DSC Group 4
Milestone 2
2/11/22

Dataset link (since we changed after Milestone 1):
https://www.kaggle.com/andrewmvd/sp-500-stocks

Delete unnecessary columns

We deleted the multiple name listings (Shortname & Longname in sp500_companies) and kept only the ticker abbreviation (Symbol in sp500_companies) column. We chose to do that because the stock abbreviation was listed in the stocks and companies data set and could be considered a key. Additionally we removed the long business summary because it was a long text description.

Clean up column values

We converted the market cap and EBITDA column to be a numerical value. Additionally we made two sets for analysis. We scaled down the EBITDA and Marketcap by 100,000,000 so that the units wouldn't be substantially higher than the other numerical values for any PCA analysis. In the other set for analysis we did not scale it down and keyed those fields against each other. Additionally we reformatted the date field in the stocks data set to match the formatting of the date column in the index dataset.

Ensure correct data types

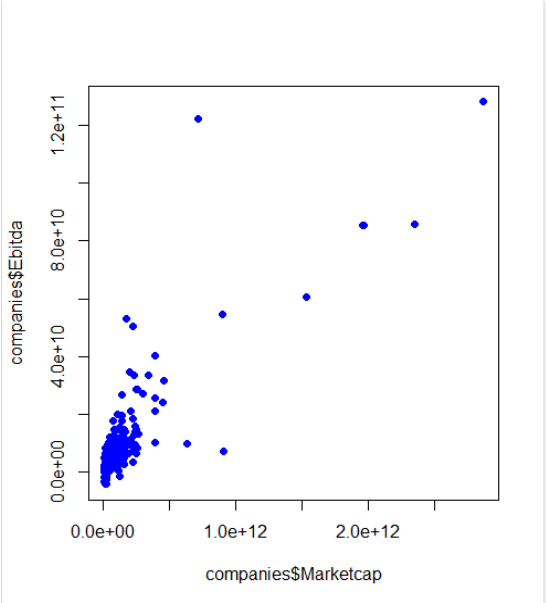
The data types were correct when brought into R. As noted above, the dates in sp_500index were converted to dates and converted to share the same formatting as sp_500stock. Additionally, the numerical fields were converted to numerical fields. The character fields came over as character data types.

Handle missing data

During our cleanup we created columns that analyzed the mean trading price over the duration of the data, as well as the median trading price. The point of this was to identify and analyze whether these points took place 50% through the 10 year sample, or if they occurred sooner or later. This would help us to identify whether a stock grew steadily throughout the 10 years or saw a balloon hike in price; meaning the price remained lower for much longer and then saw a stark increase recently. It would also help us to see whether the stock tracked below the mean price longer than it did above it. We also did this same analysis on the median and mean of the overall S&P.

Identify outliers

While none of the data is an outlier due to statistical errors or issues, when you begin to analyze and plot the data you do see some outliers within the correlation and regression. One example is when you plot EBITDA against the market cap, we see that Tesla has a much higher marketcap than expected when comparing to the others.



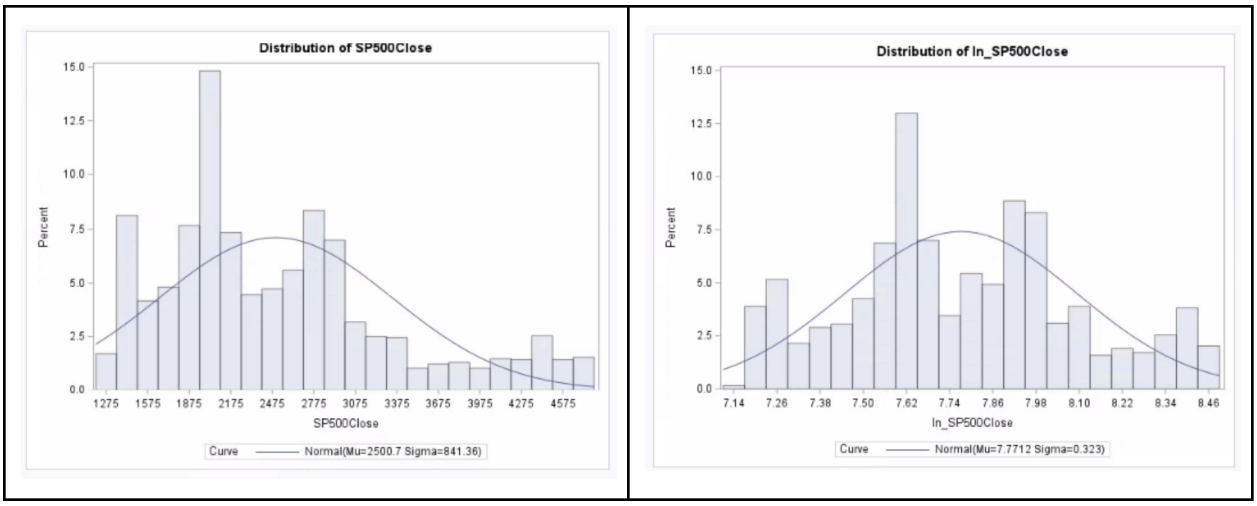
Additionally there were some issues with Googl and Goog within our data sets. While there is a technical difference on the stock exchange between these two tickers, as one is google stock bought that grants voting rights and the other is not, the actual data points within our system were identical. While they were separate in all analysis due to us using the symbol/ticker, they are still duplicated records that when comparing overall stock performance in relation to things such as market cap and S&P growth, followed the same line. Because of this, we have decided to utilize only the class a shares, which provide voting rights which fall under the GOOGL symbol. Meanwhile, we thought one interesting analysis would be to compare the discount that does exists on the Class C shares (Goog) vs the Class A voting right shares. For that sake, we have decided we are going to analyze those two symbols against one another over time to see if at any point the discount was heightened or lowered and see if there is any correlation to the S&P or the EBITDA/Marketcap within those variances.

