quantstrat 深度剖析

林佳緯

程式交易實務 – 使用 R 語言（二）

財團法人中華民國
證券暨期貨市場發展基金會
SECURITIES & FUTURES INSTITUTE

課程綱要

1. quantstrat 套件安裝
2. Time series 資料介紹
3. quantstrat 交易策略開發詳解
4. quantstrat 回測資料分析
5. 前推移動式分析(Walk Forward Analysis)

quantstrat 套件安装

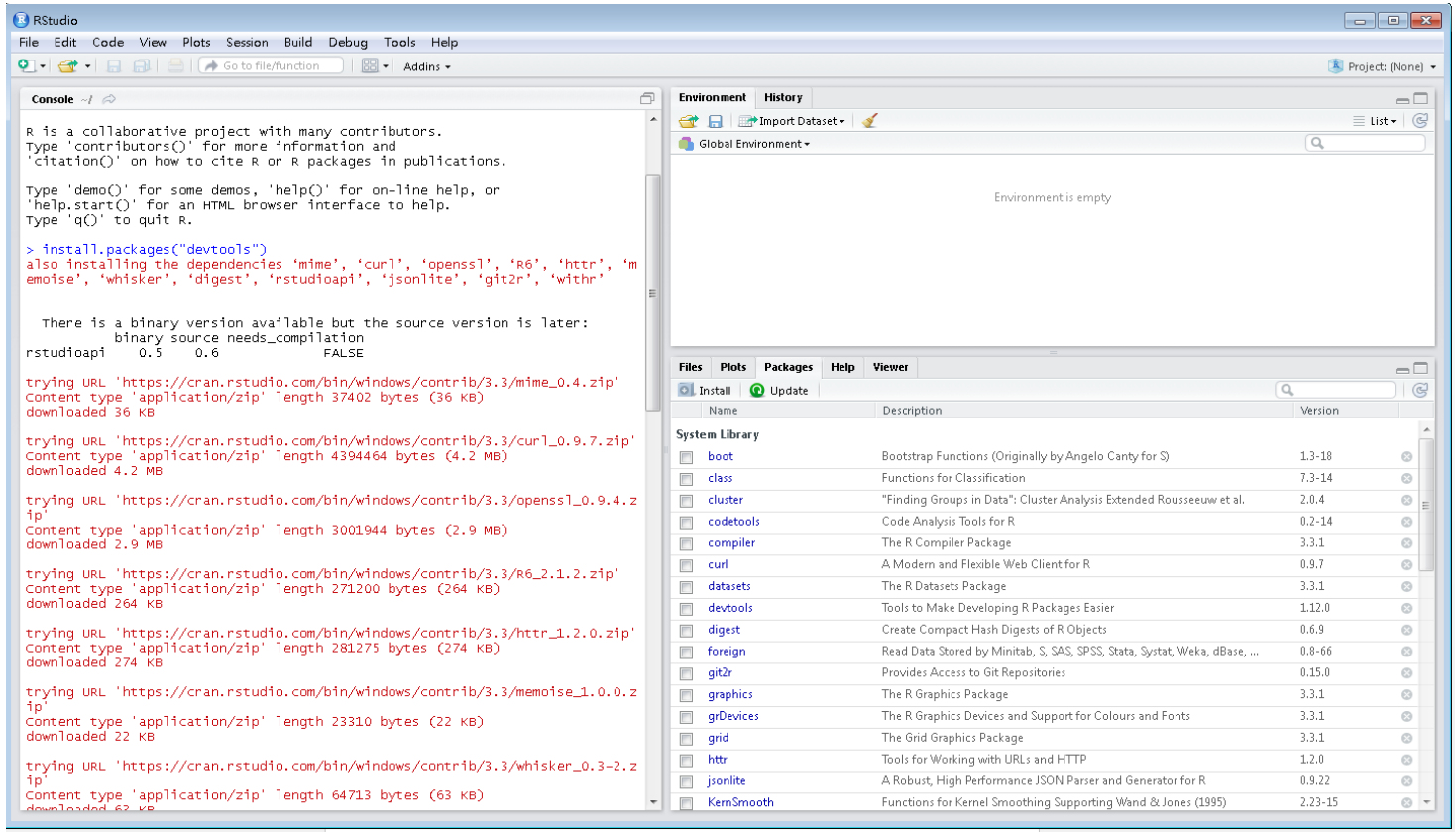
安裝計量套件指令

- 執行指令:

```
install.packages("devtools")
require(devtools)


install.packages("FinancialInstrument", repos="http://R-Forge.R-project.org")

install.packages("blotter", repos="http://R-Forge.R-project.org", type="source")

install.packages("quantstrat", repos="http://R-Forge.R-project.org", type="source")

install.packages("PortfolioAnalytics")
```

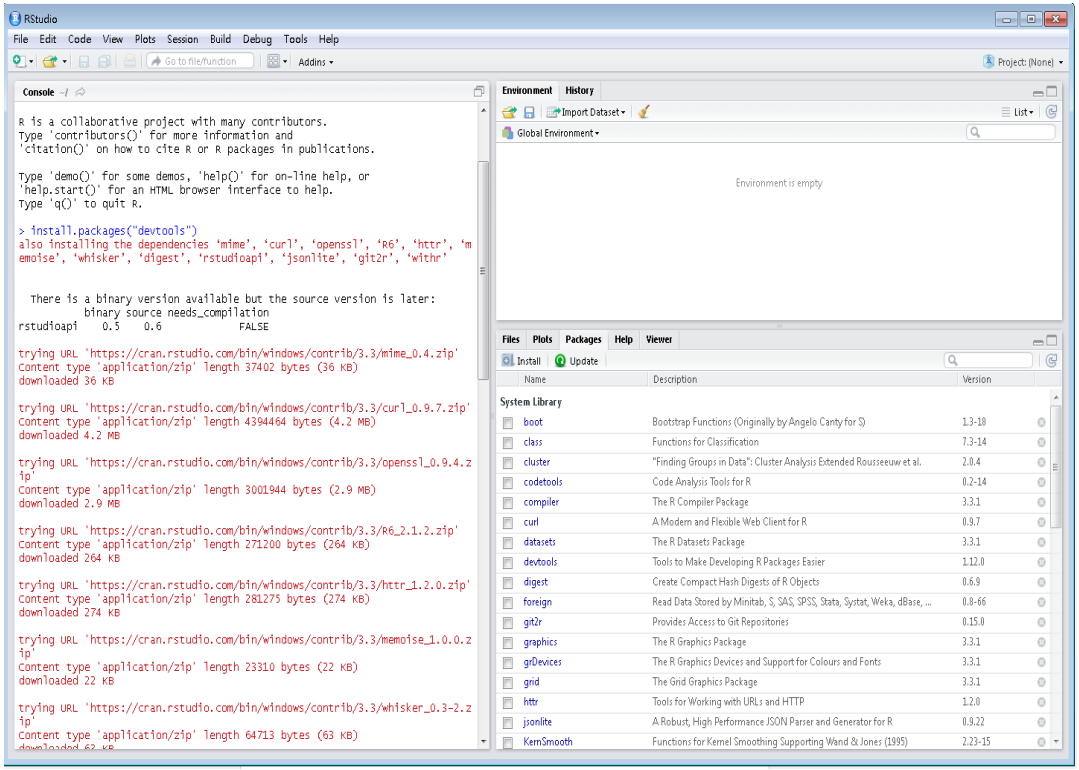
先安裝R 開發工具套件

```
install.packages("devtools")
require(devtools)
```

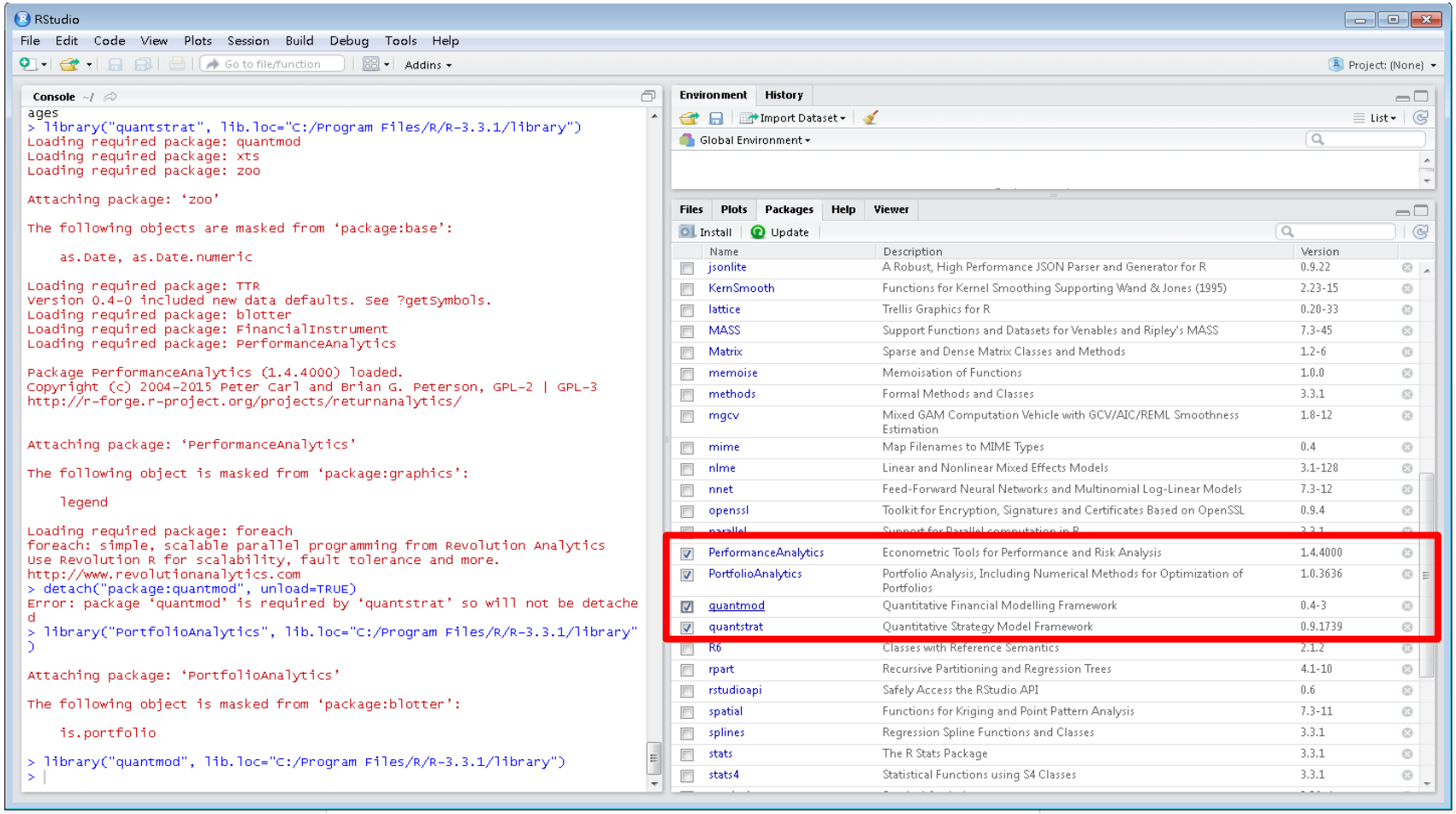


逐一安裝quantstrat相關套件

```
install.packages("FinancialInstrument",
repos="http://R-Forge.R-project.org")

install.packages("blotter",
repos="http://R-Forge.R-project.org",
type="source")

install.packages("quantstrat",
repos="http://R-Forge.R-project.org",
type="source")

install.packages("PortfolioAnalytics")
```



quantstrat 安装後檢查



測試結果

寫一段直接到**git_hub**載入程式碼的程式碼：

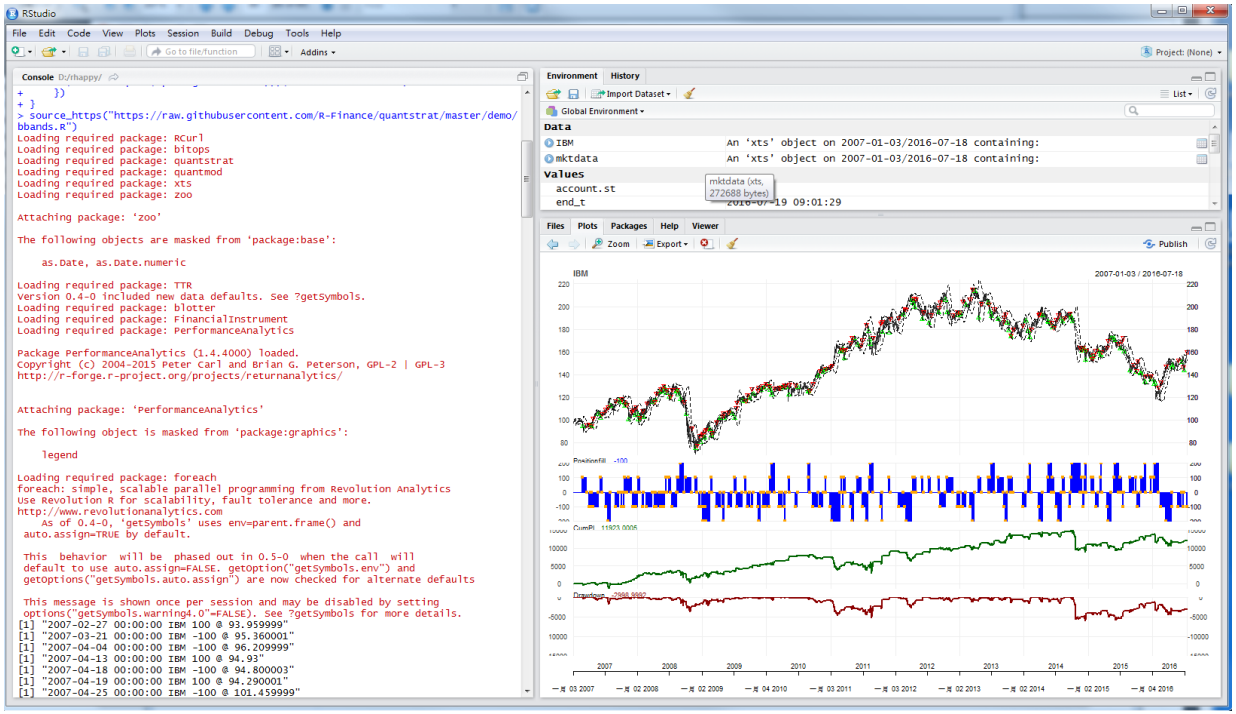
```
install.package("RCurl")

source_https <- function(url, ...) {
  # load package
  require(RCurl)

  # parse and evaluate each .R script
  sapply(c(url, ...), function(u) {
    eval(parse(text = getURL(u, followlocation = TRUE, cainfo = system.file("CurlSSL",
"cacert.pem", package = "RCurl"))), envir = .GlobalEnv)
  })
}
```

(目錄：證基會檔案\程式碼\source_https.R)

直接載入github內quanstrat的範例程式碼

```
source_https("https://raw.githubusercontent.com/R-Finance/quantstrat/master/demo/bbands.R")
```

bbands.R
為quantstrat 在 Github 上的一個布林通道策略範例

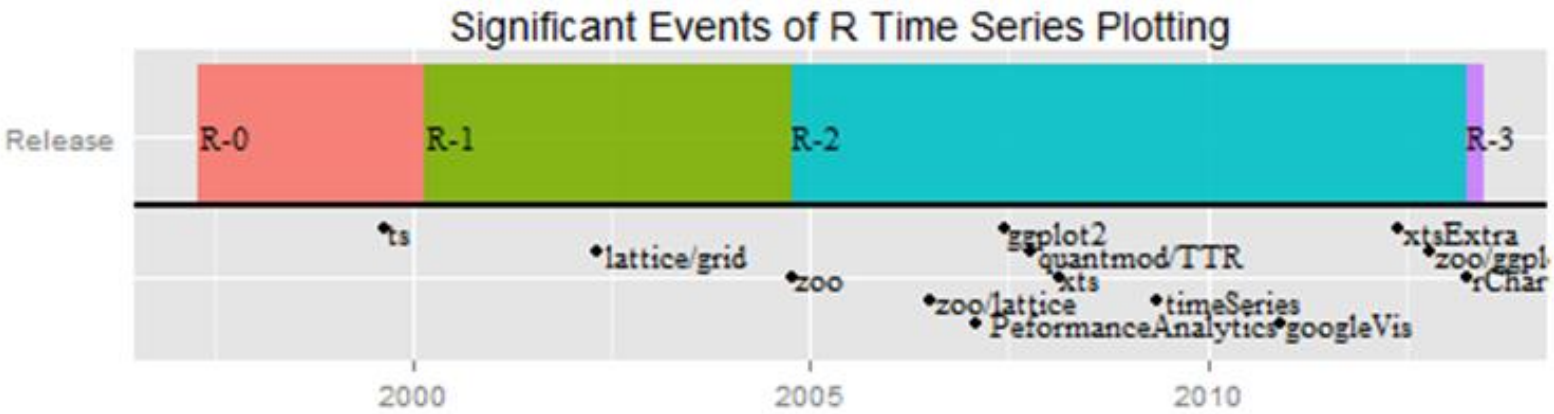


Time series資料介紹

R 時間序列套件演化時間表

- 財務資料 = 時間序列為索引資料

相關套件版本演化進程



R 時間序列套件與資料物件

Time Series Object	Package	Description
fts	**fts**	An R interfact to tslib (a time series library in C++)
its	**its**	An S4 class for handling irregular time series
irts	**tseries**	irts objects are irregular time-series objects. These are scalar or vector valued time series indexed by a time-stamp of class "POSIXct".
timeSeries	**timeSeries**	**Rmetrics** package of time series tools and utilities. Similar to the Tibco S-PLUS timeSeries class
ti	**tis**	Functions and S3 classes for time indexes and time indexed series, which are compatible with FAME frequencies
ts, mts	**stats**	Regularly spaced time series objects
zoo	**zoo**	S3 class of indexed totally ordered observations which includes irregular time series.
xts	**xts**	Extension of the zoo class

