# 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.

So 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.

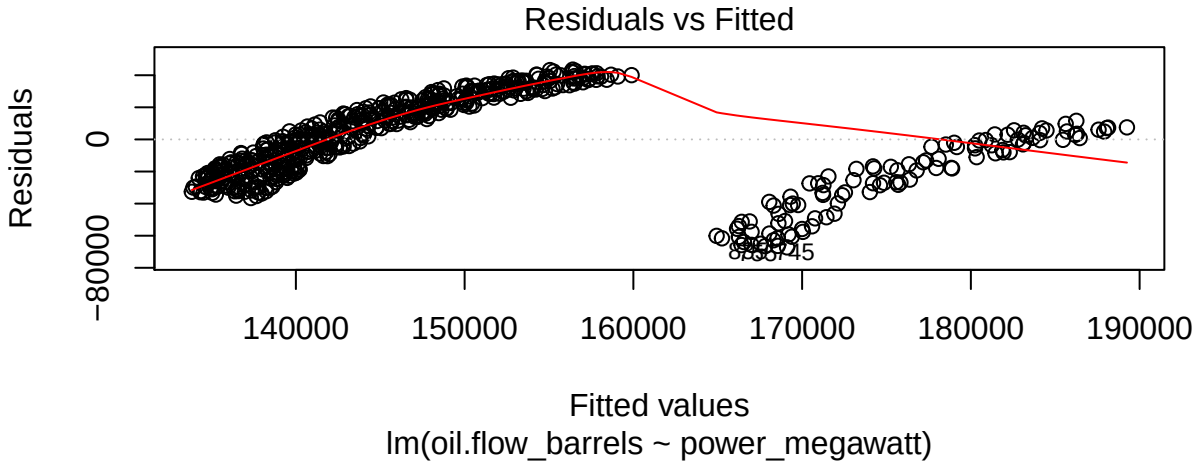
Now we can begin the creation of our model. Let's start with the easiest option and make a linear model.

