Genscape Oil Project

Jacob Townson

June 16, 2018

Problem 1

Prompt:

An oil pipeline uses pump stations to push oil over large distances. Genscape monitors the power consumption of these pump stations in Megawatts and converts this power into the amount of oil flowing through a pipeline in barrels of oil per day. We have provided you with the power consumption at a pump station and the corresponding flow rates in the pipeline (note: The flow rates are considered truth data, while the Megawatts are the actual measurements taken by Genscape). Please attempt to model the flow rate as a function of the pump station power. Discuss whether your model (or models, if you chose to change the model during the time series) is/are a good fit and explain your methodology.

Find the average monthly value for your prediction and the 'Oil Flow' columns. Create a graph comparing the predicted and actual values using the monthly averages. Please make the chart clear as if it were being presented to a customer.

My Work:

To start, I have already read the required data into R, and named the data for this problem `prob1_data`. Table 1 shows a slight glimpse at what this data looks like. What we have here is the date that the data corresponds to, the number of barrels that flowed through that pump on a given day, and the pump station power in Megawatts. Before we begin our model making process, it may be helpful to split the data into test and training datasets. We can do this with the following simple bit of code:

```
set.seed(225566)
training.data1 = sample_frac(prob1_data, size = 2/3)
test.data1 = anti_join(prob1_data, training.data1, by = 'Date')
```

Here we are using the `dplyr` package to easily organize the test and training datasets. First we set the seed so that we get the same training data every time we run this code. The training data here is $\frac{2}{3}$ of the original data, while the test data is the leftover $\frac{1}{3}$. We don't use a validation set here because our goal is to simply assess how well the model and estimation method work.

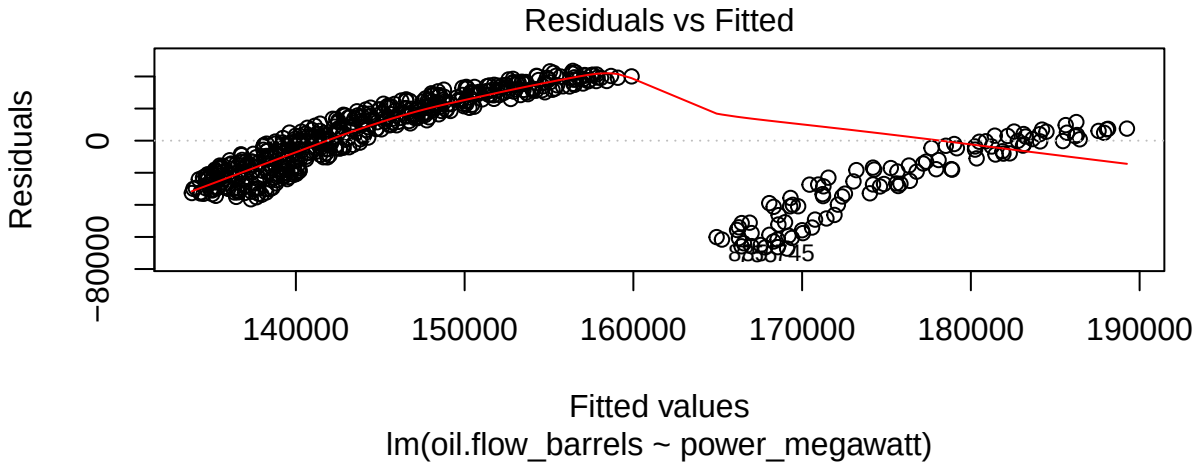
Now we can begin the creation of our model. Let's start with the easiest option and try a linear model.

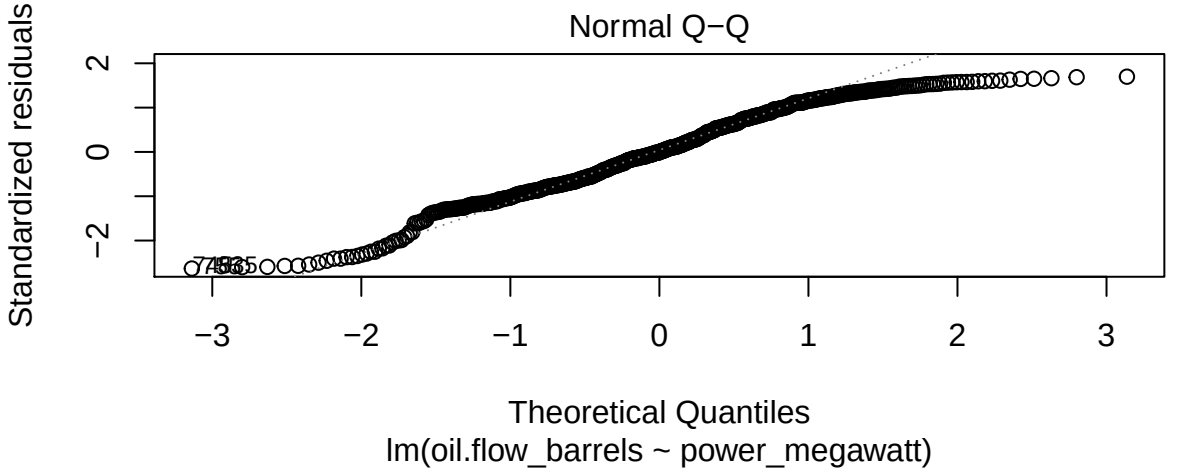
```
oil.lm = lm(oil.flow_barrels~power_megawatt, data = training.data1)
summary(oil.lm)
```

```
##
## Call:
## lm(formula = oil.flow_barrels ~ power_megawatt, data = training.data1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -67290 -19039   -178  21265  43494
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)    130070.7     1841.5   70.64   <2e-16 ***
## power_megawatt   3114.1      238.3   13.07   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25640 on 584 degrees of freedom
## Multiple R-squared:  0.2263, Adjusted R-squared:  0.2249
## F-statistic: 170.8 on 1 and 584 DF,  p-value: < 2.2e-16
```

Presented here is our summary of the linear model. Just by looking at this, we find that at a first glance this model works quite well! Notice our extremely small P values for the intercept and the `power_megawatt` variable. We also get a very small P-value for the F-statistic which is very promising. Just to make sure let's continue to check this model by looking at the residuals.





Here we can see that our residuals clearly show something is amiss. From our first residual plot, we can see that it almost seems as though our data is split into 2 parts. And from the Q-Q plot we see that we definitely have some outliers and maybe some noise. So maybe a simple linear model is not our best option.

To further our exploration here, let's look at some correlations in the data using the `pairs` function in R (Fig. 1). This plot shows some interesting facts about this data. First off, as probably expected, Date vs. the oil flow in barrels makes for a lot of noise. Then in the Date vs. the power produced in megawatts, we get what appears to be 2 blocks of noise. Notice these blocks are divided by the beginning of the year 2017. Then in the oil flow vs. power produced plots, we see that we get what look like 2 separate outcomes. Curiosity based on the date vs. power plots makes me wonder if this could possibly have something to do with the change in power produced beginning in 2017.

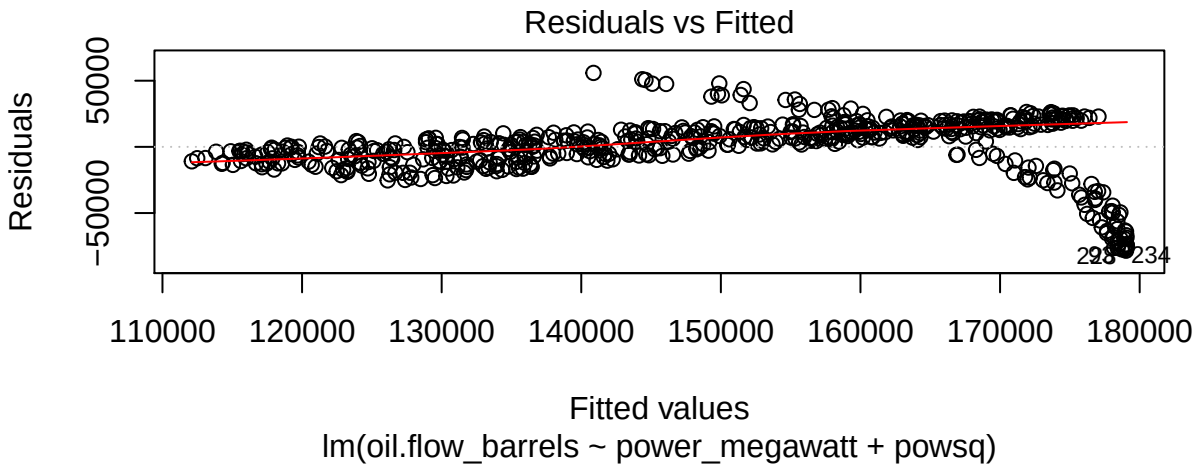
To see if the data from 2017 is causing troubles, let's remove it entirely and find what happens then (Fig. 2). Here we are getting somewhere. Notice the 2017 data must be the problem. As we noticed in our original residuals, the data is split, and now we can see how and where. However, we cannot and will not ignore this data from 2017. Instead, we will find a way to work with it.

