513 Final

Kejia Huang

December 13, 2015

2.1 Question-Stock Data

1 Pick up 10 stock tickers. Use API to download daily date for 2013 and 2014

```
library(RMySQL)
library(quantmod)

# Connect to database
conn = dbConnect(RMySQL::MySQL(),host = "fscwinsrv04.fsc.stevens.edu" ,
                 user = "a513_Kejia",password = "Hh626491",
                 dbname = "a513_kejia",port = 3306)

# Downloads ten stocks data
 MSFT <- data.frame(getSymbols ( "MSFT", from = "2013-01-01",
    to = "2014-12-31", auto.assign = FALSE  ))
 NKE <- data.frame(getSymbols ( "NKE", from = "2013-01-01",
    to = "2014-12-31", auto.assign = FALSE  ))
 PFE <- data.frame(getSymbols ( "PFE", from = "2013-01-01",
    to = "2014-12-31", auto.assign = FALSE  ))
 PG <- data.frame(getSymbols ( "PG", from = "2013-01-01",
    to = "2014-12-31", auto.assign = FALSE  ))
 TRV <- data.frame(getSymbols ( "TRV", from = "2013-01-01",
    to = "2014-12-31", auto.assign = FALSE  ))
 UTX <- data.frame(getSymbols ( "UTX", from = "2013-01-01",
    to = "2014-12-31", auto.assign = FALSE  ))
 UNH <- data.frame(getSymbols ( "UNH", from = "2013-01-01",
    to = "2014-12-31", auto.assign = FALSE  ))
 VZ <- data.frame(getSymbols ( "VZ", from = "2013-01-01",
    to = "2014-12-31", auto.assign = FALSE  ))
 V <- data.frame(getSymbols ( "V", from = "2013-01-01",
    to = "2014-12-31", auto.assign = FALSE  ))
 WMT <- data.frame(getSymbols ( "WMT", from = "2013-01-01",
    to = "2014-12-31", auto.assign = FALSE  ))
```

```
 #  Save 10 stock data into your own database as 10 tables
dbWriteTable(conn, "MSFT", MSFT)
dbWriteTable(conn, "NKE", NKE)
dbWriteTable(conn, "PFE", PFE)
dbWriteTable(conn, "PG", PG)
dbWriteTable(conn, "TRV", TRV)
dbWriteTable(conn, "UTX", UTX)
dbWriteTable(conn, "UNH", UNH)
dbWriteTable(conn, "VZ", VZ)
dbWriteTable(conn, "V", V)
dbWriteTable(conn, "WMT", WMT)

# Show table
x <- as.vector(dbGetQuery(conn, "SHOW tables"))
#    Tables_in_a513_kejia
#1                  msft
#2                   nke
#3                   pfe
#4                    pg
#5                   trv
#6                   unh
#7                   utx
#8                     v
#9                    vz
#10                  wmt

# Disconnect from database
dbDisconnect(conn)
```

2 By using SQL query, you need to combine close price from all 10 stock tables based on date. The output should not have NULL value.