範例資料準備

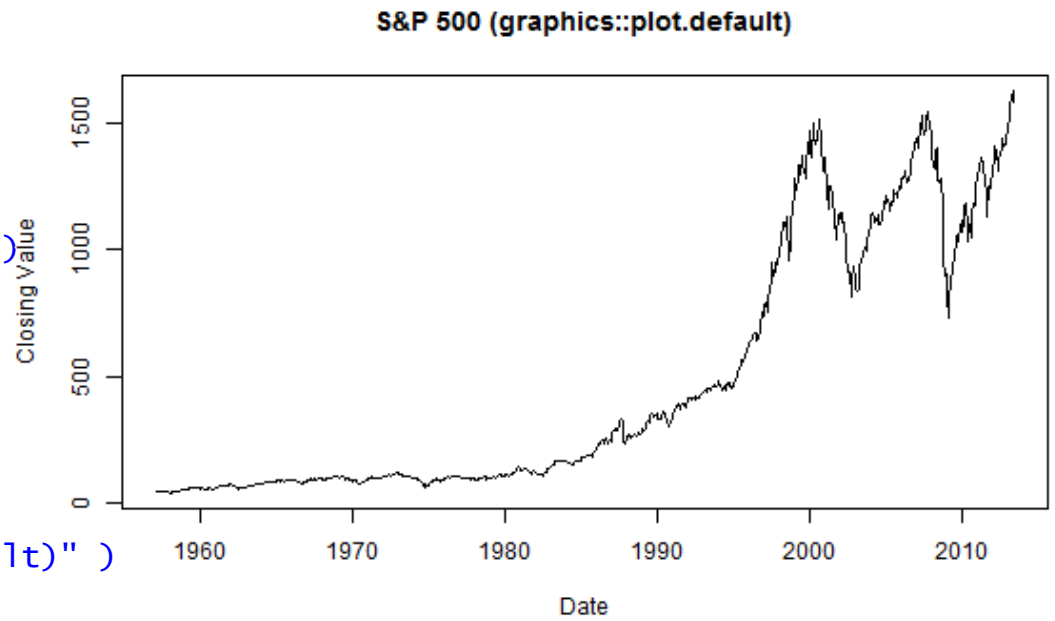
```
require(quantmod)

sp500 <- na.omit(
  getSymbols("^GSPC",
    from = "1949-12-31",
    auto.assign = FALSE)
)

sp500.monthly <- sp500[endpoints(
  sp500, on ="months")]
```

	GSPC.Open	GSPC.High	GSPC.Low	GSPC.Close	GSPC.Volume	GSPC.Adjusted
1950-01-03	16.66	16.66	16.66	16.66	1260000	16.66
1950-01-04	16.85	16.85	16.85	16.85	1890000	16.85
1950-01-05	16.93	16.93	16.93	16.93	2550000	16.93
1950-01-06	16.98	16.98	16.98	16.98	2010000	16.98
1950-01-09	17.08	17.08	17.08	17.08	2520000	17.08
1950-01-10	17.03	17.03	17.03	17.03	2160000	17.03
1950-01-11	17.09	17.09	17.09	17.09	2630000	17.09
1950-01-12	16.76	16.76	16.76	16.76	2970000	16.76
1950-01-13	16.67	16.67	16.67	16.67	3330000	16.67
1950-01-16	16.72	16.72	16.72	16.72	1460000	16.72
1950-01-17	16.86	16.86	16.86	16.86	1790000	16.86
1950-01-18	16.85	16.85	16.85	16.85	1570000	16.85

Showing 1 to 12 of 16,787 entries

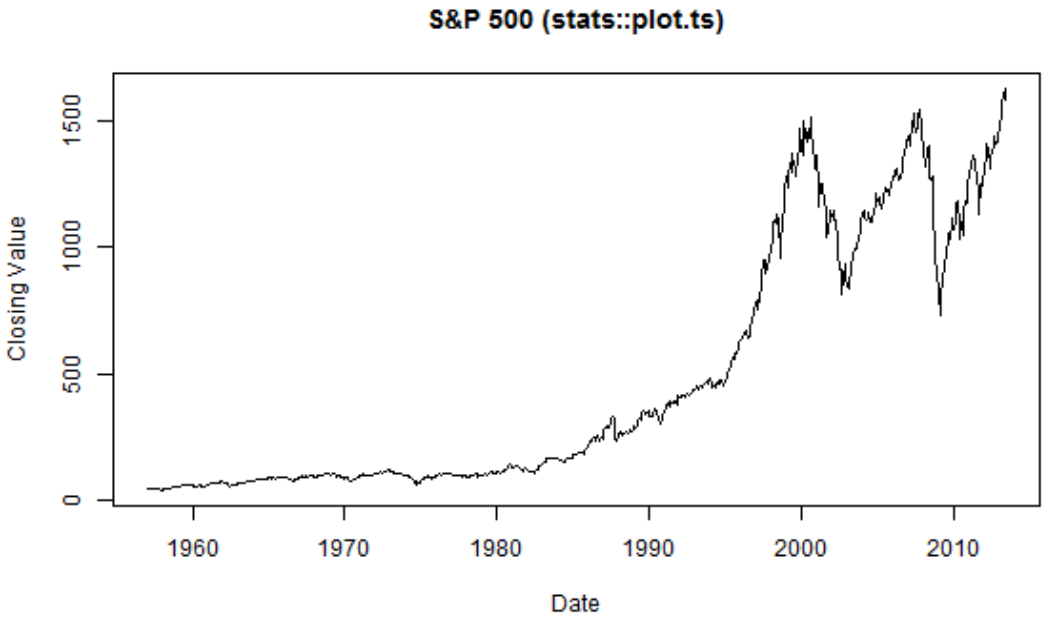
plot.default（R 內建函數）

```
sp500.df <- data.frame(
  index(sp500.monthly),
  coredata(sp500.monthly),
  stringsAsFactors=FALSE )

colnames(sp500.df ) <- c("date","sp500")

graphics::plot.default(
  x = sp500.df$date,
  y = sp500.df$sp500,
  type = "l",
  xlab = "Date",
  ylab = "Closing Value",
  main = "S&P 500 (graphics::plot.default)" )
```



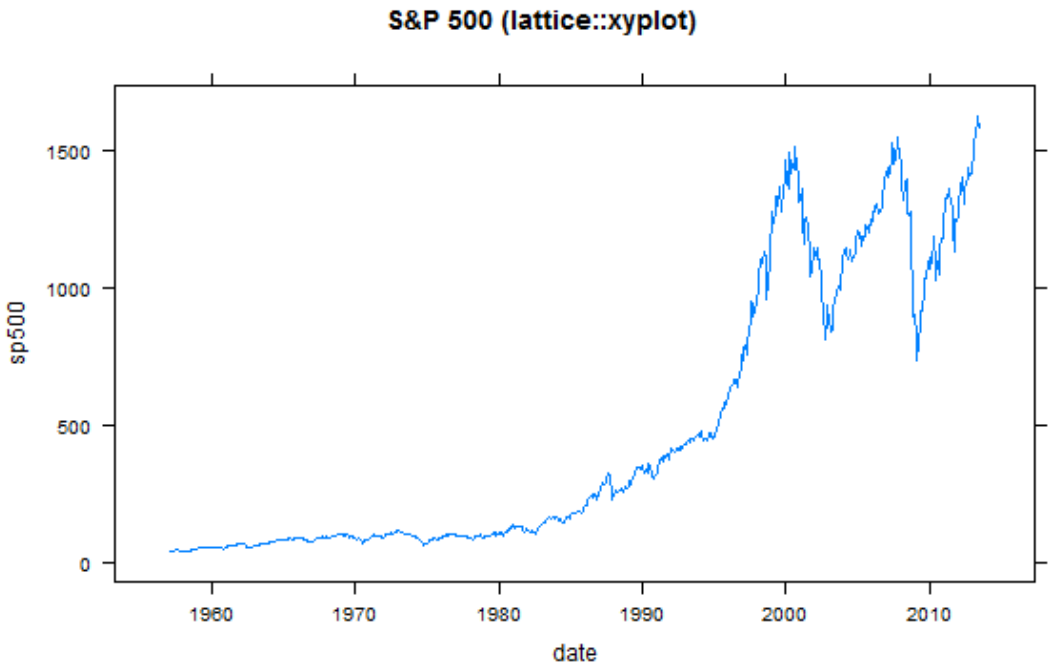
ts 1999-08

```
stats::plot.ts(
  ts(Cl(sp500.monthly),
     start = c(as.numeric(
    format(index(sp500.monthly)[1],"%Y")),
as.numeric(
    format(index(sp500.monthly)[1],"%m"))
     ),
     frequency = 12
  ),
  xlab = "Date",
  ylab = "Closing Value",
  main = "S&P 500 (stats::plot.ts)"
)
```



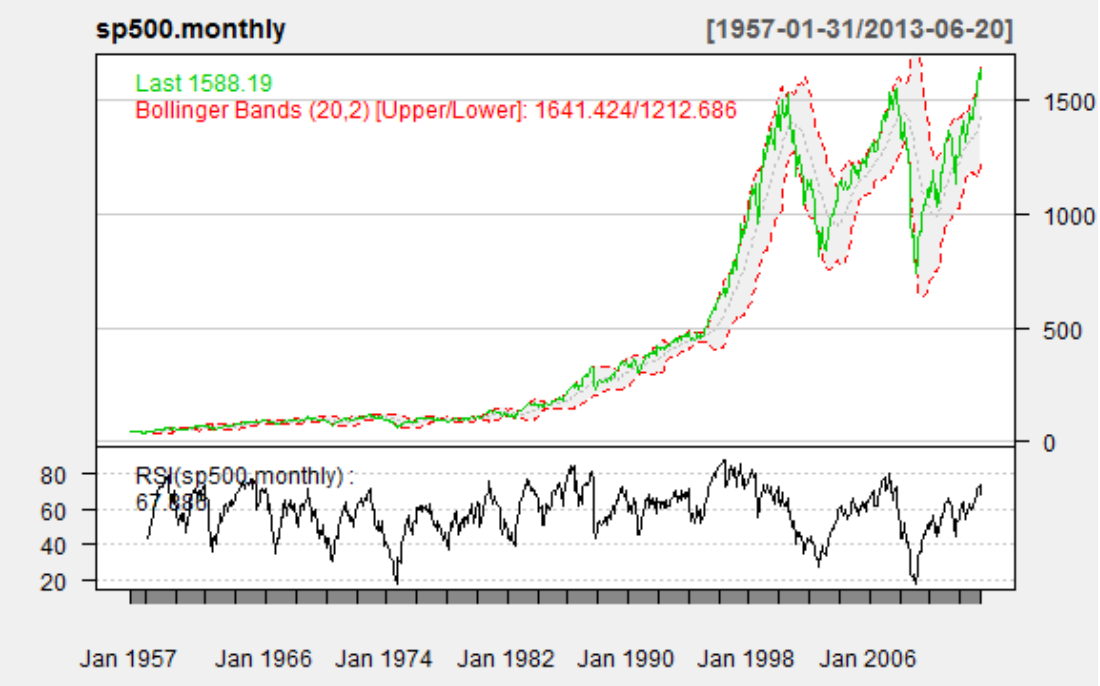
lattice and grid 2002-04

```
require(lattice)

lattice::xyplot( sp500 ~ date,
data = sp500.df,
type = "l",
main = "S&P 500 (lattice::xyplot)"
)
```



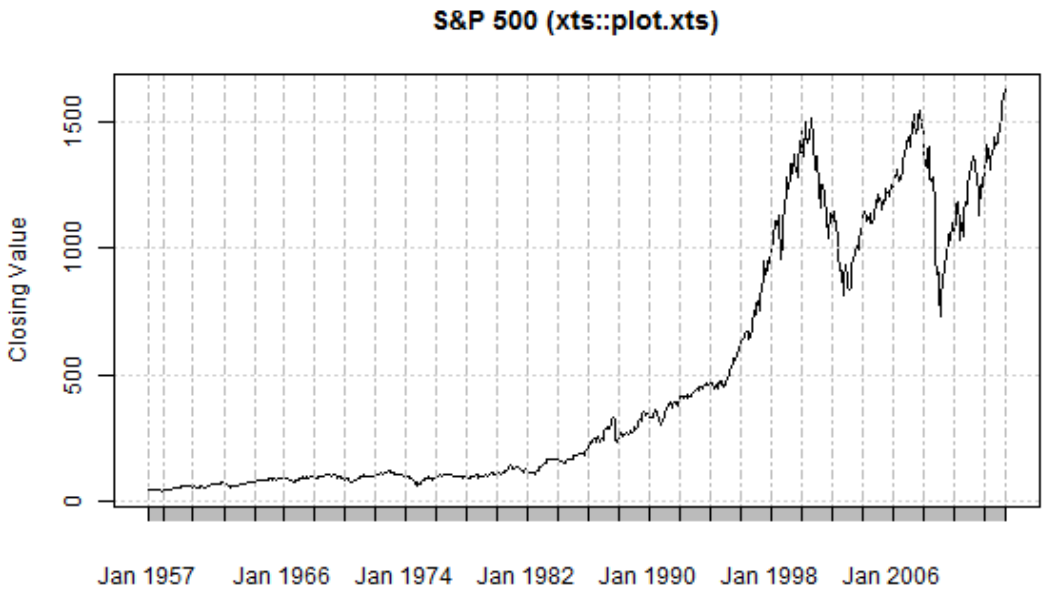
quantmod/TTR chartSeries 2007-10

```
require(quantmod)

quantmod::chartSeries(
sp500.monthly,
theme = chartTheme("white"),
TA = c(addBBands(),
addTA(RSI(sp500.monthly))
```



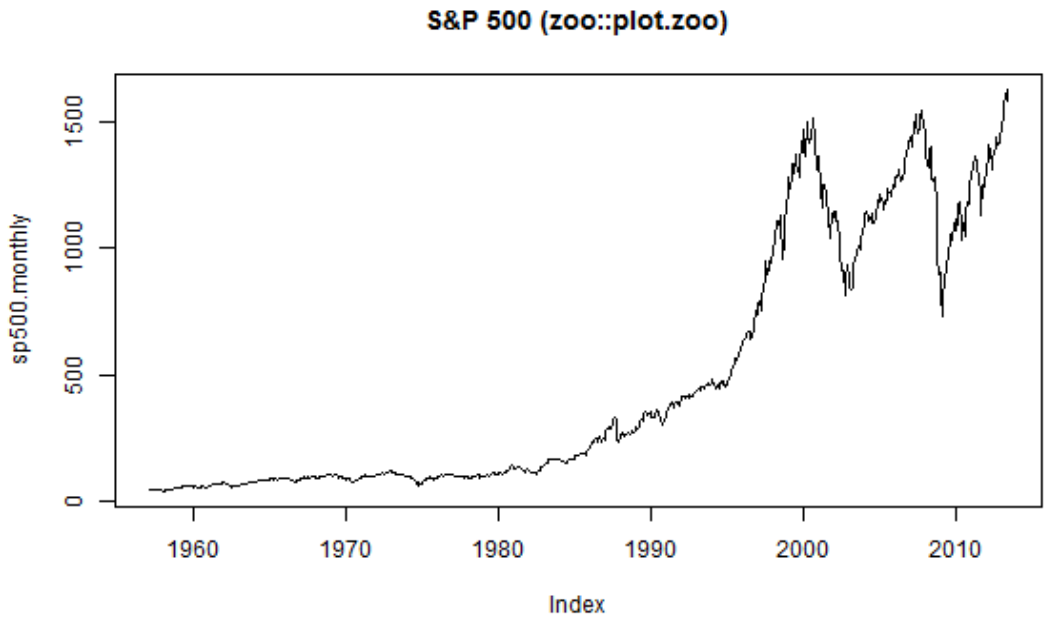
xts plot.xts 2008-02

```
xts::plot.xts( sp500.monthly,
  ylab = "Closing Value",
  main = "S&P 500 (xts::plot.xts)"
)
```



