# 2

## a

```
# Split and build model
we.train = we[we$Year <= 2017,]
we.test = we[we$Year == 2018,]
lm.model2a = lm(Wrangler.Sales ~ Year + Unemployment.Rate +
                  Wrangler.Queries + CPI.Energy + CPI.All, data=we.train)
```

Inspecting the summary reveals that all except **CPI.All** are significant. Lets drop this predictor and refit the model. The $R^2$ of this new model is fairly good with 0.824, but the $OSR^2$ is only 0.5172. Also there is suspicious results, like a negative coeffisient for the **Year** predictor. Inspecting the correlation matrix reveals that **Year, Unemployment.Rate Wrangler.Queries** are all heavily correlated. After trying different combinations , the best choice seems to be to drop **Year** and **Unemployment.Rate** based on the values of $OSR^2$ and $R^2$.

## b

```
lm.model2b = lm(Wrangler.Sales ~ Wrangler.Queries + CPI.Energy, data=we.train)
```

**Memo**

This model based on linear regression predicts yearly Wrangler sales. The model is based on a dataset of sales between 2010 and 2017. Since there was a heavy correlation between the year, unemployment rate and google searches for "jeep wrangler" in that period, only the query data is included in the predictors as this seemed to give the best $R^2$ and $OSR^2$. The model has a $R^2$ value of 0.7497. To predict the sales for a year, calculate

$$-6036.38 + 208.83x_1 + 30.65x_2 \tag{1}$$

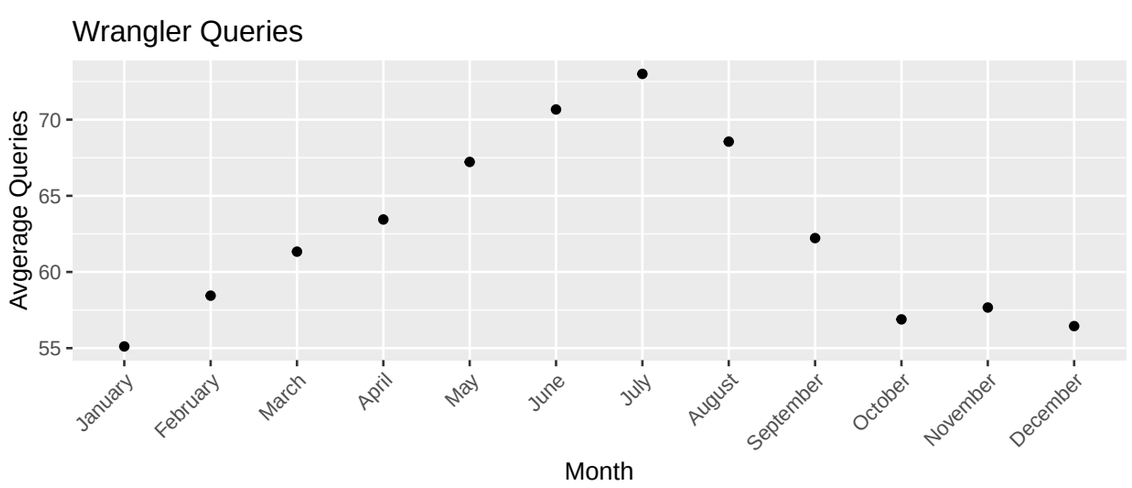
where $x_1$ is a (normalized) approximation of the number of Google searches for jeep wrangler in the United States in the given month and year and $x_2$ is the CPI index for the energy sector. The model therefore suggests that the sale is increasing with the CPI index and wrangler searches. Evaluating the model on unseen date from 2018 gives an $OSR^2$ of 0.6943. This is a bit lower than for the training data, which is expected.

