Understanding Stock Price Beahavior using an R Based Analytical Framework

Francis Kurian

11/13/2021

Introduction

Stock market behavior is a well researched area with plenty of historic information available free. Several studies link the stock price movement of a company to the perception of the market participant about that company.IBM Watson APIs enable tracking of company specific news and social media interactions through extensive text mining and help come out with a Sentiment (Negative and positive emotions expressed through words or text) Index for a business.Business Sentiment Index(BSI) is one such index calculated by TRaiCE Fintech at company level. In addition, to understand sensitivity of stock prices to Global events, Foreign Exchanges Rate is also introduced as an additional measure. Idea here is to demonstrate how to extract various relevant data elements from diverse sources, transform and load them into an analytic framework to visualize and understand the inherent relationships.

Objective

Primary objective is to develop an analysis system using various R libraries. Extracting and processing data from multiple sources, data cleaning, simplifying repetitive tasks using control structures and functions, use of data visualization techniques and application of statistical methods are the focus areas while building the framework in R. In other words, learn to write reproducible R code while analyzing stock price movement with respect to Business Sentiment Index(BSI) and Foreign Exchange Rate is the objective of this project.

List of Libraries

For this project the following Libraries are used for data validation, graphics and statistical analysis

```
library(ggplot2)
library(tidyr)
library(plyr)
library(lubridate)
library(scales)
library(reshape2)
library(summarytools) #dataframe summaries
library(ggfortify) #autoplot
library(sjPlot) #tabmodel
library(ggpubr) # wrapper for ggplot
```

Data sources, Extraction and Cleaning

Data files(.csv)used, Description and source

1.List of companies to Analyze: Internally created CSV

2.Foreign Exchange daily data for various currencies: https://www.federalreserve.gov

3.Daily Stock Prices: https://finance.yahoo.com

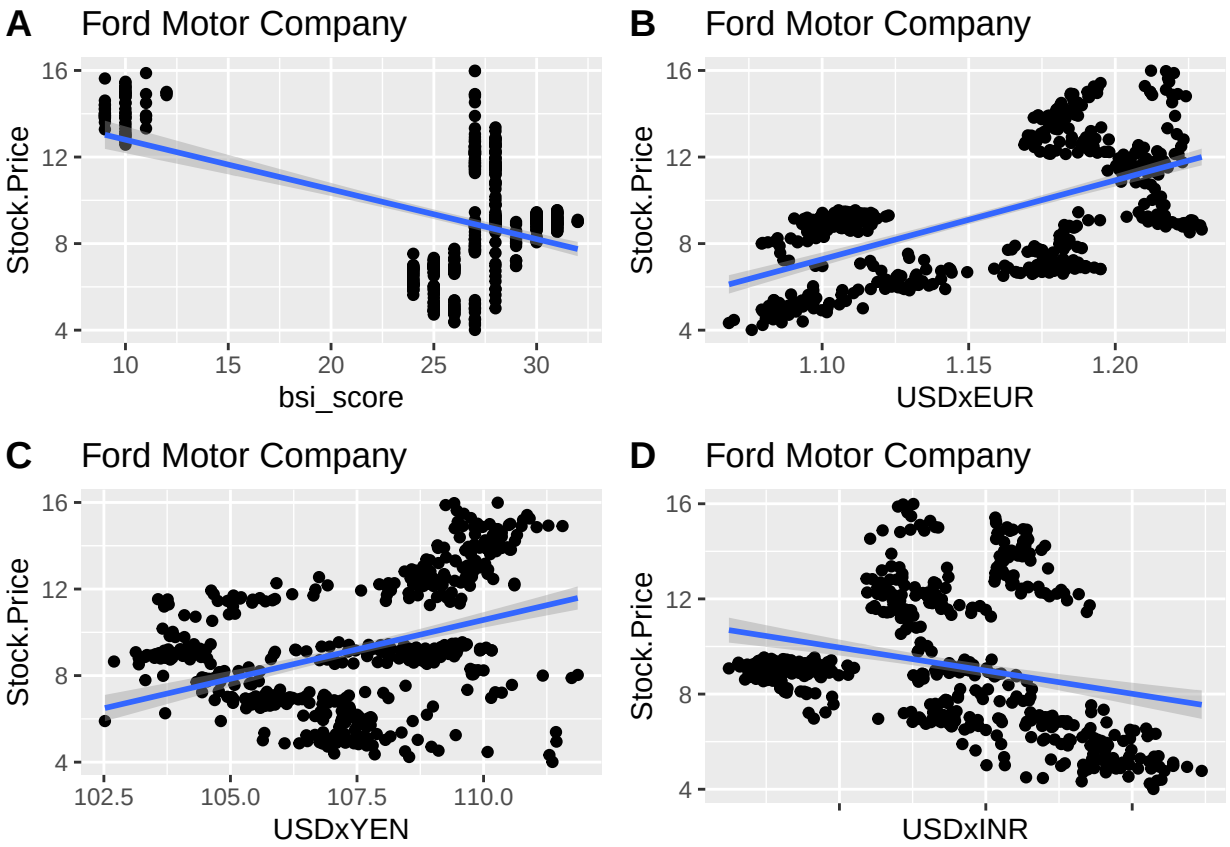
4.Business Sentiments Index(BSI,IBM Watson API based): https://www.traice.io

This project uses 6 different companies as test cases. However as long as the data files are available, companies can be added to the list and no code changes are necessary. Stock Price and BSI data will have 6 files each. Foreign Exchange is macro economic data so only one file with three different exchange rates are downloaded from the Federal Reserve. Daily data for two years(01-SEP-2019 to 31-AUG-2021) are used for the analysis.

