Regression

Benton Fariss

2023-02-17

#Dataset from: https://www.kaggle.com/datasets/neuromusic/avocado-prices?resource=download ### How does linear regression work? Linear regression models the relationship between a dependent variable and 1+ independent variables. It is a great model due to the simplistic nature of it. It is an efficient method that is very easy for users to read, comprehend, and interpret. The only problem with linear regression is that if the relationship is not linear between the variables. This can cause very powerful outliers in the data which must be dealt with properly by the user. #load the data

```
avocados <- read.csv("C:/Users/setup/Downloads/archive (1)/avocado.csv")
set.seed(1234)
```

A. Split the Data into 80 train and 20 test

```
i <- sample(1:nrow(avocados), nrow(avocados)*.8, replace=FALSE)
train <- avocados[i,]
test <- avocados[-i,]
```

B. Use 5 R functions for data exploration

This displays the structures of objects.

```
str(train)
```

```
## 'data.frame':    14599 obs. of  14 variables:
##  $ X           : int  33 14 8 31 17 50 22 42 49 26 ...
##  $ Date        : chr  "2017-05-14" "2017-09-24" "2017-11-05" "2017-05-28" ...
##  $ AveragePrice: num  1.33 1.33 1.45 1.38 1.53 0.76 1.85 1.52 1.39 1.38 ...
##  $ Total.Volume: num  117858 4405344 3492828 3916909 9298 ...
##  $ X4046       : num  51069 2350943 228843 2252110 2350 ...
##  $ X4225       : num  20280 750402 1955087 450910 3534 ...
##  $ X4770       : num  1751 6456 5588 5062 0 ...
##  $ Total.Bags  : num  44757 1297543 1303310 1208826 3415 ...
##  $ Small.Bags  : num  41342 965685 1078992 789660 1578 ...
##  $ Large.Bags  : num  3087 331517 224224 371173 1837 ...
##  $ XLarge.Bags : num  329 341 94 47994 0 ...
##  $ type        : chr  "conventional" "conventional" "conventional" "conventional" ...
##  $ year        : int  2017 2017 2017 2017 2015 2015 2017 2015 2015 2016 ...
##  $ region      : chr  "Pittsburgh" "SouthCentral" "Northeast" "Southeast" ...
```

This displays the first parts of the train dataset.

```
head(train)
```

```
##       X       Date AveragePrice Total.Volume      X4046      X4225    X4770
## 7452 33 2017-05-14         1.33    117857.60   51068.92   20279.94  1751.27
## 8016 14 2017-09-24         1.33   4405343.75 2350943.03  750401.85  6455.71
```

```
## 7162  8 2017-11-05         1.45   3492828.10  228843.35 1955086.76  5588.32
## 8086 31 2017-05-28         1.38   3916908.69 2252109.64  450910.10  5062.46
## 9196 17 2015-08-30         1.53      9298.19    2349.53    3534.12     0.00
## 623  50 2015-01-11         0.76   1128693.04  680572.11  348535.22 11900.83
##      Total.Bags Small.Bags Large.Bags XLarge.Bags         type year
## 7452   44757.47   41341.77    3086.53      329.17 conventional 2017
## 8016 1297543.16  965684.93  331517.39      340.84 conventional 2017
## 7162 1303309.67 1078991.73  224223.95       93.99 conventional 2017
## 8086 1208826.49  789659.56  371172.96    47993.97 conventional 2017
## 9196    3414.54    1577.97    1836.57        0.00      organic 2015
## 623    87684.88   67857.83   19801.95       25.10 conventional 2015
##            region
## 7452    Pittsburgh
## 8016  SouthCentral
## 7162     Northeast
## 8086     Southeast
## 9196       Atlanta
## 623  DallasFtWorth
```

This displays the last parts of the train dataset.

```
tail(train)
```

```
##        X       Date AveragePrice Total.Volume     X4046     X4225    X4770
## 15765 16 2017-09-10         1.69      8687.85    136.58   1330.54     0.00
## 8735   4 2018-02-25         1.05    352533.59  76253.45 109855.16  6642.99
## 7534   9 2017-10-29         1.29    520629.70 152604.62 128666.04 15431.79
## 1579  18 2015-08-23         1.18    400101.77   2487.19 338130.60   608.29
## 17923  9 2018-01-21         1.68      8546.30     59.53   2081.22     0.00
## 14089 23 2016-07-17         1.91     14864.69   1579.48  10303.62     0.00
##       Total.Bags Small.Bags Large.Bags XLarge.Bags         type year
## 15765    7212.80    4741.31    2471.49        0.00      organic 2017
## 8735   159781.99   76321.56   82977.10      483.33 conventional 2018
## 7534   223927.25  175517.41   47898.67      511.17 conventional 2017
## 1579    58875.69   58875.69       0.00        0.00 conventional 2015
## 17923    6405.55    2314.95    4090.60        0.00      organic 2018
## 14089    2981.59    2981.59       0.00        0.00      organic 2016
##                  region
## 15765       Indianapolis
## 8735            LasVegas
## 7534            Portland
## 1579  NorthernNewEngland
## 17923          Nashville
## 14089           SanDiego
```

This displays the names of the objects in the train dataset.

```
names(train)
```

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

This displays the dimensions of the train dataset.

```
dim(train)
```

```
## [1] 14599    14
```

This displays the number of rows in the train dataset.

```
nrow(train)
```

```
## [1] 14599
```

This displays the number of columns in the train dataset.

```
ncol(train)
```

```
## [1] 14
```

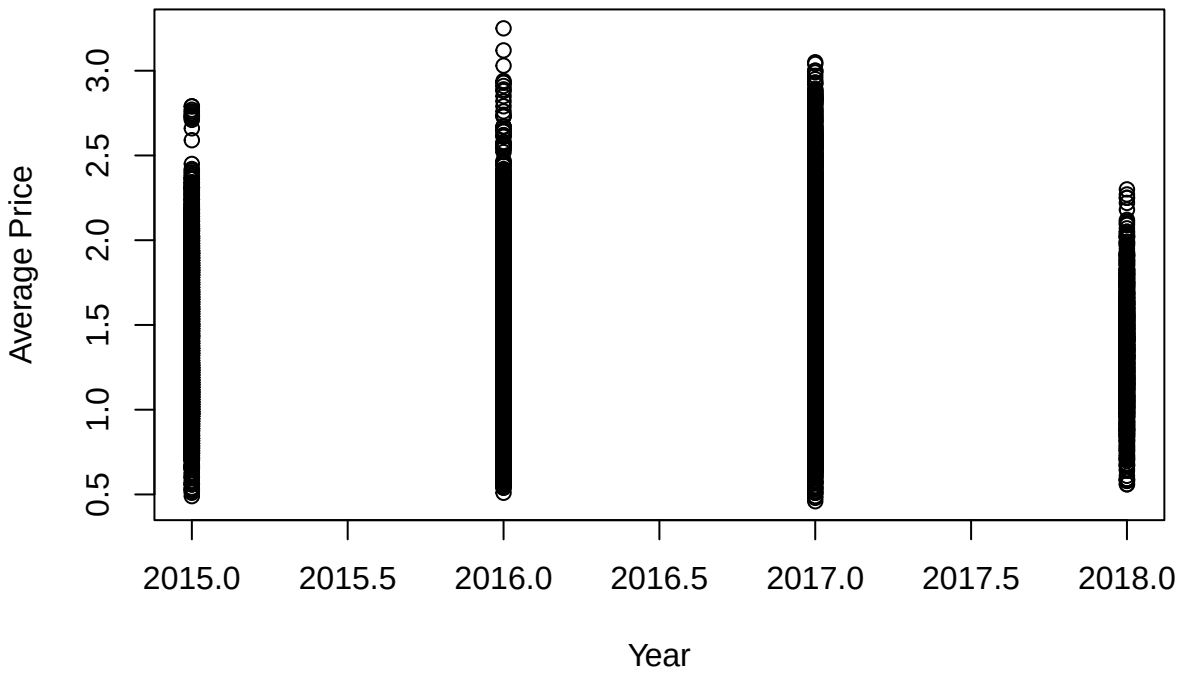
This displays the summaries of all of the values data frames.

```
summary(train)
```

