stock_price_analysis01

Stock prices analysis

Original Source

Use basic descriptive statistics to explore daily prices of stocks.

Exercise 1

Load data directly and save in a variable.

Examine its following properties: 1. columns names 2. data class and class of each column of data 3. lowest date 4. highest date 5. distinct stock symbols

```
data <- read.csv("http://www.r-exercises.com/wp-content/uploads/2016/07/data.csv", header=TRUE, sep=','
colnames(data)
```

```
## [1] "Symbol"    "Date"      "Open"      "High"      "Low"       "Close"
## [7] "Volume"    "Adj_Close"
```

```
class(data)
```

```
## [1] "data.frame"
```

```
sapply(data, class)
```

```
##    Symbol       Date       Open       High        Low      Close     Volume
##  "factor"  "factor" "numeric" "numeric" "numeric" "numeric" "integer"
## Adj_Close
## "numeric"
```

```
min(as.Date(data$Date))
```

```
## [1] "2015-07-21"
```

```
max(as.Date(data$Date))
```

```
## [1] "2016-07-20"
```

```
unique(as.character(data$Symbol))
```

```
## [1] "BAC"  "GE"   "GOOG" "X"    "YHOO"
```

Exercise 2

Get a subset of the data containing only daily opening prices of BAC in the year 2016.

```
subset(data, data$Symbol=="BAC" & as.Date(data$Date)>=as.Date('2016-01-01') & as.Date(data$Date)<=as.Da
```

```
##   [1] 14.35 14.06 13.84 13.78 13.73 13.50 13.41 13.29 13.28 12.87 12.52
##  [12] 12.93 13.19 13.37 13.07 12.57 12.77 13.05 13.84 13.60 13.62 13.74
##  [23] 13.38 13.23 13.38 13.56 13.64 13.98 14.32 14.35 14.54 14.44 14.46
##  [34] 14.95 14.60 15.03 14.76 14.98 14.83 14.60 14.54 14.64 14.60 14.02
##  [45] 13.89 13.82 14.15 14.30 14.25 14.08 14.08 13.83 14.15 14.09 14.51
##  [56] 14.58 14.73 14.92 15.02 15.02 15.02 14.87 14.93 14.56 14.26 13.85
##  [67] 14.27 13.71 13.55 13.00 12.92 13.03 13.15 13.20 13.30 13.54 13.47
##  [78] 13.49 13.49 13.54 13.73 13.41 13.77 13.67 13.80 13.68 13.22 13.51
##  [89] 13.51 13.72 13.44 13.23 13.17 13.40 13.45 13.76 13.38 13.20 12.64
## [100] 12.70 12.49 12.14 11.96 12.47 12.13 12.22 12.71 12.57 12.38 11.48
## [111] 11.46 12.42 11.99 12.67 13.32 12.89 13.28 13.74 14.05 13.66 13.59
## [122] 13.20 13.07 13.54 13.65 13.67 13.79 14.69 14.41 15.01 15.47 15.54
## [133] 15.26 15.94 15.73 16.19 16.52 16.45
```

Exercise 3

For each stock, obtain the lowest and highest closing price together with dates when these prices occured.

```
maxClose <- aggregate(data$Close, list(data$Symbol), max)
maxClose$maxDate <- data[with(data, ave(Close, Symbol, FUN=max)==Close), 2]
names(maxClose) <- c("Symbol", "Max Price", "Max Date")
minClose <- aggregate(data$Close, list(data$Symbol), min)
minClose$minDate <- data[with(data, ave(Close, Symbol, FUN=min)==Close), 2]
names(minClose) <- c("Symbol", "Min Price", "Min Date")
merge(maxClose, minClose, by="Symbol")
```

```
##   Symbol Max Price   Max Date Min Price   Min Date
## 1    BAC     18.45 2015-07-22     11.16 2016-02-11
## 2     GE     32.91 2016-07-18     23.27 2015-08-25
## 3   GOOG    776.60 2015-12-29    582.06 2015-08-25
## 4      X     21.72 2016-07-18      6.67 2016-01-27
## 5   YHOO     39.73 2015-07-21     26.76 2016-02-11
```

Exercise 4

Write a function called AvgPrice that returns average closing price for a given stock symbol and a period of time between given start and end date.

