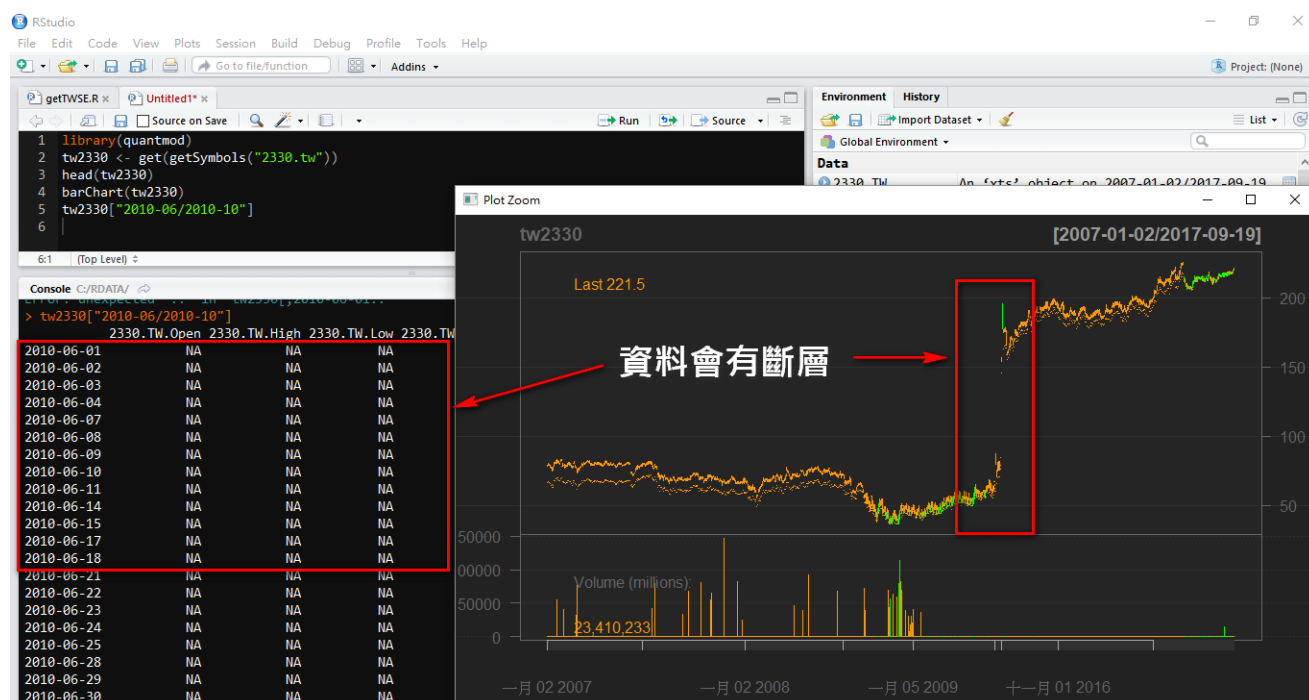


自己寫個 Function 抓證交所的股票資訊

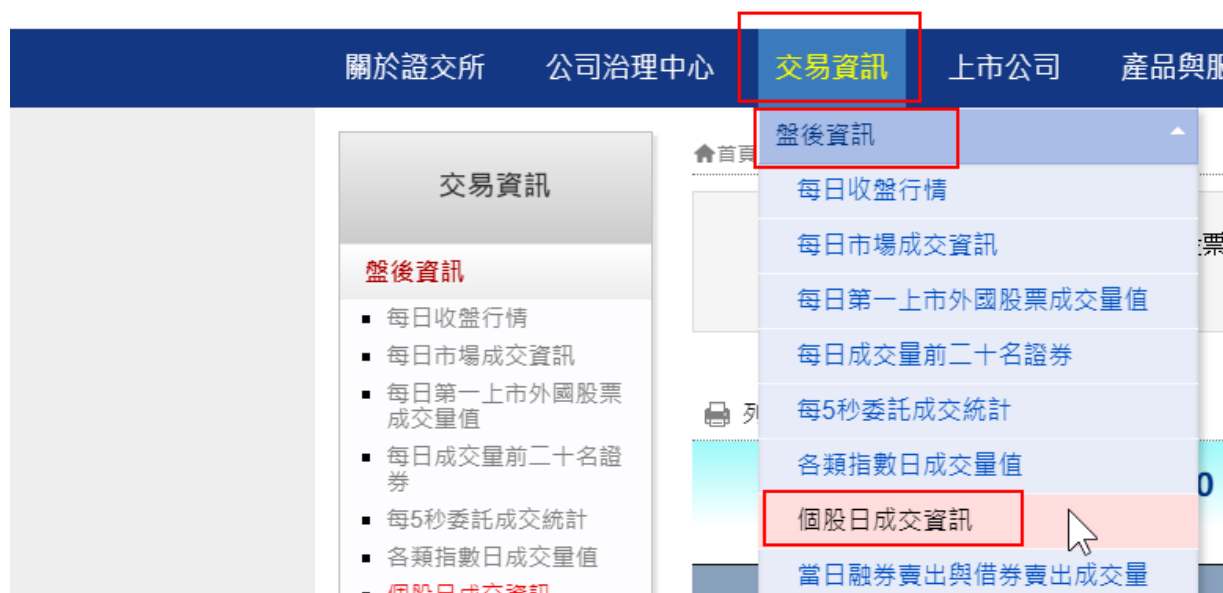
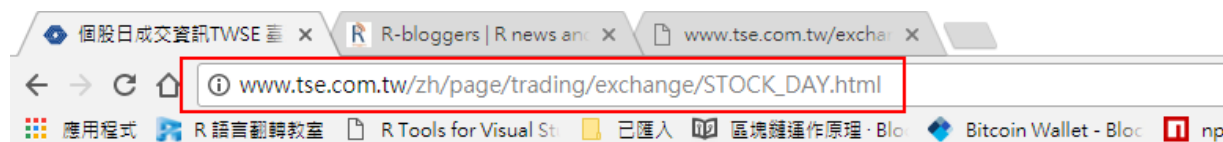
金融上的套件: quantmod, 可在 Yahoo finance、Googlefinance 等網站, 下載公開數據進行分析, 但對於台股而言, 雖然可在 Yahoo finance 抓取, 但有時資料並不完整。