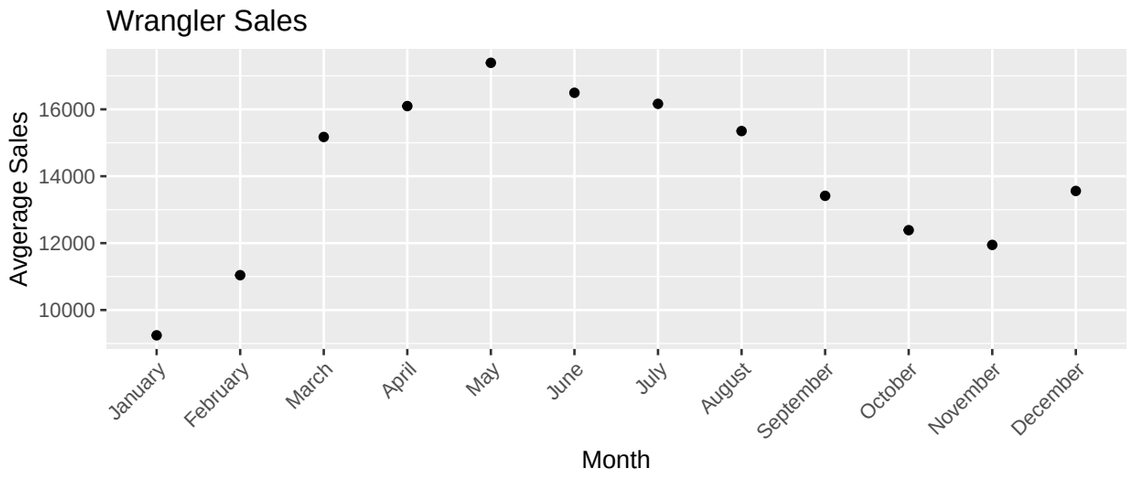
```
pred = predict(lm.model2b, newdata = we.test)
actual = we.test$Wrangler.Sales
baseline = mean(we.train$Wrangler.Sales)
OSR2 = 1 - sum((actual - pred)**2) / sum((actual - baseline)**2)
OSR2
```

```
## [1] 0.6943399
```

## c

```
g = ggplot(salesdf, aes(x=Month.FactorUnique,y=MonthlyAvgSales))
g = g + geom_point()
g = g + labs(title="Wrangler Sales", y="Avgerage Sales", x = "Month")
g = g + theme(axis.text.x = element_text(angle=45, hjust = 1))
```






The first plot indicate that there is most queries for the Wrangler during the spring and summer. Logically this should correlate with sales, and the second plot shows it does, as expected. It suggests that most americans buy a new jeep when it is warm and good conditions for driving Jeeps. The low sales in January and February might also be partially caused by overspending during the holidays.

### d

```
lm.model2d = lm(Wrangler.Sales ~  Wrangler.Queries +
                  CPI.Energy + Month.Factor, data=we.train)
pred = predict(lm.model2d, newdata = we.test)
actual = we.test$Wrangler.Sales
baseline = mean(we.train$Wrangler.Sales)
OSR2 = 1 - sum((actual - pred)**2) / sum((actual - baseline)**2)
OSR2
```

```
## [1] 0.7527684
```

This modified model now also uses the data of the months to model the seasons. Like the previous problem illustrated, the month is an important predictor for the amount of sales, and as expected the inclusion of the month improves the model. The $R^2$ is now 0.8741, and $OSR^2$ is 0.7528. The new equation for the model is

$$-1599.6 + \sum_{i=1}^{12} a_i x_i + 187 x_{13} + 21.1 x_{14} + \tag{2}$$

where $x_{13}$ is the queries and $x_{14}$ is the CPI index as before, and $x_i$ is 1 if we are in the $i$-th month (January = 1, February = 2 and so on) of the year, 0 otherwise. The coeffisients for the months is given as

```
coef(lm.model2d)
```

