MScA 37010 Final Project

Paul Whitson

12/12/2019

Summary

Beta is a measure of the volatility of a stock price as compared to some index. The objective of this project is to use R to calculate beta (measured vs. the S&P 500 index) for all the approximately 8,000 stocks in the Wharton CRSP stock database, using a full year (2018) of daily returns data.

To calculate beta, one must first calculate the daily returns (as a proportion of the prior day's closing value) of both the individual stock and the chosen index. Beta is then given by:

- covariance(Return_of_Stock, Return_of_Index) / variance(Return_of_Index)

or, equivalently, by fitting a simple linear regression model of the form:

- Return_of_Stock = alpha + beta * Return_of_Index

Data Ingestion and Cleanup

A zipped .csv file was downloaded from the CRSP website. This was then unzipped and imported into R as follows:

```
#Load libraries:
library(ggplot2)
library(reshape2)

#Read in data:
data <- read.csv("WhartonStockDataUnzipped2018.csv")

#Create working copy of data with only the columns needed:
data2 <- data.frame(cbind(Date = data$date,
                          Ticker = as.character(data$TICKER),
                          Return = as.character(data$RETX),
                          SP_Return = as.character(data$sprtrn) ))

#Convert Date column from integer to date format:
#first convert from factor to character, then convert to date format:
data2$Date <- as.character(data2$Date)
data2$Date <- paste(substr(data2$Date, 1,4), substr(data2$Date,5,6),
                    substr(data2$Date, 7,8), sep = "-")
data2$Date <- as.Date(data2$Date, '%Y-%m-%d')

#Convert Return and SP_Return to numeric (instead of "factor").
data2$Return <- as.numeric(as.character(data2$Return))
```

```
## Warning: NAs introduced by coercion
```

```
data2$SP_Return <- as.numeric(as.character(data2$SP_Return))
#Convert Ticker to character instead of factor; this is needed for logical operators later:
data2$Ticker <- as.character(data2$Ticker)

#Note:  there are 2078 NAs in the "Return" column; exclude these from analysis:
data2 <- data2[!is.na(data2$Return),]

head(data2)
```

```
##         Date Ticker    Return SP_Return
## 1 2018-01-02   JJSF -0.017454  0.008303
## 2 2018-01-03   JJSF -0.009988  0.006399
## 3 2018-01-04   JJSF  0.013813  0.004029
## 4 2018-01-05   JJSF -0.009550  0.007034
## 5 2018-01-08   JJSF  0.000742  0.001662
## 6 2018-01-09   JJSF -0.007210  0.001303
```

