Regression

Mason Cushing

2023-02-18

Linear regression works by looking at the data and finding the line of best fit. Its main weaknesses are its high bias (it always expects to find a line). Its main strength is that is is very very good at finding a line if it is there. Notice that I had to trim a good number of cases that had a ridiculous value in the data frame (i.e. someone taking an Uber around the world a few times or having a free ride). Data found here: https://www.kaggle.com/datasets/yasserh/uber-fares-dataset (https://www.kaggle.com/datasets/yasserh/uber-fares-dataset)

```
uber <- read.csv("~/Downloads/uber.csv")
uber$key<-NA
uber$X<-NA
uber$pickup_datetime<-NA
latdiff<-(uber$dropoff_latitude-uber$pickup_latitude)^2
longdiff<-(uber$dropoff_longitude-uber$pickup_longitude)^2
distances<-sqrt(latdiff+longdiff)
uber$distance<-distances
uber<-uber[uber$distance < 0.5,]
uber<-uber[uber$distance != 0,]
uber<-uber[uber$fare_amount < 100,]
uber<-uber[uber$fare_amount > 0,]
uber<-uber[uber$passenger_count < 10,]
uber<-uber[uber$passenger_count > 0,]
uber<-uber[is.na(uber$fare_amount) == FALSE,]

i <- sample(1:nrow(uber), 0.8*nrow(uber), replace=FALSE)
dtrain <- uber[i,]
dtest <- uber[-i,]
```

I will now use a few functions to explore the data. These figures are very useful later when we look at the linear models.

```
mean(uber$fare_amount)
```

```
## [1] 11.30465
```

```
median(uber$fare_amount)
```

```
## [1] 8.5
```

```
sd(uber$fare_amount)
```

```
## [1] 9.360695
```

```
mean(uber$distance)
```

```
## [1] 0.03429341
```

```
median(uber$distance)
```

```
## [1] 0.02205756
```

```
sd(uber$distance)
```

```
## [1] 0.03837389
```

```
mean(uber$passenger_count)
```

```
## [1] 1.690035
```

```
median(uber$passenger_count)
```

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

```
sd(uber$passenger_count)
```

```
## [1] 1.305717
```

These are two plots that show things about the data. Notice that the five means show that passenger count doesn't really affect fare amount.

```
mean(uber[uber$passenger_count == 1,]$fare_amount)
```

```
## [1] 11.17492
```

```
mean(uber[uber$passenger_count == 2,]$fare_amount)
```

```
## [1] 11.73478
```

```
mean(uber[uber$passenger_count == 3,]$fare_amount)
```

```
## [1] 11.43564
```

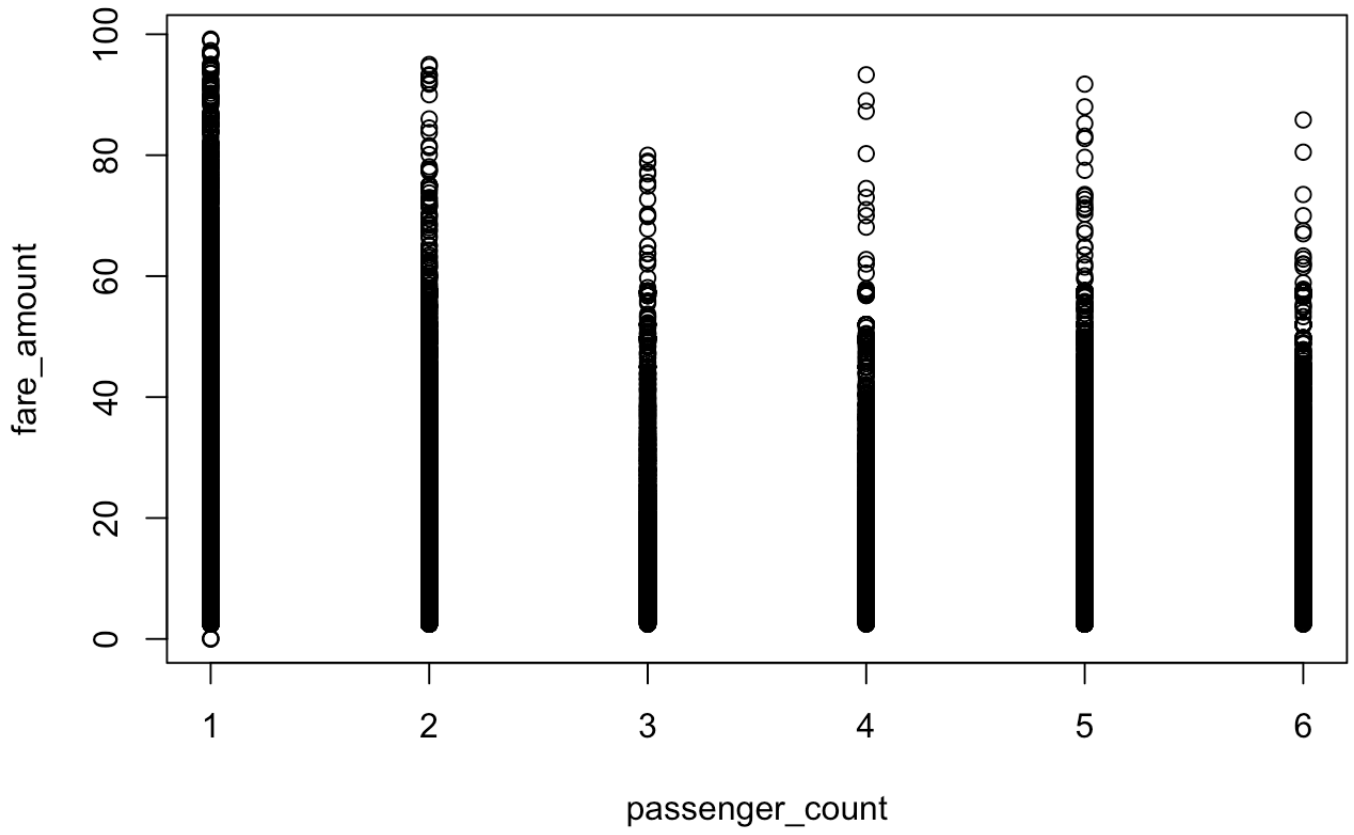
```
mean(uber[uber$passenger_count == 4,]$fare_amount)
```

```
## [1] 11.64296
```