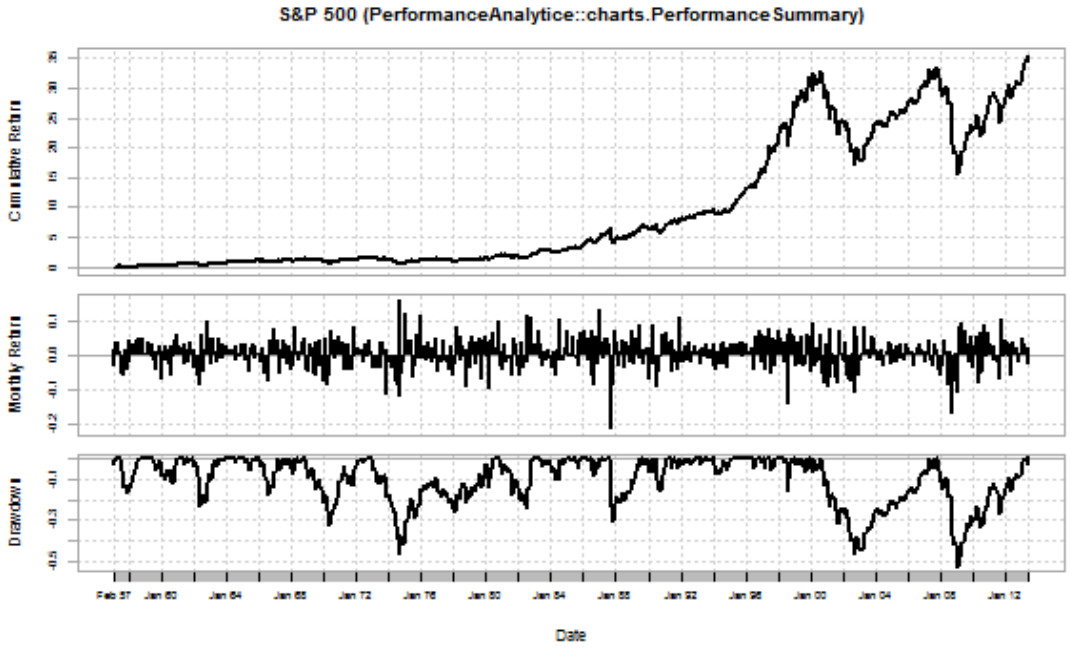
timeSeries plot 2009-05

```
require(timeSeries)

timeSeries::plot(
timeSeries(sp500.monthly$GSPC.Close),
  main = "S&P 500
(timeseries::plot)")
```



PerformanceAnalytics 2007-02

```
PerformanceAnalytics::
charts.PerformanceSummary(
ROC(sp500.monthly,
  n = 1,
  type = "discrete"),
  main = "S&P 500
(PerformanceAnalytics::charts.Perfor
manceSummary)")
```



xts 時間序列物件操作

- 資料準備

```
require(xts)
data(sample_matrix)

class(sample_matrix)
"matrix“

str(sample_matrix)
 num [1:180, 1:4] 50 50.2 50.4 50.4 50.2 ...
  attr(*, "dimnames")=List of 2
  ..$ : chr [1:180] "2007-01-02" "2007-01-03" "2007-01-04" "2007-01-05" ...
  ..$ : chr [1:4] "Open" "High" "Low" "Close“

matrix_xts <- as.xts(sample_matrix,dateFormat='Date')
str(matrix_xts)
 An ‘xts’ object on 2007-01-02/2007-06-30 containing:
  Data: num [1:180, 1:4] 50 50.2 50.4 50.4 50.2 ...
  attr(*, "dimnames")=List of 2 ..$ : NULL
  ..$ : chr [1:4] "Open" "High" "Low" "Close"
  Indexed by objects of class: [Date] TZ: UTC xts Attributes: NULL
```

xts 時間序列物件操作

- 時間指定索引

```
xts(1:10, Sys.Date()+1:10)
```

```
           [,1]
2016-09-21    1
2016-09-22    2
2016-09-23    3
2016-09-24    4
2016-09-25    5
2016-09-26    6
2016-09-27    7
2016-09-28    8
2016-09-29    9
2016-09-30   10
```

```
head(matrix_xts['2007-03'],5)
```

```
             Open     High      Low    Close
2007-03-01 50.81620 50.81620 50.56451 50.57075
2007-03-02 50.60980 50.72061 50.50808 50.61559
2007-03-03 50.73241 50.73241 50.40929 50.41033
2007-03-04 50.39273 50.40881 50.24922 50.32636
2007-03-05 50.26501 50.34050 50.26501 50.29567
```

xts 時間序列物件操作

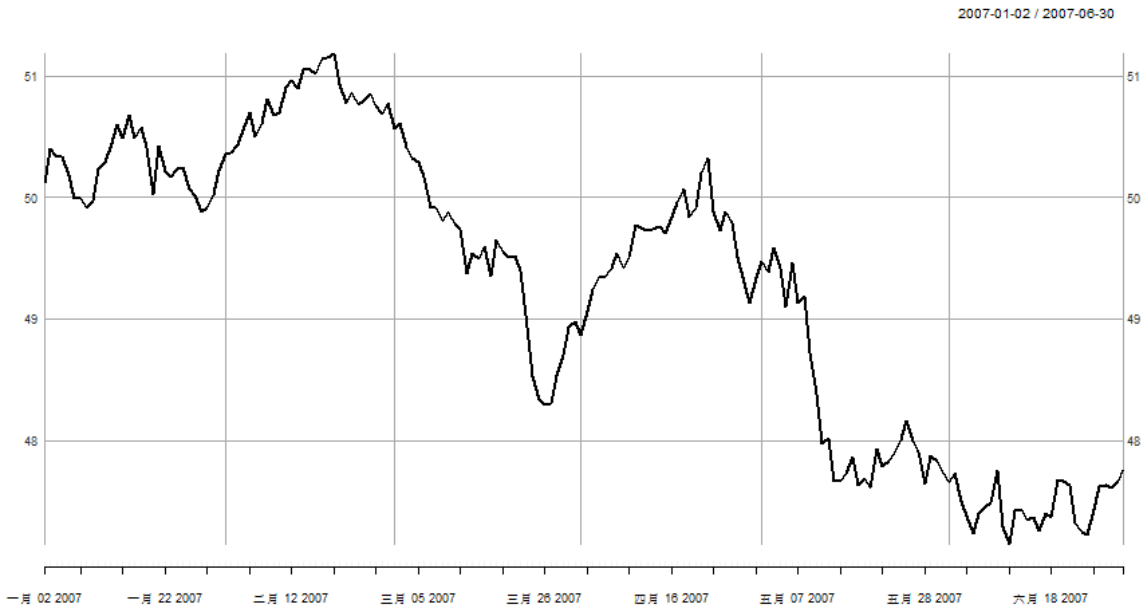
- 時間指定索引

```
first(last(matrix_xts,'1 week'),'3 days')
```

```
             Open     High      Low    Close
2007-06-25 47.20471 47.42772 47.13405 47.42772
2007-06-26 47.44300 47.61611 47.44300 47.61611
2007-06-27 47.62323 47.71673 47.60015 47.62769
```

- 畫收盤價圖表

```
plot(matrix_xts[,4],
  major.ticks='months',
  minor.ticks=FALSE,
  main=NULL,
col=1)
```



xts 時間序列物件操作

- 週期與時間區間索引

```
endpoints(matrix_xts, on='months')
```

```
[1]   0  30  58  89 119 150 180
```

```
endpoints(matrix_xts, on='weeks')
```

```
 [1]   0   6  13  20  27  34  41  48  55  62  69  76  83  90  97 104 111 118 125 132 139 146
[23] 153 160 167 174 180
```

```
to.period(matrix_xts,'months')
```

	matrix_xts.Open	matrix_xts.High	matrix_xts.Low	matrix_xts.Close
2007-01-31	50.03978	50.77336	49.76308	50.22578
2007-02-28	50.22448	51.32342	50.19101	50.77091
2007-03-31	50.81620	50.81620	48.23648	48.97490
2007-04-30	48.94407	50.33781	48.80962	49.33974
2007-05-31	49.34572	49.69097	47.51796	47.73780
2007-06-30	47.74432	47.94127	47.09144	47.76719

```
periodicity(to.period(matrix_xts,'months'))
```

```
Monthly periodicity from 2007-01-31 to 2007-06-30
```

xts 時間序列物件操作

- 時間週期apply執行

```
to.monthly(matrix_xts)
```

```
         matrix_xts.Open matrix_xts.High matrix_xts.Low matrix_xts.Close
一月 2007        50.03978        50.77336       49.76308         50.22578
二月 2007        50.22448        51.32342       50.19101         50.77091
三月 2007        50.81620        50.81620       48.23648         48.97490
四月 2007        48.94407        50.33781       48.80962         49.33974
五月 2007        49.34572        49.69097       47.51796         47.73780
六月 2007        47.74432        47.94127       47.09144         47.76719
```

```
period.apply(matrix_xts[,4],INDEX=endpoints(matrix_xts),FUN=max)
```

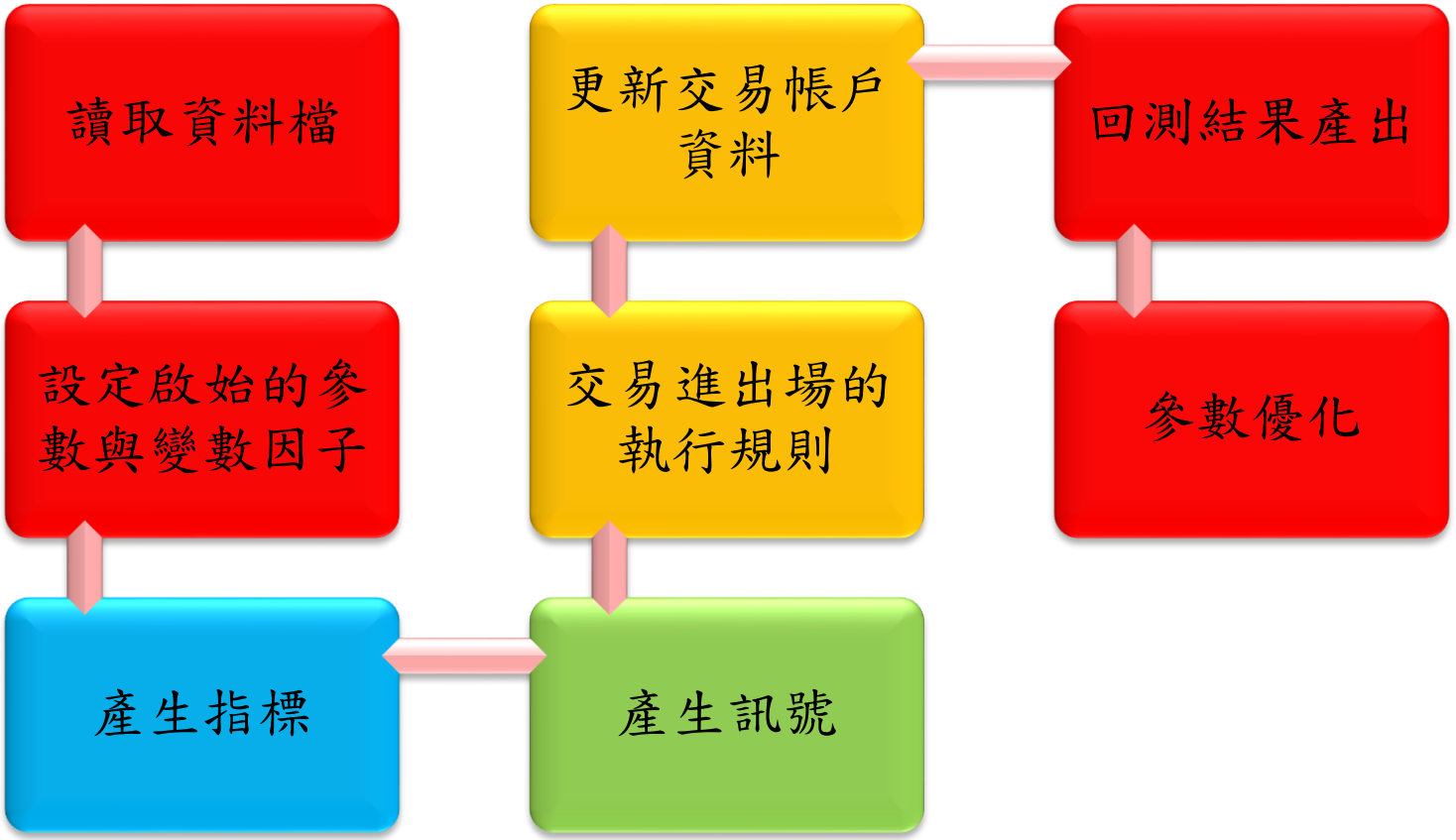
=

```
apply.monthly(matrix_xts[,4],FUN=max)
```

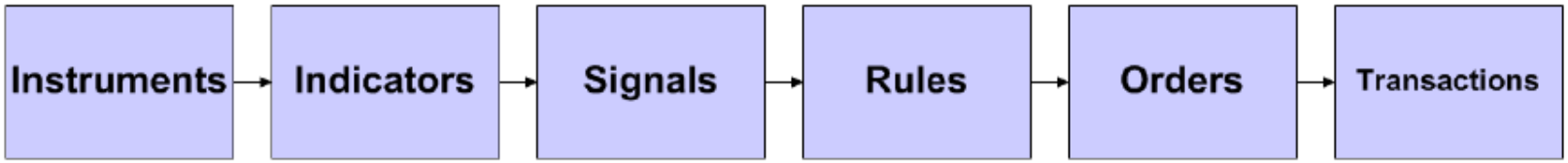
```
             Close
2007-01-31 50.67835
2007-02-28 51.17899
2007-03-31 50.61559
2007-04-30 50.32556
2007-05-31 49.58677
2007-06-30 47.76719
```

quantstrat交易策略開發詳解

quantstrat 策略回測流程



quantstrat的物件模型



- 以訊號為基礎(Signal-Based)的策略模型
  - Instruments 物件含有商品歷史資料
  - Indicators物件從商品歷史資料所衍生的量化計算值
  - 指標與市場資料間的互動產生Signals物件(例如，穿過價，某個閥值)
  - Rules物件經由商品資訊，訊號，指標判斷目前帳戶是否進行Order
  - 商品資訊與下單動作的互動，產生了Transaction

核心套件blotter主要使用函數

Initialization	
initPortf	initializes a portfolio object
initAcct	initializes an account object
Processing	
addTxn	add transactions to a portfolio
updatePortf	calculate P&L for each symbol for each period
updateAcct	calculate equity from portfolio data
updateEndEq	update ending equity for an account
getEndEq	retrieves the most recent value of the capital account
getPosQty	gets position at Date
Analysis	
chart.Posn	chart market data, position size, and cumulative P&L
PortfReturns	calculate portfolio instrument returns
getAccount	get an account object from the .blotter environment
getPortfolio	get a portfolio object from the .blotter environment
getTxns	retrieve transactions from a portfolio
tradeStats	calculate trade statistics
perTradeStats	calculate flat to flat per-trade statistics

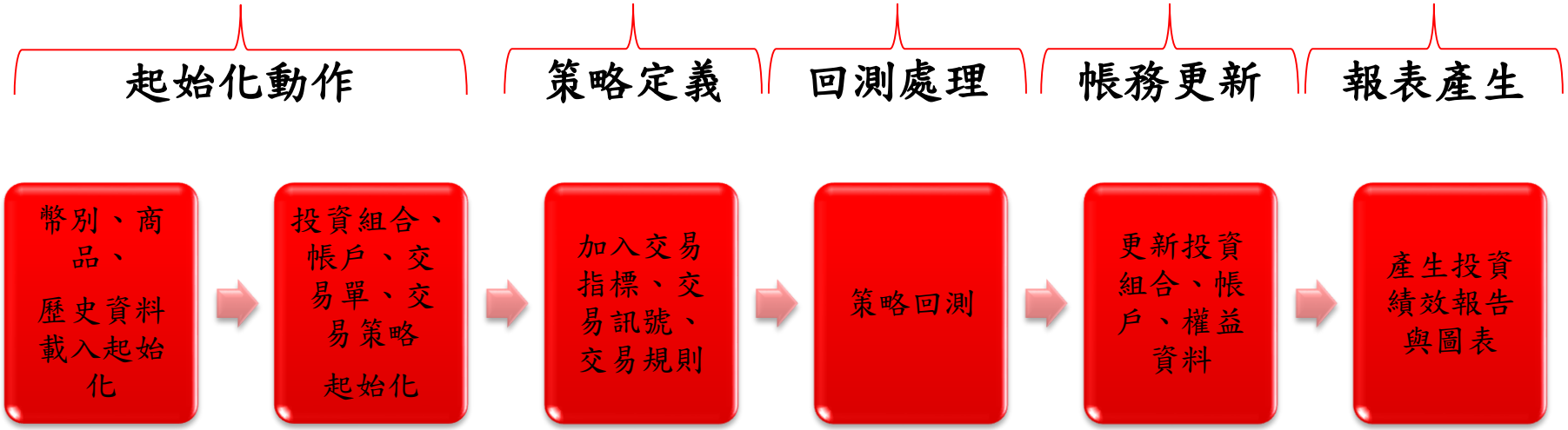
核心套件quantstrat主要使用函數

Initialization	
initOrders	initialize order container
strategy	constructor for strategy object
Strategy definition	
add.indicator	add an indicator to a strategy
add.signal	add a signal to a strategy
add.rule	add a rule to a strategy
add.distribution	add a distribution to a paramset in a strategy
add.constraint	add a constraint on 2 distributions within a paramset
Processing	
applyStrategy	apply the strategy to arbitrary market data
addPosLimit	add position and level limits at timestamp
apply.paramset	apply a paramset to the strategy
applyStrategy.rebalancing	apply the strategy to data with periodic rebalancing

