

**Research Project**

**On**

**Stock Market Analysis**

**By**

**Mohita Sharma**

**Forum Bheda**

**Labdhi Ghelani**

[ALY6015 71131 Intermediate Analytics](https://northeastern.blackboard.com/webapps/blackboard/execute/launcher?type=Course&id=_2567936_1&url=)

Fall 2018

Submitted on: December 11, 2018

**Instructor: Prof. Li Tenglong**

**Introduction**

Stock Market:  
Stock Market is a place where shares of pubic listed companies are traded. It is a collection of markets and exchanges where the regular activities of buying, selling and issuance of shares of publicly held companies take place. Buyers/Sellers can track the price change of stocks through the stock market.

How does Stock Market work:

Investors buy shares in publicly traded companies and then sell those shares back. There are many different strategies for investing, but the basic function of the stock market comes down to investors purchasing and selling previously existing shares on the New York Stock Exchange (NYSE), Nasdaq, or other stock exchanges.

To buy shares of a stock on a stock exchange, investors go through brokers, who can get an investor a stock at a fair price, at a moment's notice. Investors simply let their broker know what stock they want, how many shares they want, and usually at a general price range. That's called a "bid" and sets the stage for the execution of a trade. If an investor wants to sell shares of a stock, they tell their broker what stock to sell, how many shares, and at what price level. That process is called an "offer" or "ask price."

**Aim**

Seemingly endless data is available for Stock market. It is difficult to collect a large dataset of stock prices which is structured, cleaned and has high granularity. Stock market is the reflection of the future and a measure of the health of the economy. It is always volatile and thoughtful investment selections that meet your goals and keep individual stock and bond risks at an acceptable level is very important. Thereby we have provided a dataset which is trying to cover regression Analysis, Time series, Data Mining along with visualization.

**Objective:**

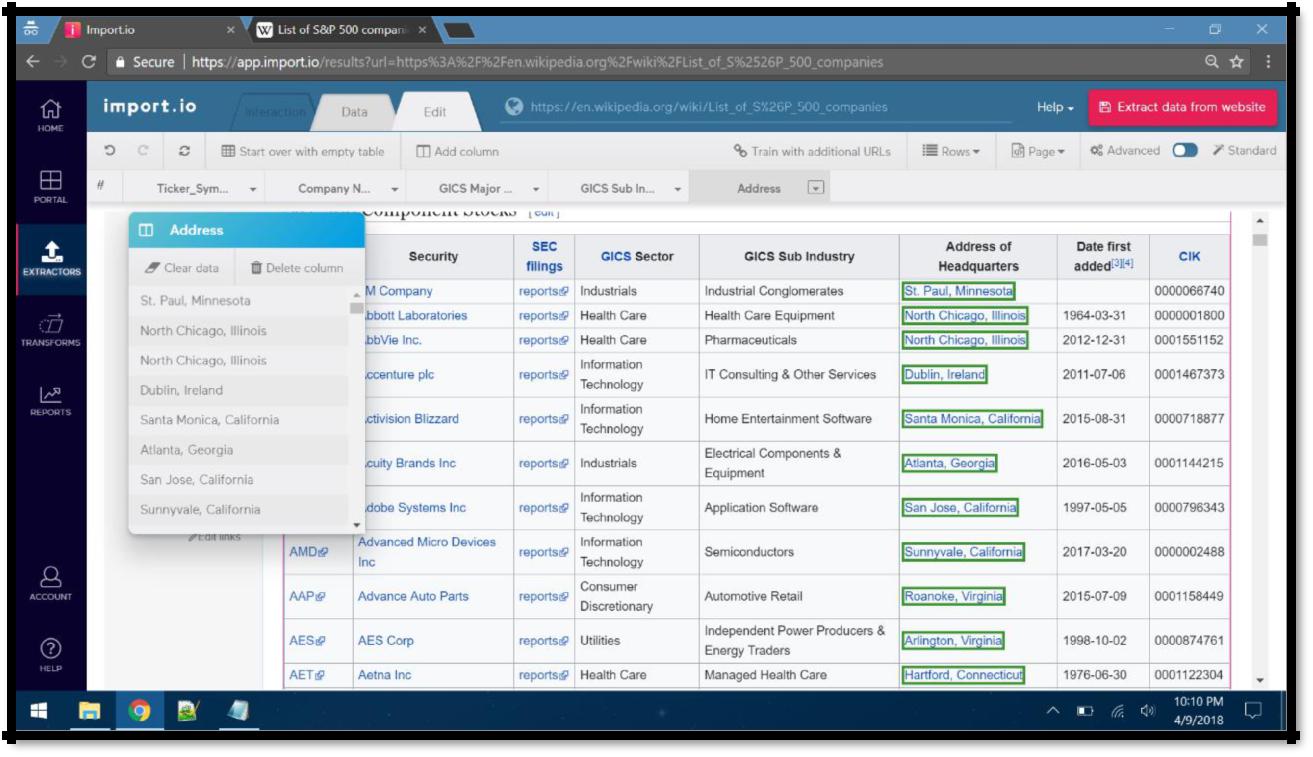
1. Collect and clean large amount of data for the year 2014-2016 on both daily and annual basis.
2. Once it is cleaned, we have created a schema and stored it in the database. We are using SQL database to store the data.
3. After the data is in the structured format, retrieve data using SQL queries.
4. We have also performed Data Analysis using R programming.
5. Additionally, performed ARIMA on highly volatile data.

**Data Collection and Cleaning**

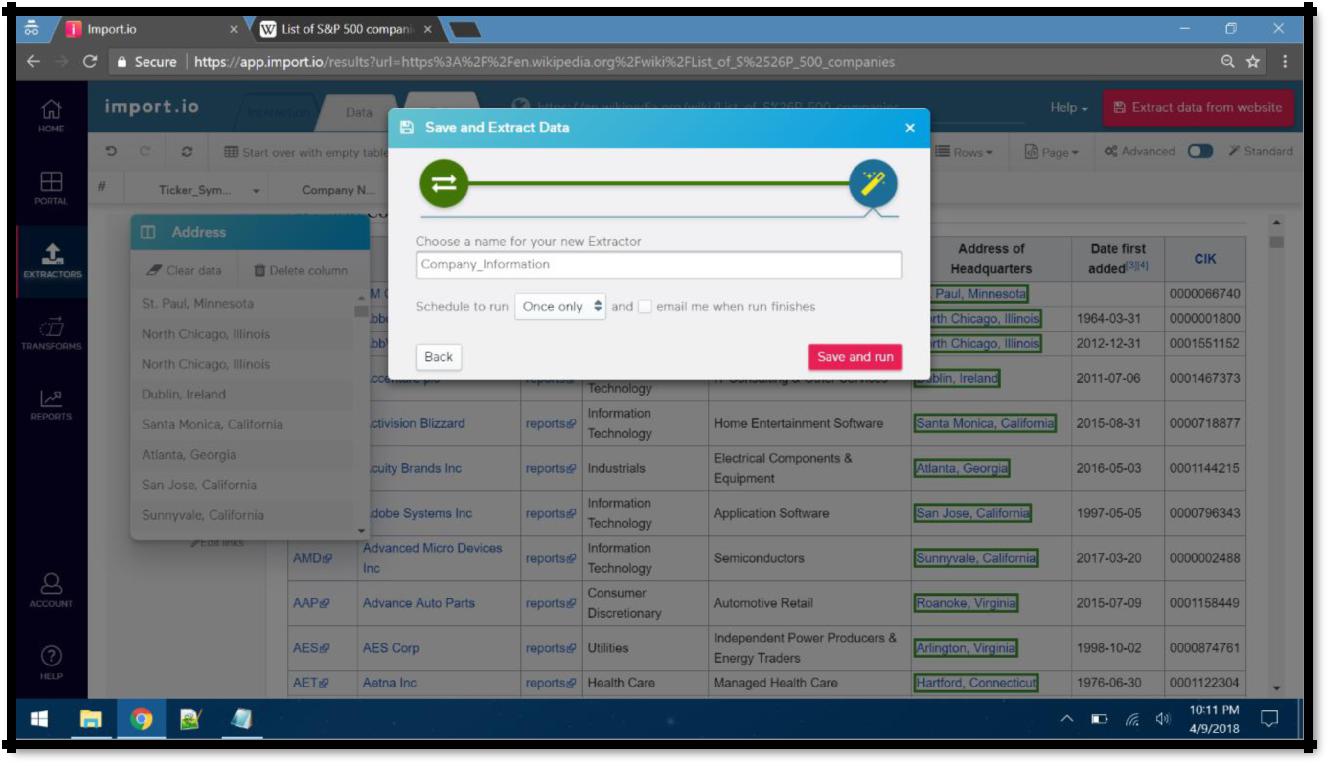
We have collected data from 2 sources and followed extraction and manipulation on both the sources:

1. Kaggle NYSE and NASDAQ data:  
   We have collected a raw dataset, which consists of daily prices of stocks from the year 2010 to the year 2016.  
   Also, we have taken another dataset which consists of Fundamentals of all the stocks. We
2. S&P Dow Jones Indices:  
   We have collected this dataset from Wikipedia. It is maintained by S&P Global company and we are using this data because it covers almost 80% of American equity market by capitalization.