Firstly, I change the column name in MySQL for convenient

```
use a513_kejia;
SELECT msft.data_date, msft_price , nke_price , pfe_price ,
pg_price, trv_price, unh_price, utx_price, v_price, vz_price, wmt_price
FROM msft
left outer join  nke on msft.data_date = nke.data_date
left outer join pfe on msft.data_date = pfe.data_date
left outer join pg on msft.data_date = pg.data_date
left outer join trv on msft.data_date = trv.data_date
left outer join unh on msft.data_date = unh.data_date
```

```
left outer join utx on msft.data_date = utx.data_date
left outer join v on msft.data_date = v.data_date
left outer join vz on msft.data_date = vz.data_date
left outer join wmt on msft.data_date = wmt.data_date;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

data_date	msft_price	nke_price	pfe_price	pg_price	trv_price	unh_pri	utx_price	v_price	vz_price	wmt_price
2013-01-03	27.25	52.369...	25.85	68.94...	73.419998	51.9...	84.30...	155.5	44.060001	68.800003
2013-01-02	27.620...	51.84	25.91	69.38...	72.860001	54.5...	84	155.380...	44.27	69.239998
2013-01-04	26.74	52.880...	25.959999	69.08...	74.059998	52.09	84.98...	156.770...	44.299999	69.059998
2013-01-07	26.690...	52.959...	25.98	68.62...	73.059998	52.09	84.57	157.889...	44.689999	68.400002
2013-01-08	26.549...	52.400...	26.02	68.51...	73.190002	51.4...	83.55...	159.360...	43.099998	68.589996
2013-01-09	26.700...	52.450...	26.469999	68.87...	73.940002	52.3...	84.55...	161.789...	43	68.57
2013-01-11	26.83	53.099...	26.52	69.22...	74.849998	52.82	85.18	161.160...	43.299999	68.629997
2013-01-18	27.25	53.290...	26.540001	69.94...	76.309998	54.5...	86.94...	158.270...	42.540001	69.199997
2013-01-16	27.040...	53.73	26.610001	69.33...	75.550003	53.66	85.57	160.199...	41.509998	69.209999
2013-01-15	27.209...	53.639...	26.620001	69.87...	75.559998	53.6...	85.95...	160.449...	41.970001	68.980003
2013-01-23	27.610...	53.09	26.65	70.69...	77.639999	55.91	88.07	159.050...	42.790001	69.489998
2013-01-22	27.15	53.48	26.68	69.94...	77.949997	56.02	87.47...	159.050...	42.939999	69.580002

3 By either r-mysql API or csv transferring, read the output from previous question into R as data frame. It should include 11 columns (1 date column and 10 stock columns) or 10 columns (if you put date as row name). The column name should be date and 10 tickers. Those 10 close price columns should only contain numeric values

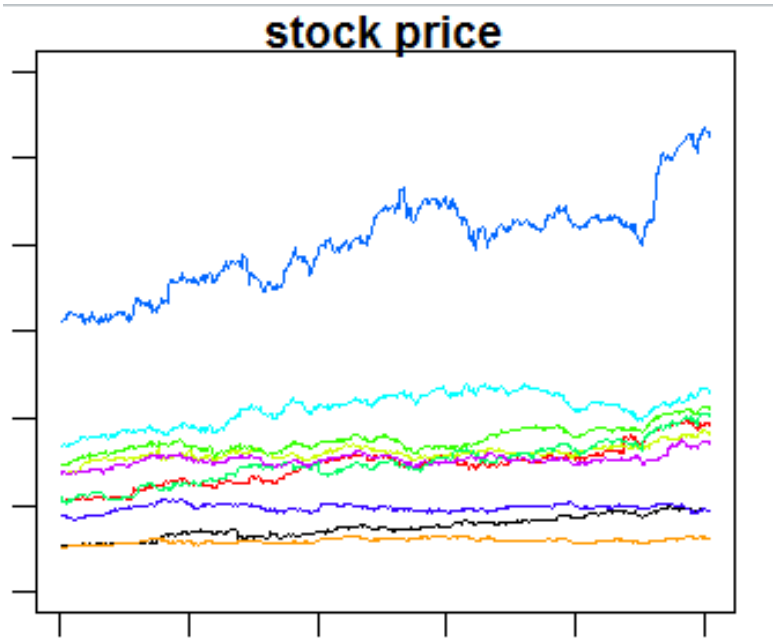
```
#Build stock matrix
date <- read.csv("date.csv", stringsAsFactors = F)
stocks10 <- cbind(date, MSFT[,4], NKE[,4], PFE[,4],
PG[,4], TRV[,4], UNH[,4], UTX[,4], V[,4], VZ[,4], WMT[,4])
colnames(stocks10) <- c("date", "MSFT", "NKE",
"PFE", "PG", "TRV", "UNH", "UTX", "V", "VZ")
```

```
> head(stocks10)
      date  MSFT   NKE   PFE    PG   TRV   UNH   UTX      V    VZ    NA
1 2013/1/2 27.62 51.84 25.91 69.39 72.86 54.54 84.00 155.38 44.27 69.24
2 2013/1/3 27.25 52.37 25.85 68.95 73.42 51.99 84.31 155.50 44.06 68.80
3 2013/1/4 26.74 52.88 25.96 69.09 74.06 52.09 84.98 156.77 44.30 69.06
4 2013/1/7 26.69 52.96 25.98 68.62 73.06 52.09 84.57 157.89 44.69 68.40
5 2013/1/8 26.55 52.40 26.02 68.51 73.19 51.40 83.55 159.36 43.10 68.59
6 2013/1/9 26.70 52.45 26.47 68.88 73.94 52.37 84.55 161.79 43.00 68.57
> }
```

4 Draw line plot based on close price on each stock. The result should be one single plot, containing 10 lines with dierent colors. You need to draw it twice by using basic plot function provided by R, and ggplot2 (with group). X axis should have date as label (e.g. 2013-01-01, or Jan 01, 2013, etc.), and y axis should have price value (e.g. 100.1, 50.5...).

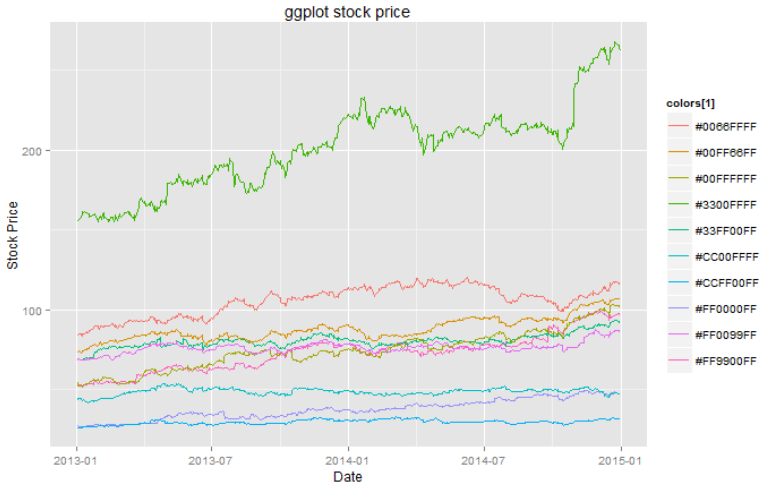
```
# Plot
par(mar=c(1,1,1,1))
colors <- rainbow(10)
```

```
plot( MSFT[,4], type="l",ylim = c(0, 300),
 xlab = "date " , ylab = "price")
lines( NKE[,4], type="l", lwd=1.5, col=colors[1] )
lines( PFE[,4], type="l", lwd=1.5, col=colors[2] )
lines( PG[,4], type="l", lwd=1.5, col=colors[3] )
lines( TRV[,4], type="l", lwd=1.5, col=colors[4] )
lines( UNH[,4], type="l", lwd=1.5, col=colors[5] )
lines( UTX[,4], type="l", lwd=1.5, col=colors[6] )
lines( V[,4], type="l", lwd=1.5, col=colors[7] )
lines( VZ[,4], type="l", lwd=1.5, col=colors[8] )
lines( WMT[,4], type="l", lwd=1.5, col=colors[9] )
title("stock price")
```




```
library(ggplot2)
xx <- rownames(MSFT)[]
xx <- as.Date(xx)
data <- data.frame(x = xx, y = stocks10[,2:11])
ggplot(data, aes(date)) +
    geom_line(aes(x = xx, y = data[,2], colour = colors[1])) +
    geom_line(aes(x = xx, y = data[,3], colour = colors[2])) +
    geom_line(aes(x = xx, y = data[,4], colour = colors[3])) +
    geom_line(aes(x = xx, y = data[,5], colour = colors[4])) +
    geom_line(aes(x = xx, y = data[,6], colour = colors[5])) +
    geom_line(aes(x = xx, y = data[,7], colour = colors[6])) +
```

```
geom_line(aes(x = xx, y = data[,8], colour = colors[7])) +
geom_line(aes(x = xx, y = data[,9], colour = colors[8])) +
geom_line(aes(x = xx, y = data[,10], colour = colors[9])) +
geom_line(aes(x = xx, y = data[,11], colour = colors[10])) +
xlab("Date") +
ylab("Stock Price") +
ggtitle("ggplot stock price")
```



2.2 Twitter Database

1 Does any twitter user have no description? If yes, how many users do not have description?

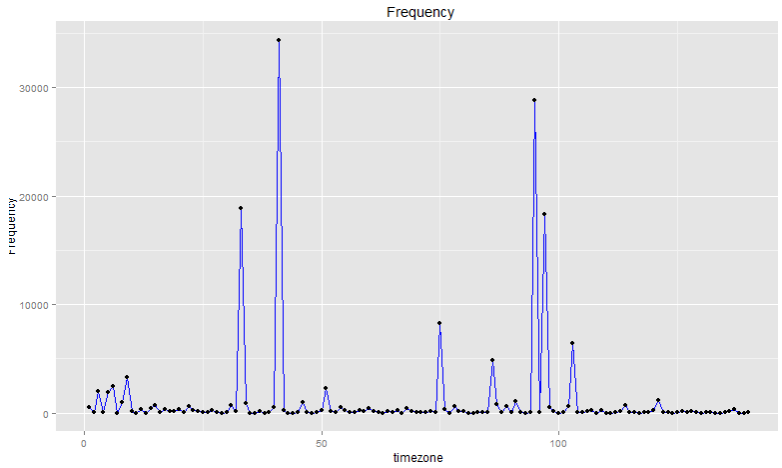
```
use fe513_twitter;
SELECT count(*) FROM twitter_profile where description = '';}
```

2 How many unique time zones did people use? Draw a histogram to show the frequency of people using dierent time zones. There are 2 columns for plot timezone name and frequency (e.g. timezone1, 100; timezone2, 150...) and the plot should have certain title and labels.

```
select time_zone, count(*) from twitter_profile group by time_zone

timez <- read.csv("twitter.csv", stringsAsFactors = F)
tiz <- table(timez)
head(tiz)
t<- data.frame(tiz)
```

```
x = t$timez
y = t$Freq
data = data.frame( c(1:140), y )
ggplot(data, aes(c(1:140),y)) + geom_line( color = "blue") + geom_point()+
    ggtitle( " Frequency")+labs(x = "timezone", y = "Frequency")
```



3,4 For those who are in Eastern time zone, get all their messages from message table. Clean those messages and eliminate meaningless words (like a, the, ...).

```
user_id IN (SELECT id FROM fe513_twitter.twitter_profile
 WHERE time_zone = 'Eastern Time (US & Canada)')
```

export as csv

```
 Connect to database
conn = dbConnect(RMySQL::MySQL(),host = "fscwinsrv04.fsc.stevens.edu" ,
                 user = "a513_Kejia",password = "Hh626491",
                 dbname = "fe513_twitter",port = 3306)

sentense <- read.csv("sentence.csv", stringsAsFactors = F)
head(sentense)
x <- c()
for( i in 1: length(sentense)){
    x <- c(x,sentense[i])
}

library(tm)
step1 <- Corpus(VectorSource(x))
```

```
step2 <- tm_map(step1, removePunctuation)

# Change all to lower case
step2 <- tm_map(step2, content_transformer(tolower))

# Delete "stop words" from the document
# Stop word can be read from existed list or customized list
stopw <- c(stopwords("en"), "legends")
step3 <- tm_map(step2, removeWords, stopw)

# Get the result of word & its frequency
step4 <- TermDocumentMatrix(step3)
step4 <- as.matrix(step4)


final <- data.frame(word = rownames(step4),freq1=step4[,1])
head(final, 10)

# Plot word cloud
library(wordcloud)
par(mfrow = c(1, 2))
wordcloud(words = final$word, freq = final$freq1,
min.freq = 2, colors=brewer.pal(8, "Dark2"))
```