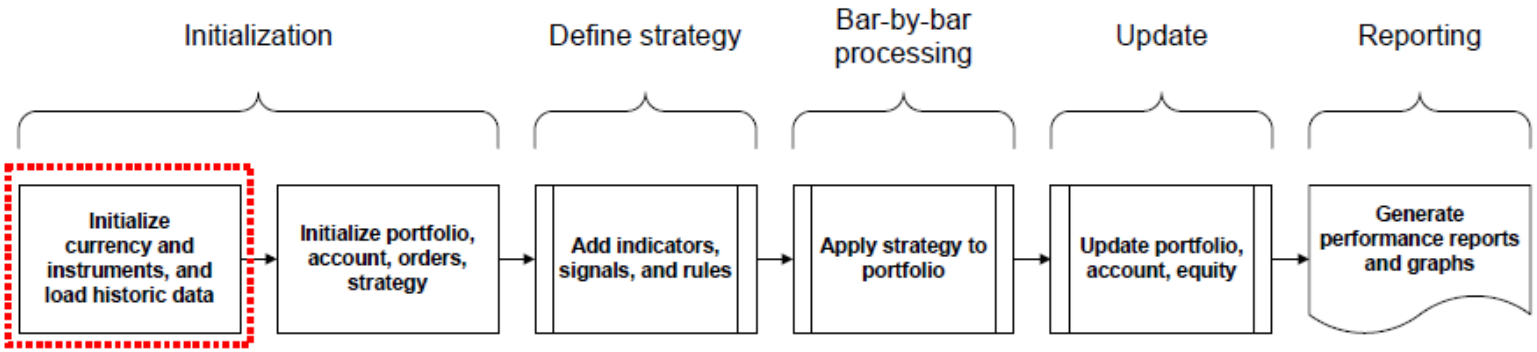
quantstrat 實作平均移動線策略

執行 (目錄: 證基會檔案\程式碼\maCross.R)

quantstrat 基本的回測流程

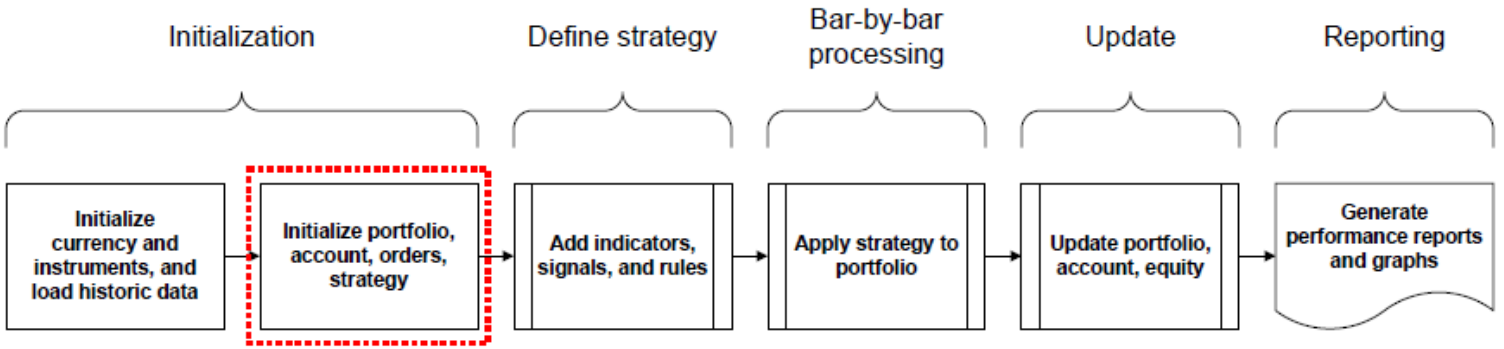


策略商品的啟始化



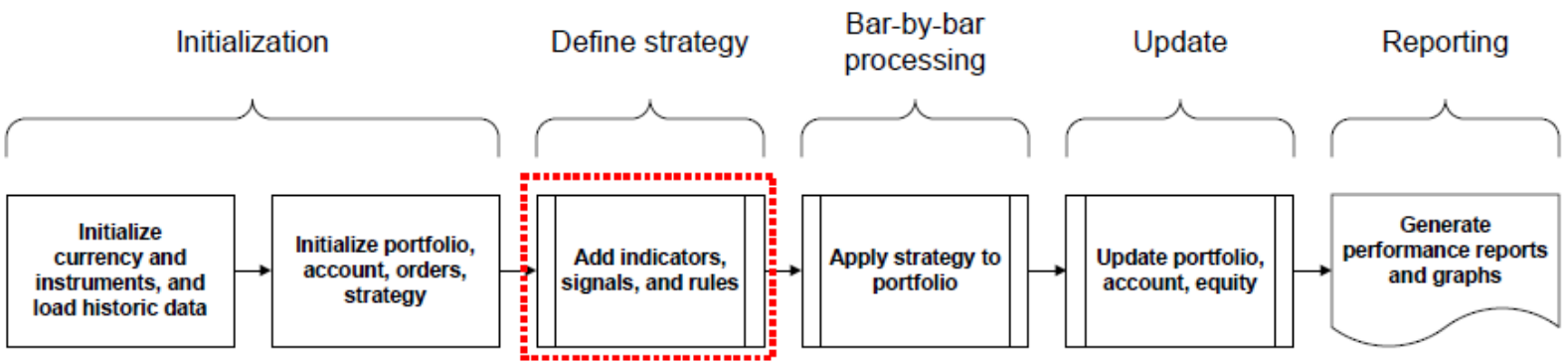

```
library(quantstrat)
ttz<-Sys.getenv('TZ')
Sys.setenv(TZ='UTC‘)

if (!exists('.blotter')) .blotter <- new.env()
if (!exists('.strategy')) .strategy <- new.env()

suppressWarnings(rm("order_book.macross",pos=.strategy))
suppressWarnings(rm("account.macross","portfolio.macross",pos=.blotter))
suppressWarnings(rm("account.st","portfolio.st","stock.str","stratMACROSS",'start_t','end_t')


stock.str='AAPL' # what are we trying it on
currency('USD')
stock(stock.str,currency='USD',multiplier=1)
startDate="2000-12-31“
initEq=1000000
```

策略的帳戶、下單、投資組合啟始化



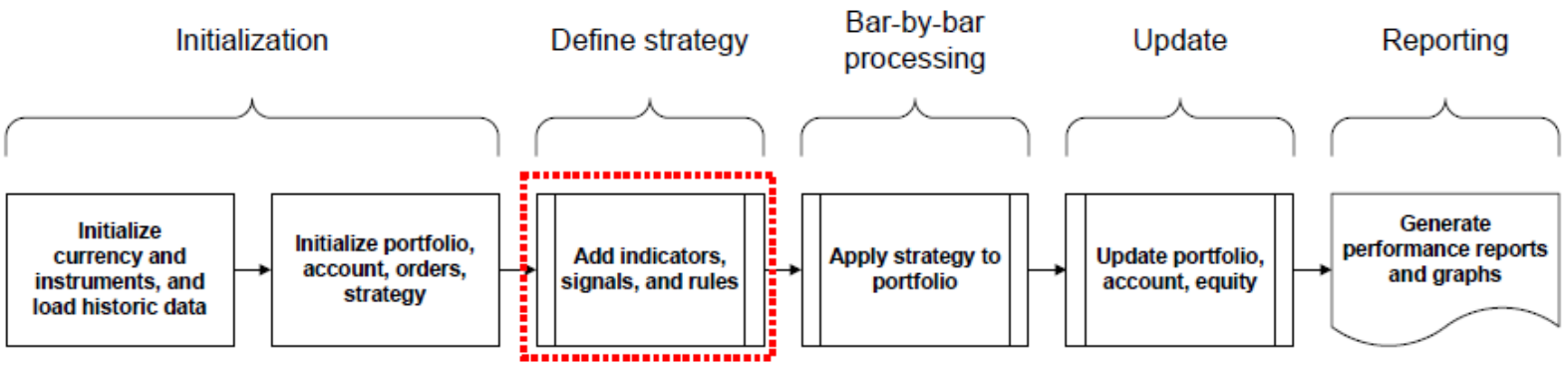

```
portfolio.st='macross'
account.st='macross‘
initPortf(portfolio.st,symbols=stock.str

initAcct(account.st,portfolios=portfolio.st, initEq=initEq)
initOrders(portfolio=portfolio.st)


stratMACROSS <- strategy(portfolio.st)
```

定義指標



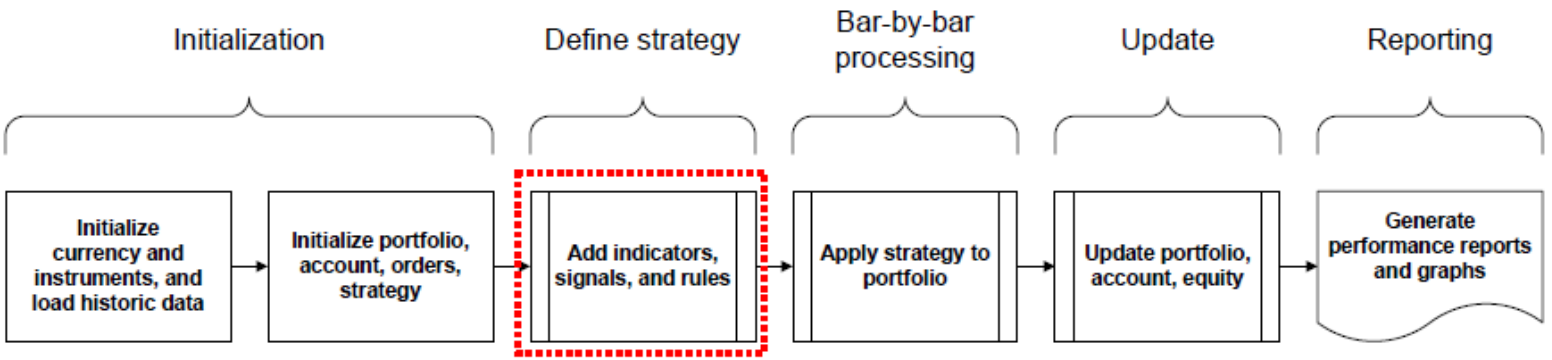

```
stratMACROSS <- add.indicator(
  strategy = stratMACROSS,
  name = "SMA",
  arguments = list(x=quote(Cl(mktdata)), n=50),
  label= "ma50" )

stratMACROSS <- add.indicator(
  strategy = stratMACROSS,
  name = "SMA",
  arguments = list(x=quote(Cl(mktdata)[,1]),
  n=200),label= "ma200")
```

定義訊號



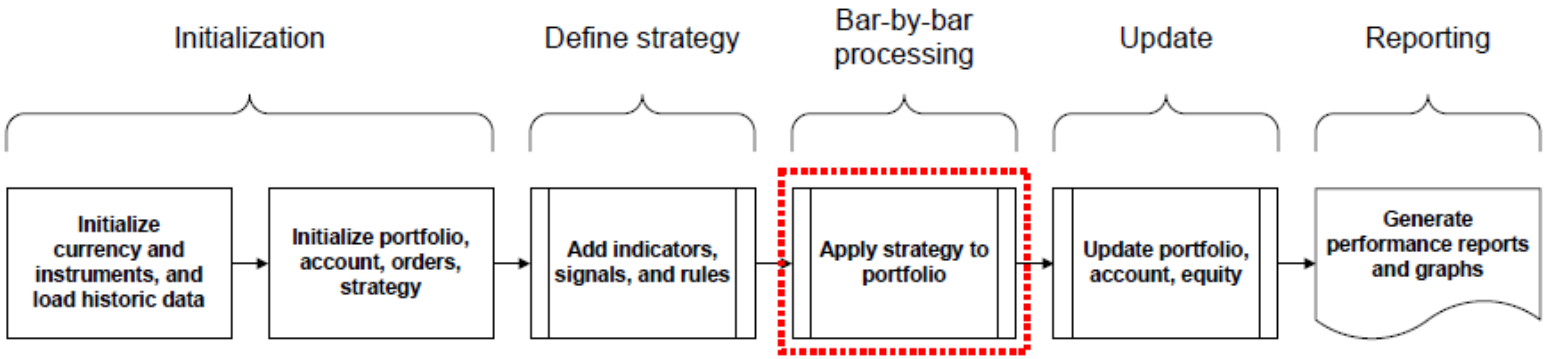

```
stratMACROSS <- add.signal(
        strategy = stratMACROSS,name="sigCrossover",
        arguments = list(columns=c("ma50","ma200"),
        relationship="gte"),
        label="ma50.gt.ma200")

stratMACROSS <- add.signal(
        strategy = stratMACROSS,name="sigCrossover",
        arguments = list(column=c("ma50","ma200"),
        relationship="lt"),
        label="ma50.lt.ma200")
```

訂定交易規則



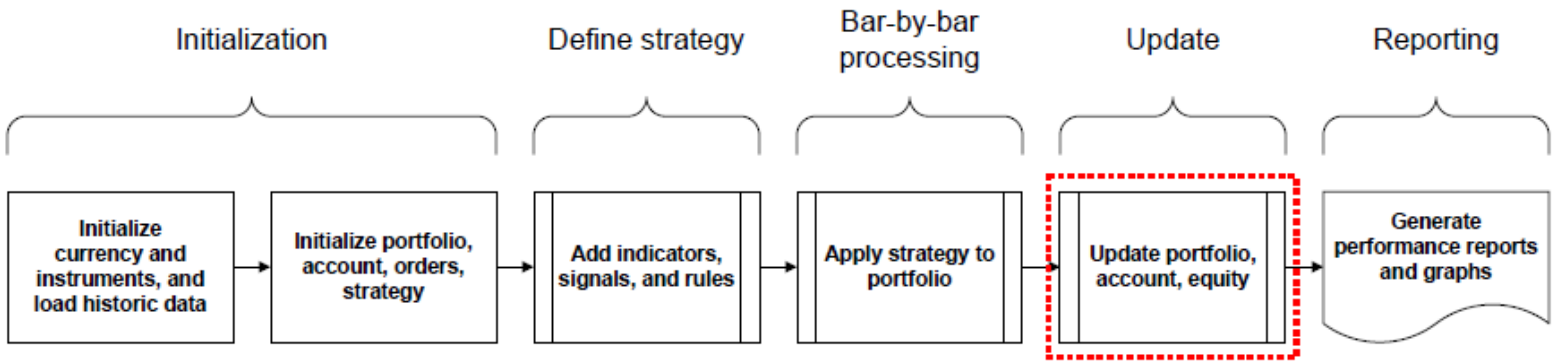

```
stratMACROSS <- add.rule(
  strategy = stratMACROSS,name='ruleSignal',
  arguments = list(sigcol="ma50.gt.ma200",
  sigval=TRUE,
  orderqty=100,
  ordertype='market',
  orderside='long'),
  type='enter')

stratMACROSS <- add.rule(
  strategy = stratMACROSS,name='ruleSignal',
  arguments = list(sigcol="ma50.lt.ma200",
  sigval=TRUE,
  orderqty='all',
  ordertype='market',
  orderside='long'),
  type='exit')
```

開始回測



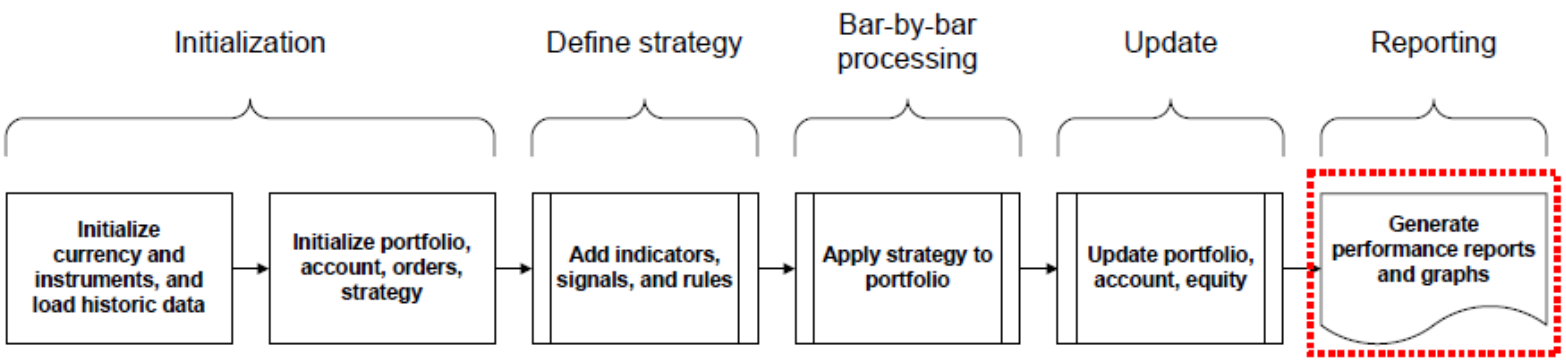

```
start_t<-Sys.time() > out<-applyStrategy(
  strategy=stratMACROSS , portfolios=portfolio.st)
[1] "2002-01-07 00:00:00 AAPL 100 @ 1.50565"
[1] "2003-05-16 00:00:00 AAPL 100 @ 1.23608"
[1] "2006-09-26 00:00:00 AAPL 100 @ 10.205548"
[1] "2008-05-19 00:00:00 AAPL 100 @ 24.143005"
[1] "2009-05-14 00:00:00 AAPL 100 @ 16.16766"
[1] "2013-09-11 00:00:00 AAPL 100 @ 63.238555"

end_t<-Sys.time() > print(end_t-start_t)
Time difference of 0.234004 secs
```

