Divya Shridar SPY Stocks Final Project BSDS100

In this data set, we will find out if we can predict the SPY symbol (SPDR S&P 500) on the NYSE Arca stock market. We will try to predict the Date as a linear function of Open, Close, High, Volume and Low values.

This data set interested me for numerous reasons. Growing up, my parents were always very invested in trading and the stock markets. When this project was presented to our Data Science class, I thought this would be the perfect opportunity to dip my toes into the field of stock investment, and decided to choose this as my topic, as it gave me time and an opportunity to get interested. Additionally, though stocks are just a fraction of the representation of the economy, the stock market was massively impacted during the pandemic, and resulted in enormous fluctuation in the market trends. I thought this would make it much more interesting to analyze.

About the data set: Contains values from November 28th, 2016, to November 26th, 2021.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
SPY_df<-read_csv('SPY.csv')
```

```
## Rows: 1259 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (6): Open, High, Low, Close, Adj Close, Volume
## date (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
nrow(SPY_df) #this tells us the observations number
```

```
## [1] 1259
```

```
summary(SPY_df) #this tells us the min, max, median, mean, 1st and 3rd quantile
```

```
##       Date                Open            High            Low
##  Min.   :2016-11-28   Min.   :219.7   Min.   :220.2   Min.   :218.3
##  1st Qu.:2018-02-28   1st Qu.:263.6   1st Qu.:265.4   1st Qu.:261.2
##  Median :2019-05-31   Median :287.3   Median :288.8   Median :285.6
##  Mean   :2019-05-30   Mean   :307.0   Mean   :308.5   Mean   :305.2
##  3rd Qu.:2020-08-27   3rd Qu.:336.0   3rd Qu.:337.9   3rd Qu.:333.3
##  Max.   :2021-11-26   Max.   :470.9   Max.   :473.5   Max.   :468.5
```

```
##      Close          Adj Close          Volume
##  Min.   :219.6   Min.   :200.5   Min.   : 20270000
##  1st Qu.:263.6   1st Qu.:248.4   1st Qu.: 55154250
##  Median :287.6   Median :275.0   Median : 69798000
##  Mean   :307.0   Mean   :296.0   Mean   : 82310930
##  3rd Qu.:335.7   3rd Qu.:328.8   3rd Qu.: 93588450
##  Max.   :469.7   Max.   :469.7   Max.   :392220700
```

```
#for each variable in the data set
spy_new<-SPY_df%>%
  mutate(day_num=1:n())%>%
  select(-`Adj Close`)%>%
  select(-Date)
#^Here we are altering the data set into a new one containing solely numeric values
#(instead of the date format,
#it is a day counter with
#day 1=11/28/2016, and we
#remove irrelevant columns like Adj Close)


#Below is some relevant information regarding the new, altered data set
#observe that all the features are numerical
nrow(spy_new)
```

```
## [1] 1259
```

```
ncol(spy_new)
```

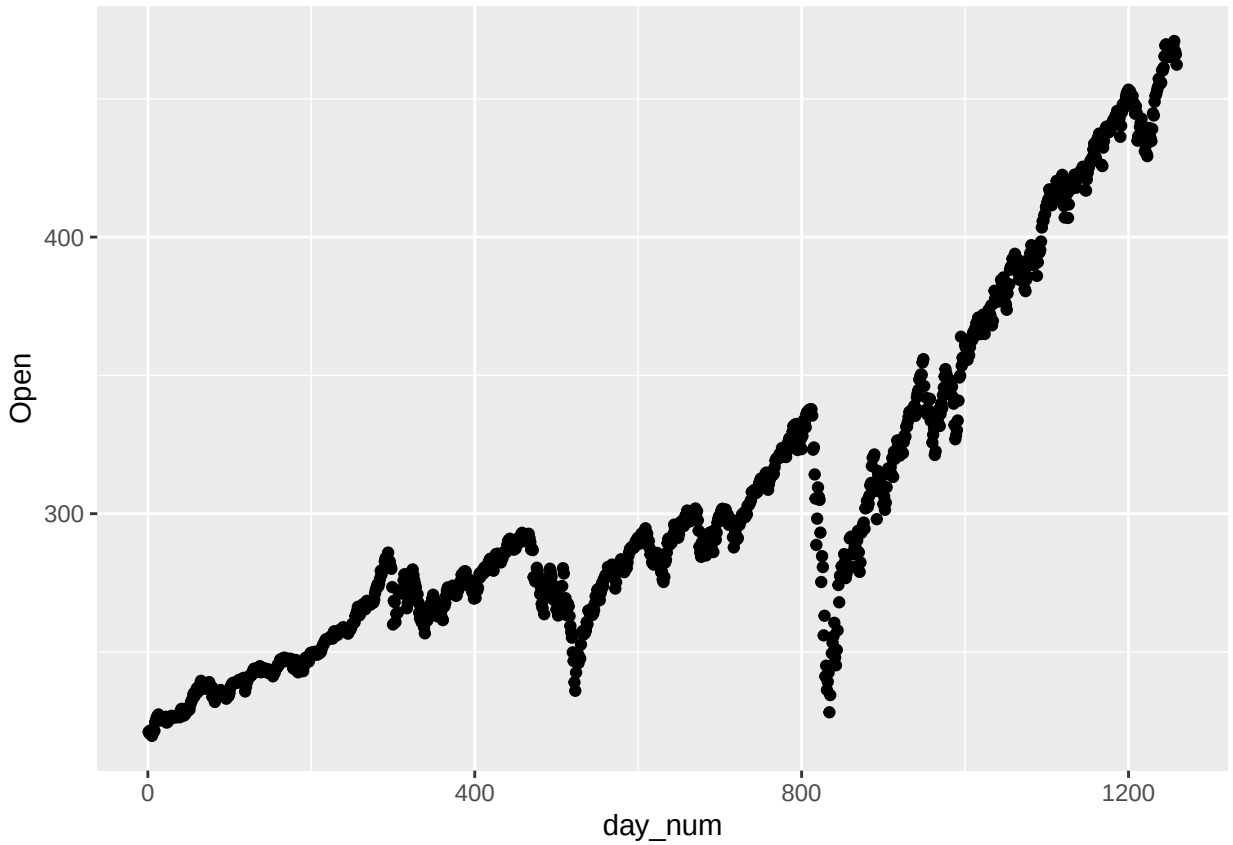
```
## [1] 6
```

```
summary(spy_new)
```

```
##       Open             High             Low             Close
##  Min.   :219.7   Min.   :220.2   Min.   :218.3   Min.   :219.6
##  1st Qu.:263.6   1st Qu.:265.4   1st Qu.:261.2   1st Qu.:263.6
##  Median :287.3   Median :288.8   Median :285.6   Median :287.6
##  Mean   :307.0   Mean   :308.5   Mean   :305.2   Mean   :307.0
##  3rd Qu.:336.0   3rd Qu.:337.9   3rd Qu.:333.3   3rd Qu.:335.7
##  Max.   :470.9   Max.   :473.5   Max.   :468.5   Max.   :469.7
##      Volume             day_num
##  Min.   : 20270000   Min.   :   1.0
##  1st Qu.: 55154250   1st Qu.: 315.5
##  Median : 69798000   Median : 630.0
##  Mean   : 82310930   Mean   : 630.0
##  3rd Qu.: 93588450   3rd Qu.: 944.5
##  Max.   :392220700   Max.   :1259.0
```

We now plot Date (or day number) against the Open value (which is the price the stock is valued at when market opens everyday). After this, we find the correlation coefficient between the day number/date and the open value each day.

```
spy_new%>%
  ggplot(aes(x=day_num, y=Open))+
  geom_point()
```



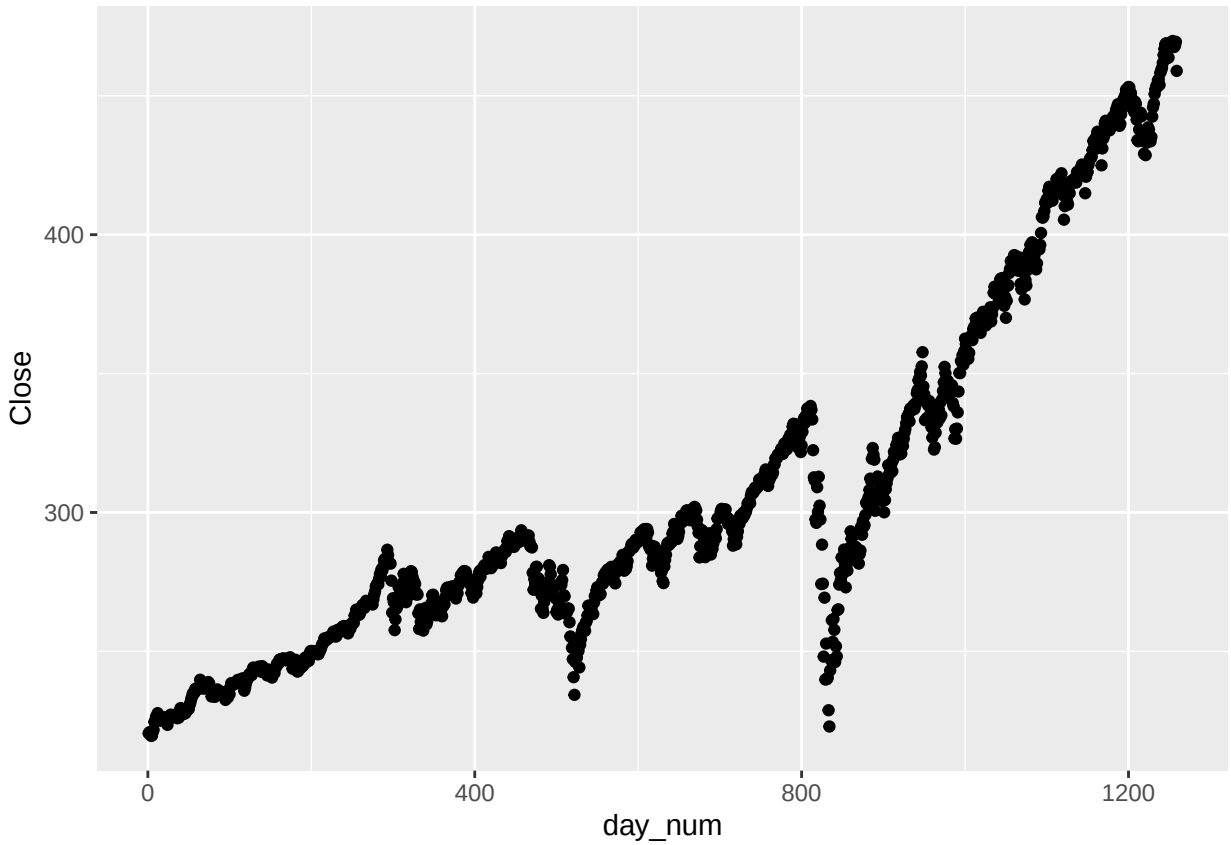

```
cor(spy_new$day_num,spy_new$Open)
```

```
## [1] 0.9036518
```

```
#0.9036518 this tells us there is a strong linear association between the date and open values.
```

We then plot Date (or day number) against the Close value (which is the price the stock is valued at when market closes everyday). After plotting these variables against each other, we find the correlation coefficient between the day number/date and the close value each day.

```
spy_new%>%
  ggplot(aes(x=day_num, y=Close))+
  geom_point()
```



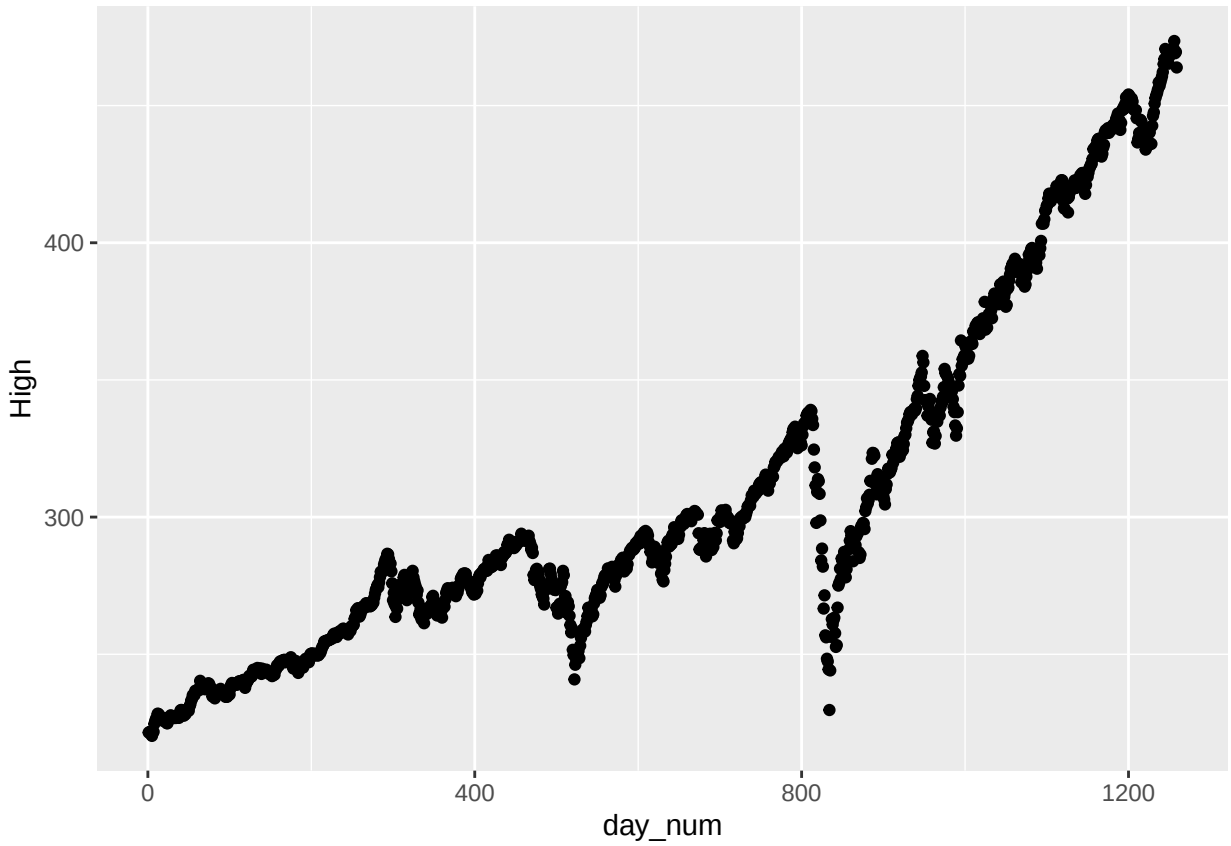

```
cor(spy_new$day_num,spy_new$Close)
```

```
## [1] 0.9037056
```

```
# 0.9037056 this tells us there is a strong linear association between the date and closing values.
```

After this, we plot Date (or day number) against the High value (which is the highest price the stock is valued at in the market during the interval of one day). After plotting these variables against one another, we find a correlation coefficient as shown below.

```
spy_new%>%
  ggplot(aes(x=day_num, y=High))+
  geom_point()
```



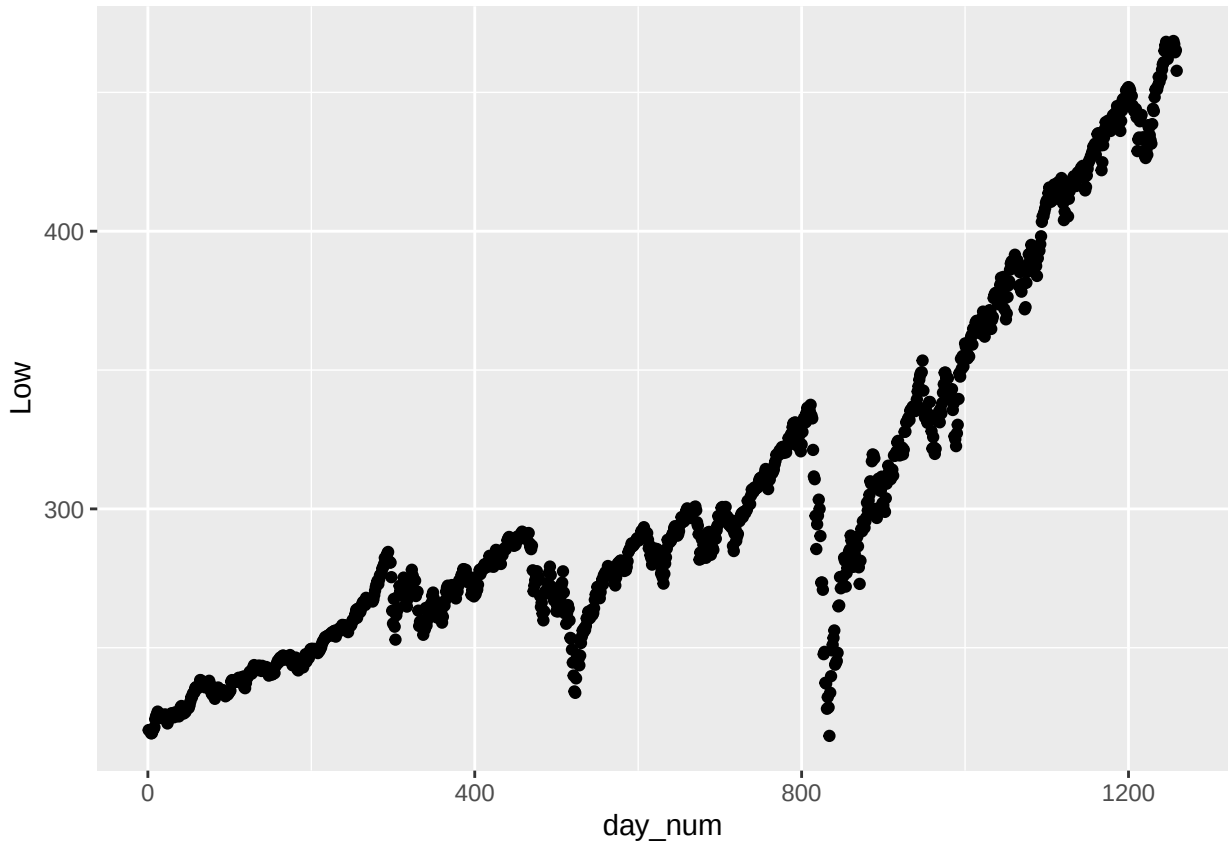

```
cor(spy_new$day_num,spy_new$High)
```

```
## [1] 0.908725
```

```
# 0.908725 this tells us there is a strong linear association between the date and high values.
```

Now, we plot Date (or day number) against the Low value (which is the lowest price the stock is valued at in the market during the interval of one day).

```
spy_new%>%
  ggplot(aes(x=day_num, y=Low))+
  geom_point()
```



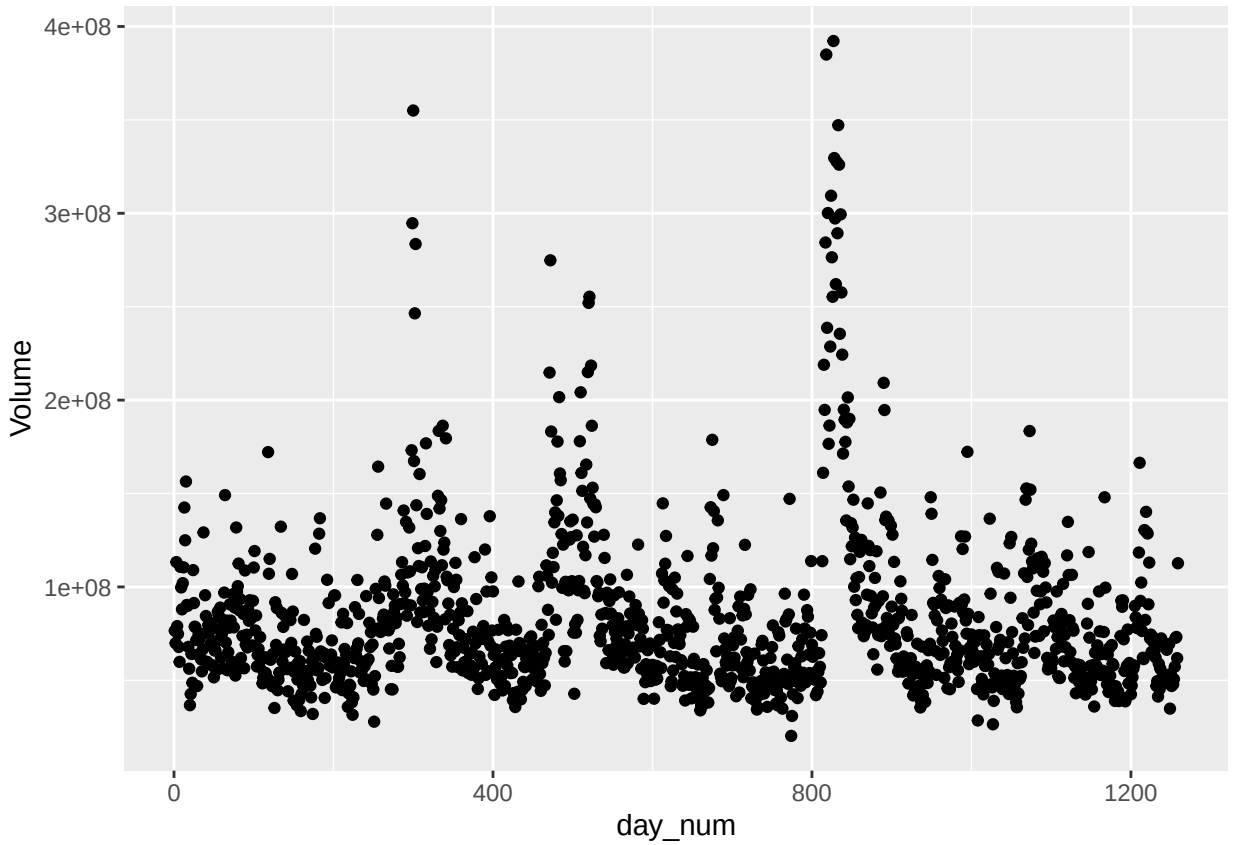

```
cor(spy_new$day_num,spy_new$Low)
```

```
## [1] 0.8985178
```

```
# 0.8985178 this tells us there is a strong linear association between the date and low values. However
```

Lastly, we plot Date (or day number) against the Volume value (which is the number of shares traded in a stock).

```
spy_new%>%
  ggplot(aes(x=day_num, y=Volume))+
  geom_point()
```




```
cor(spy_new$day_num,spy_new$Volume)
```

```
## [1] -0.01144017
```

```
# -0.01144017 this tells us there is not a strong linear association (and if so, there
#is a slightly negative correlation)
#between the date and volume values.
#Additionally, when we plot this, it is very scattered. But the fact that the
#volume numbers are high allows us to
#infer that this is a stable stock to
#invest in (as the number of shares
#traded in the stock are relatively high).
```

Now let's find a linear regression model that will predict Day_num as a function of Open, Close, High, Low and Volume.

```
linear_model_adv<-lm(day_num~Open+Close+High+Low+Volume,spy_new)
linear_model_adv
```

```
##
## Call:
## lm(formula = day_num ~ Open + Close + High + Low + Volume, data = spy_new)
##
## Coefficients:
## (Intercept)         Open        Close         High          Low       Volume
##  -8.802e+02   -6.567e+00    8.564e-01    5.163e+01   -4.119e+01   -1.111e-06
```

```
#Coefficients:
#(Intercept)         Open        Close         High          Low       Volume
```

```
#-8.802e+02   -6.567e+00    8.564e-01    5.163e+01   -4.119e+01   -1.111e-06
```

Multiple Linear Regression Model:

```
str(spy_new)
```

```
## tibble [1,259 x 6] (S3: tbl_df/tbl/data.frame)
##  $ Open   : num [1:1259] 221 221 222 221 220 ...
##  $ High   : num [1:1259] 221 221 222 221 220 ...
##  $ Low    : num [1:1259] 220 220 220 219 219 ...
##  $ Close  : num [1:1259] 220 221 220 220 220 ...
##  $ Volume : num [1:1259] 7.66e+07 6.99e+07 1.13e+08 7.90e+07 7.48e+07 ...
##  $ day_num: int [1:1259] 1 2 3 4 5 6 7 8 9 10 ...
```

```
str(SPY_df)
```

```
## spec_tbl_df [1,259 x 7] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Date     : Date[1:1259], format: "2016-11-28" "2016-11-29" ...
##  $ Open     : num [1:1259] 221 221 222 221 220 ...
##  $ High     : num [1:1259] 221 221 222 221 220 ...
##  $ Low      : num [1:1259] 220 220 220 219 219 ...
##  $ Close    : num [1:1259] 220 221 220 220 220 ...
##  $ Adj Close: num [1:1259] 201 202 201 200 201 ...
##  $ Volume   : num [1:1259] 7.66e+07 6.99e+07 1.13e+08 7.90e+07 7.48e+07 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Date = col_date(format = ""),
##   ..   Open = col_double(),
##   ..   High = col_double(),
##   ..   Low = col_double(),
##   ..   Close = col_double(),
##   ..   `Adj Close` = col_double(),
##   ..   Volume = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
summary(linear_model_adv)
```

```
##
## Call:
## lm(formula = day_num ~ Open + Close + High + Low + Volume, data = spy_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -533.54 -104.04  -30.27  108.03  516.69
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.802e+02  2.360e+01 -37.296  < 2e-16 ***
## Open        -6.567e+00  3.070e+00  -2.139   0.0326 *
## Close        8.564e-01  2.762e+00   0.310   0.7566
## High         5.163e+01  3.548e+00  14.550  < 2e-16 ***
## Low         -4.119e+01  3.319e+00 -12.410  < 2e-16 ***
## Volume      -1.112e-06  1.484e-07  -7.491 1.29e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 126.8 on 1253 degrees of freedom
## Multiple R-squared:  0.8788, Adjusted R-squared:  0.8783
## F-statistic:  1817 on 5 and 1253 DF,  p-value: < 2.2e-16
```

```
lin_model2<-lm(day_num~.-Volume,spy_new)
summary(lin_model2)
```

```
##
## Call:
## lm(formula = day_num ~ . - Volume, data = spy_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -407.84 -104.21  -31.16  111.91  390.04
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -996.932     18.112 -55.042   <2e-16 ***
## Open          -6.838      3.137  -2.180   0.0294 *
## High          35.883      2.921  12.285   <2e-16 ***
## Low          -27.345      2.817  -9.708   <2e-16 ***
## Close          3.265      2.803   1.165   0.2443
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 129.6 on 1254 degrees of freedom
## Multiple R-squared:  0.8733, Adjusted R-squared:  0.8729
## F-statistic:  2162 on 4 and 1254 DF,  p-value: < 2.2e-16
```

```
#day_num=-8.802e+02+-6.567e+00*Open+8.564e-01*Close+5.163e+01*High+-4.119e+01*Low+-1.112e-06*Volume

day_num=25
Open=225.04
High=225.83
Low=223.88
Close=225.24
Volume=91366500

single_obs<-data.frame(day_num=25,Open=225.04,High=225.83,Low=223.88,Close=225.24,Volume=91366500)
single_obs
```

```
##   day_num   Open   High    Low  Close   Volume
## 1      25 225.04 225.83 223.88 225.24 91366500
```

Our prediction for the charges for this single observation:

```
predict(lin_model2,single_obs) #181.1431
```

```
##        1
## 181.1431
```

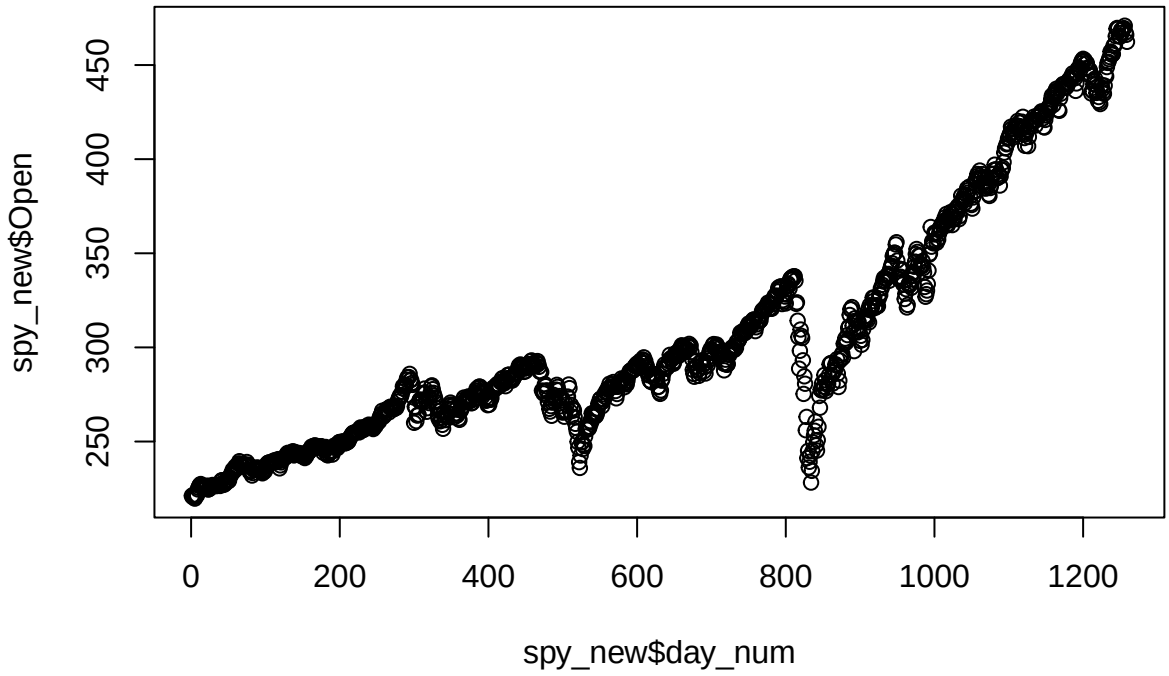
```
#Long way to obtain the prediction:
-8.802e+02+-6.567e+00*Open+8.564e-01*Close+5.163e+01*High+-4.119e+01*Low+-1.112e-06*Volume
```

```
## [1] 171.244
```

```
#181.1431
```

Nonlinearity:

```
plot(spy_new$day_num,spy_new$Open)
```




```
model_open1<-lm(Open~day_num,spy_new)
summary(model_open1)
```

```
##
## Call:
## lm(formula = Open ~ day_num, data = spy_new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -110.969  -17.935    5.979   14.309   65.515
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 207.52962    1.53447  135.25   <2e-16 ***
## day_num       0.15783    0.00211   74.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.21 on 1257 degrees of freedom
## Multiple R-squared:  0.8166, Adjusted R-squared:  0.8164
## F-statistic:  5596 on 1 and 1257 DF,  p-value: < 2.2e-16
```

```
#The R-squared is 0.8166,
#which is fairly high, allowing
#us to infer that the regression
#model fits the observed data well.
#This also shows us that as time
#passes, the open value for the stock increases.
```

```r
model_open2<-lm(Open~day_num+I(day_num^2),spy_new)
summary(model_open2)
```

```
##
## Call:
## lm(formula = Open ~ day_num + I(day_num^2), data = spy_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -96.127  -8.864   2.471  13.316  35.147
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.510e+02  1.616e+00 155.272  < 2e-16 ***
## day_num      -4.887e-02  5.925e-03  -8.248 4.03e-16 ***
## I(day_num^2)  1.640e-04  4.553e-06  36.029  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.09 on 1256 degrees of freedom
## Multiple R-squared:  0.9098, Adjusted R-squared:  0.9097
## F-statistic:  6335 on 2 and 1256 DF,  p-value: < 2.2e-16
```

```r
#The R-squared is 0.9098, which
#is quite high. This tells us that
#the regression model also fits the
#observed data well. This also shows
#us that as time passes, the open
#value for the stock increases.
model_open3<-lm(Open~day_num+I(day_num^2)+I(day_num^3),spy_new)
summary(model_open3)
```

```
##
## Call:
## lm(formula = Open ~ day_num + I(day_num^2) + I(day_num^3), data = spy_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -82.413  -4.951   1.631   8.231  31.330
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.166e+02  1.587e+00  136.50   <2e-16 ***
## day_num       2.777e-01  1.090e-02   25.47   <2e-16 ***
## I(day_num^2) -4.837e-04  2.010e-05  -24.06   <2e-16 ***
## I(day_num^3)  3.427e-07  1.049e-08   32.68   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.04 on 1255 degrees of freedom
## Multiple R-squared:  0.9513, Adjusted R-squared:  0.9512
## F-statistic:  8166 on 3 and 1255 DF,  p-value: < 2.2e-16
```

```r
#The R-squared is 0.9513, which is very high, allowing us to conclude that
#the regression model fits the observed
```

```
#data well. This also shows us that as
#time passes, the open value for the
#stock increases. As you can see, which
#each model_open variation, the r-squared
#gets higher, and we become more precise
#with the calculations. This shows us that
#the more accurate we are, the higher
#r-squared we will see (in this scenario),
#and the better correlation the variables have.
model_open4<-lm(Open~day_num+I(log(day_num)),spy_new)
summary(model_open4)
```

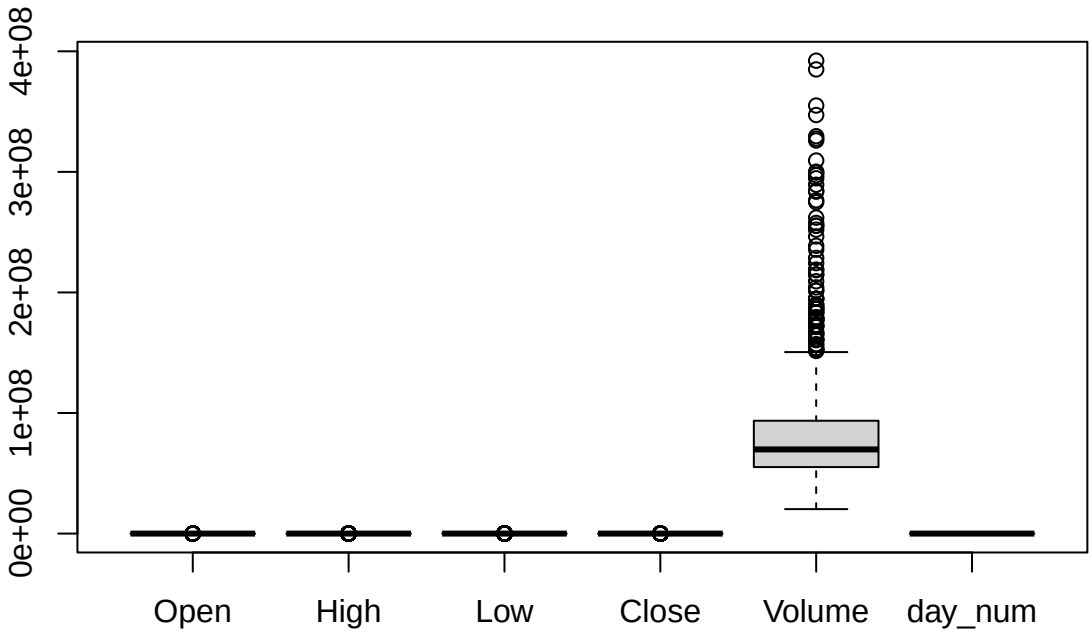
```
##
## Call:
## lm(formula = Open ~ day_num + I(log(day_num)), data = spy_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -109.02  -14.31    4.00   16.30   56.38
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     297.182233   7.086536   41.94   <2e-16 ***
## day_num           0.203528   0.004055   50.20   <2e-16 ***
## I(log(day_num)) -19.285238   1.492488  -12.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.57 on 1256 degrees of freedom
## Multiple R-squared:  0.8381, Adjusted R-squared:  0.8378
## F-statistic:  3251 on 2 and 1256 DF,  p-value: < 2.2e-16
```

```
#The R-squared is 0.8381, which is
#relatively high, allowing us to deduce
#that the regression model fits the
#observed data pretty well. This also
#shows us that as time passes, the
#open value for the stock increases.
```

I attempted to remove the outliers from the table, and re-plotting the data. However, with a correlation coefficient so high, it was safe to assume (and I put this theory to the test as well) that there were not really any outliers, and even the micro outliers that existed did not impact the data set or regression models at all.

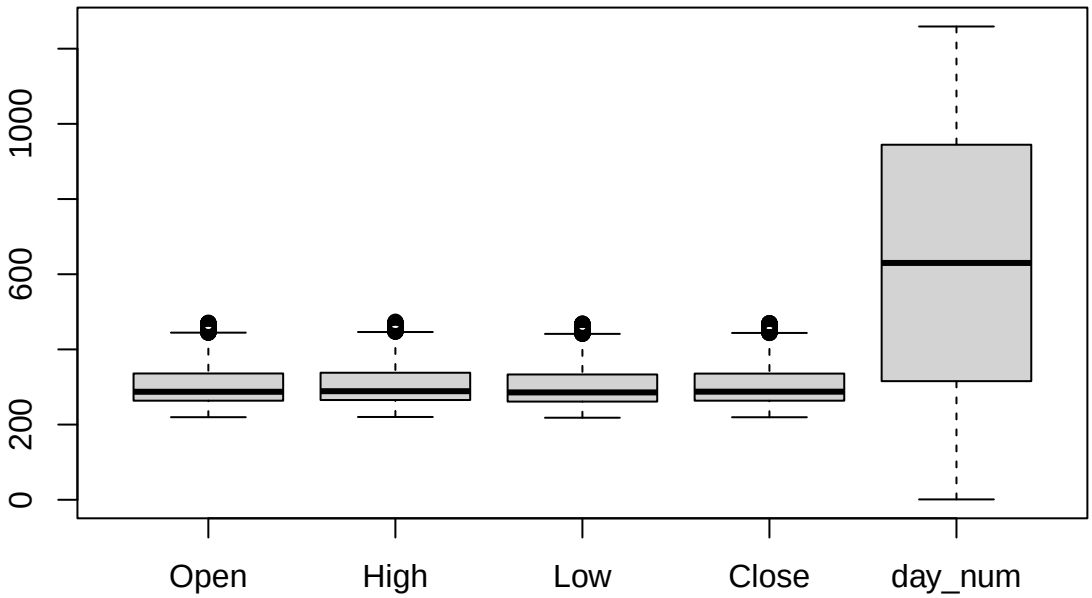
Boxplot:

```
boxplot(spy_new)
```




```
#^As you can see, volume is very different from the other variables, so we can try to remove
#volume and compare everything else in a boxplot

spy_noVolume<-spy_new%>%select(-Volume)
a<-boxplot(spy_noVolume)
```




```
summary(a) #this provides a summary of the boxplot with no volume
```

```
##       Length Class  Mode
## stats  25    -none- numeric
## n       5    -none- numeric
## conf   10    -none- numeric
## out   205    -none- numeric
## group 205    -none- numeric
## names   5    -none- character
```

Some general conclusions we can make from the graphs are that around day 800 (which is March 2020-when the Covid 19 pandemic started), there was a massive dip in all the graphs, but now, the values are at an all time high as they skyrocketed in the midst of the pandemic. Though most of the graphs and the values look very similar, this is logical as this time series analysis SPDR S&P 500 is the open/close/high/low values of each day, and realistically the values do not fluctuate much in a 24 hour window. Therefore, when you compare it on a 24 hour basis, there should not be much variation between the graphs. However, the dataset initially contained data from the past 10 years. My computer could not handle this much data, so I cut it down to the past 5 years. Finally, since this was a fairly new concept for me, in the beginning, I had a bit of difficulty figuring out the meaning behind each value. But after extensive research, I was able to grasp the meaning behind each value, and properly analyze the data.