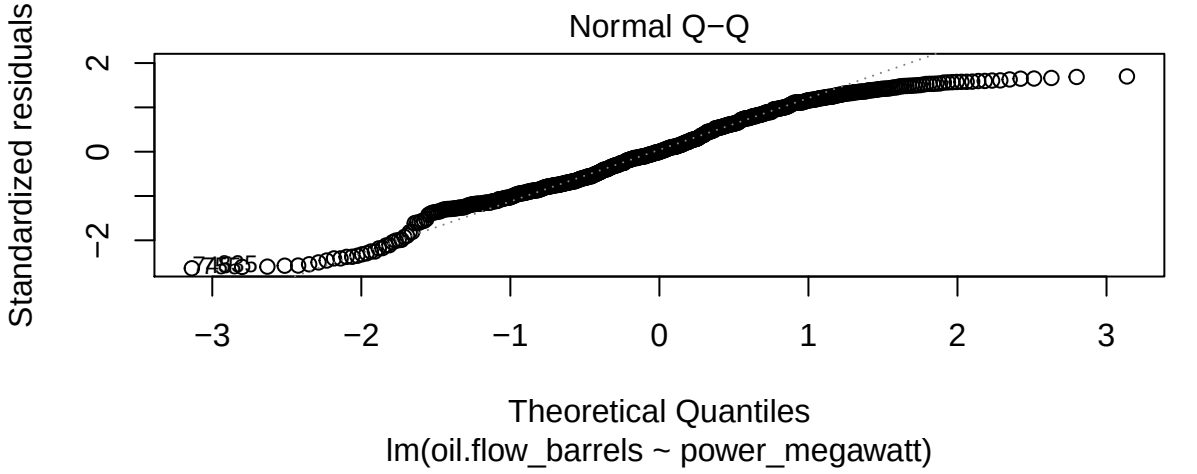
```
oil.lm = lm(oil.flow_barrels~power_megawatt, data = training.data1)
summary(oil.lm)
```

```
##
## 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 a different story. 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.

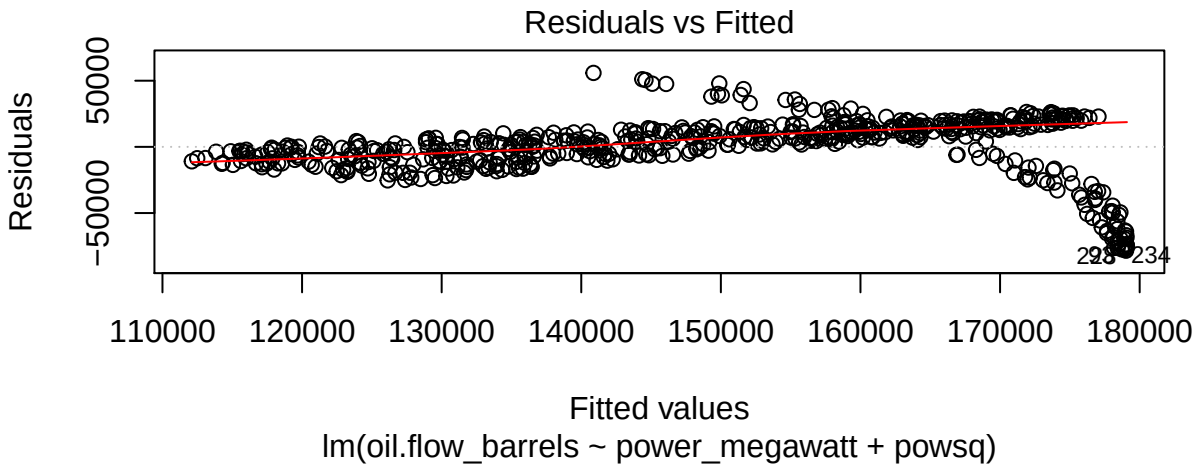
To see if the data from 2017 is causing troubles, let's remove it entirely and find what happens then (Fig. 2). Here we are getting somewhere. Notice the 2017 data must is the problem. As we noticed in our original residuals, the data is split, and now we can see how and where. However, we cannot and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

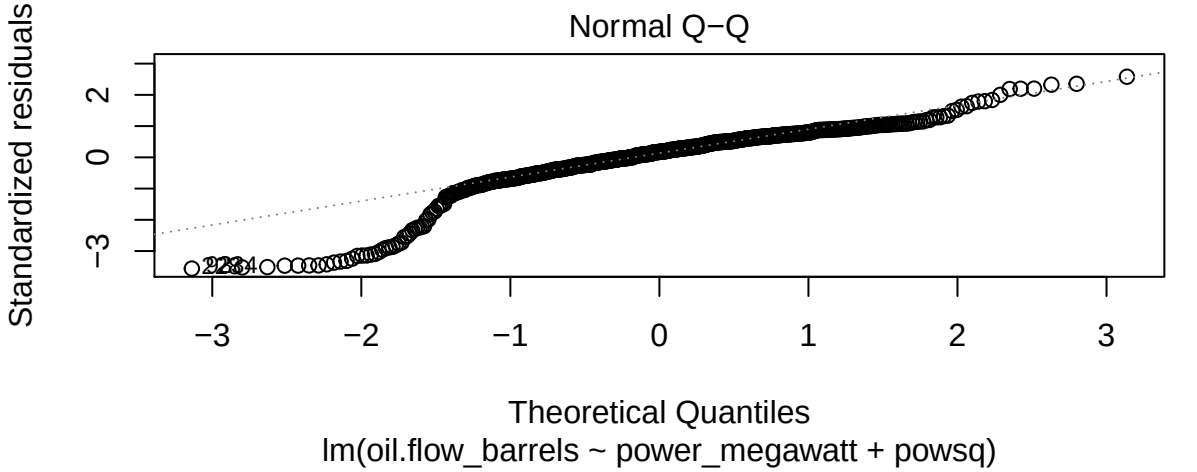
Before we attempt to use a new model, I would first like to see if this linear model could be improved by using quadratic variables. To do this, we will simply add the quadratic values for the power in megawatts to the original dataframes.

