OLS Inference

Checking Assumptions and Using Results

Daniel Vanlunen

OLS Inference

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

```
v<-fread("videos.txt")
```

1 Make a Linear Model Predicting Views from Length and Ratings

$$views = \beta_0 + \beta_1 length + \beta_2 rate + u$$

```
m1 <- lm(views~length + rate, data=v)
```

2 Check CLM Assumptions

Before conducting any inference, the CLM assumptions must be checked.

MLR 1 Linear Population Model

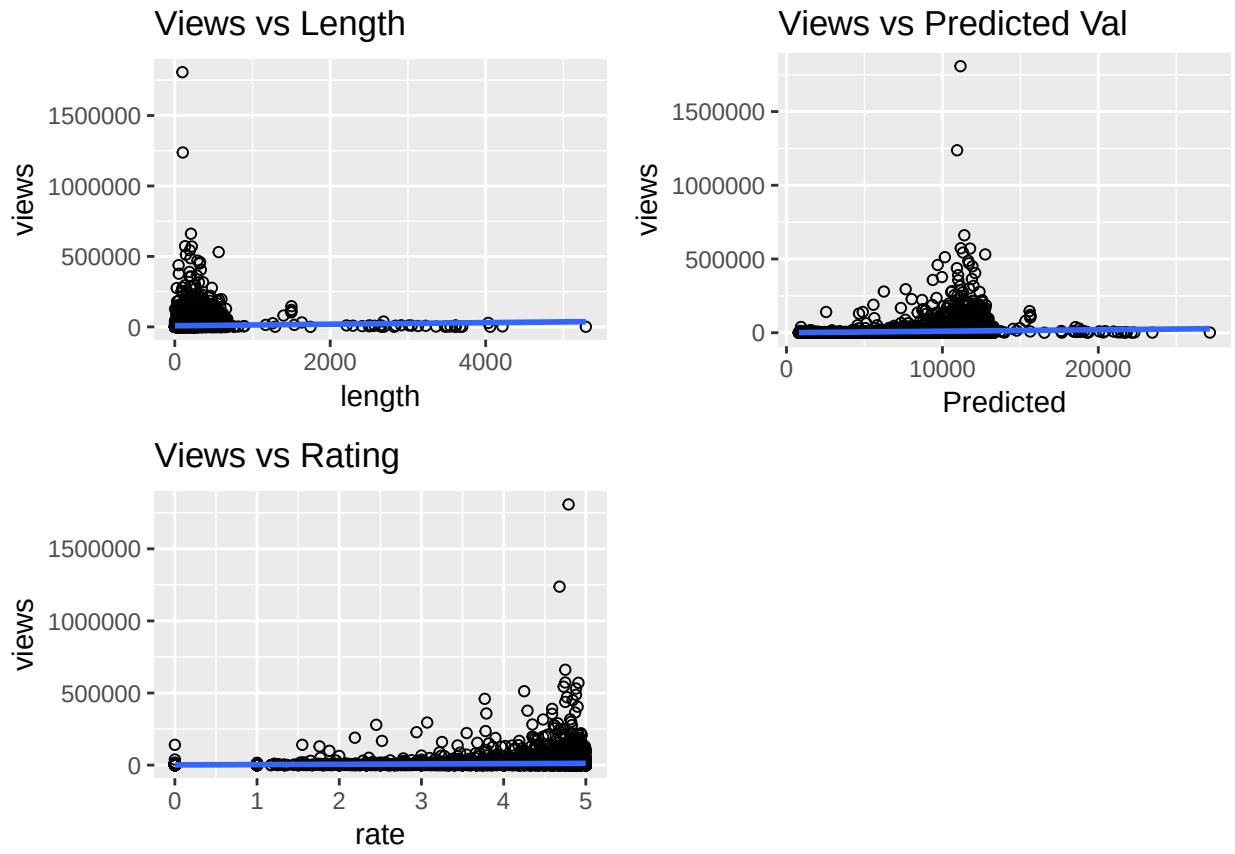
Given we have not placed an assumption on the error term, this assumption is met. However, we can examine some scatter plots to check for a linear relationship to see if linear regression makes sense (note that it could be a linear combination of non-linear functions of the features).

```
v$predict1[!is.na(v$views) &
             !is.na(v$length) &
             !is.na(v$rate)] <-fitted(m1)
p1<-ggplot(v, aes(x=length, y=views)) +
    geom_point(shape=1) +
    geom_smooth(method=lm) +
    ggtitle("Views vs Length")
p2<-ggplot(v, aes(x=rate, y=views)) +
    geom_point(shape=1) +
    geom_smooth(method=lm) +
    ggtitle("Views vs Rating")
p3<-ggplot(v, aes(x=predict1, y=views)) +
    geom_point(shape=1) +
    geom_smooth(method=lm) +
    ggtitle("Views vs Predicted Val") +
    labs(x="Predicted")

multiplot(p1,p2,p3,cols=2)
```

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

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



The relationship does not appear particularly linear so we may have to adjust function form based on assumptions later.

