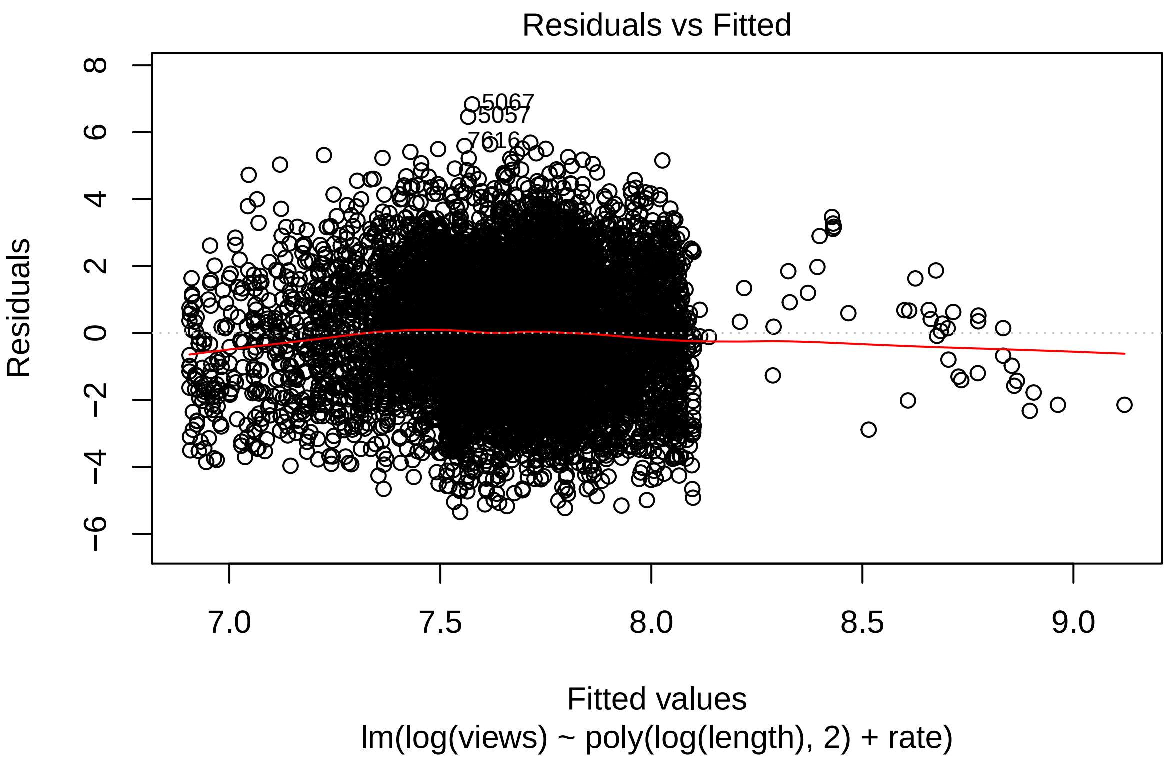
Is the assumption valid? **Yes**

Response: No response required.

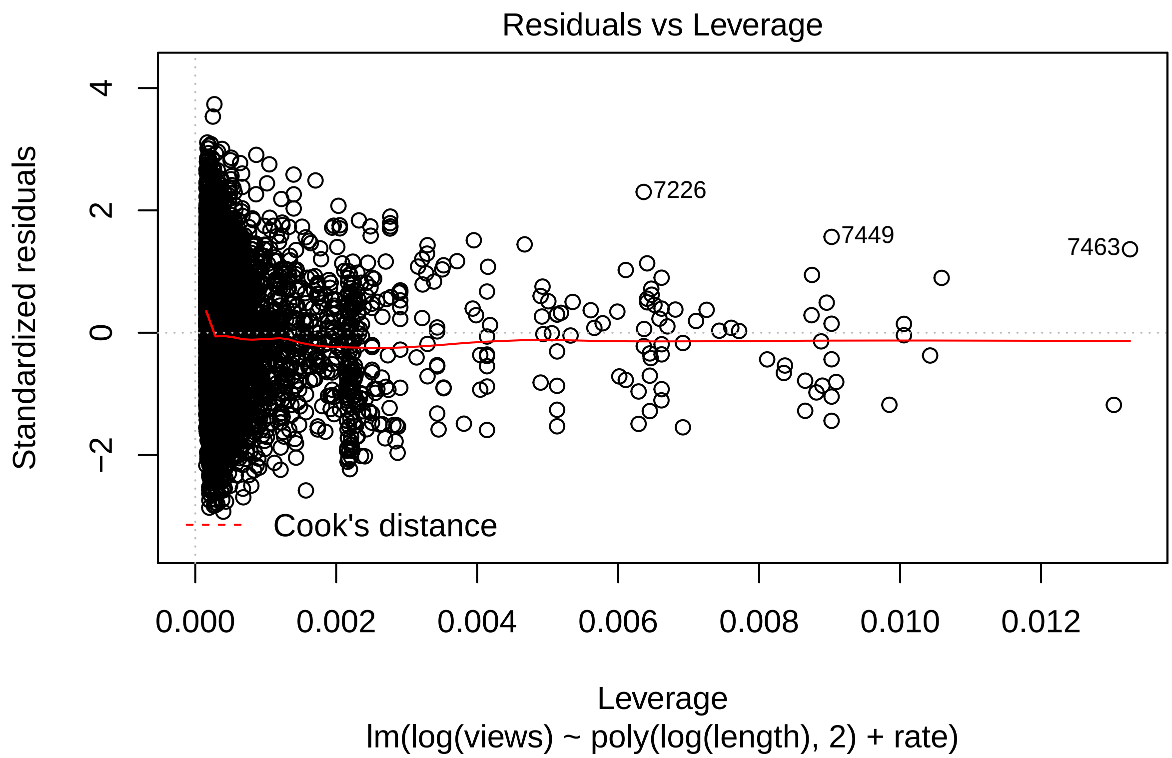
CLM 4 – Zero-Conditional Mean

To analysis whether there is a zero-conditional mean across all x's you can plot the residuals against the fitted values with the predicted conditional mean spline line across fitted values and you can also test for the less strong condition of exogeneity.

```r
plot(m1, which=1)
```




```r
plot(m1, which=5)
```




```
(cov(log(D$length),m1$residuals))
```

```
## [1] -3.612415e-18
```

```
(cov(log(D$length)^2,m1$residuals))
```

```
## [1] -2.026366e-16
```

```
(cov(D$rate,m1$residuals))
```

```
## [1] 7.378137e-15
```

The plots indicate little evidence that the zero-conditional mean assumption doesn't hold, for example, the red spline line on the residuals vs fitted values plot is fairly flat before a downturn at higher fitted values due to there being less observations.

The covariances of the three independent variables with the residuals are very close to zero indicating they are likely exogenous.

Notably no data point has a large Cook's distance, so there are no observations with undue influence on the model fit.

