HW1

3032247297

3)

```r
#install.packages(c("dplyr", "ggplot2", "GGally", "broom"))
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

##
## Attaching package: 'GGally'

## The following object is masked from 'package:dplyr':
##
##     nasa
```

```r
library(readxl)
#install.packages("car")
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```r
wrangler <- read.csv("C:/Users/Murtz.Kizilbash/Desktop/ieor142/hw1/Wrangler142-Fall2019.csv")

#wrangler
```

a)

b) the linreg equation of my model is $y = -952.18 + 257.86(x) + \epsilon$. Someone should interpret the independent variables as "for every 1 change in this variable, sales changes by the coefficient of the variable".

ii) I selected the variables based on their p values and statistical significance.

iii) Yes, the signs of the coefficients make sense because we would expect that if more people are searching for wranglers, then they are more inclined to buy one leading to more sales. As for unemployment, when unemployment goes down, people have more money to spend because they have income, therefore when unemployment goes up, sales should go down. For CPI, if it goes up, then goods become more expensive therefore people are less likely to buy an item such as a Jeep Wrangler.

iv) The model fits our training data with an r squared value of .79 so it is doing a great job of predicting the training values, this is because we are using the training data to build our model so it should fit to it pretty well.

```r
broncos <- read_excel("C:/Users/Murtz.Kizilbash/Desktop/ieor142/hw1/multiTimeline (1).xlsx", skip = 1)

rsq <- function (x, y) cor(x, y) ^ 2


wrangler$fordquery <- broncos$`ford bronco: (United States)`
wrangler.train <- filter(wrangler, Year >= 2010 & Year <= 2015 )

wrangler.test <- filter(wrangler, Year >= 2016 & Year <= 2019 )

wrangler.Indep.Vars <- wrangler[5:8]

wranglerSales.predict <- lm(WranglerSales ~ Unemployment + WranglerQueries + CPI.All + CPI.Energy, data

#summary(wranglerSales.predict)

#drop cpi energy
wranglerSales.predict1 <- lm(WranglerSales ~ Unemployment + WranglerQueries + CPI.All, data = wrangler.t
#summary(wranglerSales.predict1)

#drop cpi.all
wranglerSales.predict2 <-lm(WranglerSales ~ Unemployment + WranglerQueries, data = wrangler.train)
#summary(wranglerSales.predict2)

rsq(predict(wranglerSales.predict1, wrangler.train), wrangler.train$WranglerSales)
```

```
## [1] 0.7942801
```

b)

```r
wranglerSales.predict_season <- lm(WranglerSales ~ Unemployment + WranglerQueries + CPI.All + CPI.Energ
summary(wranglerSales.predict_season)
```

```
##
## Call:
## lm(formula = WranglerSales ~ Unemployment + WranglerQueries +
##     CPI.All + CPI.Energy + MonthFactor, data = wrangler.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2936.3  -671.6  -184.2   538.5  8256.9
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -69628.03   54007.71  -1.289  0.20262
## Unemployment            845.80    1001.10   0.845  0.40178
## WranglerQueries         175.69      61.67   2.849  0.00613 **
## CPI.All                 317.32     236.31   1.343  0.18475
## CPI.Energy              -25.28      28.71  -0.880  0.38240
## MonthFactorAugust       -62.76     944.94  -0.066  0.94728
## MonthFactorDecember    -175.82    1060.77  -0.166  0.86896
## MonthFactorFebruary   -1078.14     906.32  -1.190  0.23923
## MonthFactorJanuary    -3262.64     953.93  -3.420  0.00117 **
## MonthFactorJuly        -176.09    1003.11  -0.176  0.86128
## MonthFactorJune         313.29     948.52   0.330  0.74241
## MonthFactorMarch       -173.75     887.58  -0.196  0.84551
## MonthFactorMay         1894.71     910.63   2.081  0.04205 *
## MonthFactorNovember   -1660.69    1008.11  -1.647  0.10509
## MonthFactorOctober     -776.15     986.73  -0.787  0.43484
## MonthFactorSeptember   -945.17     890.99  -1.061  0.29333
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1535 on 56 degrees of freedom
## Multiple R-squared:  0.8698, Adjusted R-squared:  0.8349
## F-statistic: 24.94 on 15 and 56 DF,  p-value: < 2.2e-16
```

i) The new model has several coefficients that correspond to the increase or decrease in sales at any given month. the new regression equation is as follows:

$sales = -69628.03 + x(845.80 + 175.69 + 317.32 - 25.28) + monthfactor(y)$

One should interpret the coeff of the month factor variables as the increase or decrease in sales during that month. For example if the month is july then we should expect 176 fewer sales.

ii) The training set r^2 is .8698, the variables that are significant are the monthfactors for July and March, alongside Wrangler Queries.

iii) I think that including the variable MonthFactor does improve the quality of the model, however I do worry about overfitting since the statistical significance of the months was only true on 2 of the 11 months. Therefore it is hard to believe that there is extreme seasonality with Wranglers, it could just be a slight correlation.

iv) Instead of having the months as factors of one month I would slice them based on a couple of months. In this case Fall, Spring, Summer, Winter. For example I would set the month factor for isWinter to be 1 if the months that the sales we are looking at are November, December, and January. In this way we are looking at actual seasonality instead of just one month, since one month is hardly equivalent to a season. I think this new way would improve the model because we will have less coefficients ultimately in our regression equation.

v)

```
wranglerSales.final <- lm(WranglerSales ~ Unemployment + WranglerQueries + CPI.All + MonthFactor, data =

summary(wranglerSales.final)
```

```
##
## Call:
## lm(formula = WranglerSales ~ Unemployment + WranglerQueries +
##     CPI.All + MonthFactor, data = wrangler.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3009.7  -682.2  -135.0   580.2  8217.5
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -25567.033  20264.000  -1.262  0.21220
## Unemployment             72.658    479.716   0.151  0.88015
## WranglerQueries         189.436     59.548   3.181  0.00237 **
## CPI.All                 122.298     82.147   1.489  0.14206
## MonthFactorAugust       -62.480    943.075  -0.066  0.94741
## MonthFactorDecember      -7.806   1041.406  -0.007  0.99405
## MonthFactorFebruary   -1064.185    904.396  -1.177  0.24421
## MonthFactorJanuary    -3169.514    946.180  -3.350  0.00144 **
## MonthFactorJuly        -243.638    998.192  -0.244  0.80805
## MonthFactorJune         253.589    944.229   0.269  0.78923
## MonthFactorMarch       -177.858    885.813  -0.201  0.84158
## MonthFactorMay         1860.758    908.015   2.049  0.04505 *
## MonthFactorNovember   -1476.834    984.294  -1.500  0.13903
## MonthFactorOctober     -659.896    975.929  -0.676  0.50167
## MonthFactorSeptember   -893.439    887.294  -1.007  0.31823
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1532 on 57 degrees of freedom
## Multiple R-squared:  0.868,  Adjusted R-squared:  0.8356
## F-statistic: 26.78 on 14 and 57 DF,  p-value: < 2.2e-16
```

```
p = predict(wranglerSales.final, wrangler.test)

test = wrangler.test$WranglerSales

rsq <- function (x, y) cor(x, y) ^ 2

rsq(p, test)
```

```
## [1] 0.6375503
```

```
wrangler
```

```
##    MonthNumeric MonthFactor Year WranglerSales Unemployment
## 1             1     January 2010          4888          9.8
## 2             2    February 2010          5967          9.8
## 3             3       March 2010          8410          9.9
## 4             4       April 2010          8327          9.9
## 5             5         May 2010          9634          9.6
## 6             6        June 2010          8923          9.4
## 7             7        July 2010         10043          9.4
## 8             8      August 2010          7666          9.5
## 9             9   September 2010          7765          9.5
## 10           10     October 2010          7908          9.4
## 11           11    November 2010          6552          9.8
## 12           12    December 2010          8227          9.3
## 13            1     January 2011          6444          9.1
## 14            2    February 2011          7636          9.0
## 15            3       March 2011          8807          9.0
## 16            4       April 2011          9051          9.1
## 17            5         May 2011         10008          9.0
## 18            6        June 2011         11290          9.1
## 19            7        July 2011         14355          9.0
## 20            8      August 2011         12949          9.0
## 21            9   September 2011         11388          9.0
## 22           10     October 2011          9892          8.8
## 23           11    November 2011          9225          8.6
## 24           12    December 2011         11415          8.5
## 25            1     January 2012          7896          8.3
## 26            2    February 2012         18638          8.3
## 27            3       March 2012         12557          8.2
## 28            4       April 2012         12184          8.2
## 29            5         May 2012         15454          8.2
## 30            6        June 2012         14461          8.2
## 31            7        July 2012         12216          8.2
## 32            8      August 2012         13293          8.1
## 33            9   September 2012         12097          7.8
## 34           10     October 2012         11310          7.8
## 35           11    November 2012         10337          7.7
## 36           12    December 2012         11545          7.9
## 37            1     January 2013          8854          8.0
## 38            2    February 2013         10091          7.7
## 39            3       March 2013         12901          7.5
## 40            4       April 2013         13445          7.6
## 41            5         May 2013         16272          7.5
## 42            6        June 2013         16165          7.5
## 43            7        July 2013         14404          7.3
## 44            8      August 2013         15825          7.3
## 45            9   September 2013         11984          7.2
## 46           10     October 2013         11780          7.2
## 47           11    November 2013         11753          6.9
## 48           12    December 2013         12028          6.7
## 49            1     January 2014          9553          6.6
```