MLR 2 Random Sampling

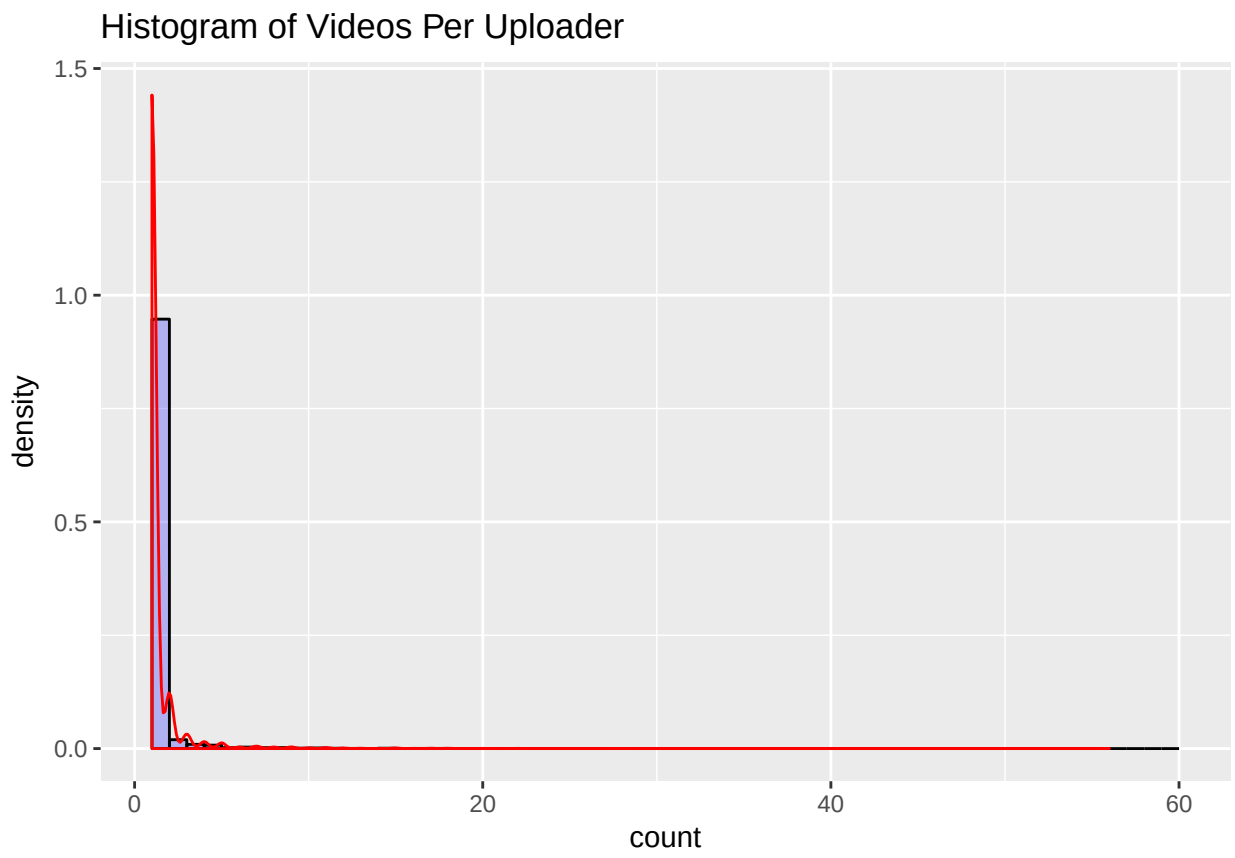
We are not given any information about where this data came from so it is hard to assess if it came from a random sample.

If this was meant to be a random sample of all videos on YouTube, we would not expect any particular uploaders to have too many videos in the sample. 1.3B people use YouTube.[1] Thus, we expect a random sample of our size not to have many users with more than 1 video.

```
ggplot(v[,.(count=.N), by=uploader],
       aes(count)) +
 geom_histogram(aes(y=..density..),
                breaks=1:60,col=1,
                fill="blue",alpha=.25) +
```

[1] https://fortunelords.com/youtube-statistics/

```
 geom_density(alpha=0.25,col=2) +
  ggtitle("Histogram of Videos Per Uploader")
```




```
head(v[,.(count=.N), by=uploader][order(-count),], n=7)
```

```
##             uploader count
## 1:           Pan93bn    56
## 2:          nikodora    28
## 3:   WWEOfficialPPVs    22
## 4:           gar6301    22
## 5:   wishinonastar07    20
## 6:          dermayon    20
## 7: erosentertainment    19
```

```
v[,.(count=.N), by=uploader][,mean(count)]
```

```
## [1] 1.325889
```

Though most users only have 1 video in the sample, 6 users have over 20 and one user has 56. This is a bit unbelievable if it is supposed to be a random sample from YouTube.

However, it may be a random sample of another population.

Given the circumstance, we will assume it is a random sample from some population and note that our inferences only apply to that population.

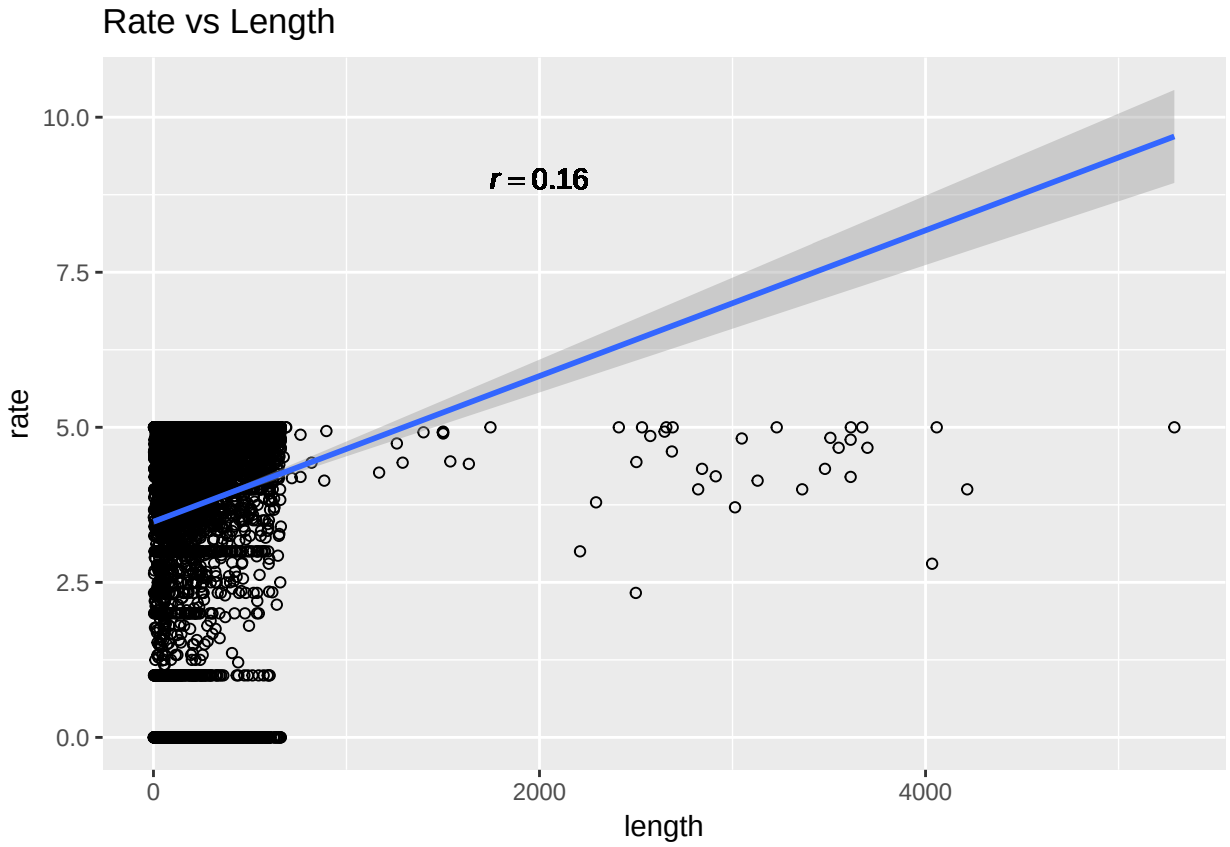
MLR 3 No perfect multicollinearity

The model did not throw an error in R, therefore there is no *perfect* multicolinearity. However, we can still check the relationship between *rate* and *length* to check for high colinearity, which would cause variance of our estimated coefficients to increase.

```
corr_eqn <- function(x,y, digits = 2) {
  corr_coef <- round(cor(x, y, use="pairwise.complete.obs"), digits = digits)
  paste("italic(r) == ", corr_coef)
}
ggplot(v, aes(x=length, y=rate)) +
    geom_point(shape=1) +
    geom_smooth(method=lm) +
    ggtitle("Rate vs Length") +
    geom_text(x = 2000, y = 9,
            label = corr_eqn(v$length,
                             v$rate), parse = TRUE)
```