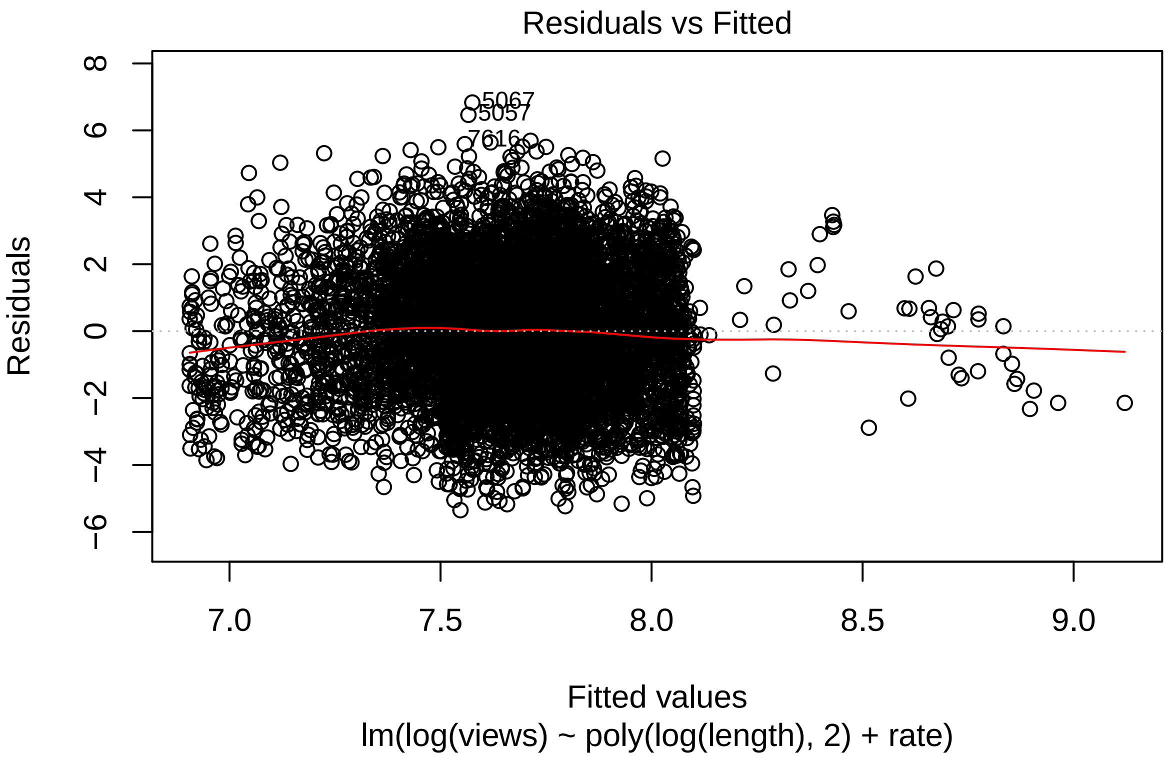
Is the assumption valid? **Yes**

Response: There is little evidence that this assumption is not valid, however, even if there was given a large sample size we are confident that due to OLS asymptotics that the coefficients are at least consistent, so no response is required.

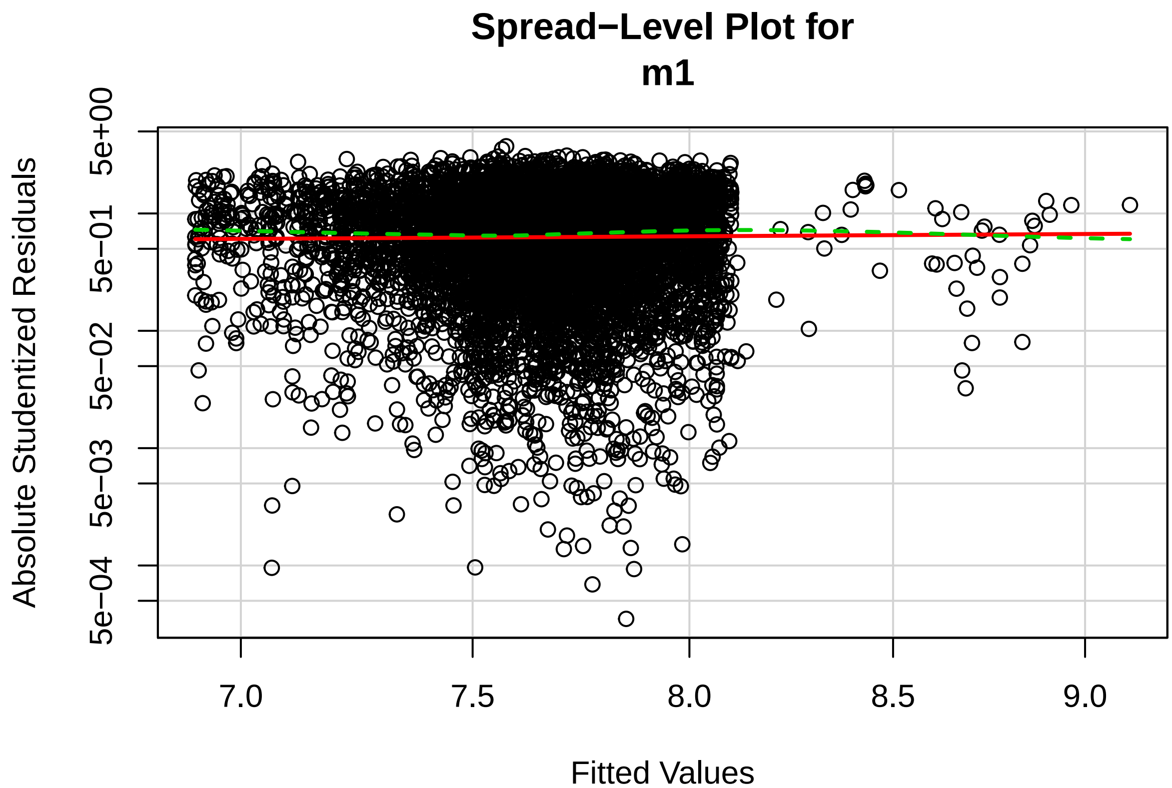
CLM 5 - Homoscedasticity

To determine whether the variance of u is fixed for all x's you can first simply view the residuals plotted against the fitted values to see whether the variance of residuals is constant across the fitted values or perform statistical tests such as Breusch-Pagan or the Score-test for non-constant error variance.

```r
# residuals vs fitted values plot
plot(m1, which=1)
```




```r
# Spread-level plot
spreadLevelPlot(m1)
```




```
##
## Suggested power transformation:  0.6165286
```

```r
# Breusch-Pagan-Test
bptest(m1)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  m1
## BP = 16.37, df = 3, p-value = 0.000952
```

```r
# Score-test for non-constant error variance
ncvTest(m1)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 1.981653    Df = 1     p = 0.1592163
```

As you can see the range of the residuals possibly widens as the fitted value rises, which is supported by a high significant p-values for the Breusch-Pagan test (although you must be careful given the large sample size). However, the Score-test is not significant, so the tests are producing mixed evidence of a heteroscedasticity problem.

Non-constant error variance does not cause biased estimates, but it does pose problems for efficiency and the usual formulas for standard errors are inaccurate. OLS estimates are inefficient because they give equal weight to all observations regardless of the fact that those with large residuals contain less information about the regression.

Is the assumption valid? **Most likely, but not 100% sure**

Response: Heteroscedasticity can be addressed by calculating robust standard errors and it is normally recommended to do so anyway, so given that the tests are inconclusive calculation of robust standard errors is recommended. Robust standard errors do not change the OLS coefficient estimates or solve the inefficiency problem, but do give more accurate p-values.

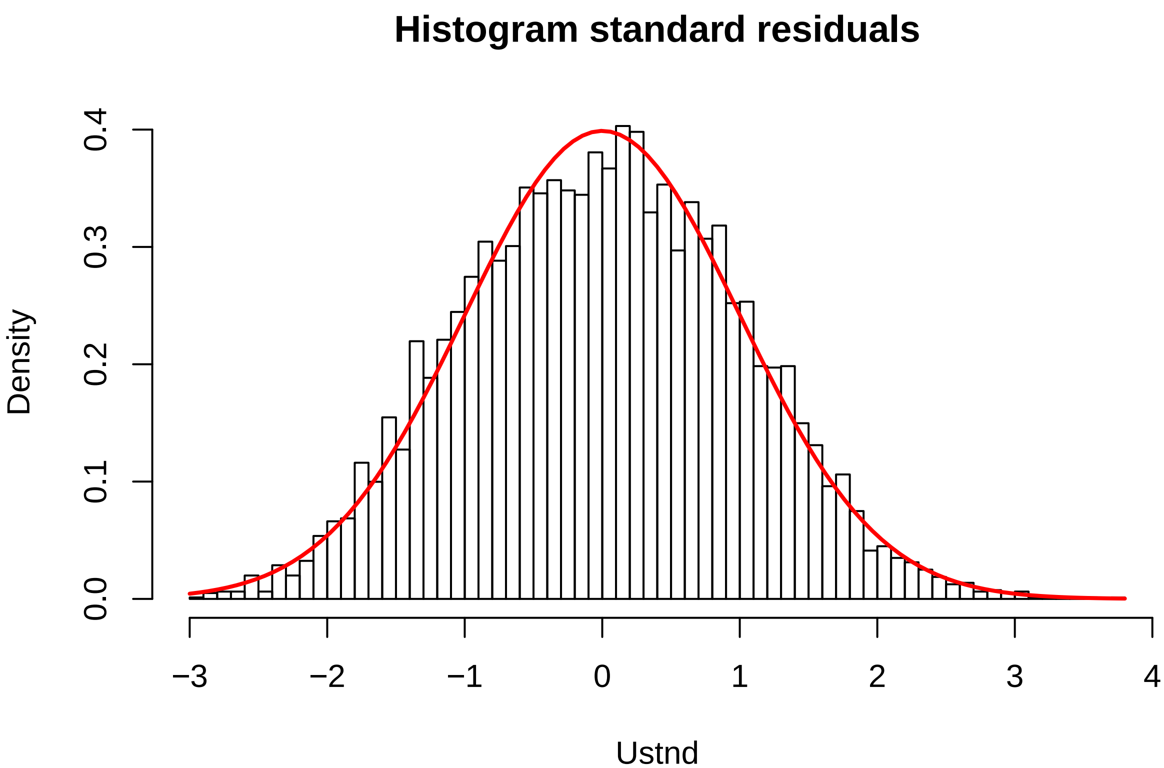
Otherwise an alternate method to resolve this problem is a Weighted Least Squares model noting that the rise in variance with fitted value is somewhat proportional to the `rate` variable.