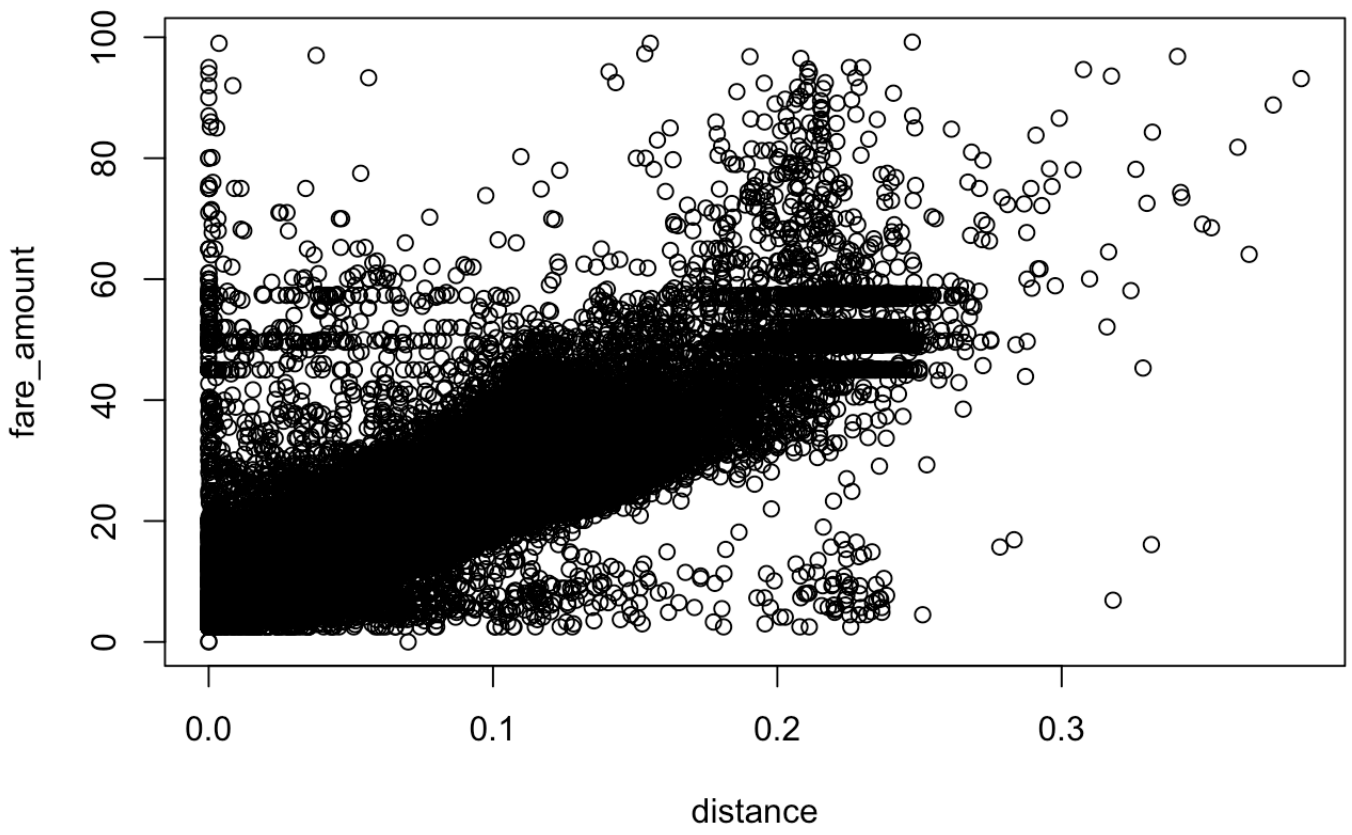
```
mean(uber[uber$passenger_count == 5,]$fare_amount)
```

```
## [1] 11.2304
```

```
plot(fare_amount~passenger_count,data=uber)
```




```
plot(fare_amount~distance,data=uber)
```