交易帳戶資料更新



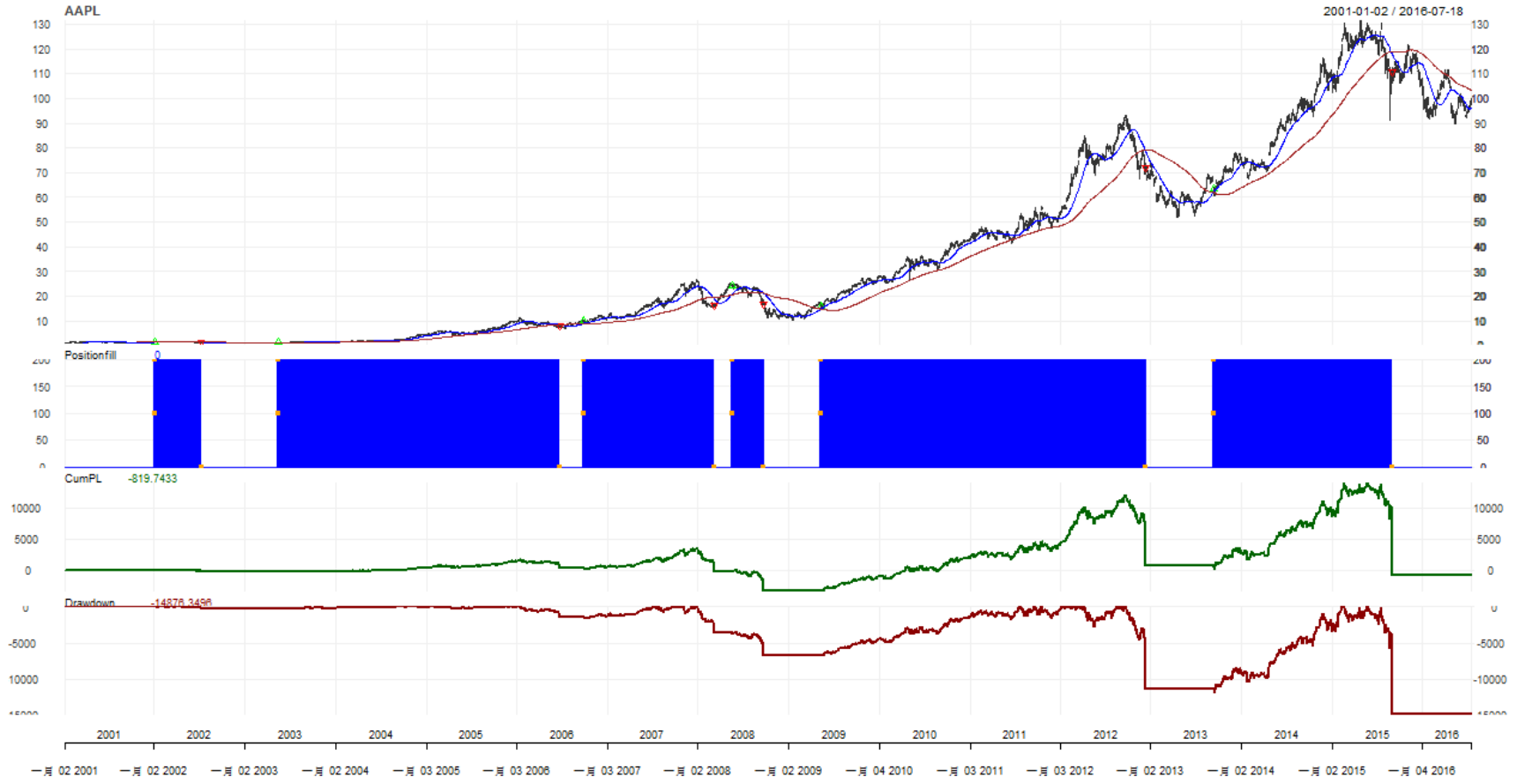

```
start_t<-Sys.time() > updatePortf(Portfolio='macross',
  Dates=paste('::',as.Date(Sys.time()),sep=''))
[1] "macross“
end_t<-Sys.time() > print("更新交易帳")
[1] "更新交易帳“
print(end_t-start_t)
Time difference of 0.07900095 secs
```

回測結果報表




```
chart.Posn(Portfolio='macross',Symbol=stock.str)

add_SMA(n=50 , on=1,col='blue')
add_SMA(n=200, on=1)
```

回測結果報表


AAPL
2001-01-02 / 2016-07-18
Positionfill
CumPL -819.7433
Drawdown -14876.3496
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
一月 02 2001
一月 02 2002
一月 02 2003
一月 02 2004
一月 03 2005
一月 03 2006
一月 03 2007
一月 02 2008
一月 02 2009
一月 04 2010
一月 03 2011
一月 03 2012
一月 02 2013
一月 02 2014
一月 02 2015
一月 04 2016

quantstrat 回測資料分析

PerformanceAnalytics 套件分析回測資料

- 資料準備

```
library('PerformanceAnalytics')

data(managers)
head(managers)
```

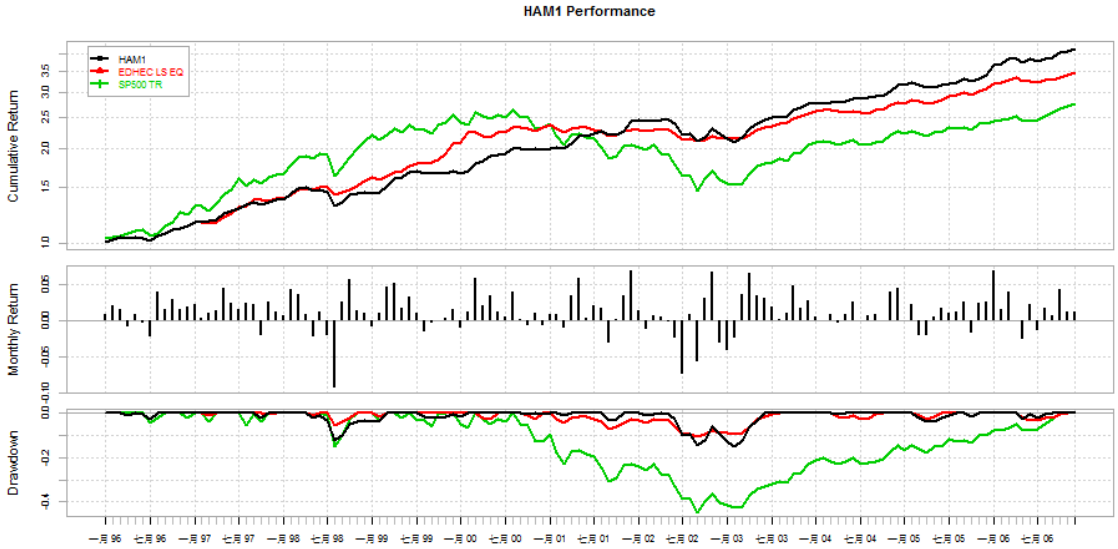
	Open	High	Low	Close
2007-01-02	50.03978	50.11778	49.95041	50.11778
2007-01-03	50.23050	50.42188	50.23050	50.39767
2007-01-04	50.42096	50.42096	50.26414	50.33236
2007-01-05	50.37347	50.37347	50.22103	50.33459
2007-01-06	50.24433	50.24433	50.11121	50.18112
2007-01-07	50.13211	50.21561	49.99185	49.99185
2007-01-08	50.03555	50.10363	49.96971	49.98806
2007-01-09	49.99489	49.99489	49.80454	49.91333
2007-01-10	49.91228	50.13053	49.91228	49.97246
2007-01-11	49.88529	50.23910	49.88529	50.23910
2007-01-12	50.21258	50.35980	50.17176	50.28519
2007-01-13	50.32385	50.48000	50.32385	50.41286

```
managers.length = dim(managers)[1]
colnames(managers)

 [1] "HAM1" "HAM2" "HAM3" "HAM4" "HAM5" "HAM6"
"EDHEC LS EQ" [8] "SP500 TR" "US 10Y TR" "US 3m TR“

manager.col = 1
peers.cols = c(2,3,4,5,6)
indexes.cols = c(7,8)
Rf.col = 10

trailing12.rows = ((managers.length - 11):managers.length)
trailing36.rows = ((managers.length - 35):managers.length)
trailing60.rows = ((managers.length - 59):managers.length)
```

PerformanceAnalytics 套件分析回測資料

- 績效摘要

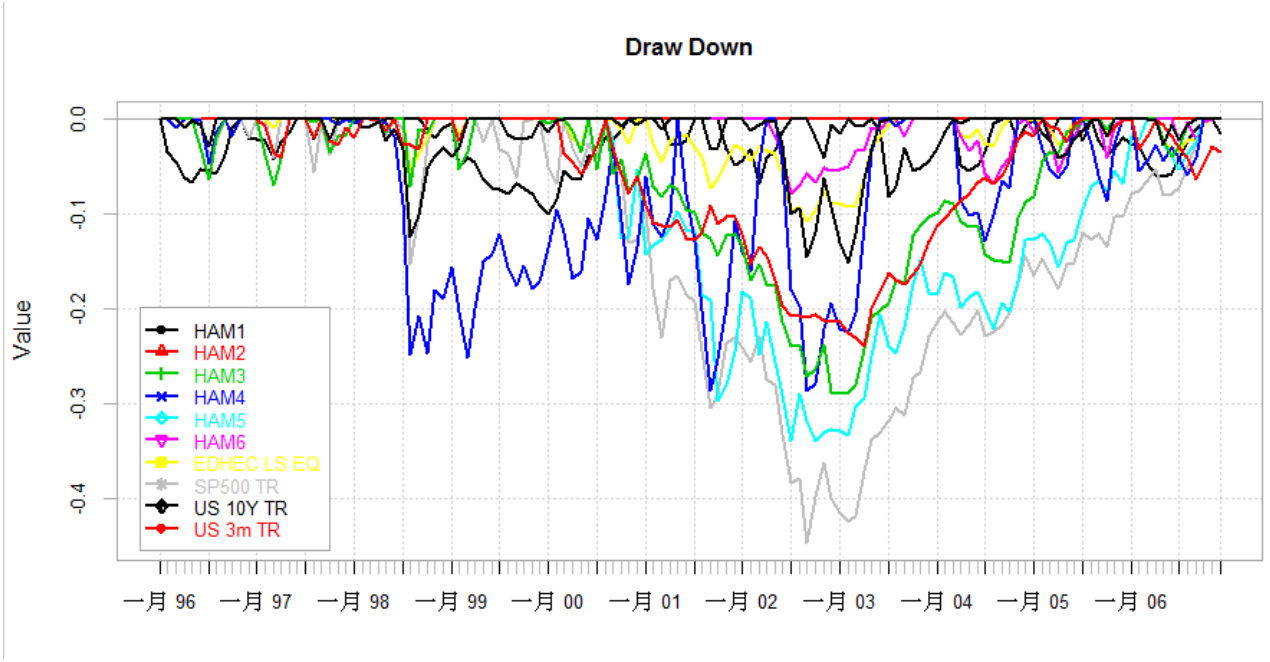
```
charts.PerformanceSummary(
  managers[,c(manager.col,indexes.cols)],
  lwd=2,
  ylog=TRUE)
```



PerformanceAnalytics 套件分析回測資料

- 權益虧損(Drawdown)

```
chart.Drawdown(managers,
               main="Draw Down",
              legend.loc="bottomleft")
```



PerformanceAnalytics 套件分析回測資料

- 逐月投資報酬率

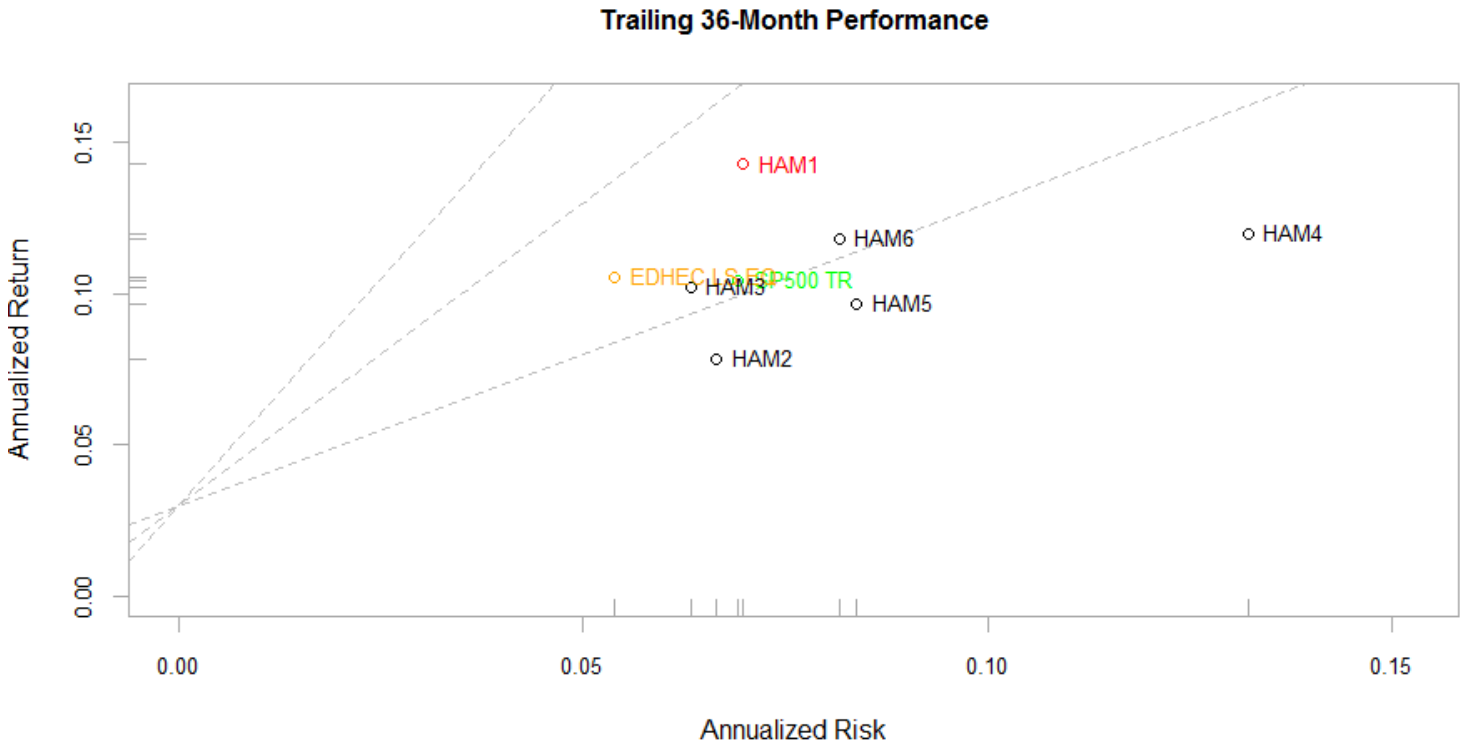
```
View(
    t(table.CalendarReturns(
    managers[,c(manager.col,indexes.cols)]
  )
 )
)
```

	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006
一月	0.7	2.1	0.6	-0.9	-1.0	0.8	1.4	-4.1	0.5	0.0	6.9
二月	1.9	0.2	4.3	0.9	1.2	0.8	-1.2	-2.5	0.0	2.1	1.5
三月	1.6	0.9	3.6	4.6	5.8	-1.1	0.6	3.6	0.9	-2.1	4.0
四月	-0.9	1.3	0.8	5.1	2.0	3.5	0.5	6.5	-0.4	-2.1	-0.1
五月	0.8	4.4	-2.3	1.6	3.4	5.8	-0.2	3.4	0.8	0.4	-2.7
六月	-0.4	2.3	1.2	3.3	1.2	0.2	-2.4	3.1	2.6	1.6	2.2
七月	-2.3	1.5	-2.1	1.0	0.5	2.1	-7.5	1.8	0.0	0.9	-1.4
八月	4.0	2.4	-9.4	-1.7	3.9	1.6	0.8	0.0	0.5	1.1	1.6
九月	1.5	2.2	2.5	-0.4	0.1	-3.1	-5.8	0.9	0.9	2.6	0.7
十月	2.9	-2.1	5.6	-0.1	-0.8	0.1	3.0	4.8	-0.1	-1.9	4.3
十一月	1.6	2.5	1.3	0.4	1.0	3.4	6.6	1.7	3.9	2.3	1.2
十二月	1.8	1.1	1.0	1.5	-0.7	6.8	-3.2	2.8	4.4	2.6	1.1
HAM1	13.6	20.4	6.1	16.1	17.7	22.4	-8.0	23.7	14.9	7.8	20.5
EDHEC LS EQ	NA	21.4	14.6	31.4	12.0	-1.2	-6.4	19.3	8.6	11.3	11.7
SP500 TR	23.0	33.4	28.6	21.0	-9.1	-11.9	-22.1	28.7	10.9	4.9	15.8