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

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

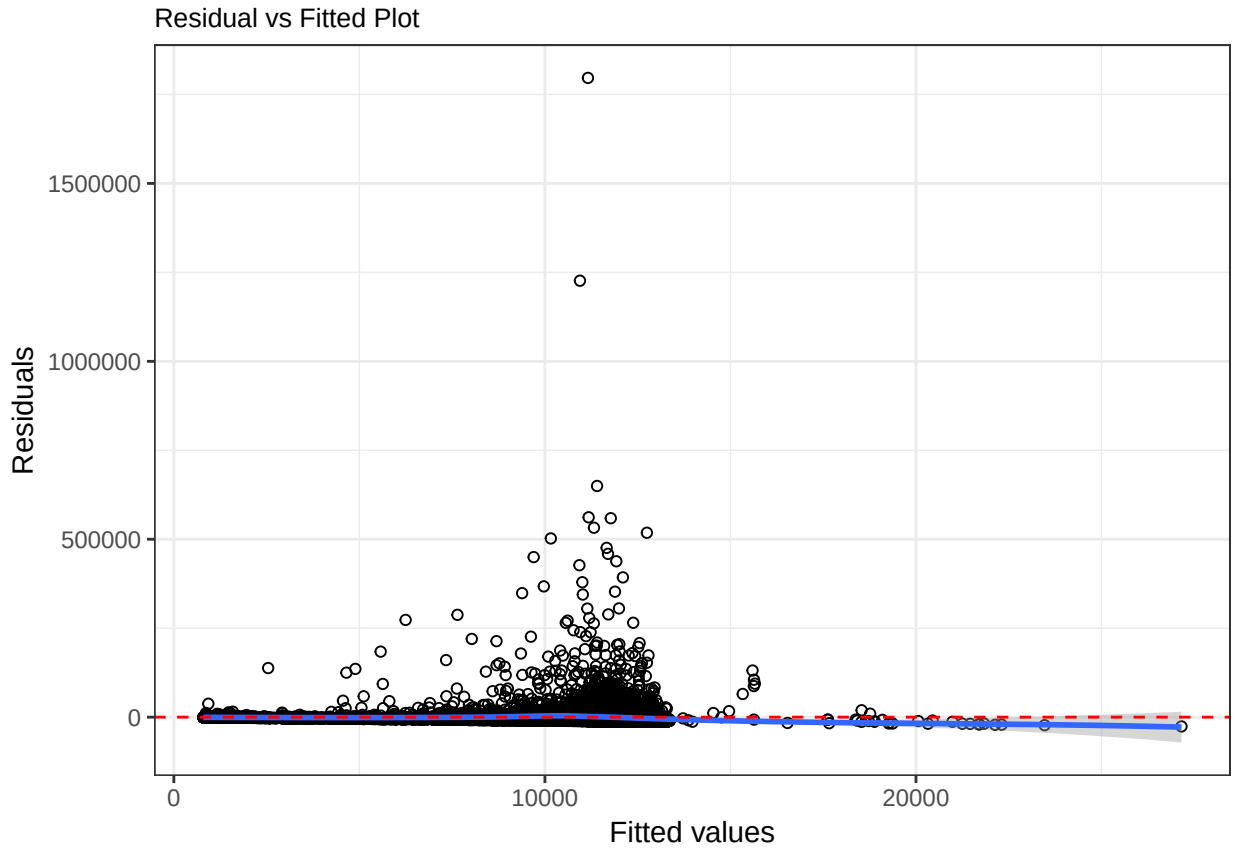


There does not appear to be a strong linear relationship.