```
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. This leads me to question the quality of this model. 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.

```
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.

```
set.seed(225566)
training.data1 = sample_frac(prob1_data, size = 2/3)
```

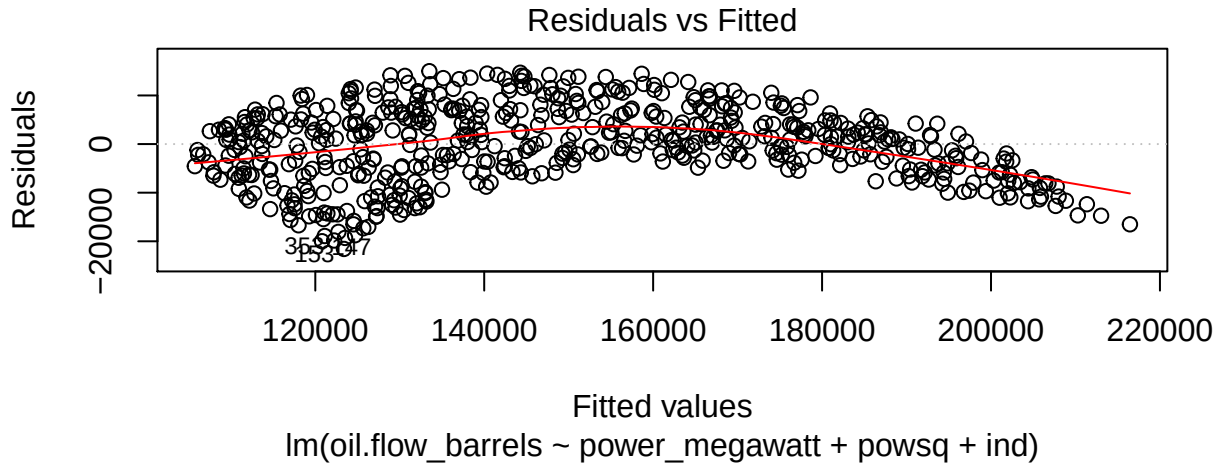
```
test.data1 = anti_join(prob1_data, training.data1, by = 'Date')
training.data1 = mutate(training.data1, powsq = power_megawatt^2)
test.data1 = mutate(test.data1, powsq = power_megawatt^2)
prob1_data.mod = mutate(prob1_data, powsq = power_megawatt^2)
```

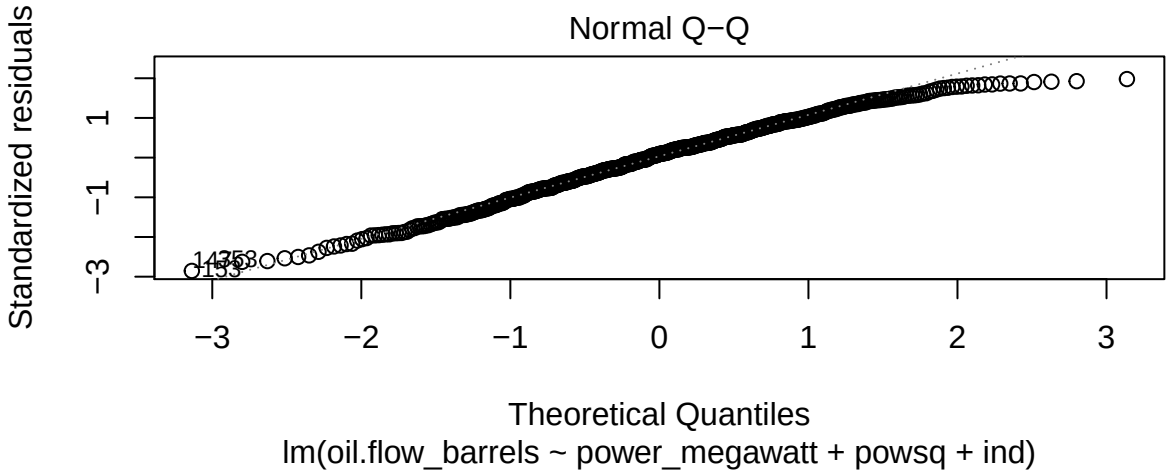
And now we can make our new model! Since the model with quadratic variables looked better than the one without, let's expand on that one.