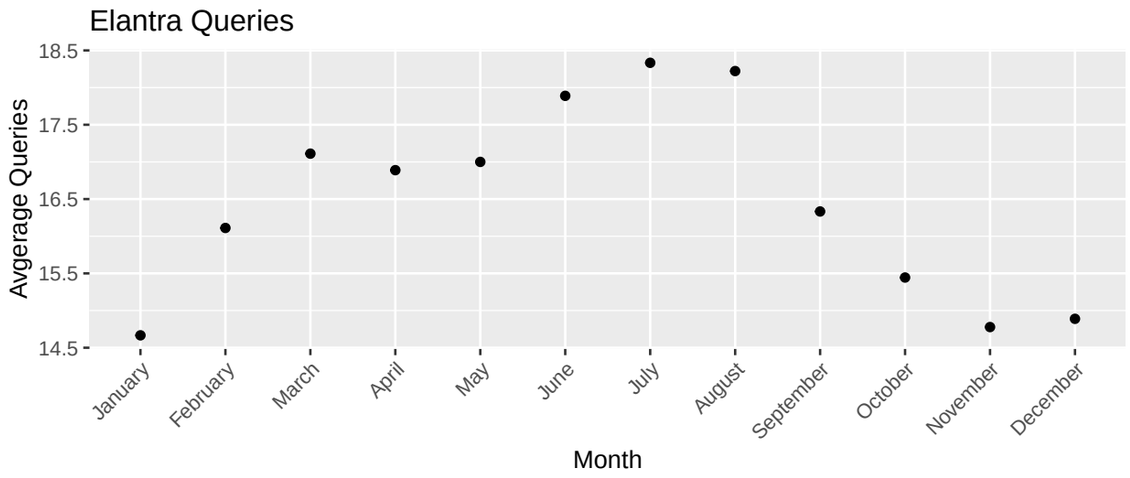
```
##           (Intercept)      Wrangler.Queries            CPI.Energy
##           -1599.60137             187.00090              21.11478
##    Month.FactorAugust  Month.FactorDecember  Month.FactorFebruary
##            -655.12347            -428.50538           -3041.48346
##   Month.FactorJanuary      Month.FactorJuly      Month.FactorJune
##           -3945.50940            -592.79952             -11.02042
##     Month.FactorMarch       Month.FactorMay  Month.FactorNovember
##            -435.61353            1439.98828           -2037.87258
##   Month.FactorOctober Month.FactorSeptember
##           -1090.13651           -1267.98333
```

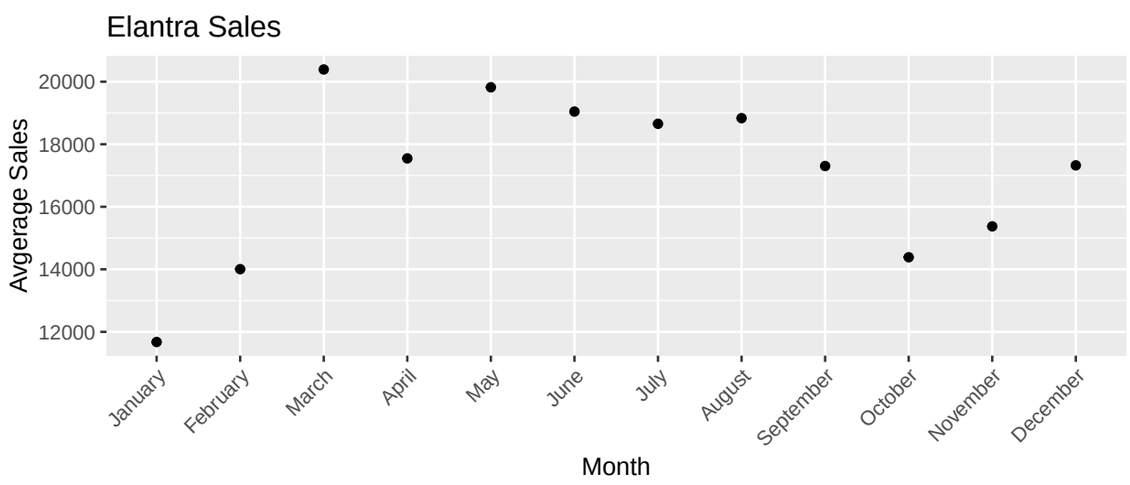
where each coeffisient is the differance in sales as compared to April ($a_4 = 0$). Of these variables, only **Wrangler.Queries**, **CPI.Energy** and the variables for January, February and November are statistically significant.