以台積電(2330)為例, 透過 getSymbols("2330.tw") 抓取回來的資料, 發現有缺漏。

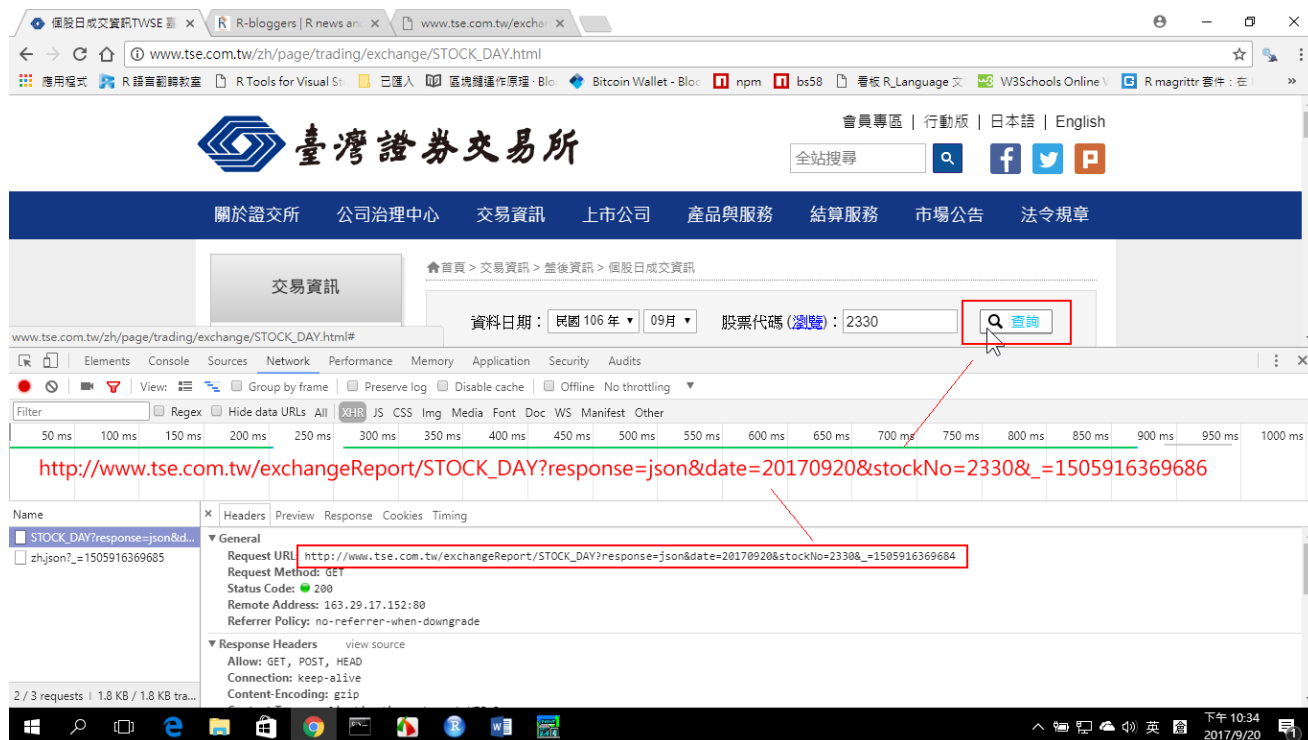


因此改抓台灣的證券交易所, 就是我們的目標。

網路上有許多類似的方法, 可抓取證交所的股票資料, 在這裏我也嚐試自己寫一個 Function 首先是利用 Google Chrome 開啟證交所網頁, 點[交易資訊] -> [盤後資訊] -> [個股日成交資訊]



按下 **F12**，可進入開發人員工具模式，有利我們觀察網頁的運作，以下圖為例
當按下[查詢]後，可發現網頁以 **GET** 方式，送出一串 **URL**，仔細觀察可發現 **URL** 中帶有個參數，其中 **response=json**，代表回傳格式為 **JSON**，**date=20170920**，代表查詢日期，實測結果發現，日期的最後 2 碼並不影響查詢結果，即 **date=20170901** 與 **date=20170920** 查回來的結果是相同的，稍後的自訂函數中，天數統一用 **01**，**stockNo=2330** 為股票代號，最後一個參數 **_=1505916369686**，為時間戳記，隨時間變動。



回傳 JSON



有了上述的基本觀念，將有利於 **Function** 的寫作。

函數：getTWSE(stockNo, from = c(year, month), to = c(year, month))，預設有 3 個參數，皆帶有預設值；當呼叫函數而未帶參數時，系統會用預設值當查詢條件。

stockNo = "2330"，預設為台積電(2330)

```
24 # =====
25 getTWSE <- function(stockNo = "2330",
26                       from = c(as.integer(format(Sys.Date(), "%Y")), as.integer(format(Sys.Date(), "%m"))),
27                       to = c(as.integer(format(Sys.Date(), "%Y")), as.integer(format(Sys.Date(), "%m")))) {
```

From、to，若未輸入，預設為查詢當下年月，如下圖

```
Console C:/RDATA/
> from = c(as.integer(format(Sys.Date(), "%Y")), as.integer(format(Sys.Date(), "%m")))
> to = c(as.integer(format(Sys.Date(), "%Y")), as.integer(format(Sys.Date(), "%m")))
> from
[1] 2017 9 ← from
> to
[1] 2017 9 ← to
> |
```

