Regression - Neo Zhao - CS4375

Linear Regression

- In Linear regression, we explore our data to find if there's a relationship between x and y. The regression estimates explain the relationship between one dependent variable and one or more independent variables. Some advantages include simple implementation as well as as the regularization of overfitted data. Some disadvantages include heavy sensitivity to outliers and likely to underfit some data.

```
library(rlang)
```

```
## Warning: package 'rlang' was built under R version 4.1.3
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
##
## 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
```

```
library(ggplot2)

# Source: https://www.kaggle.com/datasets/neuromusic/avocado-prices

Avo <- read.csv("avocado.csv")

# Grab month from date
AvoDates <- as.Date(Avo$Date, format = "%Y-%m-%d")
avoMonth <- format(AvoDates, "%m")

# Replace old date column with new numerical month column
Avo$Date <- avoMonth
```

A. Divide into 80/20 train/test

```
# Set a seed for reproducibility
set.seed(1)
i <- sample(1:nrow(Avo), nrow(Avo) * 0.8, replace = FALSE)
train <- Avo[i,]
test <- Avo[-i,]
```

B. Data Exploration

```
# 1) Summary
summary(train)
```

```
##        X             Date            AveragePrice    Total.Volume
##  Min.   : 0.00   Length:14599       Min.   :0.440   Min.   :      85
##  1st Qu.:10.00   Class :character   1st Qu.:1.100   1st Qu.:   10999
##  Median :24.00   Mode  :character   Median :1.370   Median :  109421
##  Mean   :24.34                      Mean   :1.405   Mean   :  847440
##  3rd Qu.:38.00                      3rd Qu.:1.660   3rd Qu.:  433038
##  Max.   :52.00                      Max.   :3.170   Max.   :62505647
##      X4046              X4225              X4770            Total.Bags
##  Min.   :       0   Min.   :       0   Min.   :      0.0   Min.   :       0
##  1st Qu.:     861   1st Qu.:    3099   1st Qu.:      0.0   1st Qu.:    5132
##  Median :    8787   Median :   29197   Median :    185.4   Median :   40600
##  Mean   :  291922   Mean   :  294671   Mean   :  22655.5   Mean   :  238189
##  3rd Qu.:  110964   3rd Qu.:  151172   3rd Qu.:   6140.5   3rd Qu.:  111728
##  Max.   :21620181   Max.   :20470573   Max.   :2546439.1   Max.   :19373134
##    Small.Bags         Large.Bags       XLarge.Bags           type
##  Min.   :       0   Min.   :      0   Min.   :     0.0   Length:14599
##  1st Qu.:    2871   1st Qu.:    126   1st Qu.:     0.0   Class :character
##  Median :   26941   Median :   2699   Median :     0.0   Mode  :character
##  Mean   :  180946   Mean   :  54093   Mean   :  3148.9
##  3rd Qu.:   84182   3rd Qu.:  22052   3rd Qu.:   122.8
##  Max.   :13384587   Max.   :5719097   Max.   :551693.7
##       year         region
##  Min.   :2015   Length:14599
##  1st Qu.:2015   Class :character
##  Median :2016   Mode  :character
##  Mean   :2016
##  3rd Qu.:2017
##  Max.   :2018
```

```
# 2) Find # of missing values
colSums(is.na(train))
```

```
##            X         Date AveragePrice Total.Volume         X4046         X4225
##            0            0            0            0            0             0
##        X4770   Total.Bags   Small.Bags   Large.Bags  XLarge.Bags          type
##            0            0            0            0            0             0
##         year       region
##            0            0
```

```r
# 3) str() Function
str(train)
```

```
## 'data.frame':    14599 obs. of  14 variables:
##  $ X           : int  9 42 36 8 36 45 5 0 11 11 ...
##  $ Date        : chr  "10" "03" "04" "11" ...
##  $ AveragePrice: num  1.94 1.08 1.57 1.67 0.75 0.86 1.81 1.59 1.98 1.95 ...
##  $ Total.Volume: num  6320 298088 2549 1713 1108329 ...
##  $ X4046       : num  92.7 66431.9 1072.9 334.8 583229.9 ...
##  $ X4225       : num  741.8 140242.5 11.6 94.7 105373.5 ...
##  $ X4770       : num  0 22198 0 0 13999 ...
##  $ Total.Bags  : num  5486 69215 1464 1284 405726 ...
##  $ Small.Bags  : num  4392 68589 1464 1280 118216 ...
##  $ Large.Bags  : num  1.09e+03 6.26e+02 0.00 3.33 2.88e+05 ...
##  $ XLarge.Bags : num  0 0 0 0 0 ...
##  $ type        : chr  "organic" "conventional" "organic" "organic" ...
##  $ year        : int  2017 2016 2016 2015 2017 2016 2016 2015 2016 2018 ...
##  $ region      : chr  "Tampa" "RaleighGreensboro" "MiamiFtLauderdale" "NewOrleansMobile" ...
```