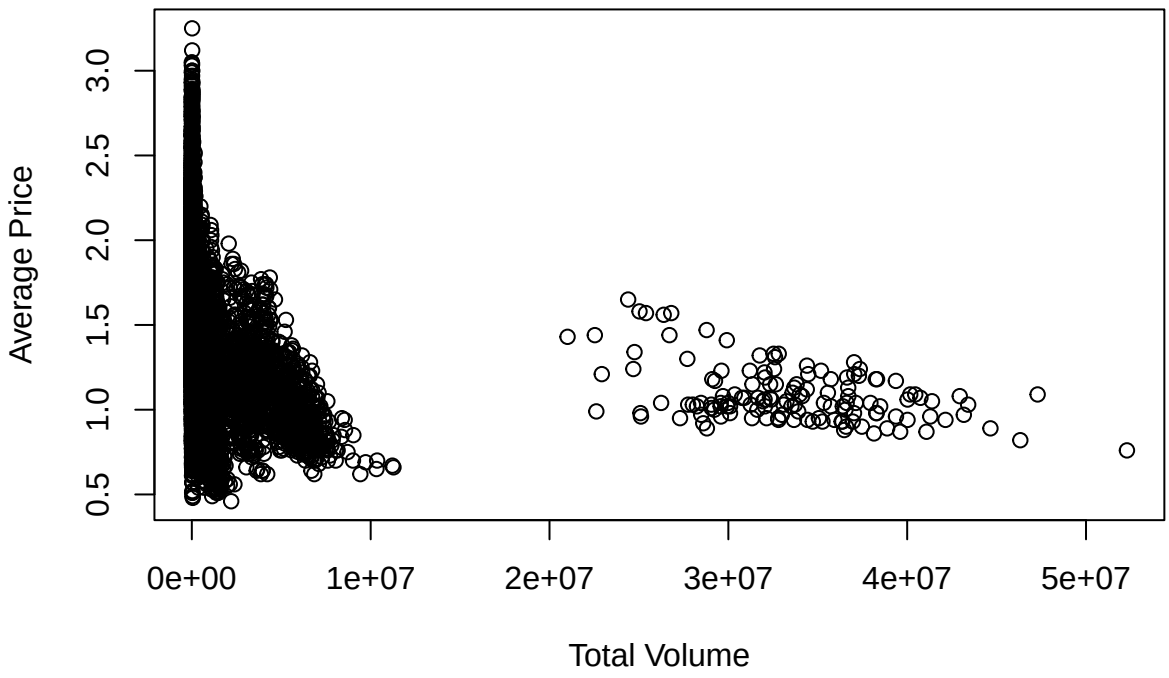
```
##        X            Date            AveragePrice    Total.Volume
##  Min.   : 0.0   Length:14599       Min.   :0.460   Min.   :     386
##  1st Qu.:10.0   Class :character   1st Qu.:1.100   1st Qu.:   10747
##  Median :24.0   Mode  :character   Median :1.370   Median :  106515
##  Mean   :24.3                      Mean   :1.407   Mean   :  852280
##  3rd Qu.:38.0                      3rd Qu.:1.660   3rd Qu.:  434161
##  Max.   :52.0                      Max.   :3.250   Max.   :52288698
##      X4046              X4225              X4770            Total.Bags
##  Min.   :       0   Min.   :       0   Min.   :      0.0   Min.   :       0
##  1st Qu.:     853   1st Qu.:    2981   1st Qu.:      0.0   1st Qu.:    5011
##  Median :    8736   Median :   28735   Median :    186.6   Median :   39676
##  Mean   :  292516   Mean   :  296887   Mean   :  23104.6   Mean   :  239770
##  3rd Qu.:  110450   3rd Qu.:  152261   3rd Qu.:   6261.2   3rd Qu.:  111540
##  Max.   :18933038   Max.   :20470573   Max.   :2546439.1   Max.   :16394524
##    Small.Bags         Large.Bags       XLarge.Bags           type
##  Min.   :       0   Min.   :      0   Min.   :     0.0   Length:14599
##  1st Qu.:    2797   1st Qu.:    126   1st Qu.:     0.0   Class :character
##  Median :   26432   Median :   2642   Median :     0.0   Mode  :character
##  Mean   :  182524   Mean   :  54185   Mean   :  3060.6
##  3rd Qu.:   83749   3rd Qu.:  21861   3rd Qu.:   132.7
##  Max.   :12540327   Max.   :4324231   Max.   :454343.7
##       year         region
##  Min.   :2015   Length:14599
##  1st Qu.:2015   Class :character
##  Median :2016   Mode  :character
##  Mean   :2016
##  3rd Qu.:2017
##  Max.   :2018
```

C. Create 2 informative graphs using the training data

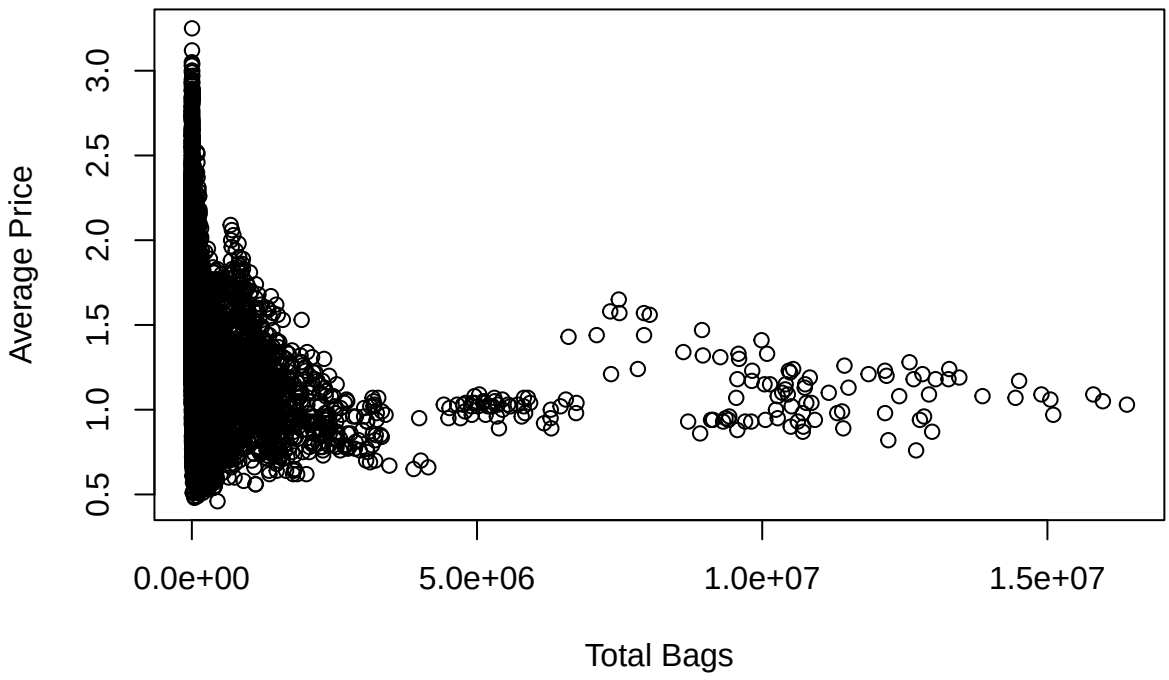
```
plot(train$year, train$AveragePrice, xlab="Year", ylab="Average Price")
```




```
plot(train$Total.Volume, train$AveragePrice, xlab="Total Volume", ylab="Average Price")
```




```
plot(train$Total.Bags, train$AveragePrice, xlab="Total Bags", ylab="Average Price")
```