行 43~49，主要是檢查 RStudio 是否已載入 jsonlite、xts 套件，未載入則印出訊息並中斷 Function，也可以改寫成未載入，則自動安裝載入套件

```
40 # =====
41 # Using this function, you must library(jsonlite) & library(xts)
42 # =====
43 packages <- gsub("package:", replacement = "", search())
44 if (!("jsonlite" %in% packages)){
45   print("Error: The 'jsonlite' package has not been loaded.")
46   return(NULL)
47 } else if (!("xts" %in% packages)){
48   print("Error: The 'xts' package has not been loaded.")
49   return(NULL)
50 }
```

改為偵測到未載入則進行安裝及載入動作

```
1 # =====
2 # Using this function, you must install.packages("jsonlite") and library(jsonlite)
3 # =====
4 packages <- gsub("package:", replacement = "", search())
5 if (!("jsonlite" %in% packages)){
6   install.packages("jsonlite") ← 安裝並載入
7   library(jsonlite)
8 }
```

行 65~68，判斷 from 及 to 的內容，若僅有年度，則補上月份

```
51 # =====
52 # parameter check & parse
53 # =====
54 if (from[1] > to[1]){
55   print("The starting year is greater than the deadline.")
56   return(NULL)
57 }
58 nowYear <- as.integer(format(Sys.Date(), "%Y"))
59 nowMonth <- as.integer(format(Sys.Date(), "%m"))
60 MM <- c("01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12")
61 queryDate <- NULL
62 historyStock <- NULL
63 startYM <- from
64 endYM <- to
65 if (length(startYM) == 1) startYM <- c(startYM, 1)
66 if (length(endYM) == 1){
67   ifelse (endYM[1] == nowYear, endYM <- c(endYM, nowMonth), endYM <- c(endYM, 12))
68 }
69 if (startYM[2] < 1 | startYM[2] > 12 | endYM[2] < 1 | endYM[2] > 12){
70   print("Month must be between 1 ~ 12.")
71   return(NULL)
72 }
```

行 77~89，計算出查詢區間的明細，待會組 URL 時，參數 date 的內容

```
76 # =====
77 if (startYM[1] == endYM[1]) {
78   queryDate <- paste0(startYM[1], MM[startYM[2]:endYM[2]], "01")
79 } else if ((endYM[1] - startYM[1]) == 1) {
80   queryDate <- c(paste0(startYM[1], MM[startYM[2]:12], "01"),
81                 paste0(endYM[1], MM[1:endYM[2]], "01"))
82 } else {
83   tmpY <- c((startYM[1]+1):(endYM[1]-1))
84   queryDate <- paste0(startYM[1], MM[startYM[2]:12], "01")
85   for (tY in tmpY){
86     queryDate <- c(QueryDate, paste0(tY, MM, "01"))
87   }
88   queryDate <- c(QueryDate, paste0(endYM[1], MM[1:endYM[2]], "01"))
89 }
```

行 95，組查詢網址，傳入 4 個參數；行 97，真正到證交所抓資料，判斷回傳的 JSON，取 stat=OK 的資料，組 OHLC 格式(Open, High, Low, Close)

```
90 # =====
91 # to TWSE get History Stock
92 # =====
93 for (qyDate in queryDate){
94   ttime <- as.character(as.integer(as.POSIXct(Sys.time()))*100)
95   twseUrl <- paste0(url, "response=", response, "&date=", qyDate, "&stockNo=", stockNo,
96                   "&_=", ttime)
97   jsonData <- fromJSON(twseUrl, flatten = TRUE)
98   if (jsonData$stat == "OK"){
99     tmpStock <- data.frame(jsonData$data[, 1],
100                           jsonData$data[, 4:7],
101                           jsonData$data[, 2:3],
102                           stringsAsFactors = FALSE)
103     historyStock <- rbind(historyStock, tmpStock)
104   }
105 }
```

