

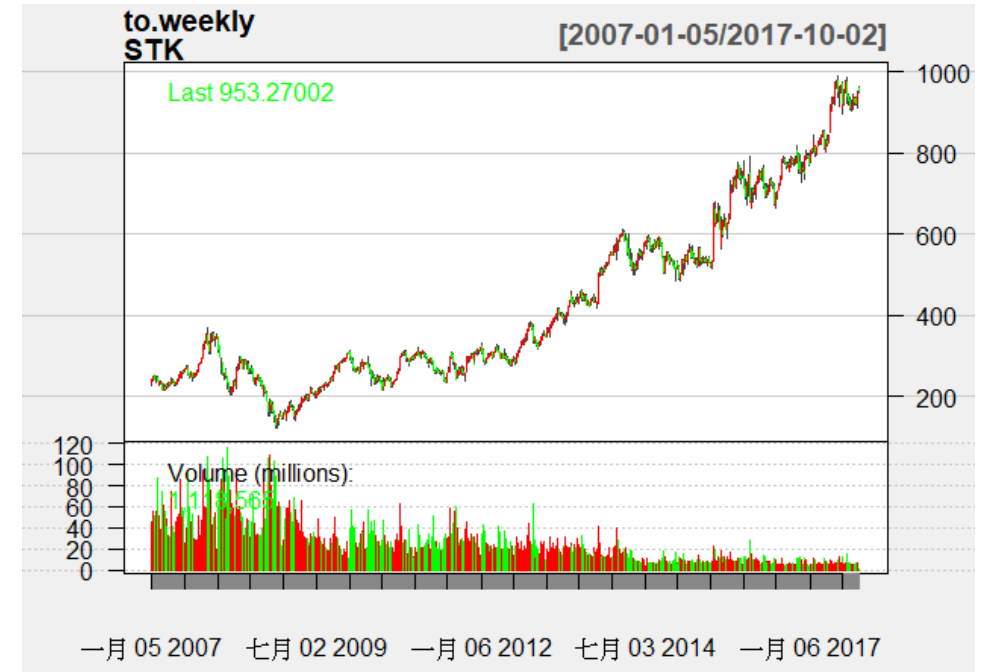
交易策略建構

學程式語言最快的方式：複製、貼上、修改
程式語言的功力 與 時間 成正比

回測方式：迴圈 v.s. 向量

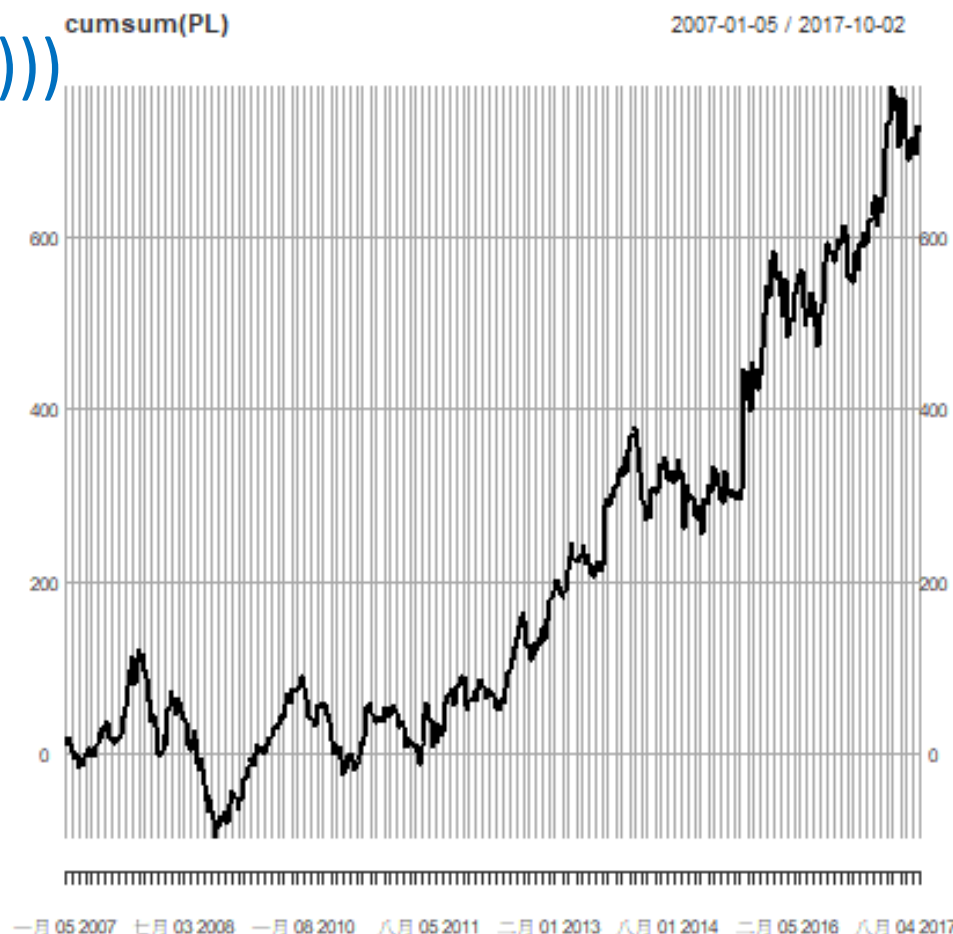
周K開盤買、收盤賣 (Yahoo Finance, GOOG)