```
## 50           2    February 2014         10640          6.7
## 51           3       March 2014         14481          6.7
## 52           4       April 2014         15389          6.2
## 53           5         May 2014         19235          6.3
## 54           6        June 2014         16439          6.1
## 55           7        July 2014         16388          6.2
## 56           8      August 2014         17988          6.2
## 57           9   September 2014         13955          5.9
## 58          10     October 2014         13665          5.7
## 59          11    November 2014         13592          5.8
## 60          12    December 2014         14003          5.6
## 61           1     January 2015         11683          5.7
## 62           2    February 2015         12911          5.5
## 63           3       March 2015         17524          5.4
## 64           4       April 2015         18849          5.4
## 65           5         May 2015         22324          5.5
## 66           6        June 2015         19159          5.3
## 67           7        July 2015         19320          5.2
## 68           8      August 2015         18160          5.1
## 69           9   September 2015         17583          5.0
## 70          10     October 2015         15751          5.0
## 71          11    November 2015         13847          5.0
## 72          12    December 2015         15591          5.0
## 73           1     January 2016         10797          4.9
## 74           2    February 2016         13234          4.9
## 75           3       March 2016         17710          5.0
## 76           4       April 2016         19003          5.0
## 77           5         May 2016         19551          4.7
## 78           6        June 2016         20060          4.9
## 79           7        July 2016         18741          4.9
## 80           8      August 2016         15290          4.9
## 81           9   September 2016         14255          4.9
## 82          10     October 2016         14469          4.8
## 83          11    November 2016         12957          4.6
## 84          12    December 2016         15721          4.7
## 85           1     January 2017         11334          4.8
## 86           2    February 2017         13641          4.7
## 87           3       March 2017         16336          4.5
## 88           4       April 2017         18841          4.4
## 89           5         May 2017         19931          4.3
## 90           6        June 2017         18839          4.3
## 91           7        July 2017         18698          4.3
## 92           8      August 2017         16808          4.4
## 93           9   September 2017         15714          4.2
## 94          10     October 2017         13391          4.1
## 95          11    November 2017         13289          4.1
## 96          12    December 2017         13700          4.1
## 97           1     January 2018         11739          4.1
## 98           2    February 2018         15936          4.1
## 99           3       March 2018         27829          4.1
## 100          4       April 2018         29776          3.9
## 101          5         May 2018         25102          3.8
## 102          6        June 2018         23110          4.0
## 103          7        July 2018         21308          3.9
```

```
## 104            8      August 2018         20168          3.8
## 105            9   September 2018         15983          3.7
## 106           10     October 2018         13318          3.8
## 107           11    November 2018         15963          3.7
## 108           12    December 2018         19800          3.9
## 109            1     January 2019         13024          4.0
## 110            2    February 2019         15001          3.8
## 111            3       March 2019         21963          3.8
## 112            4       April 2019         22422          3.6
## 113            5         May 2019         24530          3.6
## 114            6        June 2019         20055          3.7
##     WranglerQueries CPI.All CPI.Energy fordquery
## 1                30 217.488    212.807        19
## 2                33 217.281    209.624        20
## 3                34 217.353    209.326        20
## 4                35 217.403    209.219        19
## 5                36 217.290    206.631        16
## 6                38 217.199    203.764        18
## 7                38 217.605    206.877        23
## 8                38 217.923    208.770        16
## 9                34 218.275    209.832        17
## 10               32 219.035    216.710        17
## 11               37 219.590    219.496        16
## 12               34 220.472    227.130        17
## 13               39 221.187    229.258        21
## 14               40 221.898    