組查詢網址

讀取JSON

將資料組成OHLC格式

合併查詢結果

行 116~122，資料格式轉換，行 116，利用自訂函數 CNV_DATE()，透過 supply 函數將民國年轉為西元年，行 117~122，則是將 string 轉為 number，因為含千分號(,)，故先要濾掉，行 124，將 data frame 轉為 xts 格式

```
113 # =====
114 if (!is.null(historyStock)){
115   colnames(historyStock) <- c("Date", "Open", "High", "Low", "Close", "Number", "Value")
116   historyStock$Date <- sapply(historyStock$Date, CNV_DATE)
117   historyStock$Open <- as.numeric(gsub(',', replacement = '', historyStock$Open))
118   historyStock$High <- as.numeric(gsub(',', replacement = '', historyStock$High))
119   historyStock$Low <- as.numeric(gsub(',', replacement = '', historyStock$Low))
120   historyStock$Close <- as.numeric(gsub(',', replacement = '', historyStock$Close))
121   historyStock$Number <- as.numeric(gsub(',', replacement = '', historyStock$Number))
122   historyStock$Value <- as.numeric(gsub(',', replacement = '', historyStock$Value))
123   # data frame to xts
124   historyStock <- xts(historyStock[, -1], order.by = as.Date(historyStock[, 1]))
125
126   print(paste0(msg, ", Stock information, download complete. rows=", nrow(historyStock)))
127 } else{
128   print(paste0(msg, ", No Data found. "))
129 }
130 # =====
131 return(historyStock)
132 }
```

利用自訂函數, 將民國年轉為西元年

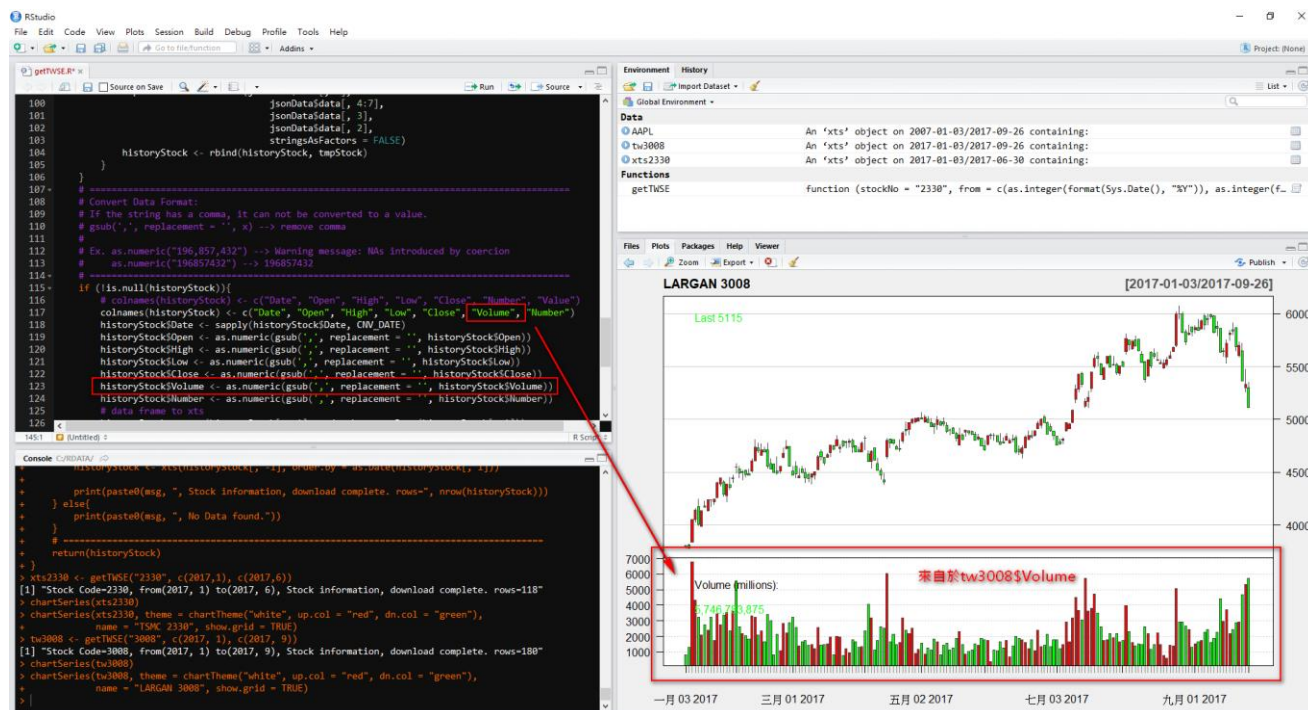
轉成xts

```
Console C:/RDATA/
> as.numeric("1,234,567,890")
[1] NA
Warning message:
NAs introduced by coercion
> as.numeric(gsub(',', replacement = '', "1,234,567,890"))
[1] 1234567890
>
```