```
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.



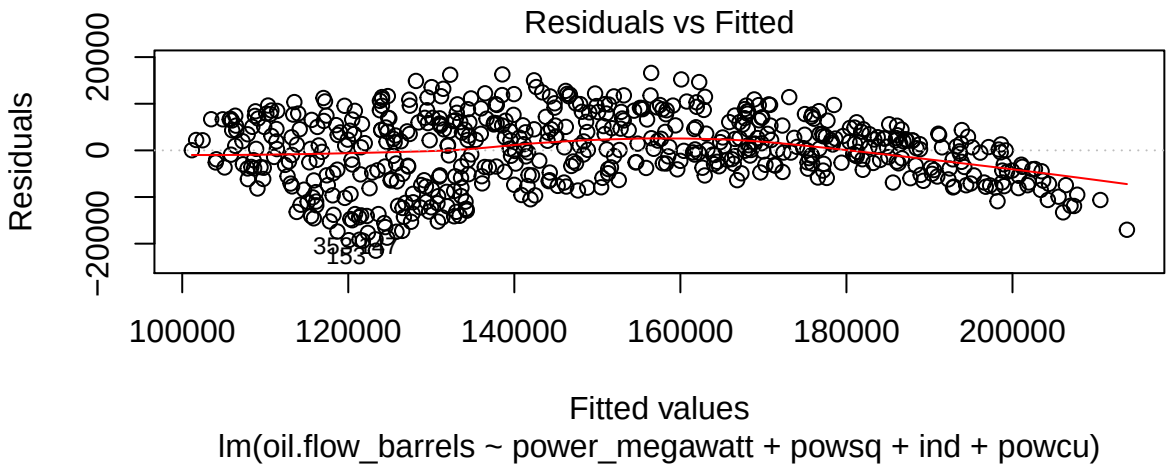


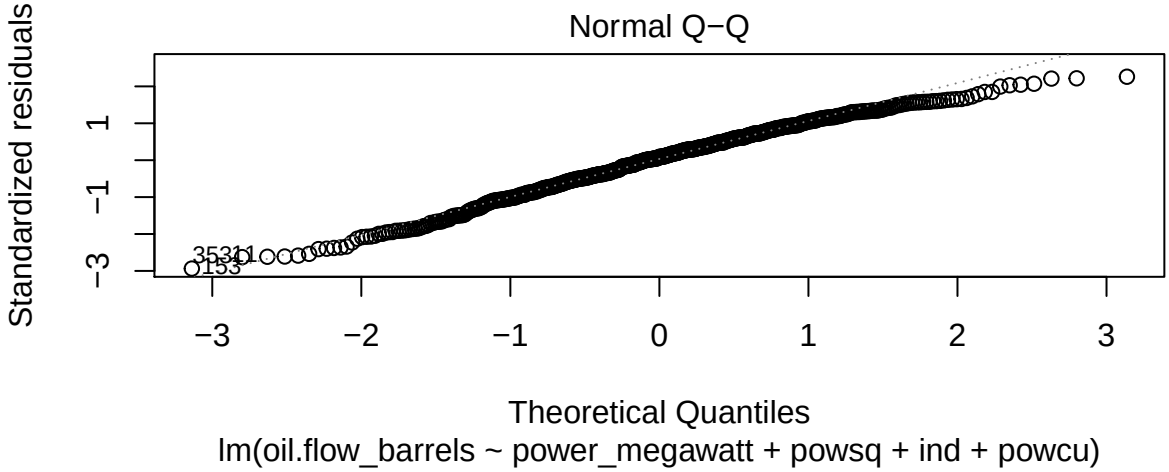

This is interesting, it seems as though we almost are still in need of additional variables. We could keep adding variables to the mix, but that may begin to make things messy. Just to check, let's try adding in the cubed power variables. To make things simple again, we will just go on and rename the 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 much better. The downside is that now our model is getting more and more complex, though the goal is always to try to keep it relatively simple. For the