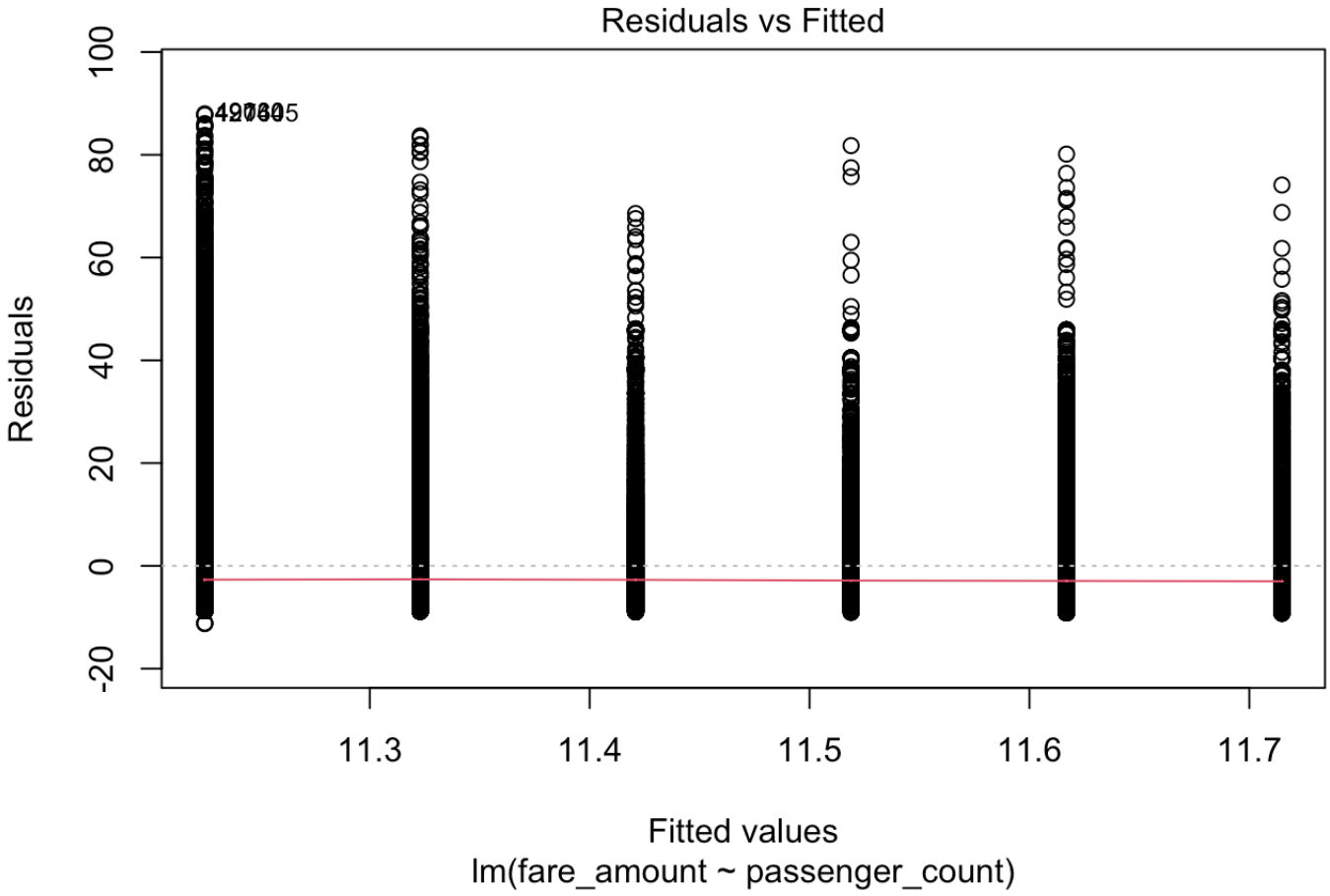


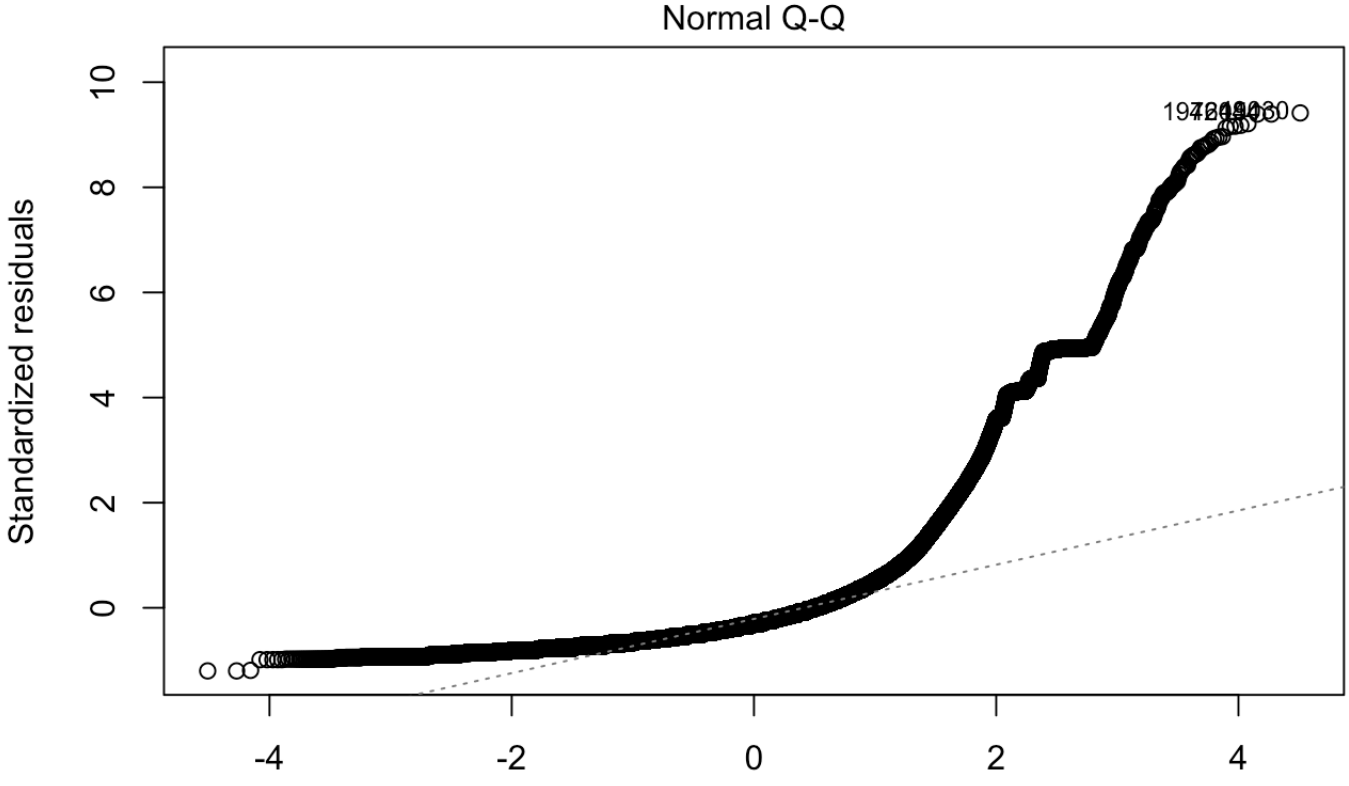
Here is our first model. We will summarize what R is learning here in code below. The human understanding is that the model's line is very flat (it says that each passenger increases fare by 9 cents, so an insignificant value). Not a good model, especially because the R^2 value is very low.

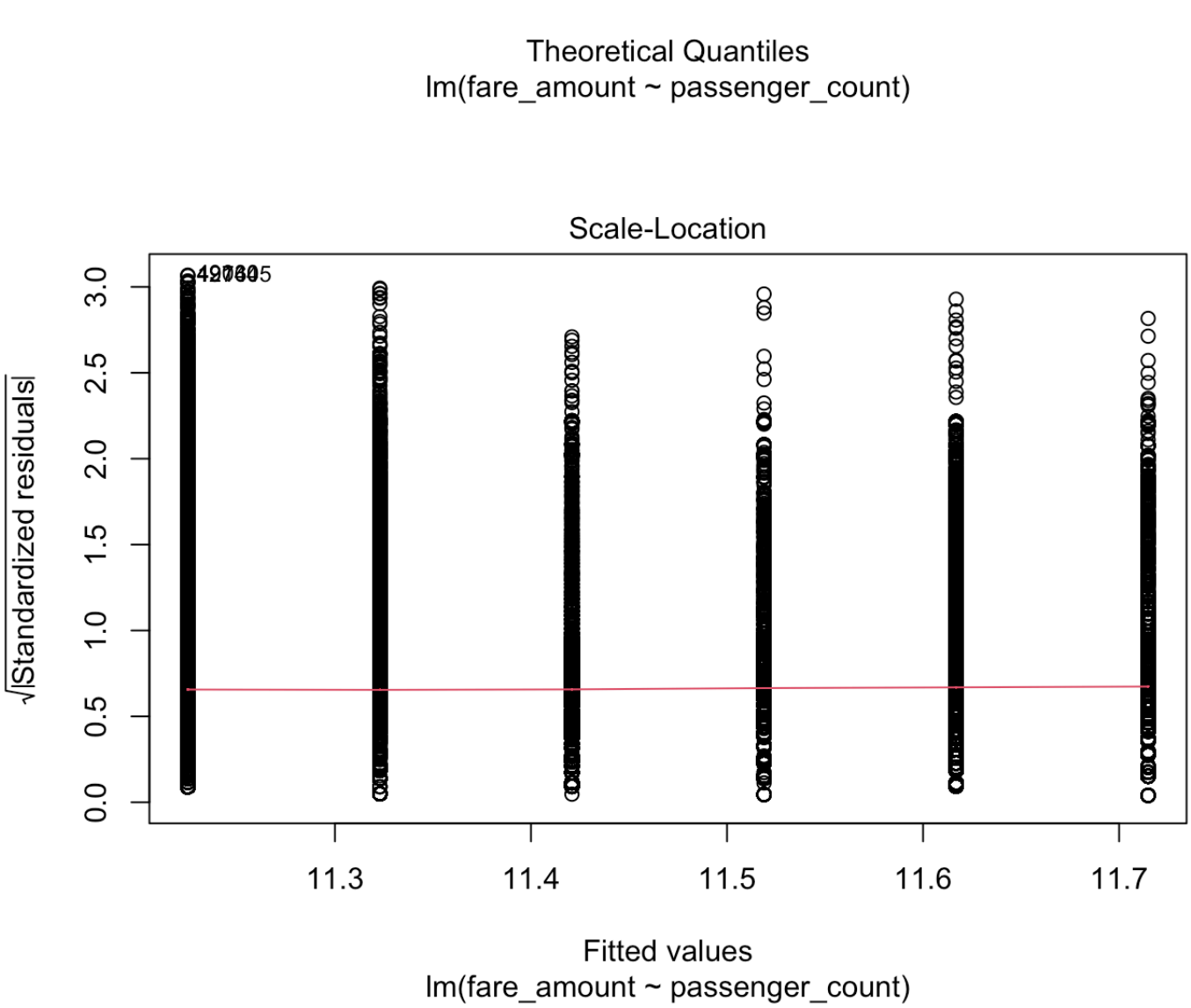
```
model <- lm(fare_amount~passenger_count,data=dtrain)
summary(model)
```

