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`. Below is a slight glimpse at what this data looks like:

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

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.data = sample_frac(prob1_data, size = 2/3)
test.data = anti_join(prob1_data, training.data, by = 'Date')
```

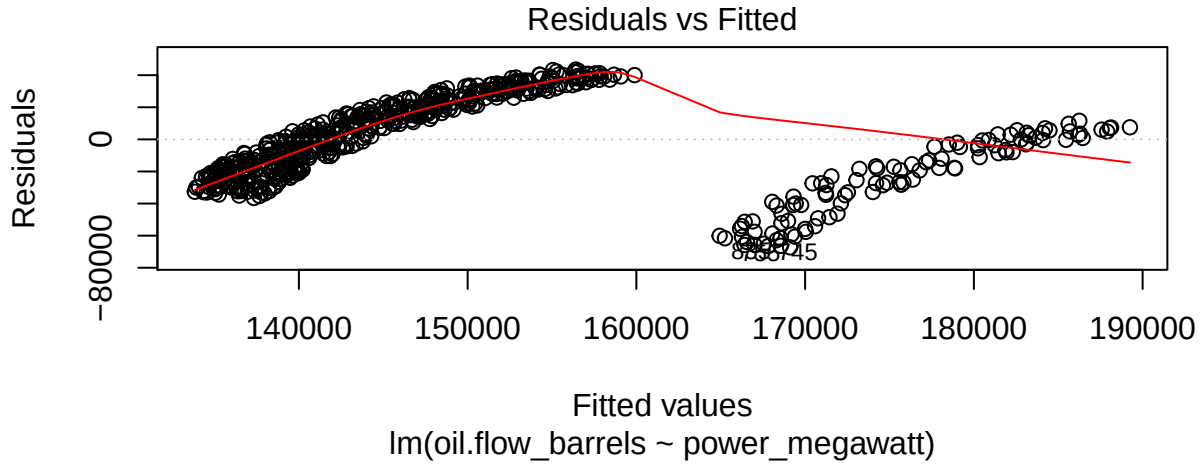
Here we are using the `dplyr` package to easily organize the test and training datasets. First we set the seed so that we get the same training data every time we run this code. The training data here is $\frac{2}{3}$ of the original data, while the test data is the leftover $\frac{1}{3}$. We don't use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

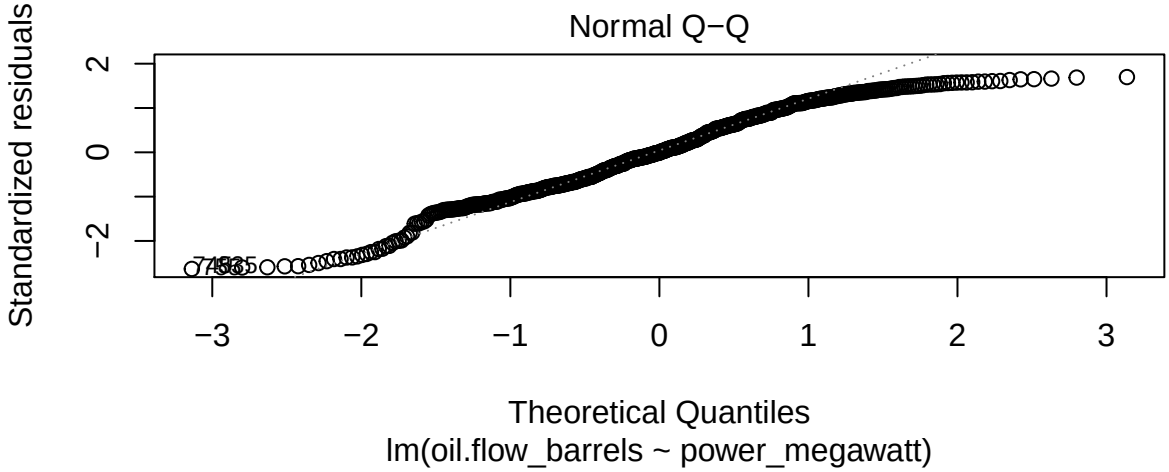
Now we can begin the creation of our model. Let's start with the easiest option and make a linear model.

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

```
##
## Call:
## lm(formula = oil.flow_barrels ~ power_megawatt, data = training.data)
##
## 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-statisctic which is very promising. Just to make sure let's continue to check this model by looking at the residuals.