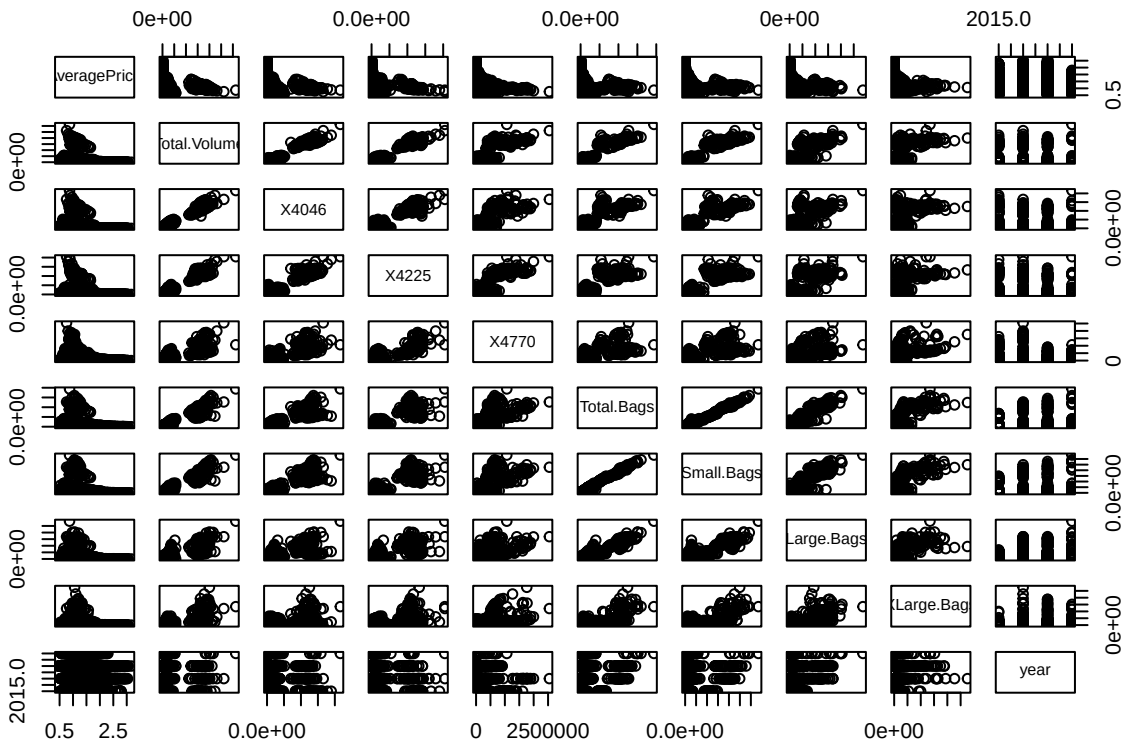
```r
# 4) names() Function
names(train)
```

```
##  [1] "X"            "Date"         "AveragePrice" "Total.Volume" "X4046"
##  [6] "X4225"        "X4770"        "Total.Bags"   "Small.Bags"   "Large.Bags"
## [11] "XLarge.Bags"  "type"         "year"         "region"
```

```r
# 5) cor() and pairs()
cor(train[,c(3:11,13)])
```

```
##               AveragePrice Total.Volume       X4046       X4225       X4770
## AveragePrice    1.00000000  -0.19378494 -0.20895817 -0.17381555 -0.17910214
## Total.Volume   -0.19378494   1.00000000  0.97678676  0.97306991  0.86714097
## X4046          -0.20895817   0.97678676  1.00000000  0.92261434  0.82614843
## X4225          -0.17381555   0.97306991  0.92261434  1.00000000  0.88228709
## X4770          -0.17910214   0.86714097  0.82614843  0.88228709  1.00000000
## Total.Bags     -0.17785748   0.96143922  0.91651988  0.90196106  0.78591456
## Small.Bags     -0.17552131   0.96553563  0.92187998  0.91200082  0.79624078
## Large.Bags     -0.17323628   0.87676019  0.83247463  0.80488870  0.69028285
## XLarge.Bags    -0.11519219   0.74336230  0.69412198  0.68212040  0.67812025
## year            0.08962285   0.01580143  0.00164992 -0.01143373 -0.03750688
##                Total.Bags  Small.Bags  Large.Bags XLarge.Bags        year
## AveragePrice -0.17785748 -0.17552131 -0.17323628 -0.11519219  0.08962285
## Total.Volume  0.96143922  0.96553563  0.87676019  0.74336230  0.01580143
## X4046         0.91651988  0.92187998  0.83247463  0.69412198  0.00164992
## X4225         0.90196106  0.91200082  0.80488870  0.68212040 -0.01143373
## X4770         0.78591456  0.79624078  0.69028285  0.67812025 -0.03750688
## Total.Bags    1.00000000  0.99400686  0.94077125  0.80281703  0.07152248
## Small.Bags    0.99400686  1.00000000  0.89838807  0.80523458  0.06405358
## Large.Bags    0.94077125  0.89838807  1.00000000  0.70676720  0.08688976
## XLarge.Bags   0.80281703  0.80523458  0.70676720  1.00000000  0.08047646
## year          0.07152248  0.06405358  0.08688976  0.08047646  1.00000000
```