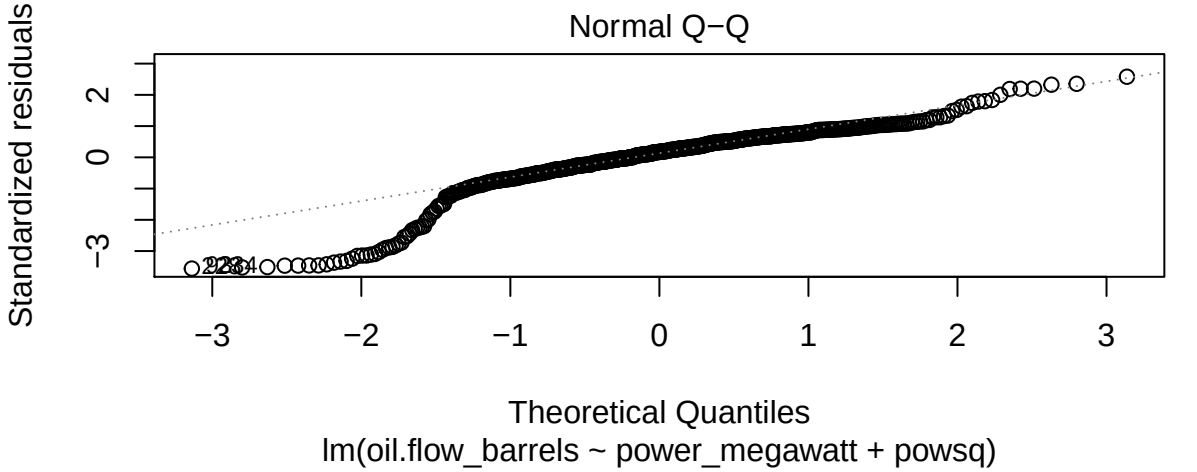
Before we attempt to use a new model, I would first like to see if this linear model could be improved by using quadratic variables. To do this, we will simply add the quadratic values for the power in megawatts to the original dataframes.

```
training.data1 = mutate(training.data1, powsq = power_megawatt^2)
test.data1 = mutate(test.data1, powsq = power_megawatt^2)
oil.lmq = lm(oil.flow_barrels~power_megawatt+powsq, data = training.data1)
summary(oil.lmq)
```

```
##
## Call:
## lm(formula = oil.flow_barrels ~ power_megawatt + powsq, data = training.data1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -78443  -8445   3852  14340  55889
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    95174.80    2917.09   32.63   <2e-16 ***
## power_megawatt 14791.60     844.42   17.52   <2e-16 ***
## powsq           -651.82      45.72  -14.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22100 on 583 degrees of freedom
## Multiple R-squared:  0.4263, Adjusted R-squared:  0.4243
## F-statistic: 216.6 on 2 and 583 DF,  p-value: < 2.2e-16
```

Here we see yet again that our P-values for the model look exceptionally nice and small. But before we jump the gun, let's check the residuals.





These residuals show a better outcome than our first model, but the Q-Q plot is still showing quite a few outliers, and we can see that our model still seems to be split in two. There is one more thing we can do to help the situation. We can make an indicator variable for any data entered in the year 2017. This indicator variable in our linear model can help distinguish the massive difference in the data depending on the year. To do this, we will go back to the handy `dplyr` package, then make a for loop to put in our indicators for the year 2017.

```
n = length(prob1_data$Date)
prob1_data = mutate(prob1_data, ind = rep(0,n))
for(i in 1:n){
  if(prob1_data$Date[i] >= '2017-01-01'){
    prob1_data$ind[i] = 1
}
}
```

Now we'll recreate our training and test datasets with these indices. We also add back in the quadratic variables, as it was clear that they will be necessary for the model.

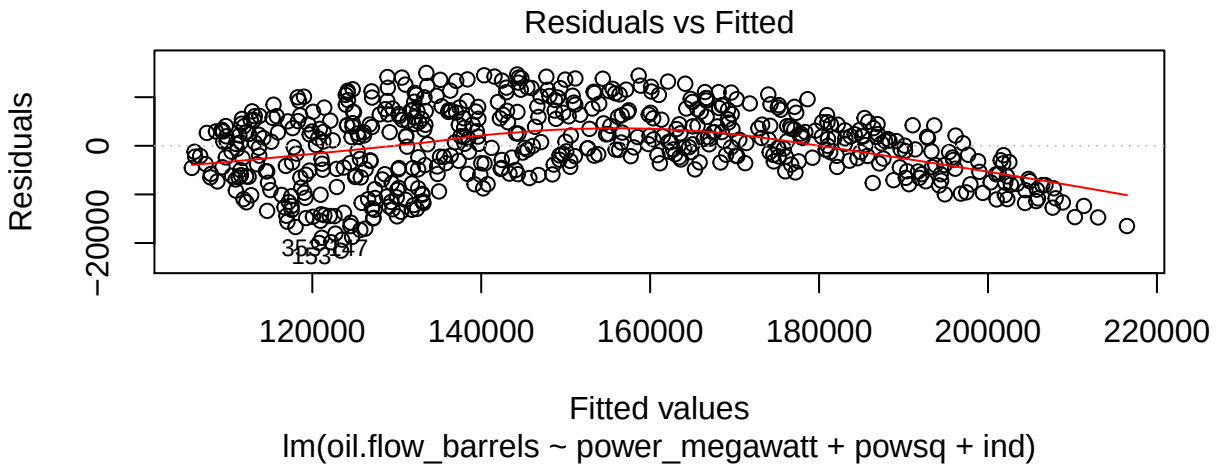
```r
set.seed(225566)
training.data1 = sample_frac(prob1_data, size = 2/3)
test.data1 = anti_join(prob1_data, training.data1, by = 'Date')
training.data1 = mutate(training.data1, powsq = power_megawatt^2)
test.data1 = mutate(test.data1, powsq = power_megawatt^2)
prob1_data.mod = mutate(prob1_data, powsq = power_megawatt^2)
```

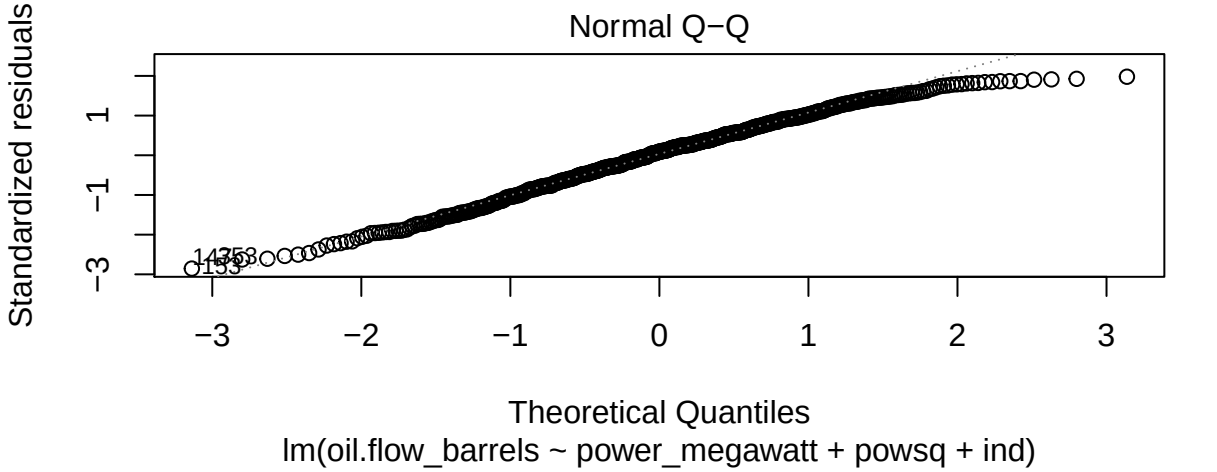
And now we can make our new model! As mentioned above, since the model with quadratic variables clearly worked better than the one without, we will expand on that one.

```r
oilyear.lm = lm(oil.flow_barrels~power_megawatt+powsq+ind, data = training.data1)
summary(oilyear.lm)
```

```
##
## Call:
## lm(formula = oil.flow_barrels ~ power_megawatt + powsq + ind,
##     data = training.data1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21574.2  -5079.4    693.4   5619.1  14991.4
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      89473.38    1010.46  88.547   <2e-16 ***
## power_megawatt   13483.98     292.10  46.162   <2e-16 ***
## powsq              -23.56      18.45  -1.277    0.202
## ind             -131417.59    2001.09 -65.673   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7626 on 582 degrees of freedom
## Multiple R-squared:  0.9318, Adjusted R-squared:  0.9314
## F-statistic:  2650 on 3 and 582 DF,  p-value: < 2.2e-16
```

Things are looking good so far. We have low P-values all around, although the P-value for the `powsq` variable is much higher than it was in our last model. Finally, let's check out the residuals.





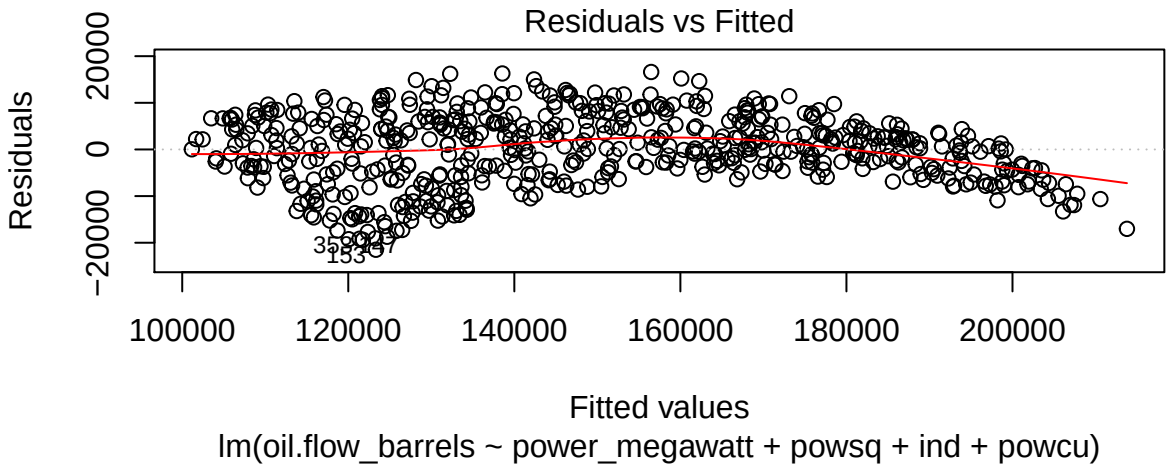
Now we are getting somewhere. The model finally is in one piece. The only downside to these residuals I would say is that we still have a slight curve in the first plot. To remedy this, let's add in one more variable, that being the power cubed. While we don't want to add too many variables in to keep the model as simple as possible, I do believe this will make it more accurate and worth the slight complexity. To save confusion on names, we will just go on and rename the model back to `oil.lm`.