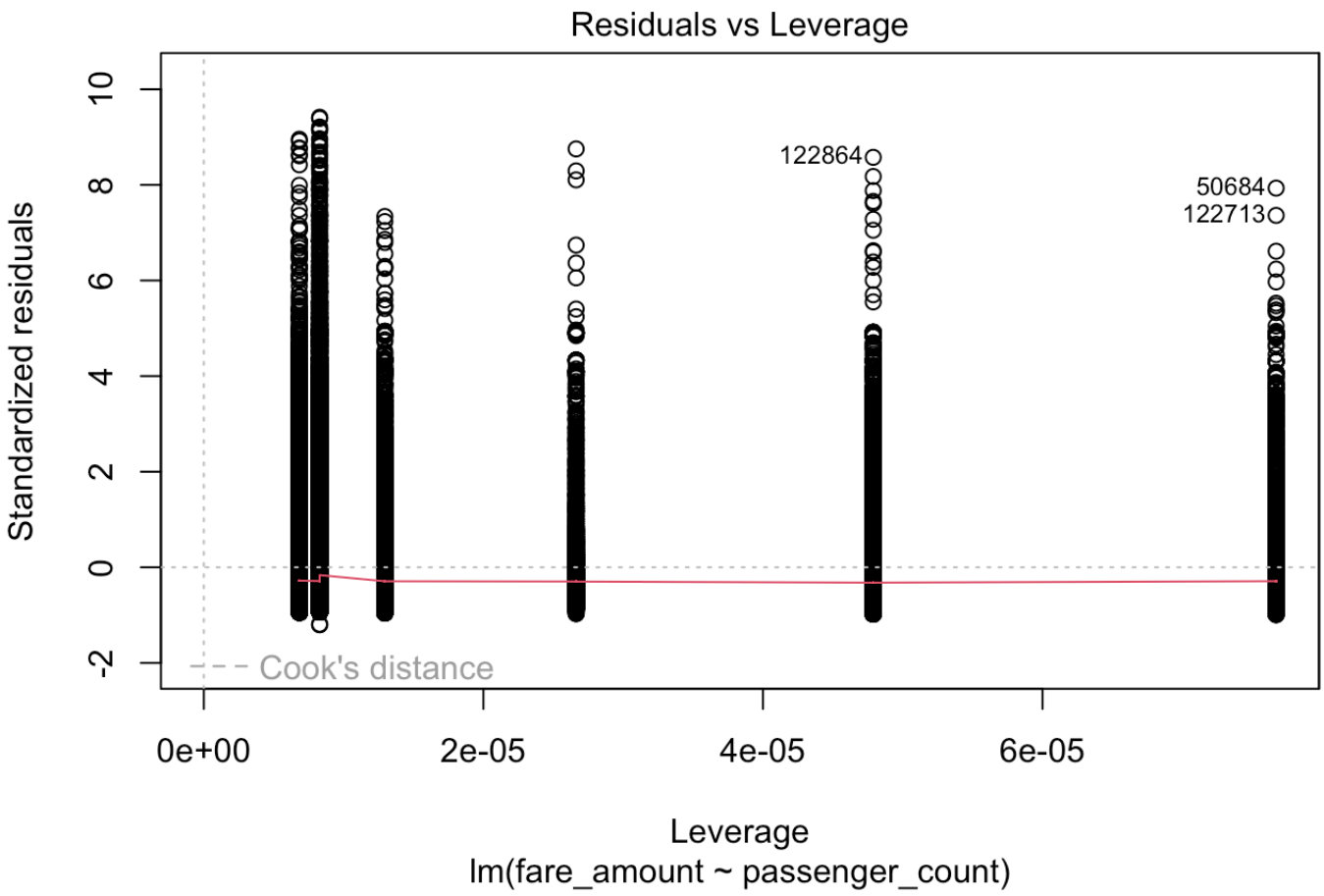
```
##
## Call:
## lm(formula = fare_amount ~ passenger_count, data = dtrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.215  -5.225  -2.725   1.275  87.975
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     11.12698    0.03887  286.23  < 2e-16 ***
## passenger_count  0.09797    0.01818    5.39 7.07e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.344 on 154472 degrees of freedom
## Multiple R-squared:  0.000188,   Adjusted R-squared:  0.0001815
## F-statistic: 29.05 on 1 and 154472 DF,  p-value: 7.067e-08
```

```
plot(model)
```


Residuals vs Fitted
Residuals
100
80
60
40
20
0
-20
11.3
11.4
11.5
11.6
11.7
Fitted values
lm(fare_amount ~ passenger_count)


Normal Q-Q
Standardized residuals
10
8
6
4
2
0
-4
-2
0
2
4


Theoretical Quantiles
lm(fare_amount ~ passenger_count)
Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
11.3
11.4
11.5
11.6
11.7
Fitted values
lm(fare_amount ~ passenger_count)



The above plot shows that the model created with just this predictor is a very bad model. The R squared value is much too small to be meaningful.

```
p<-predict(model,newdata=dtest)
mean((p-dtest$fare_amount)^2)
```

```
## [1] 88.7873
```

```
summary(model)$r.squared
```