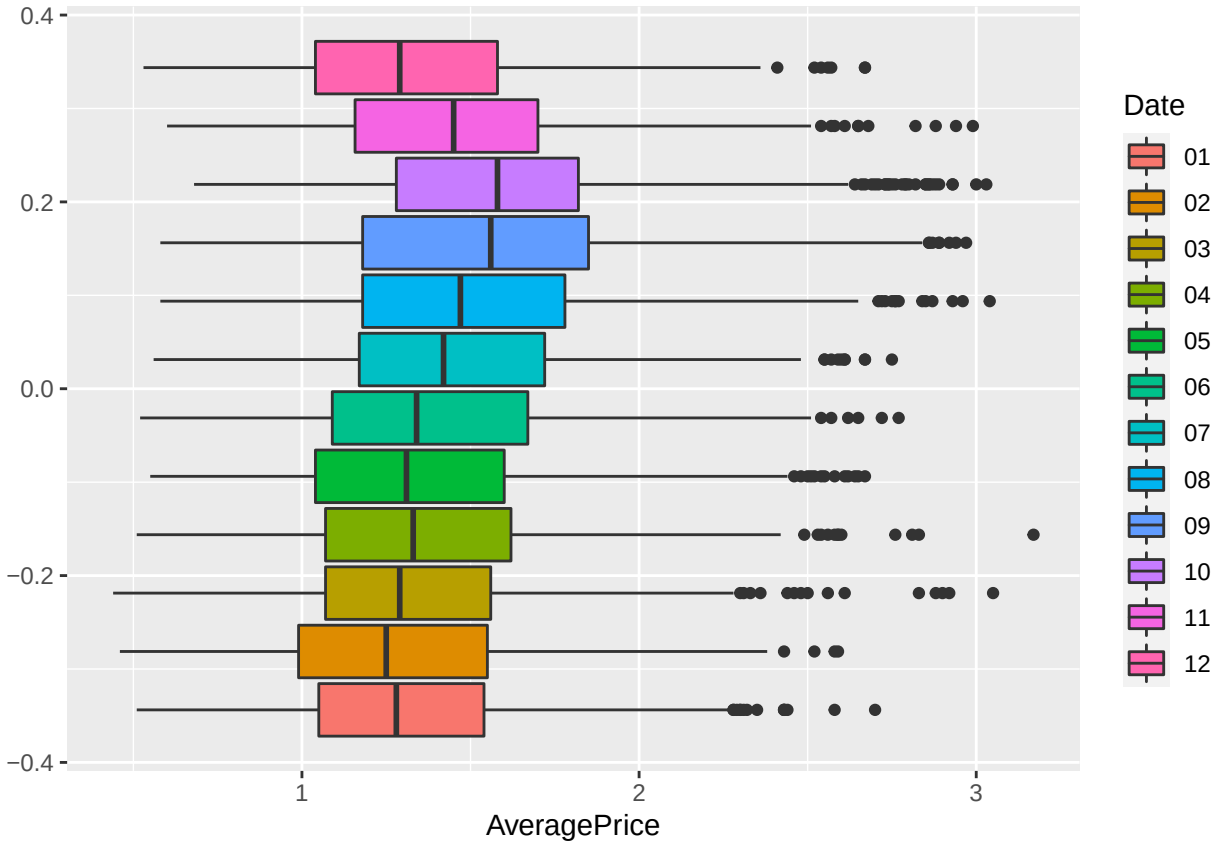
```
pairs(train[,c(3:11,13)])
```



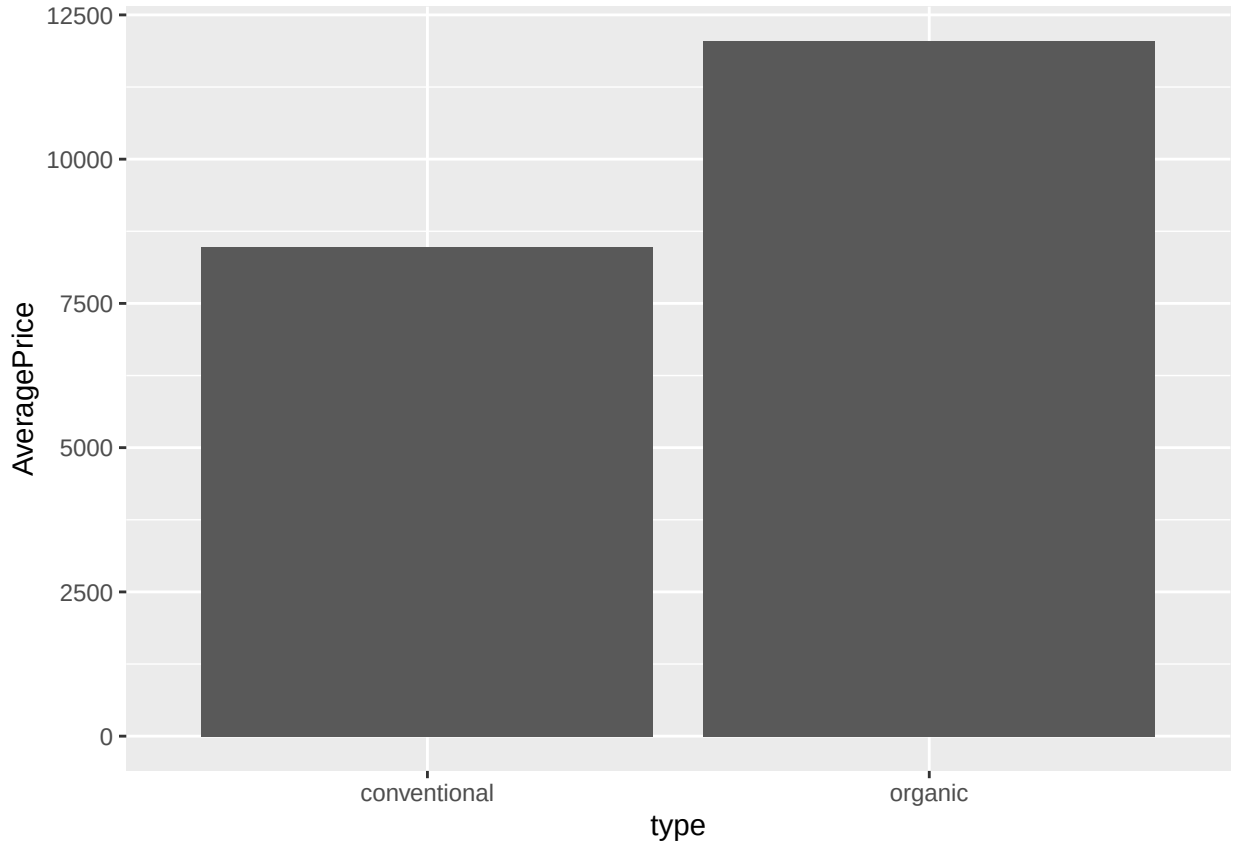
C. Informative graphs

```
# Plots

# Boxplot with ggplot
ggplot(train, aes(x = AveragePrice,
       fill = Date)) + geom_boxplot()
```




```
# Comparing types with Average Price
ggplot(data = train, aes(x = type, y = AveragePrice)) +
  geom_bar(stat = "identity")
```



D. 1) Simple Linear Regression Model + Summary

```
# First Linear Model - y = Total Volume, x = Average Price
lm1 <- lm(Total.Volume ~ AveragePrice, data = train)

# Summary
summary(lm1)
```

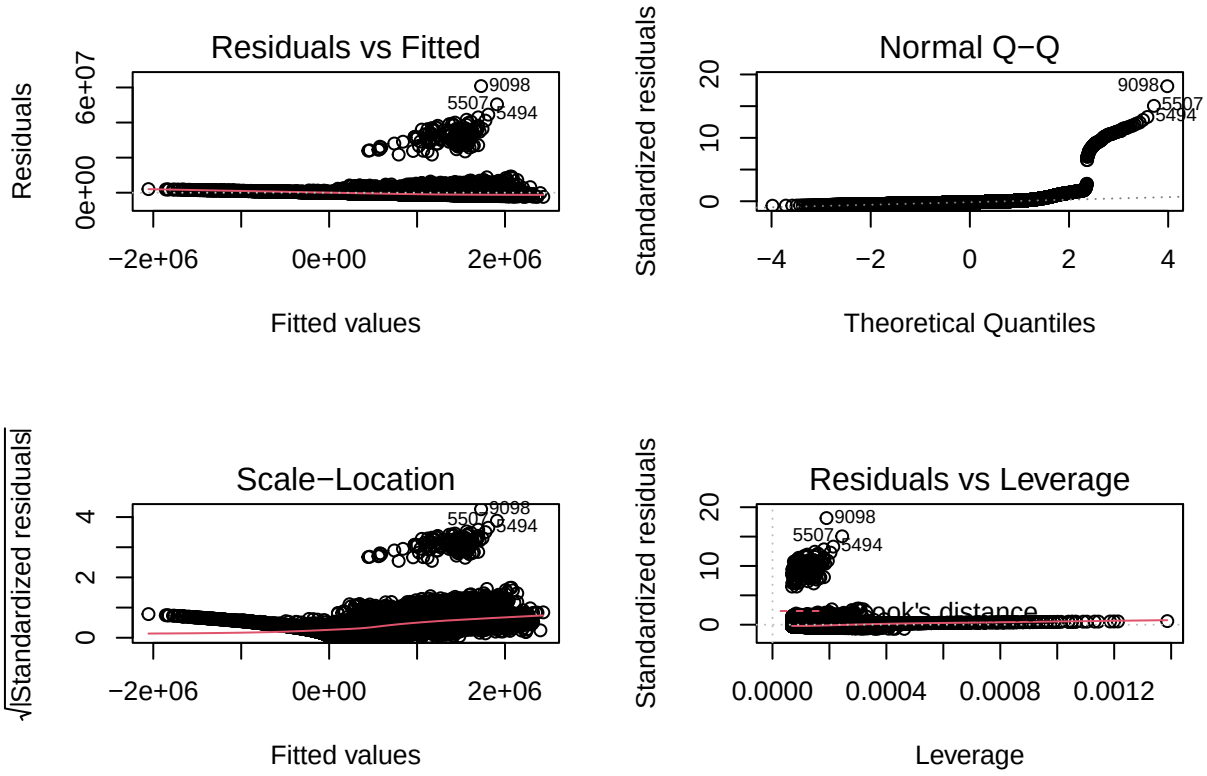
```
##
## Call:
## lm(formula = Total.Volume ~ AveragePrice, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2370951  -953630  -570796   -48655 60777796
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3158611     100733   31.36   <2e-16 ***
## AveragePrice -1644552      68910  -23.86   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3350000 on 14597 degrees of freedom
```

```
## Multiple R-squared:  0.03755,    Adjusted R-squared:  0.03749
## F-statistic: 569.5 on 1 and 14597 DF,  p-value: < 2.2e-16
```

E. Plot Residuals + Summary

- In the Residuals vs. Fitted plot, we can see there's a line which most of the model follows; however, there is a clump of outliers and cases that do not follow the line.
- In the Normal Q-Q plot... it definitely looks a little concerning where a lot of the cases do not follow the line. They are most likely the same clump that didn't follow in the Residuals vs. Fitted plot.
- In the Scale-Location plot, the residuals appear fairly random; however, somewhere between 0e+00 and 1e+06, the line becomes just slightly steeper.
- In the Residuals vs. Leverage plot, we have a rather straight red line and Cook's distance lines are not very present. We again, have the clump of outliers showing here again.

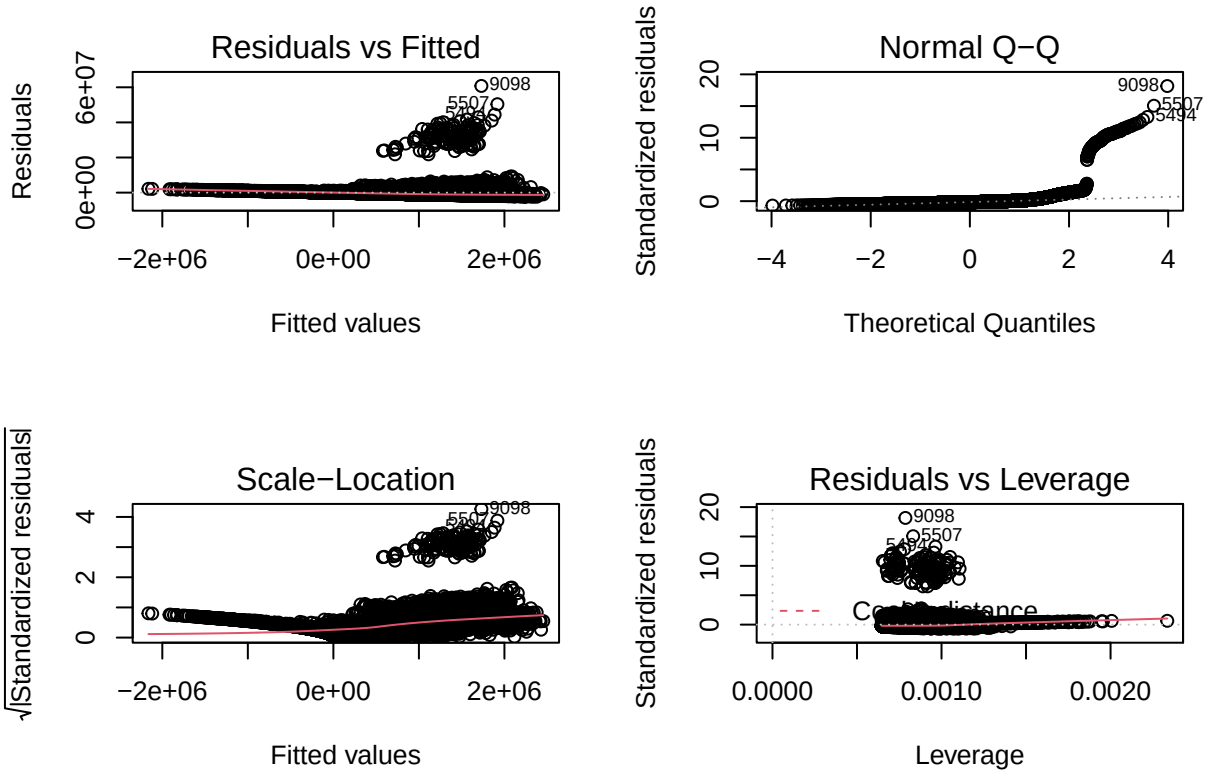
```
par(mfrow = c(2,2))
plot(lm1)
```



F. 2) Multiple Linear Regression Model + Residual Plots + Summary

```
# Second (Multiple) Linear Model - y = Total Volume, x = Average Price & Date
lm2 <- lm(Total.Volume ~ AveragePrice + Date, data = train)
```

```r
# Residual Plots
par(mfrow = c(2,2))
plot(lm2)
```




```r
# Summary
summary(lm2)
```