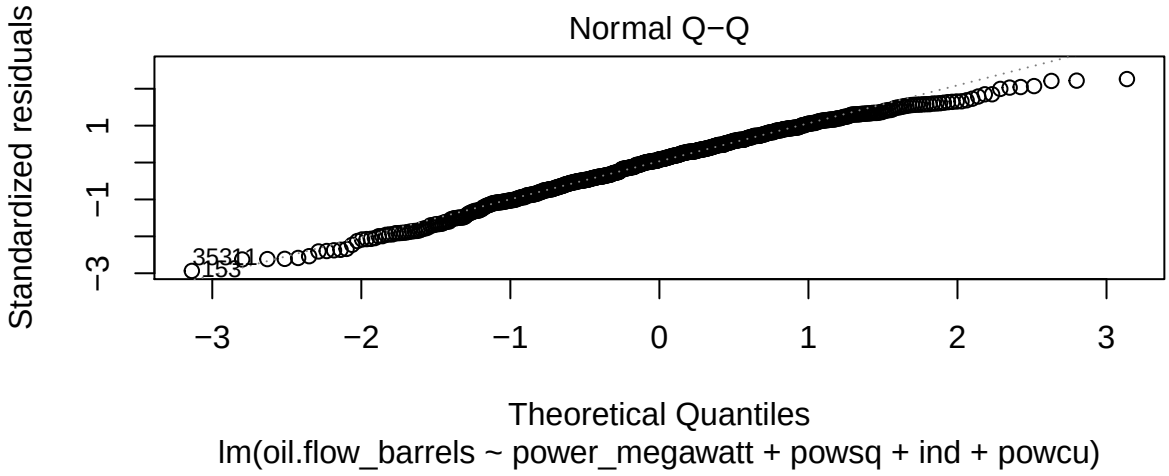
```
training.data1 = mutate(training.data1, powcu = power_megawatt^3)
test.data1 = mutate(test.data1, powcu = power_megawatt^3)
prob1_data.mod = mutate(prob1_data.mod, powcu = power_megawatt^3)
oil.lm = lm(oil.flow_barrels~power_megawatt+powsq+ind+powcu, data = training.data1)
summary(oil.lm)
```

```
##
## Call:
## lm(formula = oil.flow_barrels ~ power_megawatt + powsq + ind +
##     powcu, data = training.data1)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -21506.9  -4858.8    653.1   5370.5  16612.5
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     7.854e+04  2.012e+03  39.047  < 2e-16 ***
## power_megawatt  1.977e+04  1.050e+03  18.832  < 2e-16 ***
## powsq          -9.185e+02  1.450e+02  -6.335 4.77e-10 ***
## ind            -1.191e+05  2.769e+03 -43.013  < 2e-16 ***
## powcu           3.065e+01  4.927e+00   6.220 9.53e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7390 on 581 degrees of freedom
## Multiple R-squared:  0.936,  Adjusted R-squared:  0.9356
## F-statistic:  2126 on 4 and 581 DF,  p-value: < 2.2e-16
```

The P-values are even smaller on this model. Maybe we are on our way to making this model better. Let's check the residuals one last time.





These residuals do look better. It seems that the slight complexity was worth it.

As a final test for this model, we will now find the training and test error for the squared-error loss function $L(y, \hat{y}) = (y - \hat{y})^2$. This is standard practice, as we want to minimize this error. We will create an easy to use function in R to find this error.

```
L=function(y,y.hat){(y-y.hat)^2}
```

The `lm` function has a generic function `predict` which can be used to predict responses for new data based on a fitted model.

```
oilflow.hat = predict(oil.lm, test.data1)
```

Then the test error for this training data can be estimated from the test data using the following command.

```
obs.test.error=mean(L(test.data1$oil.flow_barrels,oilflow.hat))
obs.test.error
```

```
## [1] 50041530
```

Even though it may look large, this error of 50041530 is a decent error compared to the standard deviation of the oil flow. Thus we can conclude that this model works well. So now we move on to the final part of this problem. We must find the average monthly value for our prediction and compare it to the actual oil flow in the data given. To do this first, we must find the monthly averages of the actual data. Luckily for us, R has tools to get this done.

```
oil_ave = prob1_data %>% group_by(month=floor_date(Date, "month")) %>%
          summarize(oil.flow_barrels=mean(oil.flow_barrels))
```

The heading of this dataframe we have created is contained in Table 2 in the Appendix. As you can see, it contains the average oil flow in barrels per day for every month given in the supplied data. Now we just have to input the given power data into our model to get the day by day predictions from the model, then summarize it as we did in the code chunk above to give us our predicted data for the monthly average.

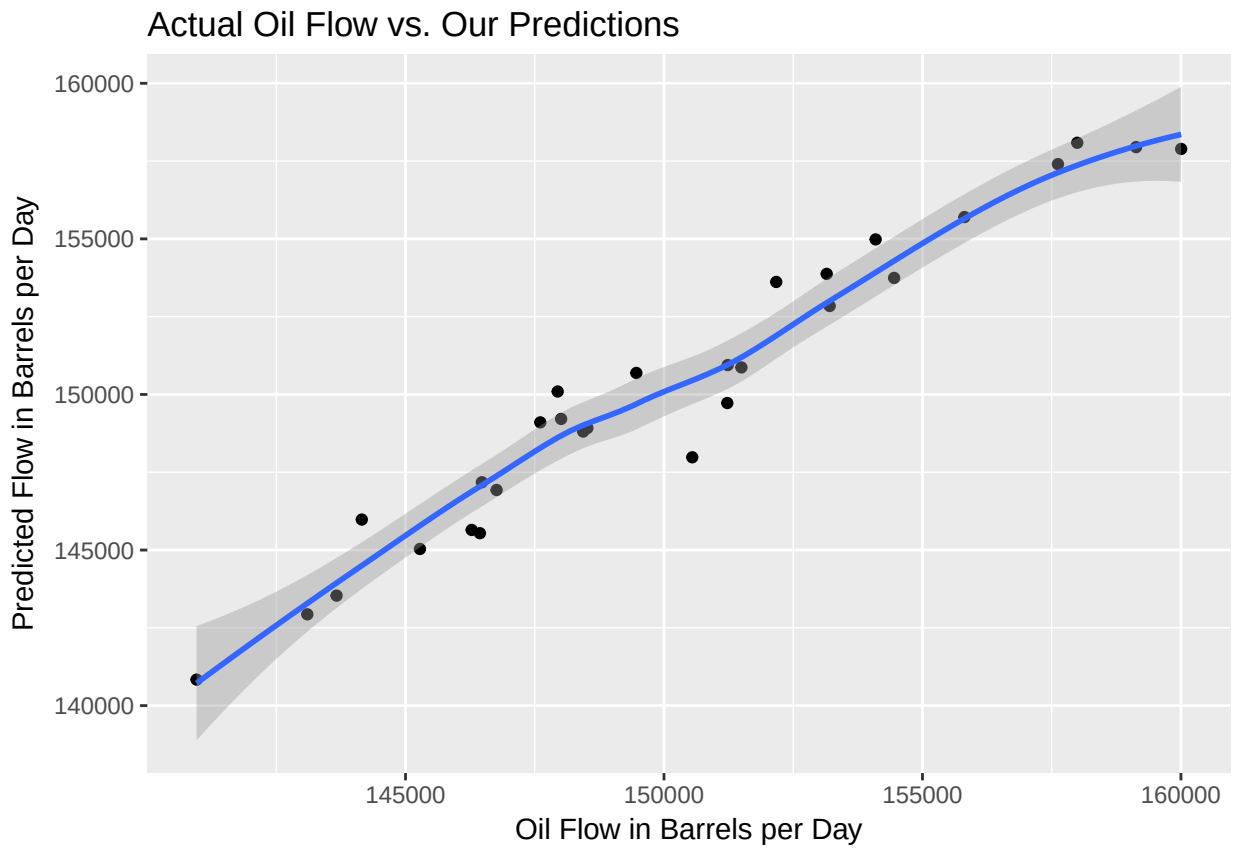
```
oil.pred = predict(oil.lm, prob1_data.mod)
oil.pred_data = data.frame(prob1_data$Date, oil.pred)
colnames(oil.pred_data) = c('Date', 'oil.predict')
oil.pred_ave = oil.pred_data %>% group_by(month=floor_date(Date, "month")) %>%
          summarize(oil.predict=mean(oil.predict))
```

Now we have our predicted monthly averages for the oil flow in barrels per day. Now we just have to compare this to the actual monthly averages for the oil flow. To do this, we first will organize the data into one clean dataframe.

```
plot_data = left_join(oil_ave,oil.pred_ave, by = 'month')
```

Finally we can make our plot. I will be using the package `ggplot2` to make this plot, as it has tools necessary to making the visual helpful and easy to understand.

```
## `geom_smooth()` using method = 'loess'
```



As you can see, this line looks very promising! While our predictions don't always line up perfectly, using the blue Loess curve, we can see that we average very close to the actual oil flow on a given day using this model. This plot is nice in its accuracy, but would also be easy to understand for consumers.