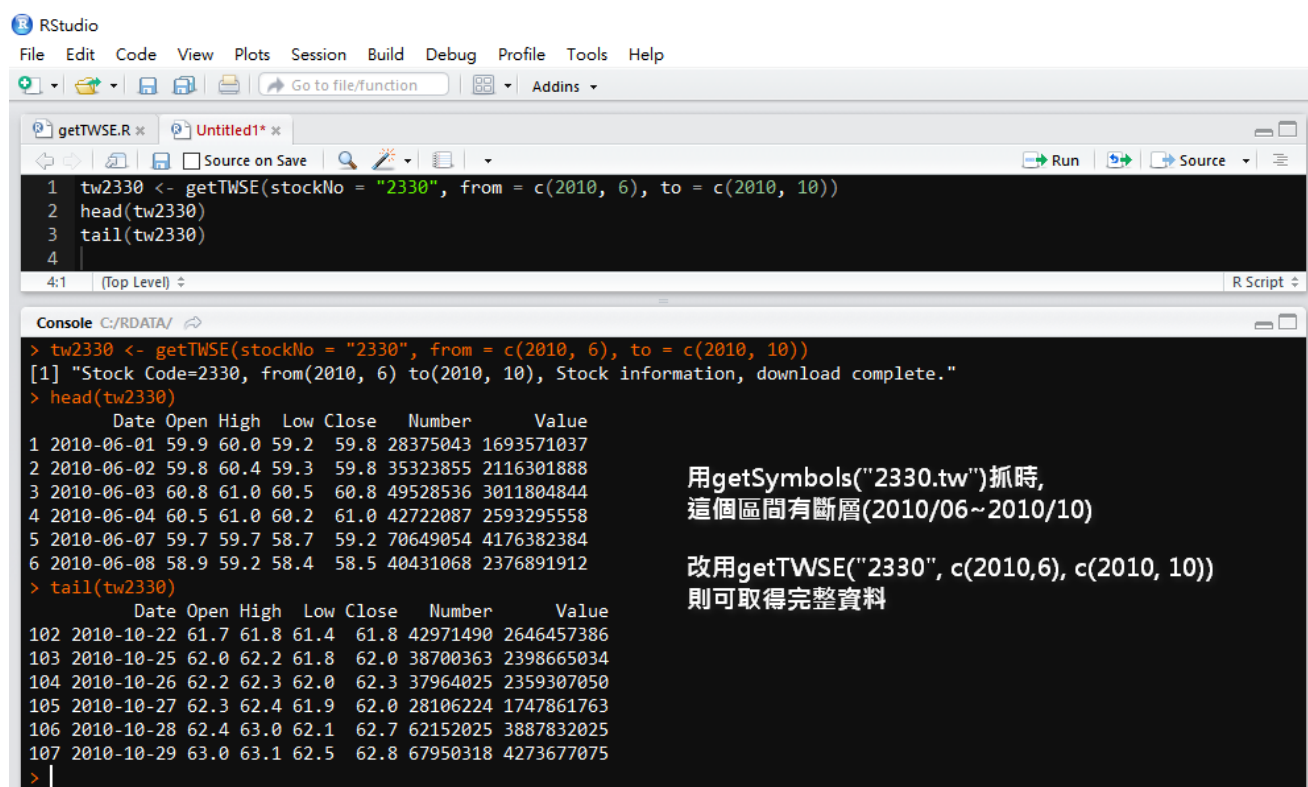
字串含千分號, 無法轉為數值

濾掉千分號, 字串轉為數值

修正: 行 117, 設定 `colnames` 時, 為了能在 `chartSeries` 畫出正確的圖形, 在資料集中定要有一個 Column 叫"Volume", 圖形才會完整。



驗測結果如下圖, 原來用 `getSymbols` 抓取為斷層的部份, 改用 `getTWSE` 可正常取得資料。



完整程式, 請至 [GitHub](#) 下載

GitHub: <https://github.com/dong945/R>

完整程式碼

```
# ++++++
# Author: TUNG-SHENG, CHEN
# DATE: 2017/09/20
# Package: jsonlite, xts
# ++++++
# install.packages(c("jsonlite", "xts"))
library(jsonlite)
library(xts)
# =====
# Parameter:
# stockNo: String, Stock code of TWSE, default: 2330 (TSMC)
# from: Vector, c(year, month), default: now
# to: Vector, c(year, month), default: now
# Examples:
# ## return '2330', this month
# getTWSE()
#
# ## return '2330', from 2016/01 ~ now
# getTWSE("2330", 2016)
#
# ## return '2330', from 2015/01 ~ 2016/12
# getTWSE("2330", 2015, 2016)
#
# ## return '2330', from 2015/06 ~ 2017/03
# getTWSE("2330", c(2105, 6), c(2017, 3))
# =====
```

```

getTWSE <- function(stockNo = "2330",
                    from = c(as.integer(format(Sys.Date(), "%Y")), as.integer(format(Sys.Date(), "%m"))),
                    to = c(as.integer(format(Sys.Date(), "%Y")), as.integer(format(Sys.Date(), "%m")))) {
# =====
# Inside Function: Change Date to yyyy-mm-dd
# =====
CNV_DATE <- function(x){
  TMP <- strsplit(x, split = "/")
  paste(as.integer(TMP[[1]][1])+1911, TMP[[1]][2], TMP[[1]][3], sep = "-")
}
# =====
response <- "json"
url <- "http://www.tse.com.tw/exchangeReport/STOCK_DAY?"
# =====
# Using this function, you must library(jsonlite) 、 library(xts)
# =====
packages <- gsub("package:", replacement = "", search())
if (!("jsonlite" %in% packages)){
  print("Error: The 'jsonlite' package has not been loaded.")
  return(NULL)
} else if (!("xts" %in% packages)){
  print("Error: The 'xts' package has not been loaded.")
  return(NULL)
}
# =====
# parameter check & parse
# =====