**Step 1**: Web Scrapping of Data using import.io:

1. We have used web scraping for Dow Jones Index data. Using import.io data scrapping tool, we have collected the necessary data for 500 companies required for this project.

1. From the given table, we have only selected columns needed for storing it in schema.   
     
   A screenshot of a computer

   Description generated with very high confidence
2. Saving the data in .csv format:

1. The final dataset has 519 rows:

A screenshot of a computer screen

Description generated with very high confidence

**Step 2:** Collecting Stock data:

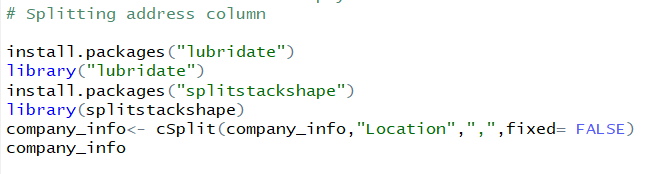
We have collected data for listed 500 companies from source 1 for both daily basis and annual basis. The daily stock dataset contains 851013 rows and annual stock file contains 1781 rows. We have also planned to work with stocks coming from the dataset, since it will help in performing analysis on huge dataset. Also, it will help in forecasting. The code for Data Collection is as:

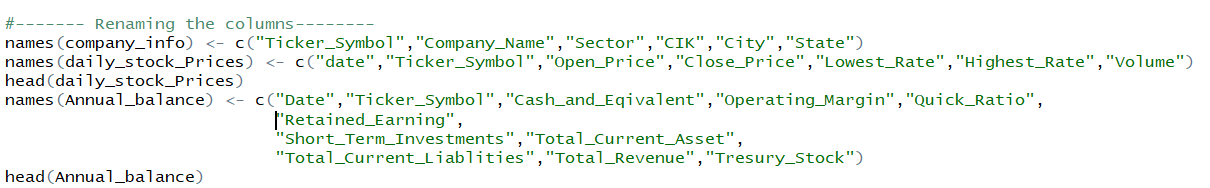
A picture containing screenshot

Description generated with very high confidence

**Step 3:** Data Cleaning:

A lot of time was put in cleaning the data:

1. Splitting the Location column:  
     
   
2. Deleting unwanted columns from the dataset:  
     
   A picture containing screenshot

   Description generated with high confidence
3. Renaming certain columns:  
     
   
4. We have also replaced missing values in columns:  
     
   A screen shot of a social media post

   Description generated with very high confidence
5. One common standards for certain Column names:  
   There are certain rows which have major sector as IT, while some have Information Technology, so we have assigned them one common name as follows:  
     
   A close up of a logo

   Description generated with high confidence
6. Cleaning date column, formatting and factorizing it into different columns for Analyzing:  
     
   Screen of a cell phone

   Description generated with high confidence
7. Creating new dataframes for volume of stocks traded every month, day, year, weekday.  
     
   A screenshot of a social media post

   Description generated with very high confidence

**Data Storage**

Once the data was collected and cleaned, next step was to create the data structure which follows 3rd Normal Form(3NF). 3NF is used in normalizing the database design to reduce the duplication of data and ensuring referential integrity.

We created 3 tables by establishing the connection with the RSQLite package.

**Step 1: Creating schema design**

1. Company Information

This table has company information

* Ticker\_Symbol
* Ticker\_Symbol is the primary key for Company Information table
* It contains companies Stock’s Name
* Company\_Name
* It shows the Name of the company a stock belongs to.
* Sector
* Provides the main domain of a company.
* Eg: Healthcare Industry, Information Technology
* City
* It provides the City in which the Headquarter of the company is located.
* State
* Contains state in which headquarter of the company is located.

1. Daily Stock Information

This schema has stock information on a daily basis.

* Daily\_Stock\_Id
* It is a primary key to the Daily Stock Information table.
* date
* The daily date for which prices are displayed.
* Ticker\_Symbol
* It is a foreign key from the Company Information table. (schema 1)
* Open\_Price
* The price at which the stock market opens for a day.
* It need not be the same as previous day’s closing price.
* Close\_Price
* The price at which the stock market closes for the day.
* Lowest\_Rate
* The lowest price of a stock for one day.
* Highest\_Rate
* The highest price of a stock for one day.
* Volume
* Total number of stocks traded in one day.

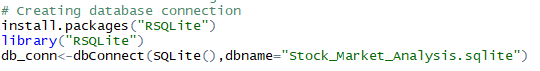
1. Annual Balance Information

This table has stock information on an annual basis.

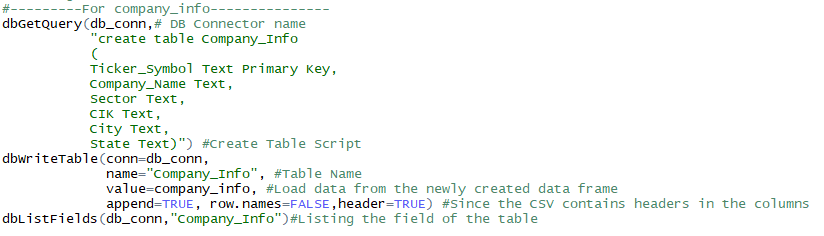
* Annual\_Stock\_Id
* It is the primary key for Annual Balance Information.
* Date
* Since the balance sheet is calculated annually, this specifies the year for which it was calculated.
* Ticker\_Symbol
* It is a foreign key from Company Information Table (schema 1).
* Cash\_and\_Eqivalent
* Value of a company's assets that are cash or can be converted into cash immediately.
* These include bank accounts, marketable securities, commercial paper, Treasury bills and short-term government bonds with a maturity date of three months or less
* Operating\_Margin
* Operating margin is a measurement of what proportion of a company's revenue is left over after paying for variable costs of production such as wages, raw materials.
* Operating margin gives analysts an idea of how much a company makes on each dollar of sales.
* Quick\_Ratio
* The quick ratio is an indicator of a company’s short-term liquidity and measures a company’s ability to meet its short-term obligations with its most liquid assets.
* Because we're only concerned with the most liquid assets, the ratio excludes inventories from current assets.
* Quick assets are current assets that can be converted to cash within 90 days or in the short-term.
* If greater than 1, liquid assets can cover for short-term investments.
* If less than 1, the company may not be able to pay off their debts
* Retained\_Earning
* Retained earnings refer to the percentage of net earnings not paid out as dividends but retained by the company to be reinvested in its core business, or to pay the debt.
* They can show a positive earnings accumulation or can turn negative and have a deficit if a current period's net loss exceeds the period's beginning retained earnings.
* Short\_Term\_Investments
* Debt incurred by a company that is due within one year.
* The value of the short-term debt account is very important when determining a company's performance.
* If the account is larger than the company's cash and cash equivalents, this suggests that the company may be in poor financial health and does not have enough cash to pay off its short-term debts.
* Total\_Current\_Asset
* The total sum of all the available assets of the company for a particular financial year.
* Total Assets and Total Current Assets are different
* Total\_Current\_Liablities
* The total sum of all the available liabilities of the company for a particular financial year.
* Total liabilities and Total Current liabilities are different.
* Total\_Revenue
* It is the top line or gross income figure from which costs are subtracted to determine net income.
* Revenue is calculated by multiplying the price at which goods or services are sold by the number of units or amount sold.
* Tresury\_Stock
* A portion of shares that a company keeps in its own treasury.
* Treasury stock may have come from a repurchase or buyback from shareholders, or it may have never been issued to the public in the first place.
* These shares don't pay dividends, have no voting rights and should not be included in shares outstanding calculations.

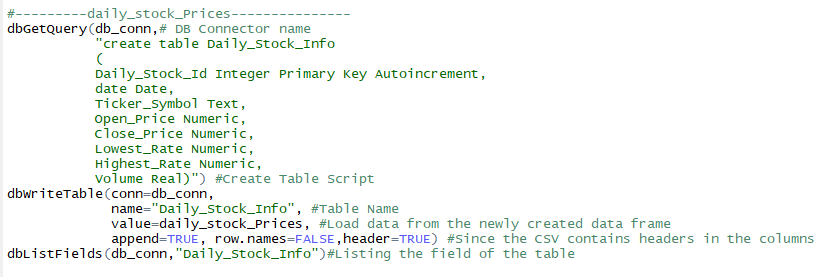
**Step 2: Setting up Connection**

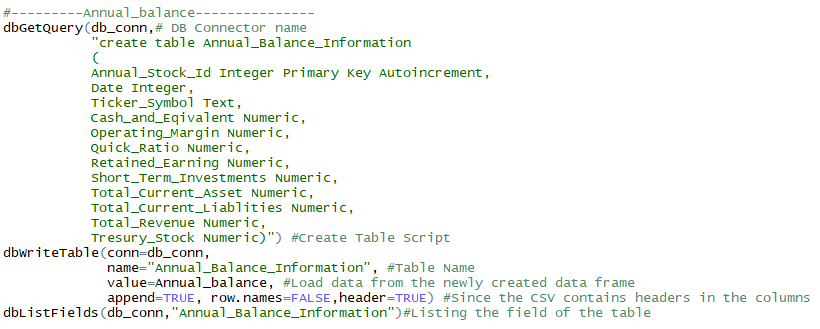
Initializing Database connection using SQLite.