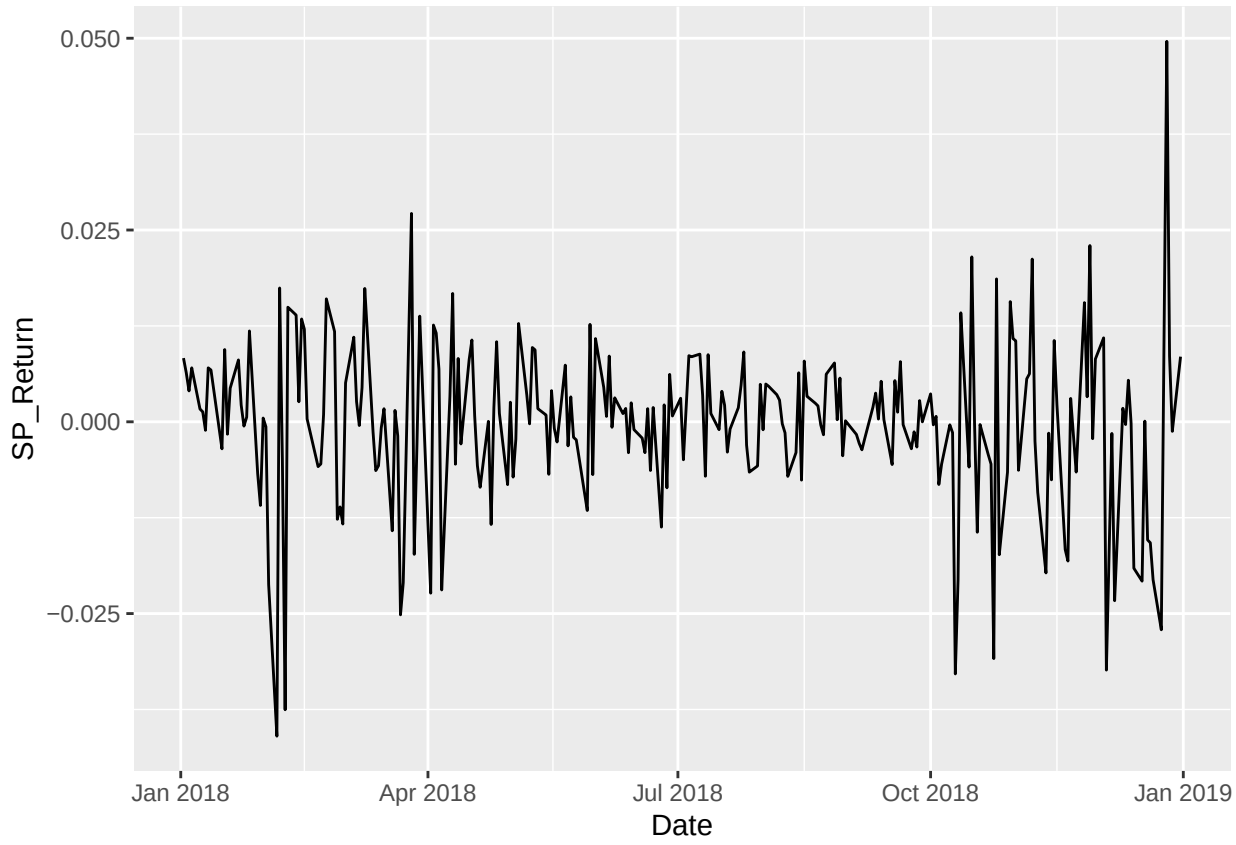
Preliminary Data Analysis: S&P 500 Index:

The output below examines the S&P 500 Index information. As shown in the histogram, the daily returns data is not normally distributed. While the median and mode are close to zero, there is a long tail to the left (negative) direction, corresponding to the period late in the year in which there were negative returns for a sustained period (as shown in the second line chart.)