Or the Spread-Level plot indicates that a power transformation of the dependent variable, in particular taking `log(view)` to the power of ~0.6 may eliminate the heteroscedasticity.

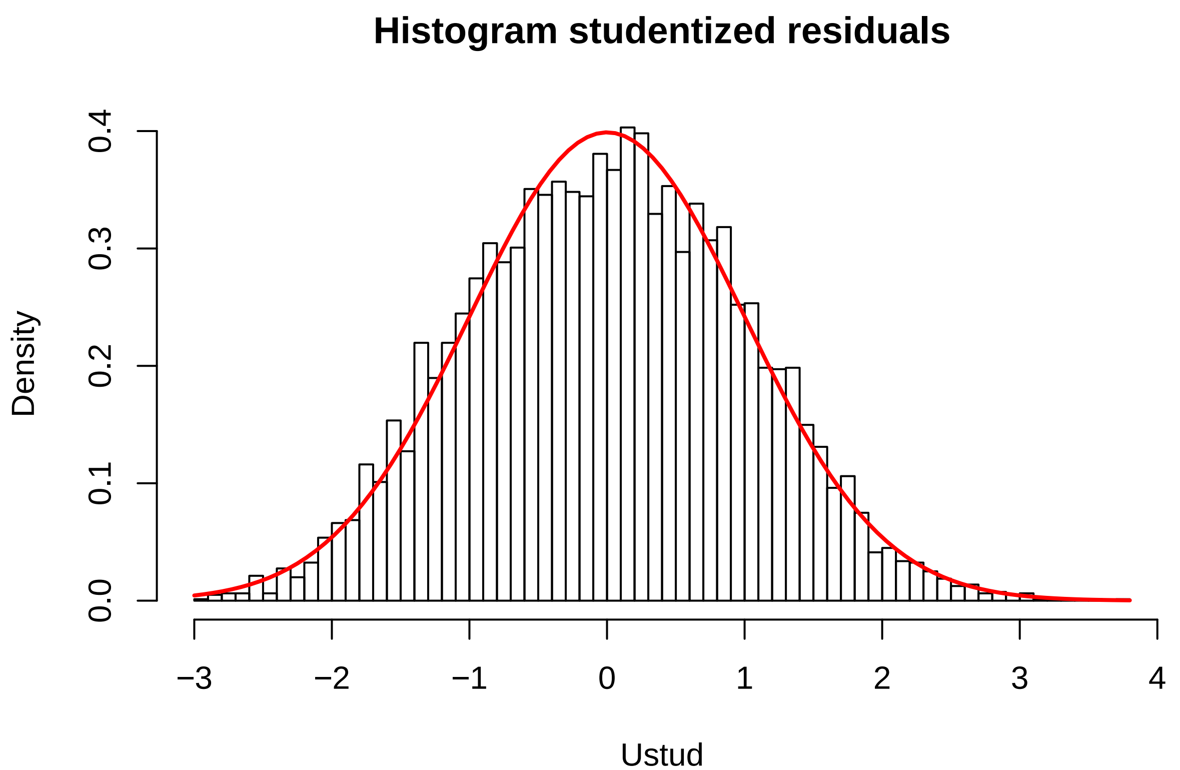
CLM 6 – Normality of residuals

To determine whether there is normality of the residuals you can use histogram or Q-Q plots of the residuals and simply visually observe whether there is normality.

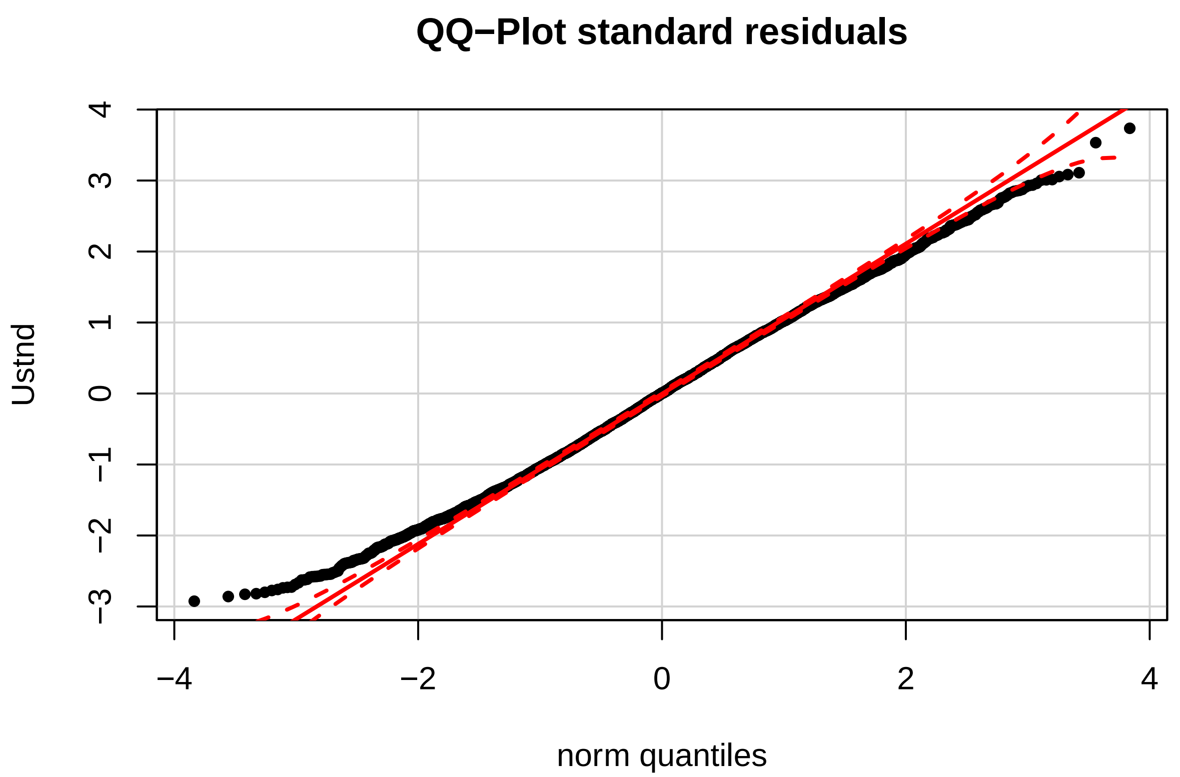
```
# normality of standard residuals
Ustnd = rstandard(m1)
hist(Ustnd, main="Histogram standard residuals", breaks = 50, freq=FALSE)
curve(dnorm(x, mean=0, sd=sd(Ustnd)), col="red", lwd=2, add=TRUE)
```




```
# normality of studentized residuals
Ustud = rstudent(m1)
hist(Ustud, main="Histogram studentized residuals", breaks = 50, freq=FALSE)
curve(dnorm(x, mean=0, sd=1), col="red", lwd=2, add=TRUE)
```