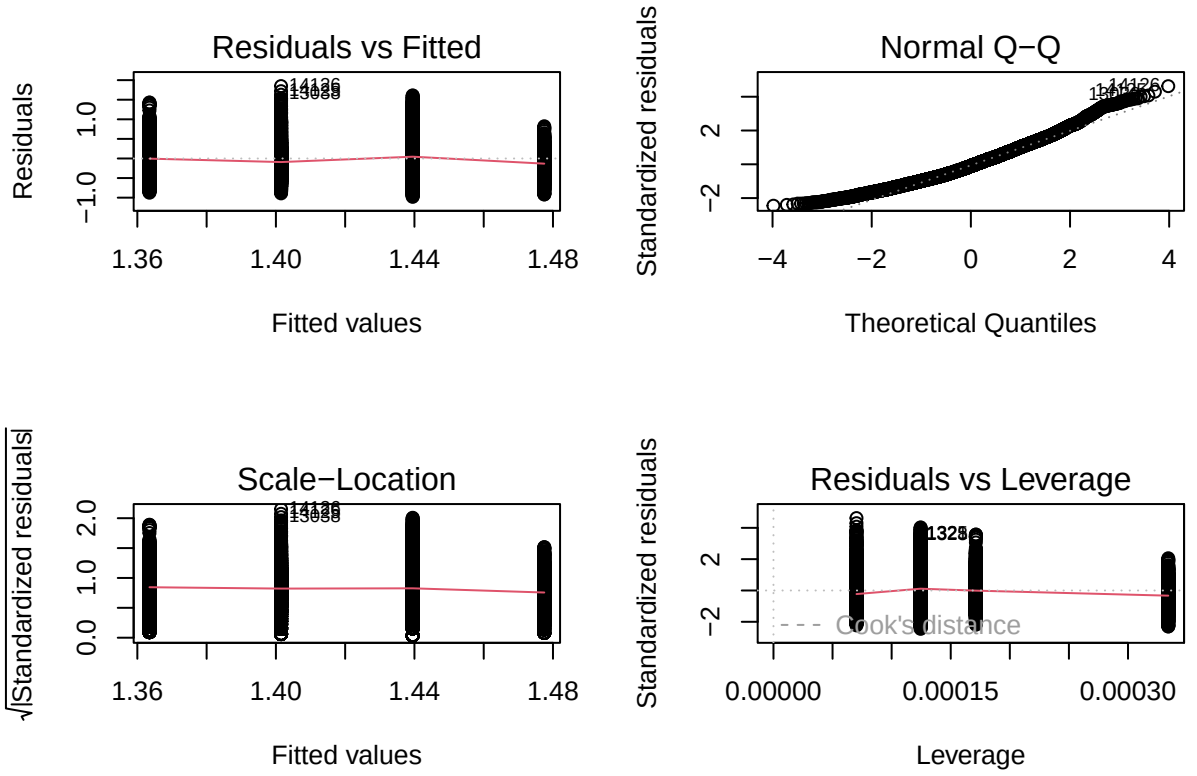
D. Build a simple linear regression model

```
lm1 <- lm(AveragePrice~year, data=train)
summary(lm1)
```

```
##
## Call:
## lm(formula = AveragePrice ~ year, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.97949 -0.29949 -0.03949  0.25151  1.84851
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -75.198198   7.101520  -10.59   <2e-16 ***
## year          0.037996   0.003522   10.79   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3997 on 14597 degrees of freedom
## Multiple R-squared:  0.007909,   Adjusted R-squared:  0.007841
## F-statistic: 116.4 on 1 and 14597 DF,  p-value: < 2.2e-16
```

I built a linear regression model to see how the years have effected the average avocado price. Given these results, they show us that there is very little correlation between the year and the average avocado price. Which a larger dataset that includes more years, I believe that this correlation would be much stronger.
E. Plot the residuals and explain.

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



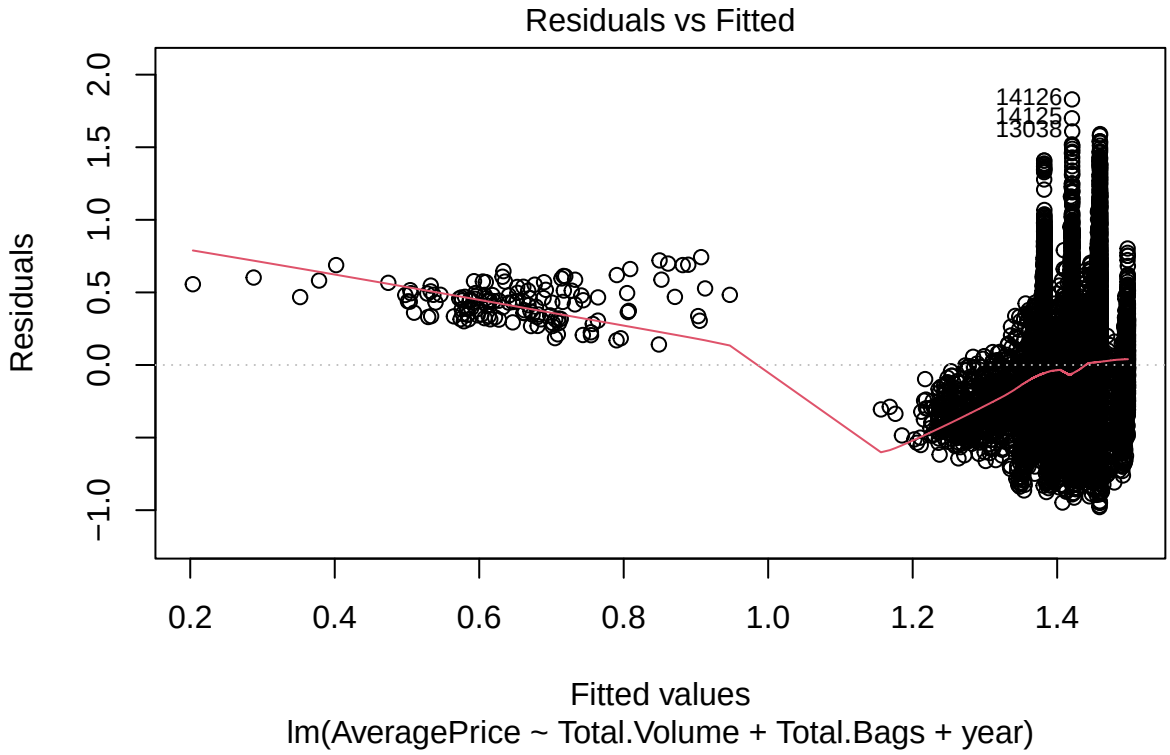
The residuals vs fitted value plot tells you whether or not there is a linear relationship. Since they are pretty spread out, that means there is no non-linear relationship however the relationship is very weak. The Q-Q Distribution plot shows that there is a normal distribution, concluding that the residuals are normally distributed. The Scale-Location plot shows us a horizontal line which means that the residuals are spread out equally. The Residuals vs Leverage plot is spread evenly and does not have any single outliers from the plot. The line is very low on the graph which is a positive in terms of significant outliers. If there were more years in the dataset, most likely there would be economic crashes that would act as outliers. ### F. Build a multiple linear regression model, output the summary and residual plots.

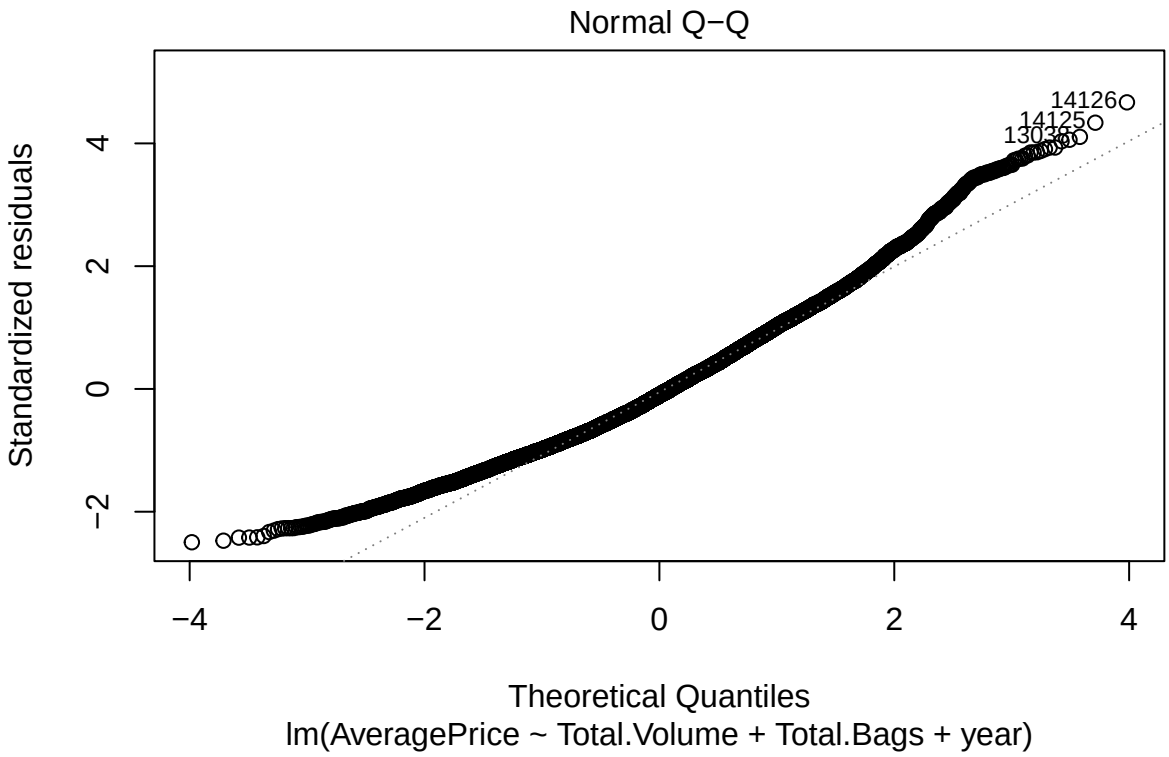
```
lm2 <- lm(AveragePrice~Total.Volume+Total.Bags+year, data=train)
summary(lm2)
```