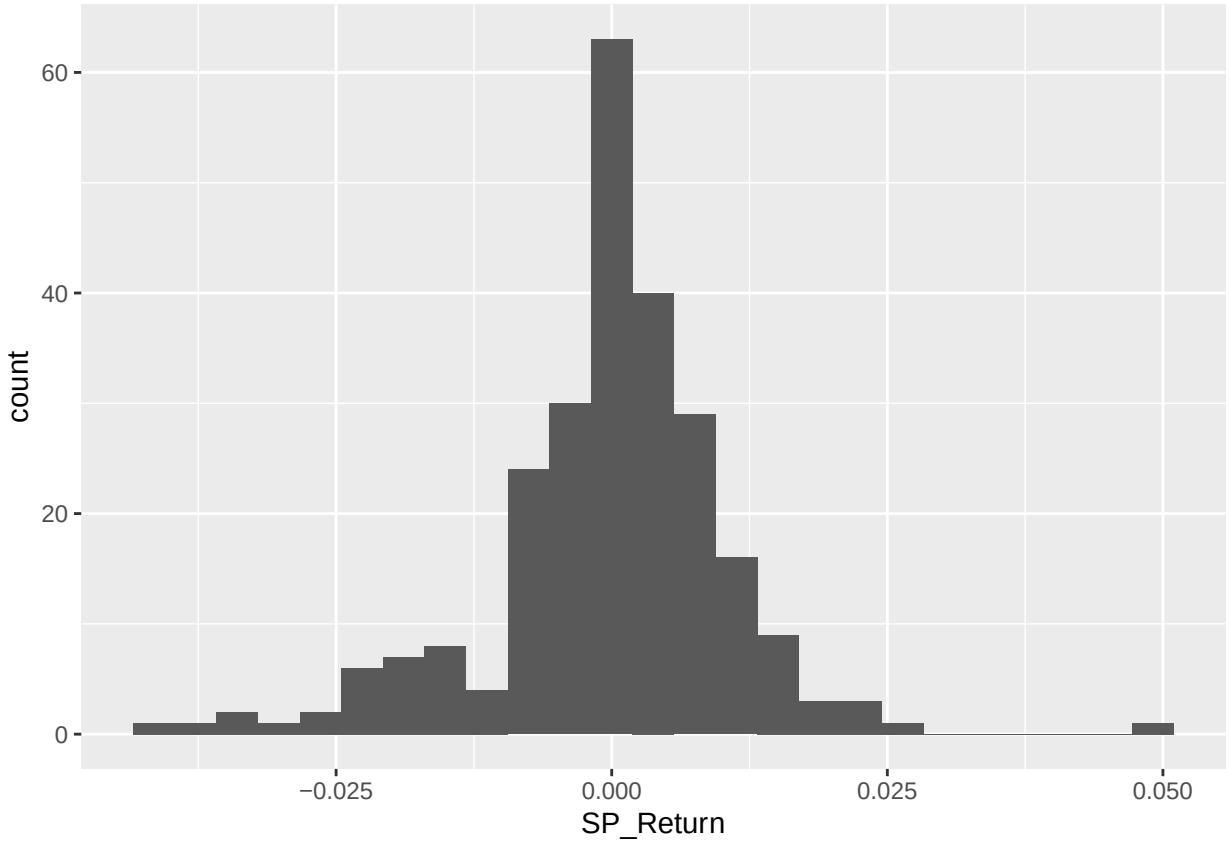
```
summary(data2$SP_Return)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0409790 -0.0049470  0.0003730 -0.0002065  0.0056000  0.0495940
```