**Step 3: Creating Tables in the database**



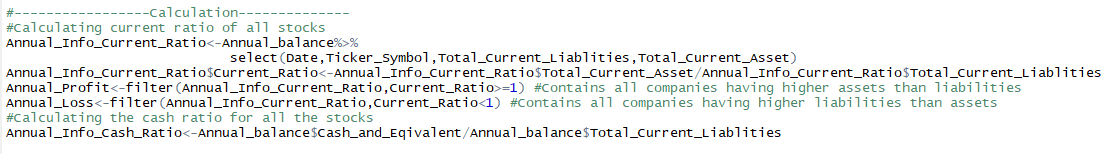




**Calculation**

We have done few calculations on Current Ratio, Cash Ratio, %Change in volume/week and %Change in price/week was not present in our dataset. This calculation helps in knowing fluctuation of stocks in the stock market per week and the liability of a company. This additional calculation makes the analysis even more interesting and helps an intelligent trader invest wisely.

* Current Ratio
* It is a ratio of Total Current Assets to Total Current Liabilities.
* If greater than 1, company has more assets than liabilities.
* If less than 1, company has more liabilities which is not good.
* If equal to 1, company has zero assets left.
* Cash Ratio
* The ratio of company’s cash and cash equivalents to its current liabilities.
* If the company is forced to pay all current liabilities immediately, this metric shows the company's ability to do so without having to sell or liquidate other assets.
* If a company's cash ratio is equal to 1, the company has the same amount of current liabilities as it does cash and cash equivalents to pay off those debts.
* If a company's cash ratio is less than 1, there are more current liabilities than cash and cash equivalents. In this situation, there is insufficient cash on hand to pay off short-term debt.
* If a company's cash ratio is greater than 1, the company has more cash and cash equivalents than current liabilities. In this situation, the company can cover all short-term debt and still have cash remaining

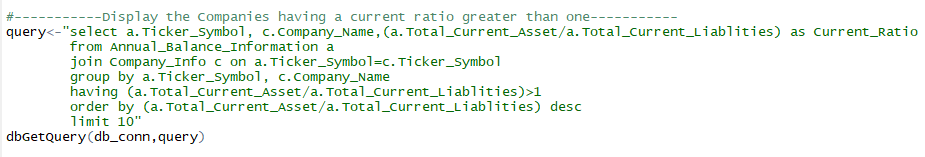


**Data Retrieving**

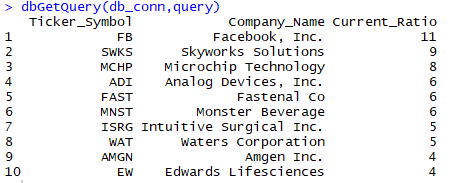
**Analysis using SQL queries**

To transform structured data into meaningful information. We performed analysis by retrieving data using SQL queries.

1. **Display the Companies having a current ratio greater than one.**



Output:

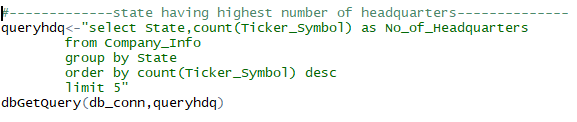


Result:

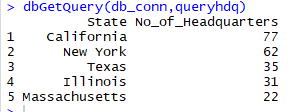
Current ratio of Facebook is 11. Facebook has the highest current ratio followed by Skyworks Solutions with a current ratio of 9.

Microchip Technology, Analog Devices, Inc., Fastenal Co, Monster Beverage, Intuitive Surgical Inc., Waters Corporation, Amgen Inc. and Edwards Lifesciences having the current ratio of greater than one. It means these 10 companies have more assets than liabilities.

1. **Which state has the highest number of headquarters?**



Output:

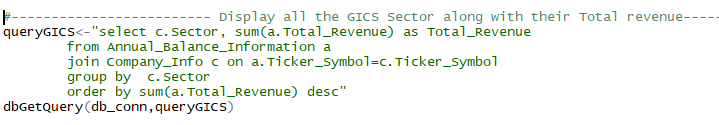


Result:

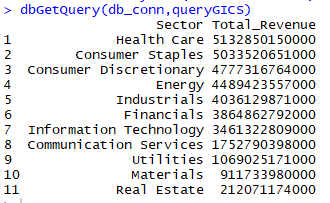
California has the highest number of Headquarters i.e. 77, which means maximum money flow from there.

New York is the second highest state with 62 no. of headquarters followed by Texas, Illinois, and Massachusetts.

1. **Display all the GICS Sector along with their Total revenue.**



Output:



Result:

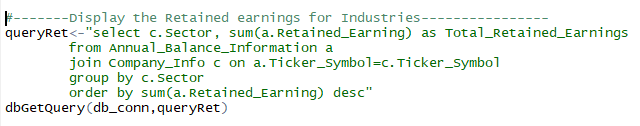
Among Sector, Health Care has the highest total revenue.

Consumer Staples is very near to Health Care.

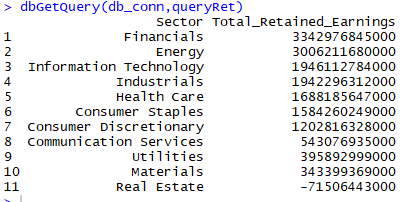
Consumer Discretionary has the third highest Revenue.

Real Estate has the lowest revenue.

1. **Display the Retained earnings for Industries**



Output:



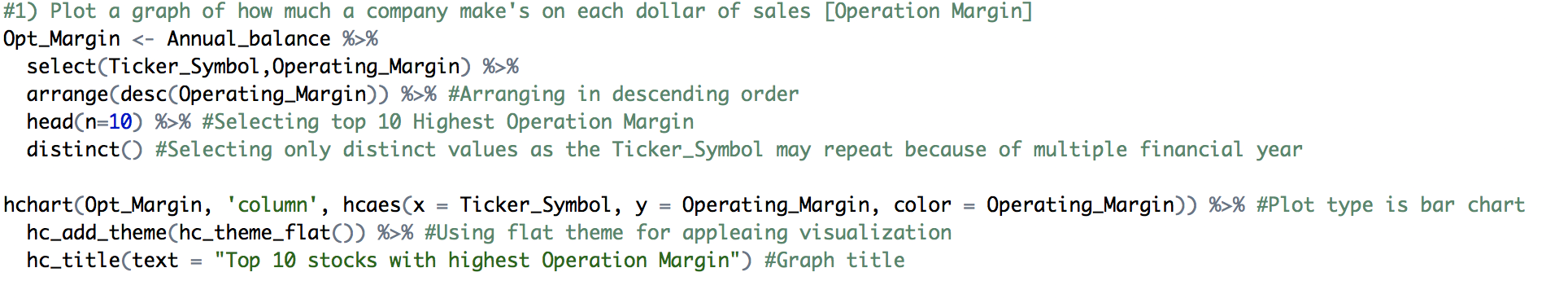
Result:

Financial Sector has retained earnings of 3342976845000. Financial Sector has the highest net earning which is not paid out as dividends but retained by the company to be reinvested in its core business whereas Real Estate have the lowest retained earnings.

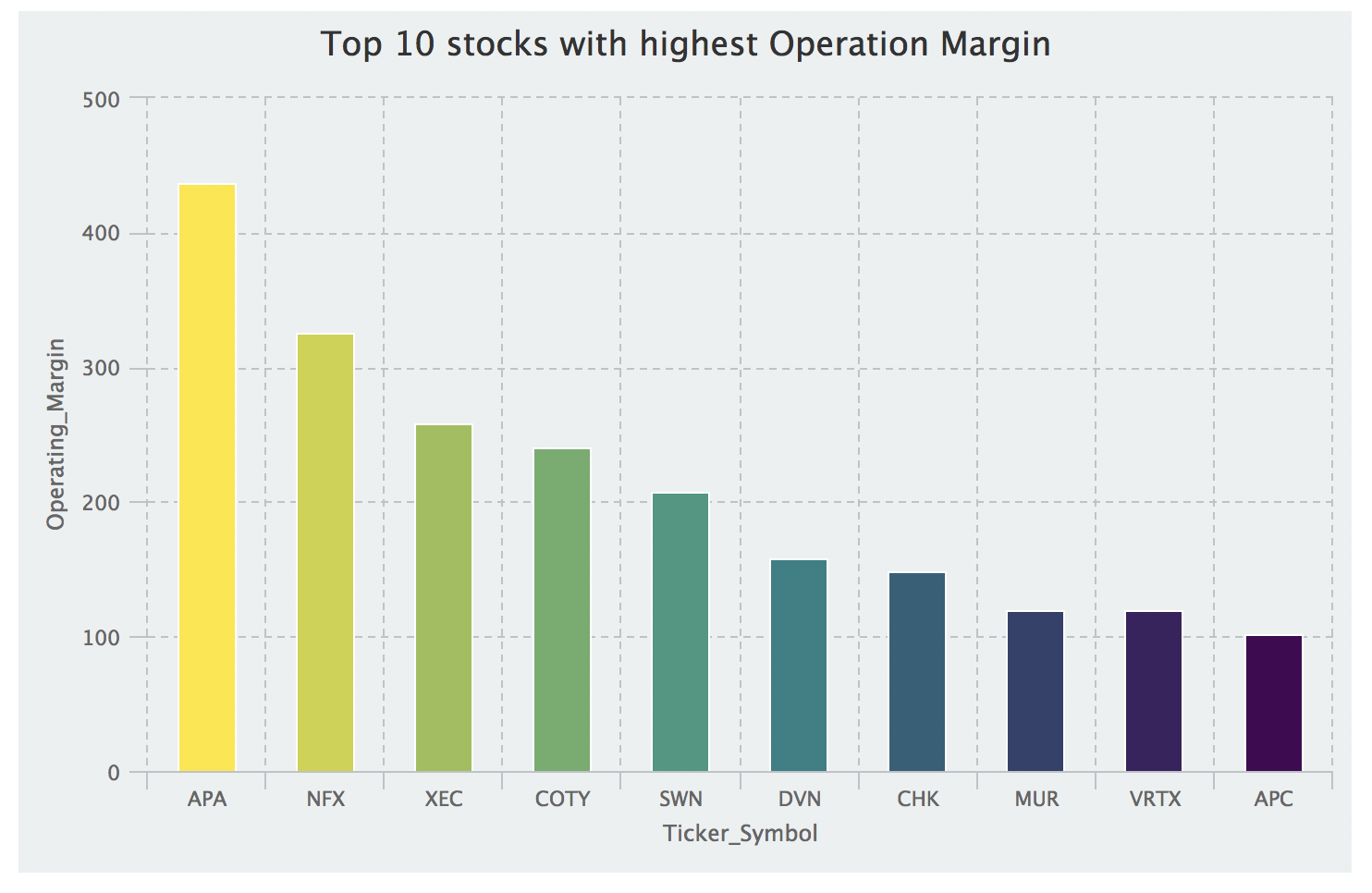
**Data Visualization using R libraries**