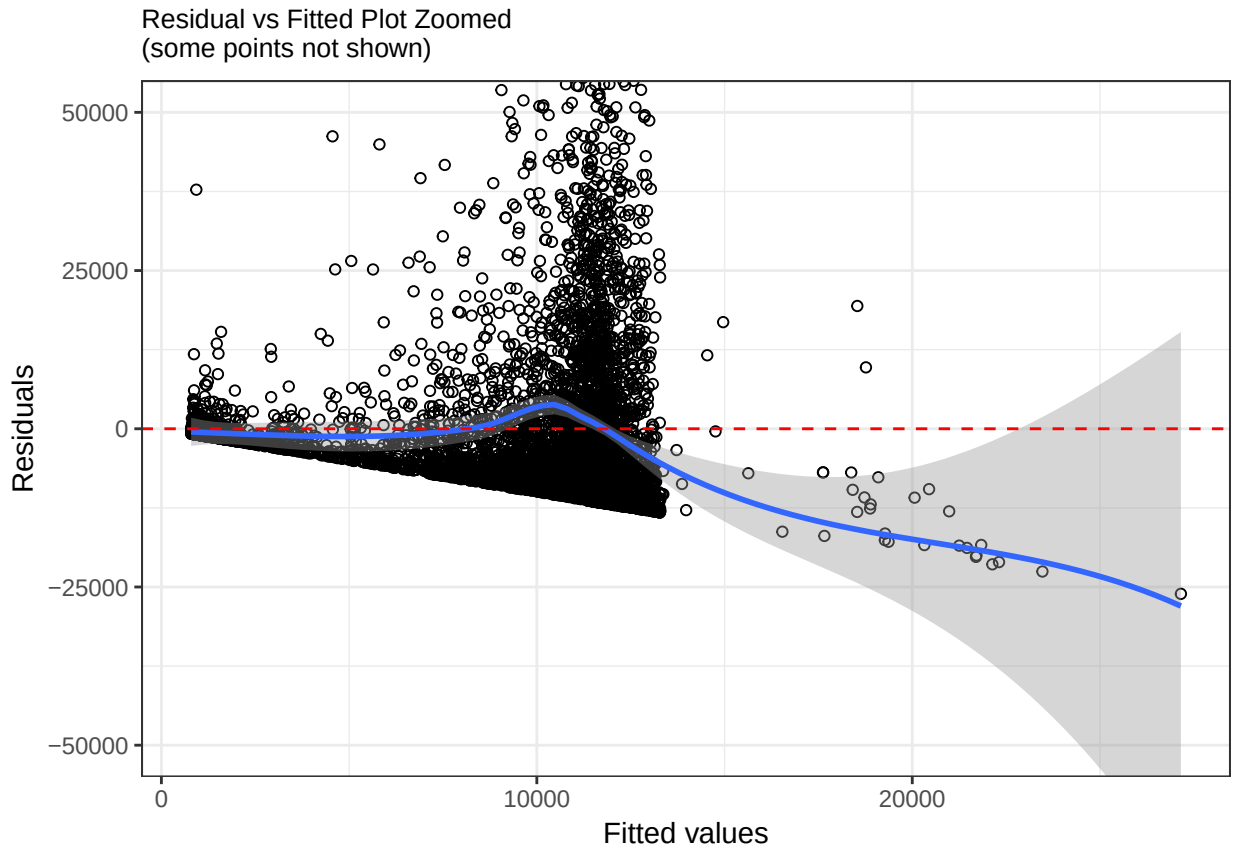
MLR 4 Zero Conditional Mean

To test this assumption, let's examine the residuals vs. fitted values plot.

```
ggplot(m1, aes(.fitted, .resid)) +
  geom_point(shape=1) + stat_smooth(method="loess") +
  geom_hline(yintercept=0, col="red", linetype="dashed") +
  xlab("Fitted values") + ylab("Residuals") +
  ggtitle("Residual vs Fitted Plot") + theme_bw() +
  theme(plot.title = element_text(size = 10))
```




```
ggplot(m1, aes(.fitted, .resid)) +
  geom_point(shape=1) + stat_smooth(method="loess") +
  geom_hline(yintercept=0, col="red", linetype="dashed") +
  xlab("Fitted values") + ylab("Residuals") +
  ggtitle("Residual vs Fitted Plot Zoomed\n(some points not shown)") + theme_bw() +
  theme(plot.title = element_text(size = 10)) + coord_cartesian(ylim=c(-50000,50000))
```

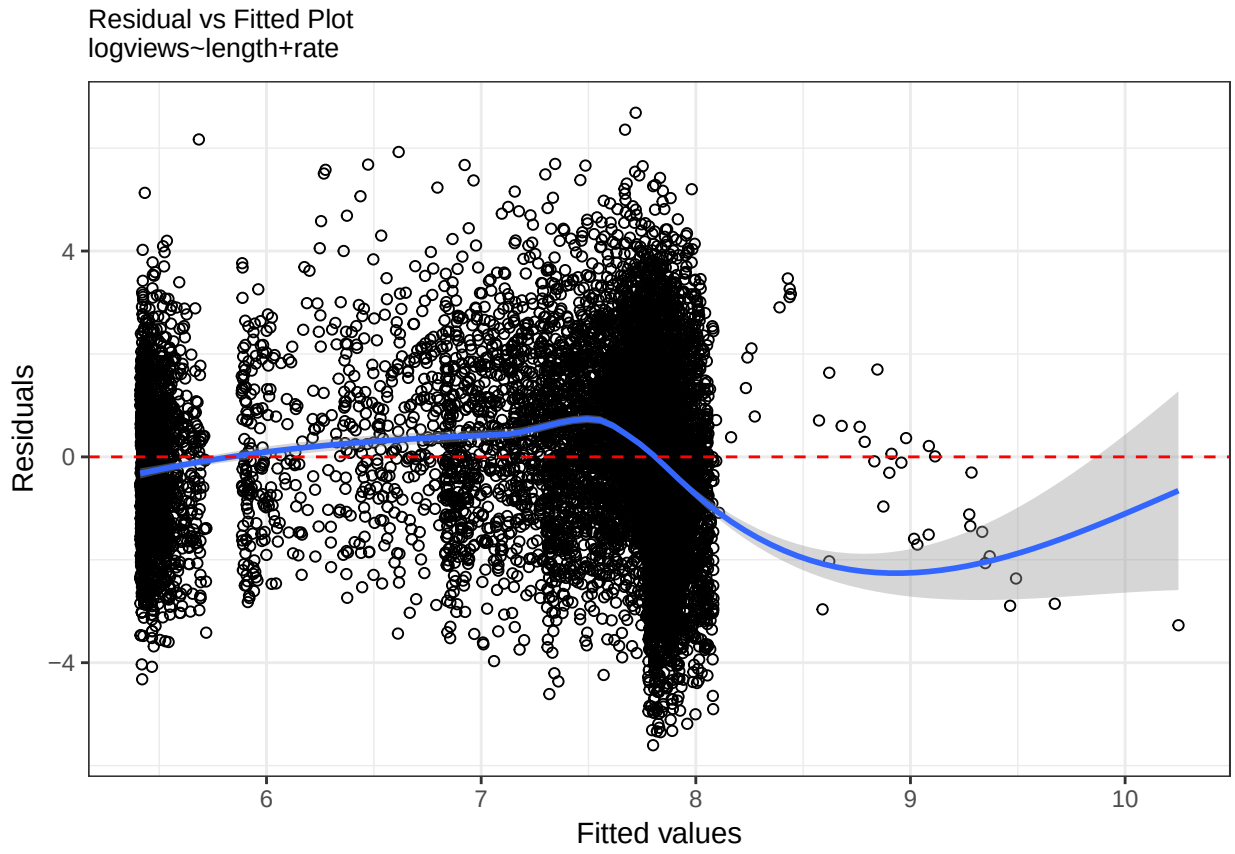


From the zoomed in plot of the residuals vs the fitted values, we see that our estimate of the condition mean of the error is not 0 for all values of the predictors. It appears that the conditional mean decreases, then increases, and decreases again as the fitted value increase.