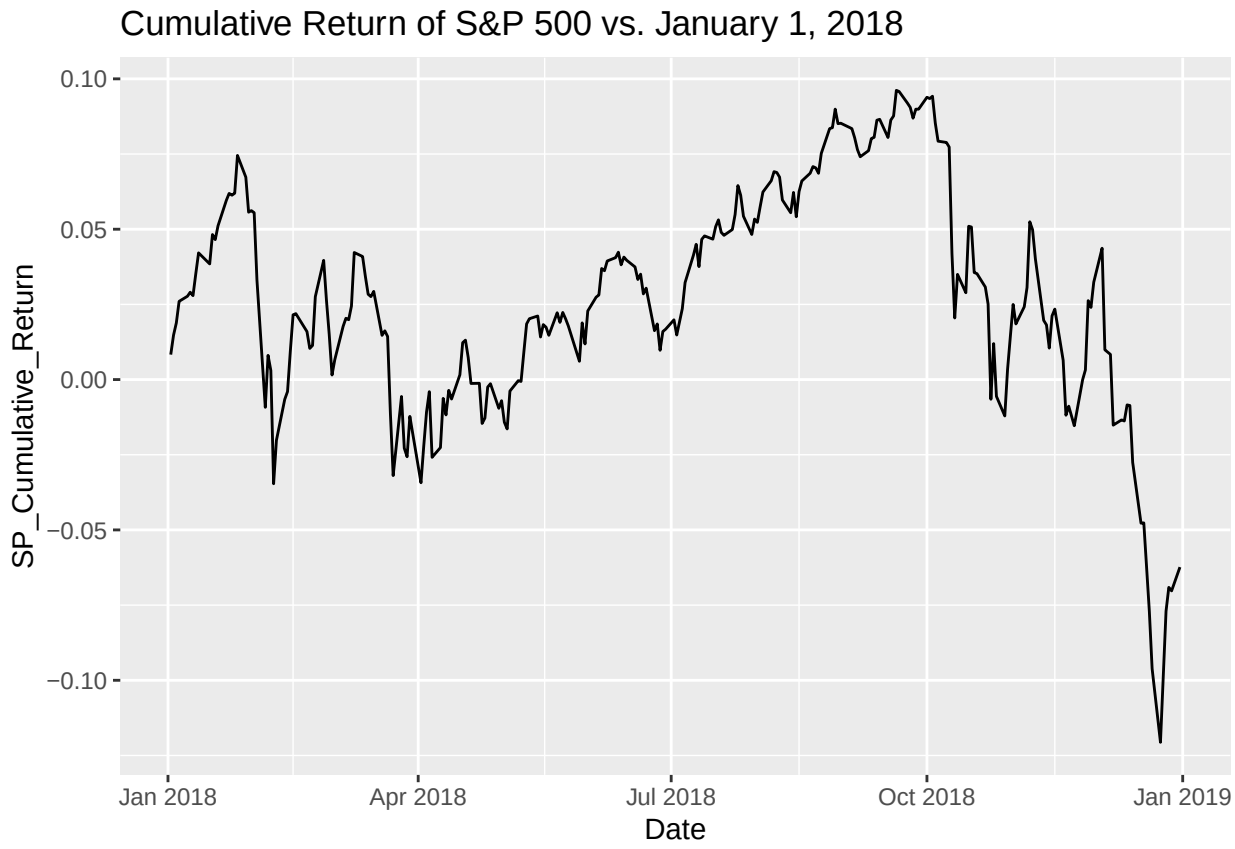
```
ggplot(data2[1:251,], aes(x=Date, y = SP_Return)) + geom_line()
```




```
ggplot(data2[1:251,], aes(x=SP_Return)) + geom_histogram(bins = 25)
```




```
#Calculate year-to-date cumulative change from January 1 value, and plot:
SPTotals <- cumprod(1+data2$SP_Return[1:251])-1
ggplot(data.frame(Date = data2$Date[1:251], SP_Cumulative_Return = SPTotals),
       aes(x=Date, y = SP_Cumulative_Return)) + geom_line() +
      labs(title = "Cumulative Return of S&P 500 vs. January 1, 2018")
```