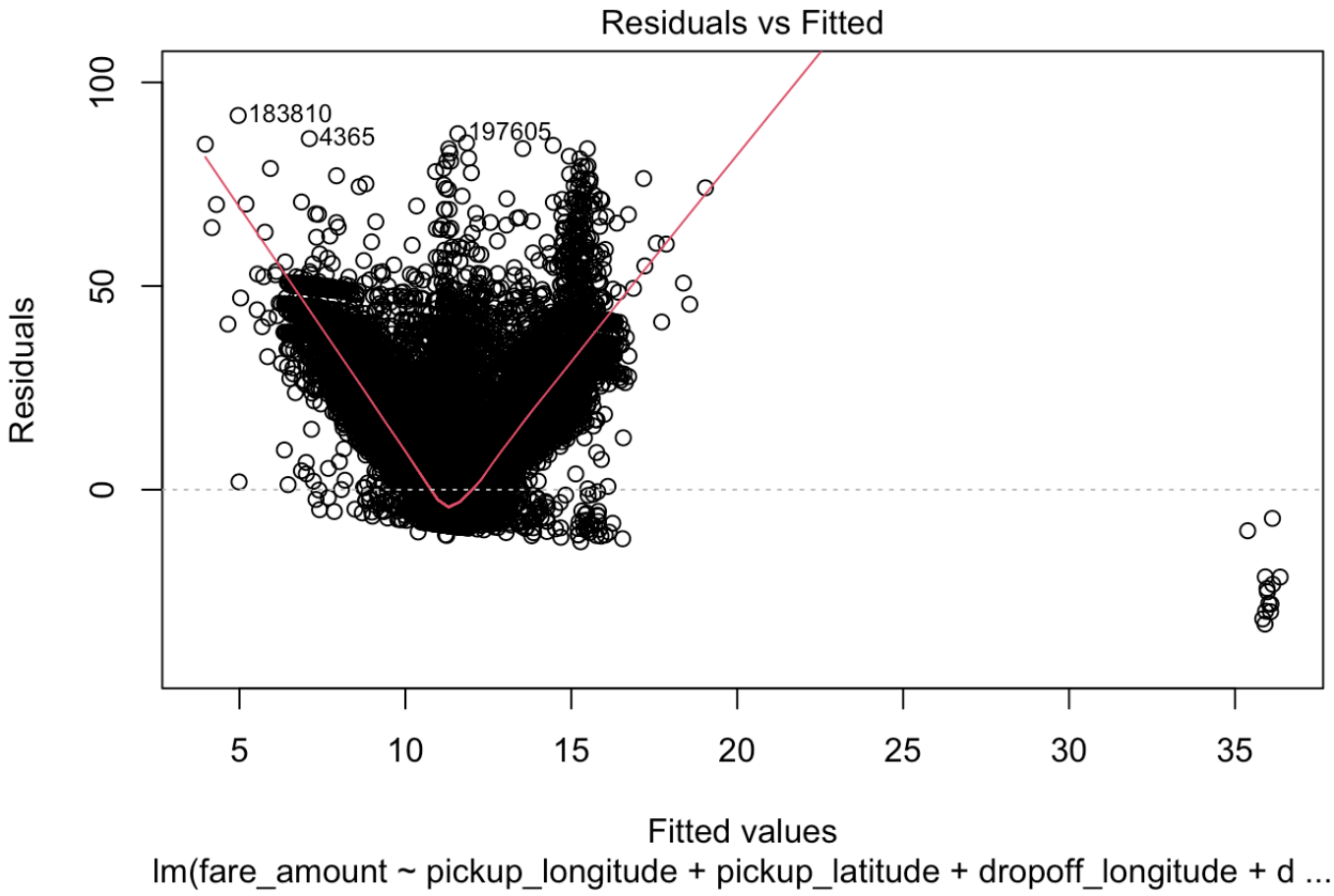
```
## [1] 0.000188018
```

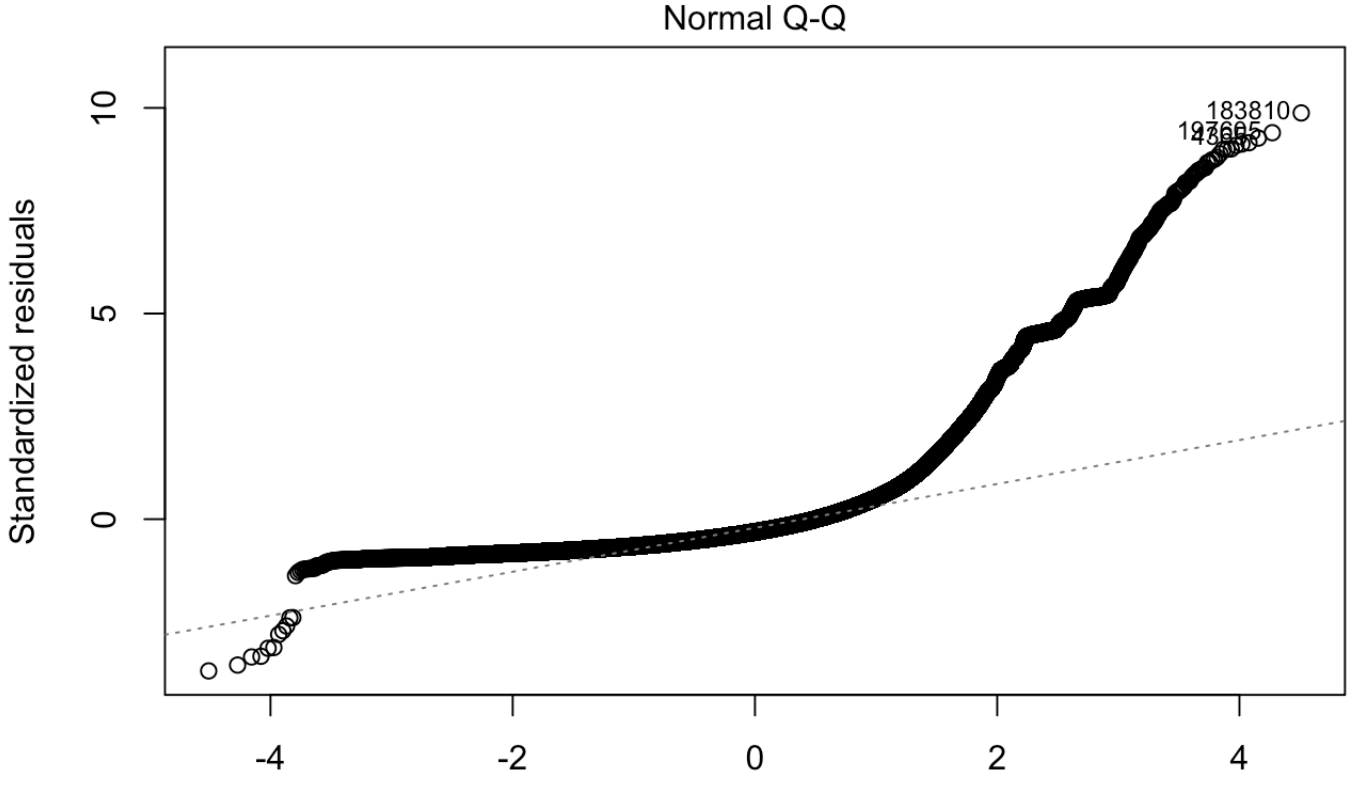
R^2 is close enough to zero to warrant scientific notation. Perhaps adding more variables will improve the model.

```
model2<-lm(fare_amount~pickup_longitude+pickup_latitude+
            dropoff_longitude+dropoff_latitude+passenger_count,data=dtrain)
summary(model2)
```

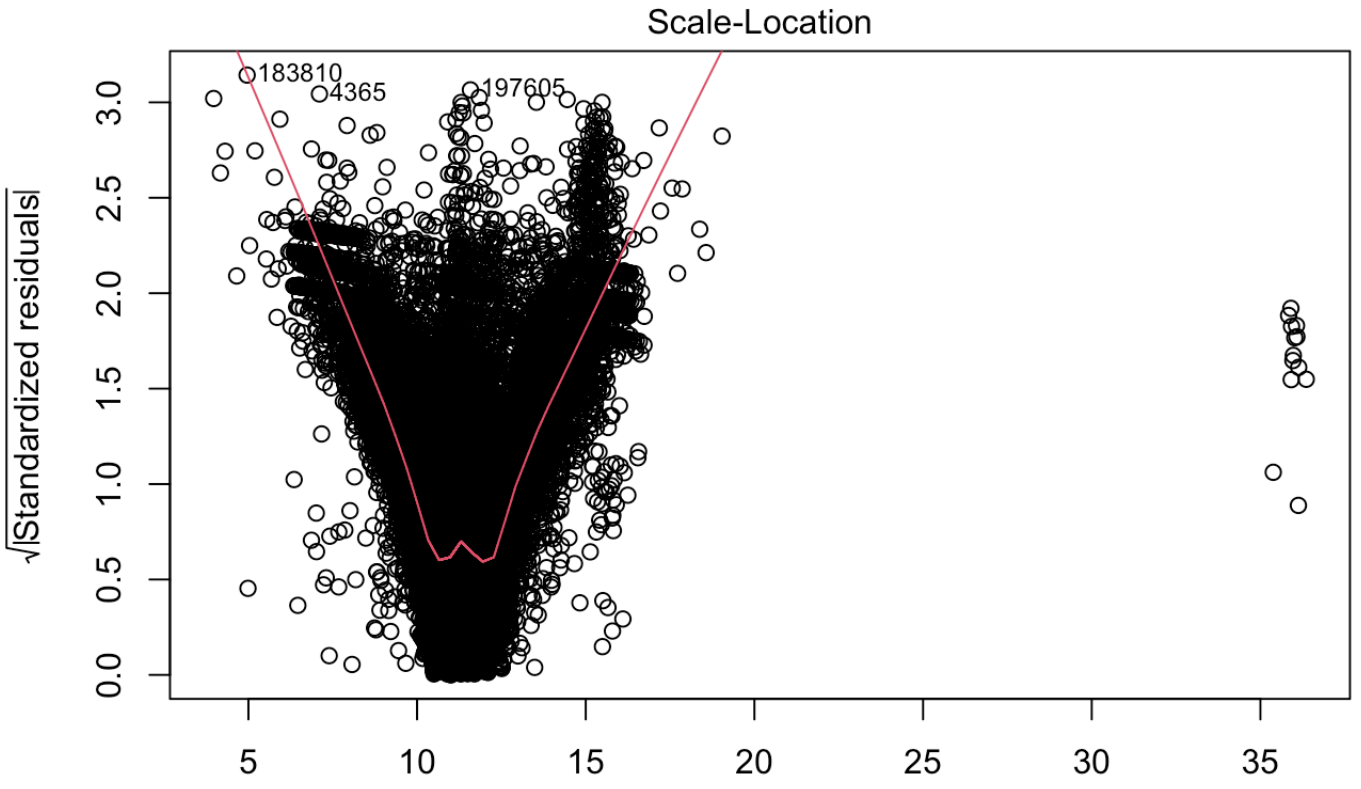
```
##
## Call:
## lm(formula = fare_amount ~ pickup_longitude + pickup_latitude +
##     dropoff_longitude + dropoff_latitude + passenger_count, data = dtrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.003  -5.313  -2.888   1.384  91.869
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        35.79287    2.55584  14.004  < 2e-16 ***
## pickup_longitude   21.08112    0.58290  36.166  < 2e-16 ***
## pickup_latitude    -0.45319    0.75898  -0.597    0.550
## dropoff_longitude -20.35394    0.58237 -34.950  < 2e-16 ***
## dropoff_latitude    1.16844    0.75703   1.543    0.123
## passenger_count     0.09743    0.01810   5.383 7.34e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.303 on 154468 degrees of freedom
## Multiple R-squared:  0.008921,   Adjusted R-squared:  0.008889
## F-statistic: 278.1 on 5 and 154468 DF,  p-value: < 2.2e-16
```