PerformanceAnalytics 套件分析回測資料

- 年化報酬落點圖

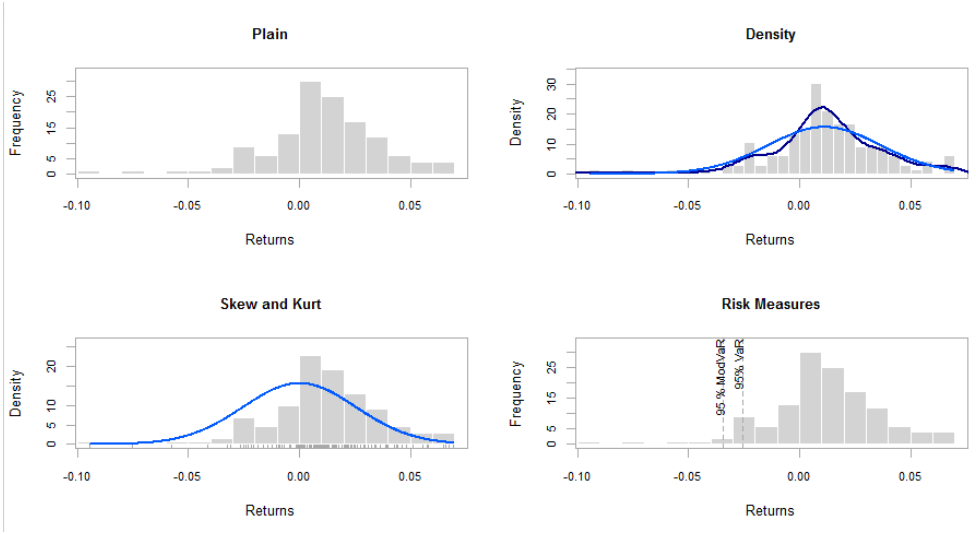
```
chart.RiskReturnScatter(managers[trailing36.rows,1:8],
  Rf=.03/12,
  main = "Trailing 36-Month Performance",
  colorset=c("red", rep("black",5), "orange", "green"))
```



PerformanceAnalytics 套件分析回測資料

- 獲利之機率動差與分佈圖

```
layout(rbind(c(1,2),c(3,4)))
chart.Histogram(managers[,1,drop=F],
main = "Plain", methods = NULL)

chart.Histogram(managers[,1,drop=F],
main = "Density", breaks=40,
methods = c("add.density",
          "add.normal"))

chart.Histogram(managers[,1,drop=F],
main = "Skew and Kurt",
methods = c("add.centered",
          "add.rug"))

chart.Histogram(managers[,1,drop=F],
main = "Risk Measures",
methods = c("add.risk"))
```



PerformanceAnalytics 套件分析回測資料

- 資本資產訂價模式(CAPM)統計量值

```
View(
  table.CAPM(
    managers[trailing36.rows, c(manager.col, peers.cols)],
    managers[ trailing36.rows, 8, drop=FALSE],
    Rf = managers[ trailing36.rows, Rf.col, drop=FALSE])
)
```

	HAM1 to SP500 TR	HAM2 to SP500 TR	HAM3 to SP500 TR	HAM4 to SP500 TR	HAM5 to SP500 TR	HAM6 to SP500 TR
Alpha	0.0051	0.0020	0.0020	0.0009	0.0002	0.0022
Beta	0.6267	0.3223	0.6320	1.1282	0.8755	0.8150
Beta+	0.8227	0.4176	0.8240	1.8430	1.0985	0.9993
Beta-	1.1218	-0.0483	0.8291	1.2223	0.5283	1.1320
R-squared	0.3829	0.1073	0.4812	0.3444	0.5209	0.4757
Annualized Alpha	0.0631	0.0247	0.0243	0.0109	0.0030	0.0271
Correlation	0.6188	0.3276	0.6937	0.5868	0.7218	0.6897
Correlation p-value	0.0001	0.0511	0.0000	0.0002	0.0000	0.0000
Tracking Error	0.0604	0.0790	0.0517	0.1073	0.0583	0.0601
Active Premium	0.0384	-0.0260	-0.0022	0.0154	-0.0077	0.0138
Information Ratio	0.6363	-0.3295	-0.0428	0.1433	-0.1319	0.2296
Treynor Ratio	0.1741	0.1437	0.1101	0.0768	0.0734	0.1045

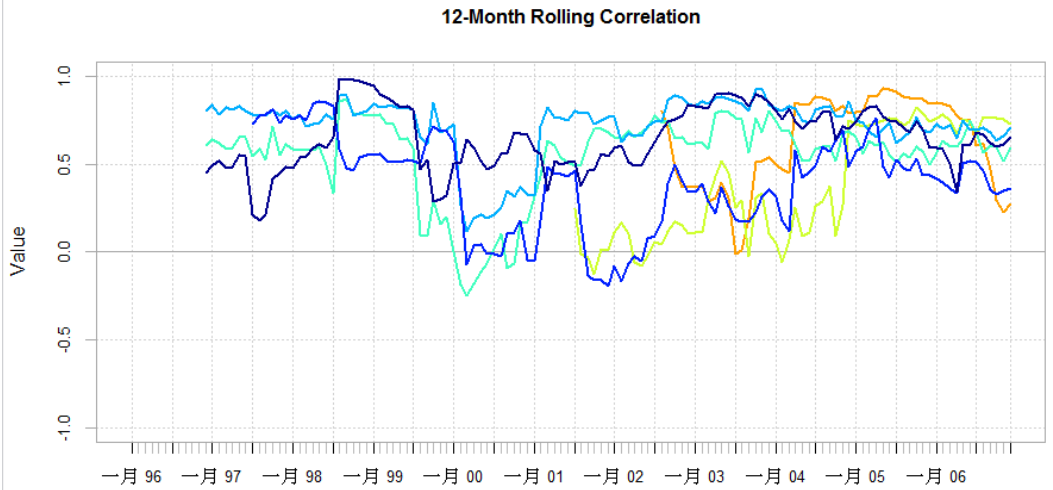
PerformanceAnalytics 套件分析回測資料

- 報酬率相關性統計

```
table.Correlation(
  managers[, c(manager.col, peers.cols)],
  managers[, 8, drop = F],
  legend.loc = "lowerleft")
```

	Correlation	p-value	Lower CI	Upper CI
HAM1 to SP500 TR	0.6600671	7.397842e-18	0.55138376	0.7467191
HAM2 to SP500 TR	0.4128282	1.715350e-06	0.25576240	0.5486602
HAM3 to SP500 TR	0.6608633	6.545409e-18	0.55236590	0.7473433
HAM4 to SP500 TR	0.5601846	2.870109e-12	0.43052170	0.6671932
HAM5 to SP500 TR	0.2844487	1.216830e-02	0.06458459	0.4779755
HAM6 to SP500 TR	0.5091542	1.735968e-05	0.30101889	0.6709863

```
chart.RollingCorrelation(
  managers[,c(manager.col, peers.cols)],
  managers[, 8, drop = FALSE],
  colorset = tim8equal,
  lwd = 2,
  main = "12-Month Rolling Correlation")
```



前推移動式分析(Walk Forward Analysis)

傳統的參數優化法

樣本內訓練資料用於優化策略參數

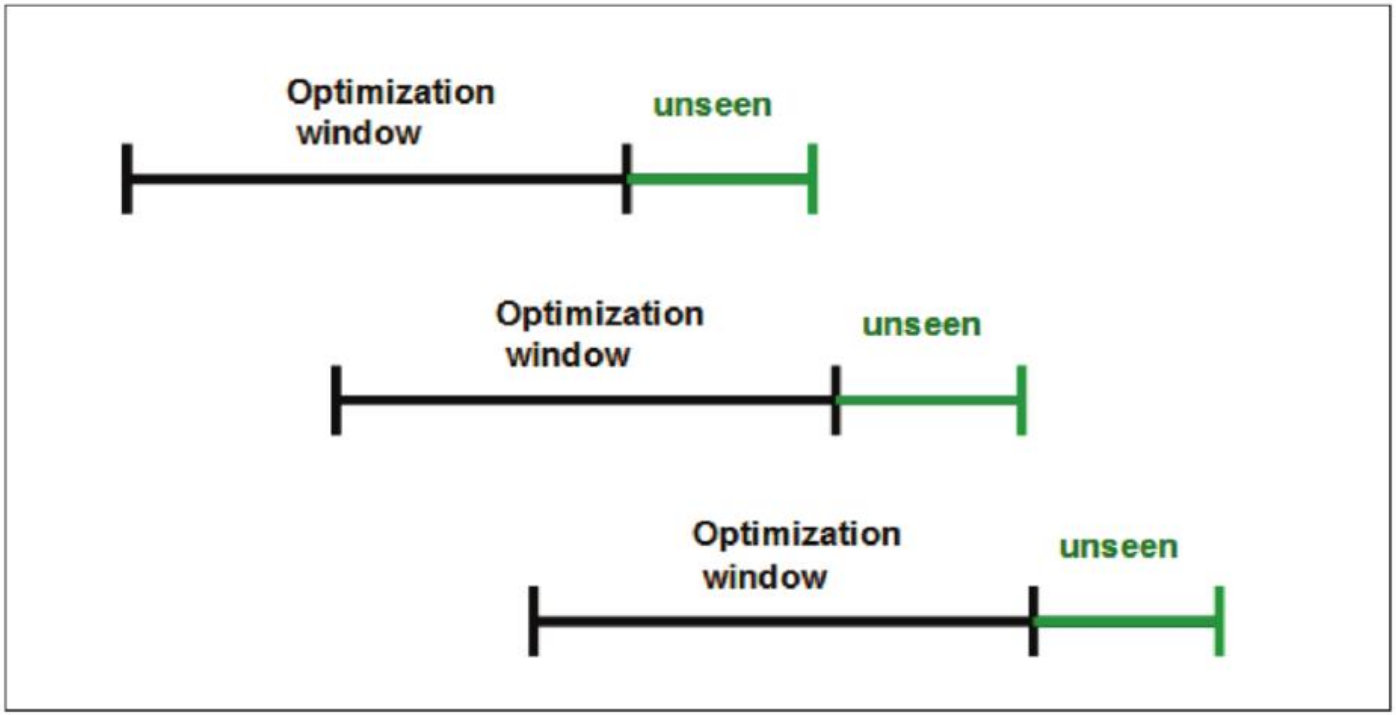
樣本外測試資料用於評估策略效能

WFA(前推移動式分析)

- WFA設計來樣本外測試時週期性的重複優化參數
  - 滾動訓練窗(固定長度遞移)
  - 定錨訓練窗(固定開始點)
- 有效利用有限樣本資料
- 讓參數能隨著市場狀態調整

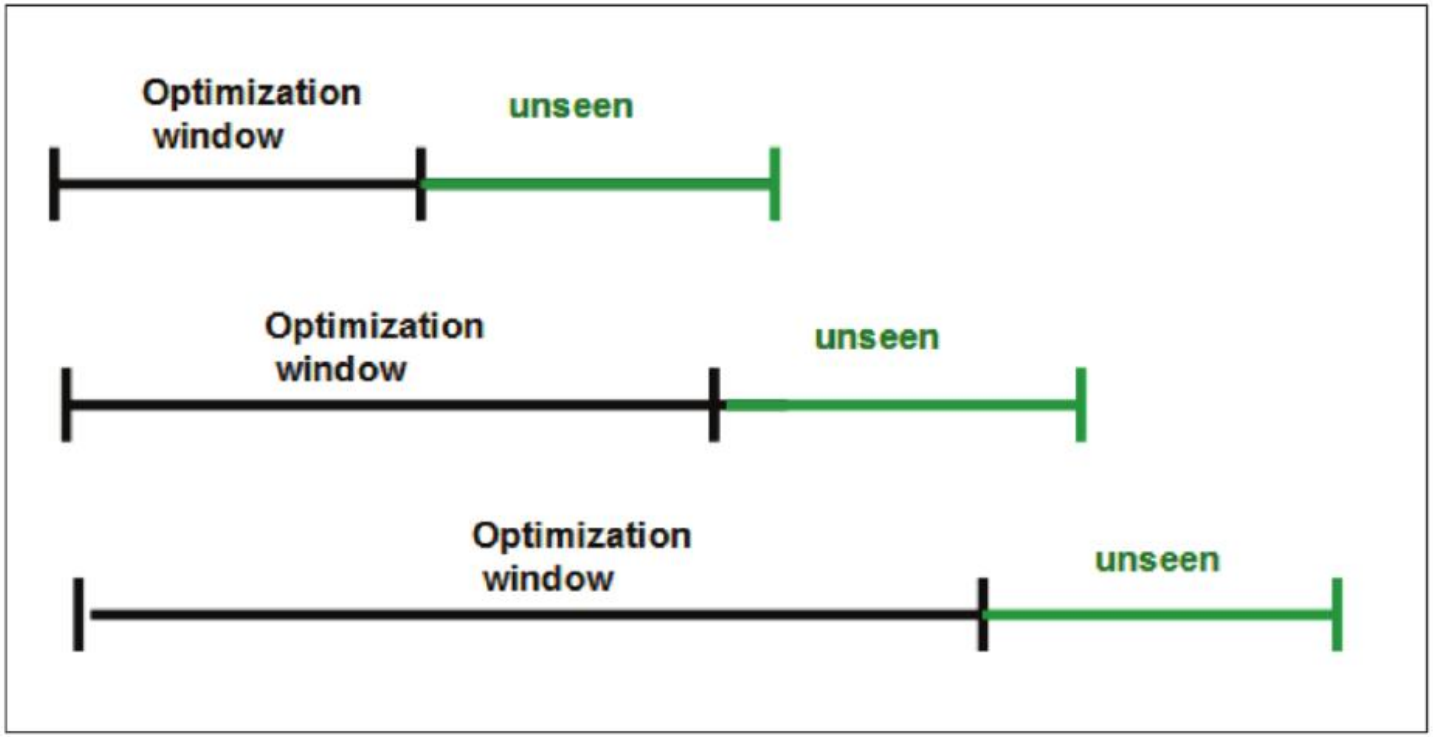
WFA(前推移動式分析法)

滾動訓練窗



WFA(前推移動式分析法)

定錨訓練窗



台股計量策略WFA實作

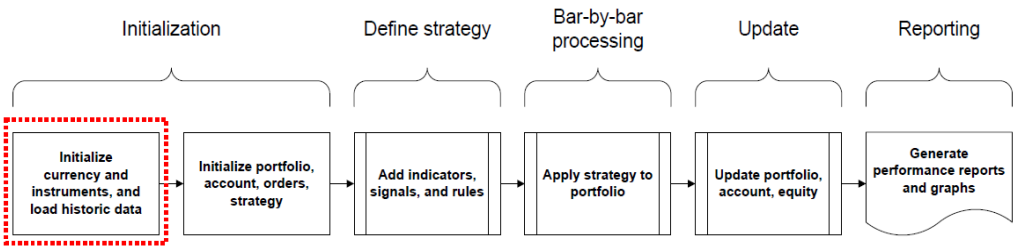
- 目的:
  - 以台積電為標的，驗證quantstrat 實作布林通道逆勢策略的回測結果，並檢驗交易內容資料正確性。
  - 測試 quantstrat之WFA滾動式資料窗格之優化參數效果

執行 (目錄: 證基會檔案\程式碼\WFA_TSMC_v1.R)

回測參數假設

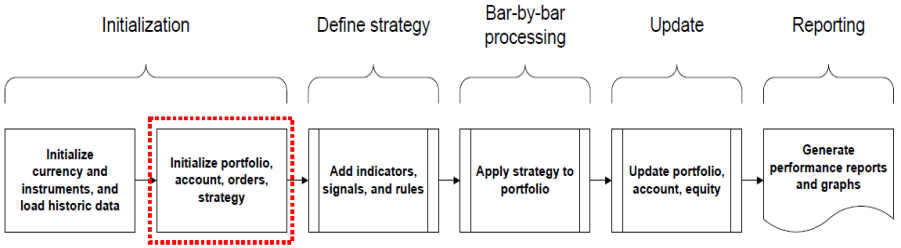
回測時間:	2007/1/1~2016/6/15
交易資料解析度:	Daily
布林通道參數:	MA: 20日
	STD: 0.5
交易邏輯:	mean-reversion
	通道過高賣，破低買
	無參數優化
本金:	TWD 1,000,000
交易成本:	TWD 120
部位管理:	1月10日
	固定比率
	無加減碼規則
WFA 方式:	滾動窗格
訓練時間:	4年
測試時間:	1 年

流程 1 系統設定啟始



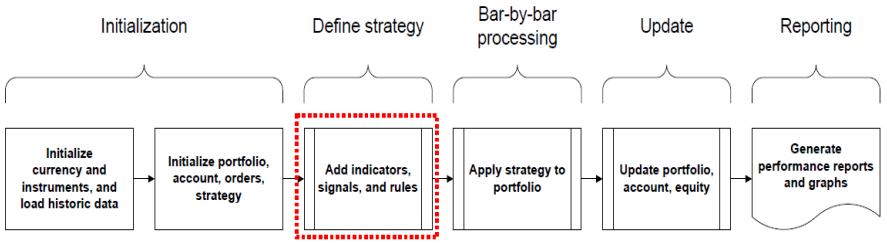

```
library(quantstrat)


rm(list=ls(all=TRUE))

###############################################################################
#1 系統啟始化 商品設定

stock.st = c("2330.TW")

currency("TWD")
stock(stock.st, currency="TWD",multiplier=1)

oldtz<-Sys.getenv('TZ')

Sys.setenv(TZ="Asia/Taipei")

# 回測時間區間
initDate = '2006-12-31'
startDate = '2007-01-01'
endDate = '2016-06-15'

initEq=1e6
tradeSize = initEq/10
```