Scaling issues

As we noted above in our data values breakdown, scaling was an issue with our data and for PCA purposes, it was important to scale down the EBITDA and the Marketcap to get it more in line with our other variables.

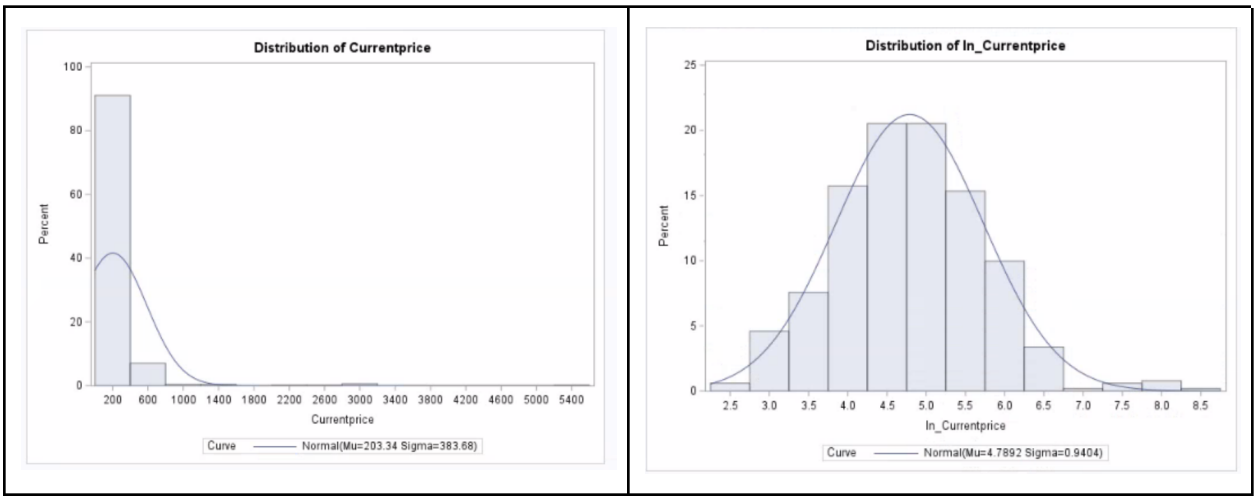
Distributions

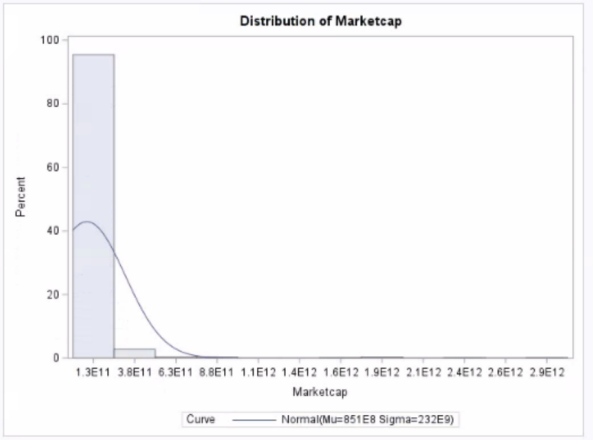
sp500_index

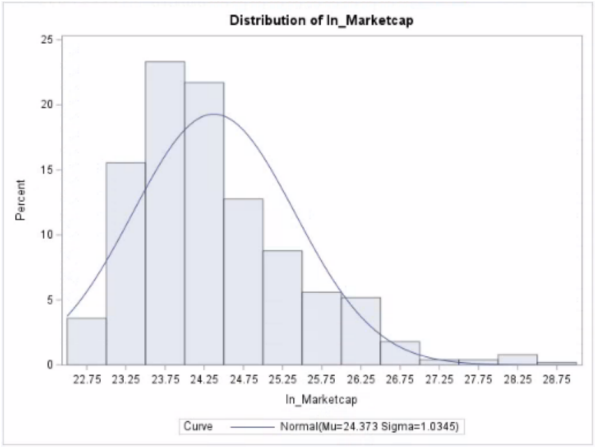


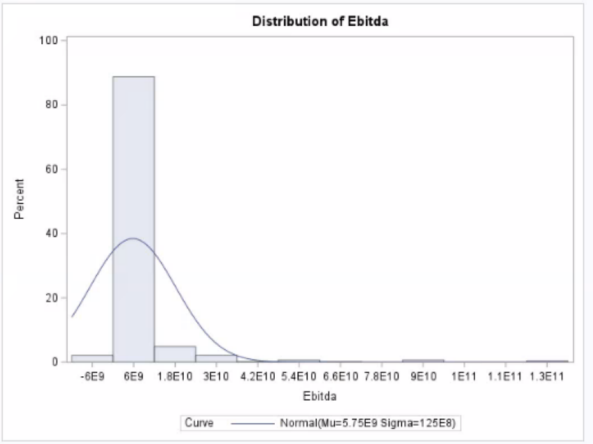
A transformation was done because the data was leaning too much right skewed so a log transformation was done. After the trasnformaiton the histogram proved to be more bimodal skewed but has normal distribution now.

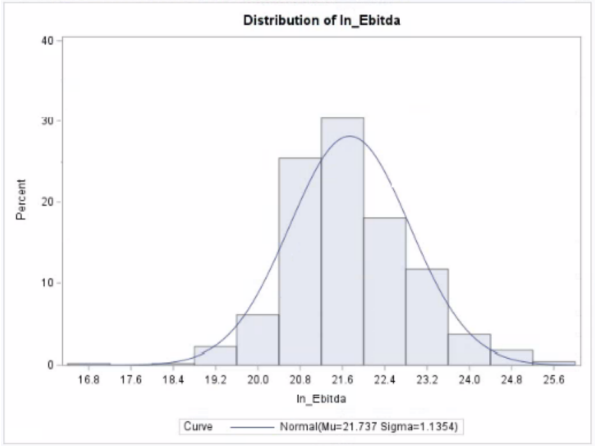
sp500_companies

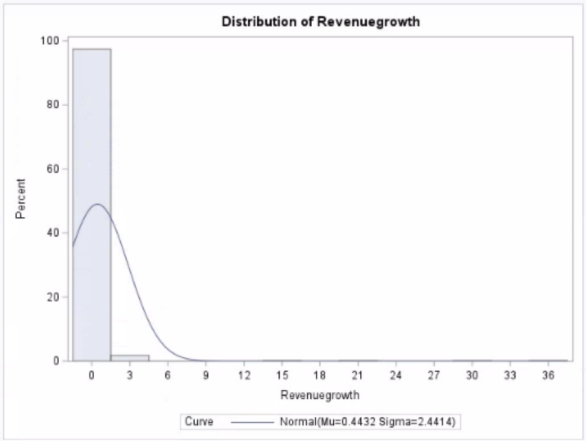


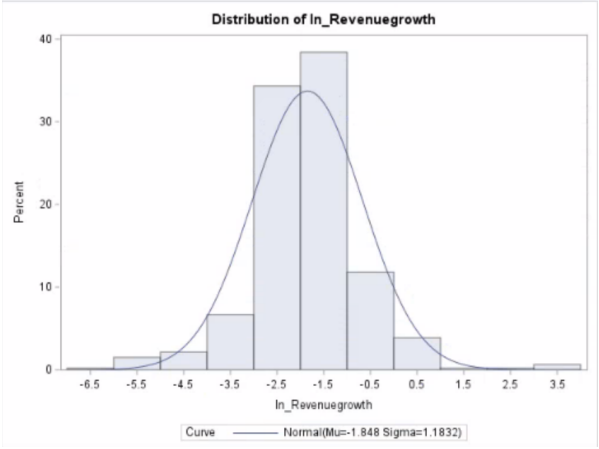

Distribution of Marketcap
Percent
100
80
60
40
20
0
1.3E11 3.8E11 6.3E11 8.8E11 1.1E12 1.4E12 1.6E12 1.9E12 2.1E12 2.4E12 2.6E12 2.9E12
Marketcap
Curve
Normal(Mu=851E8 Sigma=232E9)

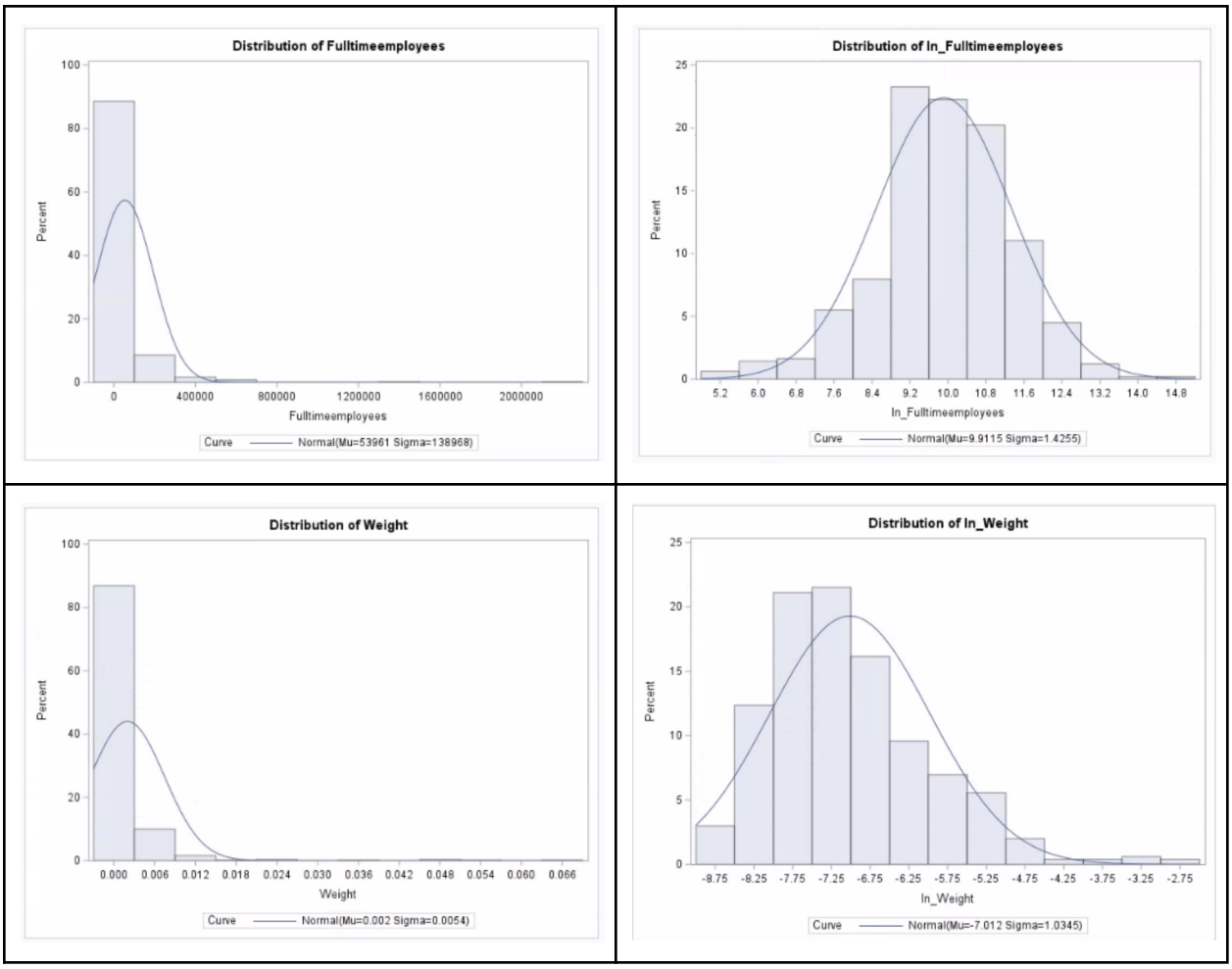

Distribution of ln_Marketcap
Percent
25
20
15
10
5
0
22.75 23.25 23.75 24.25 24.75 25.25 25.75 26.25 26.75 27.25 27.75 28.25 28.75
ln_Marketcap
Curve
Normal(Mu=24.373 Sigma=1.0345)


Distribution of Ebitda
Percent
100
80
60
40
20
0
-6E9 6E9 1.8E10 3E10 4.2E10 5.4E10 6.6E10 7.8E10 9E10 1E11 1.1E11 1.3E11
Ebitda
Curve
Normal(Mu=5.75E9 Sigma=125E8)


Distribution of ln_Ebitda
Percent
40
30
20
10
0
16.8 17.6 18.4 19.2 20.0 20.8 21.6 22.4 23.2 24.0 24.8 25.6
ln_Ebitda
Curve
Normal(Mu=21.737 Sigma=1.1354)

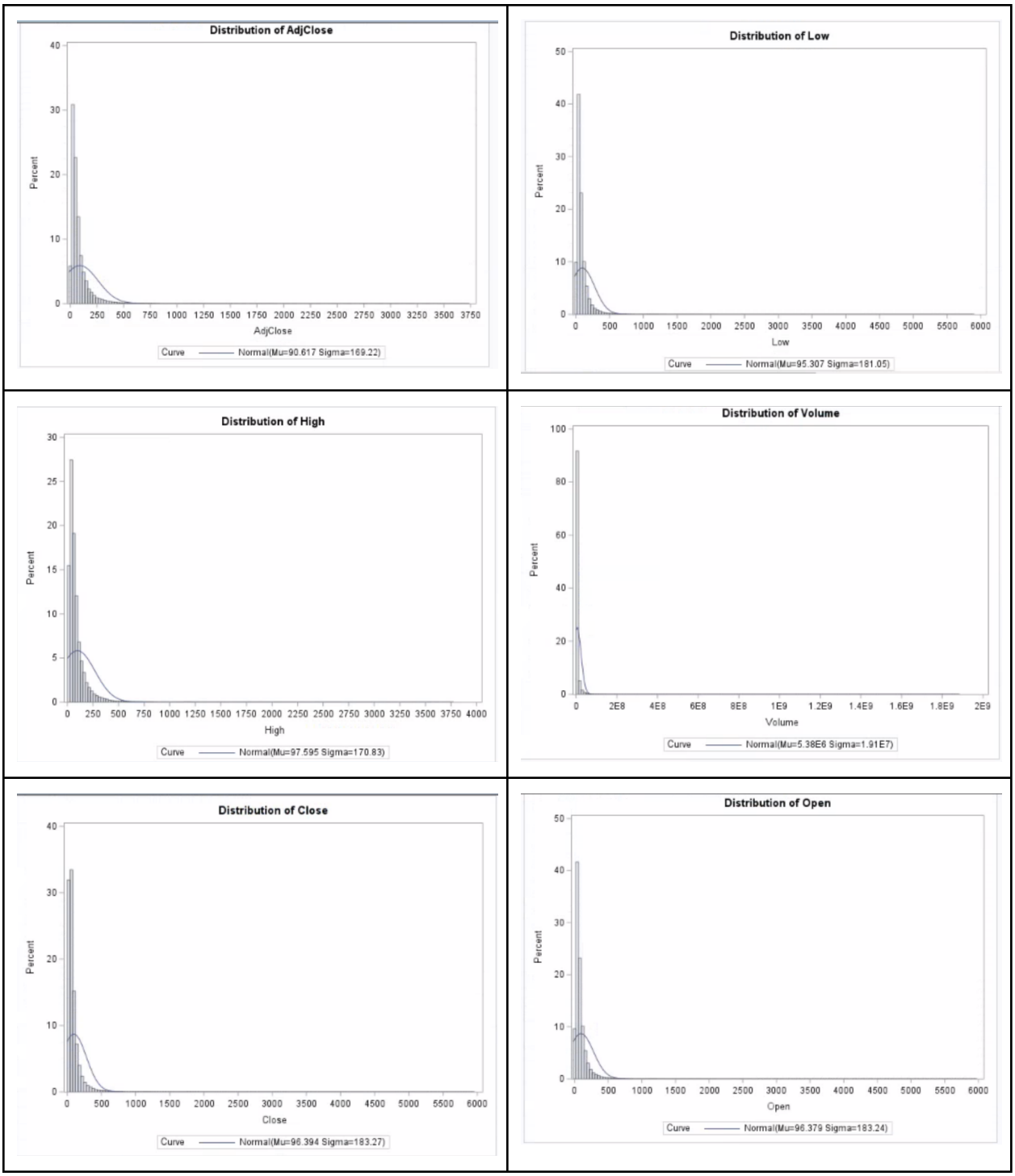

Distribution of Revenuegrowth
Percent
100
80
60
40
20
0
0 3 6 9 12 15 18 21 24 27 30 33 36
Revenuegrowth
Curve
Normal(Mu=0.4432 Sigma=2.4414)


Distribution of ln_Revenuegrowth
Percent
40
30
20
10
0
-6.5 -5.5 -4.5 -3.5 -2.5 -1.5 -0.5 0.5 1.5 2.5 3.5
ln_Revenuegrowth
Curve
Normal(Mu=-1.848 Sigma=1.1832)



Initially running the histograms on all the numerical vairbaels in the companies dataset all of the variables needed to be transformed (left images). A log transformation was done and all histograms did shift after and became all less right-skewed and more normal. The only ones that still have a little skewness but would be fine for further use are ln_Weight, and ln_Marketcap the rest of the variables have normal distrbutitions.

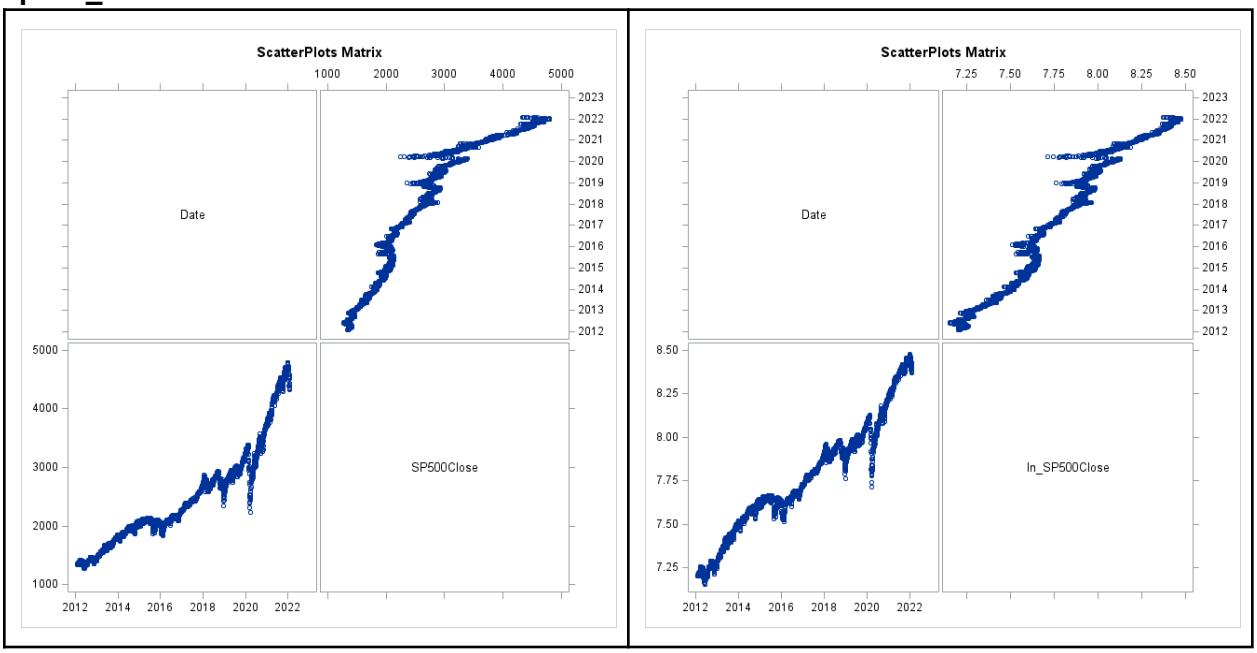
sp500_stocks



All the histograms are right skewed and they would be transformed to hopefully become normal distribution SAS was having problems transforming the data because of the number of observations of over 1.5 million being entered. If we need to use these transformed histograms for later analysis we would cut down on the amount of observations and put them in separate files and run it from there.

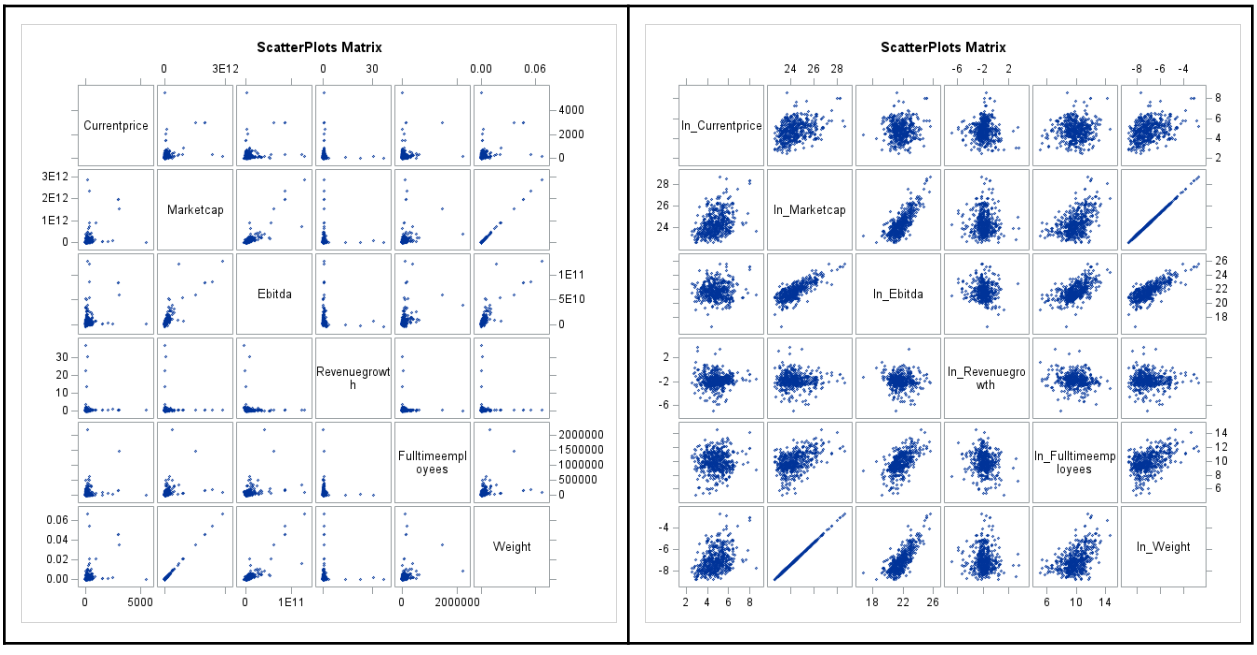
Scatter plots

sp500_index



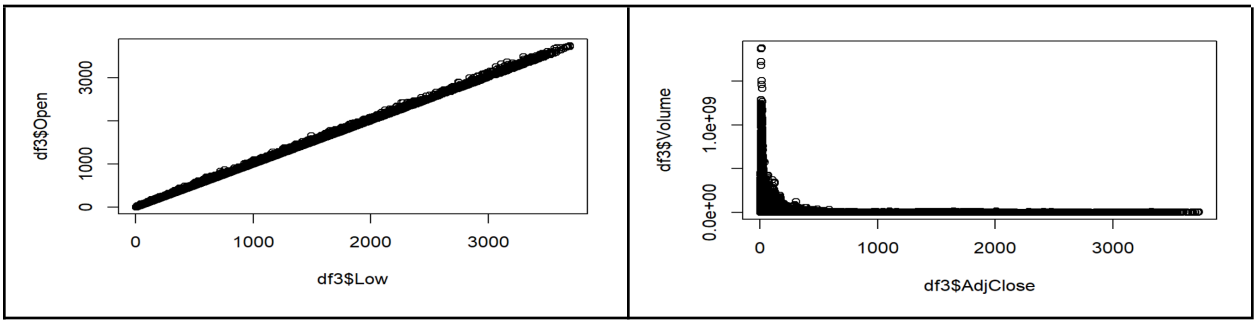
After seeing the transformation it is felt that both plots are increasing and positively increasingly linear scatter plots meaning that with the further date the higher the S&P500 was at closing time. Seeing trends in 2020 when COVID hit, both scatter plots dipped down drastically. After the world has started to live with COIVD and vaccinations came out the S&P500 closings have skyrocketed linear positively almost double from COVID's 2020 beginning. It can be seen also where the 2008 recession towards the rebound of the country in 2012 the S&P500 closing was increasing again meaning the country was rebuilding meaning a linear relationship between date and S&P500 closings.

sp500_companies



After transformation, all scatterplots become way more spaced out in data points. The data is not (left image) is not bunched but in the bottom left corner in many of the plots likely close to 0. In the tranformmed most plots show a positive trend of data points increasing the further along the log transformed graphs go and proves these variables are positvely correlated to each other. Only two plots after log transformation have straight linear shaped plots and those are ln_Marketcap and ln_Weight those are 1 correlated later in the corrleaiton matrix too.

sp500_stocks



To save all the repeat images of all the variables this is what the majority of the scatter plots (left photo) looked like before transformed. They were linear lines and showed correlation. There may be heavy linear relation from variable to variable because if there are 1.5 million observations there has to be variety in the results besides a strong positive lienar plot. The other result that was found only once was Adjclose and Volume where it made an “L” shape vertically and horizontally of 0. That does not seem linear at all because all the data is going across both axises.

Corrleation Matrices
sp500_index dataset correlation matrix

```
                 df..S.P500.Close.
df..S.P500.Close.                 1
```

The only variable that could fit into a correlation matrix for the index dataset was S&P500 Close and it is correlated to itself. There is no need for a transformation correlation matrix since it will not change the correlation value since it can run against is itself S&P500 Close. While 1 correlation value this is not multicolineairty since there is only one variable in the correlation matrix.

sp500_companies correlation matrix

```
                    df2.Currentprice df2.Marketcap
df2.Currentprice          1.0000000     0.3680546
df2.Marketcap             0.3680546     1.0000000
df2.Ebitda                       NA            NA
df2.Revenuegrowth                NA            NA
df2.Fulltimeemployees            NA            NA
df2.Weight                0.3680546     1.0000000
                    df2.Ebitda df2.Revenuegrowth
df2.Currentprice            NA                NA
df2.Marketcap               NA                NA
df2.Ebitda                   1                NA
df2.Revenuegrowth           NA                 1
df2.Fulltimeemployees       NA                NA
df2.Weight                  NA                NA
                    df2.Fulltimeemployees df2.Weight
df2.Currentprice                       NA  0.3680546
df2.Marketcap                          NA  1.0000000
df2.Ebitda                             NA         NA
df2.Revenuegrowth                      NA         NA
df2.Fulltimeemployees                   1         NA
df2.Weight                             NA  1.0000000
```

Pearson Correlation Coefficients
Prob > |r| under H0: Rho=0
Number of Observations

	ln_Currentprice	ln_Marketcap	ln_Ebitda	ln_Revenuegrowth	ln_Fulltimeemployees	ln_Weight
ln_Currentprice	1.00000 502	0.38979 <.0001 502	0.06090 0.1918 461	-0.02977 0.5215 466	0.07610 0.0924 490	0.38979 <.0001 502
ln_Marketcap	0.38979 <.0001 502	1.00000 502	0.77823 <.0001 461	-0.01852 0.6901 466	0.49614 <.0001 490	1.00000 <.0001 502
ln_Ebitda	0.06090 0.1918 461	0.77823 <.0001 461	1.00000 461	-0.09110 0.0594 429	0.58505 <.0001 451	0.77823 <.0001 461
ln_Revenuegrowth	-0.02977 0.5215 466	-0.01852 0.6901 466	-0.09110 0.0594 429	1.00000 466	-0.15463 0.0009 454	-0.01852 0.6901 466
ln_Fulltimeemployees	0.07610 0.0924 490	0.49614 <.0001 490	0.58505 <.0001 451	-0.15463 0.0009 454	1.00000 490	0.49614 <.0001 490
ln_Weight	0.38979 <.0001 502	1.00000 <.0001 502	0.77823 <.0001 461	-0.01852 0.6901 466	0.49614 <.0001 490	1.00000 502

In the companies dataset correlation matrix before transformation the only variables that were correlated were Weight and Marketcap. There feels to be no multicollinerarity in this dataset since the correlation is not too high where it could feel like multicolinearity. Since there was a transformation done in the histograms, there was no change in the correlation matrices where it can be multicollinearity.
The transformation made variables stop making each variable less correlated to itself and to other the rest of the variables in the dataset now. This means ln_Currentprice was correlated mostly with ln_Marketcap at 0.38979 but ln_Marketcap was at 0.38979 or ln_Weight 0.38979. All the values were lower in each of the variables that there was no multicollinearity. ln_Ebitda is equally correlated with ln_Marketcap and ln_Weight at 0.77823. ln_Revenuegrowth is only negatively correlated with all the other variables, the closest related variable is ln_Marketcap at -001852. ln_Fulltimeemployees the highest correlated value is ln_Marketcap at 0.49614. The only value that could have multicollineairty was ln_Marketcap with ln_Weight because the correlation was at 1.

sp_500_stocks correlation matrix

```
                df3..Adj.Close. df3.Close df3.High df3.Low
df3..Adj.Close.               1         NA        NA       NA
df3.Close                    NA          1        NA       NA
df3.High                     NA         NA         1       NA
df3.Low                      NA         NA        NA        1
df3.Open                     NA         NA        NA       NA
df3.Volume                   NA         NA        NA       NA
                df3.Open df3.Volume
df3..Adj.Close.       NA          NA
df3.Close             NA          NA
df3.High              NA          NA
df3.Low               NA          NA
df3.Open               1          NA
df3.Volume            NA           1
```

In the stocks dataset when running a correlation matrix it resulted in every variable being only correlated to itself. It does bring to notice that there could be multicollinearity since every variable in the matrix is only related to itself for 1.
There would be a transformation but again for the same reason SAS was having trouble with running all the data at once but also trying to do a correlation plot of two variables SAS could not do. The predicted result would be seeing variables show correlation to each other more than R showed of non-transformed values.

Initial OLS regression

Cannot do since we do not have a testing or training set. Also R did not want to accept a dataset sp500 stocks with over 1.5 million observations could not handle a frame over 25.2 GB. The others had errors when making due to testing/triaing set so was not used. If we want OLS for a later milestone or final we will cut down our timeline of data we are looking at.

Number of unique values per categorical variable
sp500_stocks dataset
Date: 3,050 unique Dates
Symbol: 505 unique Symbols

sp500_index dataset
Date: 2,517 unique Dates

sp500_companies dataset
Exchange: 4 unique Echanges
Symbol: 502 unique Symbols
Sector: 11 unique Sectors
Industry: 114 unique Industries
City: 235 unique Cities
State: 39 unique States
Country: 7 unique Countries

In all the datasets it will be impossible to make any dummy variables for any of the categorical variables because the number of unique values are more than two values. A dummy variable can only hold a value of either 1 or 0. Since each categorical variables have too many unique variables we will not be able to make dummy variables.