5 years of historical stock quotes

Code

Sharmin Gaziani

How does linear regression work?

Linear regression is a tool we use in machine learning for predicting the value of a variable based on other variables in the data set. One strength with linear regression is how easy it is to implement on a data set, especially in R because there is a built in function for it. It is also a great option to make sure the data set does not over-fit and can be regularized. One weakness with linear regression is the emphasis on linear relationships. This model performs very poorly on non-linear relationships and it is not realistic in the real world to only be working on linear data sets.

a. Divide into 80/20 train/set

Hide

```
set.seed(1234)
i <- sample(1:nrow(original), nrow(original)*0.80, replace=FALSE)
train <- original[i,]
test <- original[-i,]
```

Split the data up into two chunks, the larger chunk was the training data and the smaller one was the testing data.

b. The 5 R functions I used for data exploration on the training data was sum, range, correlation, standard deviation, and covariance

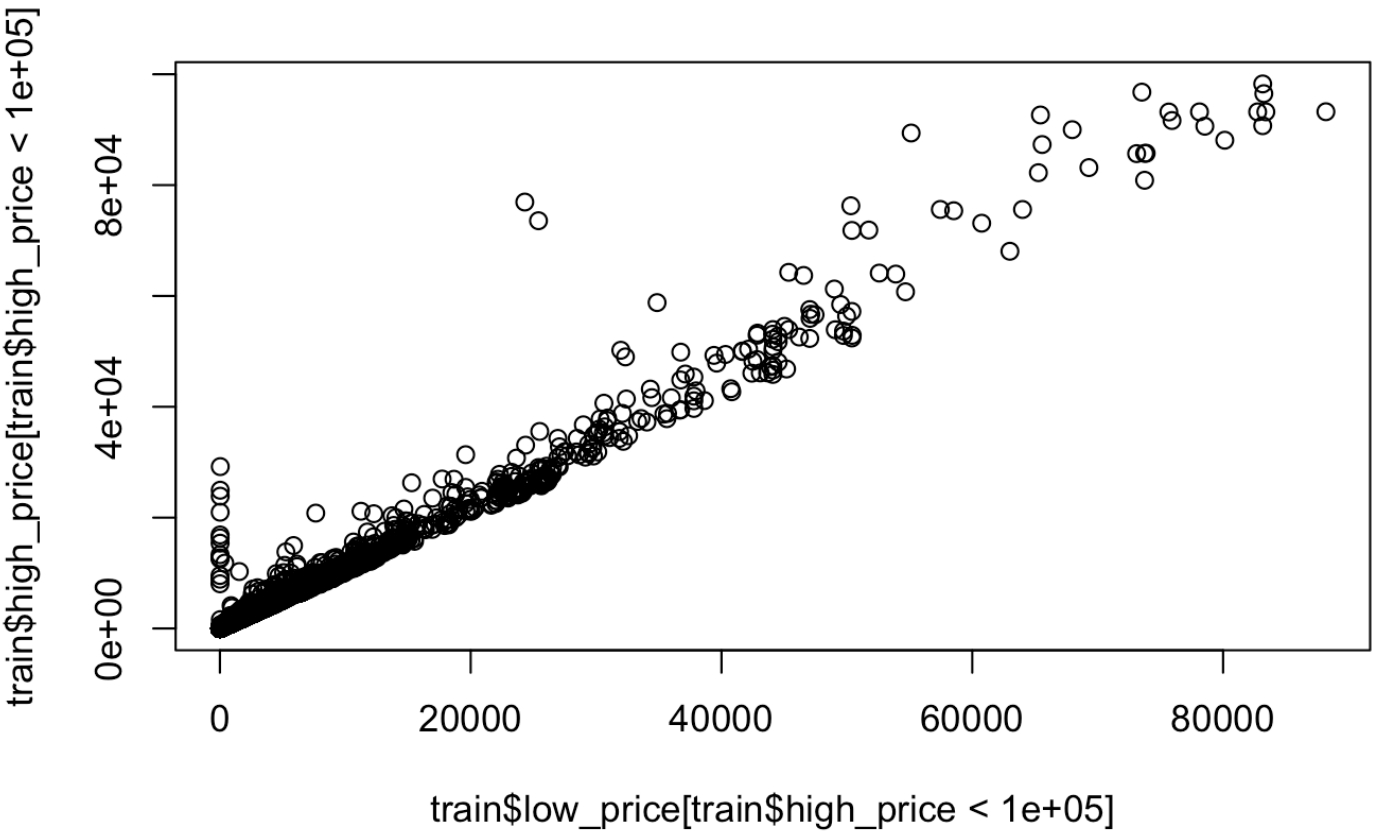
Hide

```
sum(train$high_price)
range(train$high_price)
cov(train$high_price, train$low_price)
cor(train$high_price, train$low_price)
sd(train$high_price)
```