- `library(quantmod)`
- `stock= "GOOG"`
- `duration="2007-01-01::2017-03-04"`
- `STK=get(getSymbols(stock))`
- `chartSeries(to.weekly(STK)[duration],theme = "white",up.col="red",dn.col = "green")`



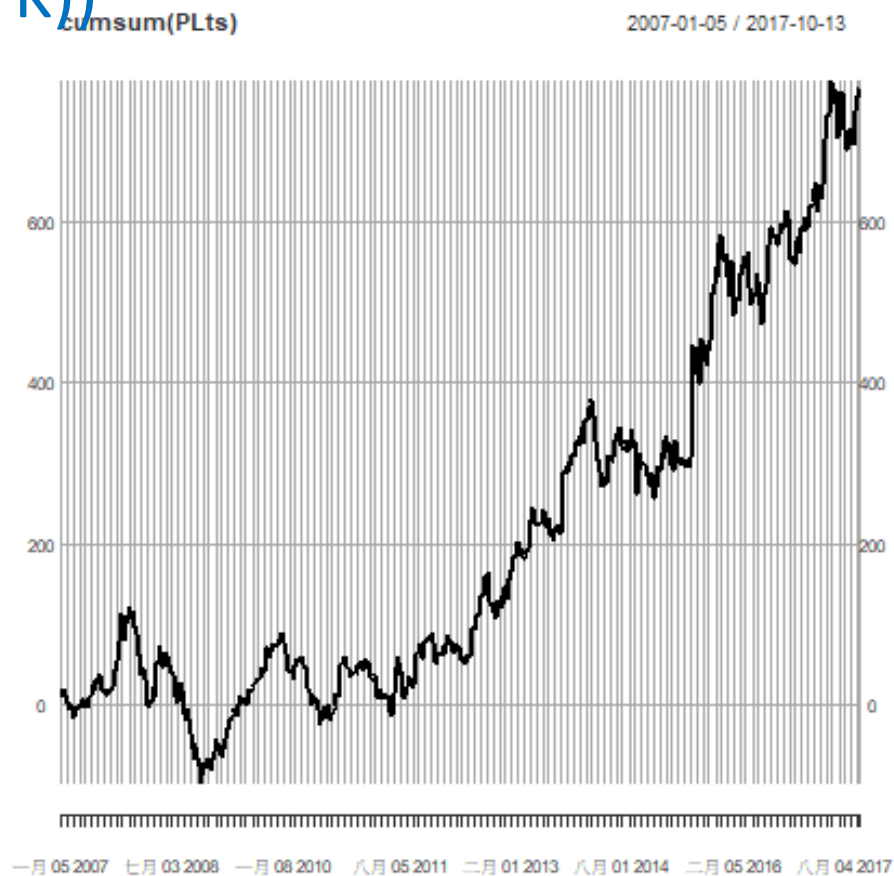
畫累計損益(時間序列)

- `STK=to.weekly(STK)`
- `View(cbind(Op(STK),Cl(STK),Cl(STK)-Op(STK)))`
- `PL=Cl(STK)-Op(STK)`
- `sum(PL)`
- `plot(cumsum(PL))`
- `class(PL)`
- `time(PL)`
- `class(time(PL))`