To perform advanced analysis, which gives more clarity in understanding the data. We have performed Data Visualization using R Libraries.

**1) Plot a graph of how much a company makes on each dollar of sales [Operation Margin]**

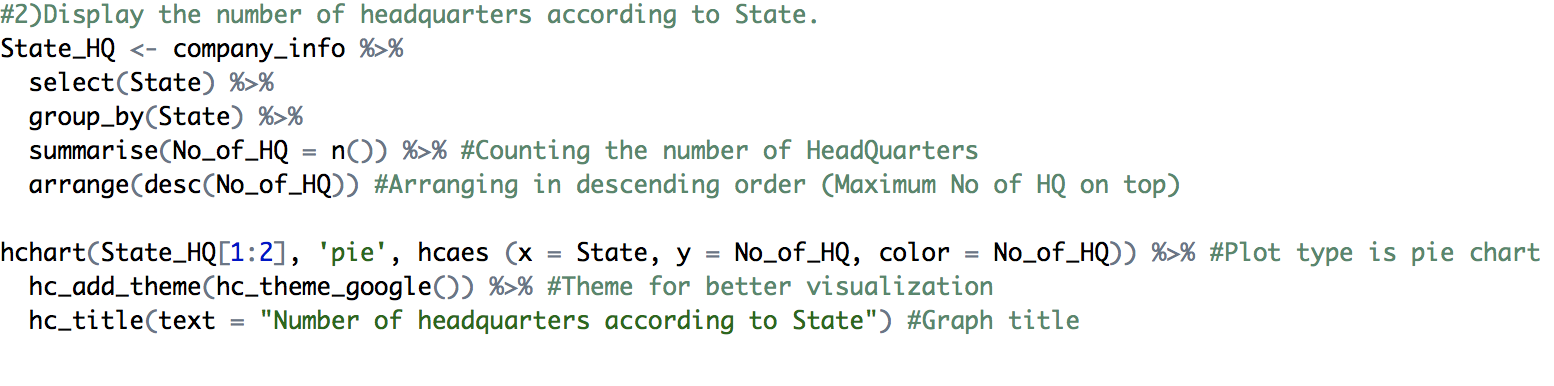


Output:

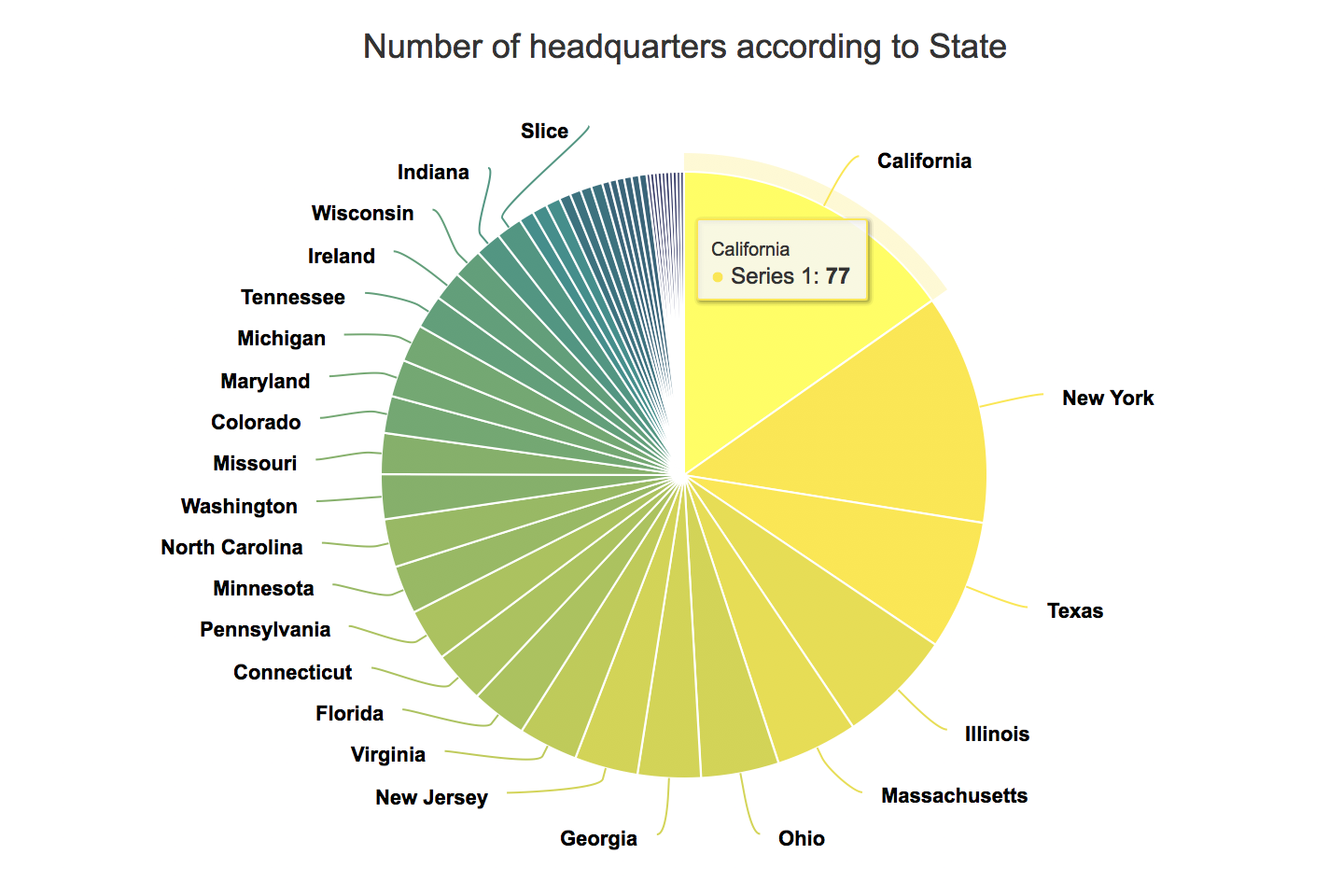


Result:  
- Operation Margin measures **profitability**. APA stocks have the highest Operation Margin which says they earn the highest profit on each dollar of revenue.  
- APA stocks belong to Apache Corporation. Investing in APA stocks seems to be a safe bet for investors. The bar graph displays top 10 stocks having highest Operation Margin.