c. 2 informative graphs using the training data

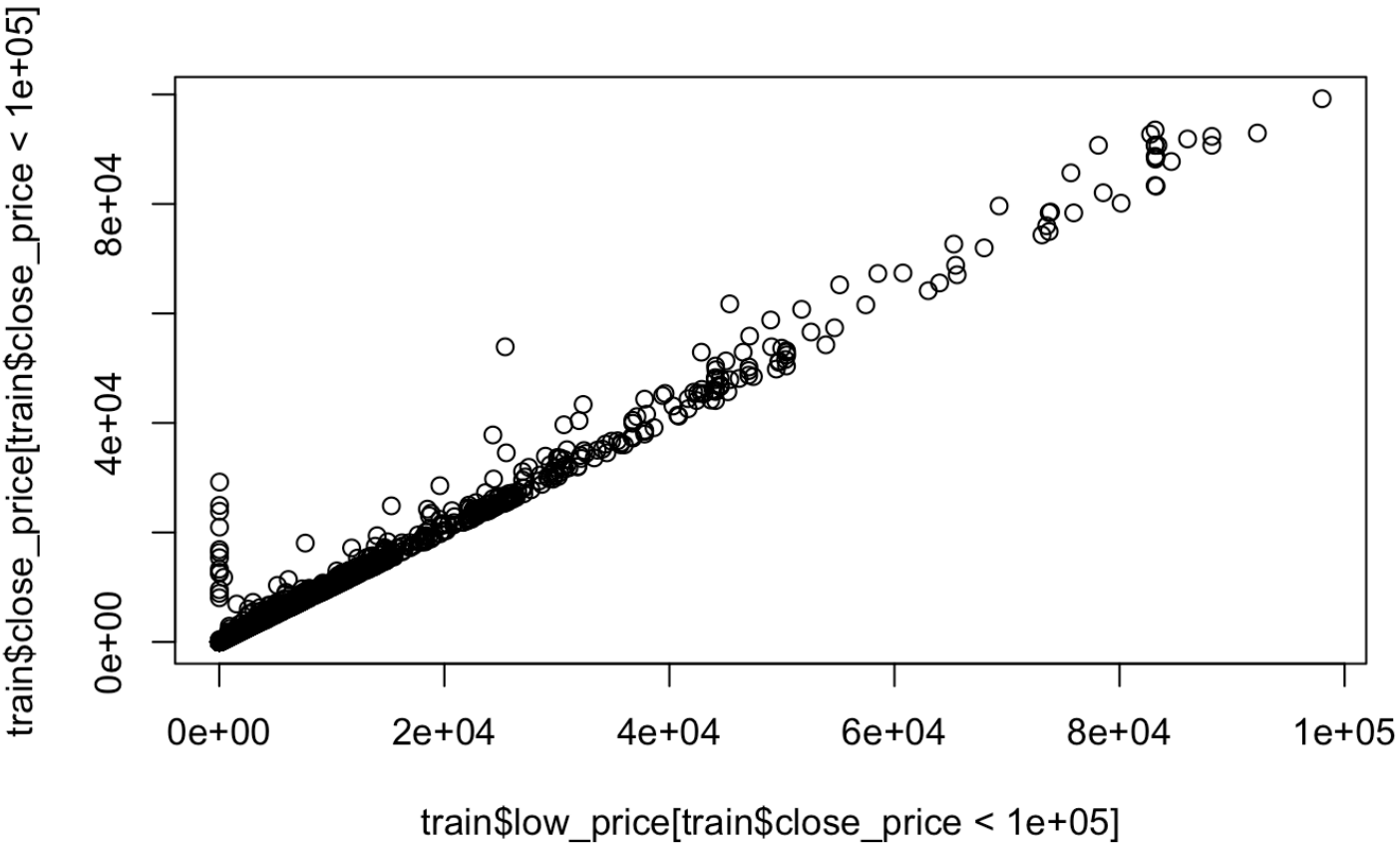
Hide

```
plot(train$low_price[train$high_price < 100000] ,
     train$high_price[train$high_price < 100000])
```