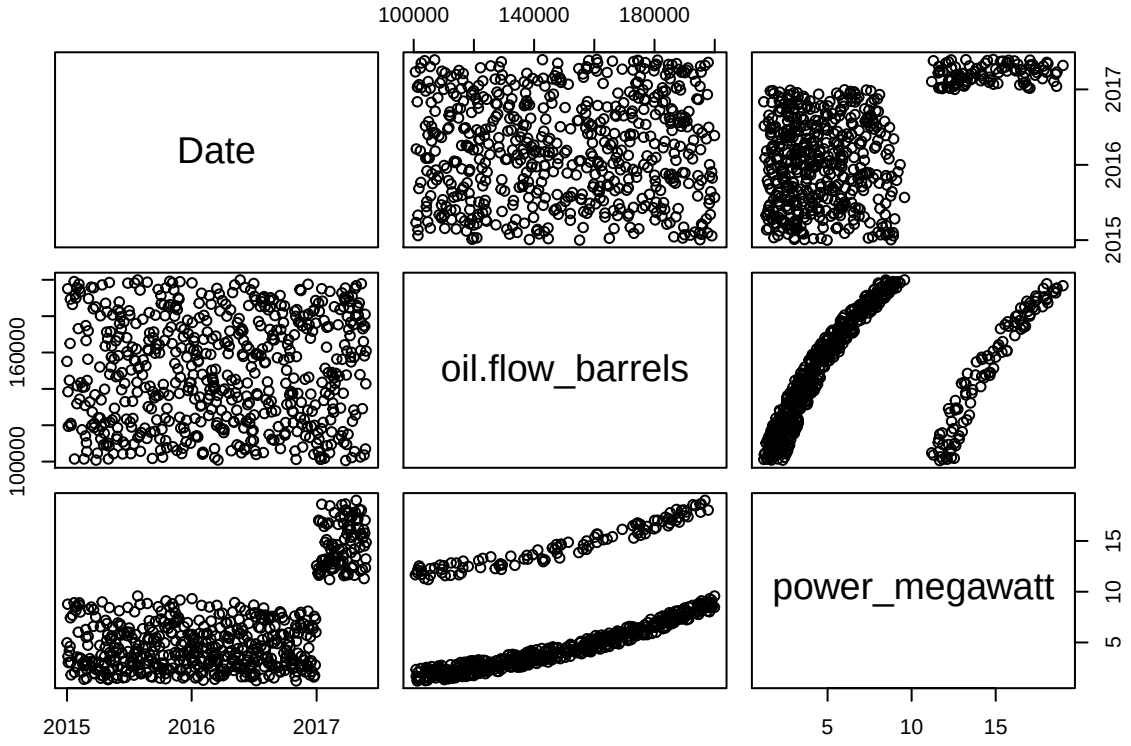




Here we can see that our residuals clearly show a different story. From our first residual plot, we can see that it almost seems as though our data is split into 2 parts. And from the Q-Q plot we see that we definitely have some outliers and maybe some noise. So maybe a simple linear model is not our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

To further our exploration here, let's look at some correlations in the data using the `pairs` function in R.

```
pairs(training.data)
```