sake of accuracy though, we will continue the use of this model, and test it to see if it works well.

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 good compared to the standard deviation of the oil flow. So I think we can conclude that this model works relatively 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.

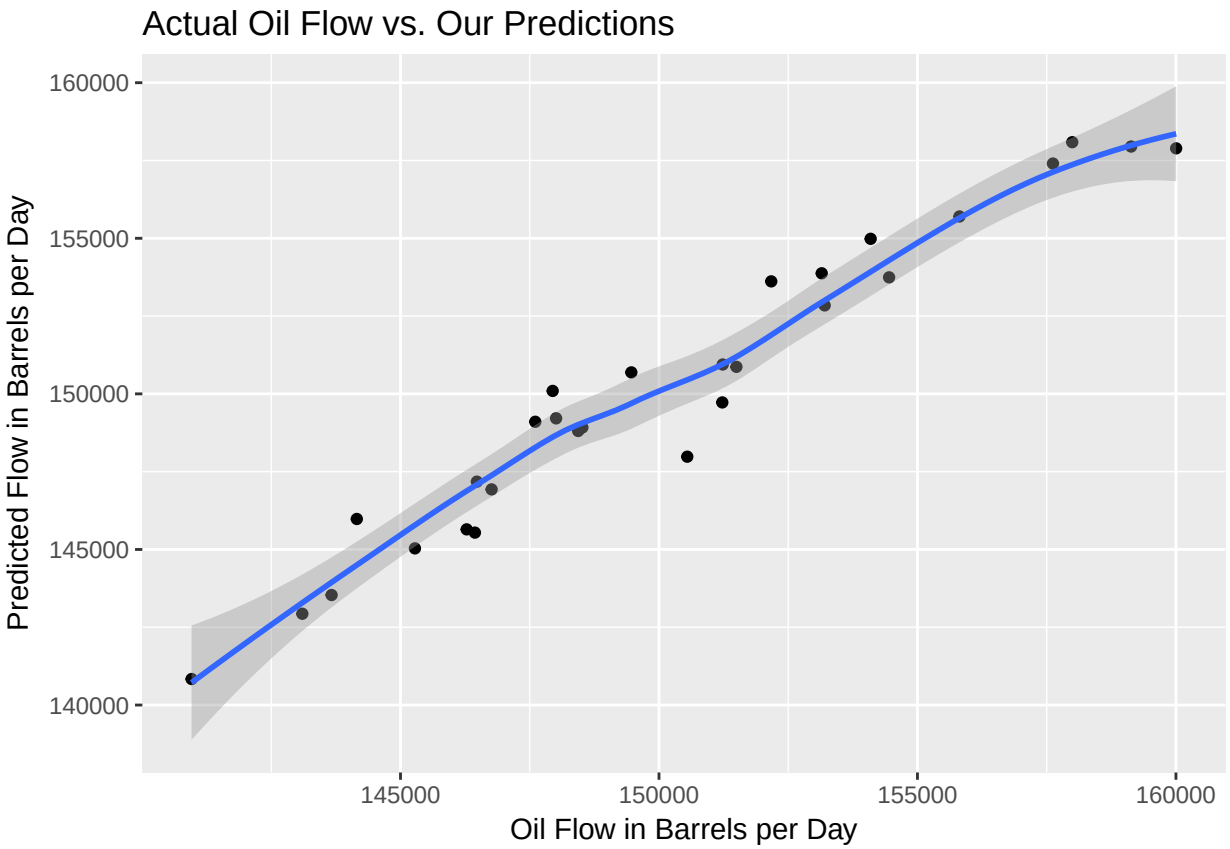
```
oil.pred = predict(oil.lm, prob1_data.mod)
oil.pred_data = data.frame(prob1_data$Date, oil.pred)
colnames(oil.pred_data) = c('Date', 'oil.predict')
oil.pred_ave = oil.pred_data %>% group_by(month=floor_date(Date, "month")) %>%
          summarize(oil.predict=mean(oil.predict))
```