Hide

```
plot(train$low_price[train$close_price < 100000] ,
     train$close_price[train$close_price < 100000])
```



Used the R plot function to create a plot for the low and high prices as well was comparing it to the prices of the stock when the market closed. Because my data set was so large, I only plotted a portion of the data set so the graphs were easier to read.

d. Simple linear regression model

Hide

```
s <- sample(1:nrow(original), nrow(original)*0.001,
replace=FALSE)
samp <- original[i,]

lm1 <- lm(samp$high_price~samp$datetime)
summary(lm1)
```

For my linear regression model, I ran the built in R function using the lm for the highest price of a stock and the time it was that high. Because my data set was so large, all the data sets were not able to fit in to the vector for the linear regression so I decided to take a sample of the original data set and create the linear model against that. The summary(lm1) gave me a good chunk of data including the residuals, the coefficients, the residual standard error, mulitple R-squared, and the F-statistic. Using the following metrics, I was able to get a better idea of my idea and what it means in terms of how they relate to each other. The standard error gave me an estimate of different variations with the coefficients and the intervals. And the last 3 lines of the

summary gave me information on how well the model would fit into the training data. The RSE was sum of squared errors and tell us if its in a positive or negative direction. The F-statistic looked at the predictors to determine if there were any significant ones present for Y.