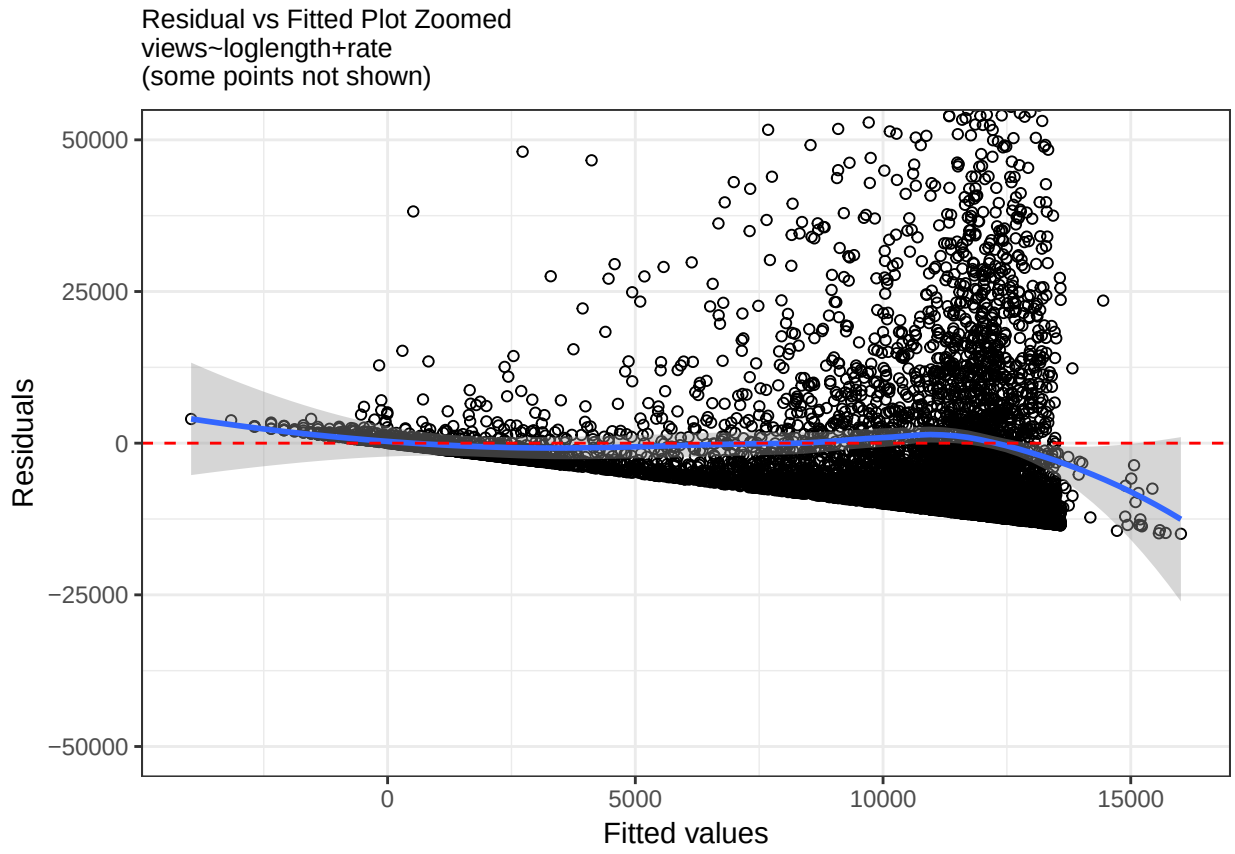
However, we do have a large sample size. So if *rate* and *length* are exogenous (assumption 4'), our coefficients will still be consistent even though they will be biased.

Another option would be to transform the function form. For example, we may want to take the natural log of view or the natural log of length.

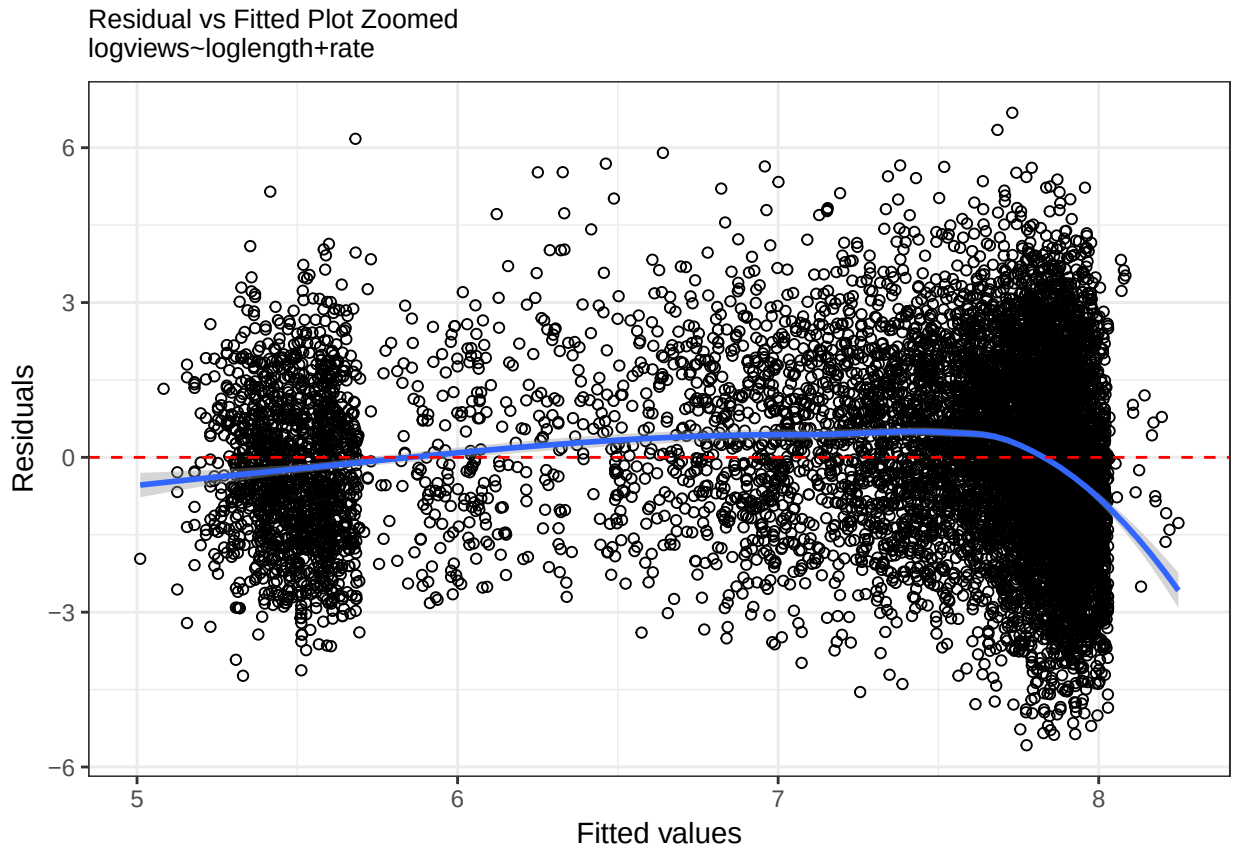
```
m2<-lm(log(views)~length + rate, data=v)
ggplot(m2, aes(.fitted, .resid)) +
  geom_point(shape=1) + stat_smooth(method="loess") +
  geom_hline(yintercept=0, col="red", linetype="dashed") +
  xlab("Fitted values") + ylab("Residuals") +
  ggtitle("Residual vs Fitted Plot\nlogviews~length+rate") + theme_bw() +
  theme(plot.title = element_text(size = 10))
```




```
m3<-lm(views~log(length)+rate, data=v)
ggplot(m3, aes(.fitted, .resid)) +
  geom_point(shape=1) + stat_smooth(method="loess") +
  geom_hline(yintercept=0, col="red", linetype="dashed") +
  xlab("Fitted values") + ylab("Residuals") +
  ggtitle("Residual vs Fitted Plot Zoomed\nviews~loglength+rate\n(some points not shown)") + theme_bw()
  theme(plot.title = element_text(size = 10)) + coord_cartesian(ylim=c(-50000,50000))
```




```
m4<-lm(log(views)~log(length)+rate, data=v)
ggplot(m4, aes(.fitted, .resid)) +
  geom_point(shape=1) + stat_smooth(method="loess") +
  geom_hline(yintercept=0, col="red", linetype="dashed") +
  xlab("Fitted values") + ylab("Residuals") +
  ggtitle("Residual vs Fitted Plot Zoomed\nlogviews~loglength+rate") + theme_bw() +
  theme(plot.title = element_text(size = 10))
```



The second of these three additional models, *m3*, that uses loglength instead of length seems get closer to a zero conditional mean. However, it still doesn't work for high numbers of views, which are the videos that are likely most interesting to study *and* it has a number of videos that it predicts a number of views less than 0. We will examine both this model and *m1*.