Preliminary Data Analysis, Part 2: Individual Stocks

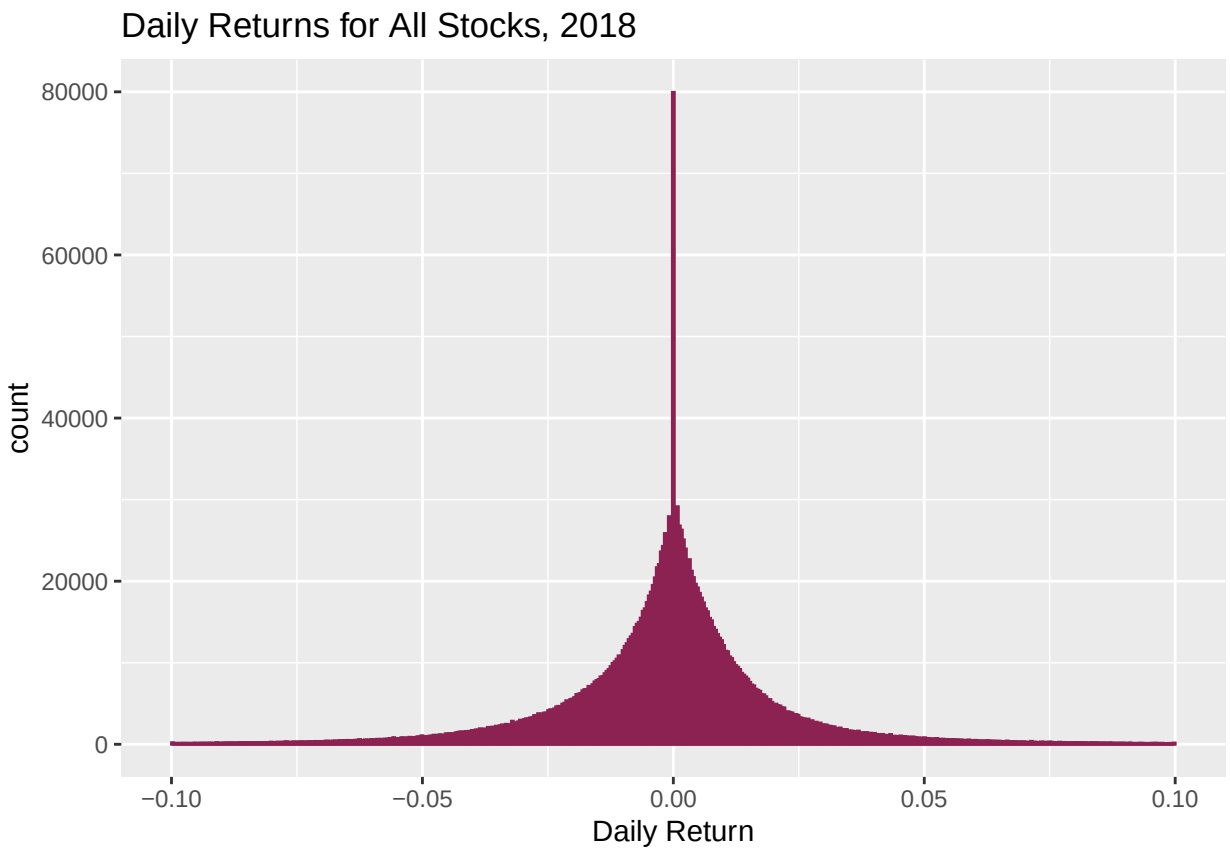
The daily returns of all ~8000 stocks were then evaluated. As shown in the histogram, these data are not normally distributed, having a high kurtosis: while fairly symmetrical, there is a large bolus of values very close to zero, and there are more data points in the tails than would be expected from a normal distribution.

```
summary(data2$Return)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.945338 -0.009434  0.000000 -0.000525  0.008197  8.977274
```

```
ggplot(data2, aes(x = Return)) +
  geom_histogram(bins = 500, color = "violetred4") +
  xlim(-0.1, 0.1) +
  labs(title = "Daily Returns for All Stocks, 2018", x = "Daily Return")
```

```
## Warning: Removed 21282 rows containing non-finite values (stat_bin).
```