Using the function, what is the average closing price of X in January of 2016?

```
AvgPrice <- function(d, symbol, start, end){
  mean(subset(d, d$Symbol==symbol & as.Date(d$Date)>=as.Date(start) & as.Date(d$Date)<=as.Date(end))$Cl
}

AvgPrice(data, "X", '2016-01-01', '2016-01-31')
```

```
## [1] 7.220526
```

Exercise 5

Write a function called WAvgPrice that returns weighted average closing price for a given stock symbol and given period of time between start and end date. (Tip: Weighted average closing price is calculated as sum(close * volume)/sum(volume))

Using the function, what is the weighted average closing price of GE in year 2016.

```
WAvgPrice <- function(d, symbol, start, end){
  tmp <- subset(d, d$Symbol==symbol & as.Date(d$Date)>=as.Date(start) & as.Date(d$Date)<=as.Date(end))
  tmp$wPrice = tmp$Close * tmp$Volume
  sum(tmp$wPrice) / sum(as.numeric(tmp$Volume))
}

WAvgPrice(data, "GE", '2016-01-01', '2016-12-31')
```

```
## [1] 29.96373
```

Exercise 6

Reshape original data frame to get data frame with dates as the first column and closing prices of stocks in other columns, each column containing one stock. Name the columns after stock symbol.

Order resulting data frame by date and save it in a variable for later use.

```
data.close <- reshape(data[c("Symbol", "Date", "Close")], timevar="Symbol", idvar="Date", direction="wi
colnames(data.close) <- c("Date", as.character(unique(data$Symbol)))
data.close <- data.close[with(data.close, order(Date)), ]
```

Exercise 7

Using the variable from the previous exercise, calculate daily return for each stock and save result in new variable for later use. Daily return is calculated as (pricet-price(t-1))/price(t-1).

Calculate average daily return for each stock.(Tip: average rate is calculated as a geometric mean)

```
data.return <- data.frame(Date=data.close$Date[-1], sapply(data.close[-1], function(x){
  diff(x) / x[-length(x)]+1
}))
sapply(data.return[-1], function(x){
  round((prod(x, na.rm=TRUE)^(1/NROW(x))-1)*100, 2)
})
```

```
##   BAC    GE  GOOG     X  YHOO
## -0.09  0.08  0.04  0.06 -0.01
```

Exercise 8

Using daily return data from previous exercise, which stock symbol has highest average risk, calculated as standard deviation of daily returns?

```
risks <- sapply(data.return[as.Date(data.return$Date)>=as.Date('2016-01-01') & as.Date(data.return$Date
names(risks)[which(risks==max(risks))]
```

```
## [1] "X"
```

Exercise 9

Write a function called LowestRisk that returns stock symbol with lowest risk in a given period of time between start and end. Call the function to obtain lowest risk stock symbol.

```
LowestRisk <- function(d, start, end){
  data.period = subset(d, as.Date(d$Date)>=as.Date(start) & as.Date(d$Date)<=as.Date(end))
  risks <- sapply(data.period[-1], sd)
  names(risks)[which(risks == min(risks))]
}
```

```
LowestRisk(data.return, '2016-01-01', '2016-01-31')
```

```
## [1] "GE"
```

Exercise 10

Using daily returns of stocks from exercise 7, calculate the correlation coefficient matrix which includes every stock symbol in the data set.

```
cor(subset(data.return, select=-c(Date)), use="pairwise.complete.obs")
```

```
##            BAC        GE      GOOG         X      YHOO
## BAC  1.0000000 0.6522450 0.4747321 0.3973756 0.4802372
## GE   0.6522450 1.0000000 0.5027091 0.4335825 0.4462588
## GOOG 0.4747321 0.5027091 1.0000000 0.2128376 0.4345335
## X    0.3973756 0.4335825 0.2128376 1.0000000 0.3039172
## YHOO 0.4802372 0.4462588 0.4345335 0.3039172 1.0000000
```