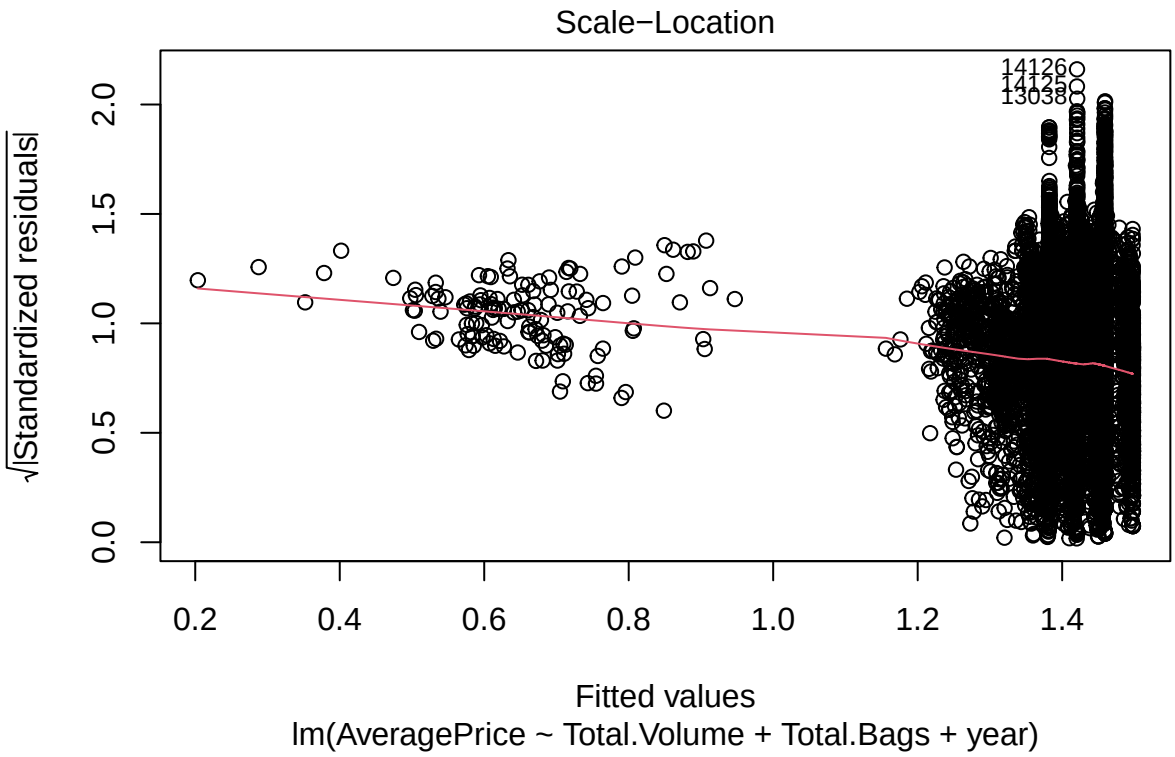
```
##
## Call:
## lm(formula = AveragePrice ~ Total.Volume + Total.Bags + year,
##     data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.97863 -0.29125 -0.03938  0.24949  1.82947
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -7.630e+01  7.109e+00 -10.732  < 2e-16 ***
```

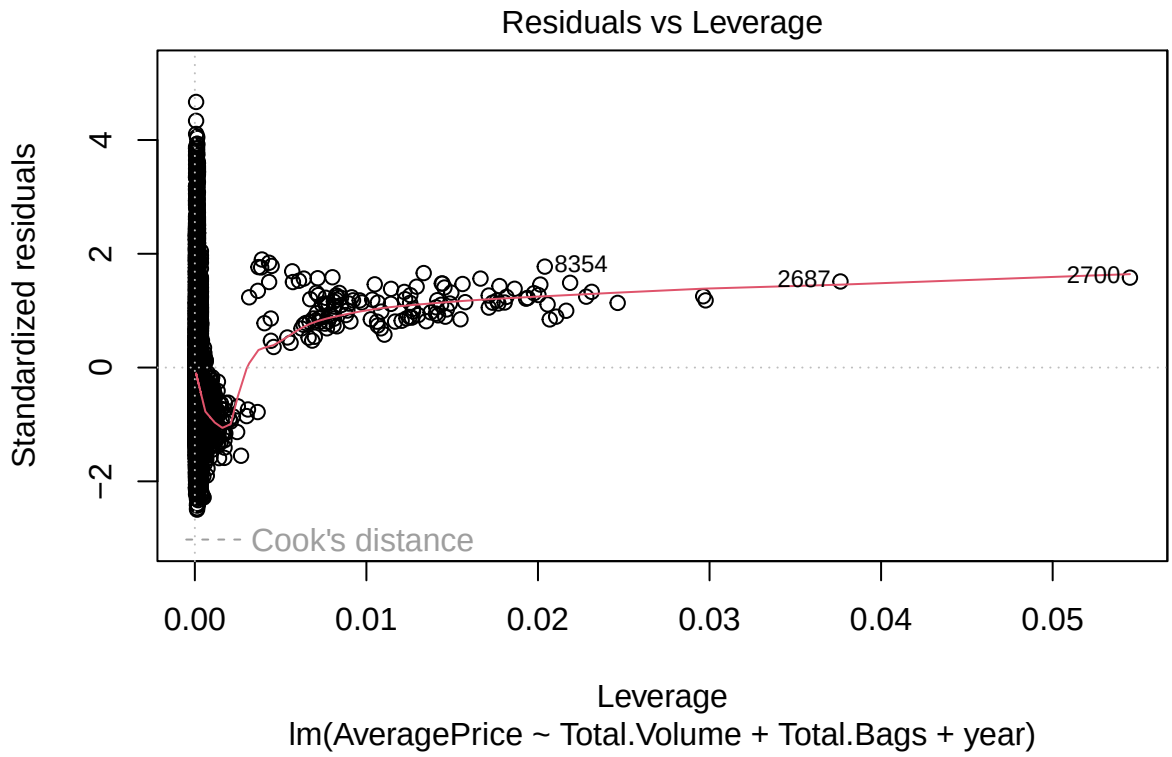
```
## Total.Volume -2.573e-08  3.431e-09  -7.499  6.8e-14 ***
## Total.Bags    1.007e-08  1.210e-08   0.832    0.405
## year          3.855e-02  3.526e-03  10.932  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3919 on 14595 degrees of freedom
## Multiple R-squared:  0.04661,    Adjusted R-squared:  0.04641
## F-statistic: 237.8 on 3 and 14595 DF,  p-value: < 2.2e-16
```

```
plot(lm2)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(AveragePrice ~ Total.Volume + Total.Bags + year)
14126
14125
−4
−2
0
2
4


Scale−Location
14126
14125
13038
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
Fitted values
lm(AveragePrice ~ Total.Volume + Total.Bags + year)



G. Build a third linear regression model using a different combination of predictors

```
lm3 <- lm(AveragePrice~region+type+Date, data=train)
summary(lm3)
```

```
##
## Call:
## lm(formula = AveragePrice ~ region + type + Date, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.99998 -0.12003  0.00317  0.12945  1.33304
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1.202e+00  2.737e-02  43.898  < 2e-16 ***
## regionAtlanta               -2.308e-01  1.983e-02 -11.640  < 2e-16 ***
## regionBaltimoreWashington -2.671e-02  1.957e-02  -1.365 0.172285
## regionBoise                 -2.172e-01  1.964e-02 -11.059  < 2e-16 ***
## regionBoston                -3.444e-02  1.957e-02  -1.760 0.078477 .
## regionBuffaloRochester      -5.488e-02  1.975e-02  -2.779 0.005466 **
## regionCalifornia            -1.729e-01  1.973e-02  -8.761  < 2e-16 ***
## regionCharlotte              3.670e-02  1.959e-02   1.874 0.060980 .
## regionChicago               -1.287e-03  1.948e-02  -0.066 0.947307
## regionCincinnatiDayton      -3.563e-01  1.960e-02 -18.180  < 2e-16 ***
## regionColumbus              -3.131e-01  1.960e-02 -15.971  < 2e-16 ***
## regionDallasFtWorth         -4.747e-01  1.983e-02 -23.942  < 2e-16 ***
```

```
## regionDenver              -3.373e-01  1.948e-02 -17.315  < 2e-16 ***
## regionDetroit             -2.925e-01  1.953e-02 -14.974  < 2e-16 ***
## regionGrandRapids         -5.547e-02  1.955e-02  -2.838 0.004545 **
## regionGreatLakes          -2.232e-01  1.945e-02 -11.479  < 2e-16 ***
## regionHarrisburgScranton  -5.669e-02  1.943e-02  -2.918 0.003528 **
## regionHartfordSpringfield  2.530e-01  1.960e-02  12.908  < 2e-16 ***
## regionHouston             -5.206e-01  1.973e-02 -26.384  < 2e-16 ***
## regionIndianapolis        -2.376e-01  1.958e-02 -12.131  < 2e-16 ***
## regionJacksonville        -5.333e-02  1.935e-02  -2.756 0.005854 **
## regionLasVegas            -1.884e-01  1.952e-02  -9.652  < 2e-16 ***
## regionLosAngeles          -3.581e-01  1.953e-02 -18.335  < 2e-16 ***
## regionLouisville          -2.681e-01  1.966e-02 -13.640  < 2e-16 ***
## regionMiamiFtLauderdale   -1.353e-01  1.940e-02  -6.977 3.15e-12 ***
## regionMidsouth            -1.637e-01  1.944e-02  -8.421  < 2e-16 ***
## regionNashville           -3.594e-01  1.979e-02 -18.159  < 2e-16 ***
## regionNewOrleansMobile    -2.627e-01  1.950e-02 -13.469  < 2e-16 ***
## regionNewYork              1.572e-01  1.968e-02   7.987 1.49e-15 ***
## regionNortheast            3.031e-02  1.941e-02   1.562 0.118349
## regionNorthernNewEngland  -8.856e-02  1.944e-02  -4.555 5.29e-06 ***
## regionOrlando             -6.238e-02  1.981e-02  -3.149 0.001642 **
## regionPhiladelphia         7.039e-02  1.941e-02   3.627 0.000288 ***
## regionPhoenixTucson       -3.373e-01  1.952e-02 -17.282  < 2e-16 ***
## regionPittsburgh          -2.135e-01  1.971e-02 -10.830  < 2e-16 ***
## regionPlains              -1.247e-01  1.981e-02  -6.293 3.19e-10 ***
## regionPortland            -2.586e-01  1.973e-02 -13.106  < 2e-16 ***
## regionRaleighGreensboro   -1.434e-02  1.989e-02  -0.721 0.471037
## regionRichmondNorfolk     -2.722e-01  1.955e-02 -13.924  < 2e-16 ***
## regionRoanoke             -3.169e-01  1.950e-02 -16.257  < 2e-16 ***
## regionSacramento           5.359e-02  1.954e-02   2.743 0.006091 **
## regionSanDiego            -1.716e-01  1.939e-02  -8.850  < 2e-16 ***
## regionSanFrancisco         2.464e-01  1.936e-02  12.726  < 2e-16 ***
## regionSeattle             -1.141e-01  1.957e-02  -5.833 5.55e-09 ***
## regionSouthCarolina       -1.560e-01  1.964e-02  -7.941 2.16e-15 ***
## regionSouthCentral        -4.655e-01  1.940e-02 -24.002  < 2e-16 ***
## regionSoutheast           -1.740e-01  1.976e-02  -8.807  < 2e-16 ***
## regionSpokane             -1.226e-01  1.951e-02  -6.280 3.48e-10 ***
## regionStLouis             -1.241e-01  1.956e-02  -6.345 2.30e-10 ***
## regionSyracuse            -5.392e-02  1.946e-02  -2.771 0.005603 **
## regionTampa               -1.774e-01  1.972e-02  -8.999  < 2e-16 ***
## regionTotalUS             -2.448e-01  1.955e-02 -12.519  < 2e-16 ***
## regionWest                -3.011e-01  1.953e-02 -15.415  < 2e-16 ***
## regionWestTexNewMexico    -2.985e-01  1.957e-02 -15.256  < 2e-16 ***
## typeorganic                4.972e-01  3.743e-03 132.825  < 2e-16 ***
## Date2015-01-11             7.605e-02  3.349e-02   2.271 0.023171 *
## Date2015-01-18             1.069e-01  3.339e-02   3.200 0.001378 **
## Date2015-01-25             1.177e-01  3.389e-02   3.472 0.000518 ***
## Date2015-02-01            -4.899e-02  3.379e-02  -1.450 0.147077
## Date2015-02-08             2.415e-02  3.339e-02   0.723 0.469502
## Date2015-02-15             7.870e-02  3.400e-02   2.315 0.020638 *
## Date2015-02-22             7.614e-02  3.400e-02   2.240 0.025134 *
## Date2015-03-01             1.450e-02  3.369e-02   0.430 0.666872
## Date2015-03-08             7.284e-02  3.411e-02   2.136 0.032711 *
## Date2015-03-15             1.108e-01  3.330e-02   3.326 0.000882 ***
## Date2015-03-22             5.617e-02  3.359e-02   1.672 0.094448 .
```

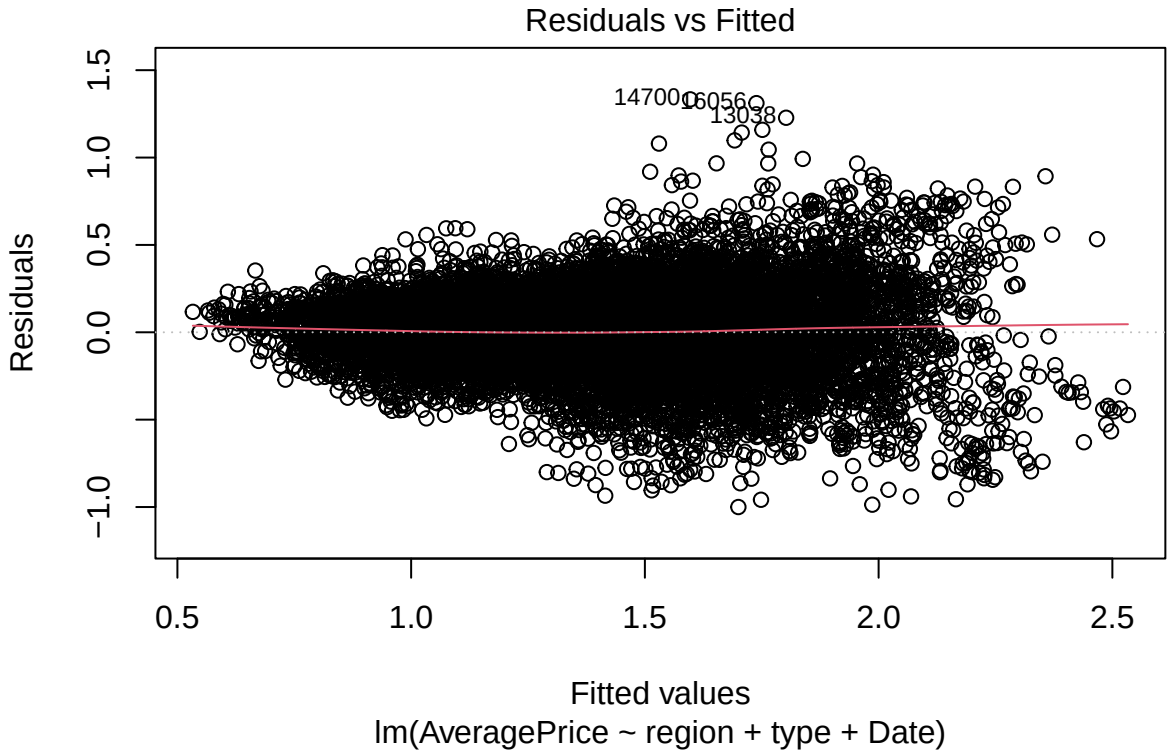
```
## Date2015-03-29            1.055e-01  3.359e-02   3.142 0.001679 **
## Date2015-04-05            1.290e-01  3.422e-02   3.769 0.000164 ***
## Date2015-04-12            5.887e-02  3.410e-02   1.726 0.084314 .
## Date2015-04-19            9.478e-02  3.400e-02   2.788 0.005314 **
## Date2015-04-26            8.278e-02  3.312e-02   2.499 0.012457 *
## Date2015-05-03           -1.045e-03  3.359e-02  -0.031 0.975185
## Date2015-05-10            2.191e-02  3.456e-02   0.634 0.526148
## Date2015-05-17            5.713e-02  3.421e-02   1.670 0.094957 .
## Date2015-05-24            8.961e-02  3.349e-02   2.676 0.007461 **
## Date2015-05-31            9.198e-02  3.321e-02   2.770 0.005614 **
## Date2015-06-07            8.402e-02  3.321e-02   2.530 0.011418 *
## Date2015-06-14            1.115e-01  3.400e-02   3.280 0.001041 **
## Date2015-06-21            1.194e-01  3.421e-02   3.489 0.000486 ***
## Date2015-06-28            1.151e-01  3.389e-02   3.396 0.000686 ***
## Date2015-07-05            1.148e-01  3.330e-02   3.447 0.000570 ***
## Date2015-07-12            1.266e-01  3.369e-02   3.759 0.000171 ***
## Date2015-07-19            9.555e-02  3.433e-02   2.783 0.005391 **
## Date2015-07-26            1.393e-01  3.389e-02   4.111 3.96e-05 ***
## Date2015-08-02            1.875e-01  3.421e-02   5.481 4.30e-08 ***
## Date2015-08-09            1.538e-01  3.421e-02   4.496 6.99e-06 ***
## Date2015-08-16            1.648e-01  3.379e-02   4.876 1.09e-06 ***
## Date2015-08-23            1.321e-01  3.468e-02   3.811 0.000139 ***
## Date2015-08-30            1.245e-01  3.456e-02   3.604 0.000315 ***
## Date2015-09-06            1.445e-01  3.410e-02   4.236 2.28e-05 ***
## Date2015-09-13            1.630e-01  3.359e-02   4.854 1.22e-06 ***
## Date2015-09-20            1.697e-01  3.410e-02   4.975 6.60e-07 ***
## Date2015-09-27            1.530e-01  3.349e-02   4.569 4.93e-06 ***
## Date2015-10-04            1.341e-01  3.410e-02   3.933 8.44e-05 ***
## Date2015-10-11            9.904e-02  3.378e-02   2.931 0.003380 **
## Date2015-10-18            1.139e-01  3.400e-02   3.351 0.000807 ***
## Date2015-10-25            9.705e-02  3.444e-02   2.818 0.004842 **
## Date2015-11-01            1.709e-02  3.400e-02   0.503 0.615151
## Date2015-11-08            4.700e-02  3.359e-02   1.399 0.161736
## Date2015-11-15            3.052e-02  3.456e-02   0.883 0.377151
## Date2015-11-22            3.106e-02  3.379e-02   0.919 0.358025
## Date2015-11-29            3.015e-02  3.400e-02   0.887 0.375246
## Date2015-12-06            4.197e-03  3.321e-02   0.126 0.899434
## Date2015-12-13            8.531e-03  3.378e-02   0.253 0.800649
## Date2015-12-20            4.418e-02  3.330e-02   1.327 0.184634
## Date2015-12-27           -5.463e-05  3.369e-02  -0.002 0.998706
## Date2016-01-03           -8.390e-02  3.389e-02  -2.475 0.013320 *
## Date2016-01-10           -7.298e-02  3.330e-02  -2.191 0.028439 *
## Date2016-01-17           -2.476e-03  3.410e-02  -0.073 0.942109
## Date2016-01-24           -6.014e-02  3.421e-02  -1.758 0.078791 .
## Date2016-01-31           -3.568e-02  3.358e-02  -1.062 0.288051
## Date2016-02-07           -1.157e-01  3.330e-02  -3.475 0.000513 ***
## Date2016-02-14           -5.842e-02  3.358e-02  -1.740 0.081958 .
## Date2016-02-21           -1.311e-02  3.330e-02  -0.394 0.693762
## Date2016-02-28           -1.900e-02  3.400e-02  -0.559 0.576304
## Date2016-03-06           -2.121e-02  3.339e-02  -0.635 0.525377
## Date2016-03-13           -8.315e-02  3.349e-02  -2.483 0.013035 *
## Date2016-03-20           -7.842e-02  3.359e-02  -2.335 0.019568 *
## Date2016-03-27           -3.337e-02  3.349e-02  -0.996 0.319108
## Date2016-04-03           -5.097e-02  3.358e-02  -1.518 0.129138
```

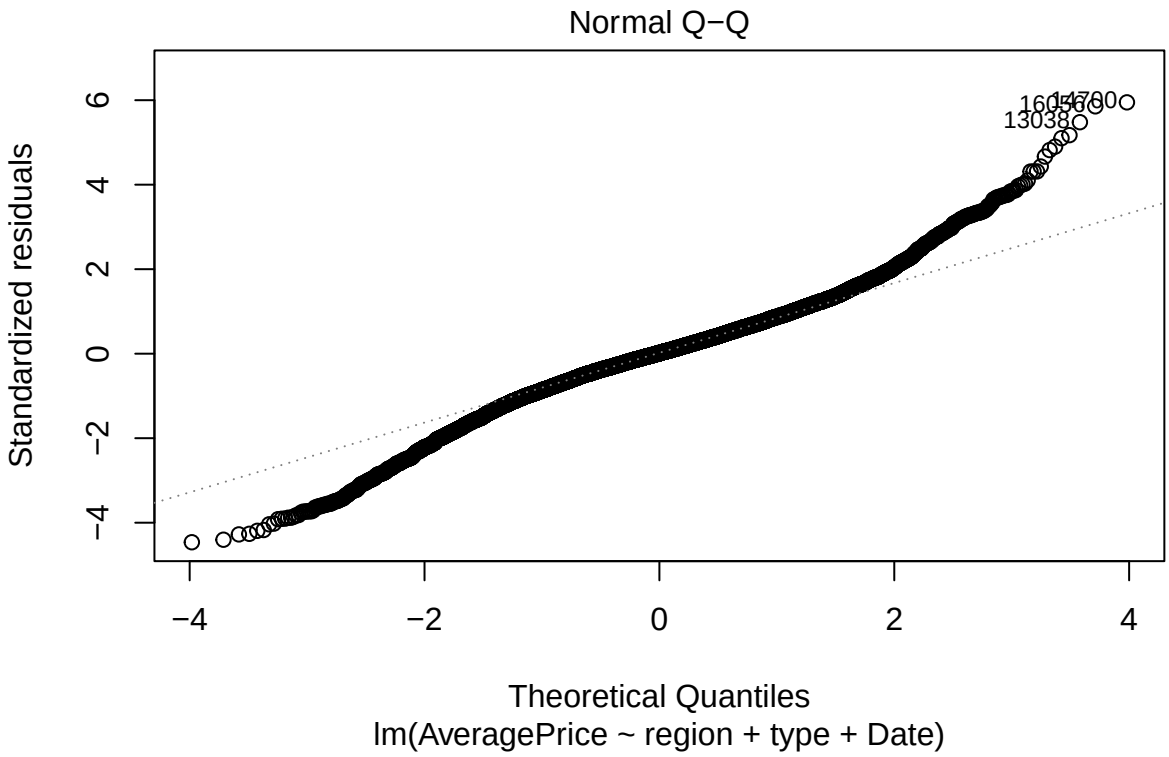
```
## Date2016-04-10          -8.753e-02  3.399e-02  -2.575 0.010041 *
## Date2016-04-17          -6.549e-02  3.369e-02  -1.944 0.051901 .
## Date2016-04-24          -1.037e-01  3.389e-02  -3.061 0.002213 **
## Date2016-05-01          -1.129e-01  3.379e-02  -3.341 0.000836 ***
## Date2016-05-08          -1.479e-01  3.379e-02  -4.378 1.21e-05 ***
## Date2016-05-15          -7.743e-02  3.389e-02  -2.285 0.022332 *
## Date2016-05-22          -5.107e-02  3.368e-02  -1.516 0.129510
## Date2016-05-29          -2.773e-02  3.410e-02  -0.813 0.416148
## Date2016-06-05          -4.950e-02  3.359e-02  -1.474 0.140533
## Date2016-06-12          -3.280e-03  3.339e-02  -0.098 0.921759
## Date2016-06-19          -2.447e-02  3.399e-02  -0.720 0.471632
## Date2016-06-26           2.294e-02  3.379e-02   0.679 0.497182
## Date2016-07-03           1.171e-03  3.339e-02   0.035 0.972031
## Date2016-07-10           7.007e-02  3.369e-02   2.080 0.037531 *
## Date2016-07-17           1.307e-01  3.303e-02   3.956 7.67e-05 ***
## Date2016-07-24           1.993e-01  3.359e-02   5.933 3.04e-09 ***
## Date2016-07-31           1.635e-01  3.330e-02   4.910 9.19e-07 ***
## Date2016-08-07           1.132e-01  3.339e-02   3.391 0.000698 ***
## Date2016-08-14           1.368e-01  3.421e-02   3.999 6.40e-05 ***
## Date2016-08-21           1.255e-01  3.359e-02   3.735 0.000188 ***
## Date2016-08-28           1.029e-01  3.456e-02   2.978 0.002908 **
## Date2016-09-04           9.843e-02  3.349e-02   2.939 0.003299 **
## Date2016-09-11           7.294e-02  3.358e-02   2.172 0.029879 *
## Date2016-09-18           1.528e-01  3.349e-02   4.562 5.10e-06 ***
## Date2016-09-25           2.409e-01  3.399e-02   7.086 1.44e-12 ***
## Date2016-10-02           2.917e-01  3.339e-02   8.736  < 2e-16 ***
## Date2016-10-09           1.963e-01  3.359e-02   5.844 5.20e-09 ***
## Date2016-10-16           1.967e-01  3.349e-02   5.873 4.37e-09 ***
## Date2016-10-23           2.366e-01  3.359e-02   7.046 1.93e-12 ***
## Date2016-10-30           4.118e-01  3.368e-02  12.225  < 2e-16 ***
## Date2016-11-06           3.421e-01  3.349e-02  10.215  < 2e-16 ***
## Date2016-11-13           2.800e-01  3.359e-02   8.337  < 2e-16 ***
## Date2016-11-20           2.294e-01  3.359e-02   6.830 8.83e-12 ***
## Date2016-11-27           2.051e-01  3.349e-02   6.125 9.31e-10 ***
## Date2016-12-04           7.273e-02  3.330e-02   2.184 0.028964 *
## Date2016-12-11           3.340e-02  3.379e-02   0.989 0.322863
## Date2016-12-18          -1.649e-02  3.444e-02  -0.479 0.632108
## Date2016-12-25           2.172e-02  3.379e-02   0.643 0.520304
## Date2017-01-01          -7.166e-03  3.312e-02  -0.216 0.828709
## Date2017-01-08          -2.047e-03  3.400e-02  -0.060 0.951983
## Date2017-01-15           3.999e-02  3.389e-02   1.180 0.237983
## Date2017-01-22          -6.569e-02  3.340e-02  -1.967 0.049184 *
## Date2017-01-29          -3.265e-02  3.389e-02  -0.963 0.335354
## Date2017-02-05          -1.333e-01  3.369e-02  -3.955 7.68e-05 ***
## Date2017-02-12          -9.052e-02  3.330e-02  -2.718 0.006569 **
## Date2017-02-19          -2.750e-02  3.369e-02  -0.816 0.414339
## Date2017-02-26          -5.221e-02  3.422e-02  -1.526 0.127113
## Date2017-03-05           8.764e-03  3.359e-02   0.261 0.794143
## Date2017-03-12           1.751e-01  3.399e-02   5.151 2.63e-07 ***
## Date2017-03-19           1.879e-01  3.411e-02   5.509 3.67e-08 ***
## Date2017-03-26           1.153e-01  3.349e-02   3.443 0.000576 ***
## Date2017-04-02           1.493e-01  3.339e-02   4.471 7.86e-06 ***
## Date2017-04-09           1.923e-01  3.422e-02   5.620 1.95e-08 ***
## Date2017-04-16           2.013e-01  3.400e-02   5.922 3.25e-09 ***
```

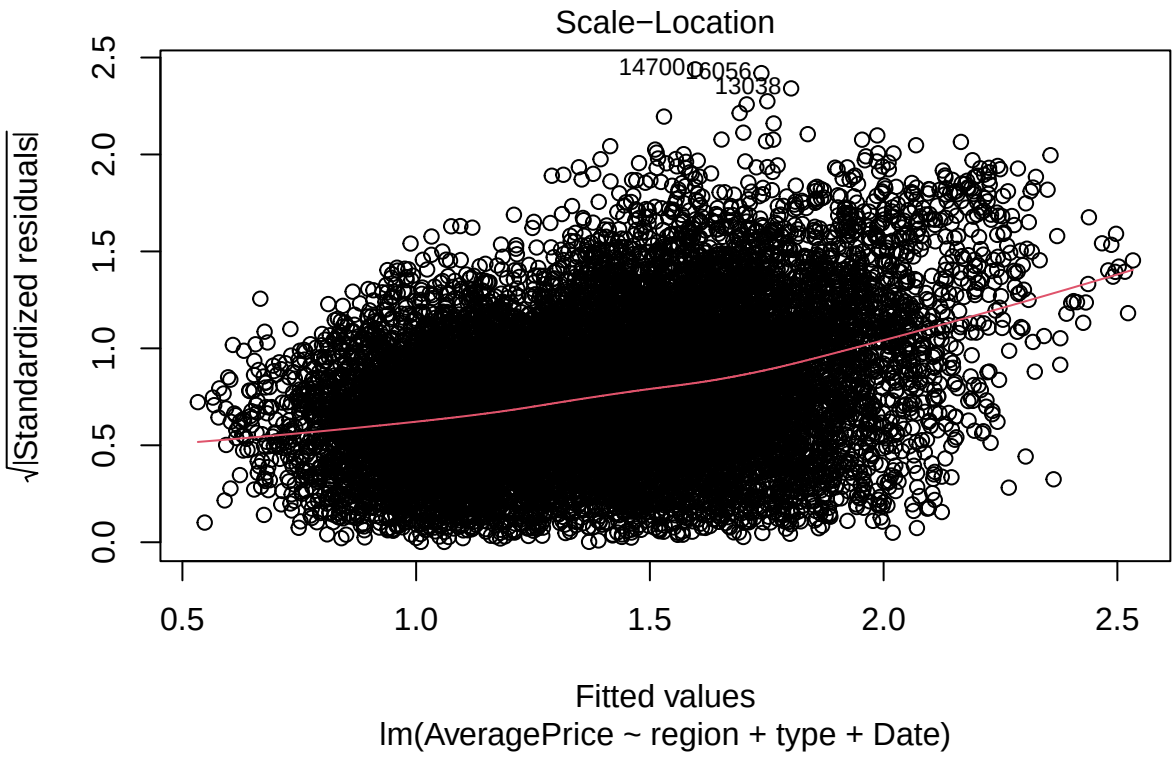
```
## Date2017-04-23            2.505e-01  3.349e-02   7.481 7.79e-14 ***
## Date2017-04-30            2.531e-01  3.410e-02   7.422 1.22e-13 ***
## Date2017-05-07            1.877e-01  3.339e-02   5.620 1.94e-08 ***
## Date2017-05-14            2.142e-01  3.369e-02   6.358 2.10e-10 ***
## Date2017-05-21            2.584e-01  3.389e-02   7.624 2.62e-14 ***
## Date2017-05-28            2.992e-01  3.379e-02   8.855  < 2e-16 ***
## Date2017-06-04            2.627e-01  3.389e-02   7.750 9.78e-15 ***
## Date2017-06-11            2.428e-01  3.433e-02   7.074 1.58e-12 ***
## Date2017-06-18            2.589e-01  3.312e-02   7.819 5.71e-15 ***
## Date2017-06-25            2.581e-01  3.421e-02   7.546 4.77e-14 ***
## Date2017-07-02            2.798e-01  3.330e-02   8.401  < 2e-16 ***
## Date2017-07-09            2.426e-01  3.433e-02   7.067 1.66e-12 ***
## Date2017-07-16            2.811e-01  3.339e-02   8.417  < 2e-16 ***
## Date2017-07-23            2.726e-01  3.400e-02   8.019 1.15e-15 ***
## Date2017-07-30            2.682e-01  3.340e-02   8.031 1.04e-15 ***
## Date2017-08-06            2.871e-01  3.432e-02   8.364  < 2e-16 ***
## Date2017-08-13            3.520e-01  3.349e-02  10.511  < 2e-16 ***
## Date2017-08-20            4.260e-01  3.369e-02  12.645  < 2e-16 ***
## Date2017-08-27            5.220e-01  3.339e-02  15.632  < 2e-16 ***
## Date2017-09-03            5.710e-01  3.368e-02  16.952  < 2e-16 ***
## Date2017-09-10            5.577e-01  3.330e-02  16.749  < 2e-16 ***
## Date2017-09-17            5.352e-01  3.399e-02  15.746  < 2e-16 ***
## Date2017-09-24            5.504e-01  3.400e-02  16.190  < 2e-16 ***
## Date2017-10-01            5.815e-01  3.368e-02  17.265  < 2e-16 ***
## Date2017-10-08            5.453e-01  3.321e-02  16.421  < 2e-16 ***
## Date2017-10-15            4.873e-01  3.421e-02  14.244  < 2e-16 ***
## Date2017-10-22            3.740e-01  3.339e-02  11.199  < 2e-16 ***
## Date2017-10-29            3.017e-01  3.444e-02   8.760  < 2e-16 ***
## Date2017-11-05            2.771e-01  3.359e-02   8.250  < 2e-16 ***
## Date2017-11-12            2.008e-01  3.444e-02   5.829 5.69e-09 ***
## Date2017-11-19            2.082e-01  3.368e-02   6.181 6.54e-10 ***
## Date2017-11-26            2.089e-01  3.421e-02   6.107 1.04e-09 ***
## Date2017-12-03            9.331e-02  3.421e-02   2.727 0.006394 **
## Date2017-12-10            4.819e-02  3.410e-02   1.413 0.157653
## Date2017-12-17            8.250e-02  3.400e-02   2.427 0.015246 *
## Date2017-12-24            1.650e-01  3.456e-02   4.776 1.81e-06 ***
## Date2017-12-31           -1.976e-02  3.369e-02  -0.587 0.557542
## Date2018-01-07            5.718e-02  3.369e-02   1.697 0.089651 .
## Date2018-01-14            1.251e-01  3.312e-02   3.777 0.000159 ***
## Date2018-01-21            8.318e-02  3.389e-02   2.454 0.014121 *
## Date2018-01-28            1.030e-01  3.359e-02   3.067 0.002166 **
## Date2018-02-04           -5.240e-02  3.359e-02  -1.560 0.118730
## Date2018-02-11            6.818e-03  3.400e-02   0.201 0.841069
## Date2018-02-18            8.194e-02  3.312e-02   2.474 0.013367 *
## Date2018-02-25            7.289e-02  3.400e-02   2.144 0.032068 *
## Date2018-03-04            6.790e-02  3.349e-02   2.027 0.042635 *
## Date2018-03-11            3.608e-02  3.400e-02   1.061 0.288557
## Date2018-03-18            1.838e-02  3.330e-02   0.552 0.581070
## Date2018-03-25            6.994e-02  3.410e-02   2.051 0.040314 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2258 on 14376 degrees of freedom
## Multiple R-squared:  0.6883, Adjusted R-squared:  0.6835
```

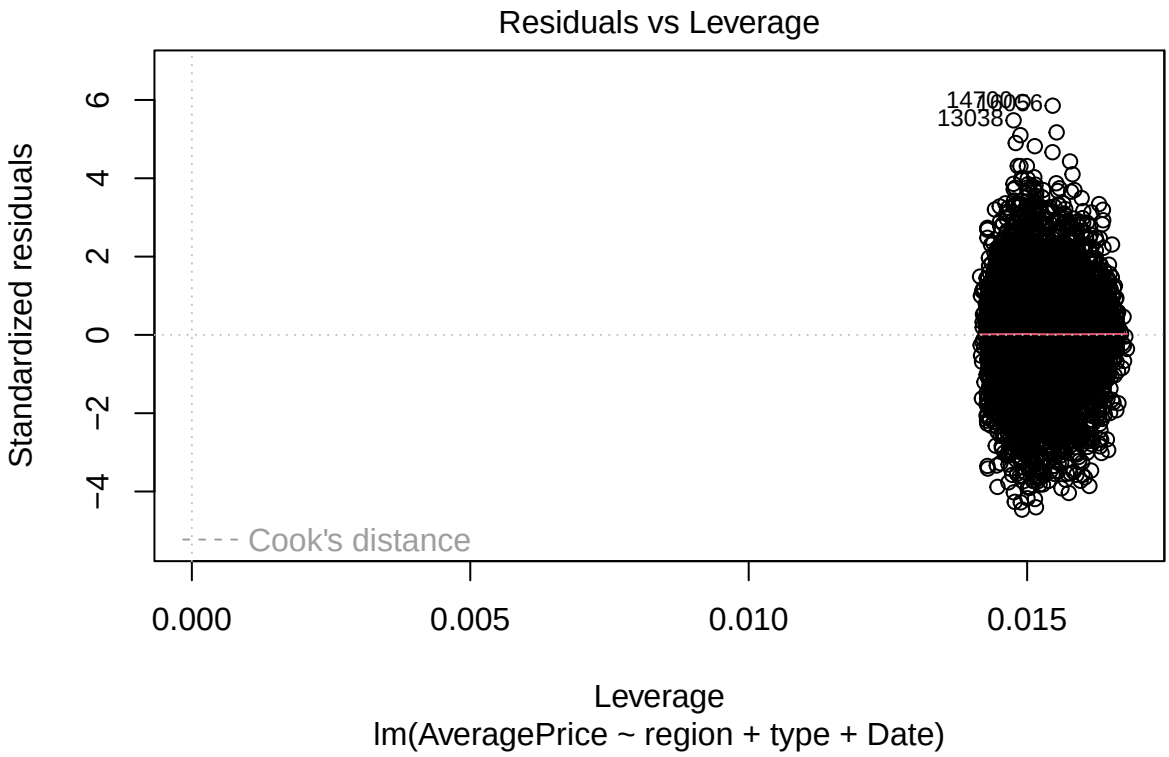
```
## F-statistic:   143 on 222 and 14376 DF,  p-value: < 2.2e-16
```

```
plot(lm3)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(AveragePrice ~ region + type + Date)
−4
−2
0
2
4
6
−4
−2
0
2
4
13038


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
14700
16056
13038
0.5
1.0
1.5
2.0
2.5
Fitted values
lm(AveragePrice ~ region + type + Date)



H. Write a paragraph comparing the results In the first model, I used the year as the predictor which gave me a .007909 R-squared value. In the second model I used the total volume, total bags, and the year as my predictors which gave me a .04661 R-squared value which is slightly better. Before doing my last model, I looked at the data more, I realized that the region, type, and date could have much better results. After successfully concluding that, I ended up with a .6883 R-squared value which is significantly better. The F-statistic of the third one also shows us that this is the most reliable model of the effects due to the larger sample size. ### I. Using the 3 models, predict and evaluate the test data using metrics correlation and mse.

```
print("Model 1:")
```

```
## [1] "Model 1:"
```

```
pred1 <- predict(lm1, newdata=test)
cor1 <- cor(pred1, test$AveragePrice)
print(paste("correlation: ", cor1))
```

```
## [1] "correlation:  0.109771621795755"
```

```
mse1 <- mean((pred1-test$AveragePrice)^2)
print(paste("mse: ", mse1))
```

```
## [1] "mse:  0.164618639639961"
```

```
rmse1 <- sqrt(mse1)
print(paste("rmse: ", rmse1))
```

```
## [1] "rmse:  0.405732226523802"
```

```
print("Model 2:")
```

```
## [1] "Model 2:"
```

```
pred2 <- predict(lm2, newdata=test)
cor2 <- cor(pred2, test$AveragePrice)
print(paste("correlation: ", cor2))
```

```
## [1] "correlation:  0.215278537798768"
```

```
mse2 <- mean((pred2-test$AveragePrice)^2)
print(paste("mse: ", mse2))
```

```
## [1] "mse:  0.158829680849729"
```

```
rmse2 <- sqrt(mse2)
print(paste("rmse: ", rmse2))
```

```
## [1] "rmse:  0.398534416142106"
```

```
print("Model 3:")
```

```
## [1] "Model 3:"
```

```
pred3 <- predict(lm3, newdata=test)
cor3 <- cor(pred3, test$AveragePrice)
print(paste("correlation: ", cor3))
```

```
## [1] "correlation:  0.827478944212141"
```

```
mse3 <- mean((pred3-test$AveragePrice)^2)
print(paste("mse: ", mse3))
```

```
## [1] "mse:  0.0525005787156866"
```

```
rmse3 <- sqrt(mse3)
print(paste("rmse: ", rmse3))
```

```
## [1] "rmse:  0.229130047605474"
```

Obviously, the first model is by far the worst model due to lack of predictors. The third model shows that it is the strongest model because it has the highest correlation. It also has a lower mse and rmse which further proves the third model be much better and have much more significant predictors. The clear difference here is that multiple regression models are far superior, and you can further improve your results by adding a more significant combination of predictors to your model.