MLR 4' Exogeneity

If we have exogeneity, both of the models we are considering will have consistent estimators of the coeficients given our large sample.

It is hard to test for exogenity. Some authors have suggested using the Durbin-Wu-Hausman test that tests for endogenity and considering the variables exogenous if you fail to reject the null hypothesis. However, this is a case of accepting the null instead of failing to reject, which is inappropriate[2]. Also the Hausman test is usually used when you have some variables you known to be exogeneous and are testing for instrumental variables.

Therefore exogeneity comes down to a subjective argument. In our models, length might be exogenous, but ratings are likely endogenous.

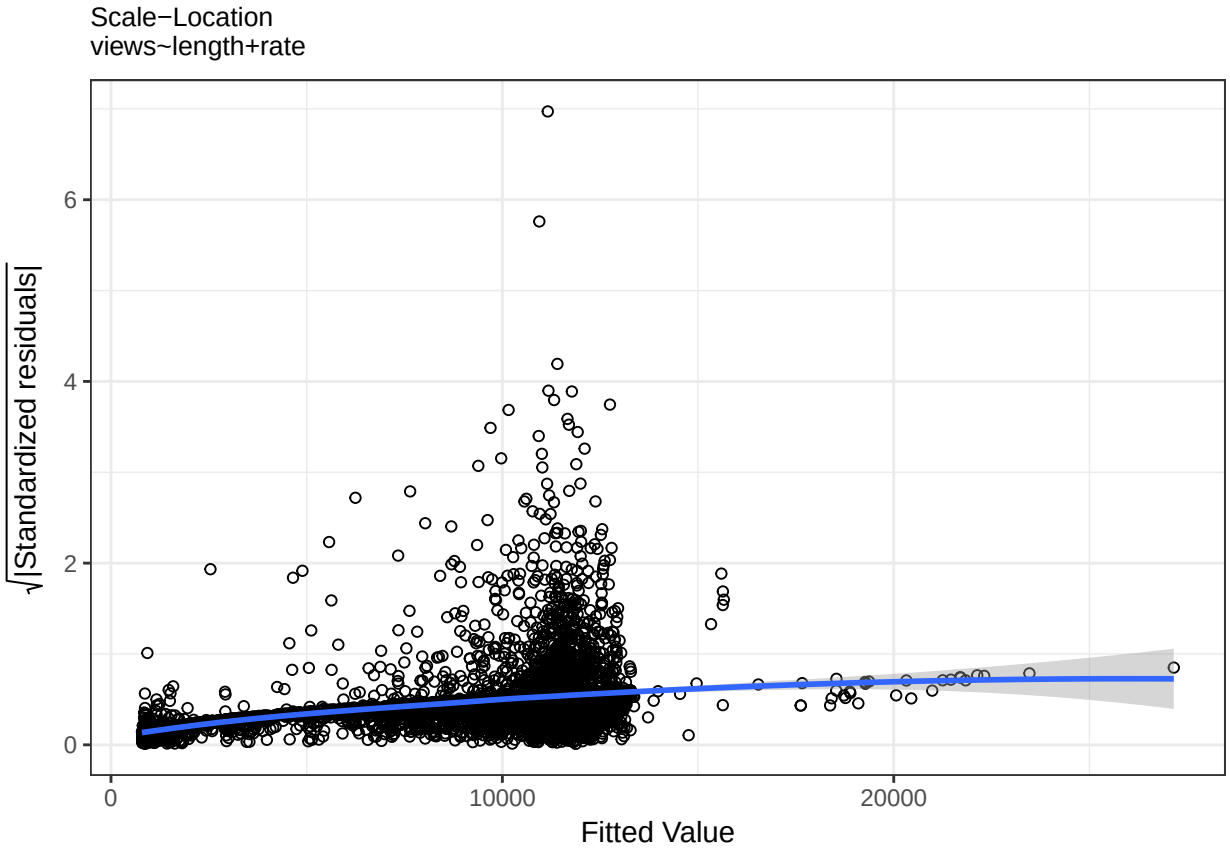
MLR 5 Homoskedasticity

The fifth assumption is homoskedasticity. This assumption is required in order to order for the model to be the Best Linear Unbiased Estimator and to quantify the standard error of the coeficient estimates using the classical formula.

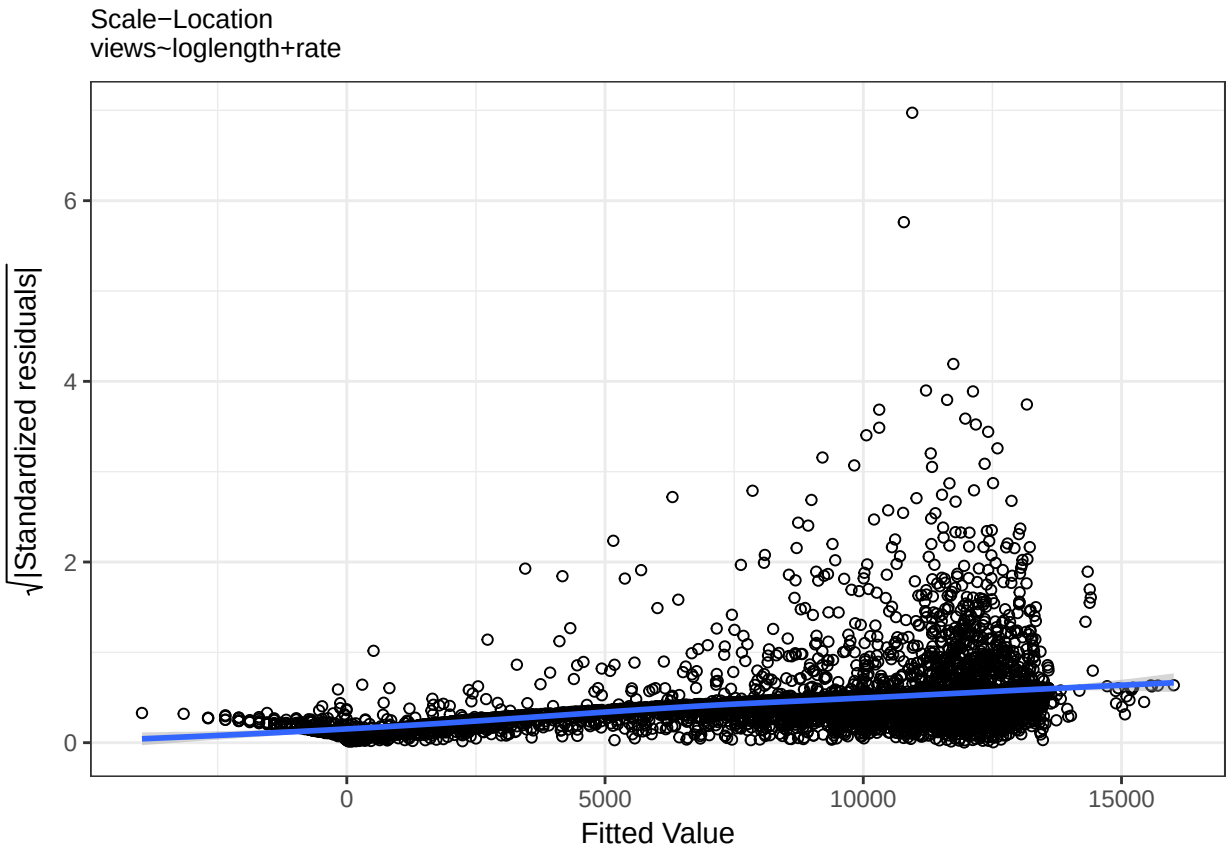
[2]http://marcfbellemare.com/wordpress/10988

