TERM PROJECT

COLLECTING, STORING AND RERIEVING DATA

BY –

Neha Firodiya

INTRODUCTION

For my term project I really wanted to work with stock market data. I previously decided to use NASDAQ Data-On-Demand as my data source but shifted to Yahoo Finance given the simplicity of the available data. I scraped the historical stock price data from multiple pages to get the stock prices data of the company of my choice, instead of directly getting it in csv format. After cleaning the data and bringing it into a proper format, I stored it in MongoDB. As end result, I performed trend analysis on the data, made a time –series model and developed a generalized polynomial function for the time series model.

CHALLENGES FACED

The biggest challenge was scraping the data. There are several methods of scraping a HTML table from a webpage and I tried two approaches. I first tried using rvest package but ran into trouble as there was a problem reading the html. Tried using read_html instead of just html for reading the url, but I couldn't get the table using html_table(). There was an issue with the XPath I was providing for sure. Switched to using RCurl and XML packages for scraping. Finally, when using xpathApply() to get the leaf elements, I was trying to simply copy and paste the XPath from the source but that wasn't working for me anywhere. I realized it's not necessary to copy and paste the XPath, I can simply search for a matching tag value. The XPath was actually not giving any results as there are symbols like "*" in the XPath you copy which are not recognized. I learned, you should always try the simple route first instead of jumping for the most efficient of preferred one and then you can build your way to the efficient methods once you know how it's done.

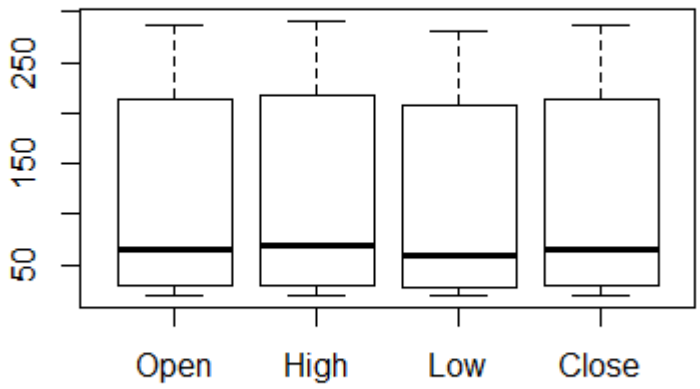
R CODE

Following is my R code for getting the data from Yahoo Finance. It can be easily changed for getting any other company's historical stock prices. A for loop was made so that results from all the pages for the company can be scraped.