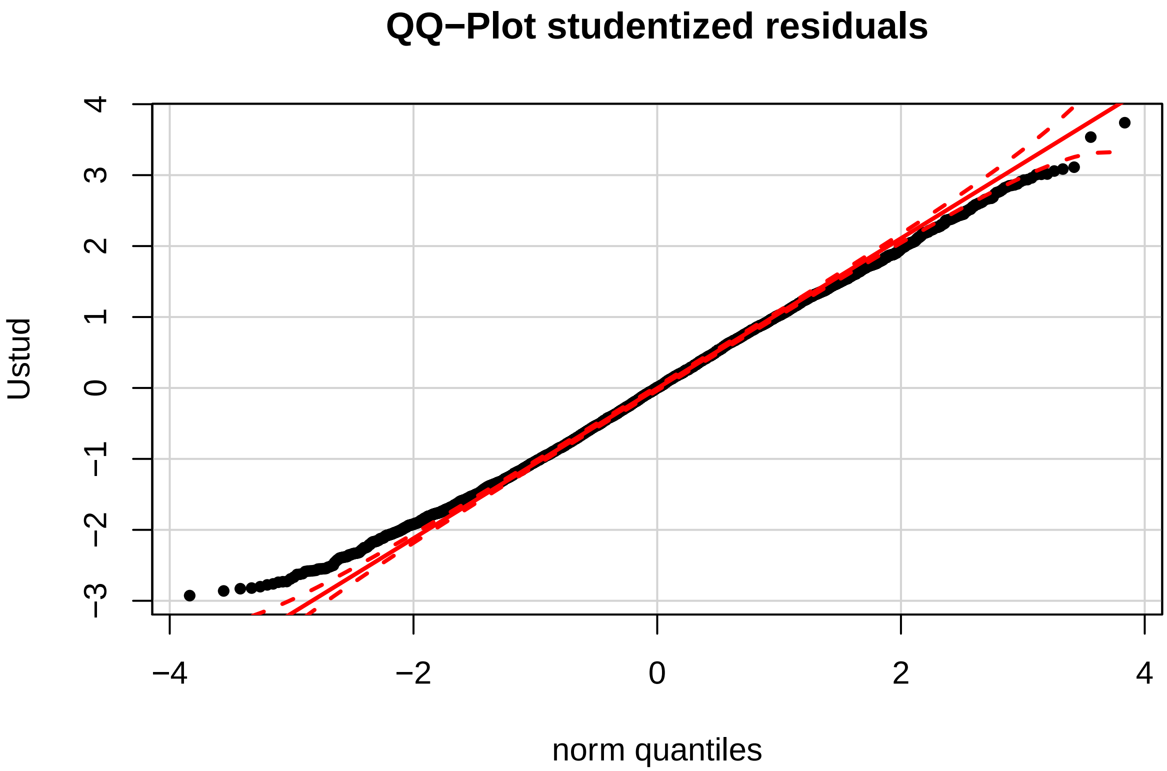



```r
# Q-Q plot standard residuals
qqPlot(Ustnd, distribution="norm", pch=20, main="QQ-Plot standard residuals")
qqline(Ustnd, col="red", lwd=2)
```




```r
# Q-Q plot studentized residuals
qqPlot(Ustud, distribution="norm", pch=20, main="QQ-Plot studentized residuals")
```

```r
qqline(Ustud, col="red", lwd=2)
```



The histograms in particular appear to be fairly normally distributed (albeit a little light in the center of the distribution, which means that the tails are a bit fatter than normal), while the Q-Q plot (which also reflects the slightly fatter tails than normal) doesn't deviate significantly from normality either, so overall the residuals appear to be fairly normal. Notably the log transformation of `views` has helped to produce normal errors.

Is the assumption valid? **Yes**

Response: No response required.

Adjusted model

To respond to the invalid assumptions I do the following:

1. Random Sampling: Reduce the sample to only include observations where `ratings` is greater than 15
2. Re-run the same model, but for:

   Model 2 - calculate robust standard errors,

   Model 3 - run a Weighted Least Squares, and

   Model 4 -transform the dependent variable with 0.6 power transformation.

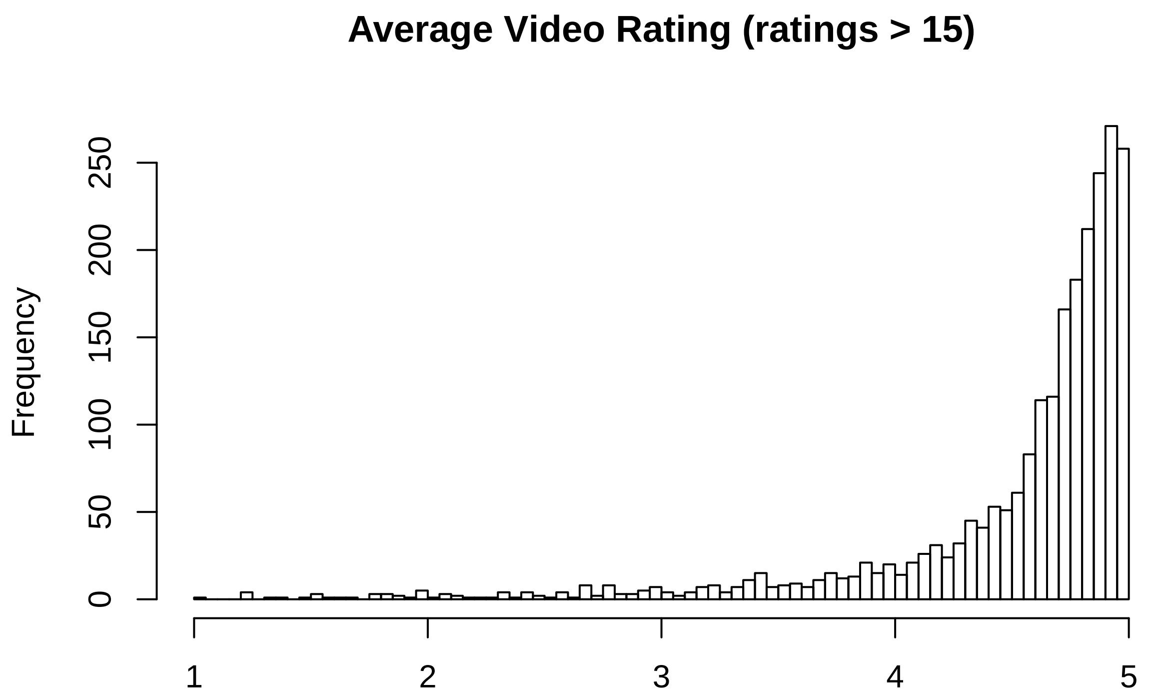
Following re-running the models with these adjustments I also quickly re-run some basic diagnostics.

```r
DA = Data[Data$ratings>15,]
DA = DA[,(5:7)]
# filter the data table to only include rows with 3 non-null observations
filter = !is.na(DA$length) | !is.na(DA$views) | !is.na(DA$rate)
```

```r
DA = DA[filter,]
summary(DA)
```

```
##      length           views               rate      
##  Min.   :   2.0   Min.   :     69   Min.   :1.000  
##  1st Qu.: 142.8   1st Qu.:   4734   1st Qu.:4.470  
##  Median : 233.0   Median :  12316   Median :4.755  
##  Mean   : 268.1   Mean   :  30570   Mean   :4.545  
##  3rd Qu.: 344.0   3rd Qu.:  30196   3rd Qu.:4.890  
##  Max.   :3229.0   Max.   :1807640   Max.   :5.000
```

```r
hist(DA$rate, main="Average Video Rating (ratings > 15)", xlab=NULL, breaks=100)
```




```r
m2 = lm(log(views) ~ poly(log(length), 2) + rate, data = DA)
coeftest(m2, vcov = vcovHC)
```

```
## 
## t test of coefficients:
## 
##                         Estimate Std. Error t value  Pr(>|t|)    
## (Intercept)             8.498195   0.269867 31.4903 < 2.2e-16 ***
## poly(log(length), 2)1 -0.943323   1.461033 -0.6457  0.518567    
## poly(log(length), 2)2  3.111471   1.386102  2.2448  0.024877 *  
## rate                    0.192053   0.058667  3.2736  0.001077 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
vcovHC(m2)
```

```
##                       (Intercept) poly(log(length), 2)1
## (Intercept)            0.07282825            0.09095665
## poly(log(length), 2)1  0.09095665            2.13461749
## poly(log(length), 2)2 -0.02495609            0.01131987
## rate                  -0.01573977           -0.02018562
```

```
##                        poly(log(length), 2)2         rate
## (Intercept)                     -0.024956093 -0.015739767
## poly(log(length), 2)1            0.011319865 -0.020185621
## poly(log(length), 2)2            1.921279426  0.005339845
## rate                             0.005339845  0.003441872
```