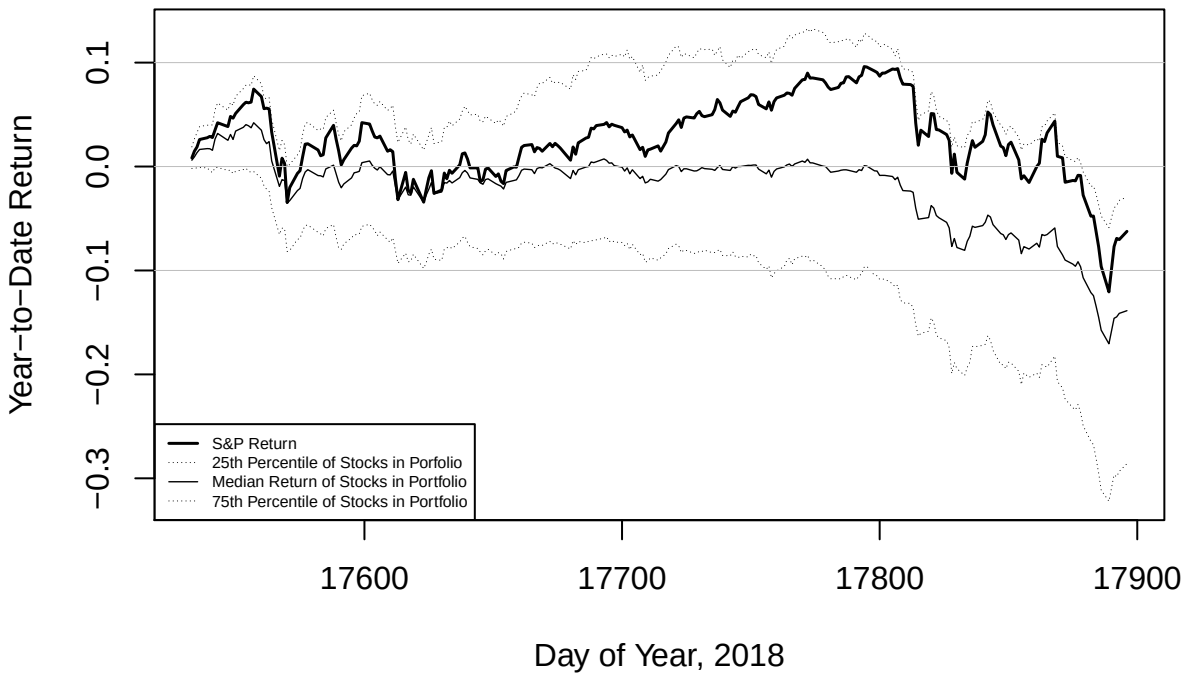
Preliminary Data Analysis, Part 3: Cumulative Returns for All Stocks

In order to understand the distribution of returns on ALL stocks in this portfolio, the year-to-date cumulative returns were calculated for every ticker symbol, for every day of the year.

Then, for each day in the year, summary statistics (minimum, maximum, and values of each quartile) were calculated and stored. Finally, the first, second (median) and third quartiles were plotted, along with the returns from the S&P 500 Index:

```
#Reshape data with the returns for every ticker symbol in its own column:
library(reshape2)
data2reshaped<-dcast(data2, Date~Ticker, value.var = "Return", mean)

#Generate a table of cumulative returns for all stocks:
#Initialize data frame:
data2cumulative <- matrix(nrow = nrow(data2reshaped), ncol = length(data2reshaped))
colnames(data2cumulative)<- names(data2reshaped)
data2cumulative[,1]<-data2reshaped$Date

#Generate cumulative change for all variables and store in data2cumulative
for (j in 2:8007) {
    data2cumulative[,j] = cumprod(data2reshaped[,j]+1)-1
}

#Initialize data frame to store quantile information for each date:
```

```r
CumulativeQuantiles = matrix(nrow = 251, ncol = 11)
colnames(CumulativeQuantiles) = c("Date", "SP_Return",
                                  "Minimum", "1st_Pct", "5th_Pct",
                                  "25th_Pct", "Median", "75th_Pct",
                                  "95th_Pct", "99th_Pct", "Maximum")
CumulativeQuantiles = as.data.frame(CumulativeQuantiles)
CumulativeQuantiles[,1]<- data2reshaped$Date
CumulativeQuantiles[,2]<- SPTotals

#For every row, generate quantiles of the year-to-date return across all ticker symbols:
for(i in 1:251) {
  CumulativeQuantiles[i,3:11] = quantile(data2cumulative[i,],
                                probs = c(0, 0.01, 0.05, 0.25, 0.5, 0.75, 0.95, 0.99, 1.0),
                                na.rm = TRUE) }

#Plot the 25th, 50th, and 75th percentiles of the year-to-date returns, along with the
#year-to-date returns of the S&P 500 Index:
matplot(CumulativeQuantiles[, 1],
        CumulativeQuantiles[, c(2, 6, 7, 8)], #Plot S&P, quartiles, and median of YTD returns
        type = "l", #all will be type "line"
        lwd = c(1.5, 0.5, 0.7, 0.5), #S&P will be heaviest, followed by median, followed by others
        lty = c(1, 3, 1, 3), #dotted for 25th and 75th, solid for others
        col = c(153, 153, 153, 153),
        xlab = "Day of Year, 2018",
        ylab = "Year-to-Date Return") #Black
legend('bottomleft', cex = 0.5,
       legend = c("S&P Return",
                  "25th Percentile of Stocks in Porfolio",
                  "Median Return of Stocks in Portfolio",
                  "75th Percentile of Stocks in Portfolio"),
       col = c(153, 153, 153, 153), lty = c(1, 3, 1, 3), lwd = c(1.5, 0.5, 0.7, 0.5))
abline(h=c(-0.1, 0, 0.1), lwd = c(0.5, 0.5, 0.5), col = "gray")
```