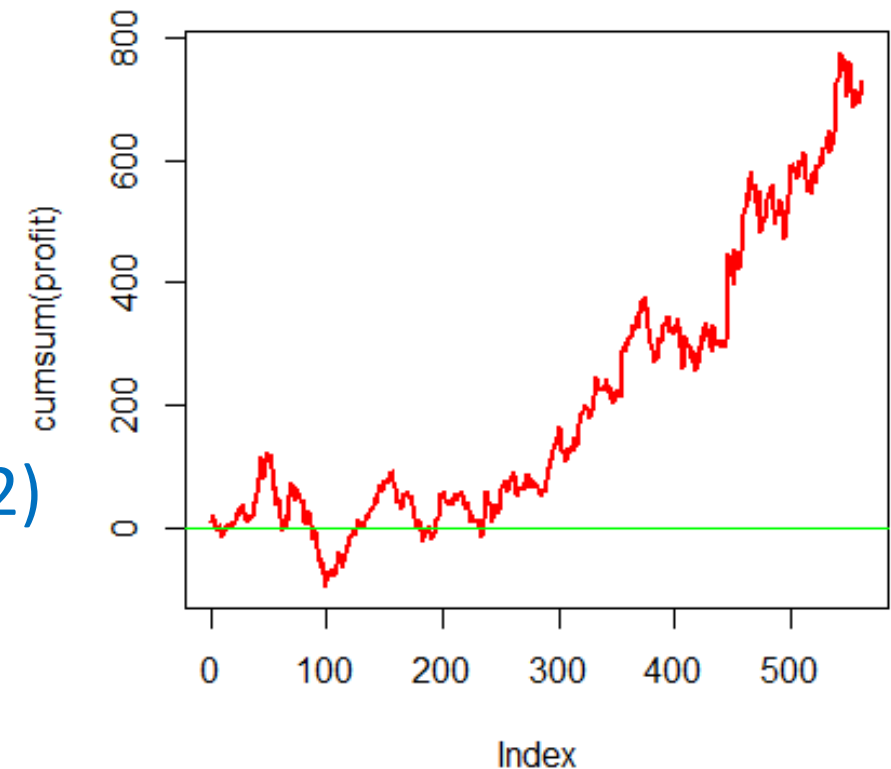
累計損益圖 (時間序列，用迴圈方式)

- `STK=to.weekly(STK)`
- `PLts=xts(numeric(length(time(STK))),time(STK))`
 - #產生時間序列的損益向量
- `for (m in as.character(time(STK))) {`
- `PLts[m]=STK[m,4]-STK[m,1]`
- `}`
- `plot(cumsum(PLts))`



累計損益圖 (非時間序列，用迴圈方式)

- `profit=setNames(numeric(length(time(STK))),time(STK))`
- `for (m in as.character(time(STK))) {`
- `profit[[m]]=STK[m,4]-STK[m,1]`
- `}`
- `plot(cumsum(profit),type="l",col="red",lwd=2)`
- `abline(h=0,col="green")`



開高買開低賣 (收盤平倉)

- 有時為了方便coding，會將時間序列轉換成矩陣
- `STK=as.matrix(STK)`
 - `##` 轉換成矩陣，依然可以用於quantmod
- 判斷開高or開低，第一天是否可以交易？
- `LastC=CI(STK)[1]`
 - `##`紀錄第一天的收盤價

開高買開低賣 (收盤平倉)

- `STK=as.matrix(to.weekly(STK))` ##轉換成周矩陣
- `LastC=Cl(STK)[1]` ##紀錄第一天的收盤價
- `for (m in 2:length(time(STK)))`
 - `if(Cl(STK)[m]>LastC){profit[m]=STK[m,4]-STK[m,1]}`
 - `if(Cl(STK)[m]<LastC){profit[m]=STK[m,1]-STK[m,4]}`
 - `LastC=Cl(STK)[m]`
 - `}`
- `plot(cumsum(profit),type="l",col="red",lwd=2)`
- `abline(h=0,col="green")`

課堂練習&回家作業

- 開高1%買進，開低1%賣出，收盤平倉的累計績效圖？
- 尋找標的，選定週期，回測開高?%，開低?%績效最好？