### e

```
lm.model2e = lm(Elantra.Sales ~  Elantra.Queries +
                  CPI.Energy + Month.Factor, data=we.train)
```

For the Elantra model, we get a $R^2$ of 0.4755 and an $OSR^2$ of $-2.452$. This is terrible. Simply guessing the mean would have been better for the test data.




**f**

```
cor(we[,c("Elantra.Sales", "Year", "Unemployment.Rate", "Elantra.Queries","CPI.Energy")])
```

```
##                   Elantra.Sales       Year Unemployment.Rate
## Elantra.Sales         1.0000000  0.2513623        -0.2869429
## Year                  0.2513623  1.0000000        -0.9842538
## Unemployment.Rate    -0.2869429 -0.9842538         1.0000000
## Elantra.Queries       0.3668348  0.6968071        -0.6464563
## CPI.Energy            0.1493700 -0.4766071         0.5025609
##                   Elantra.Queries CPI.Energy
## Elantra.Sales           0.3668348  0.1493700
## Year                    0.6968071 -0.4766071
## Unemployment.Rate      -0.6464563  0.5025609
## Elantra.Queries         1.0000000 -0.1452307
## CPI.Energy             -0.1452307  1.0000000
```

```
cor(we[,c("Wrangler.Sales", "Year", "Unemployment.Rate", "Wrangler.Queries","CPI.Energy")])
```

```
##                   Wrangler.Sales       Year Unemployment.Rate
## Wrangler.Sales         1.0000000  0.7374536        -0.7423252
## Year                   0.7374536  1.0000000        -0.9842538
## Unemployment.Rate     -0.7423252 -0.9842538         1.0000000
## Wrangler.Queries       0.8392407  0.9298112        -0.9202746
## CPI.Energy            -0.2822404 -0.4766071         0.5025609
##                   Wrangler.Queries CPI.Energy
## Wrangler.Sales           0.8392407 -0.2822404
## Year                     0.9298112 -0.4766071
## Unemployment.Rate       -0.9202746  0.5025609
## Wrangler.Queries         1.0000000 -0.4891910
## CPI.Energy              -0.4891910  1.0000000
```

We can see from the plots and correlation matrix that the Elentra Queries and Sales are much less correlated than Wrangler Queries and Sales. Also the Wrangler sales are more correlated with **CPI.Energy** than Elentra sales are. Higher correlation makes for better prediction. From a business viewpoint, I would guess the Jeep is more of a seasonal car, while the Eletra is an around the year model. So there is less patterns in

the data for our model to exploit.

I suspect we need more data that is statistically significant for the Elentra Sales to make any substantial progress. We could try playing around with interaction terms and other nonlinear feature maps in the regression model, but with insufficient data we can only get so far.

## g

If we produce more vehicles than actual demand, we would need to pay more for inventory storage and logistics. There would also be a risk of having to discount vehicles in order to clear up inventory space and reduce losses before arrival of new models. There are no direct costs when producing under demand. Still, we are losing out on business, and potentially making customers unhappy. In general I suspect the costs of producing too much is greater than making too few. So in general I would probably recommend producing less than predicted.

The current model is minimizing the loss function

$$L(w, w_0) = \sum_{i=1}^{N} (y_i - \hat{y_i})^2 \tag{3}$$

where $\hat{y_i} = \mathbf{w}^T\mathbf{x} + w_0$, which corresponds to the least square error. If we instead optimize the coeffisients over some other loss function that penalizes errors from estimating too high more, we would most likely get a more conservative model. Another option is to estimate the cost for each unit produced over demand, and the expected profit per sale. Then we could optimize the expected profit as a function of produced units based on the expected demand and standard error.