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')
```

Now we just need to make a nice plot that consumers might like to see to show the accuracy of the model.

```
## `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 would also be easy to understand for consumers who don't really care about the observed error and things like that.

So in conclusion for this problem, we managed to come up with a model that does well enough. 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 Work:

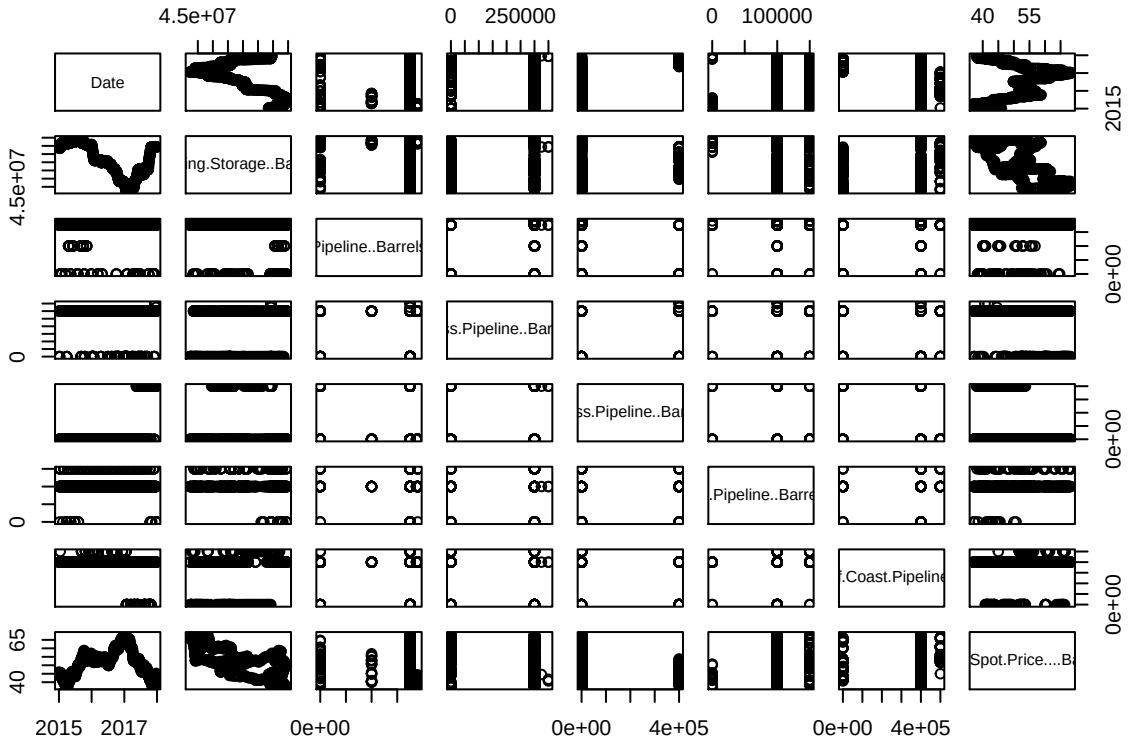
I have created a table in Excel that I read into R which is presented in Table 3. This table completes the first part of the problem, finding all of the required data and information for each pipeline. I found this information by doing some quick research on the internet. I hope that everything is correct, 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`) 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.

```
pairs(prob2_data)
```