流程 2 載入歷史資料



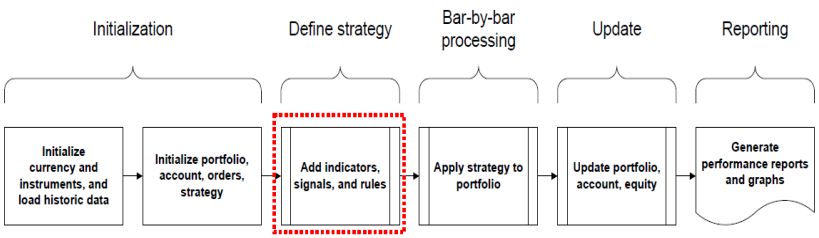

```
#################################################################
#2 台積電歷史日K資料

getSymbols(stock.st,from=startDate,to=endDate,index.class="POSIXct",adjust=T)


myTheme<-chart_theme()
myTheme$col$dn.col <-'lightblue'
myTheme$col$dn.border <- 'lightgray'
myTheme$col$up.border <- 'lightgray'
chart_Series(get(stock.st),name=stock.st,theme=myTheme)


## 定比例部位
osFixedDollar <- function(timestamp, orderqty, portfolio, symbol, ruletype, ...)
{
  pos <- getPosQty(portfolio, symbol, timestamp)
  if( isTRUE(all.equal(pos,0)) )
  {
    ClosePrice <- as.numeric(Cl(mktdata[timestamp,]))
    orderqty <- sign(orderqty)*round(tradeSize/ClosePrice,-2)
  } else {
    orderqty <- 0
  }
  return(orderqty)
}

strat.st <- "bbands"

# rm.strat(strat.st)
if (!exists('.blotter')) .blotter <- new.env()
if (!exists('.strategy')) .strategy <- new.env()
```

流程 3 交易策略設定



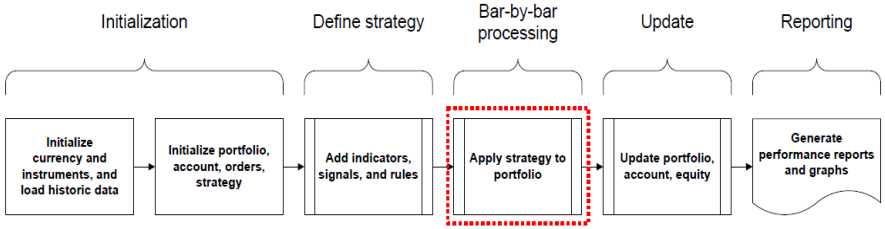

```
#################################################################
#3 交易策略設定


strategy(strat.st, store=TRUE)

add.indicator(strat.st, name = "BBands",
              arguments = list(HLC = quote(HLC(mktdata)), maType='SMA'), label='BBands')
add.signal(strat.st, name="sigCrossover",
           arguments=list(columns=c("Close","up"),relationship="gt"),
           label="Cl.gt.UpperBand")
add.signal(strat.st, name="sigCrossover",
           arguments=list(columns=c("Close","dn"),relationship="lt"),
           label="Cl.lt.LowerBand")
add.signal(strat.st, name="sigCrossover",
           arguments=list(columns=c("High","Low","mavg"),relationship="op"),
           label="Cross.Mid")


add.rule(strat.st, name='ruleSignal',
         arguments=list(sigcol="Cl.gt.UpperBand",sigval=TRUE, orderqty=-1000,
                        ordertype='market', orderside=NULL, threshold=NULL, osFUN=osFixedDollar,
                        orderset='ocoshort'),
         type='enter',label="SE")
add.rule(strat.st, name='ruleSignal',
         arguments=list(sigcol="Cl.lt.LowerBand",sigval=TRUE, orderqty= 1000,
                        ordertype='market', orderside=NULL, threshold=NULL, osFUN=osFixedDollar,
                        orderset='ocolong'),
         type='enter',label="LE")
add.rule(strat.st, name='ruleSignal',
         arguments=list(sigcol="Cross.Mid",sigval=TRUE, orderqty= 'all',
                        ordertype='market', TxnFees=-120, orderside=NULL, threshold=NULL),
         type='exit')
```

流程 4 WFA 滾動窗參數設定



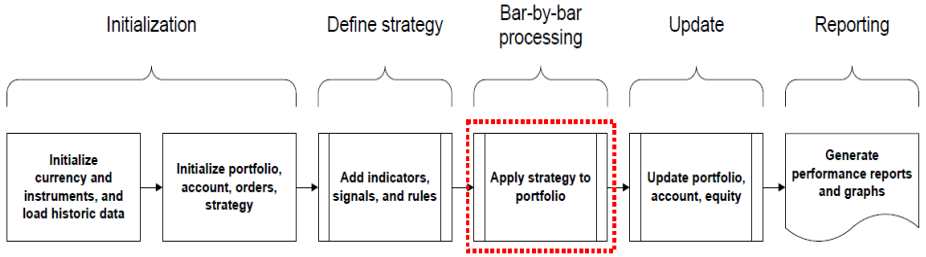

```
#################################################################################
#4 WFA 滾動窗參數設定


add.distribution(strat.st,
                 paramset.label = 'BBOPT',
                 component.type = 'indicator',
                 component.label = 'BBands',
                 variable = list(n = seq(10,30,by=5)),
                 label = 'n'
)
add.distribution(strat.st,
                 paramset.label = 'BBOPT',
                 component.type = 'indicator',
                 component.label = 'BBands',
                 variable = list(sd = seq(1,3,by=0.5)),
                 label = 'sd'
)

#
# WFA 平行CPU多核心計算設定
#
if( Sys.info()['sysname'] == "Windows" )
{
  library(doParallel)
  # uncomment line below when combine function bug is fixed for Windows
  # registerDoParallel(cores=detectCores())  ## to-be-resolved by Julian
} else {
  library(doMC)
  registerDoMC(cores=detectCores())
}
```

流程 5 執行回測




```
#############################################################
#5 開始執行回測

rm.strat("opt")
initPortf(name="opt", stock.st, initDate=initDate)
initAcct(name="opt", portfolios="opt",
         initDate=initDate, initEq=initEq)

initOrders(portfolio="opt", initDate=initDate)


results <- walk.forward(
  strategy.st=strat.st,
  paramset.label='BBOPT',
  portfolio.st="opt",
  account.st="opt",
  period='years',
  k.training=4,
  k.testing=1,
  nsamples=0,
  audit.prefix='wfa',
  anchored=FALSE,
  verbose=TRUE
)
```

流程6 WFA 回測結果




```
#########################################################
#6 WFA 回測結果

# WFA回測結果總表
PerformanceAnalytics:::textplot(t(tradeStats("opt")))

# 已實現損益
txns <- getTxns("opt",stock.st)
txns$Net.Txn.Realized.PL <- round(txns$Net.Txn.Realized.PL)

# WFA 交易明細
PerformanceAnalytics:::textplot(head(txns))
PerformanceAnalytics:::textplot(tail(txns))

# Out-of-sample 樣本外損益
plot(getPortfolio("opt")$summary$Net.Trading.PL,minor.ticks=FALSE,type="h",col=4)

# 交易進出場部位圖
chart.Posn("opt",stock.st)


list.files(pattern="^wfa.*\\.RData$")

Sys.setenv(TZ=oldtz)
```

回測統計表畫面

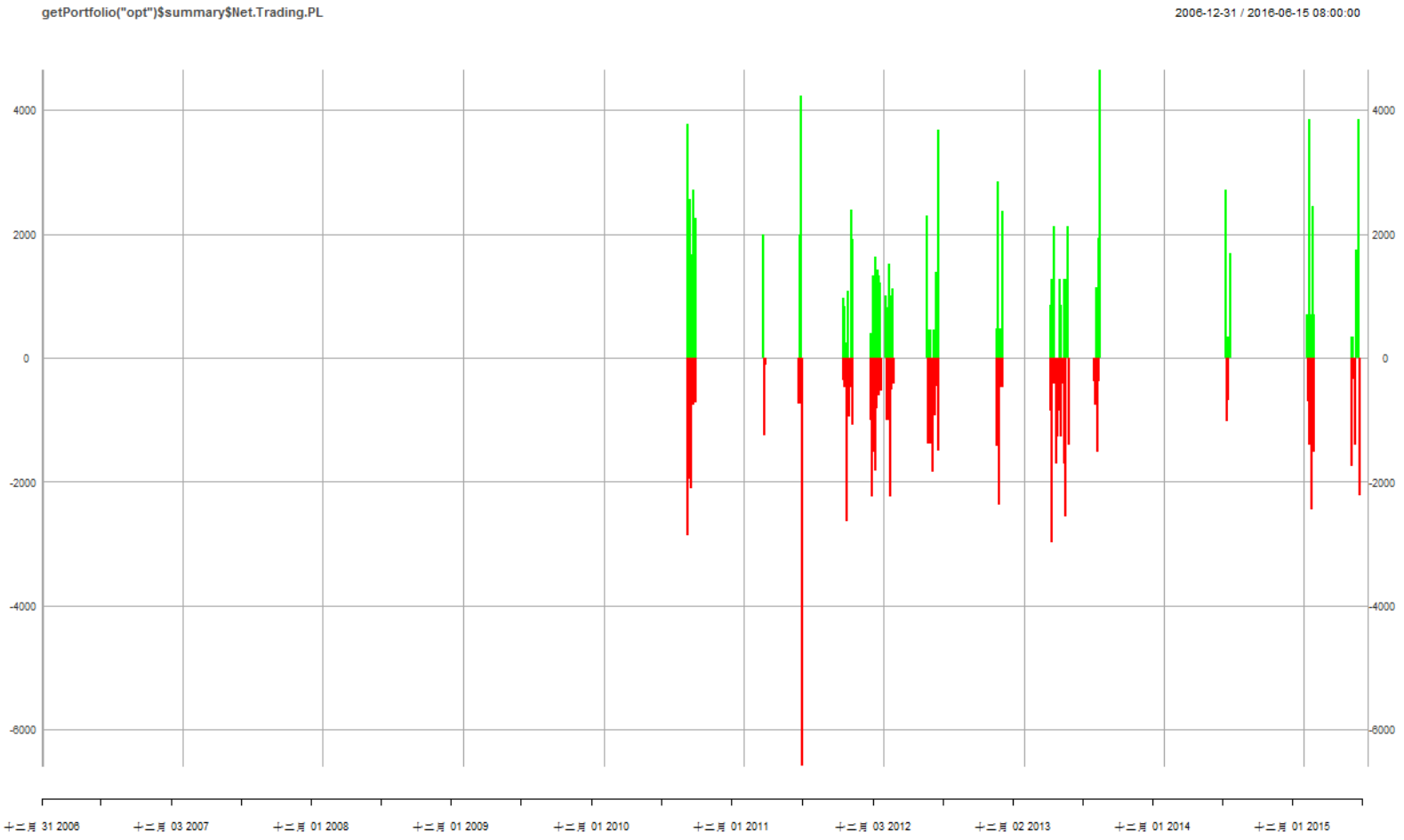
統計總表

	2330.TW
Portfolio	opt
Symbol	2330.TW
Num.Txns	26
Num.Trades	13
Net.Trading.PL	16686.45
Avg.Trade.PL	1283.573
Med.Trade.PL	1000.374
Largest.Winner	6116.895
Largest.Loser	-6518.183
Gross.Profits	25238.06
Gross.Losses	-8551.616
Std.Dev.Trade.PL	3293.933
Percent.Positive	76.92308
Percent.Negative	23.07692
Profit.Factor	2.951263
Avg.Win.Trade	2523.806
Med.Win.Trade	2417.477
Avg.Losing.Trade	-2850.539
Med.Losing.Trade	-1042.032
Avg.Daily.PL	1283.573
Med.Daily.PL	1000.374
Std.Dev.Daily.PL	3293.933
Ann.Sharpe	6.185945
Max.Drawdown	-11987.16
Profit.To.Max.Draw	1.392027
Avg.WinLoss.Ratio	0.8853788
Med.WinLoss.Ratio	2.319965
Max.Equity	18906.45
Min.Equity	-2872.07
End.Equity	16686.45

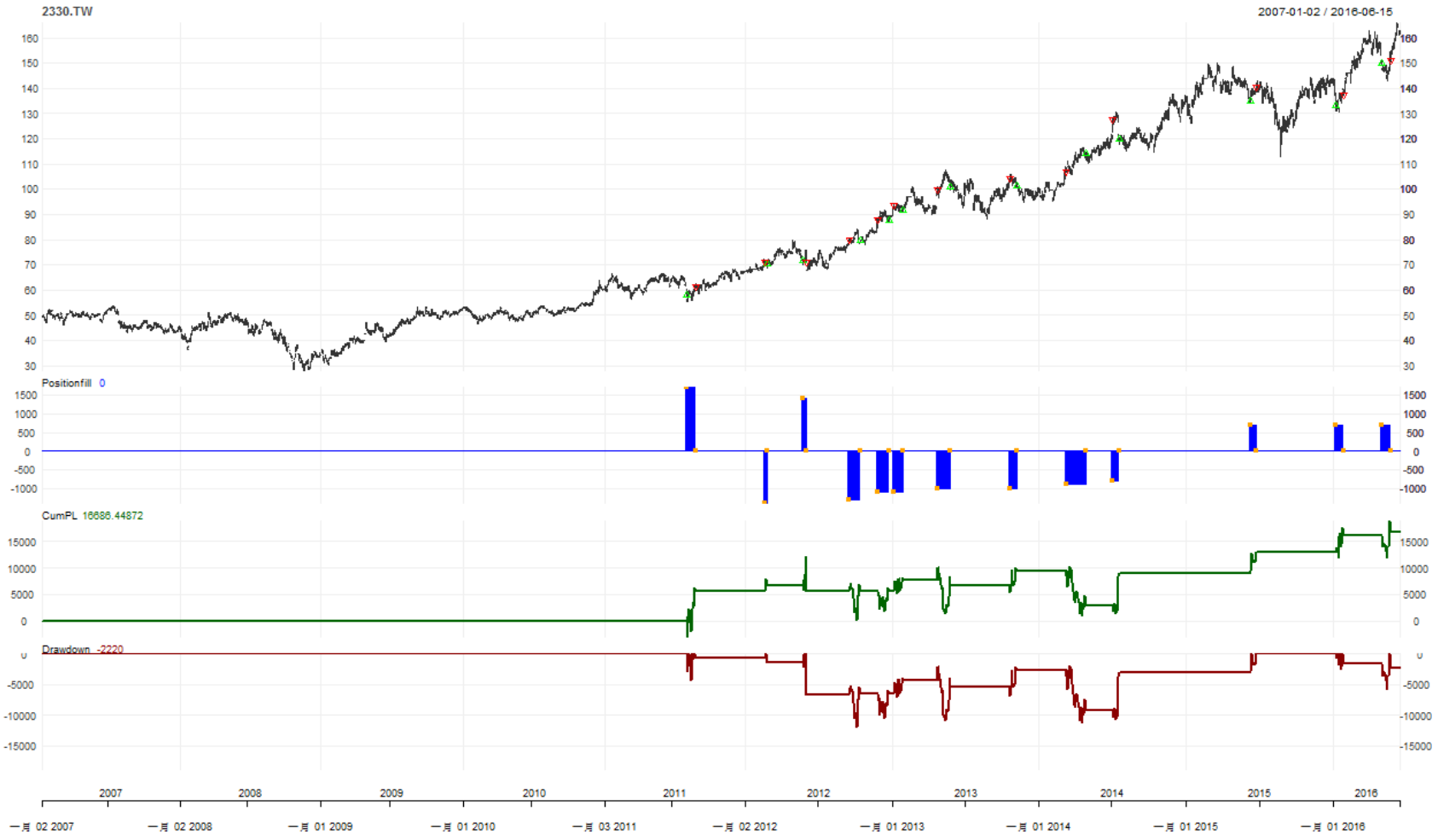
WFA樣本外交易紀錄

	Txn.Qty	Txn.Price	Txn.Fees	Txn.Value	Txn.Avg.Cost	Net.Txn.Realized.PL
2006-12-31 00:00:00	0	0	0	0	0	0
2011-08-08 08:00:00	1700	57.9749113390375	0	98557.3492763638	57.9749113390375	0
2011-08-30 08:00:00	-1700	61.3538172146256	-120	-104301.489264864	61.3538172146256	5624
2012-02-20 08:00:00	-1400	71.1348605386963	0	-99588.8047541749	71.1348605386963	0
2012-02-27 08:00:00	1400	70.334593357636	-120	98468.4307006904	70.334593357636	1000
2012-05-24 08:00:00	1400	71.6683719927366	0	100335.720789831	71.6683719927366	0

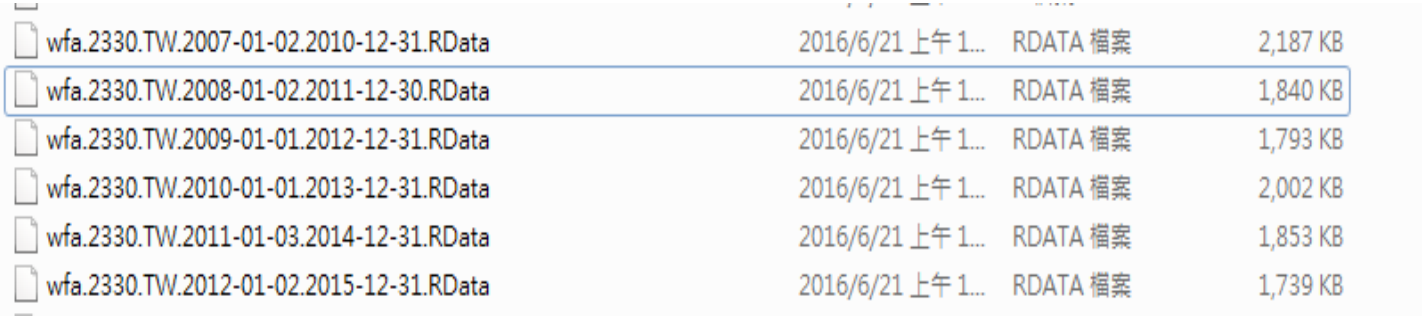
樣本外損益表


getPortfolio("opt")$summary$Net.Trading.PL
2006-12-31 / 2016-06-15 08:00:00
4000
2000
0
-2000
-4000
-6000
十二月 31 2006
十二月 03 2007
十二月 01 2008
十二月 01 2009
十二月 01 2010
十二月 01 2011
十二月 03 2012
十二月 02 2013
十二月 01 2014
十二月 01 2015