So in conclusion for this problem, we managed to come up with a model that predicts the actual outcomes well. And using the plot shown above, we can actually show others who may not know as much about the background material that this model does indeed work in most cases. My only problem with this model is that when we get into extremes, for example very high and very low values, it becomes less accurate. This was also pointed out in the Q-Q plot for the model. However, this is fairly normal for models such as this.

Problem 2

Prompt:

Cushing, Oklahoma is a large oil storage field that is critical to understanding oil supply and demand in the U.S. Cushing is connected to many large pipelines. Genscape wants you to research several pipelines to

better understand the pipeline's capacity, beginning and ending locations, and the operator/owners of the pipeline. Please create a table or list with this information for each pipeline provided.

Pipelines to research: Seaway (legacy), Dakota Access, Pony Express, White Cliffs, TransCanada Gulf Coast (aka MarketLink)

Genscape has provided sample data for each of the above pipeline's flow rates in barrels per day. We have also provided storage volumes at Cushing in Barrels. Using what you researched above, create a model using the pipeline data provided to predict storage changes at Cushing. Please note that a perfect model is not possible due to noise in the data. Please document the results of your model and explain its strengths and weaknesses.

West Texas Intermediate (WTI) price has a relationship with oil stored at Cushing (Cushing is the delivery point for the WTI NYMEX contract). WTI closing prices have been provided with their corresponding storage volumes. Please discuss any correlation you see, and any economic justification for why that relationship might exist.

My Model

I have created a table in Excel that I read into R which is presented in Table 3. This table completes the first part of the problem, finding all of the required data and information for each pipeline. I found this information by doing some quick research on the internet. I hope that everything is correct, some information was more difficult to find than others just from the way some companies had their information structured and publicly given out.

To begin the main portion of the problem, I have already created a dataframe in R containing all of the information given from Genscape for each pipeline in Cushing. The head of this dataframe (labelled in my work as `prob2_data` or `D`) is in Table 4.

Our goal here is to use this data to create a model using the pipeline data to predict storage changes at Cushing. The information found in Table 3 could help check our model as well, since we now know the maximum amounts of oil that can be pumped in barrels per day, as well as whether or not that oil is being pumped in or out of Cushing.

To start this time, let's check out any correlations we can find using the `pairs` function to give us a visual (Fig. 3). As mentioned in the prompt for this problem, there is indeed quite a bit of noise here, but it almost seems as if we can see some correlations happening. Specifically we see some interesting correlations between the barrels of oil in Cushing vs. the date, and some interesting correlations between the WTI closing prices vs. the date and the barrels at Cushing. We can play with this more later, but let's first try out some models.

Before we begin, let's divide up our data into test and training sets for this problem. For our test and training sets, we will remove the first day as we have NA values for all of the pipelines. We will also create a dataframe `D` to basically rename `prob2_data` in order to save time typing.

```
n = length(prob2_data$Date)
temp = prob2_data[2:n,]
set.seed(225566)
training.data2 = sample_frac(temp, size = 2/3)
test.data2 = anti_join(temp, training.data2, by = 'Date')
D = prob2_data
```

As before in the first problem, it is good practice to start simple, so let's try a linear regression model and see how things work out.

```
cushingoil = lm(Cushing.Storage..Barrels.~ Seaway.Pipeline..Barrels.per.day.
                + Pony.Express.Pipeline..Barrels.per.day.
                + Dakota.Access.Pipeline..Barrels.Per.Day.
                + White.Cliffs.Pipeline..Barrels.Per.Day.
```

```
                + TransCanada.Gulf.Coast.Pipeline..Barrels.Per.day.,
                data = training.data2)
summary(cushingoil)
```

```
##
## Call:
## lm(formula = Cushing.Storage..Barrels. ~ Seaway.Pipeline..Barrels.per.day. +
##     Pony.Express.Pipeline..Barrels.per.day. + Dakota.Access.Pipeline..Barrels.Per.Day. +
##     White.Cliffs.Pipeline..Barrels.Per.Day. + TransCanada.Gulf.Coast.Pipeline..Barrels.Per.day.,
##     data = training.data2)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -21434991  -6585233   -886205   8493591  16769686
##
## Coefficients:
##                                                  Estimate Std. Error
## (Intercept)                                     5.996e+07  2.467e+06
## Seaway.Pipeline..Barrels.per.day.               -1.024e+01  3.761e+00
## Pony.Express.Pipeline..Barrels.per.day.          1.780e+01  3.179e+00
## Dakota.Access.Pipeline..Barrels.Per.Day.         9.865e-01  2.683e+00
## White.Cliffs.Pipeline..Barrels.Per.Day.         -5.731e+01  1.605e+01
## TransCanada.Gulf.Coast.Pipeline..Barrels.Per.day.  1.543e+01  3.377e+00
##                                                 t value Pr(>|t|)
## (Intercept)                                      24.309  < 2e-16 ***
## Seaway.Pipeline..Barrels.per.day.                -2.723  0.00663 **
## Pony.Express.Pipeline..Barrels.per.day.           5.599 3.07e-08 ***
## Dakota.Access.Pipeline..Barrels.Per.Day.          0.368  0.71326
## White.Cliffs.Pipeline..Barrels.Per.Day.          -3.571  0.00038 ***
## TransCanada.Gulf.Coast.Pipeline..Barrels.Per.day.   4.568 5.79e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9360000 on 721 degrees of freedom
## Multiple R-squared:  0.08577,    Adjusted R-squared:  0.07943
## F-statistic: 13.53 on 5 and 721 DF,  p-value: 1.242e-12
```

The first strange thing we can note about this model is that our variables don't quite do what we would expect them to. Recall the starting and ending points for each of these pipelines. This information tells us whether or not the pipes are pumping oil in or out of Cushing. But if we notice, this linear model doesn't follow that like we would expect. We would expect that if a pipe pumps oil out of Cushing, then the estimate for the coefficient would be negative, indicating that the oil is leaving. We would expect the opposite for oil being pumped in. But notice, specifically for the White Cliffs pipeline and the TransCanada pipeline, the values are the opposite of what we would expect.

On top of this fact, we must think about the fact that the Dakota Access pipeline (we will call it DAP from here out) according to our research doesn't even go to the Cushing storage facility. And the P-value for DAP coefficient estimate is relatively high compared to all of the other coefficients. At first glance this would lead us to believe that the DAP is an unecessary variable and we should remove it. Before we jump into this decision, I would like to explore this model further.

First off, for the rest of this problem, we won't use separate training and test datasets. Cutting the data results in losing too much information in the model we create. For example, up until 05/16/2017, the DAP wasn't even pumping oil at all. After some research, we find that this is because it is a relatively new pipeline. In order to take this into account, we will not split the data as we did above.

The first thing I would like to do in this model is to test as to whether or not there is any lag in the data. By this I mean I would like to find out if pumping oil from any pipeline that pumps in or out of Cushing has lag compared to when the data says it was pumped to when we see the oil show up in the Cushing sotrage variable. If the lag is small enough, we will simply ignore it and move on. First off, let's make a new dataframe where things are a little easier to observe.

```
D2=D[-1,]
n=nrow(D)
D2[,2]=D[-1,2]-D[-n,2]
```

What we have done here is make a new dataframe `D2` that no longer shows the total amount of oil on any given day, but instead shows the change in oil stored for each day. A glimpse of this dataframe is shown in Table 5 in the Appendix. This will help make visualizing this data a little simpler.

To find if there is a significant lag, we will find a crucial point in the oil stored, and see if the oil being pumped in and out of Cushing reflects the change. To do this, let's first find a crucial point in the data. Let's plot the change in oil each day vs. the date (see Figure 4). The crucial point we will be looking at is between the two red lines on the plot. Clearly something happened here to make the amount of oil in Cushing go down. So using this, and just by looking at the data, we can see if there is indeed a significant lag. Note, the red lines mark the days 367 and 391 respectively. So if we just look at the data entries for these days, we can see that indeed oil was being pumped out, but not much was being pumped in. In fact, the exact days that we see these changes happen in the figure lines up perfectly with the days in the data. All you need to do to see this is scroll down to the 367th entry in the excel file supplied to us to confirm this. Thus we can conclude that if there is any lag, it is not significant enough to consider in our model.

Now, the problem reads: "create a model using the pipeline data provided to predict storage changes at Cushing". We will do this by actually making a model that predicts the changes in barrels of oil stored at Cushing, not just the amount being stored as we did in our first model we tried. To do this, we will go back to our `D2` dataframe.

Before we actually create the model though, we must first find out if the DAP is significant to the change in oil at Cushing. To do this, let's look at figure 5. At first glance this plot may seem daunting, but I promise it will make since. The black line here represents the amount of oil stored on each day in Cushing. The red line represents the oil being added in by White Cliffs, the green line is the oil added by Pony Express, and finally the blue line represents DAP. The way we managed to get all of these lines onto one plot was to normalize each of the variables, ie, subtract the mean, then divide the difference by the standard deviation of each. When we see the colored lines rise, that represents times that these pipelines were pumping oil into the Cushing storage facility. As we have noted, DAP doesn't start until 2017, where we see the blue lines beginning to be active. Take note that at this point, oil is still being taken out of Cushing at a fairly consistent rate. Also notice that a little before the 800th day in our data, Cushing is at an all time low in the amount of oil being stored. And even though there are no drastic changes in the oil going in or oil coming out, the amount of oil in Cushing begins to rise once the DAP began pumping. Thus I believe that we can assume that somehow, the DAP is affecting the amount of oil being stored in Cushing, seemingly by adding oil to the storage.

