

Volatility-Targeting Strategy

```
setwd("C:\\Users\\user\\Desktop")
data<-read.csv("0050MONTH.csv")
return<-data[,3] #0050 報酬
price<-data[,4] #0050 股價
MonthSd<-data[,6] #月內標準差
rt<-data[,7] #央行定存利率
ExcessReturn <- (return/100)-rt #超額報酬
alpha<-1-exp(log(exp(1),0.5)/0.5) #alpha 為 1-exp(Ln(0.5)/t)
weight<- vector(mode ="numeric",length = length(ExcessReturn)) #開一個權重的向量儲存所有權重
weight[1]<- alpha #第一期的權重為 alpha
for (i in 2:length(ExcessReturn)){
  weight[i]<-(1-alpha)^(i-1)*alpha
}
#權重公式為 wt=(1-alpha)^(t-1)*alpha
sum(weight)#權重加總等於 1

## [1] 1

#x,y 為計數器
#為了要計算預測波動度(1-alpha)^t*x0+(1-alpha)^(t-1)*alpha*x1...+(1-alpha)*alpha*xt-1+alpha*xt
#因頭尾與其他項不同 因此分開計算
x=length(MonthSd)-1
y=2
SUM<-0
while(x>=2&& y< 124){
  SUM= SUM+(1-alpha)^x*alpha*MonthSd[y]
  x=x-1
  y=y+1
}

PredictSd = SUM+(1-alpha)*MonthSd[1]+alpha*MonthSd[length(MonthSd)]
PredictSd #預測波動度

## [1] 0.9568366

Targetvolatility = 0.12 #目標波動度 預測12%
PredictWeight = Targetvolatility^2/(12*MonthSd[length(MonthSd)]^2)#預測權重 將月資料年化
PredictWeight

## [1] 0.001319896
```

```

MonthReturnPer<-return/100
MarketReturn<-unlist(data[,5])/100 #市場報酬
MonthReturn<-MonthReturnPer/100 #月報酬
capm<-lm(MonthReturn~MarketReturn)
beta<-coef(capm)[2] #計算capm的beta係數
rf<-rt/100
PortfolioReturn<-MonthReturn*weight+rf*(1-weight) #令rt為無風險利率 計算投資組合報酬
alpha1<-mean(PortfolioReturn)-mean(rf)+beta*(mean(MarketReturn)-mean(rf)) #計算超額報酬
shapeRatio<-mean(PortfolioReturn)-mean(rf)/sd(MonthReturn) #計算夏普指標
library(fTrading)

## Warning: 套件 'fTrading' 是用 R 版本 4.1.3 來建造的

## 載入需要的套件：timeDate

## Warning: 套件 'timeDate' 是用 R 版本 4.1.3 來建造的

## 載入需要的套件：timeSeries

## Warning: 套件 'timeSeries' 是用 R 版本 4.1.3 來建造的

## 載入需要的套件：fBasics

## Warning: 套件 'fBasics' 是用 R 版本 4.1.3 來建造的

mdd = maxDrawDown(MonthReturn) #計算mdd
mdd

## $maxdrawdown
## [1] 0.00053002
##
## $from
## [1] 2
##
## $to
## [1] 110

result<-rbind(mdd$maxdrawdown,alpha1,shapeRatio)
rownames(result)[1]<-"MDD"
result

##           MarketReturn
## MDD           5.300200e-04
## alpha1       -2.731148e-04
## shapeRatio   -1.762994e+02

```