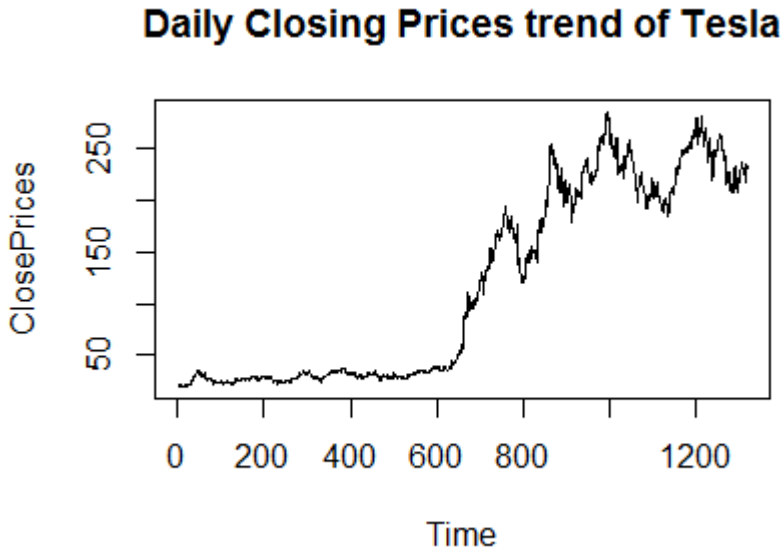
```
library(mongolite)
library(RCurl)
library(XML)
library(forecast)
library(fpp)

getwd()


### SCRAPING DATA ###

## GETTING THE DATA FROM
"http://finance.yahoo.com/q/hp?s=TSLA&a=5&b=29&c=2010&d=11&e=18&f=2015&g=d&z=
66&y=0"
##scraping data from 20 pages of yahoo finance, to get stock prices of Tesla
Motors, Inc.
```

```
url_number_list <- seq(0, length=20, by=66)  ##as the last part of url "=0"
changes by +66 for every new page
data_list <- list()  ##an empty list to store the scraped data

for(i in 1:20) {
num <- url_number_list[i]
url <-
paste("http://finance.yahoo.com/q/hp?s=TSLA&a=05&b=29&c=2010&d=11&e=18&f=2015
&g=d&z=66&y=",toString(num),sep="")
webpage <- getURL(url)
tc <- textConnection(webpage)
webpage <- readLines(tc)
close(tc)
pagetree <- htmlTreeParse(webpage, useInternalNodes = TRUE)
data_list[[i]] <- xpathApply(pagetree, "//td[@class='yfnc_tabledata1']",
xmlValue) #append to the list data_list
}

head(data_list)  #this is a list of lists


## unlist data_list
data_list <- unlist(unlist(data_list, recursive=FALSE))


## get the index of rows which have "\n  "
grep("\n" , data_list, ignore.case = FALSE)

##remove those rows
data_list<- data_list[c(-463,-926,-1389,-1852,-2315,-2778,-3241,-3704,-4167,-
4630,-5093,-5556,-6019,-6482,-6945,-7408,-7871,-8334,-8797,-9260)]

##create a dataframe
content <- as.data.frame(matrix(data_list, ncol = 7, byrow = TRUE))
colnames(content) <- c("Date", "Open","High", "Low", "Close", "Volume", " Adj
Close")

##converting columns into proper class
content$Date <- strptime(content$Date, "%b %d, %Y")
content$Date <- format(content$Date, "%Y-%m-%d")
content$Date <- as.Date(content$Date, "%Y-%m-%d")

#Converting columns 2 to 5 Factors to Numeric as we need them for analysis
for(i in 2:5)
{
  content[,i]<- as.numeric(levels(content[,i]))[content[,i]]
}

#checking class of all columns
sapply(content, class)
```

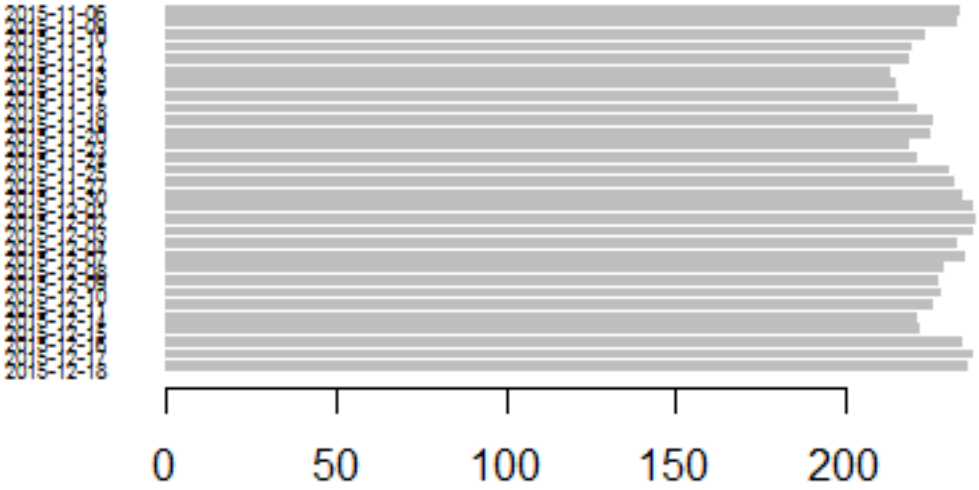
```
### ANALYSIS OF THE DATA OBTAINED ###

#Plot box plots to check 25th,50th and 75th percentile
boxplot(content[,2:5])
```