**2) Display the states and their headquarters. (Fortune 500 companies)**

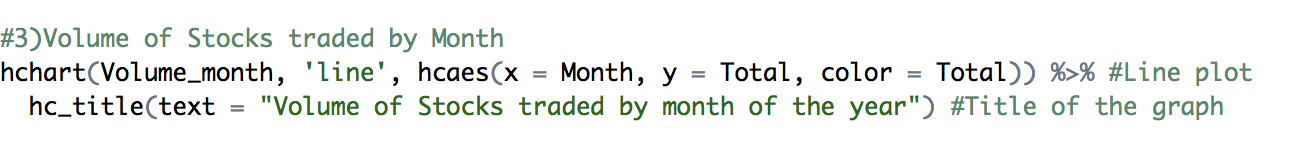


Output:

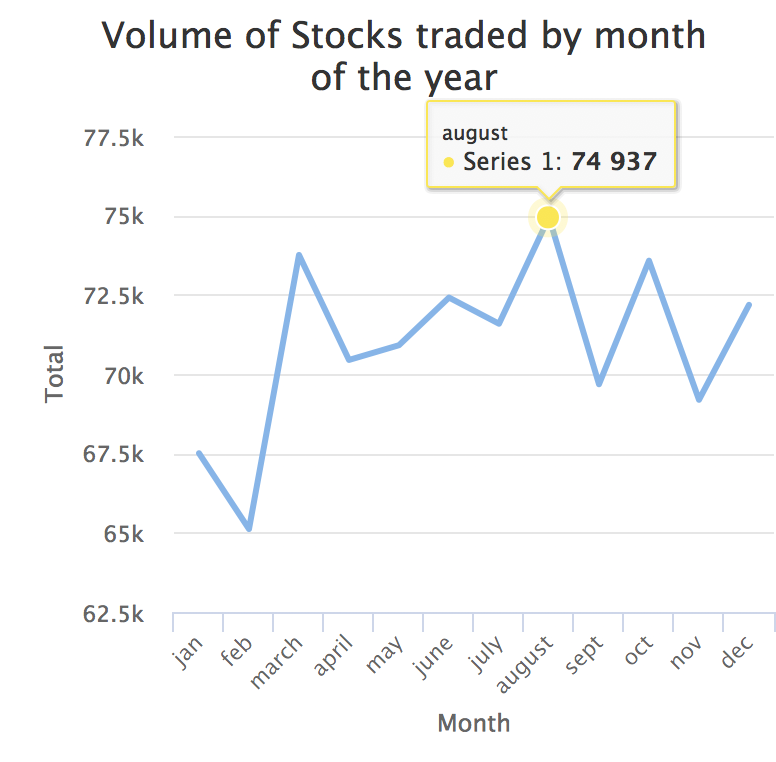


Result:  
- California has the 77 no. of headquarters from top 500 companies. Followed by New York, Texas, Illinois and Massachusetts. These are the top 5 states having highest no of headquarters.

**3) The volume of Stocks traded by Month**

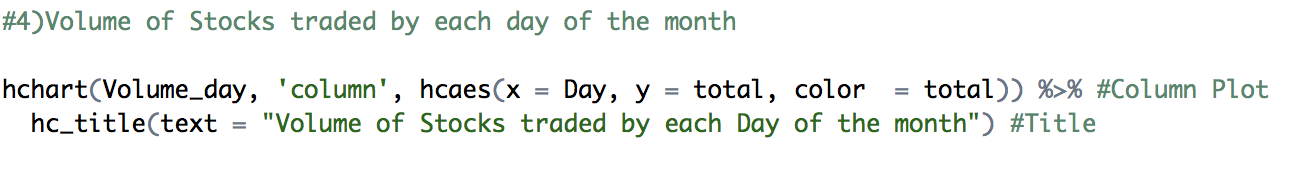


Output:

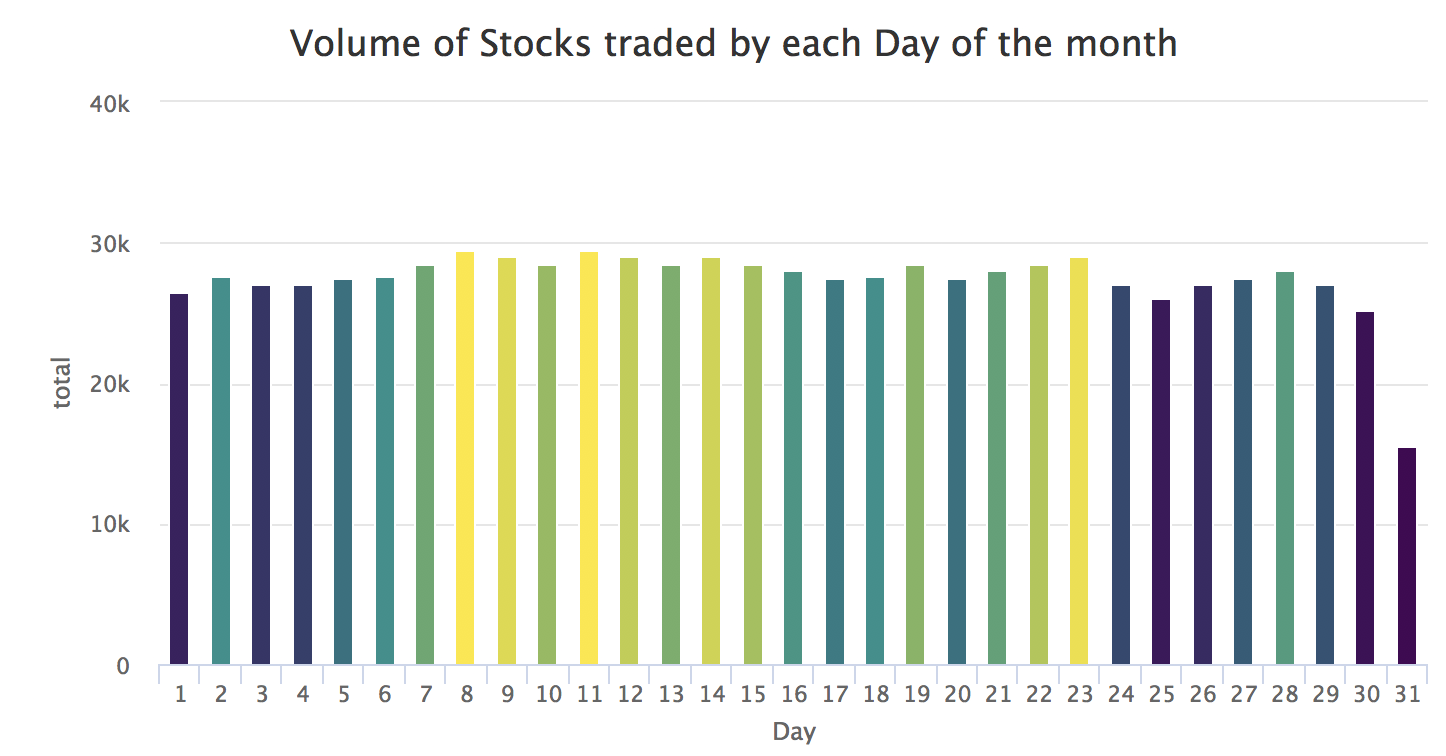


Result:  
- Line graph clearly shows that **August has the highest** volume of stocks i.e. 74937 whereas February has the lowest number of stocks i.e. 65114.  
- **January, February doesn’t** seem to be a good time at Stock Market being the start of the year. - The market seems to be very **uncertain** from **September to December**.

**4) The volume of Stocks traded by each day of the month**

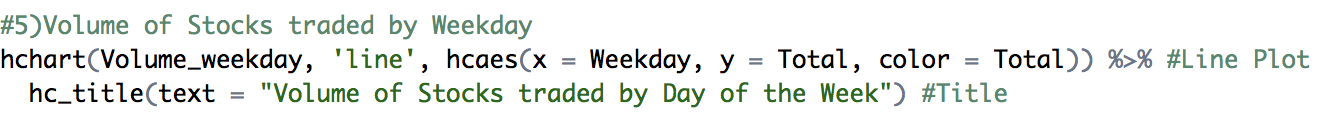


Output:

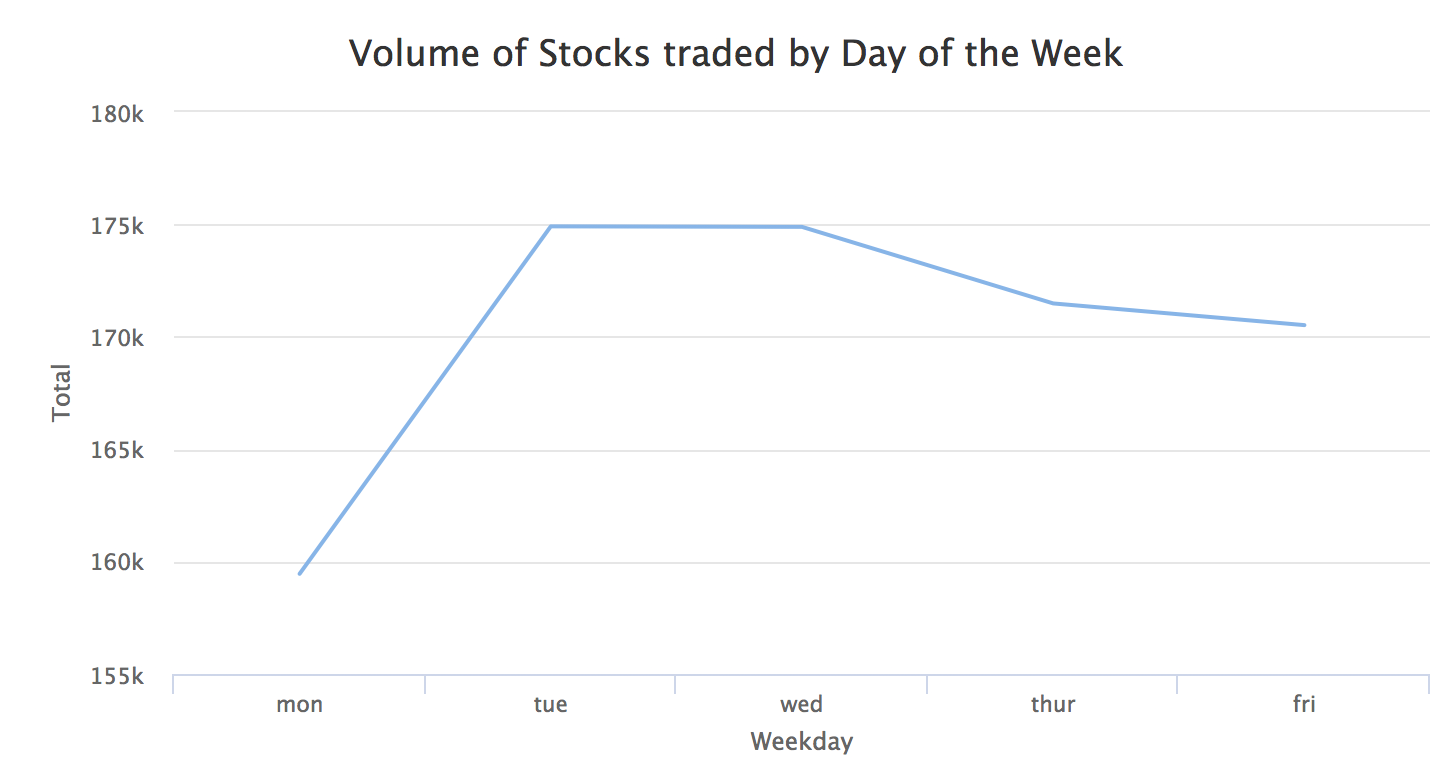


Result:  
- There is the **highest peak in the middle** of each month. There has been a record that on 11th of every month, the highest volume of stocks is traded.  
- Bars with yellow to green color says high volume of stocks has been traded, which is seen in the middle of every month. Whereas dark green to purple and blue says the **low volume** of stocks have been traded, which is seen in **start and end** of every month.  
- People get their salary at the start of every month and the first week usually they spend on clearing their dues which helps them give a clear idea on how much to invest. Thus, middle of the month is the time when people seem to invest in stock market.

**5) The volume of Stocks traded by Weekday**

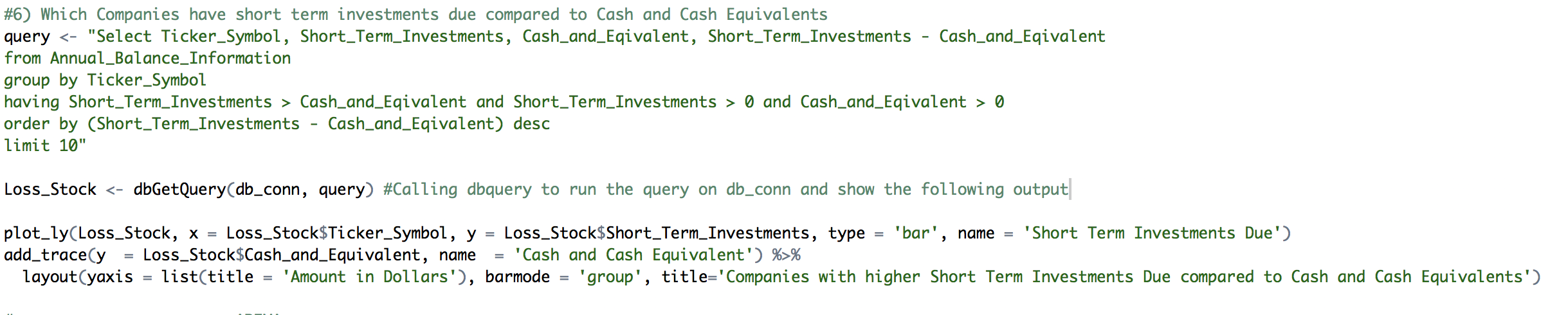


Output:

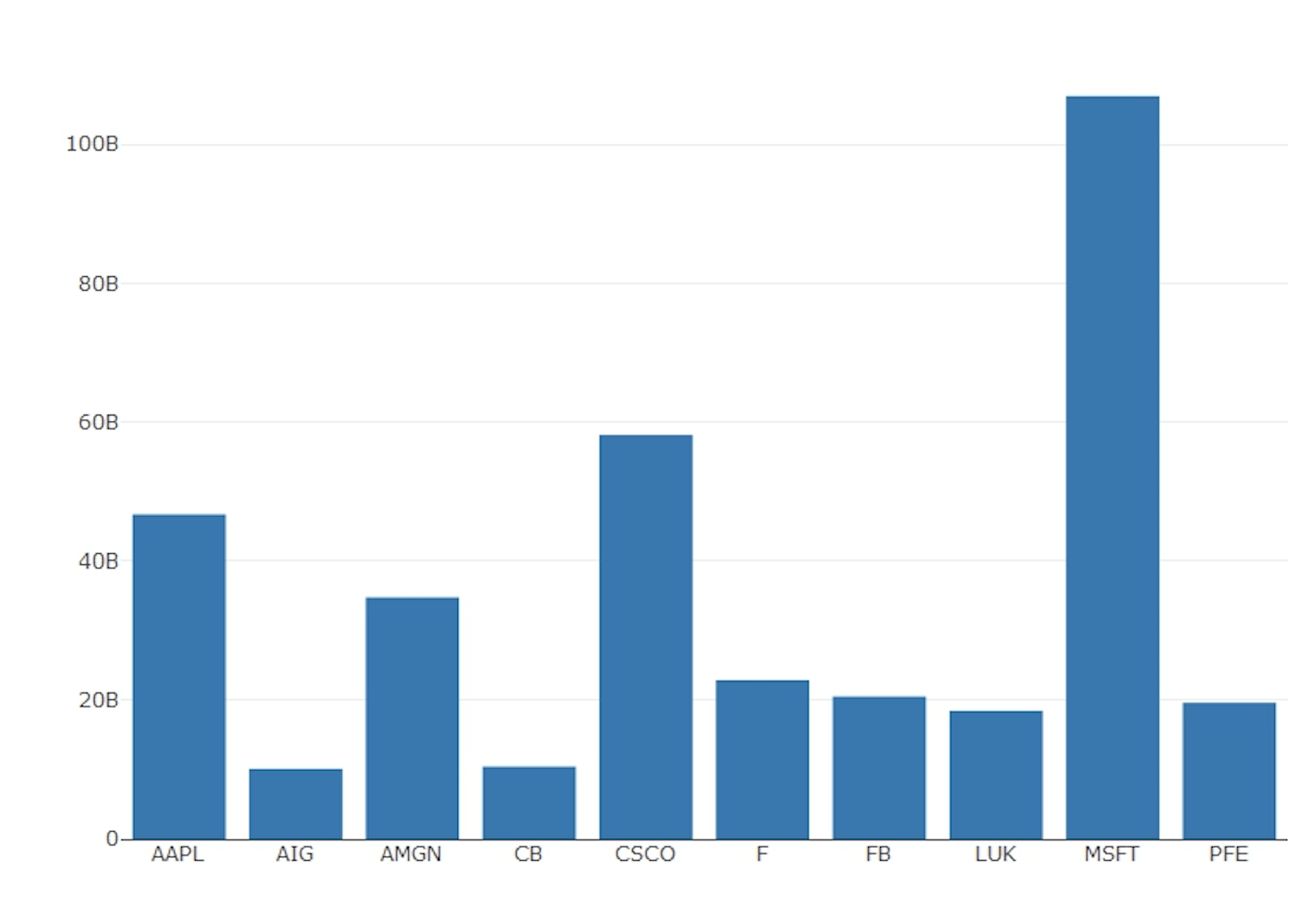


Result:  
- **Lowest volume** of stocks are traded on **Monday**. As Monday is the first day of the week and it is also the busiest day of the week as well as Monday being the first day of the week for the stock market, the market is highly volatile thus making it unpredictable for investments. - **The maximum volume** of stocks is traded on **Tuesdays and Wednesdays**. Tuesdays and Wednesdays are comparatively more predictable. Hence it is the preferred choice for the investors.

**6) Which companies have more Short-Term Investments Due compared to Cash and Cash Equivalent.**



Output:



Result:  
- Short-Term Investments Due are the **debts** incurred by the company which is due within one year. This value helps in determining a company’s performance.  
- Cash and Cash Equivalents are company’s assets that are **cash or can be converted** into cash immediately. Companies with a high amount of cash and cash equivalents are better and show higher liquidity.  
- Thus, having less short-term investments due and higher cash & cash equivalents makes company stable and reliable to buy stocks.  
- The above bar graph shows **top 10 companies** which have more short-term investments due and less cash & cash equivalent, which makes them **less reliable**. This analysis can make investors think twice before buying their stocks.

**Forecasting/ Prediction**

Using the historical data, we are forecasting the volume of stocks traded for future years. We are predicting the data using package timeseries. It tries to forecast and build ARIMA model. The packages for forecasting are as follows:

A picture containing indoor, photo, table

Description generated with high confidence

**ARIMA Model**  
Arima model is the Autoregressive integrated moving average. ARMIA model is applied in cases where data show evidence of non-stationarity, where an initial differencing step (corresponding to the "integrated" part of the model) can be applied one or more times to eliminate the non-stationarity.

ARIMA model is used to forecasting model that utilize historical information to make predictions.

Step 1:

1. Collecting the data:

A screenshot of a cell phone

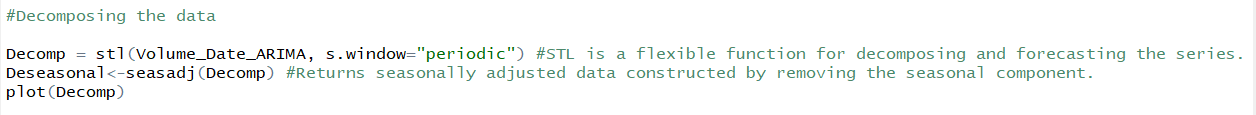
Description generated with high confidence

1. Creating Time seriesA screenshot of a social media post

   Description generated with very high confidence

A screenshot of a social media post

Description generated with very high confidence

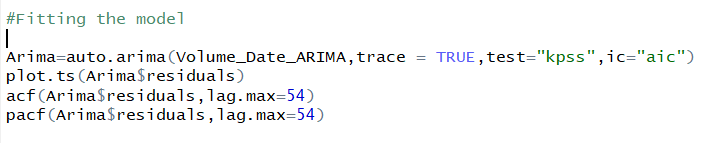
1. Decomposing the data:  
     
   

A drawing of a face

Description generated with high confidence

1. Checking the Stationarity of the model:  
   ARIMA model requires series to be stationary. ADF (Augmented Dickey-Fuller) test is a statistical test for stationarity. The test checks whether the change in Y can be explained by lagged value and a linear trend.  
   A picture containing indoor

   Description generated with high confidence
2. Checking the order of the Model:  
   ACF() plots correlation between a series and its lags. PACF () at K lag is autocorrelation function to plot the correlation between all data points that are exactly k steps apart



A screenshot of a social media post

Description generated with very high confidence

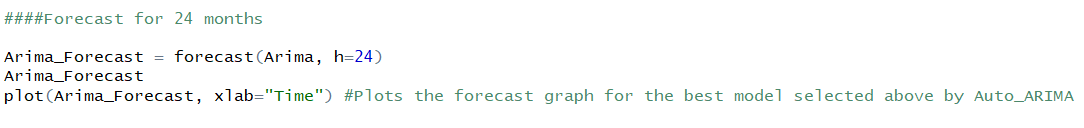
A screenshot of a social media post

Description generated with very high confidence

A screenshot of a social media post

Description generated with very high confidence

ACF and PACF residuals show no significant autocorrelation.

1. Fitting ARIMA model:  
   An auto arima function automatically selects an appropriate lag value using statistical tests and trains a linear regression model.  
     
   

A screenshot of a cell phone

Description generated with very high confidence

The above model predicts that volume of Stocks traded remains same in 2018 and 2019 compared to previous year.

**Future Scope**

1. Expanding the dataset by adding Stock Indexes, International Market Overview, Interest Rates etc. can increase the complexity in this project for which faster and efficient methods to collect, store and retrieve needs to be found.
2. Training the ARIMA model with more data to increase the accuracy. Also implementing other machine learning algorithms

**Learning Outcomes**

1. Scrapping, gathering, cleaning and consolidating techniques on the huge volume of data
2. Creating meaningful and structured schema in its 3NF and then establishing an SQLite connection is skilled.
3. Data retrieval is done using SQL queries and data visualization techniques are performed using plotly, ggthemes, highcharter packages
4. Implemented forecasting technique and learned how to build ARIMA model to predict for future.

**Appendix**

**R code for the project “Stock Market Analysis” is the following:**

#----------- Install packages----------------

install.packages("dplyr")

install.packages("tidyr")

install.packages("highcharter")

install.packages("forecast")

install.packages("timeSeries")

install.packages("tseries")

install.packages("plotly")

install.packages("timeDate")

library(highcharter)

library(tidyr)

library(dplyr)

library(forecast)

library(timeSeries)

library(tseries)

library(plotly)

library(timeDate)

###Importing the Data from Kaggle and S&P 500 compananies(wikipedia)

company\_info <- read.csv('C:/Users/forum/OneDrive/Desktop/ALY 6015/Final Project/Data/Company\_name.csv')

company\_info

daily\_stock\_Prices<- read.csv('C:/Users/forum/OneDrive/Desktop/ALY 6015/Final Project/Data/prices-split-adjusted.csv')

daily\_stock\_Prices

Annual\_balance<- read.csv('C:/Users/forum/OneDrive/Desktop/ALY 6015/Final Project/Data/fundamentals.csv')

Annual\_balance

**################## Cleaning Data : Mohita Sharma ##############################**

#------------------------- Comapny Info------------------------------

# Splitting address column

install.packages("lubridate")

library("lubridate")

install.packages("splitstackshape")

library(splitstackshape)

company\_info<- cSplit(company\_info,"Location",",",fixed= FALSE)

company\_info

#--------- Deleting columns from annual\_balance------------

Annual\_balance$'Period.Ending' <- NULL

Annual\_balance $ 'X' <- NULL

Annual\_balance $'Depreciation' <- NULL

Annual\_balance $'Gross.Profit' <- NULL

Annual\_balance $'Short.Term.Debt...Current.Portion.of.Long.Term.Debt' <- NULL

Annual\_balance $'Profit.Margin' <- NULL

Annual\_balance$'Period.Ending' <- NULL

head(Annual\_balance)

#------- Renaming the columns--------

names(company\_info) <- c("Ticker\_Symbol","Company\_Name","Sector","CIK","City","State")

names(daily\_stock\_Prices) <- c("date","Ticker\_Symbol","Open\_Price","Close\_Price","Lowest\_Rate","Highest\_Rate","Volume")

head(daily\_stock\_Prices)

names(Annual\_balance) <- c("Date","Ticker\_Symbol","Cash\_and\_Eqivalent","Operating\_Margin","Quick\_Ratio","Retained\_Earning","Short\_Term\_Investments","Total\_Current\_Asset",

"Total\_Current\_Liablities","Total\_Revenue","Tresury\_Stock")

head(Annual\_balance)

#--------------Replace missing values------------

Annual\_balance$Date[is.na(Annual\_balance$Date)] <- "1999"

head(Annual\_balance)

daily\_stock\_Prices$date[is.na(daily\_stock\_Prices$date)] <- "1999-01-01"

head(daily\_stock\_Prices)

#----------assignining a common name-----------

company\_info$'Sector'[company\_info$'Sector'=="IT"] <- "Information Technology"

company\_info

#----------------- Dividing date in Day, month, year and weekday----------------

daily\_stock\_info\_table<- daily\_stock\_Prices

daily\_stock\_Prices$date<- as.Date(daily\_stock\_Prices$date)

daily\_stock\_Prices$Day<- factor(day(as.POSIXlt(daily\_stock\_Prices$date)))

daily\_stock\_Prices$Month<- factor(month(as.POSIXlt(daily\_stock\_Prices$date)))

daily\_stock\_Prices$Year<- factor(year(as.POSIXlt(daily\_stock\_Prices$date)))

daily\_stock\_Prices$Weekday<- factor(wday(as.POSIXlt(daily\_stock\_Prices$date)))

head(daily\_stock\_Prices)

#-------------- Storing data by year, month, day and weekday---------------------------

Volume\_year <-daily\_stock\_Prices %>% filter(Year %in% c("2014","2015","2016"))%>% group\_by(Year)%>% summarize(Total =n())

Volume\_month <-daily\_stock\_Prices %>% group\_by(Month)%>% summarize(Total =n())

Month<- c("jan","feb","march","april","may","june","july","august","sept","oct","nov","dec")

Volume\_month$Month <- Month[Volume\_month$Month]

head(Volume\_month)

head(Volume\_year)

Volume\_YearMonth<-daily\_stock\_Prices %>%

group\_by(Year, Month) %>%

summarise(Total = n())

Volume\_YearMonth

Volume\_weekday <- daily\_stock\_Prices %>% group\_by(Weekday)%>% summarize(Total =n())

Weekday <-c("mon","tue","wed","thur","fri")

Volume\_weekday$Weekday <- Weekday[Volume\_weekday$Weekday]

head(Volume\_weekday)

Volume\_day <- daily\_stock\_Prices%>% group\_by(Day)%>% summarize(total =n())

head(Volume\_day)

**##################### Data Storage and Retrieval : Forum Bheda ################**

# Creating database connection

install.packages("RSQLite")

library("RSQLite")

db\_conn<-dbConnect(SQLite(),dbname="Stock\_Market\_Analysis.sqlite")