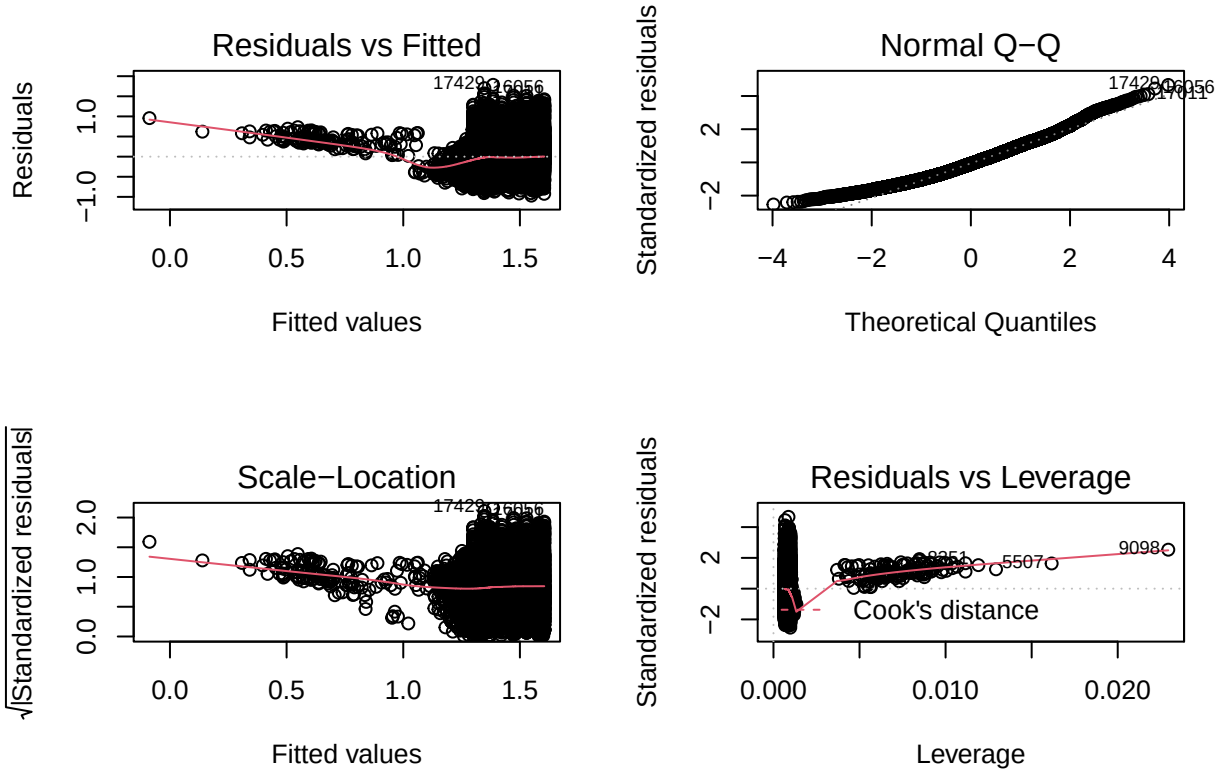
```
##
## Call:
## lm(formula = Total.Volume ~ AveragePrice + Date, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2333149  -953496  -558009   -28839 60774910
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3079402     126435  24.356   <2e-16 ***
## AveragePrice -1701150      71088 -23.930   <2e-16 ***
## Date02         131335     123461   1.064   0.2874
## Date03          -9217     121683  -0.076   0.9396
## Date04         151186     130357   1.160   0.2462
## Date05         201107     129073   1.558   0.1192
## Date06         259437     134677   1.926   0.0541 .
## Date07         296963     129253   2.298   0.0216 *
```

```
## Date08          266424     132522   2.010   0.0444 *
## Date09          317187     135967   2.333   0.0197 *
## Date10          307195     129663   2.369   0.0178 *
## Date11           91203     131665   0.693   0.4885
## Date12            7685     132097   0.058   0.9536
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3349000 on 14586 degrees of freedom
## Multiple R-squared:  0.03876,    Adjusted R-squared:  0.03797
## F-statistic: 49.01 on 12 and 14586 DF,  p-value: < 2.2e-16
```