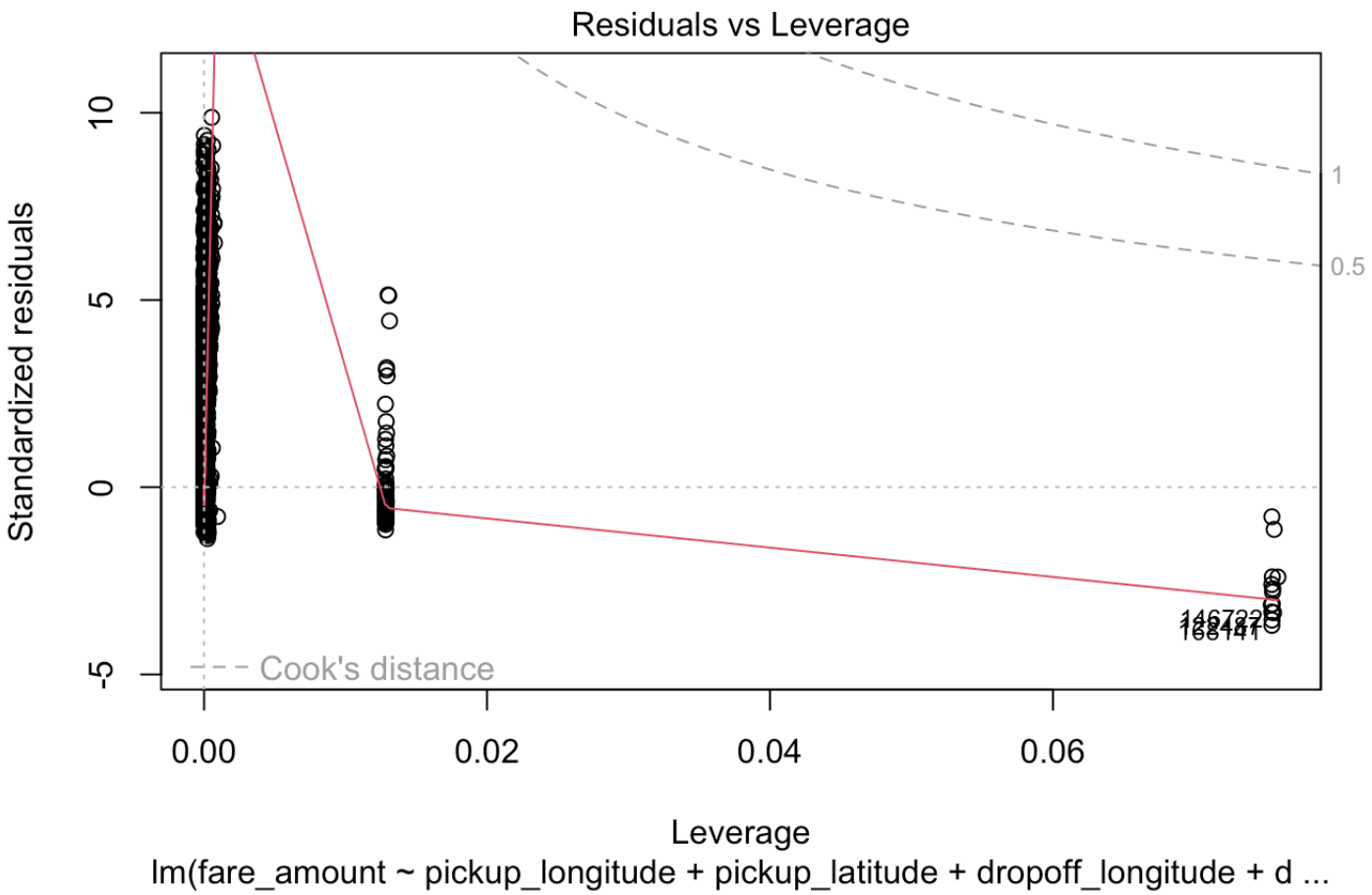
```
plot(model2)
```


Residuals vs Fitted
Residuals
100
50
0
183810
4365
197605
5
10
15
20
25
30
35
Fitted values
lm(fare_amount ~ pickup_longitude + pickup_latitude + dropoff_longitude + d ...


Normal Q-Q
Standardized residuals
10
5
0
183810
-4
-2
0
2
4

Theoretical Quantiles
lm(fare_amount ~ pickup_longitude + pickup_latitude + dropoff_longitude + d ...





This did technically help, but the model isn't very good yet (see the R^2 value and the above commentary). Intuitively this makes sense. The longitude won't affect the fare in a linear fashion (especially since the world isn't flat).

```
p2<-predict(model2,newdata=dtest)
mean((p2-dtest$fare_amount)^2)
```

```
## [1] 87.6119
```

```
summary(model2)$r.squared
```