232.068        19
## 15               40 223.046    240.079        18
## 16               41 224.093    247.977        20
## 17               43 224.806    250.744        18
## 18               47 224.806    245.534        17
## 19               51 225.395    246.187        18
## 20               51 226.106    246.880        20
## 21               46 226.597    248.550        18
## 22               43 226.750    246.655        17
## 23               44 227.169    247.640        18
## 24               42 227.223    243.353        18
## 25               45 227.842    244.876        21
## 26               45 228.329    248.898        20
## 27               48 228.807    249.742        20
## 28               51 229.187    249.677        18
## 29               53 228.713    241.806        19
## 30               56 228.524    235.897        19
## 31               57 228.590    233.568        18
## 32               56 229.918    244.987        19
## 33               52 231.015    252.987        19
## 34               43 231.638    256.017        17
## 35               43 231.249    248.819        17
## 36               43 231.221    244.708        16
## 37               45 231.679    245.025        17
## 38               49 232.937    255.696        19
## 39               53 232.282    246.595        20
## 40               53 231.797    240.473        20
## 41               59 231.893    240.468        18
## 42               62 232.445    242.711        20
```

```
## 43                   63 232.900    242.986        19
## 44                   59 233.456    244.833        20
## 45                   53 233.544    242.745        18
## 46                   48 233.669    241.954        18
## 47                   48 234.100    242.718        18
## 48                   48 234.719    245.733        31
## 49                   53 235.288    250.340        38
## 50                   58 235.547    249.925        30
## 51                   61 236.028    249.961        28
## 52                   63 236.468    249.864        49
## 53                   68 236.918    249.213        29
## 54                   70 237.231    249.714        29
## 55                   73 237.498    248.744        30
## 56                   68 237.460    245.699        57
## 57                   61 237.477    241.610        39
## 58                   57 237.430    237.061        46
## 59                   55 236.983    229.016        60
## 60                   53 236.252    218.536        45
## 61                   57 234.718    199.471        46
## 62                   63 235.236    202.079        39
## 63                   69 236.005    206.148        38
## 64                   69 236.156    202.898        33
## 65                   76 236.974    209.120        30
## 66                   78 237.684    212.476        29
## 67                   85 238.053    212.324        31
## 68                   83 238.028    208.870        34
## 69                   76 237.506    197.324        31
## 70                   67 237.781    196.014        27
## 71                   62 238.016    194.365        27
## 72                   62 237.817    190.299        28
## 73                   72 237.833    186.122        33
## 74                   78 237.469    176.407        49
## 75                   80 238.038    181.074        39
## 76                   83 238.827    185.405        31
## 77                   87 239.464    188.401        31
## 78                   93 240.167    193.068        37
## 79                   90 240.150    190.089        32
## 80                   86 240.602    189.795        29
## 81                   79 241.051    191.772        31
## 82                   74 241.691    195.824       100
## 83                   71 242.029    195.519        40
## 84                   73 242.772    200.266        42
## 85                   76 243.780    206.048        72
## 86                   79 243.961    203.170        43
## 87                   83 243.749    201.526        41
## 88                   87 244.051    202.399        38
## 89                   87 243.962    198.596        37
## 90                   93 244.182    198.265        37
## 91                   97 244.390    197.349        53
## 92                   89 245.297    202.338        39
## 93                   78 246.418    211.137        34
## 94                   77 246.587    207.771        37
## 95                   82 247.332    213.134        31
## 96                   81 247.901    214.055        33
```

```
## 97                  77 248.884    217.542        41
## 98                  83 249.369    218.955        38
## 99                  85 249.498    215.801        48
## 100                 89 249.956    217.690        51
## 101                 96 250.646    220.967        40
## 102                 96 251.134    222.361        44
## 103                100 251.597    222.269        49
## 104                 92 251.879    223.341        47
## 105                 83 252.010    221.077        42
## 106                 76 252.794    225.612        43
## 107                 76 252.760    219.295        45
## 108                 76 252.723    213.565        52
## 109                 77 252.673    206.842        58
## 110                 82 253.113    207.755        53
## 111                 86 254.148    214.963        55
## 112                 89 254.958    221.286        51
## 113                 89 255.155    219.937        51
## 114                 96 255.305    214.847        52
```