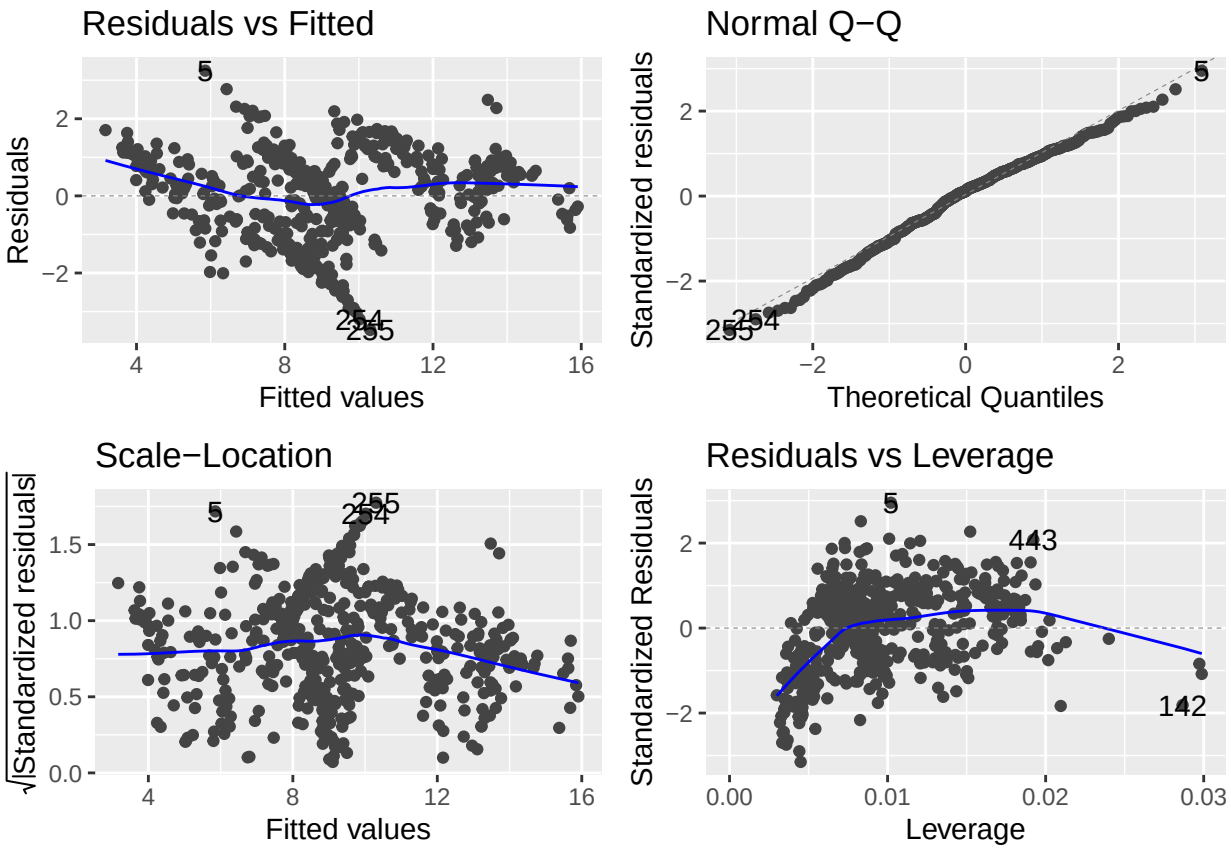
Company Level Analysis -User Interface

To Analyze a company stock price behavior(Stock.Price) users can call this function with just changing the stock ticker.Business Sentiments Index(BSI Score), US Dollar vs Euro (USDxEUR) Exchange Rate, USD vs Yen (USDxYEN)Exchange Rate, USD vs Indian Rupee (USDxINR) Exchange Rate are used as independent variables. The graphs and multiple regression results will be presented for analysis and inference. Two companies : GE and Ford are used for testing.

```
source("./src/fn_data_preperation.R")
fn.analysis("F")
```

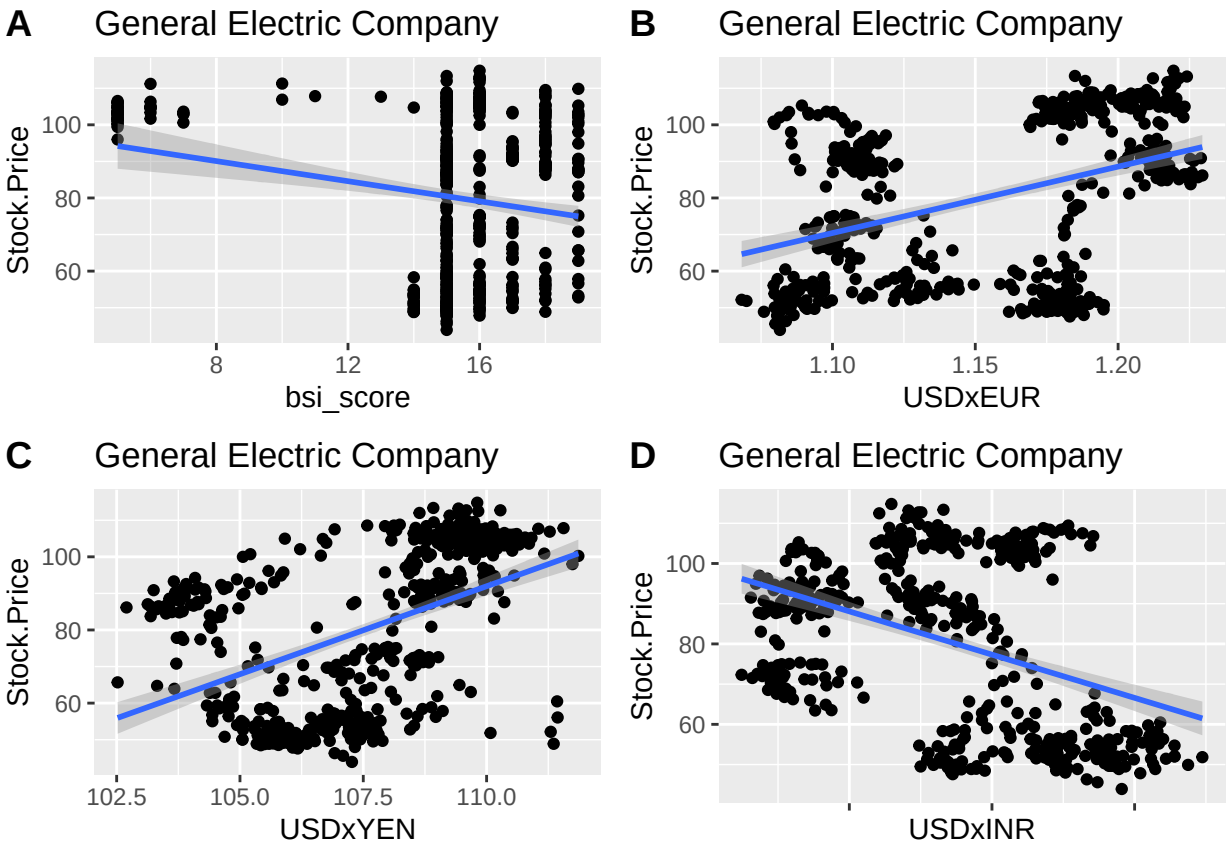



```
## MULTIPLE REGRESSION ANALYSIS RESULTS
## Call:
## lm(formula = Stock.Price ~ bsi_score + USDxEUR + USDxYEN + USDxINR,
##     data = df_all)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4760 -0.6867  0.1285  0.7732  3.2462
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -53.04353    6.21800  -8.531   <2e-16 ***
## bsi_score    -0.13374    0.01185 -11.282   <2e-16 ***
## USDxEUR      43.87533    1.29572  33.862   <2e-16 ***
## USDxYEN       0.65139    0.03109  20.951   <2e-16 ***
## USDxINR      -0.74859    0.03840 -19.495   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.106 on 493 degrees of freedom
## Multiple R-squared:  0.8544, Adjusted R-squared:  0.8532
## F-statistic:   723 on 4 and 493 DF,  p-value: < 2.2e-16
```

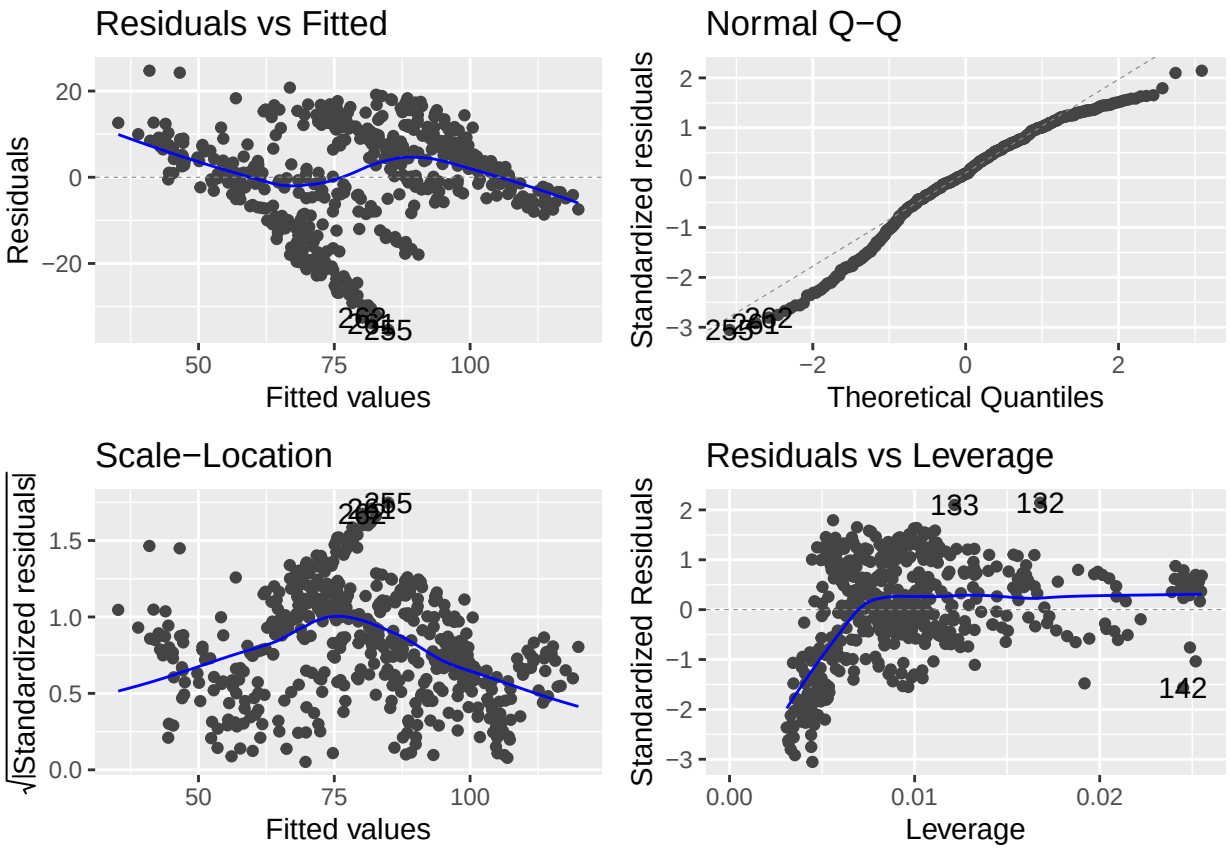



```
## [1] TRUE
```

```
fn.analysis("GE")
```




```
## MULTIPLE REGRESSION ANALYSIS RESULTS
## Call:
## lm(formula = Stock.Price ~ bsi_score + USDxEUR + USDxYEN + USDxINR,
##     data = df_all)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.418  -6.164   0.936   8.423  24.720
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -577.9548    50.9904 -11.335   <2e-16 ***
## bsi_score      0.1159     0.1905   0.608    0.543
## USDxEUR      300.7839    13.0462  23.055   <2e-16 ***
## USDxYEN        6.4028     0.2780  23.028   <2e-16 ***
## USDxINR       -5.1636     0.3490 -14.797   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.63 on 493 degrees of freedom
## Multiple R-squared:  0.7071, Adjusted R-squared:  0.7047
## F-statistic: 297.6 on 4 and 493 DF,  p-value: < 2.2e-16
```




```
## [1] TRUE
```

```
#Other companies readily available
# fn.analysis("BA")
# fn.analysis("FE")
# fn.analysis("KHC")
# fn.analysis("OXY")
```

Summary of the Analysis

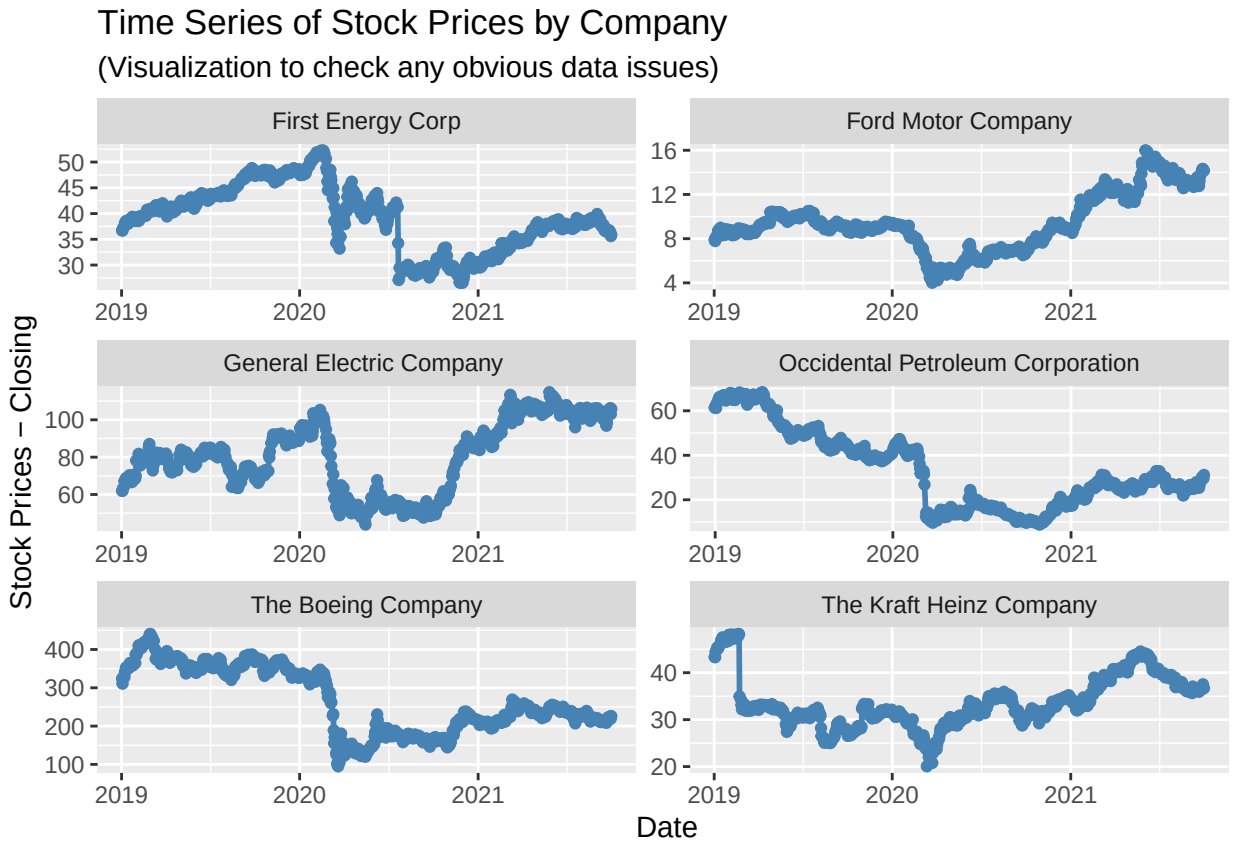
Scatterplot and multiple regression analysis demonstrate a significant relationship between Stock Prices and Business Sentiment Index and Stock Prices and foreign exchange rates in case of Ford Motor company.Adjusted R square of 85% is a good fit. BSI is negatively related to Stock prices whereas all foreign Exchange rate series (except INR) are positively related. All independent variables are significant at 99% level, based on the t test. In case of the company GE, BSI was not a significant predictor but foreign exchange rates were. The same analysis can be repeated for any 6 companies in our test list by calling the function "fn.analysis())" repeatedly with simply changing the ticker (F, GE, BA, FE, KHC, OXY).The analysis can be repeated for any listed companies as long as we ensure the data is available for the time period. This is a rudimentary analysis with multiple regression where significant effort was spent on reading and cleaning data from various sources and made several data series available for various advanced algorithms that R offers. In the future, time series properties can be tested and models like co-integration can be explored and that will add more analytic content to this project.

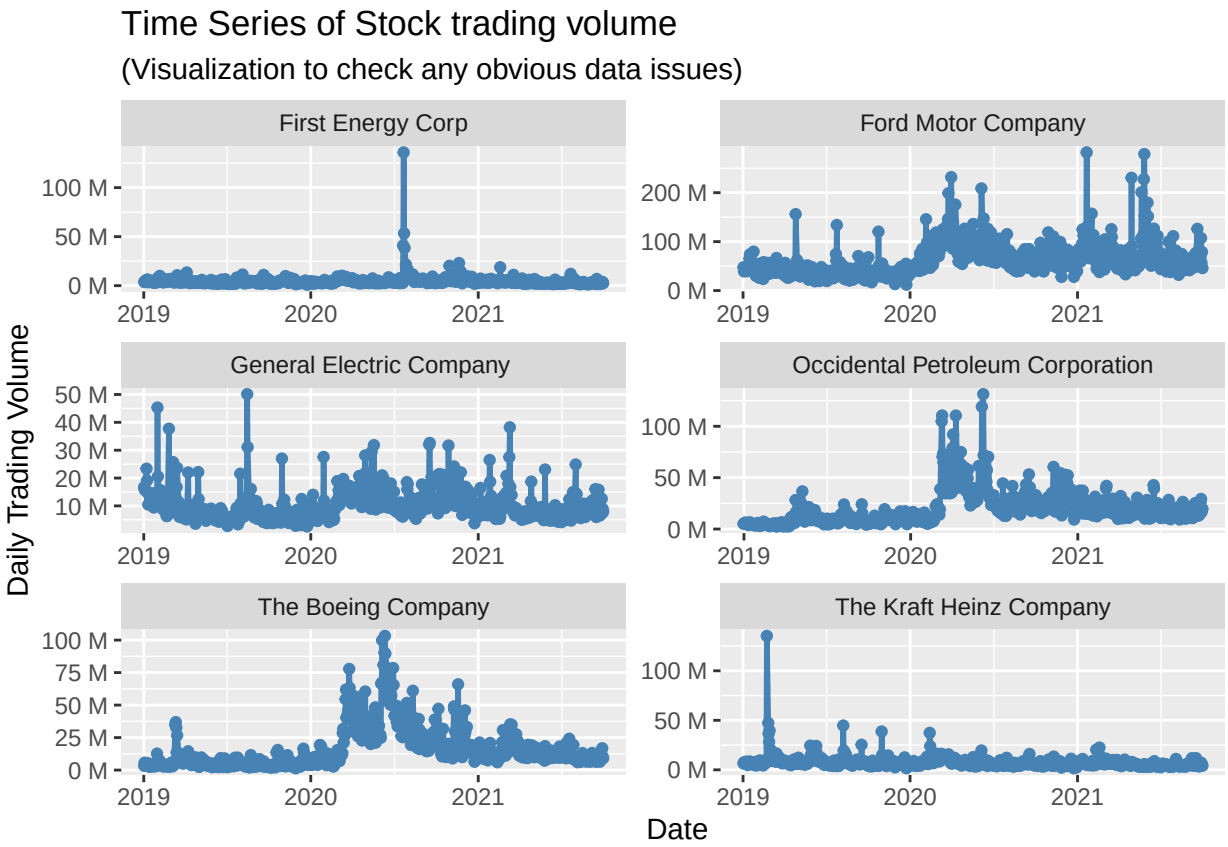
Data Cleaning and Diagnostics Process behind the Analysis

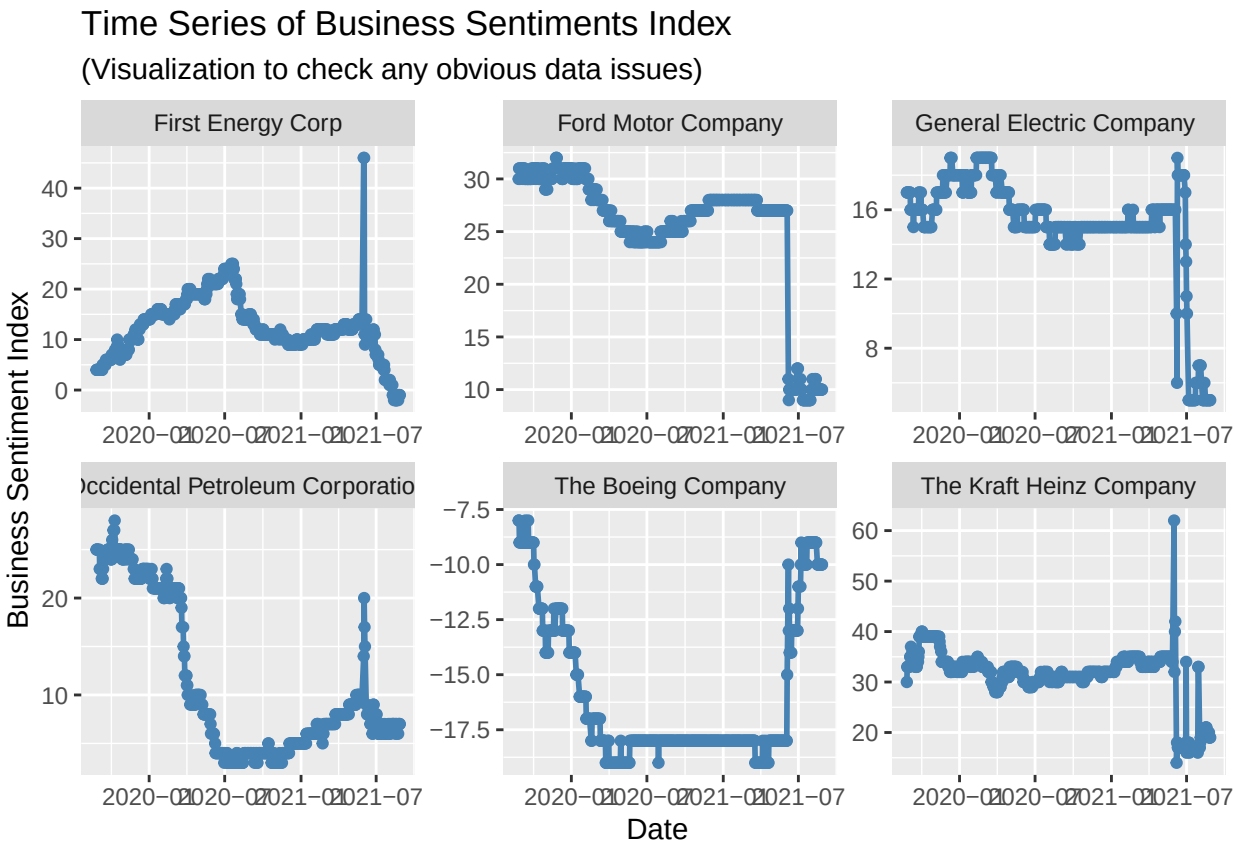
Processing CSV files(Items 1-4 in Data sources, Extraction and Cleaning sections listed above)

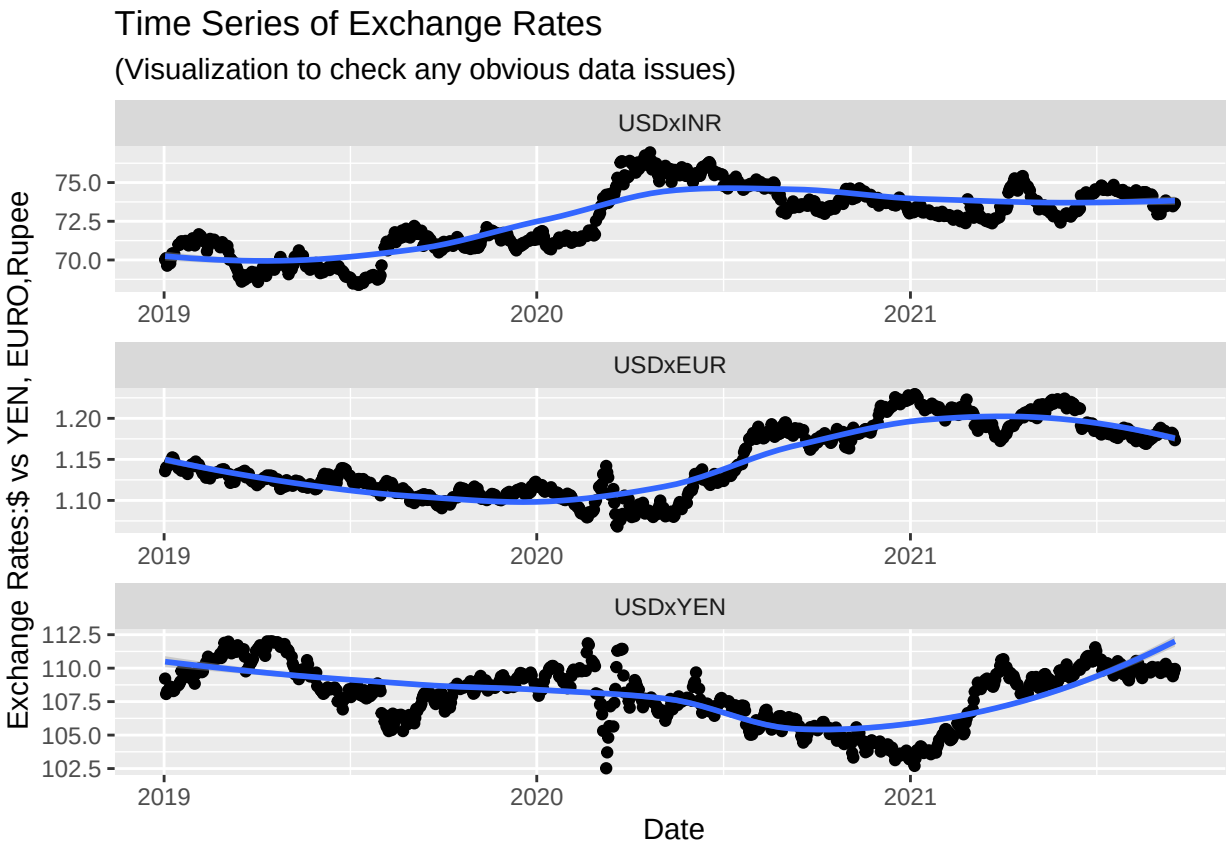
Validates source csv files are existing for the companies and alerts the user otherwise.Process the files iterative Displays data summaries and charts to detect anomalies in the data

```
fn.data_diagnostics()
```




Time Series of Stock trading volume
(Visualization to check any obvious data issues)
First Energy Corp
Ford Motor Company
General Electric Company
Occidental Petroleum Corporation
The Boeing Company
The Kraft Heinz Company
Daily Trading Volume
Date
100 M
50 M
0 M
200 M
40 M
30 M
20 M
10 M
75 M
25 M
2019
2020
2021


Time Series of Business Sentiments Index
(Visualization to check any obvious data issues)
First Energy Corp
Ford Motor Company
General Electric Company
ccidental Petroleum Corporatio
The Boeing Company
The Kraft Heinz Company
Business Sentiment Index
Date
2020-01
2020-07
2021-01
2021-07



Analysis Function - R Script

```
fn.analysis <- function(company){

CompanyNames <- './data/CompanyNames.csv'

# error trap and validation added - Peer review

  if (class(try(read.csv(CompanyNames, header = T),silent=T)) == "try-error") {
    print(message("Please change your Knit/Knit directory to 'Project Directory or
    Current Working Directory'. This error normally appears when  R Markdown knit/Knit
    Direcotry is set to 'Document directory'"))
   break  }
  else {
    df_companies <- read.csv(CompanyNames, header = T)
  }



  ForexFile <- './data/FederalReserve_CurrencyXchangeRate.csv'
  headers <- read.csv(ForexFile, skip = 5 , header = F, nrows = 1) #header info at row #6
  df_forex <- read.csv(ForexFile, skip = 6, header = F) #data begins at row #7
  colnames(df_forex) <- headers #apply headers
  df_forex$period <- as.Date(df_forex$`Time Period`) #format as date
```

```r
df_forex$USDxINR <- as.numeric(gsub("ND","",df_forex$RXI_N.B.IN))#clean & fmt as nbr
df_forex$USDxEUR <- as.numeric(gsub("ND","",df_forex$`RXI$US_N.B.EU`))
df_forex$USDxYEN <- as.numeric(gsub("ND","",df_forex$RXI_N.B.JA))

df_forex <- subset(df_forex,select = c(period, USDxINR, USDxEUR, USDxYEN))
df_forex <- df_forex[!duplicated(df_forex$period),] # remove duplicates
df_forex2 <- melt(df_forex, id.vars="period") # for all in one graphs

sapply(df_forex, class)



for (i in df_companies$ticker) {

  if (!file.exists(paste0("./data/",i, ".csv"))) {
    print("Please remove the Company. Files Missing for:")
    print(i)
    break
  }

  #process stock price/volume files from data directory
  #pick the csv file of every company in the input list and create a dataframe in a loop

  df_c <- read.csv(paste0("./data/",i, ".csv"), header = T)
  df_c$period <- as.Date(df_c$Date) # add a date field
  df_c$ticker <- i # attach ticker
  df_companies_subset <- subset(df_companies, ticker==i)
  df_c <- merge(x=df_c, y=df_companies_subset, by="ticker",all.x=TRUE )#to get name
  assign(paste0("df_",i), df_c) #create a new data frame for later use

  # Appends company data frames to one for graphs.

  if (exists("df_c_all")) {
    df_c_all <- rbind(df_c_all, df_c)

  }else {
    df_c_all=df_c}

  # Process Sentiments Index from data directory

  df_bsi <- read.csv(paste0("./data/",i, "_BSI.csv"), header = T)
  df_bsi$period <- as.Date(parse_date_time(df_bsi$created_date,c('%Y-%m-%d','%m/%d/%y')))
  df_bsi$ticker <- i # attach ticker
  df_bsi <- merge(x=df_bsi, y=df_companies_subset, by="ticker",all.x=TRUE )#to get name
  df_bsi <- df_bsi[c('ticker','period','name','bsi_score')]

  assign(paste0("df_",i,"_BSI"), df_bsi) #create a new data frame for later use

  # create a data frame to append  every data frame for graphs

  if (exists("df_bsi_all")) {
    df_bsi_all <- rbind(df_bsi_all, df_bsi)
```

```
   }else {
     df_bsi_all <- df_bsi}

 }

 #### End of data extraction and cleaning process ####



fn.regress <- function(company){

  #create local dataframes based on the company names

  if (!exists(paste0("df_",company))) {
    print("Company Datafiles Missing.Process Terminated")
    return(FALSE)
  }else {

    df_b  <- get(paste0("df_",company,"_BSI"))
    df_s  <- get(paste0("df_",company))
    df_cname <- df_companies[df_companies[,1]  == company, ]
  }


  # check all three datasets for data issues and make it global

  d1 <- dfSummary(df_forex, style = "grid", plain.ascii = TRUE)
  # view(dfSummary(df_forex, style = "grid", plain.ascii = TRUE))
  d2 <- dfSummary(df_s, style = "grid", plain.ascii = TRUE)
  d3 <- dfSummary(df_b, style = "grid", plain.ascii = TRUE)

  df_forex <- na.omit(df_forex)
  d4 <- dfSummary(df_forex, style = "grid", plain.ascii = TRUE)

  # cat(df_cname[1,2])
  # print(d1)
  # print(d2)
  # print(d4)

  # merge all three datasets

  df_all <- merge(df_b, df_forex, by="period",all.x=TRUE )  # merge with Forex file

  df_all <- merge(x=df_all, y=df_s, by="period",all.x=TRUE )  # merge with stock price

  df_all <- subset(na.omit(df_all),select=c(period,    ticker.x,   name.x, bsi_score,
                                 USDxINR,  USDxEUR,    USDxYEN,    Close,  Volume))

  # dfSummary(df_all, style = "grid", plain.ascii = TRUE)

  df_all <- rename(df_all, c("ticker.x"="ticker", "name.x"="CompanyName",
                             "Close"="Stock.Price", "Volume"="Stock.Volume"))
```

```
  d5 <- dfSummary(df_all, style = "grid", plain.ascii = TRUE) #output saved global

  # Scatterplot the variable relationship to visualize.Output saved global

  # scatter <- ggplot(df_all, aes(x=df_all[,var.X], y=df_all[,var.Y])) +
  #   geom_point()+
  #   geom_smooth(formula = y ~ x,method=lm)+ ggtitle(df_cname[1,2] ) +
  #   xlab(var.X) + ylab(var.Y)

  scatter1 <- ggplot(df_all, aes(x=bsi_score, y=Stock.Price)) +
    geom_point()+
    geom_smooth(formula = y ~ x,method=lm)+ ggtitle(df_all[1,3] )

  scatter2 <- ggplot(df_all, aes(x=USDxEUR, y=Stock.Price)) +
    geom_point()+
    geom_smooth(formula = y ~ x,method=lm)+ ggtitle(df_all[1,3] )

  scatter3 <- ggplot(df_all, aes(x=USDxYEN, y=Stock.Price)) +
    geom_point()+
    geom_smooth(formula = y ~ x,method=lm)+ ggtitle(df_all[1,3] )


  scatter4 <- ggplot(df_all, aes(x=USDxINR, y=Stock.Price)) +
    geom_point()+
    geom_smooth(formula = y ~ x,method=lm)+ ggtitle(df_all[1,3] )


  scatter <-  ggarrange(scatter1, scatter2, scatter3, scatter4 + rremove("x.text"),
            labels = c("A", "B", "C","D"),
            ncol = 2, nrow = 2)


  # Fit the simple regression model and create a summary


  multi.fit <- lm(Stock.Price~bsi_score+USDxEUR+USDxYEN+USDxINR, data=df_all)
  std_results <- summary(multi.fit) # std model results
  tab_results <- tab_model(multi.fit) # tabulated results
  residual_plot <- autoplot(multi.fit) # residuals plot

  print(scatter)



  cat("MULTIPLE REGRESSION ANALYSIS RESULTS")
  print(std_results)
  print(residual_plot)

  return(TRUE)
}

fn.regress(company)
```

```
}
```

Data Diagnostics Function - R Script

```
fn.data_diagnostics <- function(){
 # this section of the code is re-used in analysis program to create data
  CompanyNames <- './data/CompanyNames.csv'

  # error trap and validation added - Peer review

  if (class(try(read.csv(CompanyNames, header = T),silent=T)) == "try-error") {
    print(message("Please change your Knit/Knit directory to
    'Project Directory or Current Working Directory'.
    This error normally appears when
    R Markdown knit/Knit Direcotry is set to 'Document directory'"))
    break  }
  else {
    df_companies <- read.csv(CompanyNames, header = T)
  }

  # print(head(df_companies))


  ForexFile <- './data/FederalReserve_CurrencyXchangeRate.csv'
  headers <- read.csv(ForexFile, skip = 5 , header = F, nrows = 1) #header at row #6
  df_forex <- read.csv(ForexFile, skip = 6, header = F) #data begins at row #7
  colnames(df_forex) <- headers #apply headers
  df_forex$period <- as.Date(df_forex$`Time Period`) #format as date

  df_forex$USDxINR <- as.numeric(gsub("ND","",df_forex$RXI_N.B.IN)) ##clean & frmt as nbr
  df_forex$USDxEUR <- as.numeric(gsub("ND","",df_forex$`RXI$US_N.B.EU`))
  df_forex$USDxYEN <- as.numeric(gsub("ND","",df_forex$RXI_N.B.JA))

  df_forex <- subset(df_forex,select = c(period, USDxINR, USDxEUR, USDxYEN))
  df_forex <- df_forex[!duplicated(df_forex$period),] # remove duplicates
  df_forex2 <- melt(df_forex, id.vars="period") # for all in one graphs

  # sapply(df_forex, class)
  # head(df_forex)
  # summary(df_forex)



  for (i in df_companies$ticker) {

    if (!file.exists(paste0("./data/",i, ".csv"))) {
      print("Please remove the Company. Files Missing for:")
      print(i)
      break
    }

    #process stock price/volume files from data directory
```

```
   #pick the csv file of every company in the input list and create a dataframe in a loop

   df_c <- read.csv(paste0("./data/",i, ".csv"), header = T)
   df_c$period <- as.Date(df_c$Date) # add a date field
   df_c$ticker <- i # attach ticker
   df_companies_subset <- subset(df_companies, ticker==i)
   df_c <- merge(x=df_c, y=df_companies_subset, by="ticker",all.x=TRUE )#to get name
   assign(paste0("df_",i), df_c) #create a new data frame for later use

   # Appends company data frames to one for graphs.

   if (exists("df_c_all")) {
     df_c_all <- rbind(df_c_all, df_c)

   }else {
     df_c_all=df_c}

   # Process Sentiments Index from data directory

   df_bsi <- read.csv(paste0("./data/",i, "_BSI.csv"), header = T)
   df_bsi$period <- as.Date(parse_date_time(df_bsi$created_date,c('%Y-%m-%d','%m/%d/%y')))
   df_bsi$ticker <- i # attach ticker
   df_bsi <- merge(x=df_bsi, y=df_companies_subset, by="ticker",all.x=TRUE )##to get name
   df_bsi <- df_bsi[c('ticker','period','name','bsi_score')]

   assign(paste0("df_",i,"_BSI"), df_bsi) #create a new data frame forlater use

   # create a data frame to append  every data frame for graphs

   if (exists("df_bsi_all")) {
     df_bsi_all <- rbind(df_bsi_all, df_bsi)

   }else {
     df_bsi_all <- df_bsi}

 }

 #### End of data extraction and cleaning process ####

# print( summary(df_c_all) )# Stock price series
# print( summary(df_bsi_all)) #BSI index series

 print( ggplot(data = df_c_all, aes(period, Close)) +
          geom_line(color = "steelblue", size = 1) +
          geom_point(color = "steelblue") +
          labs(title = "Time Series of Stock Prices by Company",
               subtitle = "(Visualization to check any obvious data issues)",
               y = "Stock Prices - Closing ", x = " Date") +
          facet_wrap(~name,scales="free",ncol=2))

 print(ggplot(data = df_c_all, aes(period, Volume)) +
         geom_line(color = "steelblue", size = 1) +
         geom_point(color = "steelblue") +
```

```
          labs(title = "Time Series of Stock trading volume",
               subtitle = "(Visualization to check any obvious data issues)",
               y = "Daily Trading Volume", x = " Date") +
          facet_wrap(~name,scales="free",ncol=2) +
          scale_y_continuous(labels = label_number(suffix = " M", scale = 1e-6)))

  print(ggplot(data = na.omit(df_bsi_all), aes(period,bsi_score)) +
          geom_line(color = "steelblue", size = 1) +
          geom_point(color = "steelblue") +
          labs(title = "Time Series of Business Sentiments Index",
               subtitle = "(Visualization to check any obvious data issues)",
               y = "Business Sentiment Index", x = " Date") +
          facet_wrap(~name,scales="free",ncol=3) )

  print( ggplot(na.omit(df_forex2), aes(period,value)) +
           geom_point() +
           stat_smooth(formula = y ~ x,method=loess) +
           facet_wrap(~variable,scales="free",ncol=1)+
           labs(title = "Time Series of Exchange Rates",
                subtitle = "(Visualization to check any obvious data issues)",
                y = "Exchange Rates:$ vs YEN, EURO,Rupee", x = " Date"))
}
```

References

1. Understanding Diagnostic Plots for Linear Regression Analysis. https://data.library.virginia.edu/diagnostic-plots