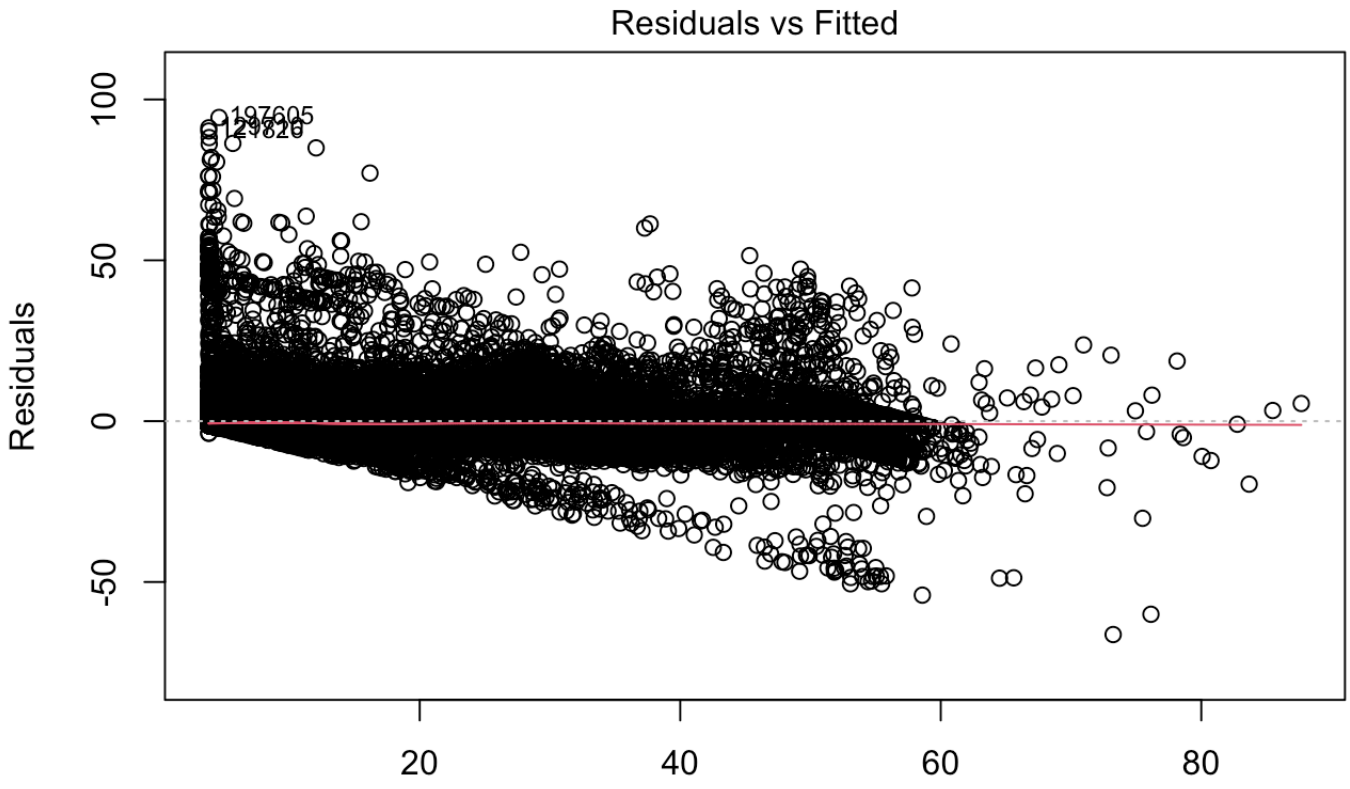
```
## [1] 0.008921494
```

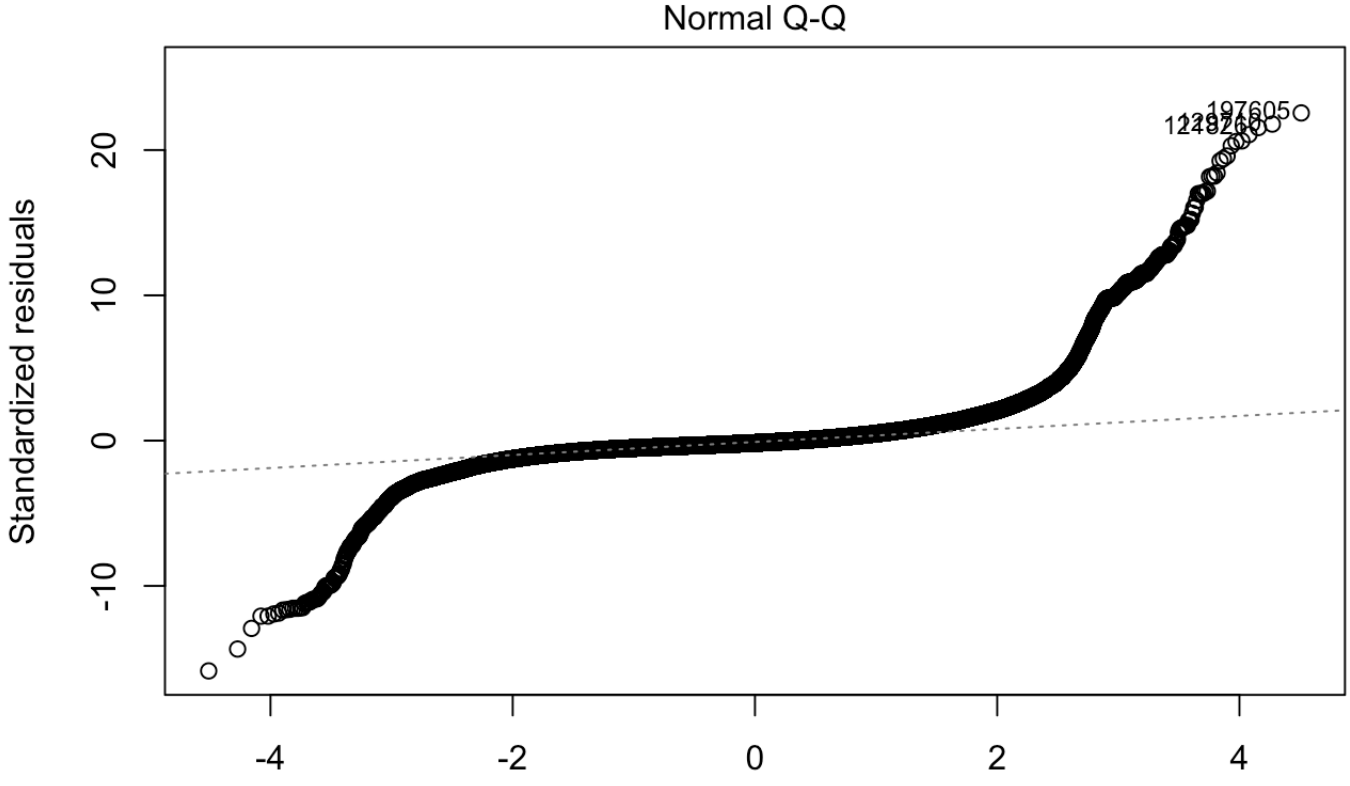
Let's try using the distance. Intuitively, this should help our model a lot.

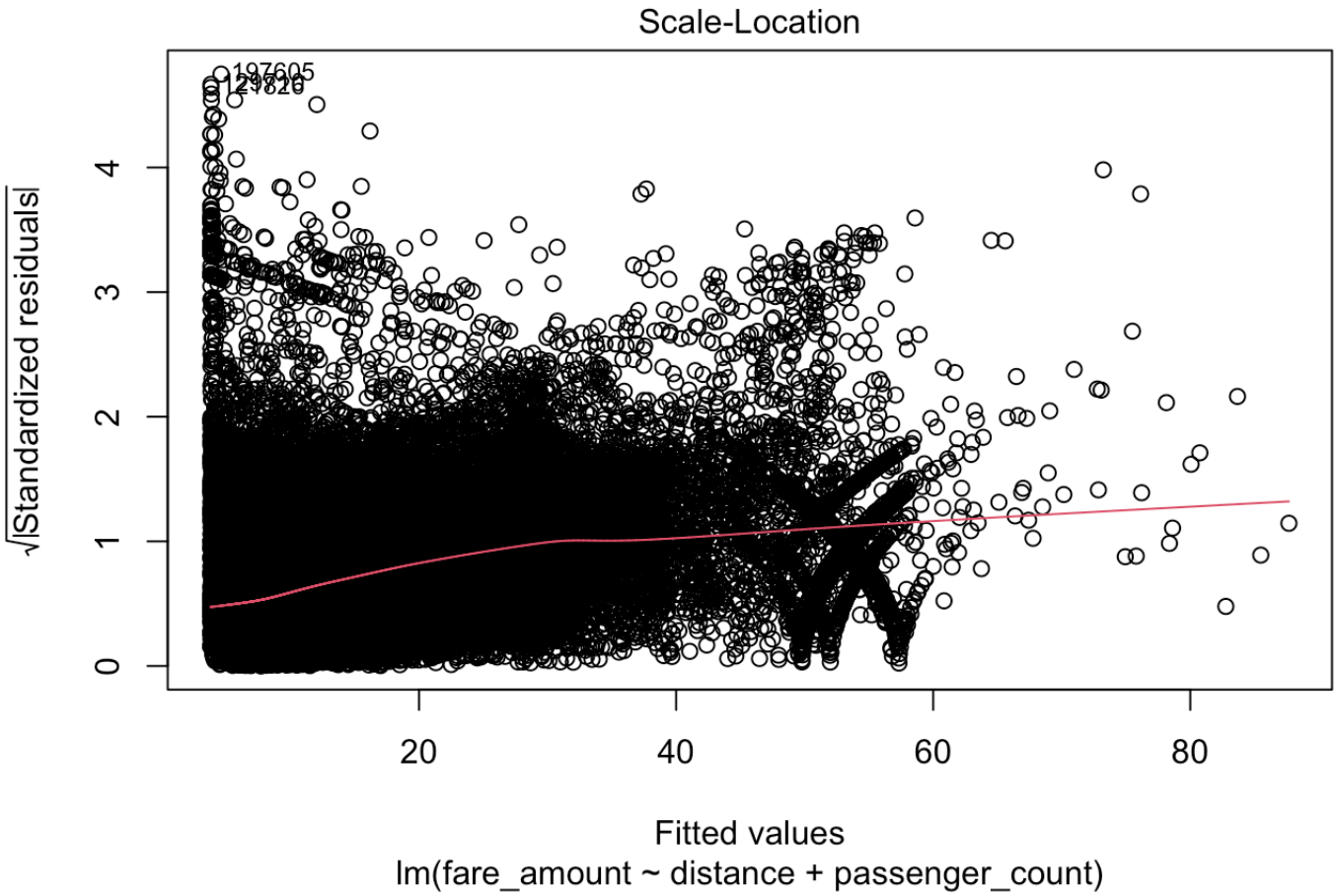
```
model3<-lm(fare_amount~distance+passenger_count,data=dtrain)
summary(model3)
```