As shown in the chart above, the S&P 500 index performed better than the median of the 8000+ stocks; in fact, toward the end of the year, the S&P 500 was performing better than approximately 75% of these stocks.

Calculating Beta Values

In order to calculate betas, a regression model was fit between the daily returns of each individual stock and the daily returns of the S&P 500 Index. The slope parameters (betas) were then stored in a separate data frame for analysis:

```
#Extract unique ticker symbols and compute number of occurrences of each:
TickerSymbols <- as.character(unique(data2$Ticker))
OccurrencesPerTickerSymbol<-table(data2$Ticker)

#initialize data frame with 3 columns:  Ticker, Alpha, and Beta
ModelCoefficients <- matrix(ncol = 3, nrow = length(TickerSymbols))
colnames(ModelCoefficients) <- c("Ticker", "Alpha", "Beta")
ModelCoefficients <- as.data.frame(ModelCoefficients)

#For each ticker symbol, calculate alpha and beta by fitting a linear model
#between the Return of the individual stock and the SP_return.
#Write the ticker symbol, alpha, and beta to data frame ModelCoefficients.
#Some ticker symbols have only a few data points;
#if ticker symbol has 5 or more data points, fit model and write coefficients to
#data frame ModelCoefficients.If it does not, skip and move on to next ticker symbol.
```

```
for (i in 1:length(TickerSymbols)) {

  if (OccurrencesPerTickerSymbol[TickerSymbols[i]] >= 5) {

  ModelCoefficients[i,1] <- TickerSymbols[i]
  ModelCoefficients[i, 2:3] <- summary(lm(Return~SP_Return,
      data2[data2$Ticker == TickerSymbols[i],]))$coefficients[1:2,1] }

}
```

Analyzing Distribution of Beta Values

We may first calculate an "overall" value of beta for data from ALL stocks analyzed vs. the S&P 500 index. This produces an estimate of beta for the entire market of 0.676.

Note also that the estimate of alpha is negative. This is consistent with the observation above that well over half of the stocks in this dataset under-performed the S&P index; in other words, the individual stocks demonstrated some downward "drift" independent of their correlation with the S&P index.

```
#calculate a beta for all stocks together:

summary(lm(Return~SP_Return, data2))$coefficients[1:2,1]
```

```
##   (Intercept)     SP_Return
## -0.0003856274  0.6761193082
```