To evaluate this assumption, let's look at the scale-location plots.

```
ggplot(m1, aes(.fitted, sqrt(abs(.stdresid)))) +
  geom_point(na.rm=TRUE,shape=1) +
  stat_smooth(method="loess", na.rm = TRUE) +
  xlab("Fitted Value") +
  ylab(expression(sqrt("|Standardized residuals|"))) +
  ggtitle("Scale-Location\nviews~length+rate") + theme_bw() +
  theme(plot.title = element_text(size = 10))
```




```
ggplot(m3, aes(.fitted, sqrt(abs(.stdresid)))) +
  geom_point(na.rm=TRUE,shape=1) +
  stat_smooth(method="loess", na.rm = TRUE) +
  xlab("Fitted Value") +
  ylab(expression(sqrt("|Standardized residuals|"))) +
  ggtitle("Scale-Location\nviews~loglength+rate") + theme_bw() +
  theme(plot.title = element_text(size = 10))
```



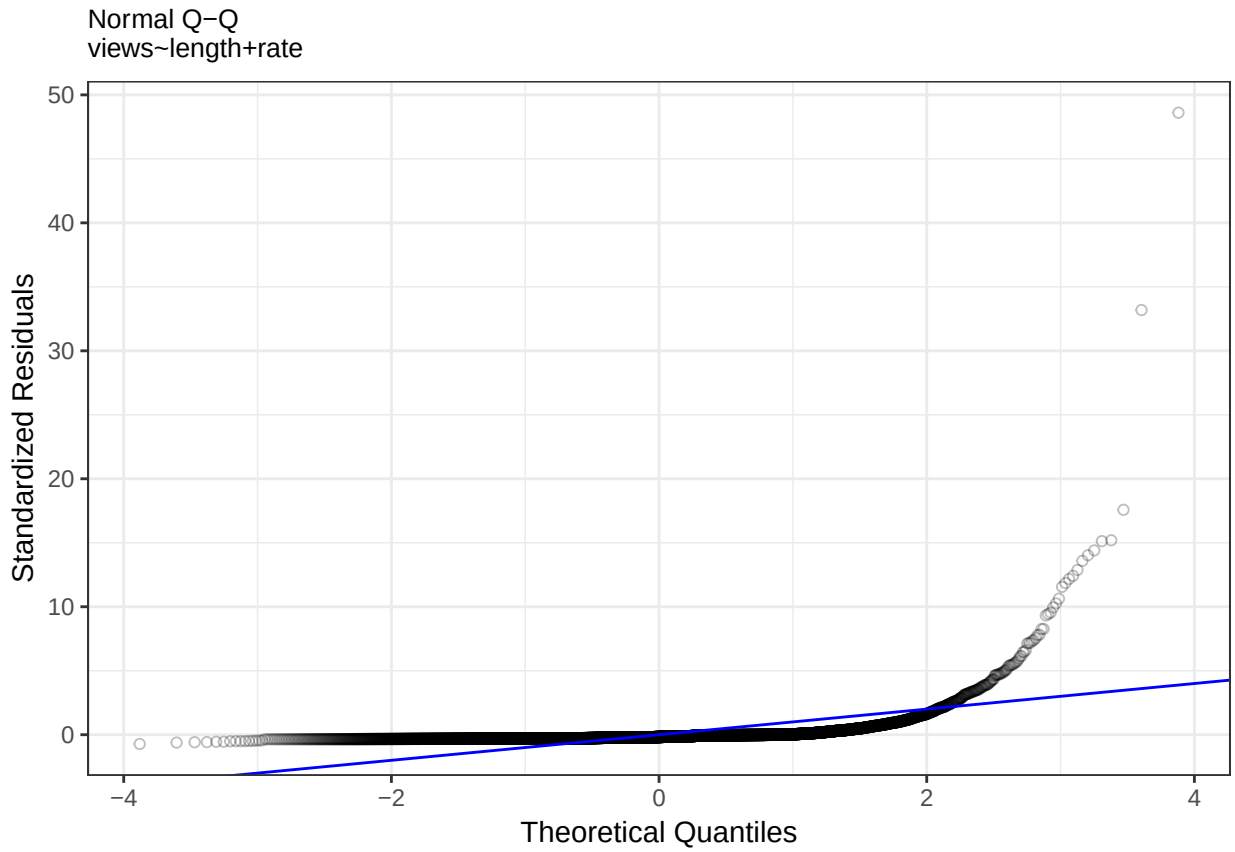
Both the (i) residuals vs fitted values and (ii) scale-location plots indicate there is heteroskedasticity in both models.

As a result, we will have to use heteroskedasticity-robust standard errors (Huber-White). These will still be consistent with our large data set given assumptions 1-3 and 4'.