The training set r^2 is .7943, the test set data has an r^2 of .63. Based on the r squared value of our model, I do not think it will provide much to Jeep, considering that the r squared value is very low. Maybe a linear model is not a great fit for this dataset and I think that we can increase the r squared value if we were to have a less granulated coefficient array for the season, instead of it being one month it should be a collection of months that represent a season.

d) I would maybe look at the search queries for a competing model to the jeep wrangler, one that I looked at was the ford bronco. I would suspect that queries for the bronco are inversely correlated with sales of the jeep, rationale being that if more people are looking up information on the bronco and are interested in buying the bronco, that means less individuals are interested in competing brands or models, in this case the Jeep Wrangler.

```r
wranglerSales.bronco <- lm(WranglerSales ~ Unemployment + WranglerQueries + CPI.All + MonthFactor + ford

z = predict(wranglerSales.bronco, wrangler.test)

rsq(z,test)
```

```
## [1] 0.6419495
```

```r
summary(wranglerSales.bronco)
```

```
##
## Call:
## lm(formula = WranglerSales ~ Unemployment + WranglerQueries +
##     CPI.All + MonthFactor + fordquery, data = wrangler.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3025.9  -692.2   -95.6   591.0  8228.0
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)          -27578.393  21540.812  -1.280  0.20572
## Unemployment            166.623    579.460   0.288  0.77475
## WranglerQueries         194.629     62.570   3.111  0.00294 **
## CPI.All                 125.746     83.637   1.503  0.13834
## MonthFactorAugust       -95.256    957.221  -0.100  0.92109
## MonthFactorDecember      69.808   1082.455   0.064  0.94881
## MonthFactorFebruary   -1033.651    917.612  -1.126  0.26478
## MonthFactorJanuary    -3149.789    956.204  -3.294  0.00172 **
## MonthFactorJuly        -248.636   1006.430  -0.247  0.80577
## MonthFactorJune         268.853    953.298   0.282  0.77896
## MonthFactorMarch       -154.426    896.537  -0.172  0.86386
## MonthFactorMay         1885.607    919.263   2.051  0.04493 *
## MonthFactorNovember   -1418.875   1011.623  -1.403  0.16626
## MonthFactorOctober     -584.155   1016.934  -0.574  0.56798
## MonthFactorSeptember   -852.139    905.426  -0.941  0.35067
## fordquery                 8.465     28.757   0.294  0.76957
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1545 on 56 degrees of freedom
## Multiple R-squared:  0.8682, Adjusted R-squared:  0.8329
## F-statistic:  24.6 on 15 and 56 DF,  p-value: < 2.2e-16
```

The resulting r squared value is .6419, which means it has increased and has thus added some predictive value. Looking at the table however, there is a very high p value associated with the ford query, indicating it is not as significant of a variable as we may think. I think that ultimately because there are so many other options other than ford bronco for a substitute, this does not help our model, if we were to replace this with queries for any other competing model to the wrangler it might make it more accurate.