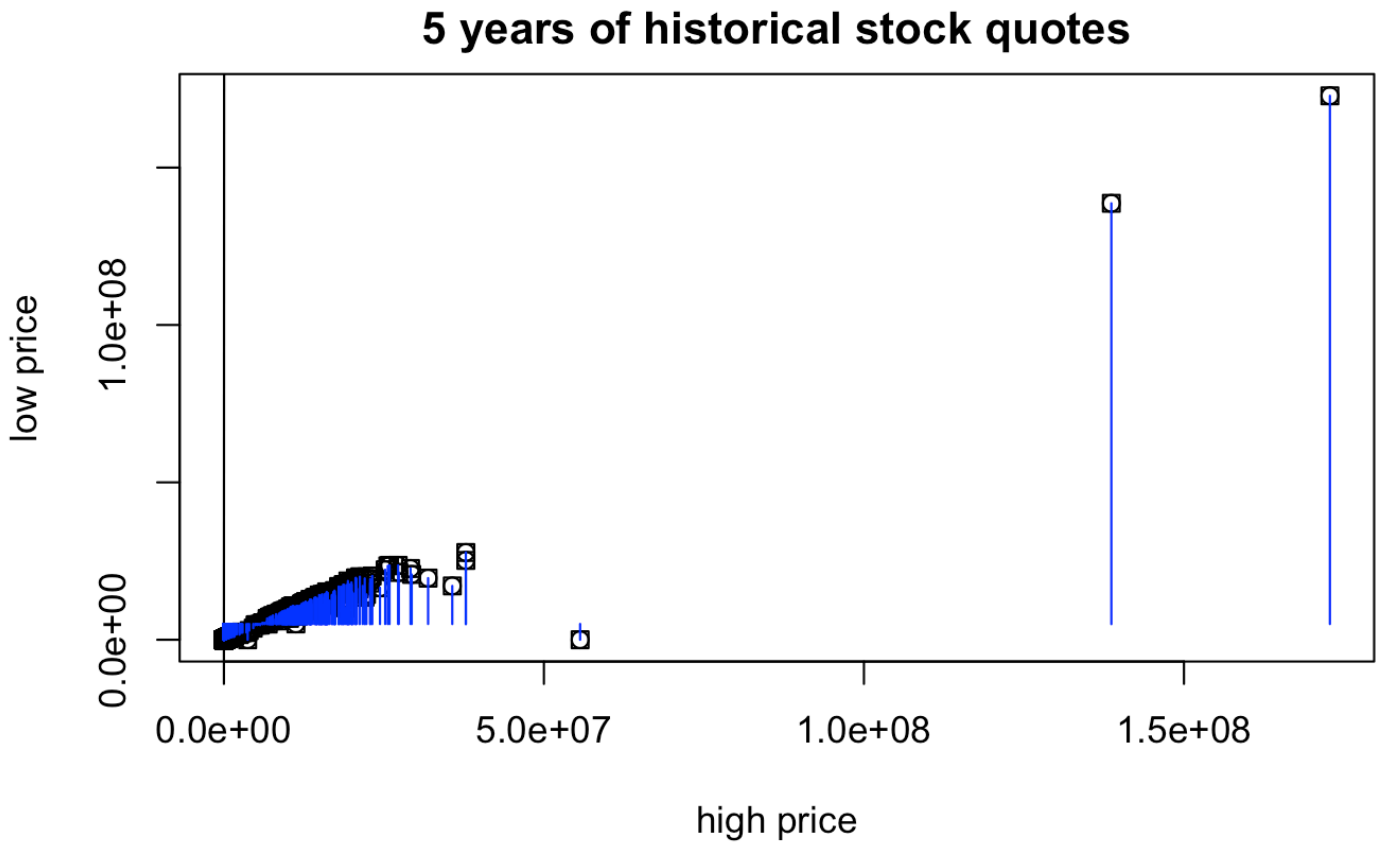
e. Plot the residuals

Hide

```
plot(original$high_price, original$low_price, main="5 years of historical stock quote
s",
     xlab="high price", ylab="low price")
abline(lm1)
```

Hide

```
points(original$high_price, original$low_price, pch=0)
segments(original$high_price, original$low_price, original$high_price, pred, col="blu
e")
```



After plotting the residuals, we have to make sure the residuals represent the data properly and whether or not it fits in the model properly. After looking at this plot, we can see that the residuals are in a linear pattern because the points are formed around the straight diagonal line. However, the plots are not equally

distributed. I would consider this residual plot a “good” model because all the points are not scattered aimlessly around the graph and meet the regression assumptions well. Because it is a “good” model, it is also considered a Normal Q-Q because the points are surrounded by the straight line.

f. Multiple linear regression model with multiple predictors

Hide

```
s2 <- sample(1:nrow(original), nrow(original)*0.001,
replace=FALSE)
samp2 <- original[i,]

lm2 <- lm(samp2$open_price~samp2$close_price)
summary(lm2)
```

```

Call:
lm(formula = samp2$open_price ~ samp2$close_price)

Residuals:
      Min        1Q    Median        3Q       Max
-10465076        -5        -4        -4  14835811

Coefficients:
                   Estimate Std. Error   t value Pr(>|t|)
(Intercept)       4.164e+00  3.937e+00     1.058     0.29
samp2$close_price 1.010e+00  3.771e-05 26780.936   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9835 on 6241534 degrees of freedom
Multiple R-squared:  0.9914,    Adjusted R-squared:  0.9914
F-statistic: 7.172e+08 on 1 and 6241534 DF,  p-value: < 2.2e-16
```