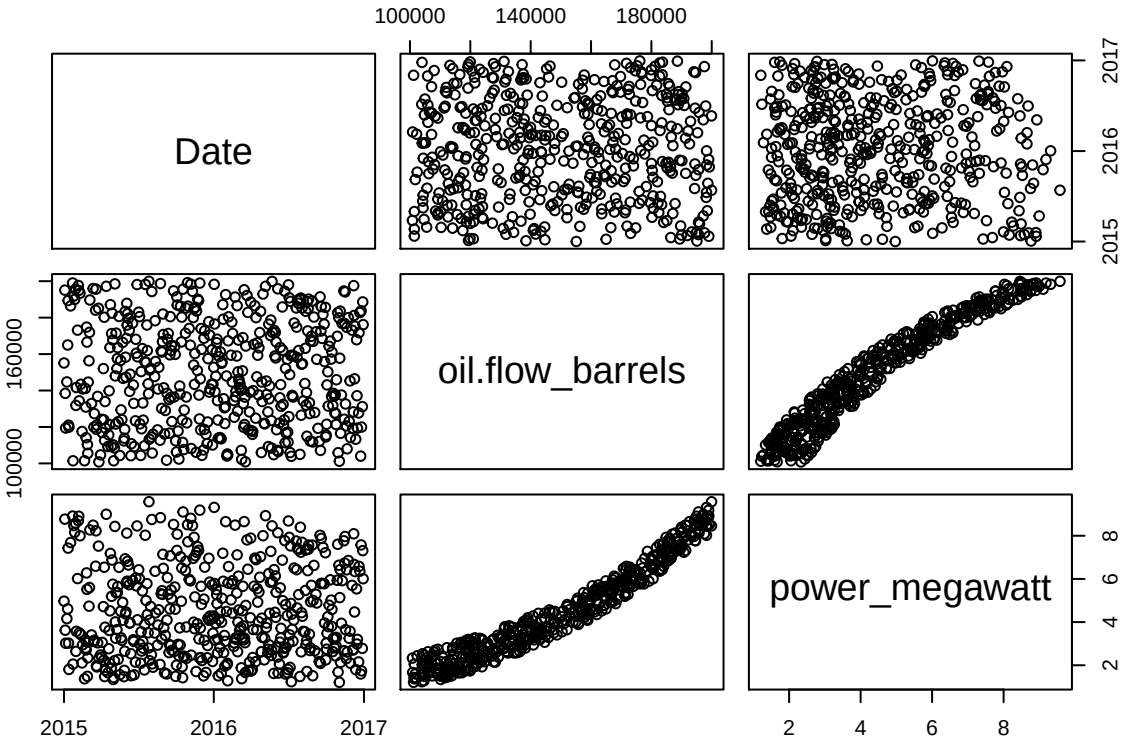


This plot shows some interesting facts about this data. First off, as probably expected, Date vs. the oil flow

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.

```
rem_year = filter(training.data, Date < '2017-01-01')
pairs(rem_year)
```



Now we're 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.

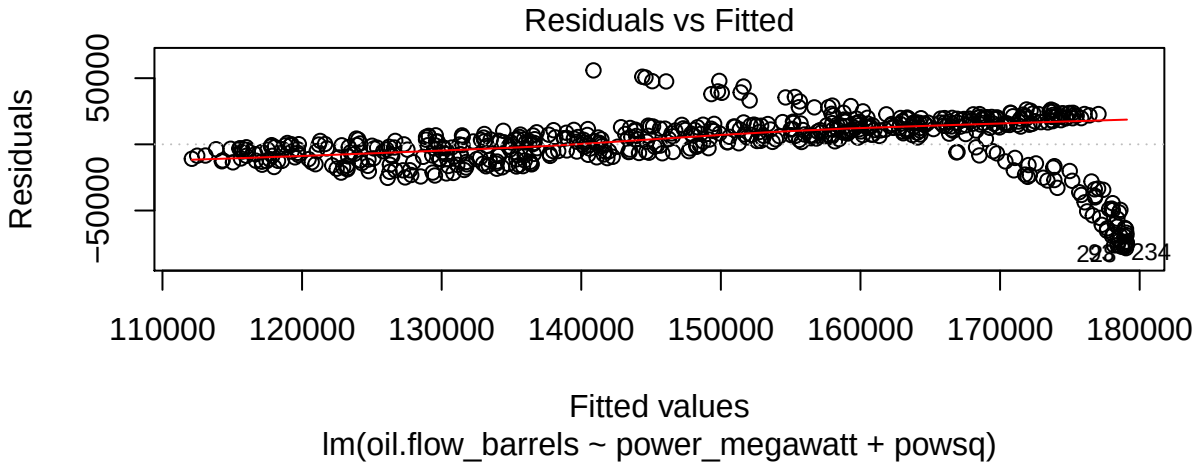
Before we attempt to use a new model, I would first like to see if this linear model could be improved by using quadratic variables. To do this, we will simply add the quadratic values for the power in megawatts to the original dataframes.