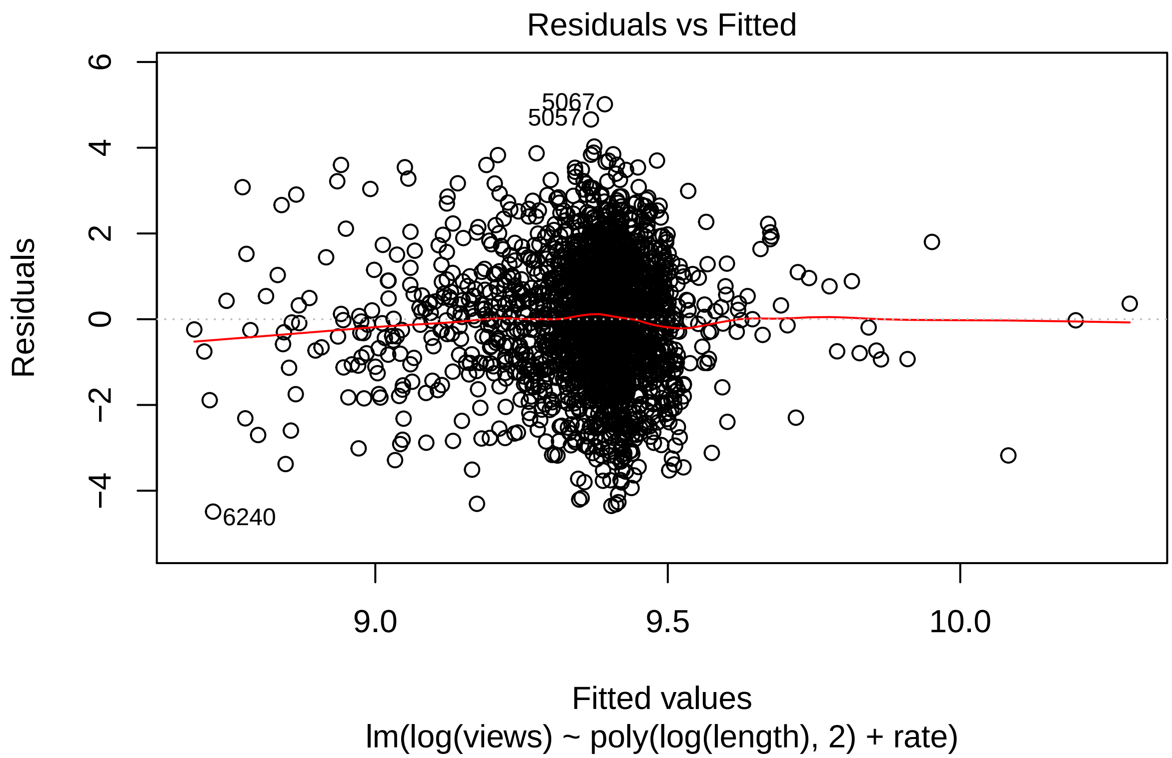
```r
(se.m2 = sqrt(diag(vcovHC(m2))))
```

```
##           (Intercept) poly(log(length), 2)1 poly(log(length), 2)2
##            0.26986711            1.46103302            1.38610224
##                  rate
##            0.05866747
```

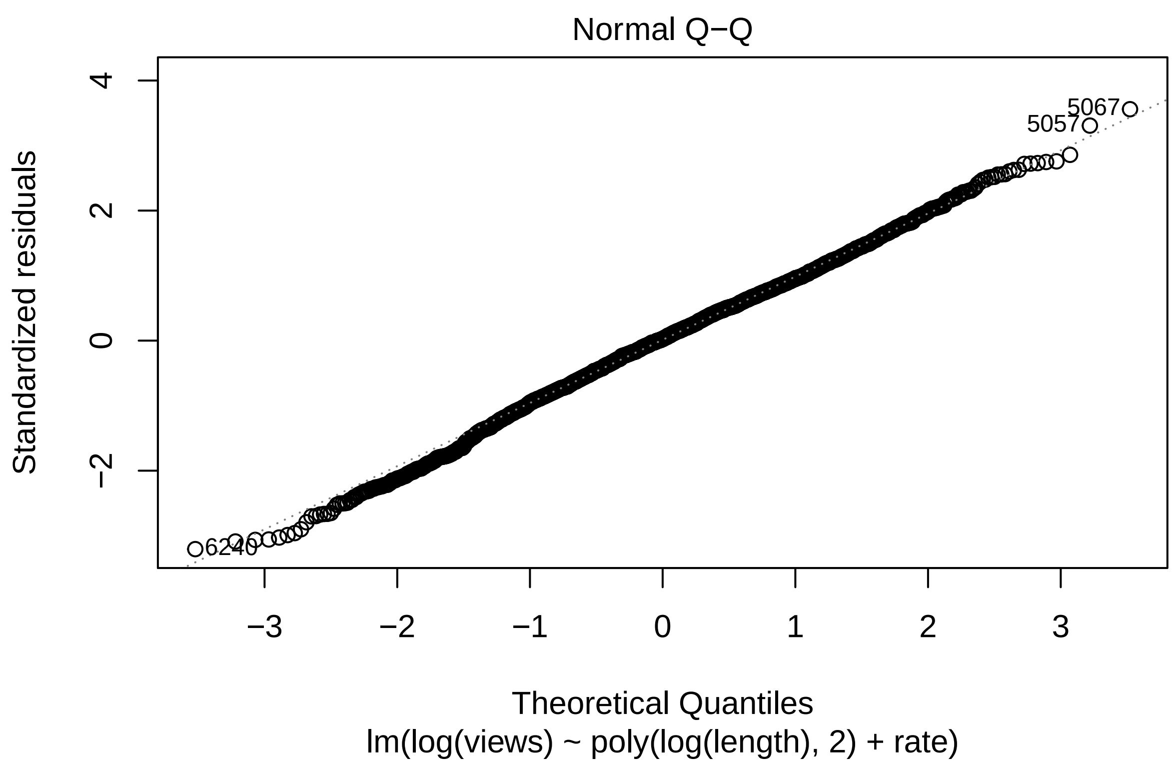
```r
summary(m2)
```

```
##
## Call:
## lm(formula = log(views) ~ poly(log(length), 2) + rate, data = DA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4886 -0.9052  0.0345  0.9410  5.0152
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            8.49819    0.23309  36.458  < 2e-16 ***
## poly(log(length), 2)1 -0.94332    1.46597  -0.643 0.519977
## poly(log(length), 2)2  3.11147    1.41217   2.203 0.027669 *
## rate                   0.19205    0.05088   3.775 0.000164 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.41 on 2332 degrees of freedom
## Multiple R-squared:  0.007819,   Adjusted R-squared:  0.006542
## F-statistic: 6.126 on 3 and 2332 DF,  p-value: 0.0003798
```