Hide

```
lm3 <- lm(samp2$high_price~samp2$close_price)
summary(lm3)
```

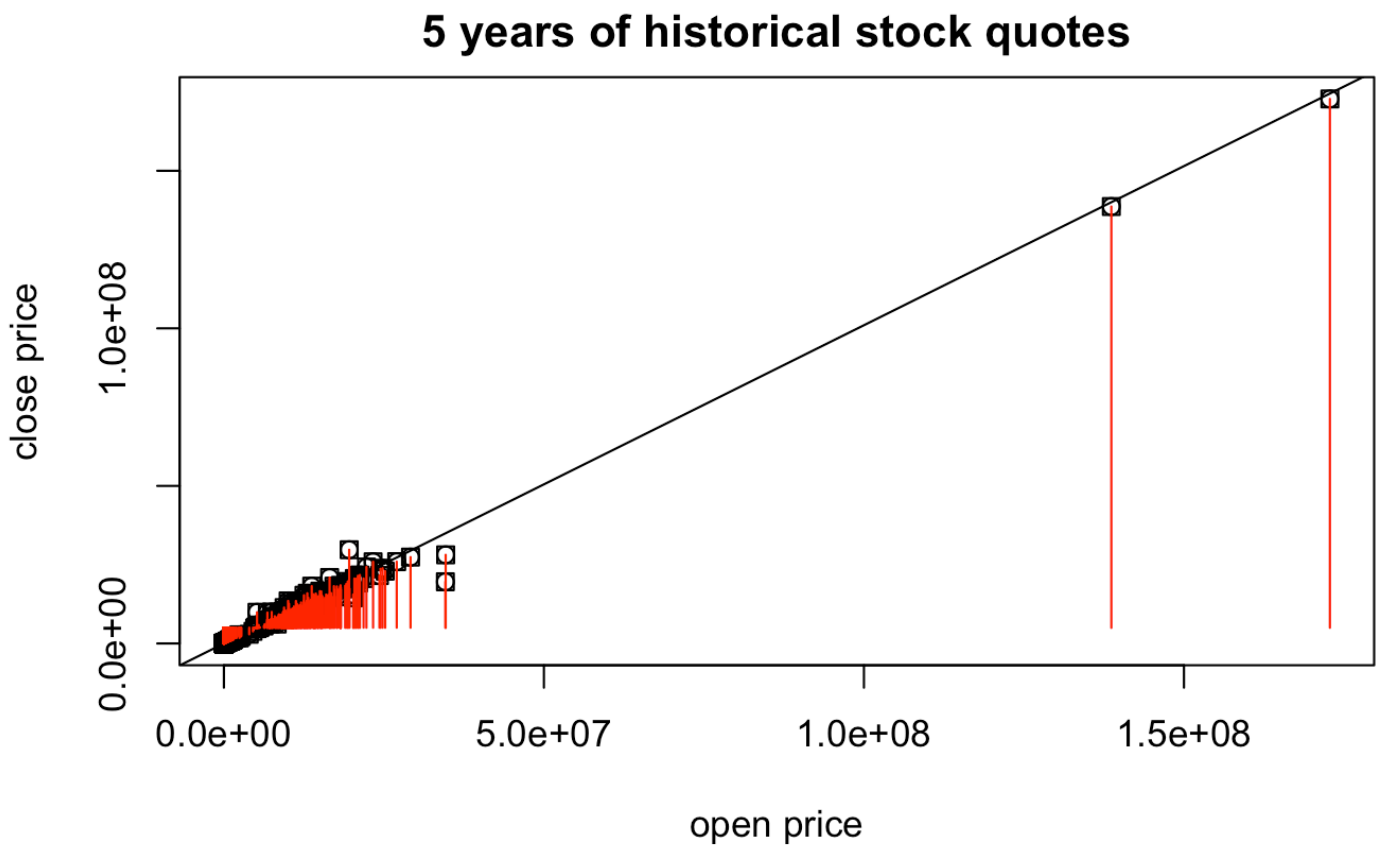
```
Call:
lm(formula = samp2$high_price ~ samp2$close_price)

Residuals:
      Min        1Q    Median        3Q       Max
-10348890       -25       -24       -23  55654768

Coefficients:
                   Estimate Std. Error   t value Pr(>|t|)
(Intercept)       2.305e+01  9.880e+00     2.333   0.0196 *
samp2$close_price 1.060e+00  9.465e-05 11198.307   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 24680 on 6241534 degrees of freedom
Multiple R-squared:  0.9526,    Adjusted R-squared:  0.9526
F-statistic: 1.254e+08 on 1 and 6241534 DF,  p-value: < 2.2e-16
```

Hide

```
plot(original$open_price, original$close_price, main="5 years of historical stock quo
tes",
     xlab="open price", ylab="close price")
abline(lm2)
```

Hide

```
points(original$open_price, original$close_price, pch=0)
segments(original$open_price, original$close_price, original$open_price, pred, col="r
ed")
```

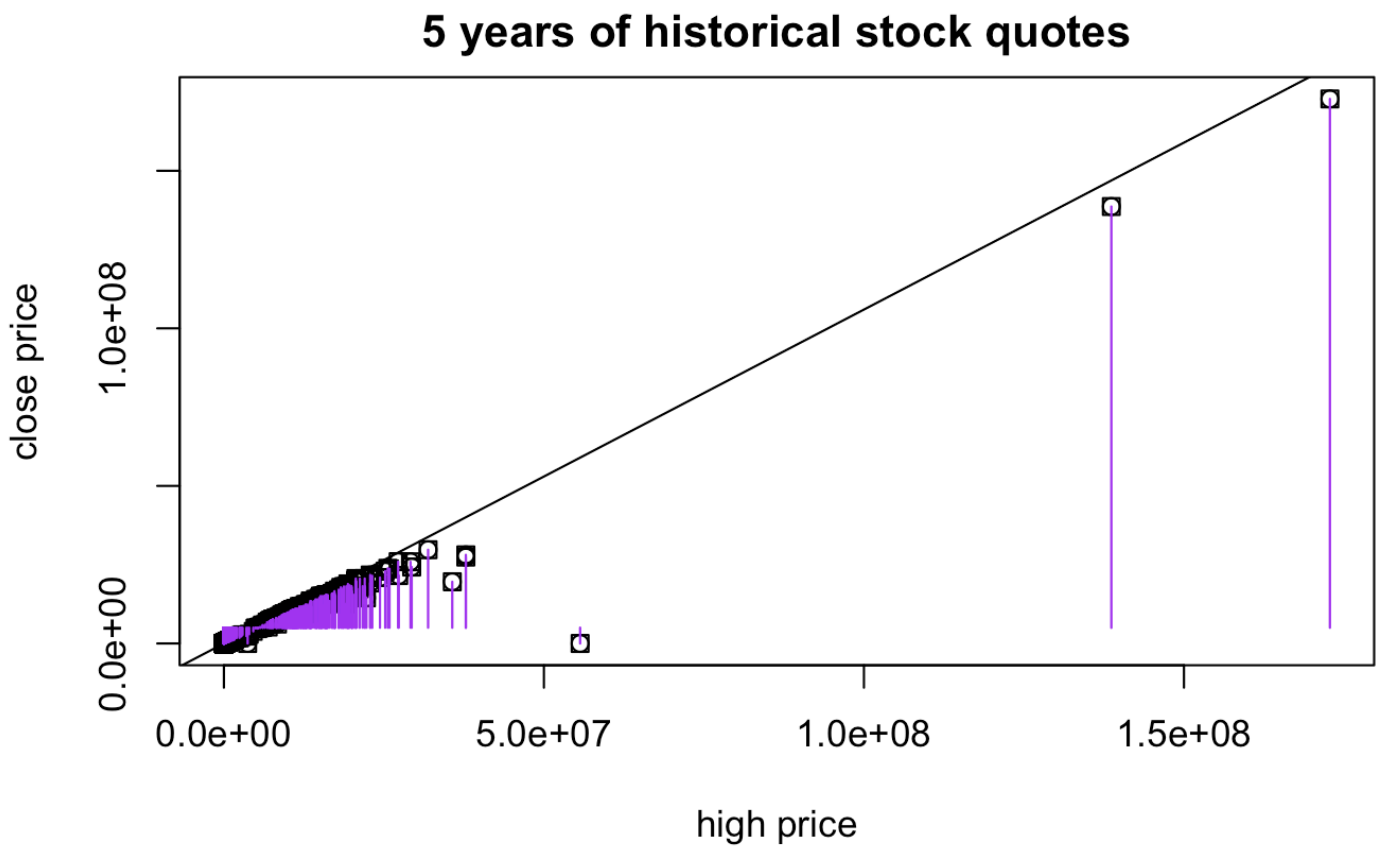


Hide

```
plot(original$high_price, original$close_price, main="5 years of historical stock quo
tes",
     xlab="high price", ylab="close price")
abline(lm3)
```

Hide

```
points(original$high_price, original$close_price, pch=0)
segments(original$high_price, original$close_price, original$high_price, pred, col="p
urple")
```



I created a multiple linear regression model with multiple predictors and did an output on the summary and plotted the residuals for each regression model. Similar to the previous linear regression model where I plotted the residuals, I am shown very similar plots with multiple predictors as well. Both plots have a linear relationship because the plots are surrounded by the straight line and moving in a positive direction. They both fit the model very well and are also classified as a Normal Q-Q because of the way the plots are surrounding the line. The summary also gave me similar results too with the coefficients, the residual standard error, mulitple R-squared, and the F-statistic. For my plots I used the high price for a stock symbol, the price it was at when the market closed, and the price when the market opened.

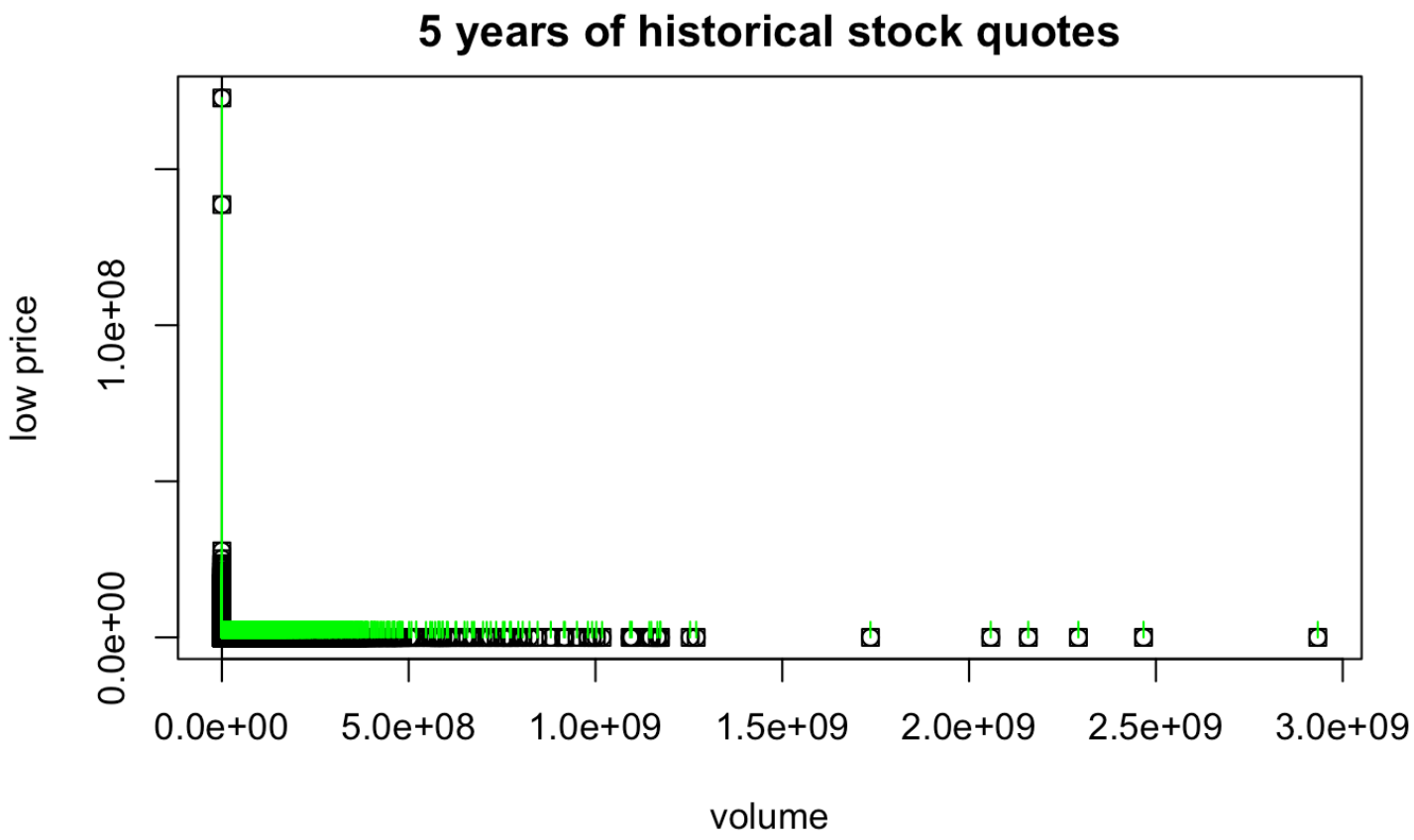
g. Third linear regression model with different predictors