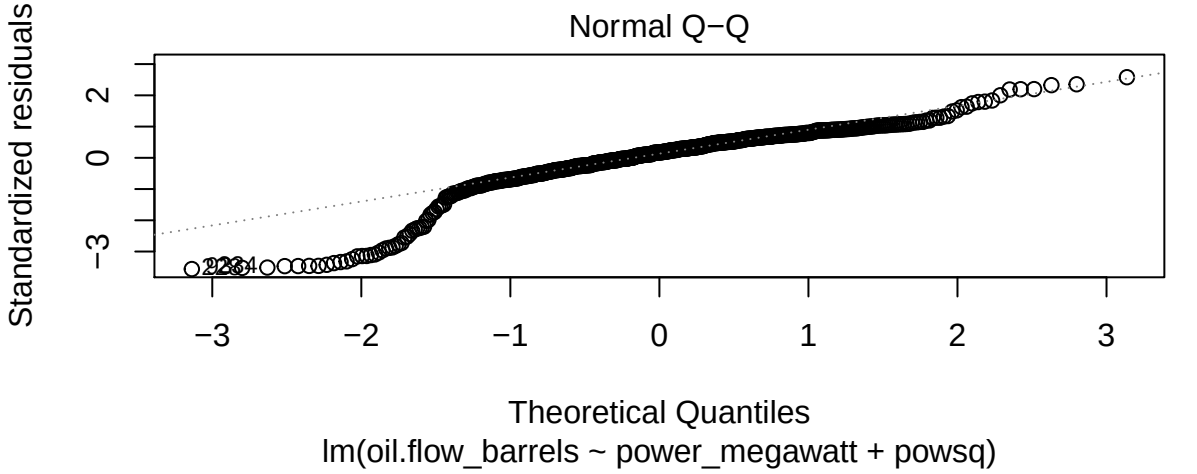
```
training.data = mutate(training.data, powsq = power_megawatt^2)
test.data = mutate(test.data, powsq = power_megawatt^2)
oil.lmq = lm(oil.flow_barrels~power_megawatt+powsq, data = training.data)
summary(oil.lmq)
```

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

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 funciton `predict` which can be used to predict responses for new data based on a fitted model.

```
oilflow.hat = predict(oil.lmq, test.data)
```

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.data$oil.flow_barrels,oilflow.hat))
obs.test.error
```

```
## [1] 378776529
```

Clearly the linear model is not working as well as we may have hoped. Let's try making a random forest model using the `caret` package.

```
set.seed(225566)
ctrl=trainControl(method="boot632")
rf.model=train(oil.flow_barrels~power_megawatt, data=training.data, method="rf",
               ntree=5000, trControl=ctrl, tuneGrid=data.frame(mtry = 1:8))
rf.model
```

```
## Random Forest
##
## 586 samples
##   1 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 586, 586, 586, 586, 586, 586, ...
## Resampling results across tuning parameters:
##
##   mtry  RMSE      Rsquared   MAE
##   1     6349.884  0.9478769  4970.053
##   2     6350.298  0.9478734  4969.866
##   3     6347.432  0.9479022  4968.202
##   4     6349.730  0.9478711  4968.929
##   5     6347.861  0.9479034  4968.698
##   6     6350.055  0.9478759  4969.006
##   7     6351.948  0.9478433  4970.789
##   8     6349.676  0.9478792  4969.600
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 3.
```

```
oilflow.fit=predict(rf.model,training.data)
mean(L(training.data$oil.flow_barrels,oilflow.hat))
```

```
## [1] 1246419185
```

```r
oilflow.hat=predict(rf.model,test.data)
mean(L(test.data$oil.flow_barrels,oilflow.hat))
```

```
## [1] 45149023
```

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: