Data Science project

Seth Holtzman

2023-02-17

```r
library(readr) #loads functions to read CSV file
library(dplyr) #loads data wrangling functions
```

```
##
## 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(lubridate) #loads library for date functions
```

```
##
## Attaching package: 'lubridate'

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

```r
library(ggplot2) #loads library for creating graphs
```

```r
dataset <- read_csv("ACC_with_indicators_.csv") #creates a new dataset from a CSV file
```

```
## Rows: 130348 Columns: 59
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (58): open, high, low, close, volume, sma5, sma10, sma15, sma20, ema5, ...
## dttm  (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.
```

```r
glimpse(dataset) # give basic information about the dataset
```

```
## Rows: 130,348
## Columns: 59
## $ date         <dttm> 2015-02-02 09:00:00, 2015-02-02 09:05:00, 2015-02-02 09:~
## $ open         <dbl> 1528.50, 1527.40, 1521.30, 1520.65, 1521.20, 1526.40, 152~
## $ high         <dbl> 1529.95, 1528.00, 1526.70, 1522.90, 1526.10, 1529.00, 152~
## $ low          <dbl> 1526.05, 1516.00, 1521.00, 1519.80, 1516.25, 1524.90, 152~
## $ close        <dbl> 1527.40, 1521.95, 1521.55, 1520.25, 1526.10, 1525.85, 152~
## $ volume       <dbl> 4678, 10165, 8078, 4733, 4636, 6921, 4016, 3411, 4339, 45~
## $ sma5         <dbl> 1538.82, 1532.81, 1527.52, 1523.93, 1523.45, 1523.14, 152~
```

```
## $ sma10       <dbl> 1543.015, 1540.670, 1538.205, 1535.725, 1533.440, 1530.98~
## $ sma15       <dbl> 1542.017, 1541.213, 1540.317, 1538.997, 1537.407, 1536.39~
## $ sma20       <dbl> 1539.838, 1539.285, 1538.723, 1538.125, 1537.680, 1537.29~
## $ ema5        <dbl> 1535.519, 1530.996, 1527.847, 1525.315, 1525.577, 1525.66~
## $ ema10       <dbl> 1539.097, 1535.980, 1533.356, 1530.973, 1530.087, 1529.31~
## $ ema15       <dbl> 1539.604, 1537.397, 1535.416, 1533.520, 1532.593, 1531.75~
## $ ema20       <dbl> 1539.456, 1537.789, 1536.242, 1534.719, 1533.898, 1533.13~
## $ upperband   <dbl> 1558.726, 1551.261, 1539.577, 1530.626, 1529.016, 1527.90~
## $ middleband  <dbl> 1538.82, 1532.81, 1527.52, 1523.93, 1523.45, 1523.14, 152~
## $ lowerband   <dbl> 1518.914, 1514.359, 1515.463, 1517.234, 1517.884, 1518.37~
## $ HT_TRENDLINE <dbl> 1538.860, 1538.309, 1537.711, 1537.118, 1536.611, 1536.12~
## $ KAMA10      <dbl> 1540.020, 1536.725, 1533.658, 1530.938, 1530.152, 1529.31~
## $ KAMA20      <dbl> 1541.903, 1541.425, 1540.940, 1540.403, 1540.148, 1539.92~
## $ KAMA30      <dbl> 1537.003, 1536.848, 1536.707, 1536.488, 1536.414, 1536.30~
## $ SAR         <dbl> 1556.7, 1556.7, 1556.7, 1556.7, 1556.7, 1556.7, 1556.7, 1~
## $ TRIMA5      <dbl> 1538.556, 1531.850, 1526.983, 1523.611, 1522.506, 1522.65~
## $ TRIMA10     <dbl> 1545.733, 1543.333, 1539.832, 1535.672, 1531.660, 1528.27~
## $ TRIMA20     <dbl> 1542.356, 1542.721, 1542.749, 1542.429, 1541.785, 1540.79~
## $ ADX5        <dbl> 63.50025, 65.22281, 66.60086, 68.08255, 70.31235, 65.9936~
## $ ADX10       <dbl> 43.08269, 43.02950, 42.98162, 43.15536, 43.92183, 43.2383~
## $ ADX20       <dbl> 23.56147, 23.51581, 23.47244, 23.52041, 23.81910, 23.7502~
## $ APO         <dbl> 5.3641026, 3.3846154, 2.1554487, 0.5192308, -0.9365385, -~
## $ CCI5        <dbl> -99.35118, -95.62378, -69.27688, -74.05418, -19.27861, 16~
## $ CCI10       <dbl> -177.940526, -152.686623, -103.639049, -91.253096, -67.23~
## $ CCI15       <dbl> -158.980673, -187.520270, -147.915094, -129.167972, -95.6~
## $ macd510     <dbl> -3.57851714, -4.98376750, -5.50882724, -5.65832485, -4.51~
## $ macd520     <dbl> -3.9370330, -6.7927615, -8.3948593, -9.4042243, -8.321643~
## $ macd1020    <dbl> -0.3585282, -1.8090041, -2.8860403, -3.7459062, -3.811072~
## $ macd1520    <dbl> 0.1491268, -0.3905232, -0.8250017, -1.1978330, -1.3046239~
## $ macd1226    <dbl> 0.2310039, -1.1830192, -2.3092992, -3.2690992, -3.5171579~
## $ MOM10       <dbl> -13.70, -23.45, -24.65, -24.80, -22.85, -24.60, -26.45, -~
## $ MOM15       <dbl> -6.10, -12.05, -13.45, -19.80, -23.85, -15.25, -19.85, -1~
## $ MOM20       <dbl> -4.60, -11.05, -11.25, -11.95, -8.90, -7.65, -8.45, -4.80~
## $ ROC5        <dbl> -1.48666516, -1.93621134, -1.70865633, -1.16694838, -0.15~
## $ ROC10       <dbl> -0.888975407, -1.517406497, -1.594231018, -1.605126048, -~
## $ ROC20       <dbl> -0.300261097, -0.720808871, -0.733950939, -0.779924292, -~
## $ PPO         <dbl> 0.34875560, 0.22010675, 0.14020862, 0.03378526, -0.060947~
## $ RSI14       <dbl> 36.78894, 32.80217, 32.52359, 31.58474, 39.98093, 39.7563~
## $ RSI8        <dbl> 28.65076, 24.06281, 23.74389, 22.62985, 37.66994, 37.3156~
## $ slowk       <dbl> 4.838951, 7.147969, 12.588612, 17.267679, 36.098460, 55.7~
## $ slowd       <dbl> 23.965380, 9.649550, 8.191844, 12.334754, 21.984917, 36.3~
## $ fastk       <dbl> 4.936015, 15.909091, 16.920732, 18.973214, 72.401434, 75.~
## $ fastd       <dbl> 4.838951, 7.147969, 12.588612, 17.267679, 36.098460, 55.7~
## $ fastksr     <dbl> 0.00000, 0.00000, 0.00000, 0.00000, 100.00000, 97.32562, ~
## $ fastdsr     <dbl> 0.00000, 0.00000, 0.00000, 0.00000, 33.33333, 65.77521, 9~
## $ ULTOSC      <dbl> 43.34687, 41.44844, 36.64834, 30.13957, 41.14588, 42.4213~
## $ WILLR       <dbl> -95.06399, -84.09091, -85.16043, -88.63636, -72.99465, -7~
## $ ATR         <dbl> 5.282946, 5.762736, 5.758254, 5.568379, 5.874209, 5.74748~
## $ Trange      <dbl> 3.90, 12.00, 5.70, 3.10, 9.85, 4.10, 2.50, 5.45, 1.90, 5.~
## $ TYPPRICE    <dbl> 1527.800, 1521.983, 1523.083, 1520.983, 1522.817, 1526.58~
## $ HT_DCPERIOD <dbl> 25.92900, 25.59547, 25.18456, 25.34973, 26.30800, 27.8824~
## $ BETA        <dbl> 0.47946558, 0.20001935, 0.45094862, 0.56033278, -0.058313~
```

```r
head(dataset) #displays first few rows of the dataset
```

```
## # A tibble: 6 x 59
##   date                 open  high   low close volume  sma5 sma10 sma15 sma20
##   <dttm>              <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 2015-02-02 09:00:00 1528. 1530. 1526. 1527.   4678 1539. 1543. 1542. 1540.
## 2 2015-02-02 09:05:00 1527. 1528  1516  1522.  10165 1533. 1541. 1541. 1539.
## 3 2015-02-02 09:10:00 1521. 1527. 1521  1522.   8078 1528. 1538. 1540. 1539.
## 4 2015-02-02 09:15:00 1521. 1523. 1520. 1520.   4733 1524. 1536. 1539. 1538.
## 5 2015-02-02 09:20:00 1521. 1526. 1516. 1526.   4636 1523. 1533. 1537. 1538.
## 6 2015-02-02 09:25:00 1526. 1529  1525. 1526.   6921 1523. 1531. 1536. 1537.
## # i 49 more variables: ema5 <dbl>, ema10 <dbl>, ema15 <dbl>, ema20 <dbl>,
## #   upperband <dbl>, middleband <dbl>, lowerband <dbl>, HT_TRENDLINE <dbl>,
## #   KAMA10 <dbl>, KAMA20 <dbl>, KAMA30 <dbl>, SAR <dbl>, TRIMA5 <dbl>,
## #   TRIMA10 <dbl>, TRIMA20 <dbl>, ADX5 <dbl>, ADX10 <dbl>, ADX20 <dbl>,
## #   APO <dbl>, CCI5 <dbl>, CCI10 <dbl>, CCI15 <dbl>, macd510 <dbl>,
## #   macd520 <dbl>, macd1020 <dbl>, macd1520 <dbl>, macd1226 <dbl>, MOM10 <dbl>,
## #   MOM15 <dbl>, MOM20 <dbl>, ROC5 <dbl>, ROC10 <dbl>, ROC20 <dbl>, ...
```

```r
dataset2 <- dataset %>%
  dplyr::select(date,open,high,low,close,volume,sma5,sma10,sma20,ema5,ema10,ema20,MOM10,MOM15,MOM20,ROC5
glimpse(dataset2) #gives basic infomrmation about the new dataset
```

```
## Rows: 130,348
## Columns: 19
## $ date   <dttm> 2015-02-02 09:00:00, 2015-02-02 09:05:00, 2015-02-02 09:10:00,~
## $ open   <dbl> 1528.50, 1527.40, 1521.30, 1520.65, 1521.20, 1526.40, 1525.35, ~
## $ high   <dbl> 1529.95, 1528.00, 1526.70, 1522.90, 1526.10, 1529.00, 1527.85, ~
## $ low    <dbl> 1526.05, 1516.00, 1521.00, 1519.80, 1516.25, 1524.90, 1525.35, ~
## $ close  <dbl> 1527.40, 1521.95, 1521.55, 1520.25, 1526.10, 1525.85, 1525.55, ~
## $ volume <dbl> 4678, 10165, 8078, 4733, 4636, 6921, 4016, 3411, 4339, 4538, 73~
## $ sma5   <dbl> 1538.82, 1532.81, 1527.52, 1523.93, 1523.45, 1523.14, 1523.86, ~
## $ sma10  <dbl> 1543.015, 1540.670, 1538.205, 1535.725, 1533.440, 1530.980, 152~
## $ sma20  <dbl> 1539.838, 1539.285, 1538.723, 1538.125, 1537.680, 1537.297, 153~
## $ ema5   <dbl> 1535.519, 1530.996, 1527.847, 1525.315, 1525.577, 1525.668, 152~
## $ ema10  <dbl> 1539.097, 1535.980, 1533.356, 1530.973, 1530.087, 1529.317, 152~
## $ ema20  <dbl> 1539.456, 1537.789, 1536.242, 1534.719, 1533.898, 1533.132, 153~
## $ MOM10  <dbl> -13.70, -23.45, -24.65, -24.80, -22.85, -24.60, -26.45, -17.80,~
## $ MOM15  <dbl> -6.10, -12.05, -13.45, -19.80, -23.85, -15.25, -19.85, -16.00, ~
## $ MOM20  <dbl> -4.60, -11.05, -11.25, -11.95, -8.90, -7.65, -8.45, -4.80, -10.~
## $ ROC5   <dbl> -1.48666516, -1.93621134, -1.70865633, -1.16694838, -0.15701668~
## $ ROC10  <dbl> -0.888975407, -1.517406497, -1.594231018, -1.605126048, -1.4751~
## $ ROC20  <dbl> -0.300261097, -0.720808871, -0.733950939, -0.779924292, -0.5798~
## $ BETA   <dbl> 0.47946558, 0.20001935, 0.45094862, 0.56033278, -0.05831327, 0.~
```

```r
head(dataset2,10) #displays first 10 rows
```

```
## # A tibble: 10 x 19
##    date                 open  high   low close volume  sma5 sma10 sma20  ema5
##    <dttm>              <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 2015-02-02 09:00:00 1528. 1530. 1526. 1527.   4678 1539. 1543. 1540. 1536.
##  2 2015-02-02 09:05:00 1527. 1528  1516  1522.  10165 1533. 1541. 1539. 1531.
##  3 2015-02-02 09:10:00 1521. 1527. 1521  1522.   8078 1528. 1538. 1539. 1528.
##  4 2015-02-02 09:15:00 1521. 1523. 1520. 1520.   4733 1524. 1536. 1538. 1525.
##  5 2015-02-02 09:20:00 1521. 1526. 1516. 1526.   4636 1523. 1533. 1538. 1526.
```

```
##  6 2015-02-02 09:25:00 1526. 1529  1525. 1526.   6921 1523. 1531. 1537. 1526.
##  7 2015-02-02 09:30:00 1525. 1528. 1525. 1526.   4016 1524. 1528. 1537. 1526.
##  8 2015-02-02 09:35:00 1526. 1531  1526. 1530.   3411 1526. 1527. 1537. 1527.
##  9 2015-02-02 09:40:00 1531. 1531. 1529  1529.   4339 1527. 1526. 1536. 1528.
## 10 2015-02-02 09:45:00 1529. 1530. 1524. 1527    4538 1528. 1526. 1535. 1528.
## # i 9 more variables: ema10 <dbl>, ema20 <dbl>, MOM10 <dbl>, MOM15 <dbl>,
## #   MOM20 <dbl>, ROC5 <dbl>, ROC10 <dbl>, ROC20 <dbl>, BETA <dbl>
```

Question #1 How does volume affect the difference between the open and closing price?

First we will do a little bit of changing to the dataset to get the things we need such as a percent change between the open and close price to find the differnce between them.

```
#manipulations of dataset needed later for desired graphs
#goal is to get first price of a day and last price of a day and add percent change column of price
openClose <- dataset2 %>%
  filter((hour(date) == 9 & minute(date) == 55) | (hour(date) == 3 & minute(date) == 45) | (wday(date) =
  dplyr::select(date,open,close,volume) %>%  #keeps only the columns we will use
  mutate(type = if_else(hour(date) == 9,"Open","Close"), # adds column to say whether it is open or clo
         percent_change = (((open-close)/open))*100, #adds columns of percent change between open and c
         percent_change_type = #adds column for percent change type(positive or negative)
           if_else(percent_change>0,"Positive","Negative"),
         ABS_percent_change = abs(percent_change)) # adds column for the absolute value of percent chan
openClose$percent_change_type <- factor(openClose$percent_change_type,levels = c("Positive","Negative")

openClose$type <- factor(openClose$type, levels = c("Open","Close"))
head(openClose,10) #displays first ten rows
```

```
## # A tibble: 10 x 8
##    date                 open close volume type  percent_change
##    <dttm>              <dbl> <dbl>  <dbl> <fct>          <dbl>
##  1 2015-02-02 09:55:00 1525  1515.  17773 Open          0.636
##  2 2015-02-03 03:45:00 1515  1524.  15118 Close        -0.561
##  3 2015-02-03 09:55:00 1512. 1512.  70611 Open          0.0364
##  4 2015-02-04 03:45:00 1512  1512.  12365 Close         0.0231
##  5 2015-02-04 09:55:00 1490. 1490   13522 Open          0.0335
##  6 2015-02-05 03:45:00 1494  1499.   6663 Close        -0.318
##  7 2015-02-05 09:55:00 1504. 1500.   9039 Open          0.246
##  8 2015-02-06 03:45:00 1503  1501.   5719 Close         0.120
##  9 2015-02-06 09:55:00 1510. 1513.   4090 Open         -0.192
## 10 2015-02-09 03:45:00 1500  1504.   3761 Close        -0.247
## # i 2 more variables: percent_change_type <fct>, ABS_percent_change <dbl>
```

```
summary(openClose$percent_change) #gives 6 number summary of percent change column (to help bound data
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -3.44068 -0.20067  0.00290  0.01226  0.21021  4.16552
```

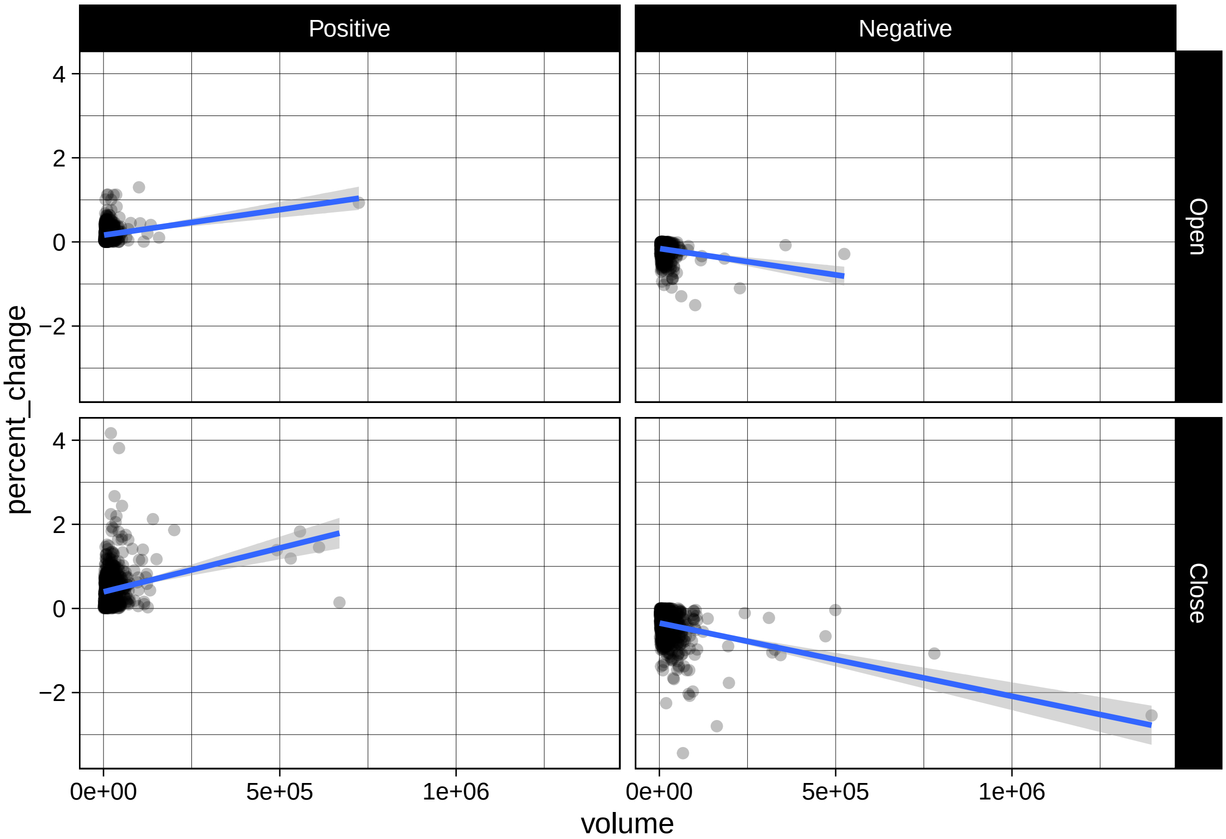
```
summary(openClose$volume) #gives 6 number summary of volume column
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     525    6413   10584   19322   20332 1396017
```

Next we graph our data in a way to hopefully show some correlations

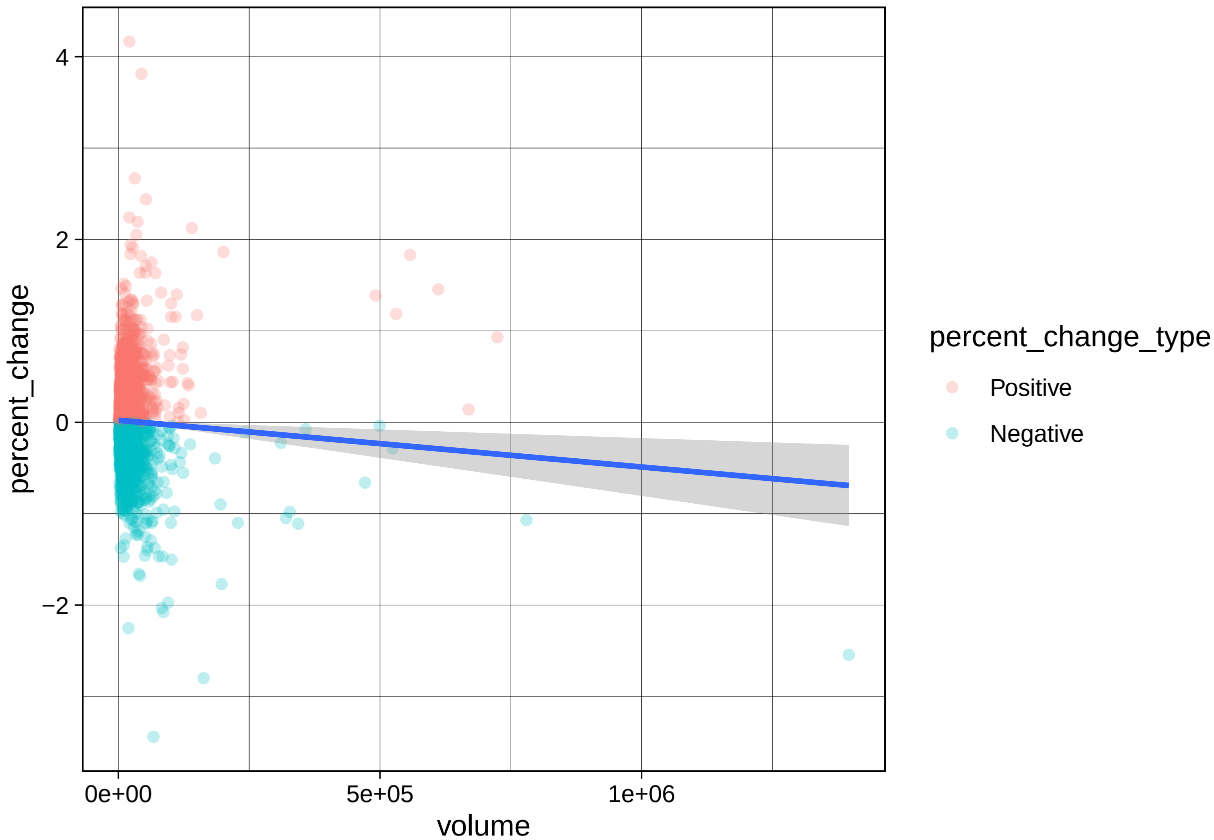
```
VolumeVOpenCloseFacet <- ggplot(data = openClose, aes(x = volume, y = percent_change)) + geom_point(alp

VolumeVOpenCloseFacet #displays graph
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
VolumeVOpenClose <- ggplot(data = openClose, aes(x = volume, y = percent_change)) + geom_point(alpha = 0
VolumeVOpenClose #displays the graph
```

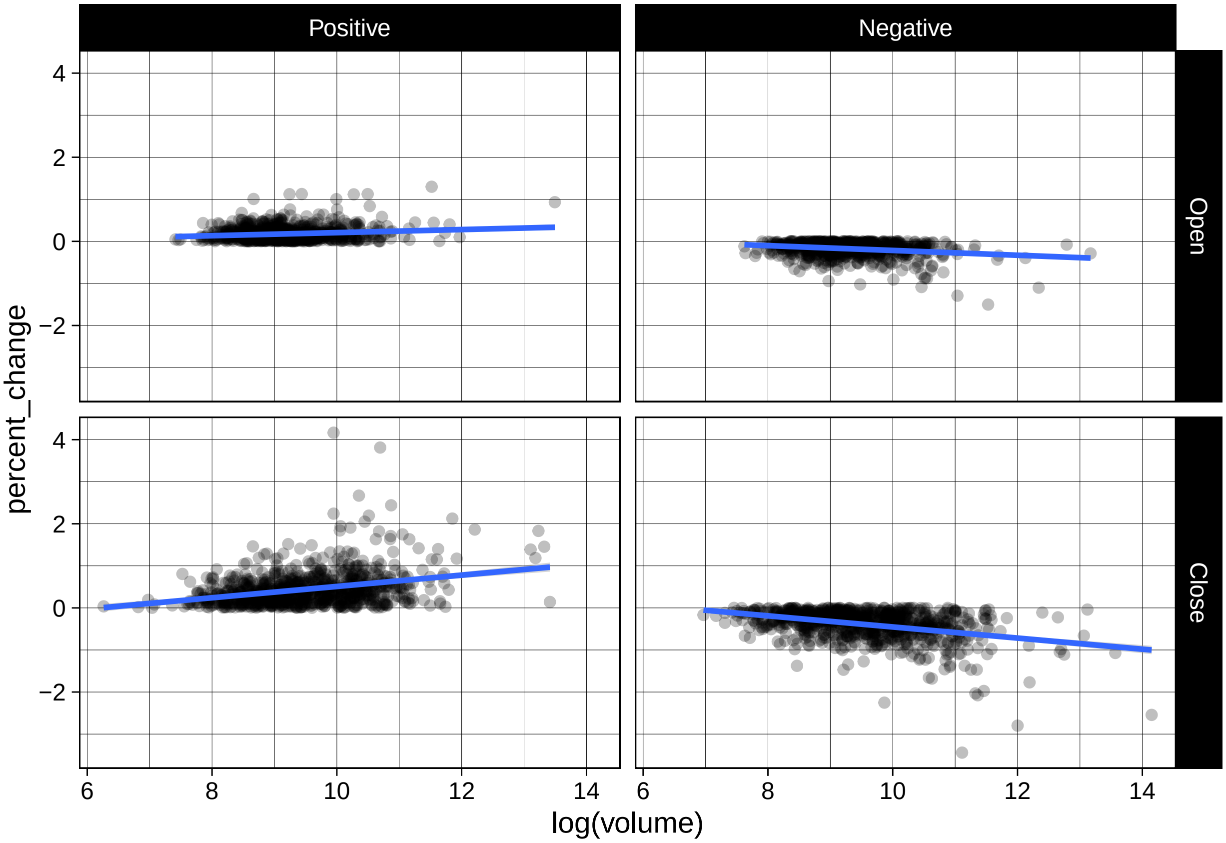
```
## `geom_smooth()` using formula = 'y ~ x'
```



This data is a bit zoomed out so next we try using a logarthmic scale for volume

```
graph1log <- ggplot(data = openClose, aes(x = log(volume), y = percent_change)) + geom_point(alpha = 0.

#plots the log of volume on the x axis and the percent change on the y axis, creates a scatter plot of
graph1log
```

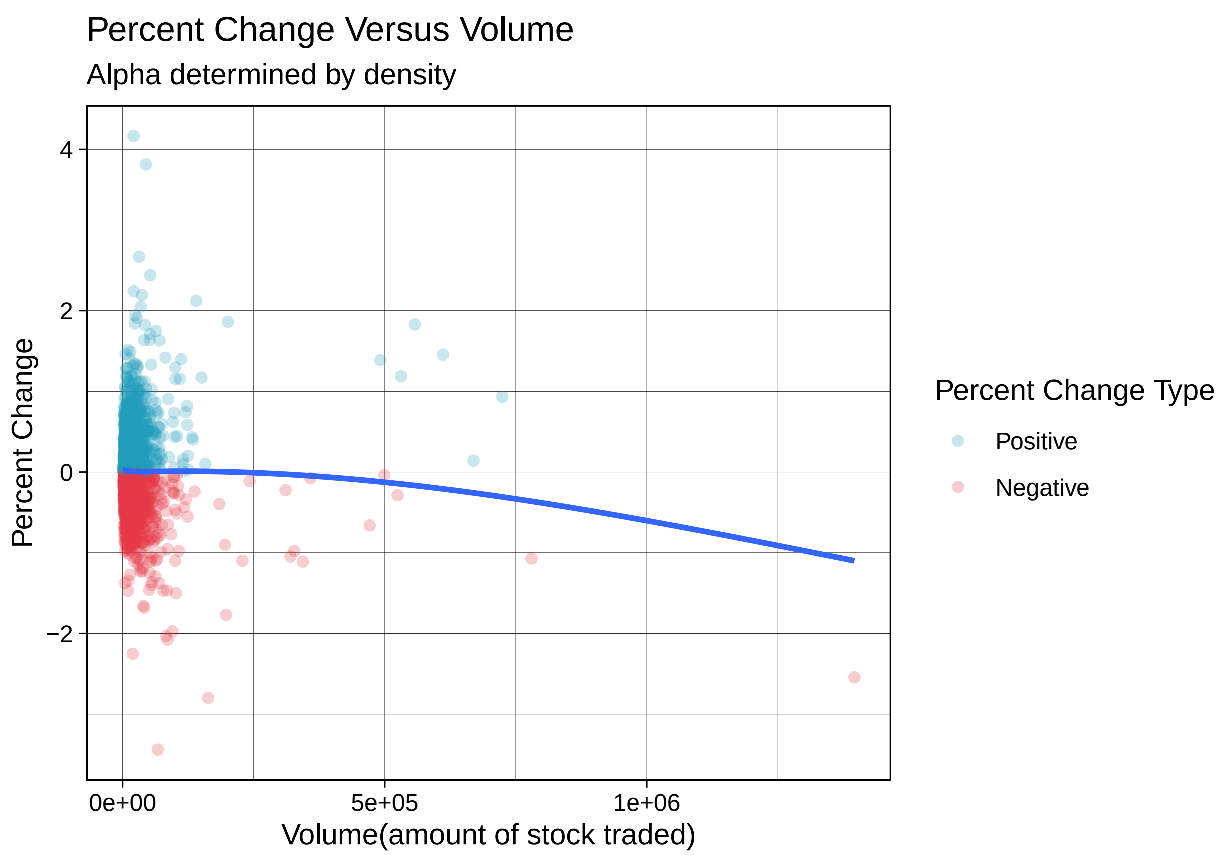
```
## `geom_smooth()` using formula = 'y ~ x'
```



Next we split apart the faceted graph from above to make it larger scale_fill_manual(values = c("#1d3557","#e63946"))

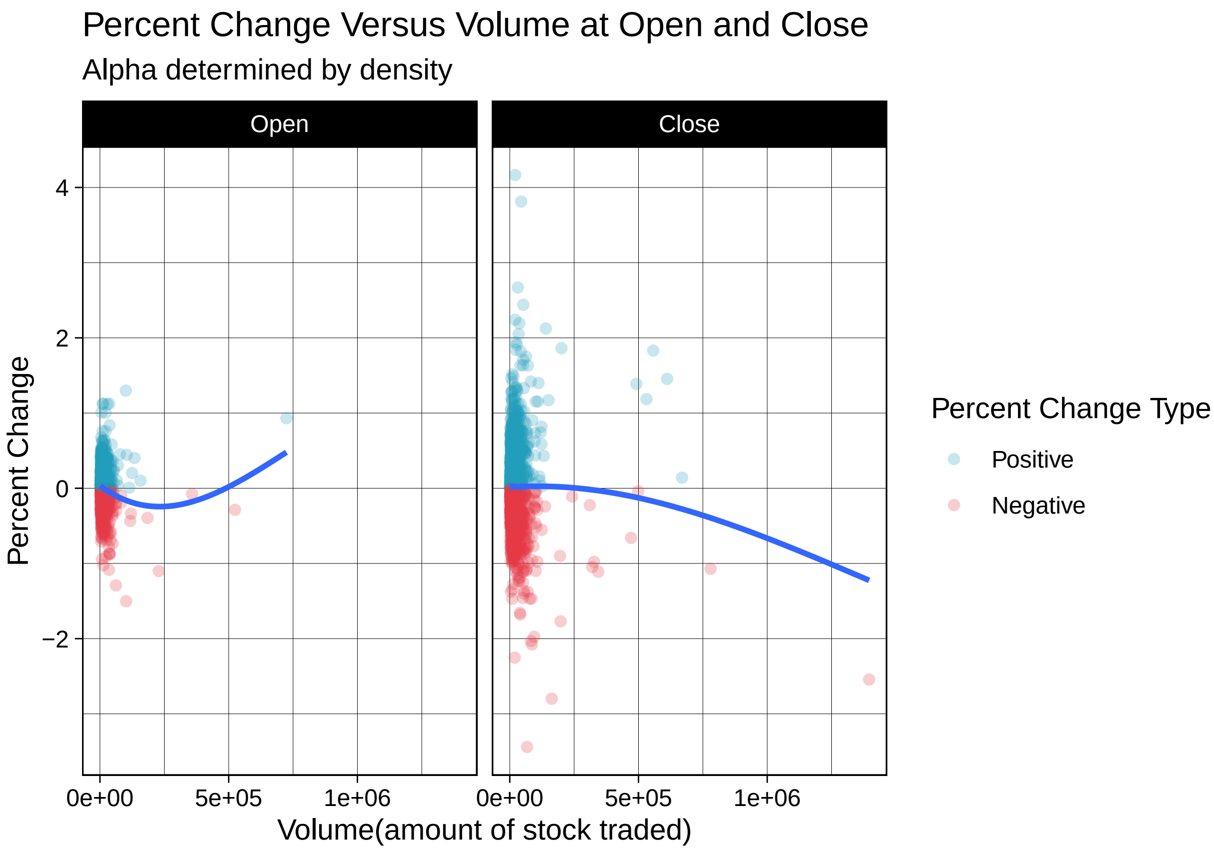
```
question1.1 <- ggplot(data = openClose, aes(x = volume, y = percent_change)) + geom_point(alpha = 0.25,a

question1.2 <- ggplot(data = openClose, aes(x = volume, y = percent_change)) + geom_point(alpha = 0.25,a


openClosePositive <- openClose %>%
  filter(percent_change_type == "Positive") #takes only the positive percent change types

question1.2Positive <- ggplot(data = openClosePositive, aes(x = volume, y = percent_change)) +geom_poin

#plots the log of volume on the x axis and the percent change on the y axis, creates a scatter plot of



openCloseNegative <- openClose %>%
  filter(percent_change_type == "Negative") #takes only the negative percent change values

question1.2Negative <- ggplot(data = openCloseNegative, aes(x = volume, y = percent_change)) +geom_poin
#plots the log of volume on the x axis and the percent change on the y axis, creates a scatter plot of

#displays the graphs created above
question1.1
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



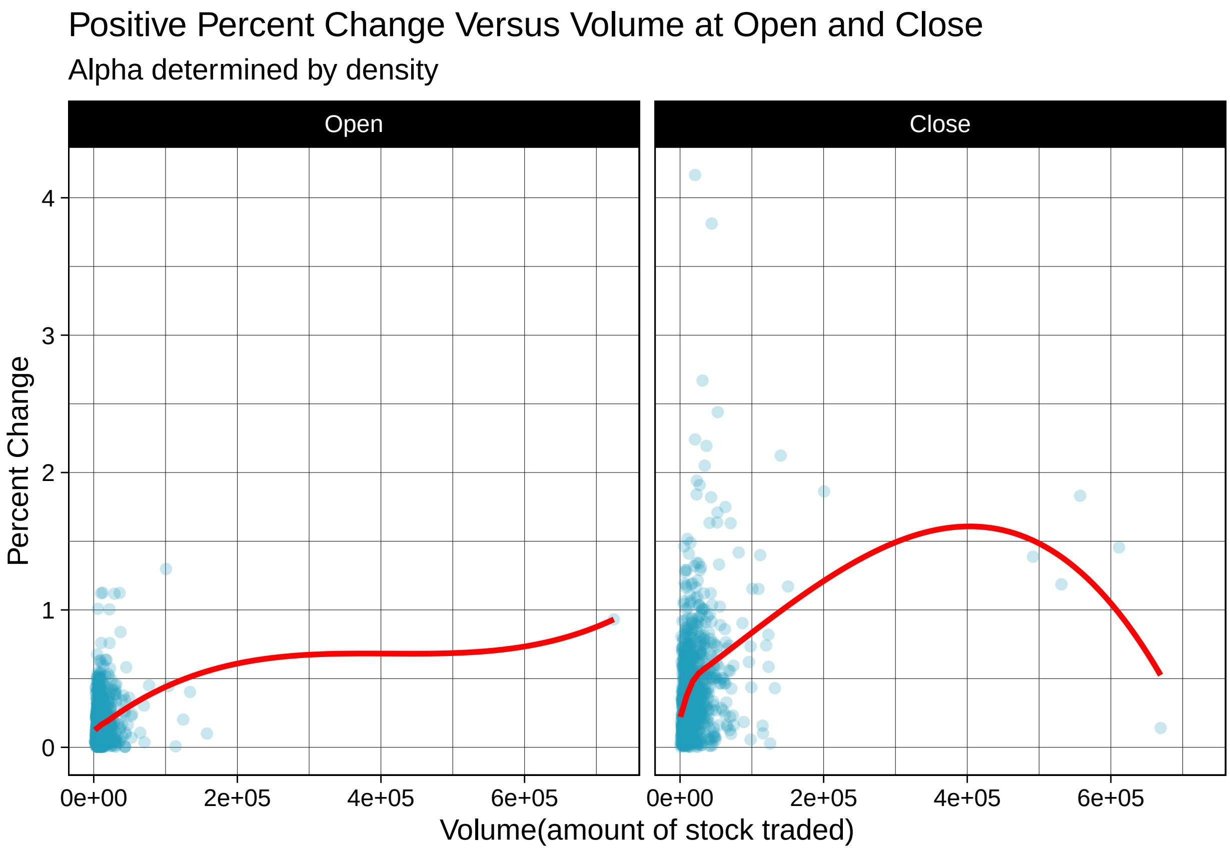

```
question1.2
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



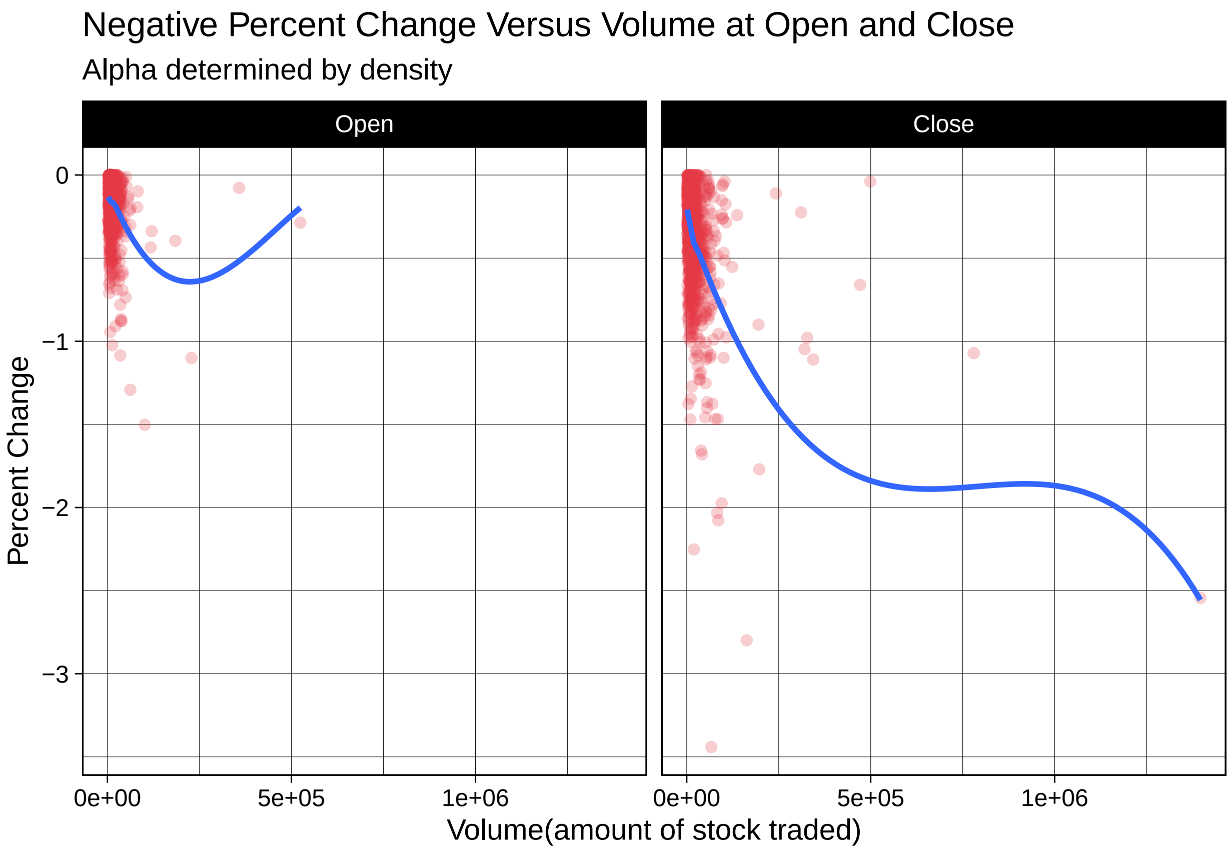

```
question1.2Positive
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```




```
question1.2Negative
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



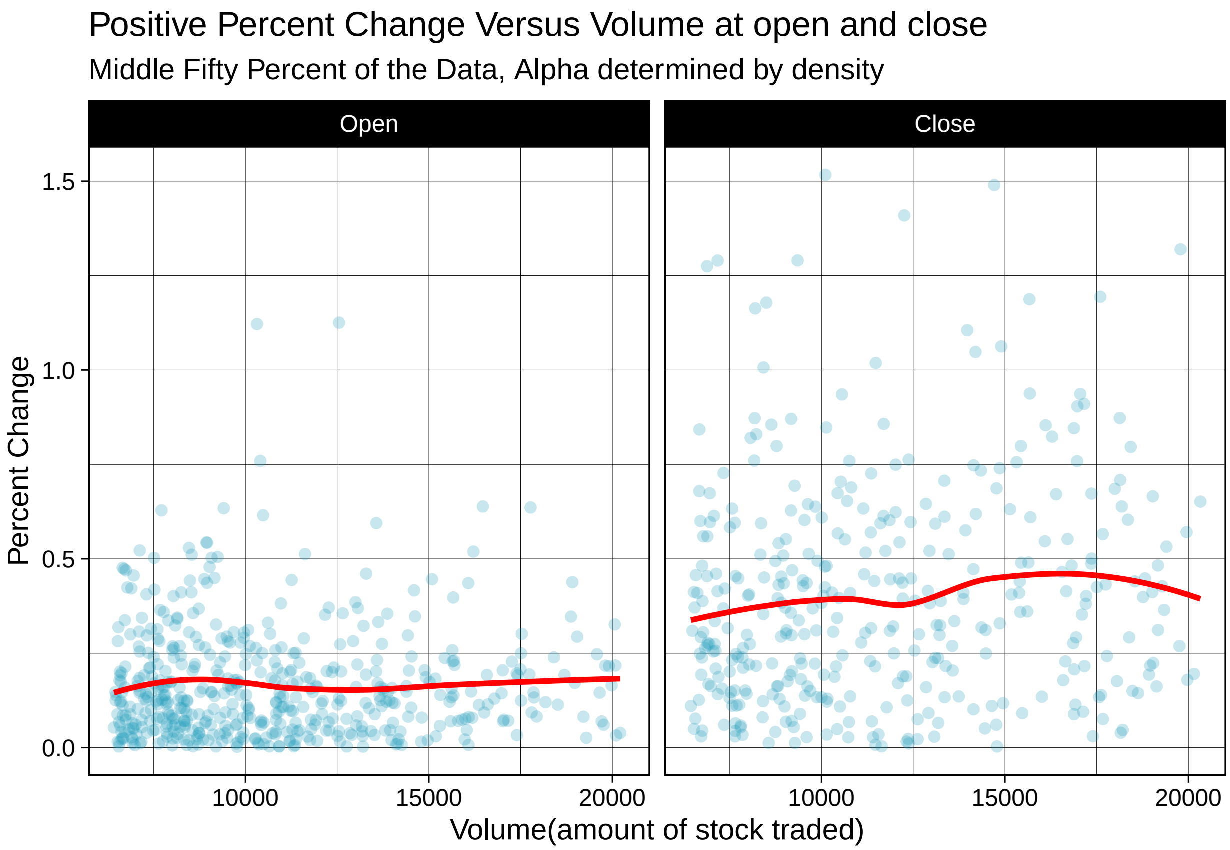
While these graphs show some promise they are a little bit zoomed out still so to improve them I will try removing some outliers. Another issue with the using a logarithmic scale and removing outliers is adding the proper labels to make it more clear what the graph actually represents. I also plan to clean up these graphs a bit and add some nice formatting. The outliers add a lot of noise but are also extremely important in the anylsis of stock price(after all they are the biggest movers) so finding a balance is key here.

```
quantile(openClose$volume) #gives the 1st, 2nd, 3rd and 4th quantile of the dataset
```

```
##        0%       25%       50%       75%      100%
##     525.0    6413.0   10584.5   20332.5 1396017.0
```

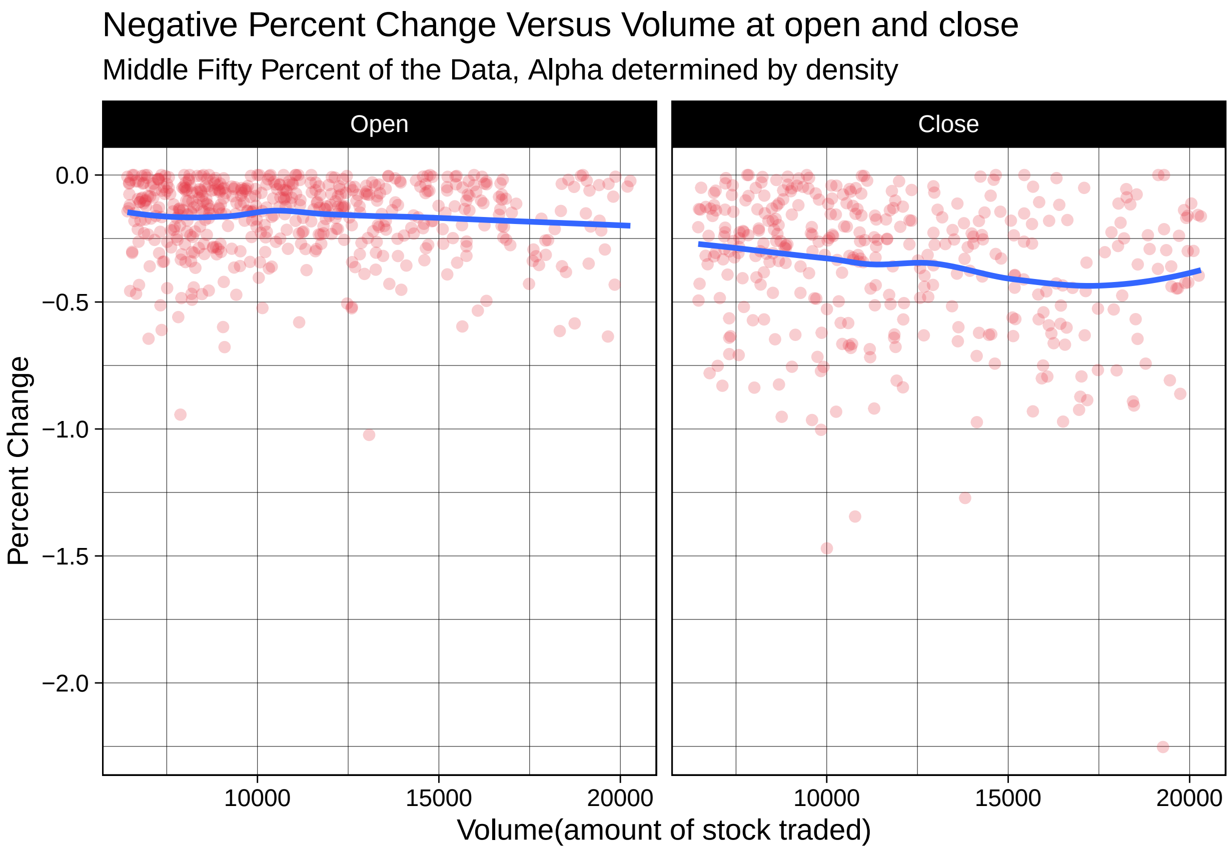
```
openClose2 <- openClose %>%
  filter(volume >= 6413 & volume <= 20332) #keeps only the middle fifty percent of the dataset

question2.1 <- ggplot(data = openClose2, aes(x = volume, y = percent_change)) +geom_point(alpha = 0.15,

question2.2 <- ggplot(data = openClose2, aes(x = volume, y = percent_change)) + geom_point(alpha = 0.25

openClosePositive2 <- openClose2 %>%
  filter(percent_change_type == "Positive") #takes only the positive percent change types

openClosePositiveScatterPlot2 <- ggplot(data = openClosePositive2, aes(x = volume, y = percent_change))

#plots the log of volume on the x axis and the percent change on the y axis, creates a scatter plot of

openClosePositiveScatterPlot2 #displays our graph
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



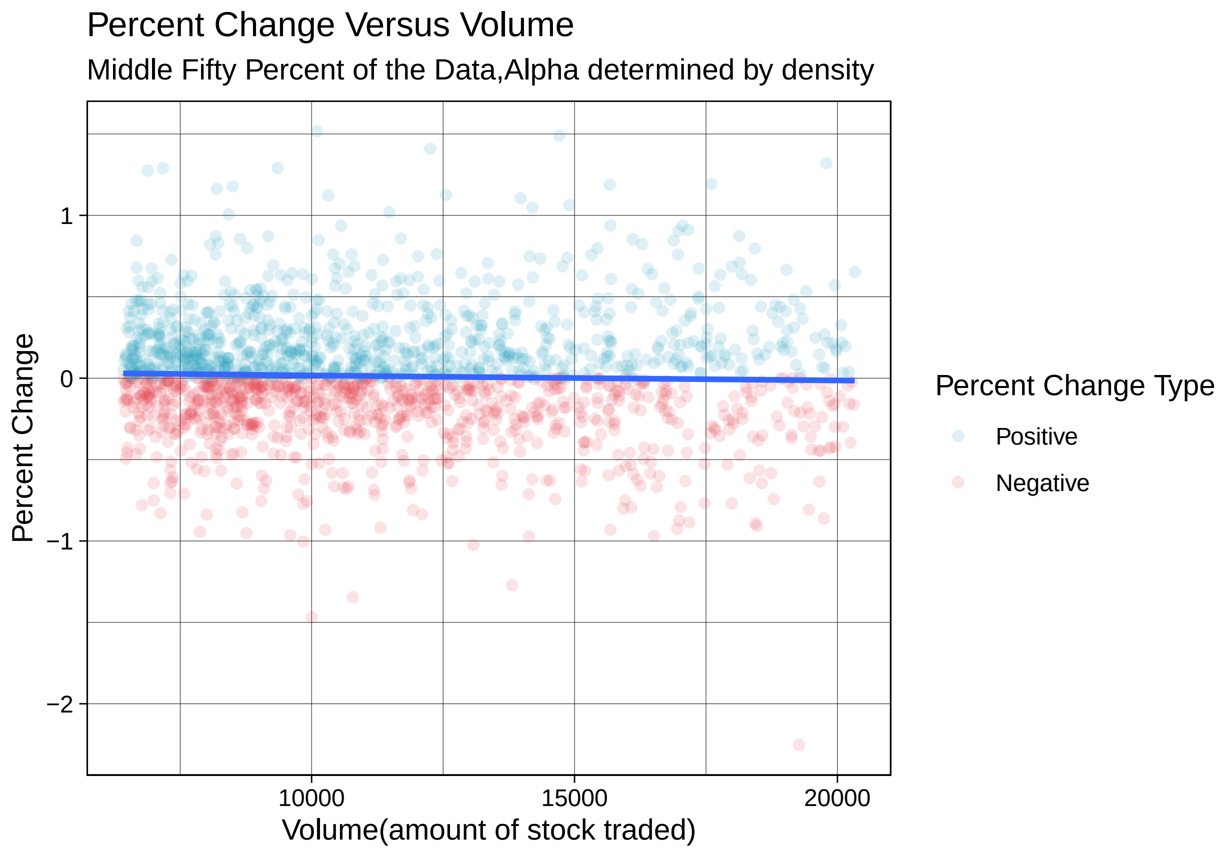

```
openCloseNegative2 <- openClose2 %>%
  filter(percent_change_type == "Negative") #takes only the negative percent change values

openCloseNegativeScatterPlot2 <- ggplot(data = openCloseNegative2, aes(x = volume, y = percent_change))
#plots the log of volume on the x axis and the percent change on the y axis, creates a scatter plot of

openCloseNegativeScatterPlot2 #displays the graph
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



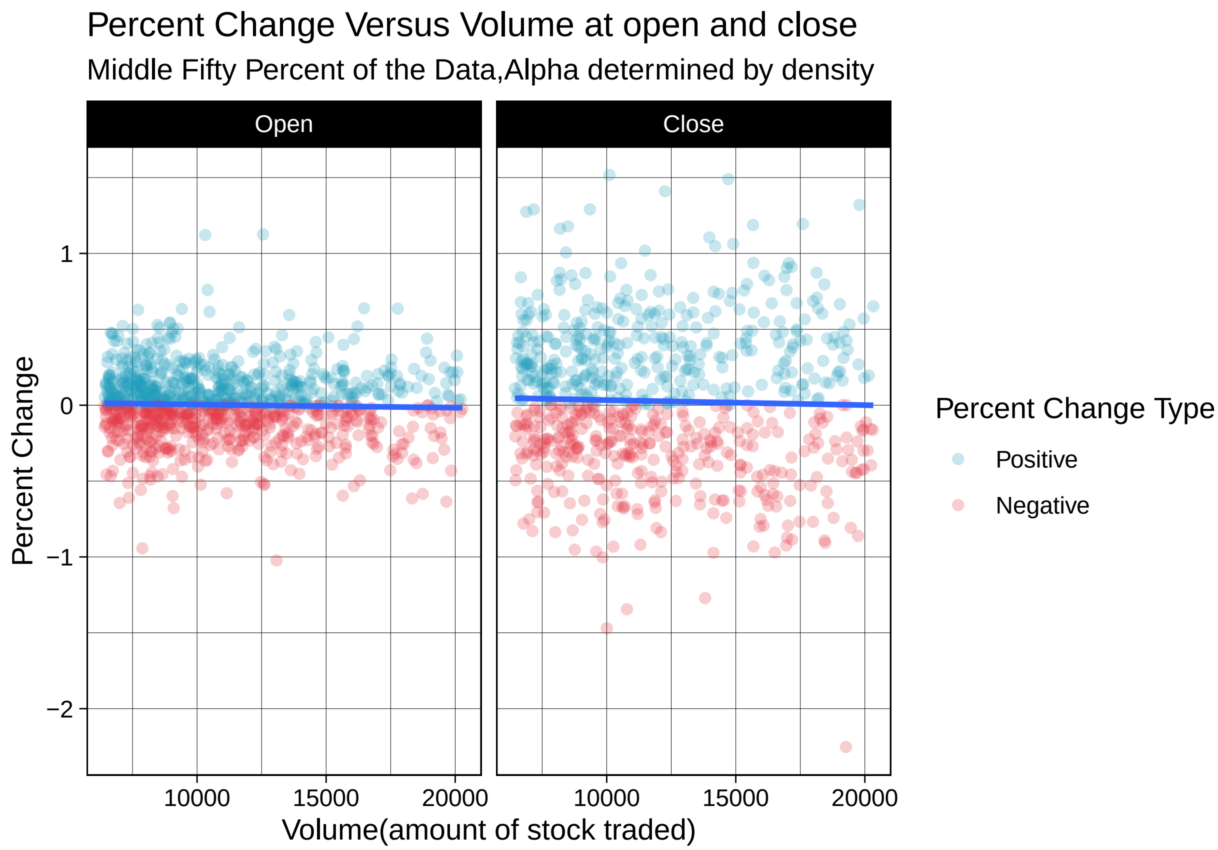

```
question2.1
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```




```
question2.2
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



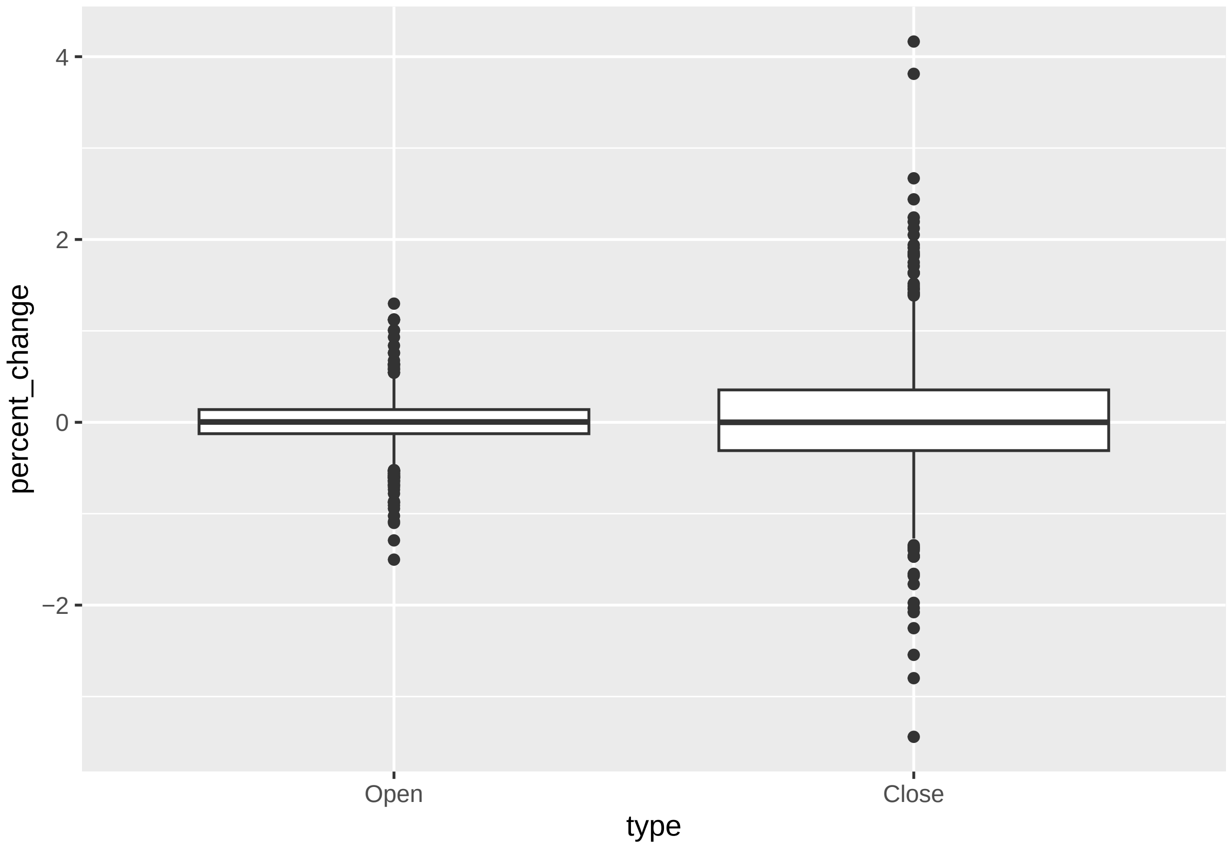

```
openCloseOpen <- openClose %>%
  filter(type == "open")
quantile(openCloseOpen$percent_change, probs = seq(0,1,0.10))
```

```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
##   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
```

```
openCloseClose <- openClose %>%
  filter(type == "close")
quantile(openCloseClose$percent_change, probs = seq(0,1,0.10))
```

```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
##   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
```

```
ggplot(data = openClose, aes(x = type, y = percent_change)) +geom_boxplot()
```



Question 2 Which seasons(Fall,winter,Spring, Summer) have the greatest change in stock price and what direction does it trend?

First we will start with some dataset manipulation in order to add the proper season to the dataset and remove outliers from percent change to make the graph more "zoomed in"

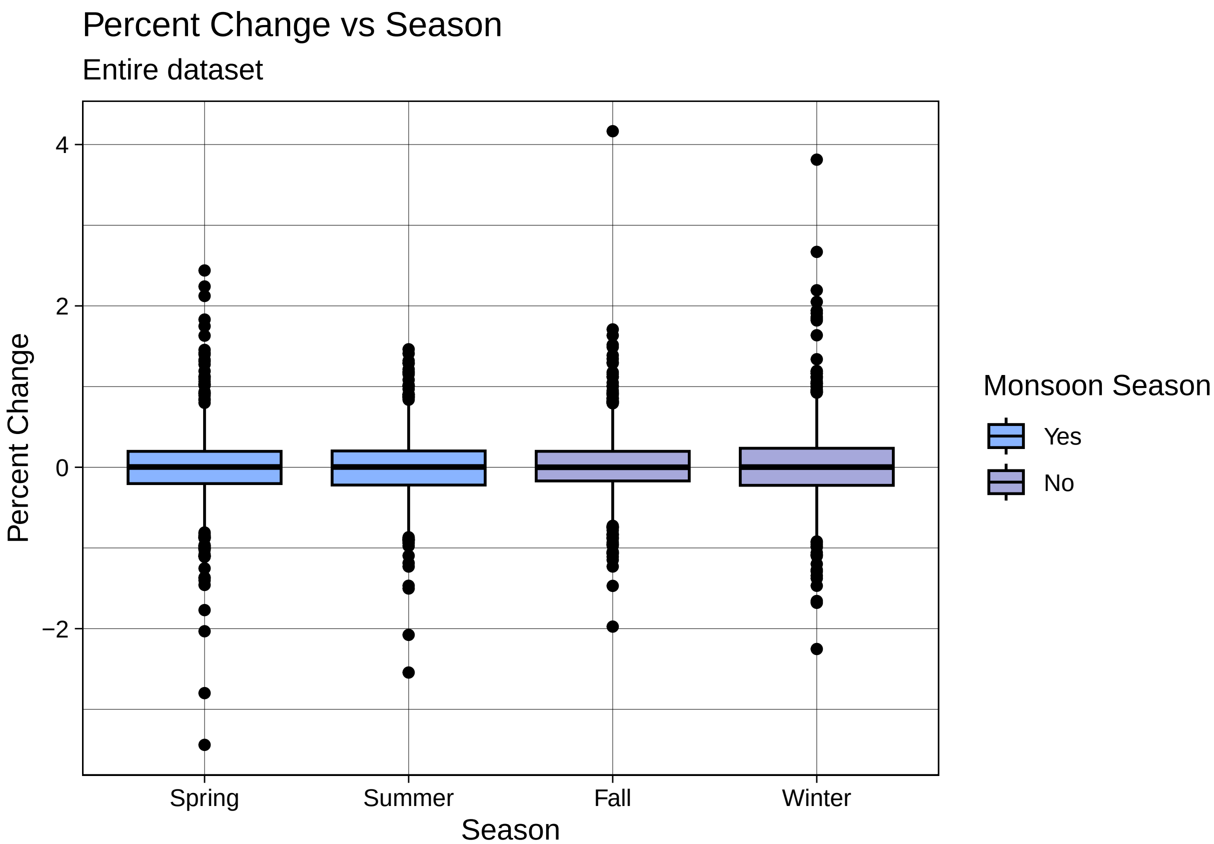
```
dataset4 <- openClose %>%
  mutate(day = wday(date, label = TRUE, abbr = FALSE), #creates new column based on day of week
         season = #creates new column based on season
if_else(month(date)>=4 & month(date)<=6,"Spring", # if month is 3-5 labels it spring
  if_else(month(date)>=7 & month(date)<=9,"Summer", # if month is 6-8 labels it summer
    if_else(month(date)>=10 & month(date)<=12,"Fall","Winter")))) %>%   # if month is between 9-11 labe
  mutate(Moonsoon_Season = if_else(season == "Spring" | season == "Summer","Yes","No")) #adds column fo
dataset4$Moonsoon_Season <- factor(dataset4$Moonsoon_Season, levels = c("Yes","No")) #changes order of 
dataset4$season <- factor(dataset4$season, levels = c("Spring","Summer","Fall","Winter")) #changes orde


quantile(dataset4$percent_change) #shows the quartiles for dataset4
```

```
##          0%         25%         50%         75%        100%
## -3.44067797 -0.20066780  0.00289984  0.21020593  4.16551724
```

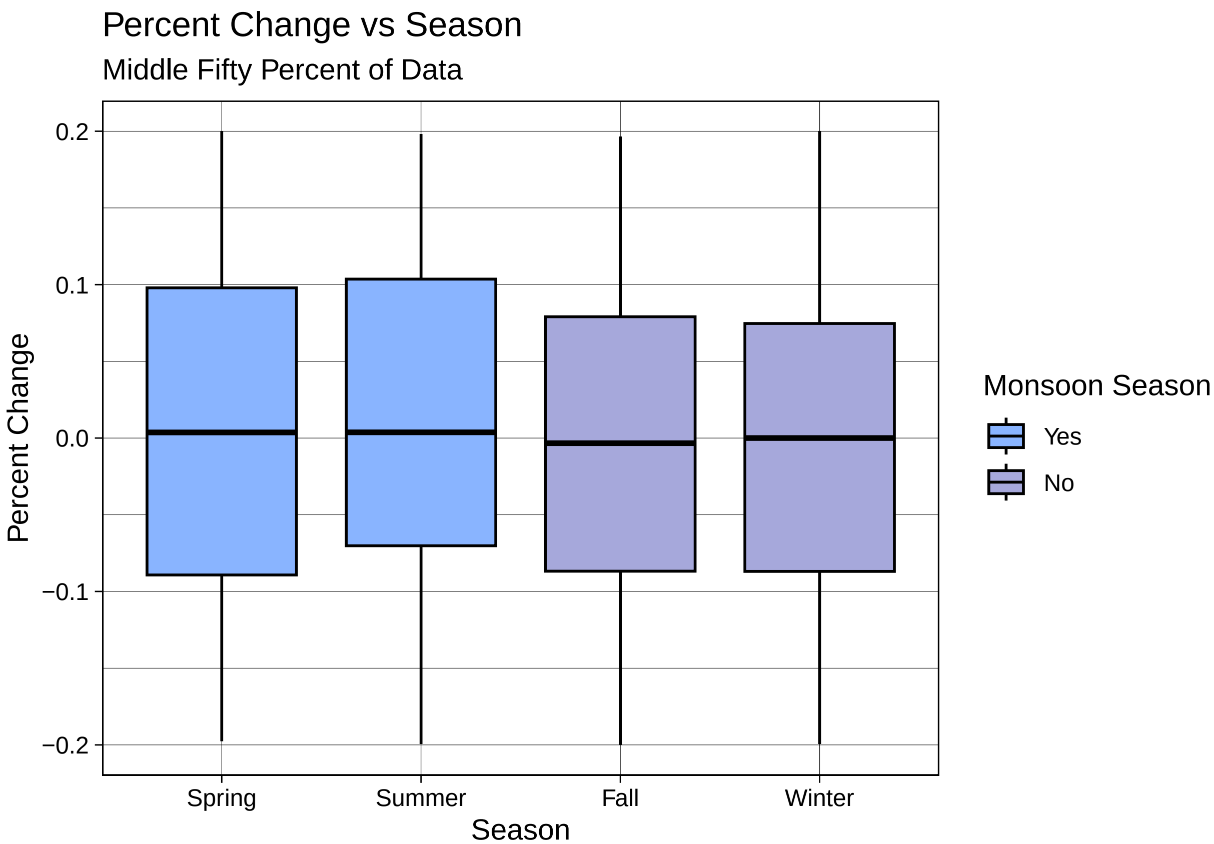
```
dataset3 <- dataset4 %>% #creates new dataset from dataset4
  filter(percent_change>-0.2 & percent_change<0.2) #keeps only the middle fifty percent of the data
```

```
#c("darkblue","darkblue","lightblue","lightblue")
SeasonPercentChangeBoxPlot2 <- ggplot(data = dataset4, aes(x = season, y = percent_change, fill = Moons
```

```r
SeasonPercentChangeBoxPlot2 #displays graph
```



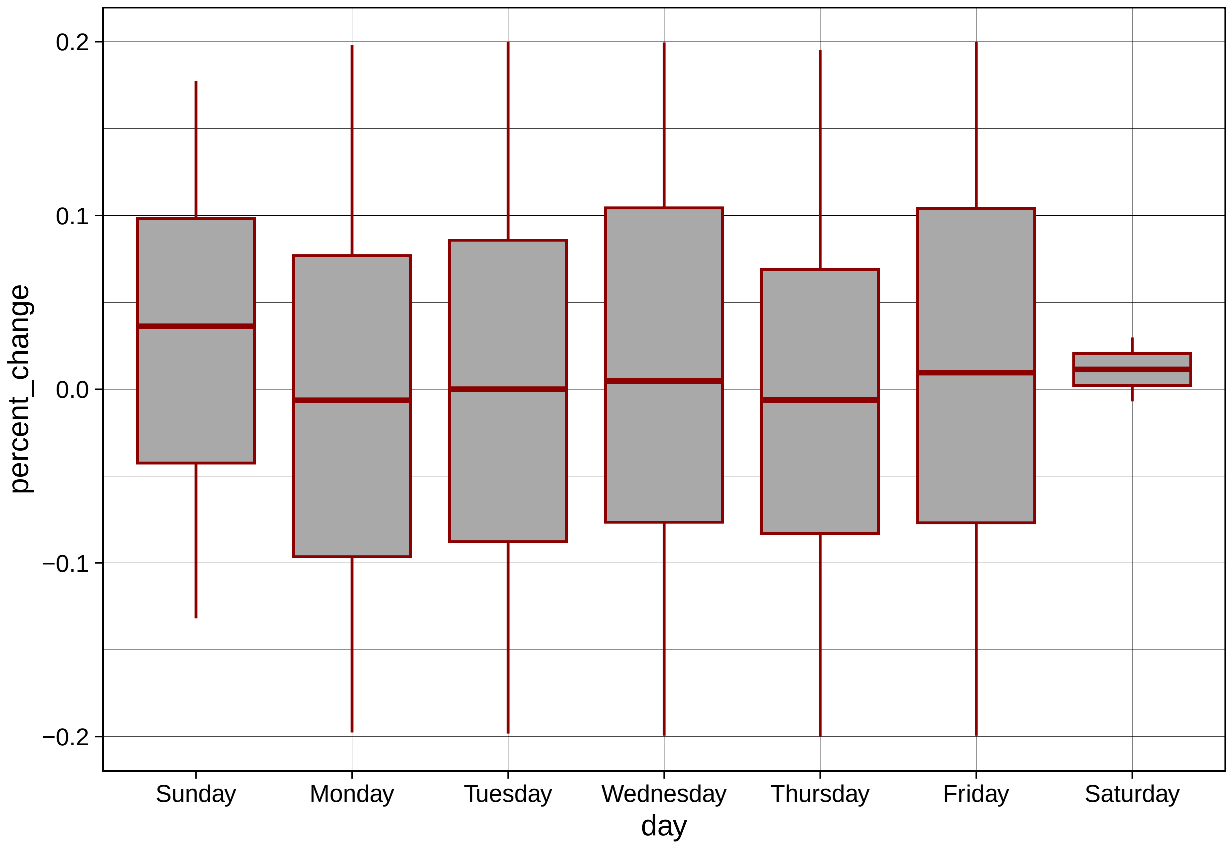

```r
SeasonPercentChangeBoxPlot <- ggplot(data = dataset3, aes(x = season, y = percent_change, fill = Moonso
SeasonPercentChangeBoxPlot #displays graph
```



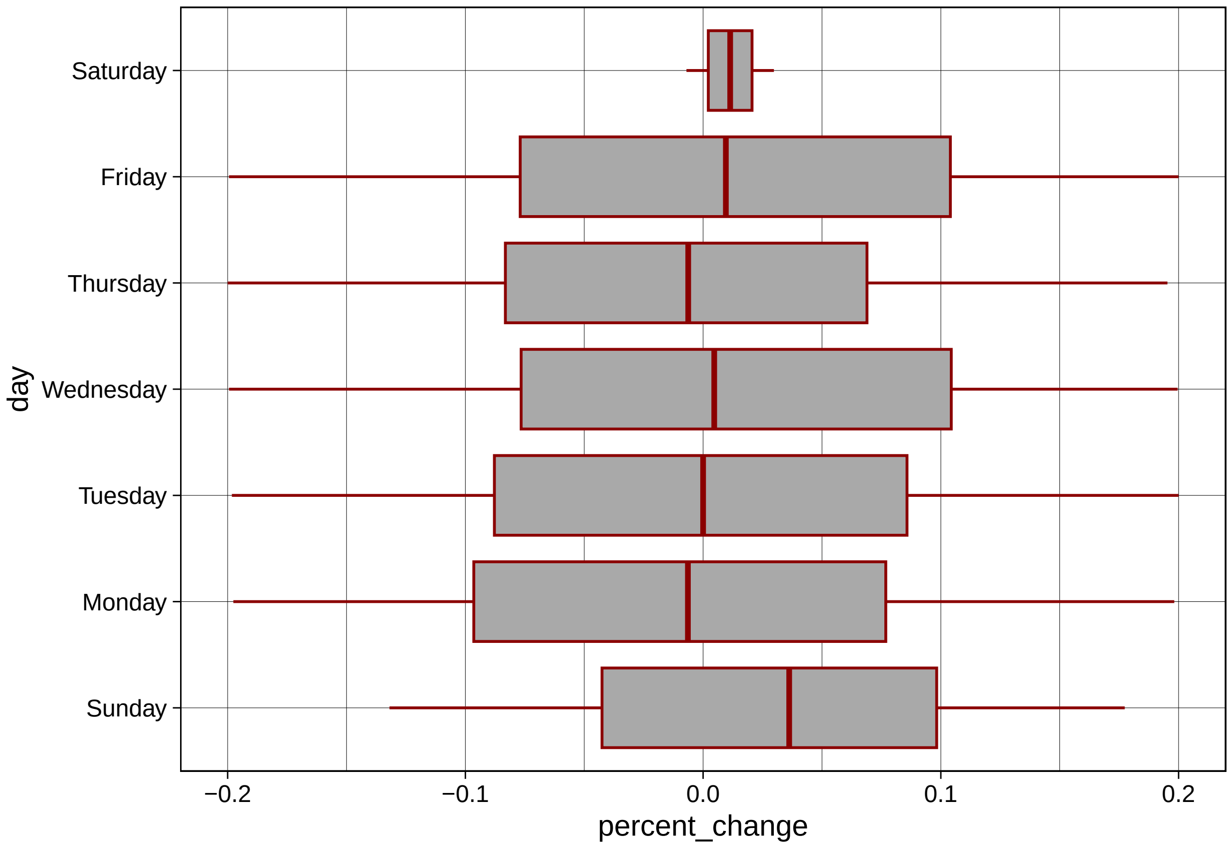
This graph can be improved visually to be more appealing. We also removed percent change less than -0.2 and greater than 0.2 While this does create a nicer looking graph these outliers are important to represent. The higher/lower the percent change the more the stock moves so these outliers are very important and cannot be ignored. I think dealing with the outliers in a more elegant way will be the best way to improve these results.

#IGNORE BELOW————> #from another one of my questions I do no think I will persue but I decided to keep the code because I already wrote it and maybe I will need it in the future

```
graph3 <- ggplot(data = dataset3, aes(x = day,y = percent_change)) + geom_boxplot(color = "darkred", fil
graph3
```




```
graph3.2 <- graph3 +coord_flip()
graph3.2
```




```
graph3testData <- dataset2 %>%
  mutate(percent_change = (((open-close)/open))*100,percent_change_type = if_else(percent_change>0,"Posi
```