Hide

```
plot(original$volume, original$low_price, main="5 years of historical stock quotes",
     xlab="volume", ylab="low price")
abline(lm1)
```

Hide

```
points(original$volume, original$low_price, pch=0)
segments(original$volume, original$low_price, original$volume, pred, col="green")
```



Hide

```
lm4 <- lm(y~volume+high_price)
```

```
Error in eval(predvars, data, env) : object 'y' not found
```

h. Comparing results

When creating my third linear regression model, I noticed this was the first time I did not get a linear relationship with my variables and the plot looked different. For volume and low price, I noticed that the points on the plot do not increase much and stays low on the graph. This showed me that the two variables do not have a linear relationship. The points are chunked together and in a straight line but do not increase in a positive direction. Although my points were in a straight line, I went ahead and did a polynomial regression to see what my data would look like. I used two variables, volume and high price with a for loop to plot the models of the degree. After looking at both results, I still do think that the linear regression model made the most sense with my data because I still saw the points line up on a line whereas the polynomial regression did not fit the model as well. I know with every other combination of predictors and variables I used, my linear regression model was always a linear relationship which is why I think this model works the best. And I now know that the volume and low price variable have a non-linear relationship.

i. Predict and evaluate test data

Hide

```
pred <- predict(lm2, newdata=test)

cor <- cor(pred, original$high_price)
cor

mse <- mean((pred - original$high_price)^2)
mse

#test data
lm2 <- lm(original$high_price~original$low_price, data=pred)
summary(lm2)
```

I did a summary output of the linear regression using the predicted data to see how the computer's estimation came to be. I also used the correlation metric and the mse to better understand the data. The correlation is used to see how much change in one variable can effect the change in another variable. The two variables are correlated with one another because if the price is too low, the high price will not be as high for the stock price and vice versa. The mse showed us the mean squared error which is useful because we are comparing two models from the same data. I think these results happend because both variables have a linear relatioship to one another, based on the linear regression models used, and they are both correlated based on the summary shown.