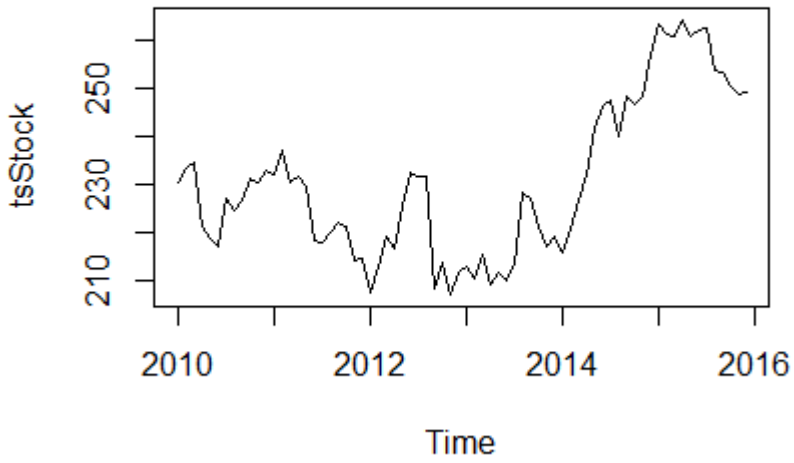

```
# trend
plot(rev(content$Close),type='l',xlab='Time',ylab='ClosePrices',main='Daily
Closing Prices trend of Tesla')
```



```
# Bar plot Highest price analysis for 30 days
barplot(content$High[1:30], names.arg=content$Date[1:30], horiz=TRUE, las=1,
cex.names=0.5, border=NA)
```

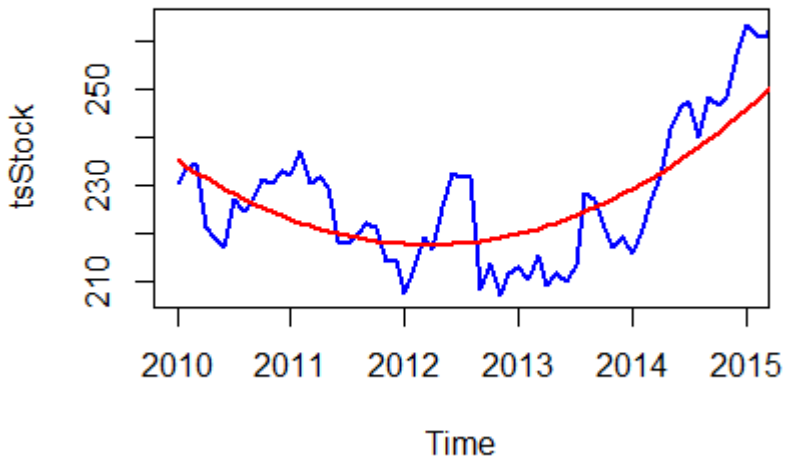



```
# time series model
tsStock <- ts(content$Close, start=c(2010,1), end=c(2015,12), frequency=12)
tsStock #check data
plot(tsStock)  #plot data
```



Above graph shows the variation in the stock prices since 2010. There has been a lot of fluctuation given it's a start-up. But, as we can see the there was a high in the stock prices in the year 2015.

```
# Creating a polynomial model
t1 <- seq(2010, 2015, length=length(tsStock))
t12 <- t1^7
polystock <- lm(tsStock ~ t1 + t12)
tsStock_trend <- ts(polystock$fit, start=c(2010, 1), frequency=12)
plot(tsStock, lw=2, col="blue", xlim=c(2010, 2015))
lines(tsStock_trend, lw=2, col="red")
```