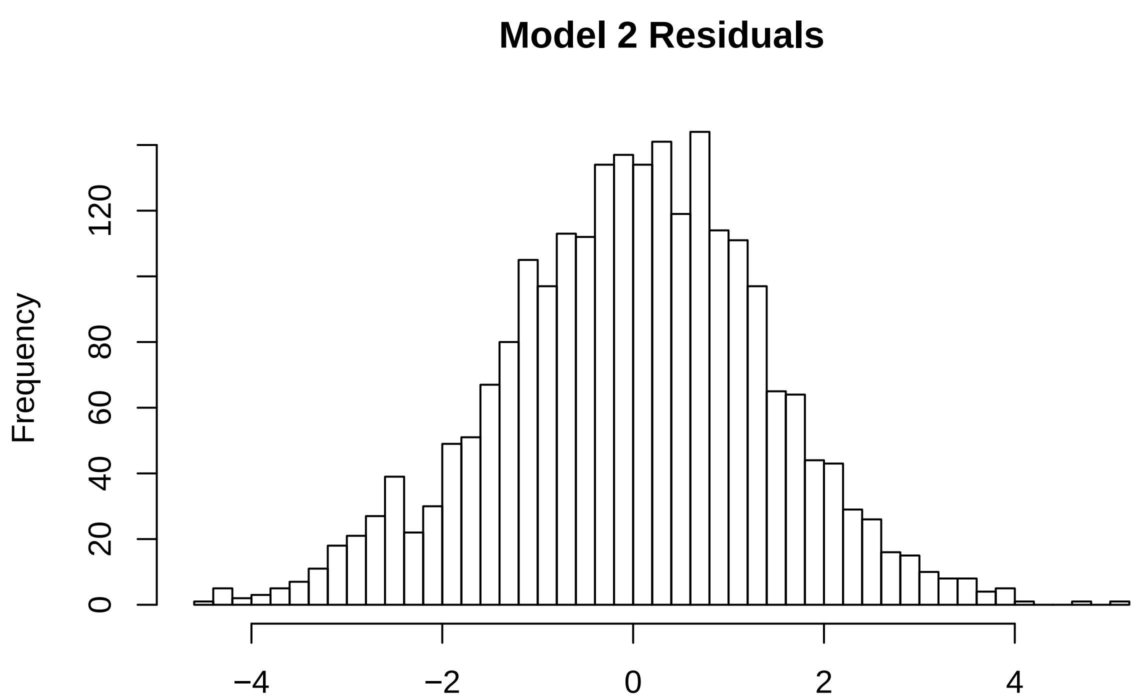
```r
plot(m2, which=1)
```




```
plot(m2, which=2)
```



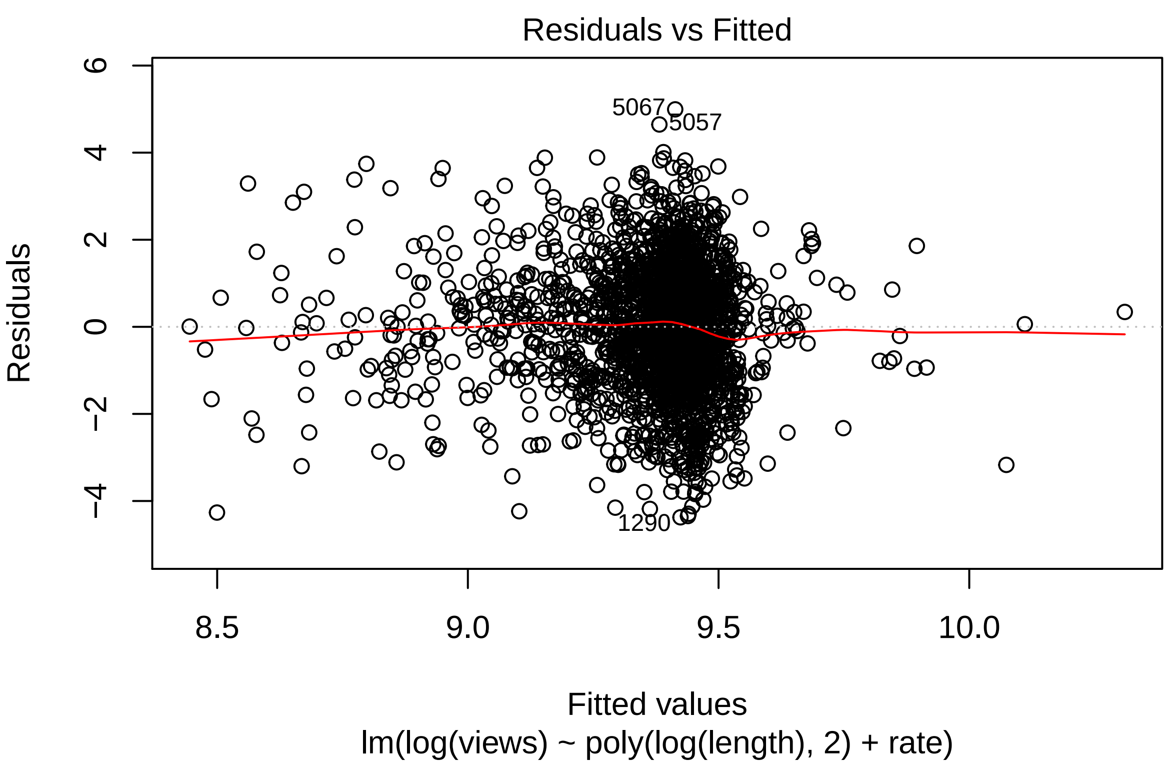

```
hist(m2$residuals, main="Model 2 Residuals", xlab=NULL, breaks = 50)
```



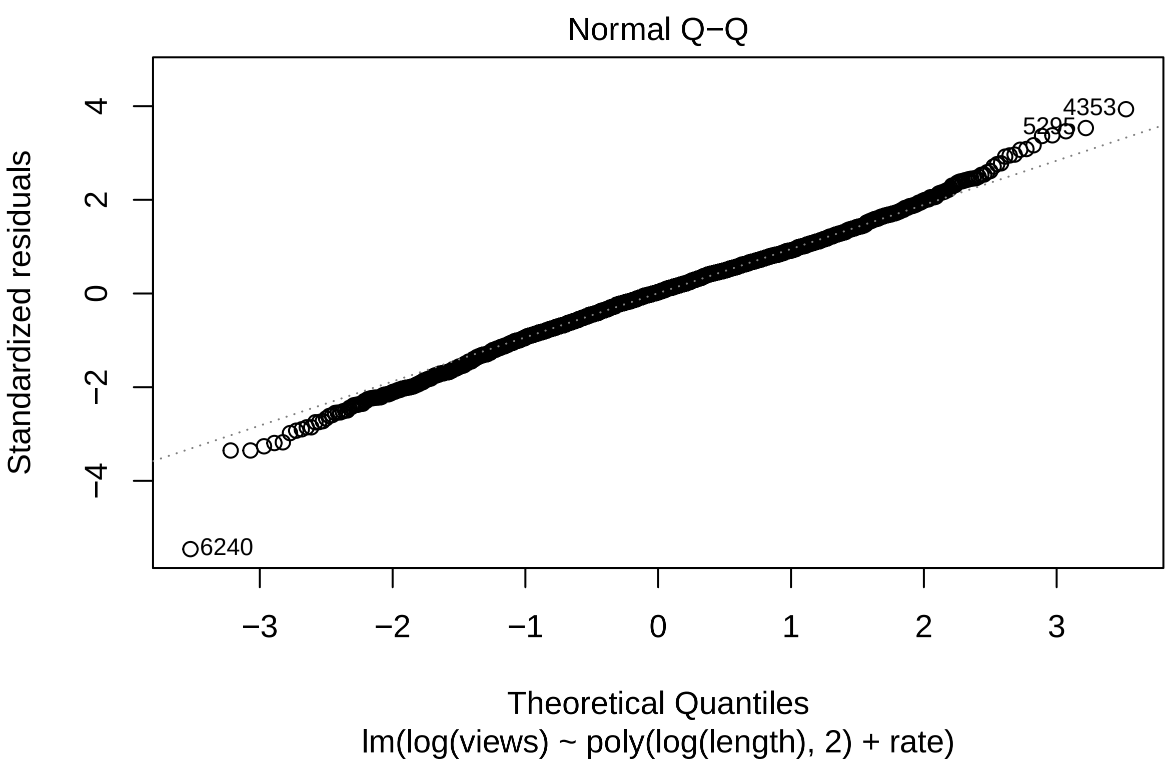

```r
m3 = lm(log(views) ~ poly(log(length), 2) + rate, data = DA, weights=1/rate)
summary(m3)
```

```
##
## Call:
## lm(formula = log(views) ~ poly(log(length), 2) + rate, data = DA,
##     weights = 1/rate)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6576 -0.4264  0.0209  0.4369  2.6438
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            8.17793    0.17801  45.940  < 2e-16 ***
## poly(log(length), 2)1 -1.07160    1.44616  -0.741   0.4588
## poly(log(length), 2)2  3.00295    1.37901   2.178   0.0295 *
## rate                   0.26252    0.03972   6.610 4.75e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.679 on 2332 degrees of freedom
## Multiple R-squared:  0.02048,    Adjusted R-squared:  0.01922
## F-statistic: 16.25 on 3 and 2332 DF,  p-value: 1.87e-10
```