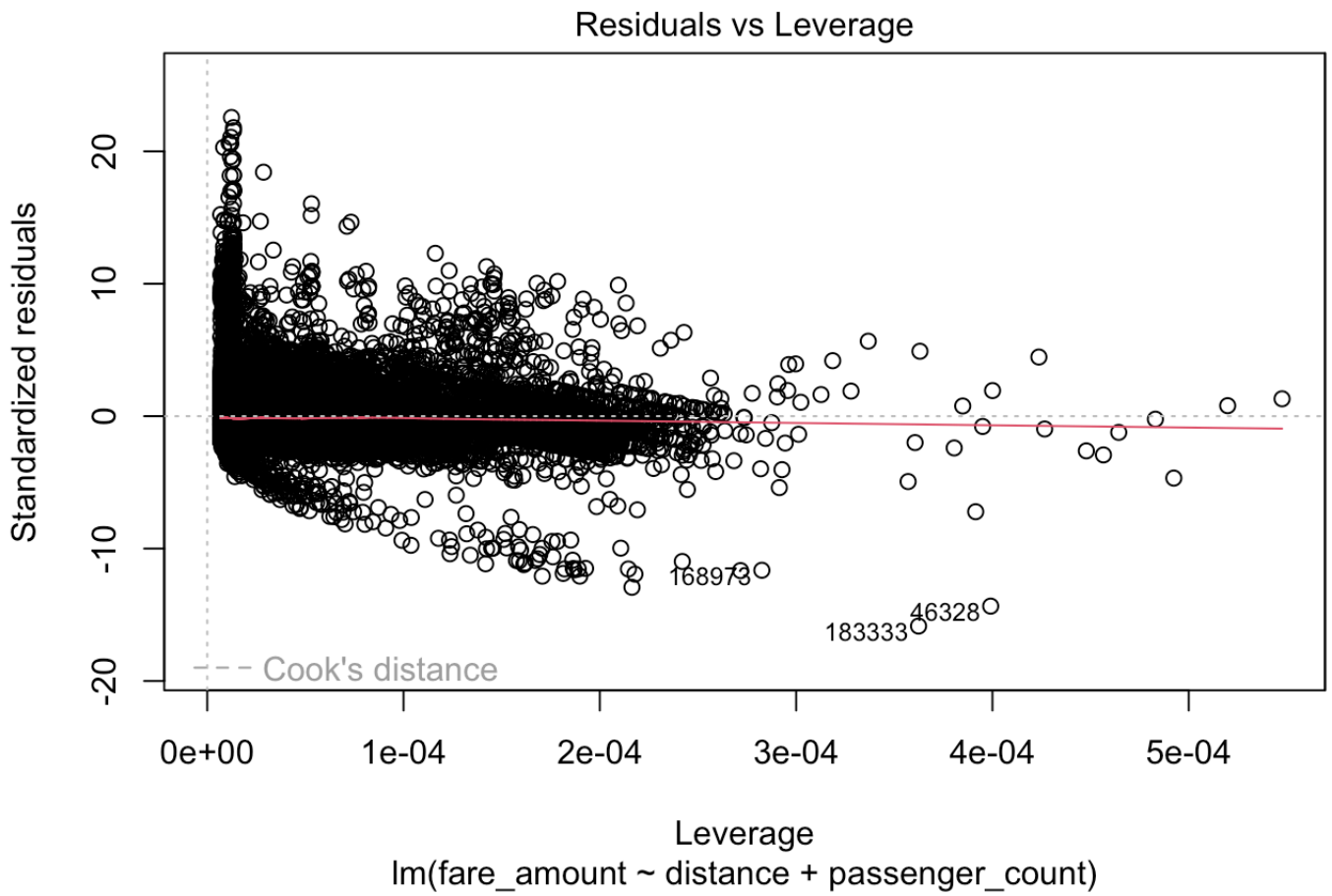
```
##
## Call:
## lm(formula = fare_amount ~ distance + passenger_count, data = dtrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -66.333  -1.655  -0.664   0.876  94.409
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.762e+00  1.978e-02 190.138  < 2e-16 ***
## distance        2.182e+02  2.781e-01 784.455  < 2e-16 ***
## passenger_count 3.757e-02  8.143e-03   4.613 3.96e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.186 on 154471 degrees of freedom
## Multiple R-squared:  0.7994, Adjusted R-squared:  0.7994
## F-statistic: 3.078e+05 on 2 and 154471 DF,  p-value: < 2.2e-16
```

```
plot(model3)
```


Residuals vs Fitted
Residuals
100
50
0
-50
197605
124820
20
40
60
80
Fitted values
lm(fare_amount ~ distance + passenger_count)


Normal Q-Q
Standardized residuals
20
10
0
-10
197605
-4
-2
0
2
4


Theoretical Quantiles
lm(fare_amount ~ distance + passenger_count)
Scale-Location
√|Standardized residuals|
197605
124820
0
1
2
3
4
20
40
60
80
Fitted values
lm(fare_amount ~ distance + passenger_count)



And we now have an R^2 value near 0.8. This indicates that a good linear trend is present at least to some degree (at least much more than was there previously). We will now summarize our findings below.

```
p3<-predict(model3,newdata=dtest)
mean((p3-dtest$fare_amount)^2)
```

```
## [1] 16.81776
```

```
s1<-summary(model)
s2<-summary(model2)
s3<-summary(model3)
mean(s1$residuals^2)
```

```
## [1] 87.31009
```

```
mean(s2$residuals^2)
```

```
## [1] 86.54743
```

```
mean(s3$residuals^2)
```

```
## [1] 17.51905
```

The above statistics reflect the error of the model (basically a sum of how far the model was off for each data point that has been adjusted somewhat). As we saw earlier, this statistic tells us very clearly that our third model is vastly superior to our earlier two models. This is the case because model3 has much less error and much more linear trending (due to MSE shown above and R^2 value explained further above).

My results were actually very good considering how the data was when I first started with it. The data had roughly 7000 useless rows out of 200000, and filtering those was quite difficult. The results are also very intuitive, no variables given had any surprise correlation with anything, and distance was a very good predictor of how much the fare was. The MSE for each model (test values and train values) were nearly identical, and the third model had roughly 5x less MSE than the others, and a reasonable R^2 value.

The residual graphs also reflect the story told by the rest of the data, and while interpreting them is not very intuitive, looking at the graphs for the last model over the other two shows a clear difference and is a good example of what the graphs should and should not look like.