Thus we cannot rule the DAP out of our model, meaning that we must include everything we have. However, this does make it difficult to find a single model that works the best that is also all encompassing. To wrap things up, I will present 3 models, each with their own benefits and weaknesses. The first of which will be centered around the data before DAP was added into the mix.

```
bDAP=lm(D2[1:865,2]~D2[1:865,3]+D2[1:865,4]+D2[1:865,6]+D2[1:865,7])
summary(bDAP)
```

```
##
## Call:
## lm(formula = D2[1:865, 2] ~ D2[1:865, 3] + D2[1:865, 4] + D2[1:865,
##     6] + D2[1:865, 7])
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -319579  -52965    4419   52132  366861
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.837e+05  1.815e+04  15.636   <2e-16 ***
## D2[1:865, 3] -8.099e-01  2.858e-02 -28.334   <2e-16 ***
## D2[1:865, 4]  9.874e-01  2.175e-02  45.391   <2e-16 ***
## D2[1:865, 6]  1.009e+00  1.143e-01   8.825   <2e-16 ***
## D2[1:865, 7] -1.005e+00  2.931e-02 -34.272   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 68240 on 860 degrees of freedom
## Multiple R-squared:  0.8284, Adjusted R-squared:  0.8276
## F-statistic:  1038 on 4 and 860 DF,  p-value: < 2.2e-16
```

As we can see here, this model is in line much more with what we expected out of a model for this situation. All of our P-values are low, and the estimates for the coefficients are positive and negative where we expect them to be. The residuals are plotted as well in the appendix in Figure 6. While these residuals are not extremely impressive, the are not bad either. They at least show that the coefficients cannot be rejected and that we are close to normality in the Q-Q plot. Next up we will look at a model for after DAP came into the situation.

```
aDAP=lm(D2[866:n,2]~D2[866:n,3]+D2[866:n,4]+D2[866:n,5]+D2[866:n,6]+D2[866:n,7])
summary(aDAP)
```

```
##
## Call:
## lm(formula = D2[866:n, 2] ~ D2[866:n, 3] + D2[866:n, 4] + D2[866:n,
##     5] + D2[866:n, 6] + D2[866:n, 7])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -293779  -46287    4150   54618  201143
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.386e+05  3.328e+04  13.178  < 2e-16 ***
## D2[866:n, 3] -6.368e-01  4.476e-02 -14.228  < 2e-16 ***
## D2[866:n, 4]  7.725e-01  5.215e-02  14.812  < 2e-16 ***
## D2[866:n, 5] -8.130e-02  3.035e-02  -2.679  0.00795 **
## D2[866:n, 6] -1.481e-01  2.376e-01  -0.623  0.53374
## D2[866:n, 7] -1.069e+00  3.356e-02 -31.837  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 85990 on 219 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.8585, Adjusted R-squared:  0.8552
## F-statistic: 265.7 on 5 and 219 DF,  p-value: < 2.2e-16
```

Here things become a little less accurate. This could be for multiple reasons, but my educated guess is that is has something to do with the small amount of data we have after the DAP is created. On top of this, we still

technically don't know the capacity in which the DAP affects the oil storage at Cushing, we only know that it has some effect. We can notice in the summary that P-values aren't what we would expect. For example, number 6 in the variables is the White Cliffs pipeline. Before this, we have never questioned the impact of this pipeline on our model, but now we are getting a very high P-value, one that most (if not all) people would say means that we should remove the variable entirely. Considering all of these factors, I would not use this model. In case the reader would like to see them, the residuals are plotted in Figure 7.

The final model we will discuss is the all encompassing model. This model will use all of the data at our disposal.

```
cushing = lm(D2$Cushing.Storage..Barrels.~D2$Seaway.Pipeline..Barrels.per.day.
          +D2$Pony.Express.Pipeline..Barrels.per.day.
          +D2$Dakota.Access.Pipeline..Barrels.Per.Day.
          +D2$White.Cliffs.Pipeline..Barrels.Per.Day.
          +D2$TransCanada.Gulf.Coast.Pipeline..Barrels.Per.day.)
summary(cushing)
```

```
##
## Call:
## lm(formula = D2$Cushing.Storage..Barrels. ~ D2$Seaway.Pipeline..Barrels.per.day. +
##     D2$Pony.Express.Pipeline..Barrels.per.day. + D2$Dakota.Access.Pipeline..Barrels.Per.Day. +
##     D2$White.Cliffs.Pipeline..Barrels.Per.Day. + D2$TransCanada.Gulf.Coast.Pipeline..Barrels.Per.day
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -336987  -51673    2464   54026  363053
##
## Coefficients:
##                                                    Estimate Std. Error
## (Intercept)                                       3.238e+05  1.570e+04
## D2$Seaway.Pipeline..Barrels.per.day.             -7.458e-01  2.328e-02
## D2$Pony.Express.Pipeline..Barrels.per.day.        9.558e-01  2.053e-02
## D2$Dakota.Access.Pipeline..Barrels.Per.Day.      -4.908e-02  1.724e-02
## D2$White.Cliffs.Pipeline..Barrels.Per.Day.        6.813e-01  1.053e-01
## D2$TransCanada.Gulf.Coast.Pipeline..Barrels.Per.day. -1.051e+00  2.064e-02
##                                                   t value Pr(>|t|)
## (Intercept)                                        20.625  < 2e-16 ***
## D2$Seaway.Pipeline..Barrels.per.day.              -32.032  < 2e-16 ***
## D2$Pony.Express.Pipeline..Barrels.per.day.         46.546  < 2e-16 ***
## D2$Dakota.Access.Pipeline..Barrels.Per.Day.        -2.846  0.00451 **
## D2$White.Cliffs.Pipeline..Barrels.Per.Day.          6.467 1.51e-10 ***
## D2$TransCanada.Gulf.Coast.Pipeline..Barrels.Per.day. -50.906  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 73990 on 1084 degrees of freedom
## Multiple R-squared:  0.8398, Adjusted R-squared:  0.839
## F-statistic:  1136 on 5 and 1084 DF,  p-value: < 2.2e-16
```

This model looks much better in terms of the P-values. The DAP is still high compared to others, however it is low enough that I don't think it should be rejected. What is odd that the DAP got a negative coefficient, although it is very small. The main strength of this model is that it is all inclusive, and still relatively accurate even so. In Figure 8 you can see the residuals. Notice that excluding some outliers in the Q-Q plot, this is the most accurate model we've had thus far. The weakness of this model though is that I feel that if new data were to be added, it would not hold up. Simply put it's because more data is coming in every day that actually accounts for the DAP better than the first 865 entries of the data we were given does. So even

though this may be accurate in the short term, it would definitely need to be updated over time. Even so, I would argue that this is the best model yet.

Bonus Model

I didn't want to leave the reader thinking that all I was good for was a linear regression model, so I wanted to include this as well. Below is the method using the `caret` package to make a random forest model. This model uses all of the pipeline data and uses machine learning techniques to decide what would make an accurate model.

```
## random forest using caret
set.seed(213874)
ctrl=trainControl(method="boot632")
rf.model=train(Cushing.Storage..Barrels.~ Seaway.Pipeline..Barrels.per.day. +
               Pony.Express.Pipeline..Barrels.per.day.
               + Dakota.Access.Pipeline..Barrels.Per.Day.
               + White.Cliffs.Pipeline..Barrels.Per.Day.
               + TransCanada.Gulf.Coast.Pipeline..Barrels.Per.day.,
               data=training.data2, method="rf",ntree=5000, trControl=ctrl,
               tuneGrid=data.frame(mtry = 1:8))
rf.model
```

```
## Random Forest
##
## 727 samples
##   5 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 727, 727, 727, 727, 727, 727, ...
## Resampling results across tuning parameters:
##
##   mtry  RMSE     Rsquared   MAE
##   1     9178146  0.1507753  7783878
##   2     8980361  0.1611287  7355255
##   3     8961815  0.1628147  7242476
##   4     8962159  0.1632021  7219487
##   5     8963765  0.1632438  7208493
##   6     8963693  0.1632420  7208842
##   7     8963642  0.1632680  7208205
##   8     8963775  0.1632675  7208596
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 3.
```

```
cush.fit=predict(rf.model,training.data2)
mean(L(training.data2$Cushing.Storage..Barrels.,cush.fit))
```

```
## [1] 7.745711e+13
```

```
cush.hat=predict(rf.model,test.data2)
mean(L(test.data2$Cushing.Storage..Barrels.,cush.hat))
```