```

```

if (from[1] > to[1]){
  print("The starting year is greater than the deadline.")
  return(NULL)
}
nowYear <- as.integer(format(Sys.Date(), "%Y"))
nowMonth <- as.integer(format(Sys.Date(), "%m"))
MM <- c("01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12")
queryDate <- NULL
historyStock <- NULL
startYM <- from
endYM <- to
if (length(startYM) == 1) startYM <- c(startYM, 1)
if (length(endYM) == 1){
  ifelse (endYM[1] == nowYear, endYM <- c(endYM, nowMonth), endYM <- c(endYM, 12))
}
if (startYM[2] < 1 | startYM[2] > 12 | endYM[2] < 1 | endYM[2] > 12){
  print("Month must be between 1 ~ 12.")
  return(NULL)
}
msg <- paste0("Stock Code=", stockNo,
              ", from(", startYM[1], ", ", startYM[2], ")",
              " to(", endYM[1], ", ", endYM[2], ")")
# =====
if (startYM[1] == endYM[1]) {
  queryDate <- paste0(startYM[1], MM[startYM[2]:endYM[2]], "01")
} else if ((endYM[1] - startYM[1]) == 1) {
  queryDate <- c(paste0(startYM[1], MM[startYM[2]:12], "01"),

```



```

        paste0(endYM[1], MM[1:endYM[2]], "01"))
    } else {
        tmpY <- c((startYM[1]+1):(endYM[1]-1))
        queryDate <- paste0(startYM[1], MM[startYM[2]:12], "01")
        for (tY in tmpY){
            queryDate <- c(QueryDate, paste0(tY, MM, "01"))
        }
        queryDate <- c(QueryDate, paste0(endYM[1], MM[1:endYM[2]], "01"))
    }
# =====
# to TWSE get History Stock
# =====
for (qyDate in queryDate){
    ttime <- as.character(as.integer(as.POSIXct(Sys.time()))*100)
    twseUrl <- paste0(url, "response=", response, "&date=", qyDate, "&stockNo=", stockNo,
                      "&_=", ttime)
    jsonData <- fromJSON(twseUrl, flatten = TRUE)
    if (jsonData$stat == "OK"){
        tmpStock <- data.frame(jsonData$data[, 1],
                               jsonData$data[, 4:7],
                               jsonData$data[, 2:3],
                               stringsAsFactors = FALSE)
        historyStock <- rbind(historyStock, tmpStock)
    }
}
# =====
# Convert Data Format:

```

```

# If the string has a comma, it can not be converted to a value.
# gsub(',', replacement = "", x) --> remove comma
#
# Ex. as.numeric("196,857,432") --> Warning message: NAs introduced by coercion
#      as.numeric("196857432") --> 196857432
# =====
if (!is.null(historyStock)){
  colnames(historyStock) <- c("Date", "Open", "High", "Low", "Close", "Volume", "Number")
  historyStock$Date <- sapply(historyStock$Date, CNV_DATE)
  historyStock$Open <- as.numeric(gsub(',', replacement = "", historyStock$Open))
  historyStock$High <- as.numeric(gsub(',', replacement = "", historyStock$High))
  historyStock$Low <- as.numeric(gsub(',', replacement = "", historyStock$Low))
  historyStock$Close <- as.numeric(gsub(',', replacement = "", historyStock$Close))
  historyStock$Volume <- as.numeric(gsub(',', replacement = "", historyStock$Volume))
  historyStock$Number <- as.numeric(gsub(',', replacement = "", historyStock$Number))
  # data frame to xts
  historyStock <- xts(historyStock[, -1], order.by = as.Date(historyStock[, 1]))

  print(paste0(msg, ", Stock information, download complete. rows=", nrow(historyStock)))
} else{
  print(paste0(msg, ", No Data found."))
}
# =====
return(historyStock)
}

```

測試抓取大立光股價

```
# =====  
# TEST  
# =====  
library(quantmod)  
tw3008 <- getTWSE("3008", 2017)  
head(tw3008)  
tail(tw3008)  
nrow(tw3008)  
barChart(tw3008)  
# =====  
# use chartTheme, set up.col = 'red'  dn.col = 'green'  
# =====  
chartSeries(tw3008, theme = chartTheme("white", up.col = "red", dn.col = "green"),  
            name = "大立光 3008", show.grid = TRUE)
```

Plot Zoom

— □ ×

LARGAN 3008

[2017-01-03/2017-09-26]

Last 5115