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.

```
set.seed(225566)
training.data2 = sample_frac(prob2_data, size = 2/3)
test.data2 = anti_join(prob2_data, training.data2, by = 'Date')
```

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..Ba
                + 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
## -21342860  -6648862   -994911   8419572  17341011
##
## Coefficients:
##                                                  Estimate Std. Error
## (Intercept)                                     5.800e+07  2.359e+06
## Seaway.Pipeline..Barrels.per.day.              -9.183e+00  3.562e+00
## Pony.Express.Pipeline..Barrels.per.day.         2.063e+01  3.109e+00
## Dakota.Access.Pipeline..Barrels.Per.Day.        3.036e+00  2.515e+00
## White.Cliffs.Pipeline..Barrels.Per.Day.        -5.433e+01  1.616e+01
## TransCanada.Gulf.Coast.Pipeline..Barrels.Per.day.  1.724e+01  3.011e+00
##                                                 t value Pr(>|t|)
## (Intercept)                                      24.584  < 2e-16 ***
## Seaway.Pipeline..Barrels.per.day.                -2.578 0.010129 *
## Pony.Express.Pipeline..Barrels.per.day.           6.635 6.36e-11 ***
## Dakota.Access.Pipeline..Barrels.Per.Day.          1.207 0.227885
## White.Cliffs.Pipeline..Barrels.Per.Day.          -3.361 0.000817 ***
## TransCanada.Gulf.Coast.Pipeline..Barrels.Per.day.   5.725 1.52e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9207000 on 720 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.1143, Adjusted R-squared:  0.1081
## F-statistic: 18.58 on 5 and 720 DF,  p-value: < 2.2e-16
```

---

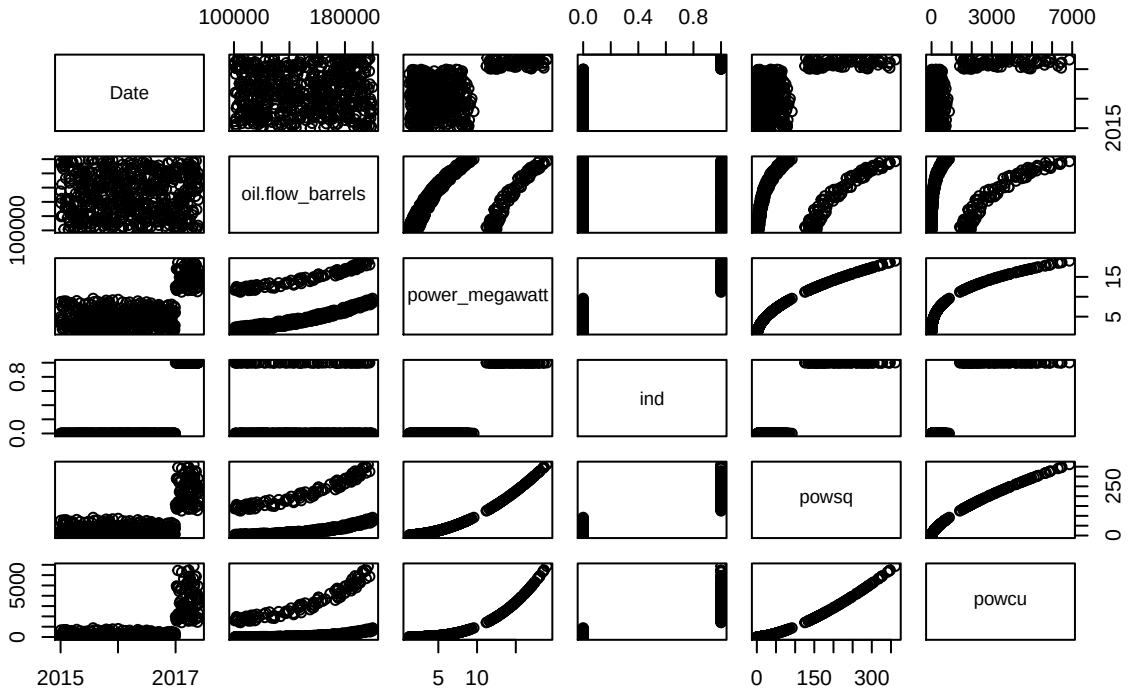
# Appendix

Table 1: Problem 1 Data Glimpse

| Date | oil.flow_barrels | power_megawatt | ind |
|---|---|---|---|
| 2015-01-01 | 155117 | 4.969950 | 0 |
| 2015-01-02 | 155002 | 5.228080 | 0 |
| 2015-01-03 | 195091 | 8.769649 | 0 |
| 2015-01-04 | 138447 | 3.624437 | 0 |
| 2015-01-05 | 119406 | 3.021225 | 0 |
| 2015-01-06 | 173907 | 6.359465 | 0 |

```
pairs(training.data1, main = 'Figure 1')
```

**Figure 1**




```
rem_year = filter(training.data1, Date < '2017-01-01')
pairs(rem_year, main = 'Figure 2')
```

# Figure 2

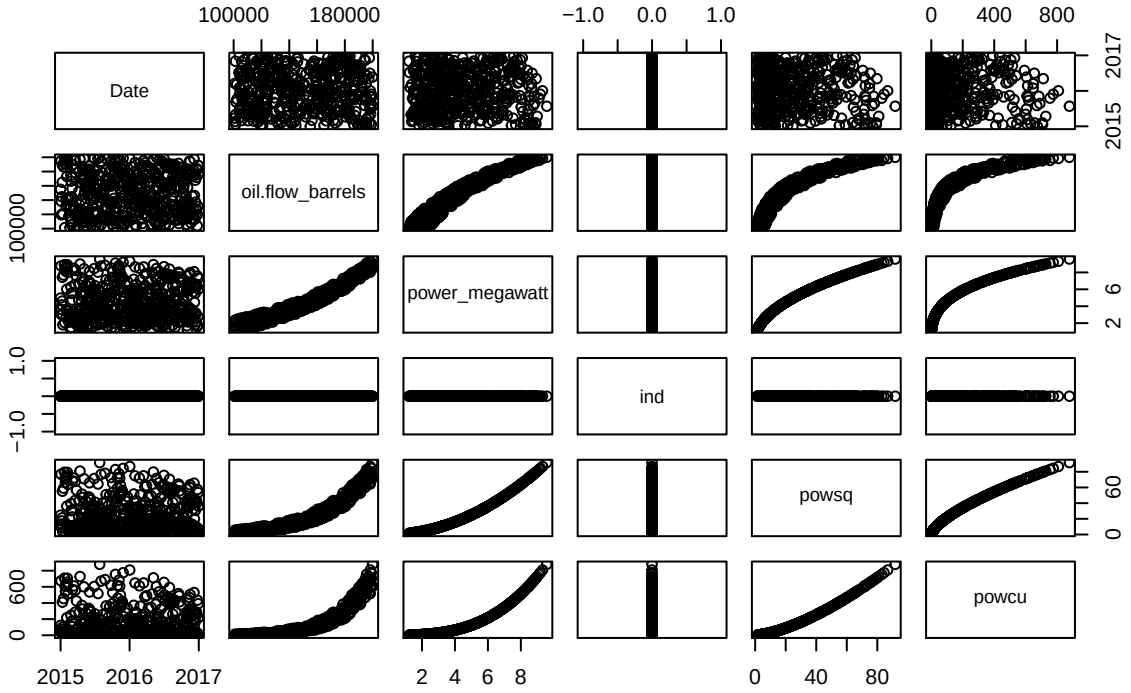



Table 2: Average Oil Flow per Month

| 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 |

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 |

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 |