交易部位紀錄


2330.TW
2007-01-02 / 2016-06-15
Positionfill 0
CumPL 16686.44872
Drawdown -2220
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
一月 02 2007
一月 02 2008
一月 01 2009
一月 01 2010
一月 03 2011
一月 02 2012
一月 01 2013
一月 01 2014
一月 01 2015
一月 01 2016

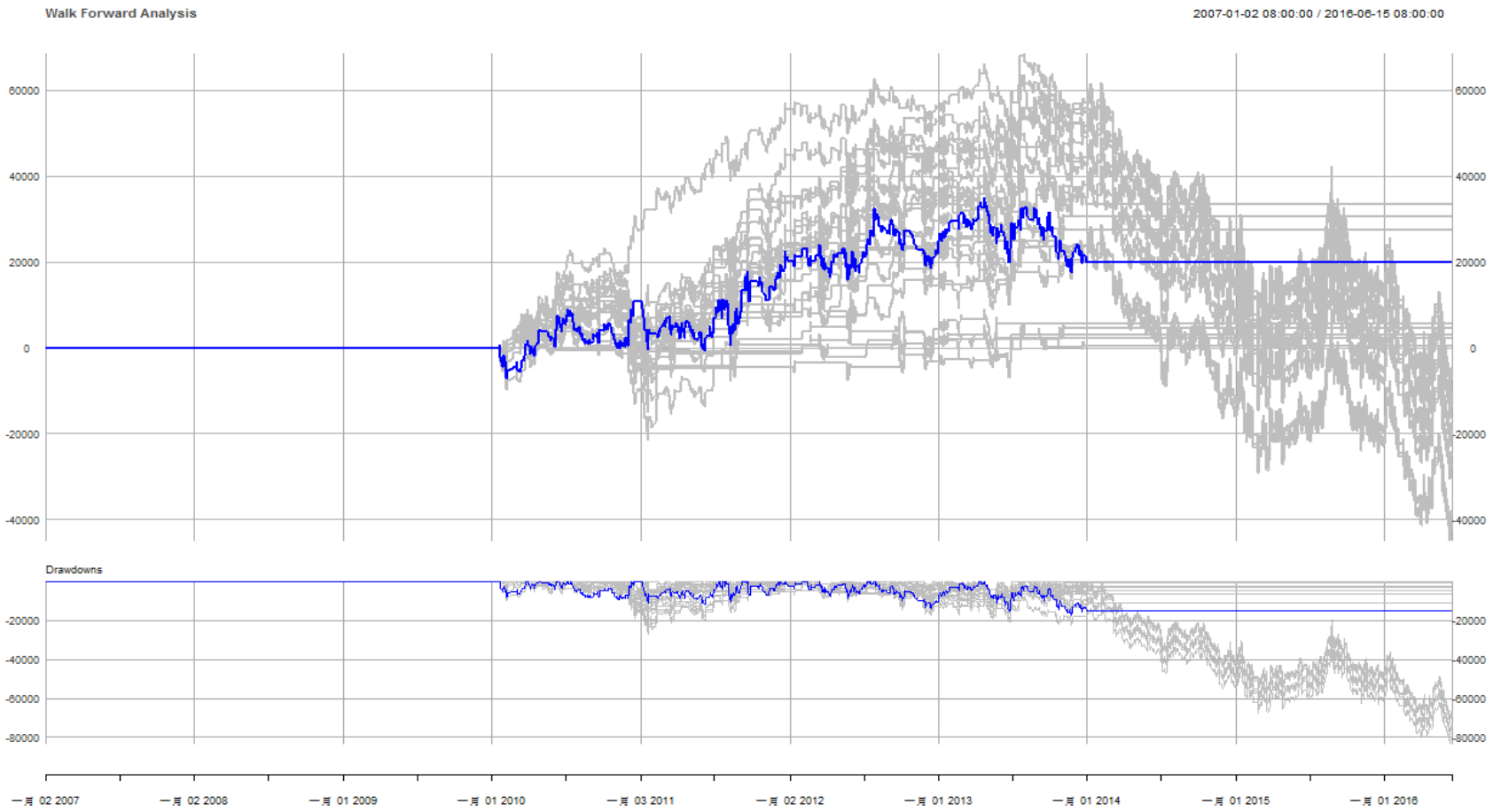
樣本外交易紀錄檔案



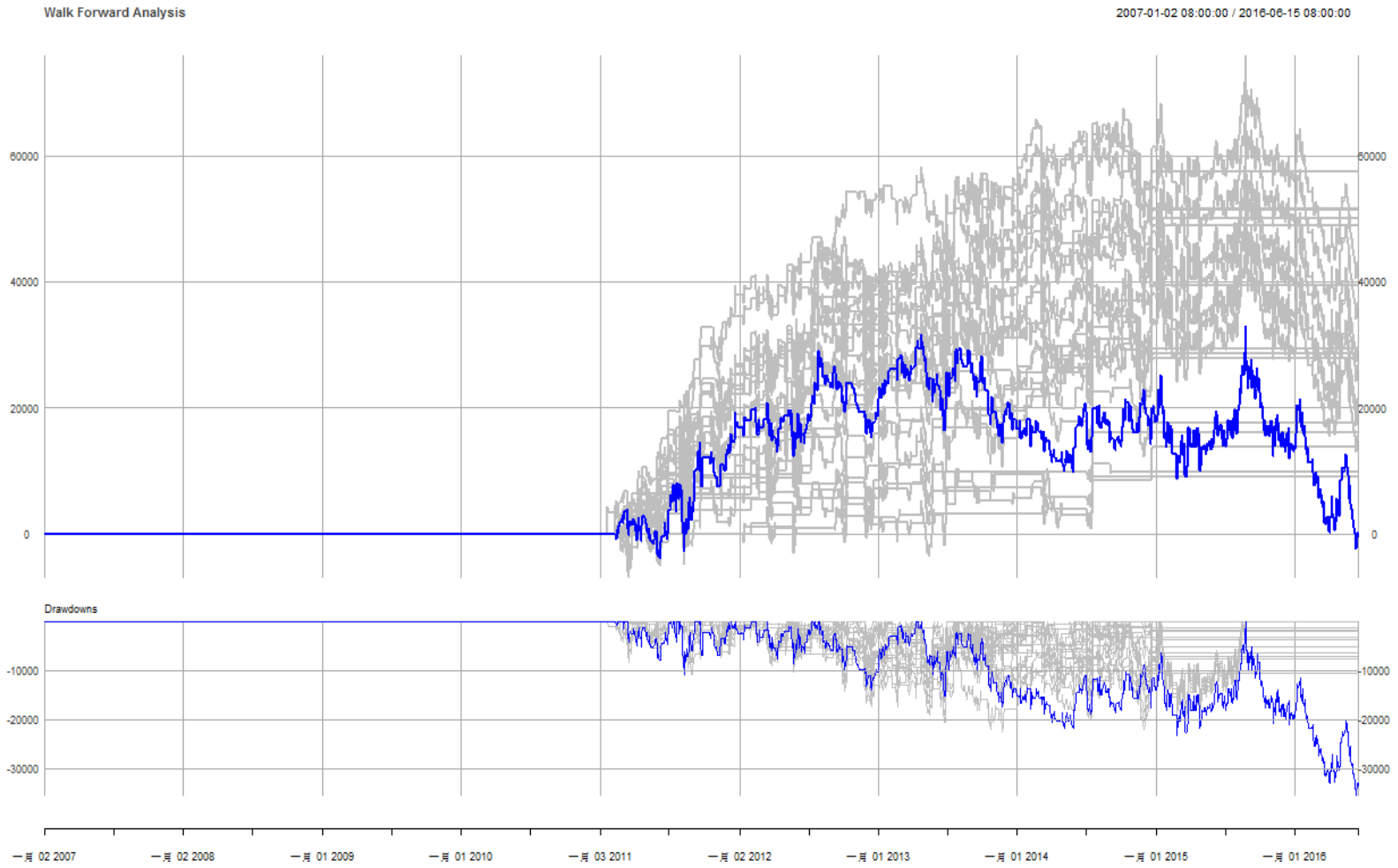
交易資料檢查驗證，使用**load** 指令:

load(wfa.2003.tw.xxxx-xx-xx.xxxx-xx-xx.Rdata)

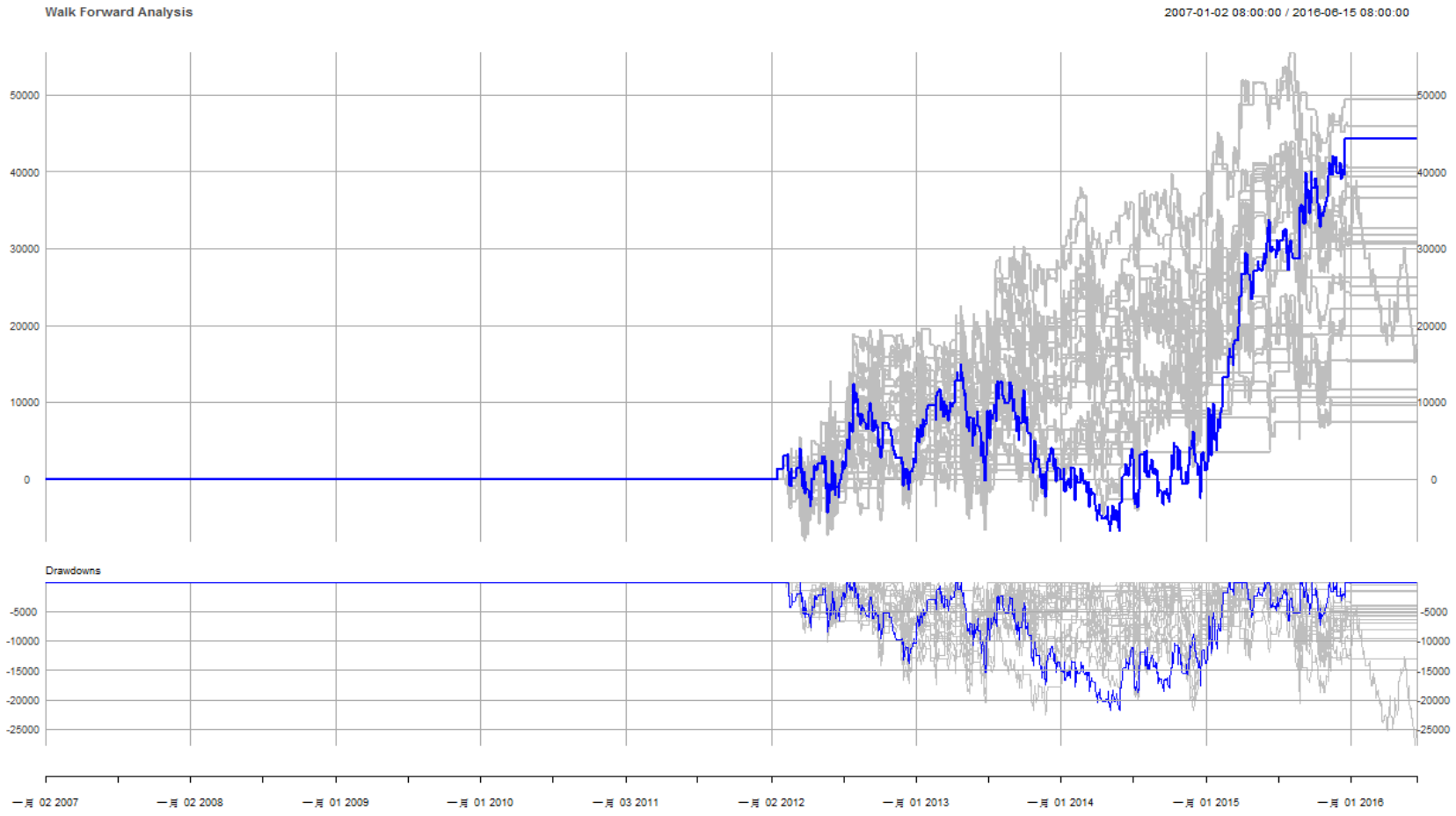
2010~2013 推進回測圖



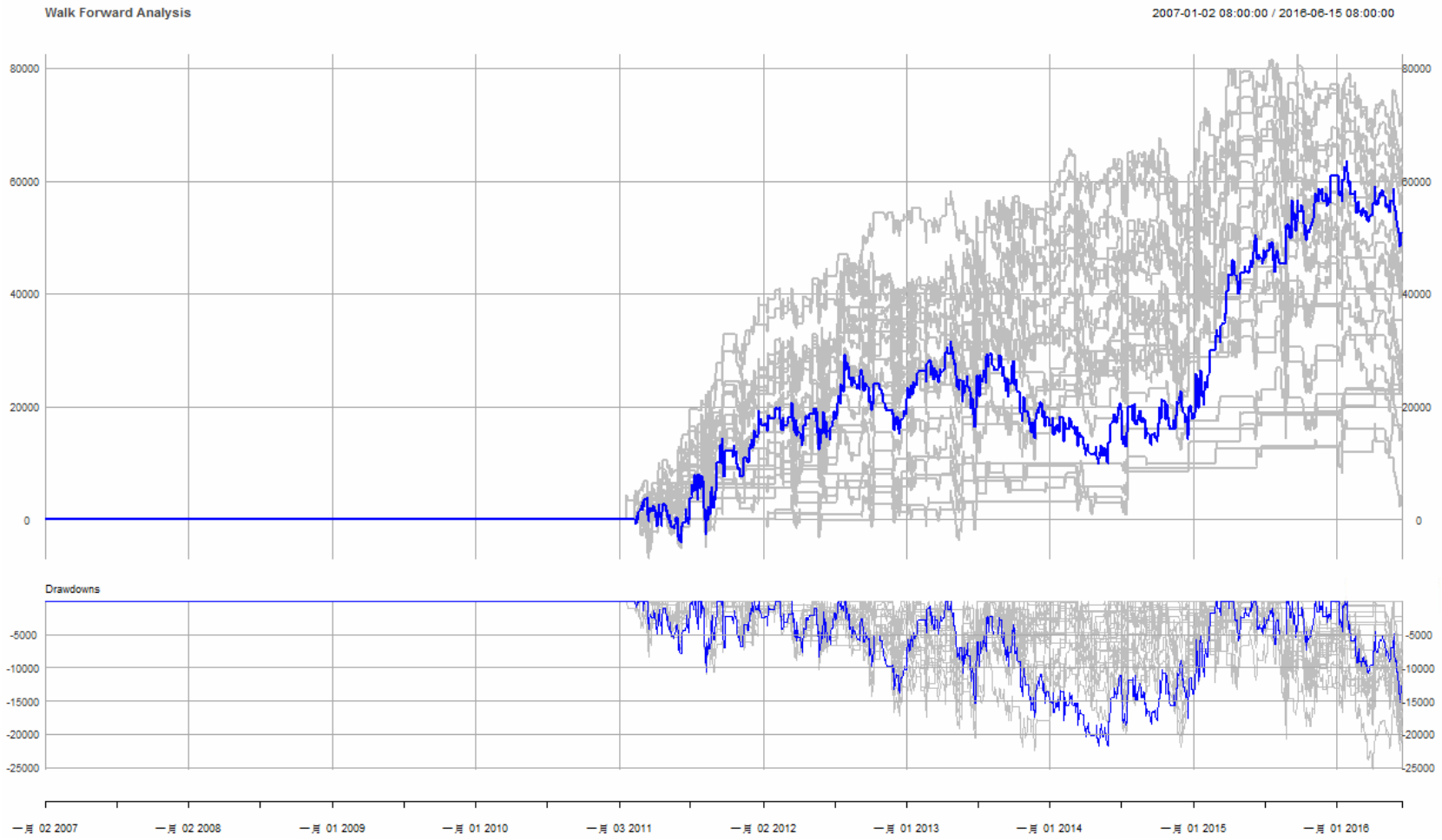
2011~2014 推進回測圖



2012~2015 推進回測圖



總推進回測績效表


Walk Forward Analysis
2007-01-02 08:00:00 / 2016-06-15 08:00:00
80000
60000
40000
20000
0
Drawdowns
-5000
-10000
-15000
-20000
-25000
一月 02 2007
一月 02 2008
一月 01 2009
一月 01 2010
一月 03 2011
一月 02 2012
一月 01 2013
一月 01 2014
一月 01 2015
一月 01 2016

結論與效益

- 提供逆勢交易策略基本模板，後續做為平行運算優化同類型策略之基礎。

- 單純布林通道逆勢交易，其投資報酬率明顯低於buy-and-hold方法，也遠比定存低，證明不是可交易策略，但:
  - 透過基因演算，找出收益率更高的參數組合。
  - 可做為多策略內特定條件之輔助策略。

- WFA 效益，淨利由原始1.66萬，增加為5萬，證明具有顯著效益，以此模式再進行大量統計檢定其model robustness。

課後討論