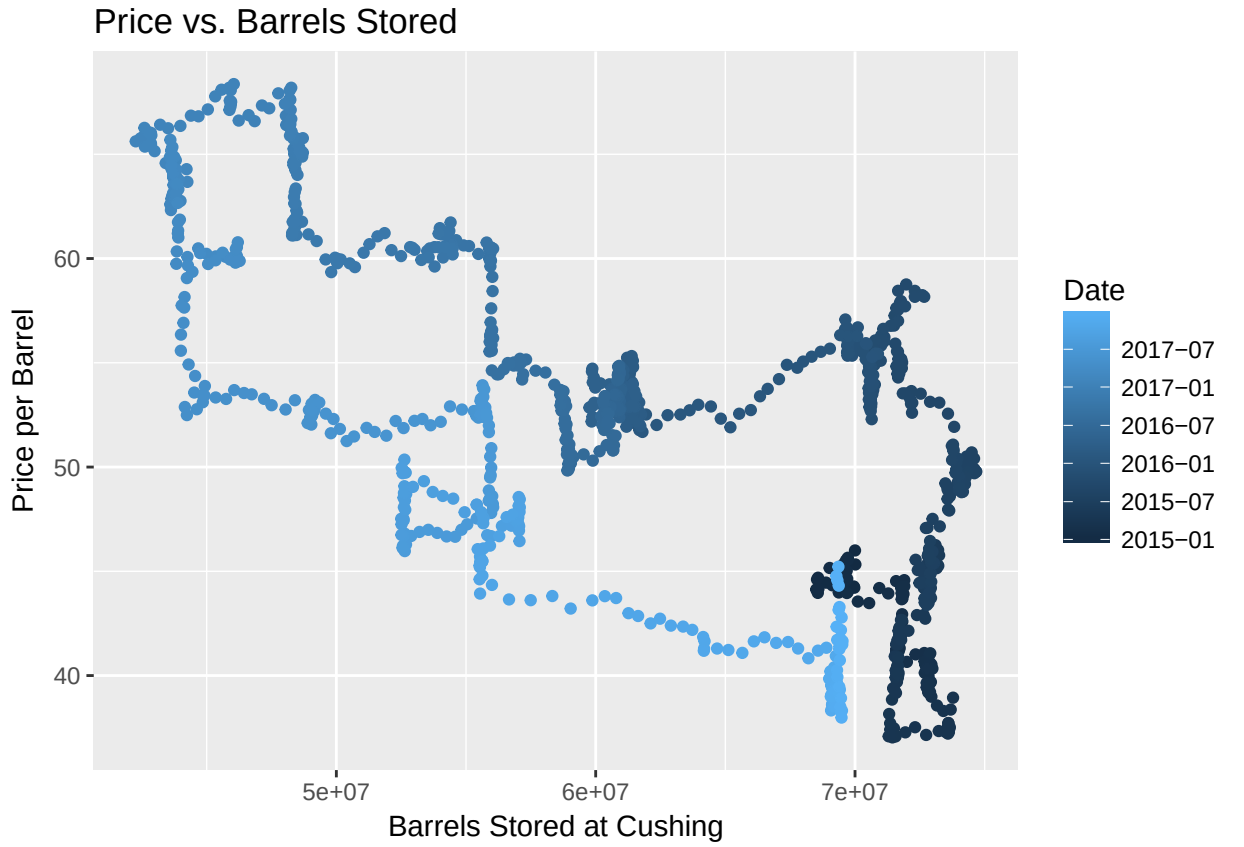
```
## [1] 7.22922e+13
```

Before finishing this project, I tried quite a few different types of models for this problem. This one gave me

the lowest training and test errors (7.745711×10^{13} and 7.22922×10^{13} respectively), which is why I have chosen to include it. Both are actually quite small compared to the standard deviation. The downside of this model though is that it's not exactly easy to use, and consumers may not like the fact that it isn't easy to understand. But if we could make a way to hide the unnecessary details, say in perhaps a shiny app or something of the sort, this model could be used to it's fullest capacity without bogging down consumers in its complexity. It is also unfortunate that I couldn't find the test error for my final linear model I created to compare it to this one. If the data was expanded and more was added though, this could be done easily.

Price Correlation Discussion

Here we will discuss the last part of this problem; the correlation we found between the price of oil, the barrels being stored at Cushing, and the date. To do this, we will look at the below plot.



Here we can see the correlation very easily. First note that the darker colored points are the earlier dates, and the colors get lighter as time goes on. And, as we would expect, the more barrels stored at Cushing, the less the price of oil, and vice versa. This is simple supply and demand logic. What's interesting is how seemingly in the more recent dates, the price is even lower per barrel related to how many barrels are stored at Cushing. This could be for many reasons, maybe the addition of the DAP, but without more information, it is hard to say. Nonetheless, it is very interesting to note.

Appendix

Problem 1

Figure 1

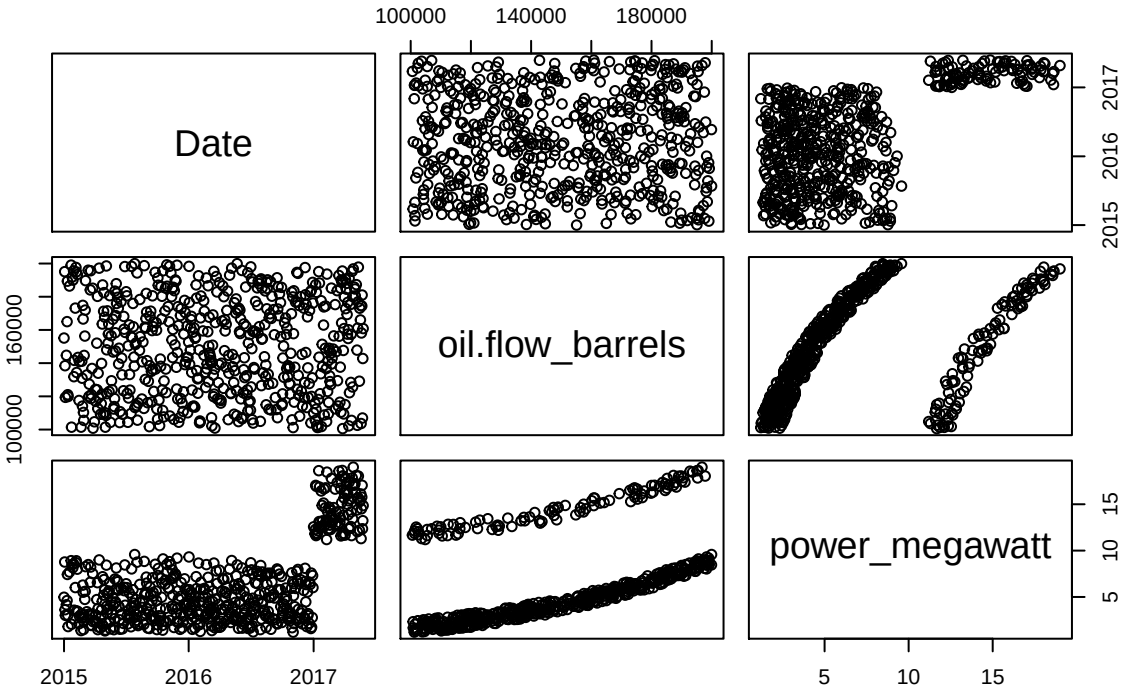


Table 1: Problem 1 Data Glimpse

Date	oil.flow_barrels	power_megawatt
2015-01-01	155117	4.969950
2015-01-02	155002	5.228080
2015-01-03	195091	8.769649
2015-01-04	138447	3.624437
2015-01-05	119406	3.021225
2015-01-06	173907	6.359465

Figure 2

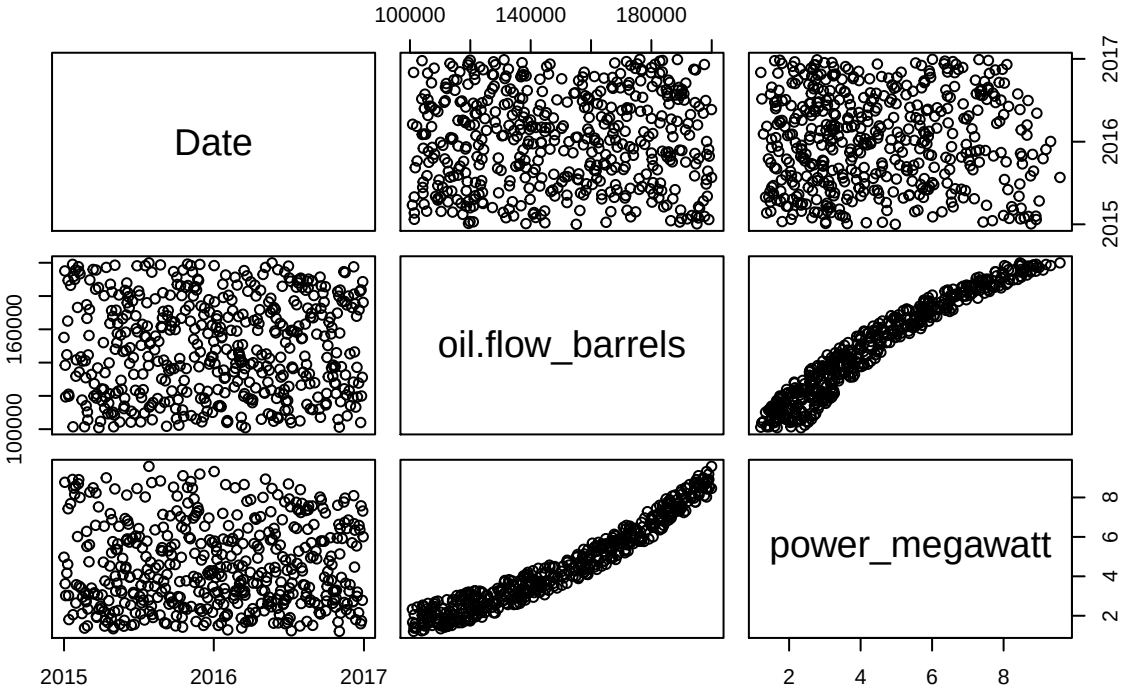


Table 2: Average Oil Flow per Month

month	oil.flow_barrels
2015-01-01	154094.3
2015-02-01	146279.9
2015-03-01	148517.9
2015-04-01	146762.1
2015-05-01	148436.6
2015-06-01	151498.0

Problem 2

Table 3: Cushing, OK Pipeline Data

Pipeline	Pipeline.Capacity..BPD.	Beginning.Location	Ending.Location	Owner	Operator
Seaway (legacy)	400000	Cushing, Oklahoma	Houston Texas	Endbridge Inc.	Enterprise Products Partners L.P.
Dakota Access	570000	Brakken, North Dakota	Pakota, Illinois	Energy Transfer Crude Oil Company, LLC	Dakota Access, LLC
Pony Express	230000	Guernsey, Wyonming	Cushing, Oklahoma	Tallgrass Energy Partners	Tallgrass Energy Partners
White Cliffs	215000	Denver Basin, Colorado	Cushing, Oklahoma	SemGroup	Rose Rock Midstream
TransCanada Gulf Coast (MarketLink)	700000	Cushing, Oklahoma	Port Arthur, Texas; Houston, Texas	TransCanada	Marketlink LLC

Figure 3

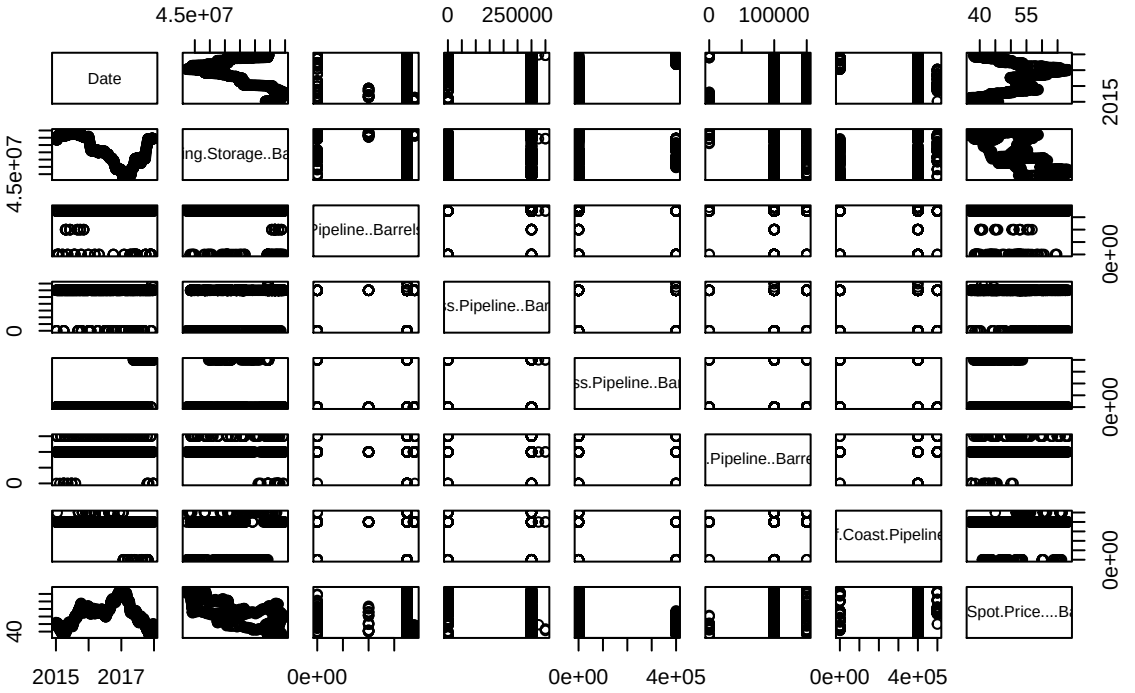


Table 4: Problem 2 Data Glimpse

Date	Cushing.Storage..Barrels.	Seaway.Pipeline..Barrels.per.day.	Pony.Express.Pipeline..Barrels.per.day.	Dakota.Access.Pipeline..Barrels.Per.Day.	White.Cliffs.Pipeline..Barrels.Per.Day.	TransCanada.Gulf.Coast.Pipeline..Barrels.Per.day.	WTI.Spot.Price....Barrel.
2015-01-01	70000000	NA	NA	NA	NA	NA	46.00000
2015-01-02	70013944	350000	3e+05	0	1e+05	4e+05	45.32784
2015-01-03	69952898	350000	3e+05	0	1e+05	4e+05	44.27039
2015-01-04	69971671	350000	3e+05	0	1e+05	4e+05	44.21480
2015-01-05	69724741	350000	0e+00	0	1e+05	4e+05	44.02187
2015-01-06	69374326	350000	0e+00	0	0e+00	4e+05	43.98455

Table 5: Modified Data with Differences

	Date	Cushing.Storage..Barrels.	Seaway.Pipeline..Barrels.per.day.	Pony.Express.Pipeline..Barrels.per.day.	Dakota.Access.Pipeline..Barrels.Per.Day.	White.Cliffs.Pipeline..Barrels.Per.Day.	TransCanada.Gulf.Coast.Pipeline..Barrels.Per.day.	WTI.Spot.Price....Barrel.
2	2015-01-02	13944.34	350000	3e+05	0	1e+05	4e+05	45.32784
3	2015-01-03	-61046.23	350000	3e+05	0	1e+05	4e+05	44.27039
4	2015-01-04	18772.88	350000	3e+05	0	1e+05	4e+05	44.21480
5	2015-01-05	-246929.50	350000	0e+00	0	1e+05	4e+05	44.02187
6	2015-01-06	-350415.25	350000	0e+00	0	0e+00	4e+05	43.98455
7	2015-01-07	-247730.88	350000	0e+00	0	1e+05	4e+05	44.32990

Figure 4: Change in Oil Stored Each Day

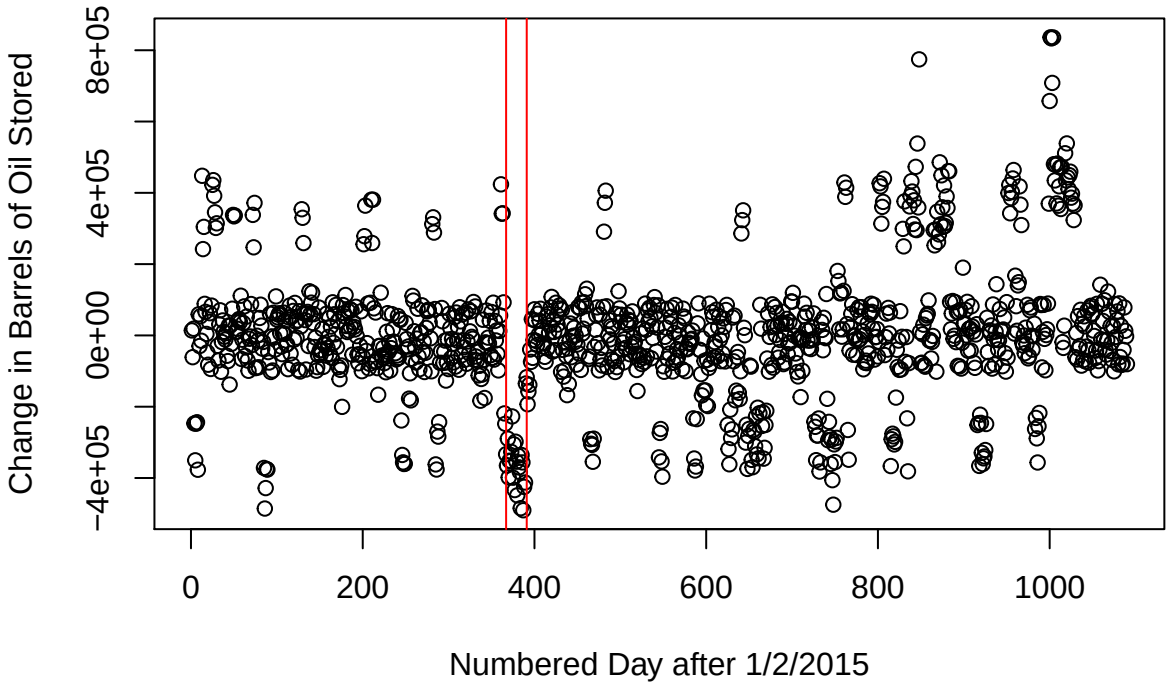


Figure 5: Change in Oil Amount at Cushing

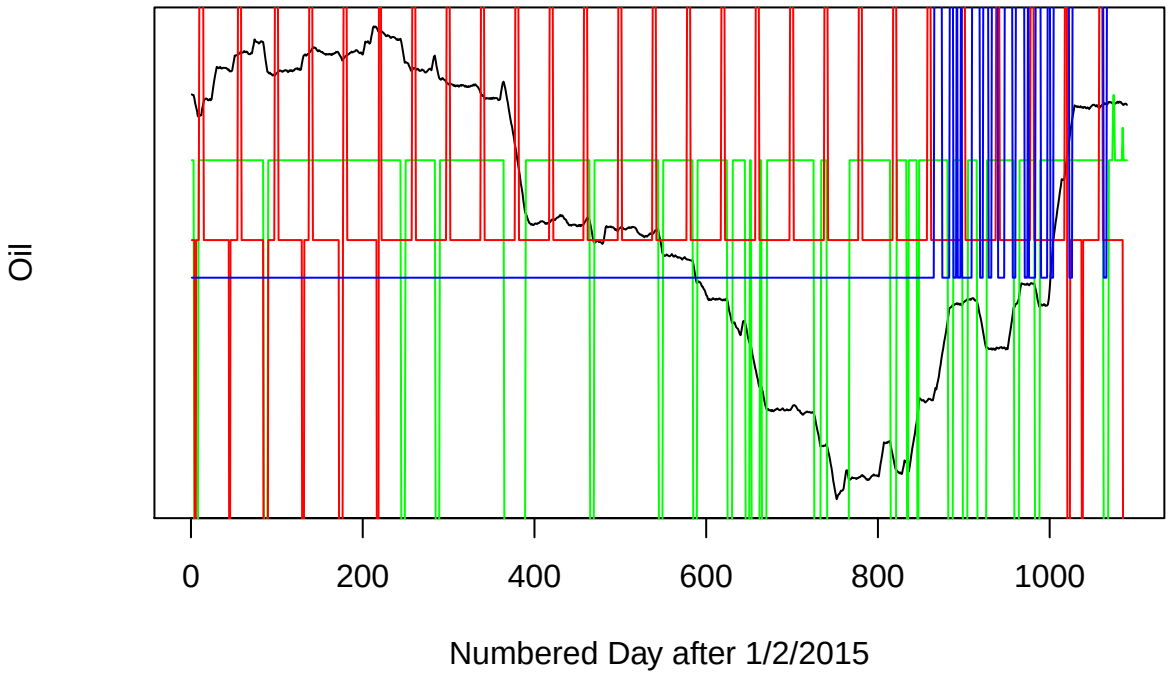


Figure 6

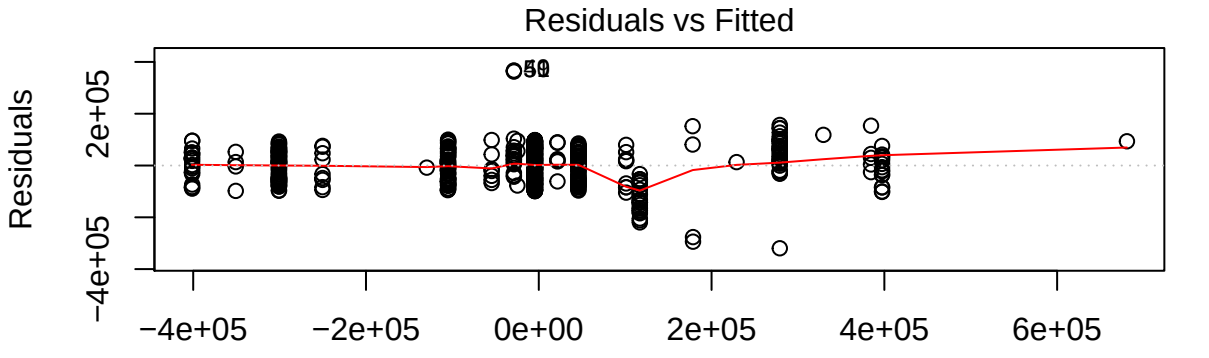


Figure 6

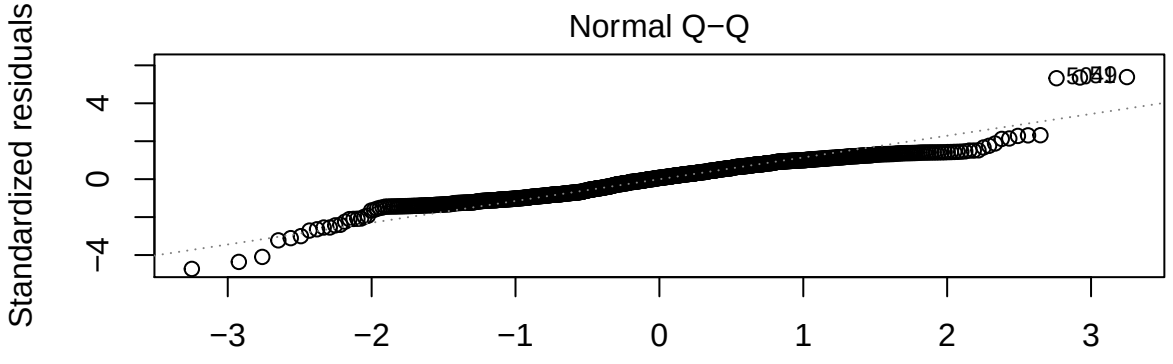


Figure 7

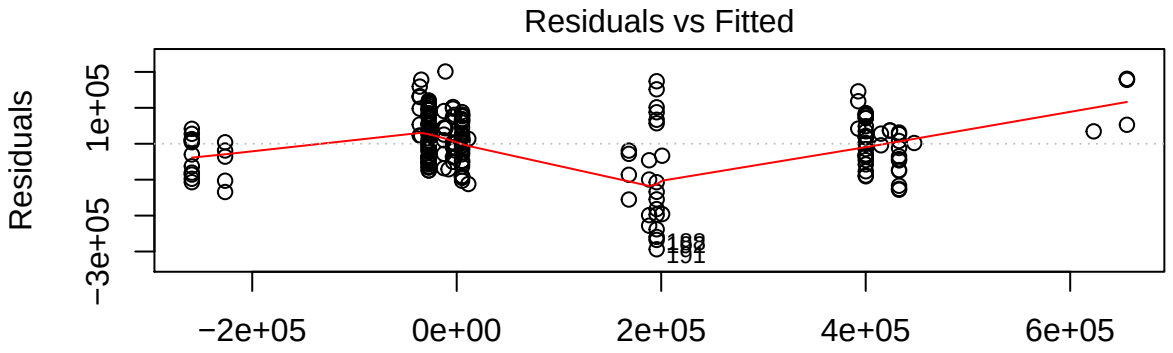


Figure 7

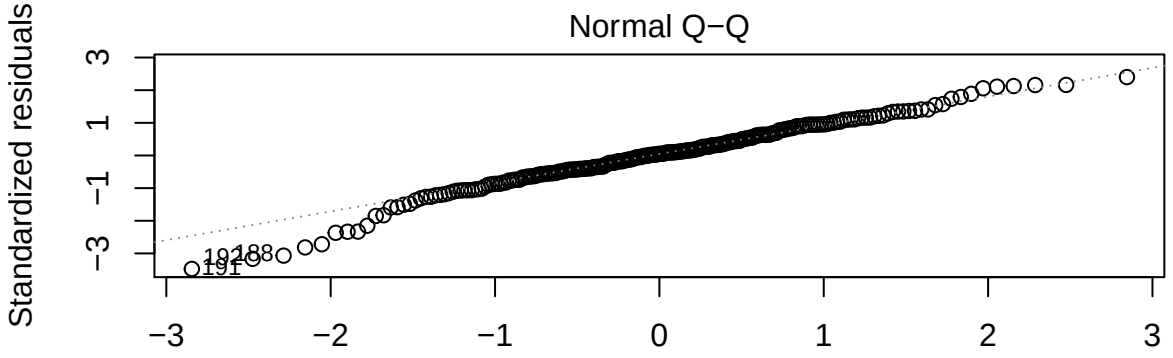


Figure 8

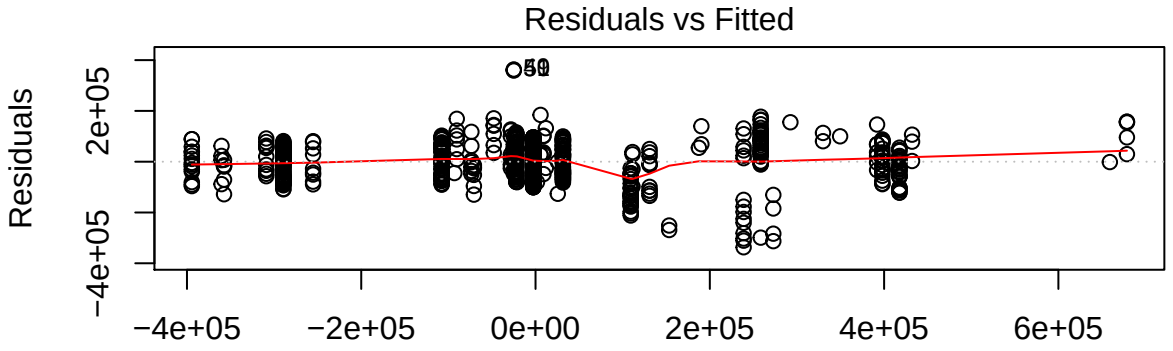
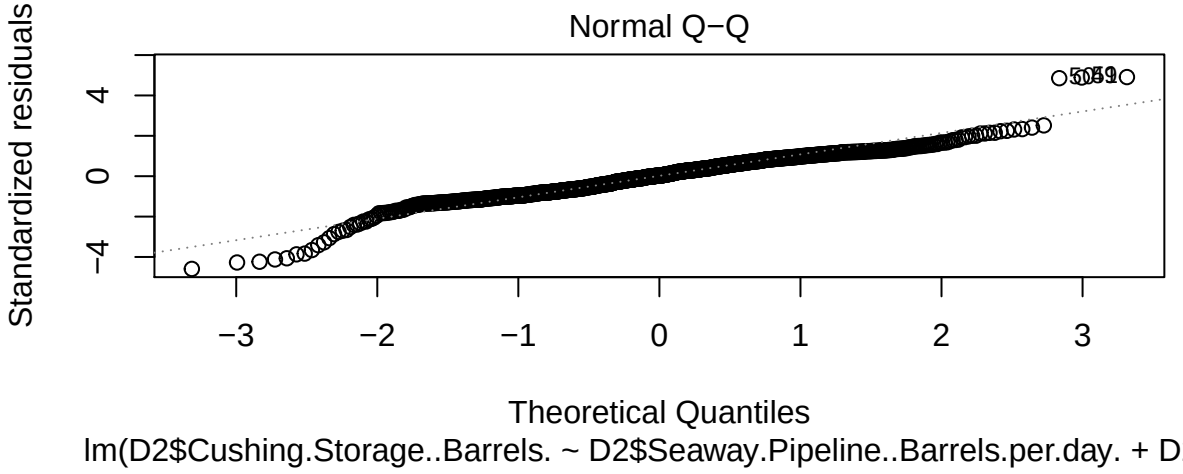


Figure 8


Normal Q−Q
Standardized residuals
4
0
−4
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(D2$Cushing.Storage..Barrels. ~ D2$Seaway.Pipeline..Barrels.per.day. + D2 ...