#Creating tables in the database

#---------For company\_info---------------

dbGetQuery(db\_conn,# DB Connector name

"create table Company\_Info

(

Ticker\_Symbol Text Primary Key,

Company\_Name Text,

Sector Text,

CIK Text,

City Text,

State Text)") #Create Table Script

dbWriteTable(conn=db\_conn,

name="Company\_Info", #Table Name

value=company\_info, #Load data from the newly created data frame

append=TRUE, row.names=FALSE,header=TRUE) #Since the CSV contains headers in the columns

dbListFields(db\_conn,"Company\_Info")#Listing the field of the table

#---------daily\_stock\_Prices---------------

dbGetQuery(db\_conn,# DB Connector name

"create table Daily\_Stock\_Info

(

Daily\_Stock\_Id Integer Primary Key Autoincrement,

date Date,

Ticker\_Symbol Text,

Open\_Price Numeric,

Close\_Price Numeric,

Lowest\_Rate Numeric,

Highest\_Rate Numeric,

Volume Real)") #Create Table Script

dbWriteTable(conn=db\_conn,

name="Daily\_Stock\_Info", #Table Name

value=daily\_stock\_Prices, #Load data from the newly created data frame

append=TRUE, row.names=FALSE,header=TRUE) #Since the CSV contains headers in the columns

dbListFields(db\_conn,"Daily\_Stock\_Info")#Listing the field of the table

#---------Annual\_balance---------------

dbGetQuery(db\_conn,# DB Connector name

"create table Annual\_Balance\_Information

(

Annual\_Stock\_Id Integer Primary Key Autoincrement,

Date Integer,

Ticker\_Symbol Text,

Cash\_and\_Eqivalent Numeric,

Operating\_Margin Numeric,

Quick\_Ratio Numeric,

Retained\_Earning Numeric,

Short\_Term\_Investments Numeric,

Total\_Current\_Asset Numeric,

Total\_Current\_Liablities Numeric,

Total\_Revenue Numeric,

Tresury\_Stock Numeric)") #Create Table Script

dbWriteTable(conn=db\_conn,

name="Annual\_Balance\_Information", #Table Name

value=Annual\_balance, #Load data from the newly created data frame

append=TRUE, row.names=FALSE,header=TRUE) #Since the CSV contains headers in the columns

dbListFields(db\_conn,"Annual\_Balance\_Information")#Listing the field of the table

#-----------------Calculation----------------------------------------------------------------------------------

#Calculating current ratio of all stocks

Annual\_Info\_Current\_Ratio<-Annual\_balance%>%

select(Date,Ticker\_Symbol,Total\_Current\_Liablities,Total\_Current\_Asset)

Annual\_Info\_Current\_Ratio$Current\_Ratio<-Annual\_Info\_Current\_Ratio$Total\_Current\_Asset/Annual\_Info\_Current\_Ratio$Total\_Current\_Liablities

Annual\_Profit<-filter(Annual\_Info\_Current\_Ratio,Current\_Ratio>=1) #Contains all companies having higher assets than liabilities

Annual\_Loss<-filter(Annual\_Info\_Current\_Ratio,Current\_Ratio<1) #Contains all companies having higher liabilities than assets

#Calculating the cash ratio for all the stocks

Annual\_Info\_Cash\_Ratio<-Annual\_balance$Cash\_and\_Eqivalent/Annual\_balance$Total\_Current\_Liablities

#-----------Display the Companies having a current ratio greater than one-----------

query<-"select a.Ticker\_Symbol, c.Company\_Name,(a.Total\_Current\_Asset/a.Total\_Current\_Liablities) as Current\_Ratio

from Annual\_Balance\_Information a

join Company\_Info c on a.Ticker\_Symbol=c.Ticker\_Symbol

group by a.Ticker\_Symbol, c.Company\_Name

having (a.Total\_Current\_Asset/a.Total\_Current\_Liablities)>1

order by (a.Total\_Current\_Asset/a.Total\_Current\_Liablities) desc

limit 10"

dbGetQuery(db\_conn,query)

#-------------State having highest number of headquarters--------------

queryhdq<-"select State,count(Ticker\_Symbol) as No\_of\_Headquarters

from Company\_Info

group by State

order by count(Ticker\_Symbol) desc

limit 5"

dbGetQuery(db\_conn,queryhdq)

#------------------------- Display all the GICS Sector along with their Total revenue-----------

queryGICS<-"select c.Sector, sum(a.Total\_Revenue) as Total\_Revenue

from Annual\_Balance\_Information a

join Company\_Info c on a.Ticker\_Symbol=c.Ticker\_Symbol

group by c.Sector

order by sum(a.Total\_Revenue) desc"

dbGetQuery(db\_conn,queryGICS)

#-------Display the Retained earnings for Industries----------------

queryRet<-"select c.Sector, sum(a.Retained\_Earning) as Total\_Retained\_Earnings

from Annual\_Balance\_Information a

join Company\_Info c on a.Ticker\_Symbol=c.Ticker\_Symbol

group by c.Sector

order by sum(a.Retained\_Earning) desc"

dbGetQuery(db\_conn,queryRet)

**############# Data Analysis: Labdhi Ghelani ############################**

#1) Plot a graph of how much a company make's on each dollar of sales [Operation Margin]

Opt\_Margin <- Annual\_balance %>%

select(Ticker\_Symbol,Operating\_Margin) %>%

arrange(desc(Operating\_Margin)) %>%

head(n=10) %>%

distinct()

hchart(Opt\_Margin, 'column', hcaes(x = Ticker\_Symbol, y = Operating\_Margin, color = Operating\_Margin)) %>%

hc\_add\_theme(hc\_theme\_flat()) %>%

hc\_title(text = "Top 10 stocks with highest Operation Margin")

#2)Display the number of headquarters according to State.

State\_HQ <- company\_info %>%

select(State) %>%

group\_by(State) %>%

summarise(No\_of\_HQ = n()) %>%

arrange(desc(No\_of\_HQ))

hchart(State\_HQ[1:2], 'pie', hcaes (x = State, y = No\_of\_HQ, color = No\_of\_HQ)) %>%

hc\_add\_theme(hc\_theme\_google()) %>%

hc\_title(text = "Number of headquarters according to State")

#3)Volume of Stocks traded by Month

hchart(Volume\_month, 'line', hcaes(x = Month, y = Total, color = Total)) %>%

hc\_title(text = "Volume of Stocks traded by month of the year")

#4)Volume of Stocks traded by each day of the month

hchart(Volume\_day, 'column', hcaes(x = Day, y = total, color = total)) %>%

hc\_title(text = "Volume of Stocks traded by each Day of the month")

#5)Volume of Stocks traded by Weekday

hchart(Volume\_weekday, 'line', hcaes(x = Weekday, y = Total, color = Total)) %>%

hc\_title(text = "Volume of Stocks traded by Day of the Week")

#6) Which Companies have short term investments due compared to Cash and Cash Equivalents

query <- "Select Ticker\_Symbol, Short\_Term\_Investments, Cash\_and\_Eqivalent, Short\_Term\_Investments - Cash\_and\_Eqivalent

from Annual\_Balance\_Information

group by Ticker\_Symbol

having Short\_Term\_Investments > Cash\_and\_Eqivalent and Short\_Term\_Investments > 0 and Cash\_and\_Eqivalent > 0

order by (Short\_Term\_Investments - Cash\_and\_Eqivalent) desc

limit 10"

Loss\_Stock <- dbGetQuery(db\_conn, query)

plot\_ly(Loss\_Stock, x = Loss\_Stock$Ticker\_Symbol, y = Loss\_Stock$Short\_Term\_Investments, type = 'bar', name = 'Short Term Investments Due')

add\_trace(y = Loss\_Stock$Cash\_and\_Equivalent, name = 'Cash and Cash Equivalent') %>%

layout(yaxis = list(title = 'Amount in Dollars'), barmode = 'group', title='Companies with higher Short Term Investments Due compared to Cash and Cash Equivalents')

**################### ARIMA : Mohita Sharma #######################**

Volume\_data\_arima<- daily\_stock\_Prices %>% group\_by(date) %>% summarize(Total = n())

#Create Time series

Volume\_Date\_ARIMA = ts(na.omit(Volume\_data\_arima$Total), start=c(2015,1), end=c(2018,12),frequency=24) #Preparing time series

plot(Volume\_Date\_ARIMA)

#Decomposing the data

Decomp = stl(Volume\_Date\_ARIMA, s.window="periodic") #STL is a flexible function for decomposing and forecasting the series.

Deseasonal<-seasadj(Decomp) #Returns seasonally adjusted data constructed by removing the seasonal component.

plot(Decomp)

# To check if mean is stationary

adf.test(Volume\_Date\_ARIMA,alternative="stationary")

# p-value is greater than 0.05 hence we can say that mean is not constant

#Fitting the model

Arima=auto.arima(Volume\_Date\_ARIMA,trace = TRUE,test="kpss",ic="aic")

plot.ts(Arima$residuals)

acf(Arima$residuals,lag.max=54)

pacf(Arima$residuals,lag.max=54)

####Forecast for 24 months

Arima\_Forecast = forecast(Arima, h=24)

Arima\_Forecast

plot(Arima\_Forecast, xlab="Time") #Plots the forecast graph for the best model selected above by Auto\_ARIMA

**References**

* RStudio. (n.d.). Retrieved from <https://db.rstudio.com/databases/sqlite/>
* O'Connell, B. (2018, October 03). How Do Stocks and the Stock Market Work? Retrieved from <https://www.thestreet.com/investing/stocks/how-do-stocks-work-14726850>
* Highcharts for R users. (2018, August 17). Retrieved from <https://www.highcharts.com/blog/data-science/highcharts-for-r-users/>
* Kunst, J. (n.d.). H|1i|0g|3h|2c|1h|2a|1r|3t|2e|1r|2{rpackage}. Retrieved from <http://jkunst.com/highcharter/>
* Data Visualization with Highcharter in R. (n.d.). Retrieved from <https://www.datacamp.com/community/tutorials/data-visualization-highcharter-r>
* Contributors, D. C. (n.d.). Retrieved from <https://datacarpentry.org/R-ecology-lesson/05-r-and-databases.html>