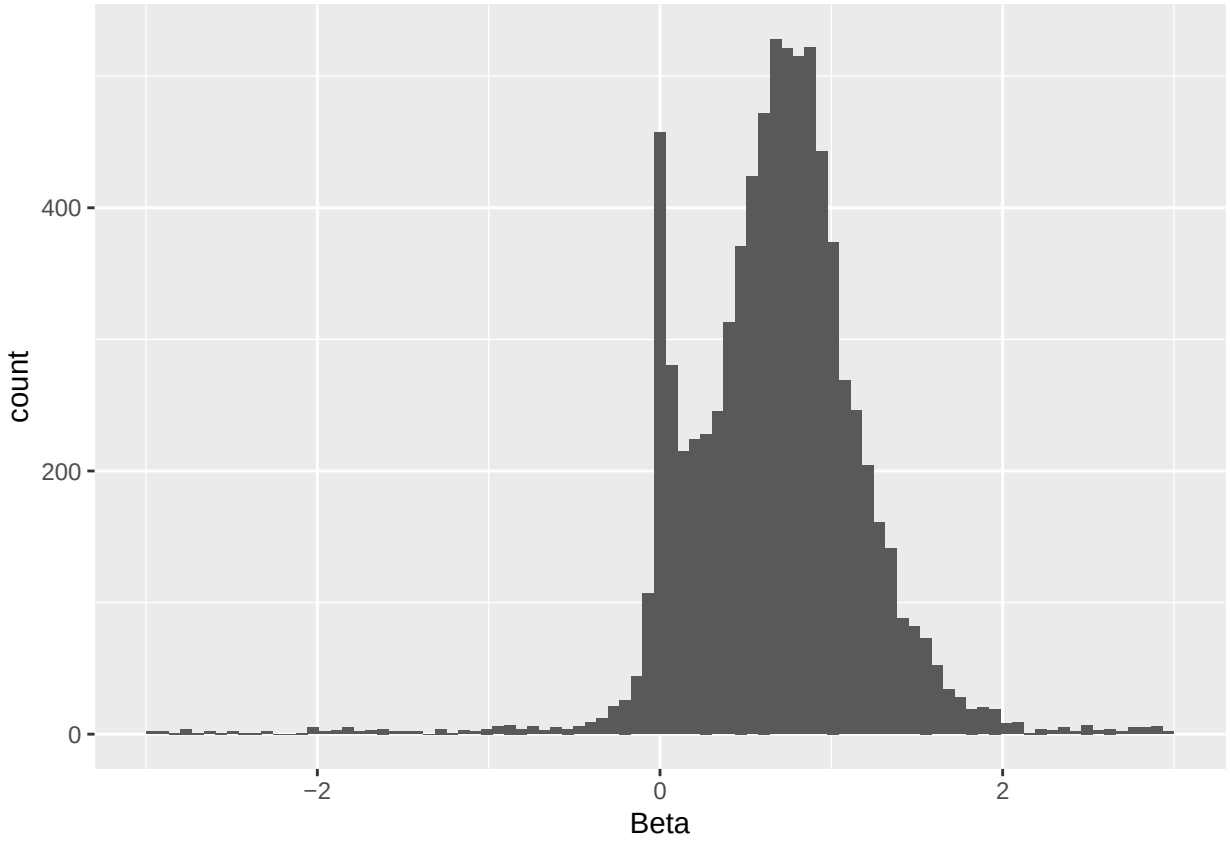
We may now evaluate the distribution of beta values for all 8000+ individual stocks in the data set.

The mean and median values of the betas are 0.667 and 0.697, respectively; this is consistent with the estimate of beta (0.676) from the analysis above.

Note that the distribution of the beta values exhibits significant deviations from normality. - There are more outliers than would be expected from a normal distribution. Some of these may be due to stocks with relatively few data points in the dataset, which may make estimation of their beta value unreliable.
- Note also that there is a spike of beta values very close to zero, indicating securities with no correlation with the market. One possible explanation could be that these securities are demonstrating very little change over time; it may also be that the data for these securities is erroneous.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## -6.5849  0.3735  0.6971  0.6674  0.9580  7.0152      17
```

```
## Warning: Removed 48 rows containing non-finite values (stat_bin).
```



Summary and Conclusions

This project demonstrated that beta values for a large number of stocks can be calculated easily using a simple loop structure in R.

In future work, it is recommended to dig more deeply into the data set to validate the data for stocks whose beta values were either: - very close to zero, or - very far from the median (>2 or <-2)