The red line on the graph shows how the polynomial function behaves. The blue plot is the time series plot.

```
### STORING THE DATA IN MONGODB ###

mongoData <- mongo("content")              # creatign a connection with MongoDB
mongoData$insert(content)                  # inserting data
mongoData$export(file("content.txt"))      # exporting into a text file
```


```
{ "_id" : { "$oid" : "5674c5729546012038001367" }, "Date" : "2015-12-10", "Open" : 412, "High" : 402, "Low" : 436, "Close" : 431, "Volume" : 379, " Adj Close" : 431 }
{ "_id" : { "$oid" : "5674c5729546012038001368" }, "Date" : "2015-12-09", "Open" : 419, "High" : 396, "Low" : 421, "Close" : 415, "Volume" : 520, " Adj Close" : 415 }
{ "_id" : { "$oid" : "5674c5729546012038001369" }, "Date" : "2015-12-08", "Open" : 425, "High" : 406, "Low" : 439, "Close" : 428, "Volume" : 442, " Adj Close" : 428 }
{ "_id" : { "$oid" : "5674c572954601203800136a" }, "Date" : "2015-12-07", "Open" : 426, "High" : 470, "Low" : 449, "Close" : 480, "Volume" : 532, " Adj Close" : 480 }
{ "_id" : { "$oid" : "5674c572954601203800136b" }, "Date" : "2015-12-04", "Open" : 478, "High" : 456, "Low" : 458, "Close" : 471, "Volume" : 426, " Adj Close" : 471 }
{ "_id" : { "$oid" : "5674c572954601203800136c" }, "Date" : "2015-12-03", "Open" : 497, "High" : 477, "Low" : 505, "Close" : 493, "Volume" : 478, " Adj Close" : 493 }
{ "_id" : { "$oid" : "5674c572954601203800136d" }, "Date" : "2015-12-02", "Open" : 508, "High" : 487, "Low" : 516, "Close" : 489, "Volume" : 481, " Adj Close" : 489 }
{ "_id" : { "$oid" : "5674c572954601203800136e" }, "Date" : "2015-12-01", "Open" : 474, "High" : 485, "Low" : 513, "Close" : 510, "Volume" : 605, " Adj Close" : 510 }
{ "_id" : { "$oid" : "5674c572954601203800136f" }, "Date" : "2015-11-30", "Open" : 477, "High" : 458, "Low" : 469, "Close" : 469, "Volume" : 429, " Adj Close" : 469 }
{ "_id" : { "$oid" : "5674c5729546012038001370" }, "Date" : "2015-11-27", "Open" : 474, "High" : 450, "Low" : 453, "Close" : 483, "Volume" : 251, " Adj Close" : 483 }
{ "_id" : { "$oid" : "5674c5729546012038001371" }, "Date" : "2015-11-25", "Open" : 389, "High" : 439, "Low" : 418, "Close" : 444, "Volume" : 639, " Adj Close" : 444 }
{ "_id" : { "$oid" : "5674c5729546012038001372" }, "Date" : "2015-11-24", "Open" : 340, "High" : 362, "Low" : 360, "Close" : 356, "Volume" : 417, " Adj Close" : 356 }
{ "_id" : { "$oid" : "5674c5729546012038001373" }, "Date" : "2015-11-23", "Open" : 353, "High" : 342, "Low" : 358, "Close" : 354, "Volume" : 422, " Adj Close" : 354 }
{ "_id" : { "$oid" : "5674c5729546012038001374" }, "Date" : "2015-11-20", "Open" : 406, "High" : 386, "Low" : 352, "Close" : 390, "Volume" : 694, " Adj Close" : 390 }
{ "_id" : { "$oid" : "5674c5729546012038001375" }, "Date" : "2015-11-19", "Open" : 383, "High" : 393, "Low" : 416, "Close" : 403, "Volume" : 418, " Adj Close" : 403 }
{ "_id" : { "$oid" : "5674c5729546012038001376" }, "Date" : "2015-11-18", "Open" : 334, "High" : 364, "Low" : 349, "Close" : 399, "Volume" : 459, " Adj Close" : 399 }