G. 3) Third Linear Regression Model With Different Combination of Predictors + Residual Plots + Summary

```
# Third Linear Model - y = Average Price, x = Total Volume & Date
lm3 <- lm(AveragePrice ~ Total.Volume + Date, data = train)

# Residual Plots
par(mfrow = c(2,2))
plot(lm3)
```



```r
# Summary
summary(lm3)
```

```
##
## Call:
## lm(formula = AveragePrice ~ Total.Volume + Date, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.96911 -0.28933 -0.03298  0.24680  1.78490
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.333e+00  9.762e-03 136.567  < 2e-16 ***
## Total.Volume -2.221e-08  9.280e-10 -23.930  < 2e-16 ***
## Date02       -3.300e-02  1.410e-02  -2.339 0.019325 *
## Date03        1.469e-02  1.390e-02   1.057 0.290752
## Date04        5.195e-02  1.489e-02   3.489 0.000486 ***
## Date05        3.715e-02  1.475e-02   2.520 0.011761 *
## Date06        8.948e-02  1.537e-02   5.821 5.96e-09 ***
## Date07        1.516e-01  1.472e-02  10.304  < 2e-16 ***
## Date08        1.954e-01  1.506e-02  12.979  < 2e-16 ***
## Date09        2.468e-01  1.540e-02  16.021  < 2e-16 ***
## Date10        2.712e-01  1.465e-02  18.518  < 2e-16 ***
## Date11        1.404e-01  1.500e-02   9.360  < 2e-16 ***
## Date12        1.286e-02  1.509e-02   0.852 0.394312
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3826 on 14586 degrees of freedom
## Multiple R-squared:  0.09629,    Adjusted R-squared:  0.09554
## F-statistic: 129.5 on 12 and 14586 DF,  p-value: < 2.2e-16
```

H. Compare Results + Summary

- I believe Model 3 shows the best comparison; however, overall I don't think the data set is very good with further cleaning and inspection of the "clumps" of outliers that are present across the models. Over all the models, they're a little concerning; however, our third model has the best looking Normal Q-Q plot.

I. Predict and Evalute the Test Data using Metrics Correlation & MSE + Compare

- For MSE, you can see that lm1 and lm2 have quite large MSEs while lm3 is very close to 0. In this case, I believe lm3, which compares Average Price to Total Volume is the best model.
- Our target values are not the same as we use different predictors each time
- All of the values in Model 1 and 2 are quite large. Model 3 is the best.

```r
# Predict

# First Model
```

```
pred1 <- predict(lm1, newdata = test)
cor1 <- cor(pred1, test$Total.Volume)
mse1 <- mean((pred1 - test$Total.Volume)^2)
rmse1 <- sqrt(mse1)

head(pred1)
```

```
##         1         5        11        12        17        23
##  971356.2 1053583.8 1316712.1 1053583.8 1333157.7 1333157.7
```

```
cor1
```

```
## [1] 0.1890867
```

```
mse1
```

```
## [1] 1.254199e+13
```

```
rmse1
```

```
## [1] 3541467
```

```
# Second Model
pred2 <- predict(lm2, newdata = test)
cor2 <- cor(pred2, test$Total.Volume)
mse2 <- mean((pred2 - test$Total.Volume)^2)
rmse2 <- sqrt(mse2)

head(pred2)
```

```
##         1         5        11        12        17        23
##  824558.1  993133.5 1481309.1 1209125.2 1508312.8 1488088.8
```

```
cor2
```

```
## [1] 0.1886567
```

```
mse2
```

```
## [1] 1.25438e+13
```

```
rmse2
```

```
## [1] 3541723
```

```
# Third Model
pred3 <- predict(lm3, newdata = test)
cor3 <- cor(pred3, test$AveragePrice)
mse3 <- mean((pred3 - test$AveragePrice)^2)
rmse3 <- sqrt(mse3)

head(pred3)
```

```
##        1        5       11       12       17       23
## 1.344645 1.472469 1.602547 1.602999 1.577788 1.482484
```

```
cor3
```

```
## [1] 0.3000805
```

```
mse3
```

```
## [1] 0.1486408
```

```
rmse3
```

```
## [1] 0.3855397
```