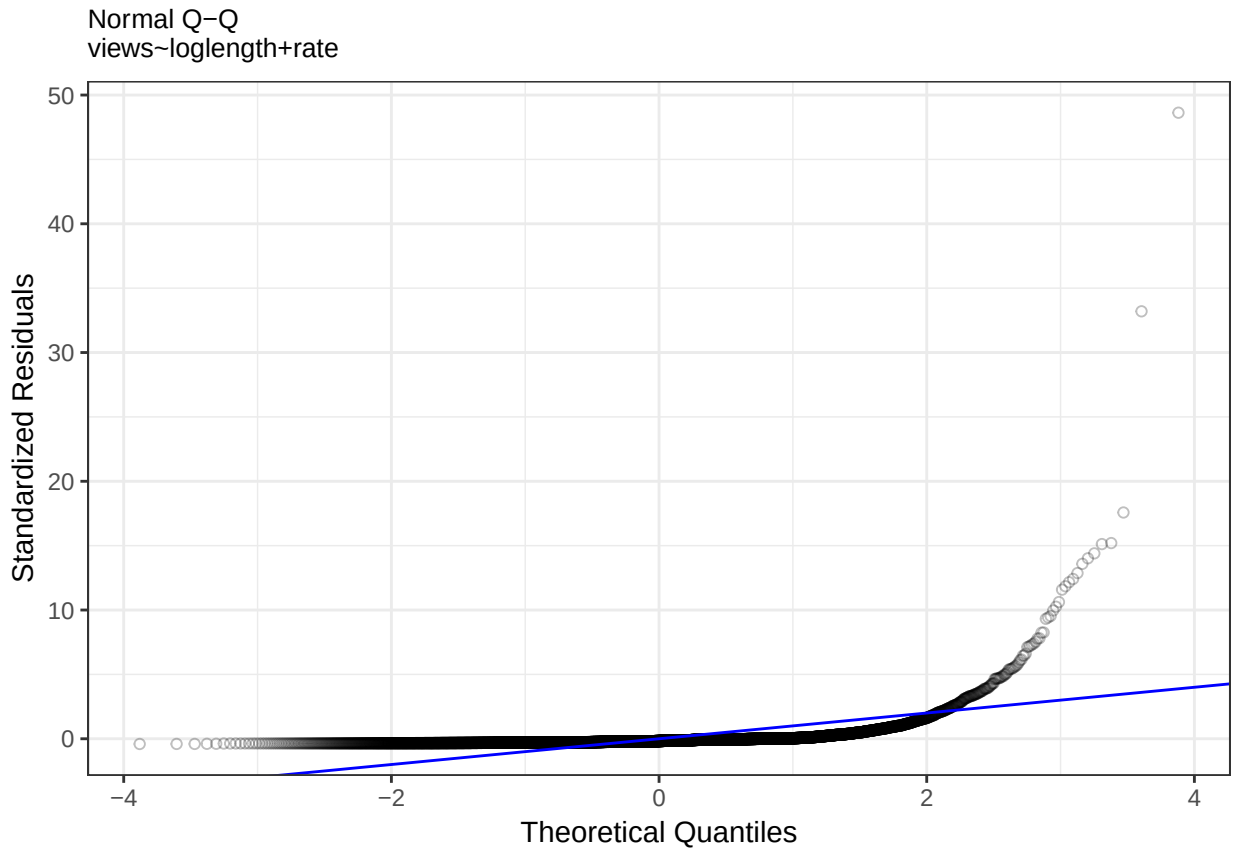
MLR 6 Normality of Errors

The final CLM assumption is that the errors follow a nomral distribution. The best way to evaluate this assumption is with a Normal Q-Q plot of the residuals.

```
ggplot(m1, aes(qqnorm(.stdresid, plot.it = F)[[1]], .stdresid)) +
  geom_point(na.rm = TRUE,shape=1,alpha=.25) +
  geom_abline(color="blue") + xlab("Theoretical Quantiles") +
  ylab("Standardized Residuals") + ggtitle("Normal Q-Q\nviews~length+rate") +
  theme_bw() + theme(plot.title = element_text(size = 10))
```




```
ggplot(m3, aes(qqnorm(.stdresid, plot.it = F)[[1]], .stdresid)) +
  geom_point(na.rm = TRUE,shape=1,alpha=.25) +
  geom_abline(color="blue") + xlab("Theoretical Quantiles") +
  ylab("Standardized Residuals") + ggtitle("Normal Q-Q\nviews~loglength+rate") +
  theme_bw() + theme(plot.title = element_text(size = 10))
```



The curvature of the Normal Q-Q plot for the residuals indicates the error is far from normally distributed for both models. However, we have a large sample and therefore can rely on the CLT that implies the coefficients will have an asymptotically normal sampling distribution (given we have met assumptions 1, 2, 3, and 4')

3 Model Results

```
se.m1 = sqrt(diag(vcovHC(m1)))
quantile(v$length,c(.99),na.rm=T)
```

```
## 99%
## 652
```

```
se.m3 = sqrt(diag(vcovHC(m3)))
```

```
stargazer(m1,m3, omit.stat = "f",
          se=list(se.m1,se.m3),
          star.cutoffs = c(.05,.01,.001),
          header=F)
```

Model 1 views ~ length + rate

The coefficient on length is 2.996 and is marginally significant. 99 percent of videos in the sample have length less than or equal to 652. This leads me to believe length is measured in seconds because a majority of YouTube videos are less than 10 minutes. To tell whether this coefficient is practically significant or not, we need some measure of what it means to have a significant change on the number of views. If we are trying to

Table 1:

	Dependent variable:	
	views	
	(1)	(2)
length	2.996*	
	(1.223)	
log(length)		1,164.624***
		(255.241)
rate	2,103.880***	1,998.631***
	(126.528)	(127.689)
Constant	789.033**	−3,965.785***
	(277.176)	(1,150.226)
Observations	9,609	9,609
R^2	0.011	0.012
Adjusted R^2	0.011	0.012
Residual Std. Error (df = 9606)	36,962.390	36,950.620
Note:	*p<0.05; **p<0.01; ***p<0.001	

predict viral views, for example, it is not significant because a majority of videos will only be predicted to have about 3X650=1950 or fewer more views from length (vs the baseline of a length 0 video). However, if you are examining smaller fan bases, an additional thousand or so views is quite practically significant.

The coefficient on rating is 2103.880 and is very significant. This isn't surprising that more highly rated videos are viewed more often. Again the practical significance depends on our standard of number of views, but it is clear that rating has more of a practical impact than length. Each unit change in rating as more of an effect than a change in length equal to the range where 99% of the data points fall.

Model 3 views ~ loglength + rate

Again we see both slope coefficients are positive and significant. However, loglength now has a very significant coefficient instead of a marginally significant one. In terms of prediction, it appears the models perform about the same with R square values that are very similar. The loglength coefficient indicates that a 1% increase in length yields approximately 1,165 more views. This seems practically significant, especially when videos have shorter lengths and can have large percentage increases in length.

Overall, I believe model 1 to be better than model 3. This is because the two models face the same drawbacks in terms of the CLM assumptions (though model 3 is closer to meeting assumption 4). The models have nearly equal prediction power. However, model 3 yields the odd prediction value of negative views and the coefficient on length is hard to interpret for low values of length.