{ "_id" : { "$oid" : "5674c5729546012038001377" }, "Date" : "2015-11-17", "Open" : 339, "High" : 321, "Low" : 344, "Close" : 334, "Volume" : 387, " Adj Close" : 334 }
{ "_id" : { "$oid" : "5674c5729546012038001378" }, "Date" : "2015-11-16", "Open" : 266, "High" : 314, "Low" : 288, "Close" : 335, "Volume" : 476, " Adj Close" : 335 }
{ "_id" : { "$oid" : "5674c5729546012038001379" }, "Date" : "2015-11-13", "Open" : 328, "High" : 299, "Low" : 297, "Close" : 277, "Volume" : 567, " Adj Close" : 277 }
{ "_id" : { "$oid" : "5674c572954601203800137a" }, "Date" : "2015-11-12", "Open" : 357, "High" : 341, "Low" : 350, "Close" : 330, "Volume" : 475, " Adj Close" : 330 }
{ "_id" : { "$oid" : "5674c572954601203800137b" }, "Date" : "2015-11-11", "Open" : 356, "High" : 344, "Low" : 354, "Close" : 366, "Volume" : 557, " Adj Close" : 366 }
{ "_id" : { "$oid" : "5674c572954601203800137c" }, "Date" : "2015-11-10", "Open" : 405, "High" : 378, "Low" : 367, "Close" : 342, "Volume" : 717, " Adj Close" : 342 }
{ "_id" : { "$oid" : "5674c572954601203800137d" }, "Date" : "2015-11-09", "Open" : 480, "High" : 455, "Low" : 440, "Close" : 421, "Volume" : 624, " Adj Close" : 421 }
{ "_id" : { "$oid" : "5674c572954601203800137e" }, "Date" : "2015-11-06", "Open" : 472, "High" : 457, "Low" : 474, "Close" : 490, "Volume" : 413, " Adj Close" : 490 }
{ "_id" : { "$oid" : "5674c572954601203800137f" }, "Date" : "2015-11-05", "Open" : 471, "High" : 460, "Low" : 472, "Close" : 487, "Volume" : 708, " Adj Close" : 487 }
{ "_id" : { "$oid" : "5674c5729546012038001380" }, "Date" : "2015-11-04", "Open" : 423, "High" : 453, "Low" : 444, "Close" : 484, "Volume" : 316, " Adj Close" : 484 }
{ "_id" : { "$oid" : "5674c5729546012038001381" }, "Date" : "2015-11-03", "Open" : 332, "High" : 309, "Low" : 310, "Close" : 292, "Volume" : 1134, " Adj Close" : 292 }
{ "_id" : { "$oid" : "5674c5729546012038001382" }, "Date" : "2015-11-02", "Open" : 285, "High" : 319, "Low" : 304, "Close" : 333, "Volume" : 634, " Adj Close" : 333 }
{ "_id" : { "$oid" : "5674c5729546012038001383" }, "Date" : "2015-10-30", "Open" : 310, "High" : 291, "Low" : 276, "Close" : 275, "Volume" : 700, " Adj Close" : 275 }
{ "_id" : { "$oid" : "5674c5729546012038001384" }, "Date" : "2015-10-29", "Open" : 320, "High" : 303, "Low" : 338, "Close" : 324, "Volume" : 237, " Adj Close" : 324 }
{ "_id" : { "$oid" : "5674c5729546012038001385" }, "Date" : "2015-10-28", "Open" : 316, "High" : 301, "Low" : 312, "Close" : 331, "Volume" : 446, " Adj Close" : 331 }
{ "_id" : { "$oid" : "5674c5729546012038001386" }, "Date" : "2015-10-27", "Open" : 336, "High" : 329, "Low" : 306, "Close" : 316, "Volume" : 582, " Adj Close" : 316 }
{ "_id" : { "$oid" : "5674c5729546012038001387" }, "Date" : "2015-10-26", "Open" : 317, "High" : 320, "Low" : 335, "Close" : 337, "Volume" : 563, " Adj Close" : 337 }
{ "_id" : { "$oid" : "5674c5729546012038001388" }, "Date" : "2015-10-23", "Open" : 337, "High" : 316, "Low" : 307, "Close" : 295, "Volume" : 680, " Adj Close" : 295 }
{ "_id" : { "$oid" : "5674c5729546012038001389" }, "Date" : "2015-10-22", "Open" : 318, "High" : 318, "Low" : 320, "Close" : 326, "Volume" : 460, " Adj Close" : 326 }