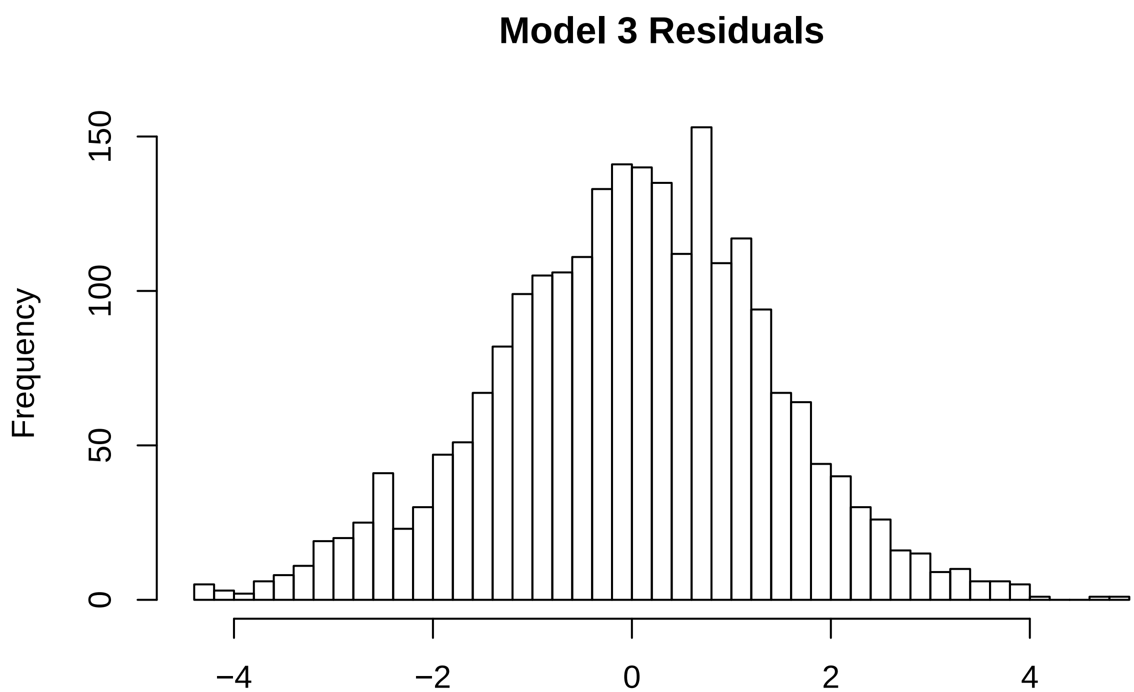
```r
plot(m3, which=1)
```




```
plot(m3, which=2)
```



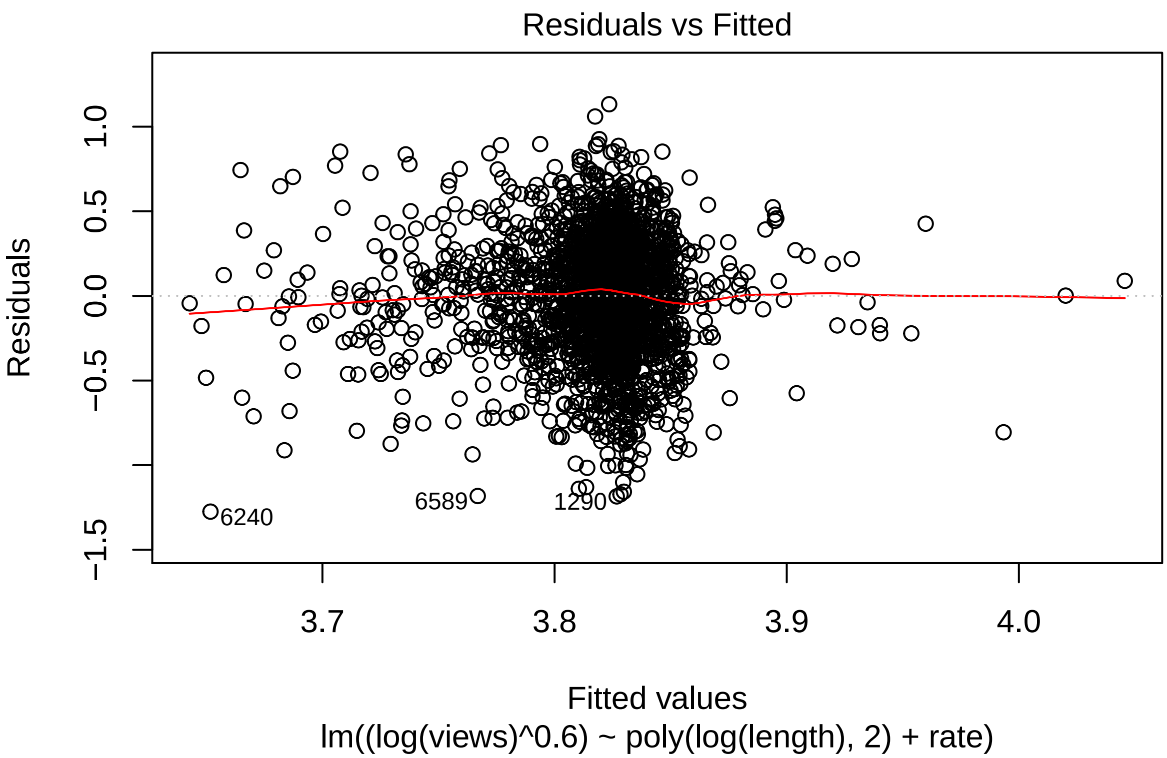

```
hist(m3$residuals, main="Model 3 Residuals", xlab=NULL, breaks = 50)
```