{ "_id" : { "$oid" : "5674c572954601203800138a" }, "Date" : "2015-10-21", "Open" : 321, "High" : 313, "Low" : 316, "Close" : 314, "Volume" : 673, " Adj Close" : 314 }
{ "_id" : { "$oid" : "5674c572954601203800138b" }, "Date" : "2015-10-20", "Open" : 427, "High" : 404, "Low" : 259, "Close" : 332, "Volume" : 341, " Adj Close" : 332 }
{ "_id" : { "$oid" : "5674c572954601203800138c" }, "Date" : "2015-10-19", "Open" : 417, "High" : 443, "Low" : 442, "Close" : 436, "Volume" : 420, " Adj Close" : 436 }
{ "_id" : { "$oid" : "5674c572954601203800138d" }, "Date" : "2015-10-16", "Open" : 402, "High" : 437, "Low" : 434, "Close" : 429, "Volume" : 687, " Adj Close" : 429 }
{ "_id" : { "$oid" : "5674c572954601203800138e" }, "Date" : "2015-10-15", "Open" : 346, "High" : 367, "Low" : 355, "Close" : 401, "Volume" : 465, " Adj Close" : 401 }
{ "_id" : { "$oid" : "5674c572954601203800138f" }, "Date" : "2015-10-14", "Open" : 385, "High" : 360, "Low" : 362, "Close" : 344, "Volume" : 528, " Adj Close" : 344 }
{ "_id" : { "$oid" : "5674c5729546012038001390" }, "Date" : "2015-10-13", "Open" : 330, "High" : 371, "Low" : 340, "Close" : 367, "Volume" : 793, " Adj Close" : 367 }
{ "_id" : { "$oid" : "5674c5729546012038001391" }, "Date" : "2015-10-12", "Open" : 401, "High" : 373, "Low" : 361, "Close" : 340, "Volume" : 622, " Adj Close" : 340 }
{ "_id" : { "$oid" : "5674c5729546012038001392" }, "Date" : "2015-10-09", "Open" : 386, "High" : 382, "Low" : 382, "Close" : 396, "Volume" : 918, " Adj Close" : 396 }
{ "_id" : { "$oid" : "5674c5729546012038001393" }, "Date" : "2015-10-08", "Open" : 467, "High" : 438, "Low" : 426, "Close" : 428, "Volume" : 916, " Adj Close" : 428 }
{ "_id" : { "$oid" : "5674c5729546012038001394" }, "Date" : "2015-10-07", "Open" : 506, "High" : 481, "Low" : 470, "Close" : 488, "Volume" : 960, " Adj Close" : 488 }
{ "_id" : { "$oid" : "5674c5729546012038001395" }, "Date" : "2015-10-06", "Open" : 549, "High" : 537, "Low" : 535, "Close" : 564, "Volume" : 798, " Adj Close" : 564 }
```


This is the .txt file of the exported database in MongoDB

END OF PROJECT

FUTURE SCOPE

- The code can be modified to get data every day and keep updating the data in the MongoDb database.
- With knowledge of prediction models, a predictive analysis of stock prices can be done using the data.
- Financial Risk Analysis can also be done using the data, given we have a good statistical knowledge of how to deal with it and how to build the models.

LEARNINGS

- Several concepts in R like scraping data from webpages, how to read a HTML document, XML parsing, data shaping, regex were revisited during the course of this project.
- I was introduced to two packages - forecast and fpp which are used for analysis.
- Also, I was introduced to Quantmod package which is used specifically for quantitative financial modelling.
- R has a very wide application from data collection to analysis and the simplicity and power of using it makes it a desirable choice.
- Learned the basic approaches to programming in the process of learning R.