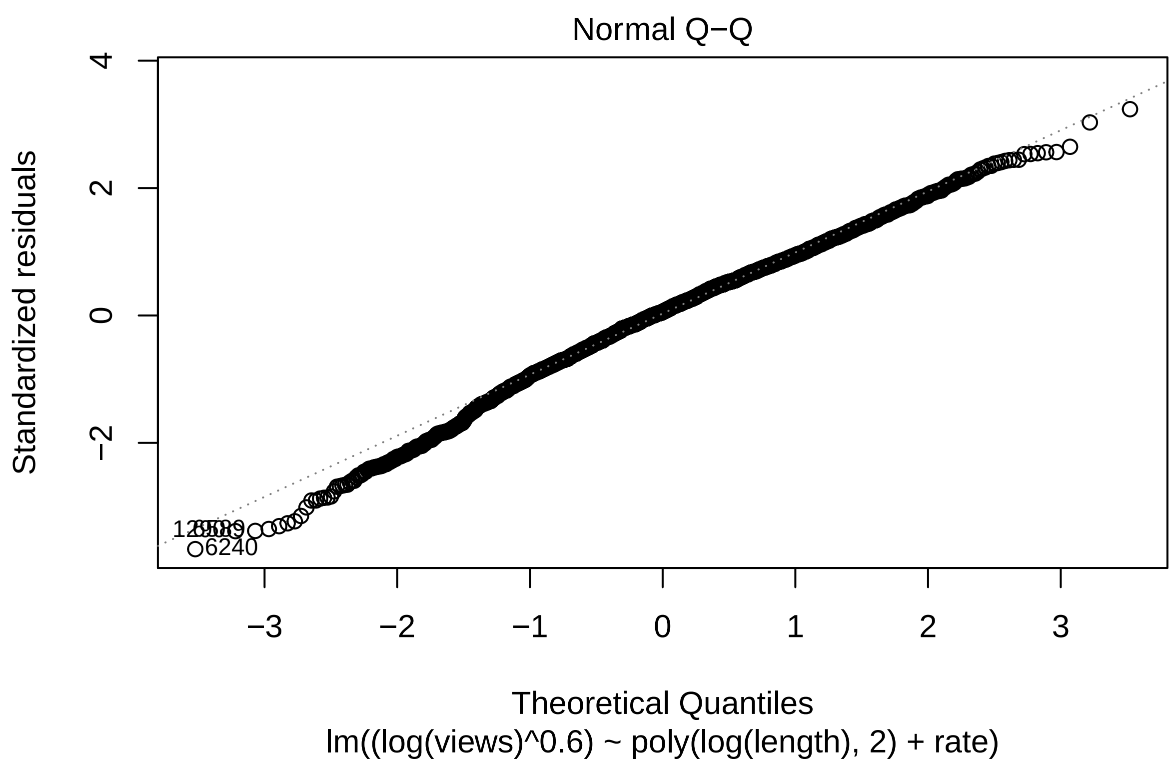

```r
m4 = lm((log(views)^0.6) ~ poly(log(length), 2) + rate, data = DA)
summary(m4)
```

```
##
## Call:
## lm(formula = (log(views)^0.6) ~ poly(log(length), 2) + rate,
##     data = DA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.27463 -0.21571  0.01848  0.23689  1.13274
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              3.59379    0.05786  62.111  < 2e-16 ***
## poly(log(length), 2)1 -0.22070    0.36390  -0.606   0.5443
## poly(log(length), 2)2  0.77295    0.35054   2.205   0.0276 *
## rate                   0.04935    0.01263   3.907  9.6e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.35 on 2332 degrees of freedom
## Multiple R-squared:  0.008268,   Adjusted R-squared:  0.006992
## F-statistic: 6.481 on 3 and 2332 DF,  p-value: 0.0002298
```

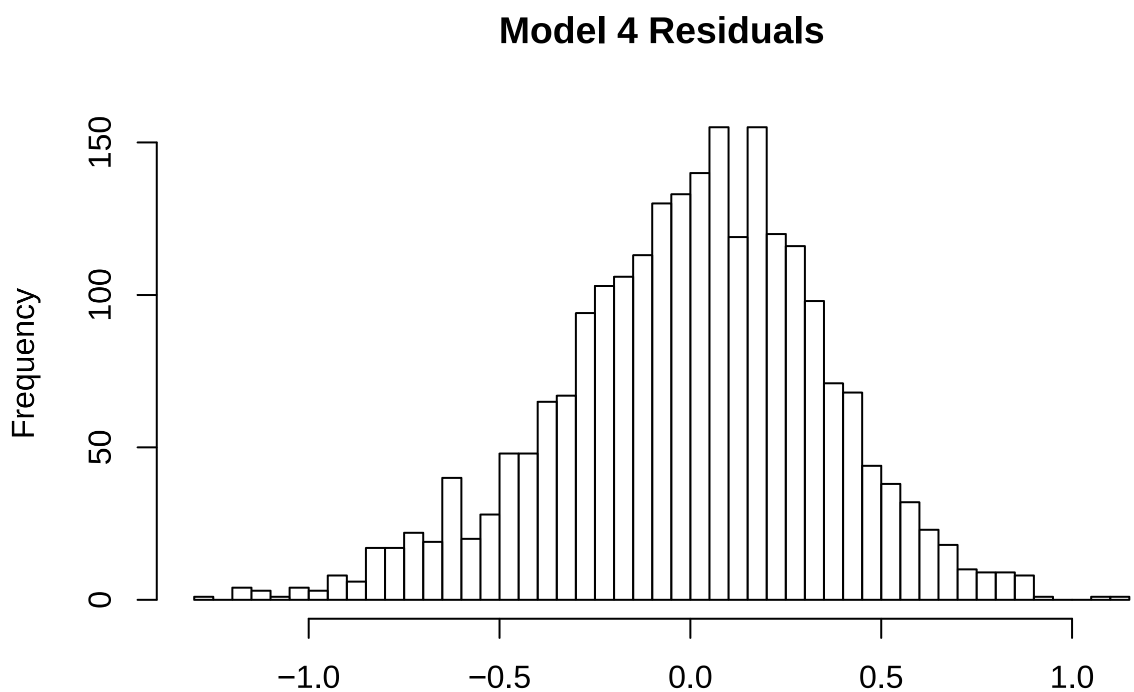
```r
plot(m4, which=1)
```




```r
plot(m4, which=2)
```




```r
hist(m4$residuals, main="Model 4 Residuals", xlab=NULL, breaks = 50)
```




```
se.m1 = coef(summary(m1))[, "Std. Error"]
se.m3 = coef(summary(m3))[, "Std. Error"]
se.m4 = coef(summary(m4))[, "Std. Error"]
```

On balance the diagnostics for **Model 2** are indicative that all 6 CLM assumptions are now met

While the other two models, which are just out of interest, suggest the Weighted Least Squares is also an OK approach, while the power transformation at the least leads to non-normality of the residuals, so appears less successful (it also it ends up with non-intuitive coefficients, so is not a great approach).

The Model Results

The results of all four models are reported in the table below:

```
stargazer(m1, m2, m3, m4, type = "latex",
title = "Linear Models Predicting Views",
se = list(se.m1, se.m2, se.m3, se.m4), omit.stat=c("f","ser"),
          star.cutoffs = c(0.05, 0.01, 0.001))
```

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Tue, Aug 15, 2017 - 00:39:38

Observations:

1. All independent variables come up as statistically significant in the original model (**Model 1**), which is likely due to the size of the sample and invalid standard errors more so than any valid relationship having been found.
2. Also the coefficients in **Model 1** cannot be relied upon either because of random sampling problems in my view.

Table 1: Linear Models Predicting Views

	Dependent variable:			
	log(views)			(log(views)^0.6)
	(1)	(2)	(3)	(4)
poly(log(length), 2)1	13.354***	−0.943	−1.072	−0.221
	(1.865)	(1.461)	(1.446)	(0.364)
poly(log(length), 2)2	7.917***	3.111*	3.003*	0.773*
	(1.829)	(1.386)	(1.379)	(0.351)
rate	0.152***	0.192**	0.263***	0.049***
	(0.024)	(0.059)	(0.040)	(0.013)
Constant	6.994***	8.498***	8.178***	3.594***
	(0.109)	(0.270)	(0.178)	(0.058)
Observations	8,013	2,336	2,336	2,336
R^2	0.016	0.008	0.020	0.008
Adjusted R^2	0.016	0.007	0.019	0.007
Note:				*p<0.05; **p<0.01; ***p<0.001

3. Correcting for random sampling and robust errors produces a reliable **Model 2** with the same linear model specification. Notably, average video rating (`rate`) remains statistically significant at the 1% level, while the quadratic modeling of `log(length)` has now become significant only at a 5% level. These findings are consistent with my expected intuition that a linear relationship between average rating (`rate`) and `views` seems highly plausible, while a relationship between `views` and `length` seems less obvious.
4. The sign on the squared `log(length)` term is also positive, which is opposite to my initial flimsy intuition.
5. **Model 3** has quite similar parameters to **Model 2**, but has better explanatory power, so is possibly a better specification.
6. As highlighted by `r-squared` all models explain only 2% or less of the variation in `log(views)`, so the models are not overly useful for predicting the number of `views`.
7. For **Model 3** a 1 rating point (`rate`) increase for a video predicts a 26.3% increase in video `views`, so this is also a practically significant result.