University of Cape Town

STA4016H

Portfolio theory

Minimum Variance Backtesting and Out-of-Sample Performance

Author:
Chongo Nkalamo

Student Number:
NKLCHO001

March 24, 2019

Contents

1 Introduction **2**

2 Part I **3**

2.1 Sample Error when Estimating the Sharpe Ratio 3

2.2 The Maximum of the Sample 7

2.3 Minimum Backtest Length 8

3 Part II **10**

3.1 Experiment I: In Sample and Out of Sample Sharpe Ratios. 10

3.2 Experiment II: Out-Of-Sample Backtesting using a Rolling Window. . 14

4 Conclusion **16**

5 References **18**

6 Appendix **19**

1 Introduction

In this assignment we backtest a buy and hold investment strategy on 10 assets to find the optimal specification of the investment stragtegy. The buy and hold strategy is implemented in two ways discussed in two separate experiments in part two. Thereafter, comparison between the two methods or specification is discussed and recommendations as to which of the methods performed best will be given. These recommendations will be based off out-of-sample (OOS) performance and Sharpe ratio performance and all this is discussed in part II of this paper. Part I of this paper gives discussions and proofs of theoretical principals of Sharpe ratios and backtesting.

2 Part I

2.1 Sample Error when Estimating the Sharpe Ratio

Statement : The distribution of the estimated annualized Sharpe ratio $\hat{SR}$ converges asymptotically ($n \to \infty$) to

$$\hat{SR} \to N\left(SR, 1 + \frac{\frac{SR^2}{2q}}{n}\right)$$

for n $\to$ ∞, where n is the number data points (here years) used to estimate the statistic.

Proof :

To show the above the assertion, we make use of use of the asymptotic properties of MLE's as outlined in Stewart et al[3].
We begin with the assumption that excess returns follow a normal distribution. Therefore, the pdf of a single return is given by

$$f(R_i, \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(R_i-\mu)^2}{2\sigma^2}}$$

Consequently the log-likelihood of n excess returns, we will be

$$l(\theta) = -\frac{1}{2}nlog(\sigma^2) - \frac{1}{2}nlog(2\pi) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(R_i - \mu)^2$$

Taking partial derivatives we obtain the following estimates

$$\frac{\partial l(\theta)}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(R_i - \mu) = 0$$
$$\implies \hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} Ri$$

$$\frac{\partial l(\theta)}{\partial \mu} = -\frac{n}{\sigma^2} + \frac{1}{2(\sigma^2)^2}\sum_{i=1}^{n}(R_i - \mu)^2 = 0$$
$$\implies \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} R_i$$

Computing the fisher information matrix we get

$$\mathbf{I}(\theta_0) = \begin{pmatrix} \frac{n}{\hat{\sigma^2}} & 0 \\ 0 & \frac{n}{2\hat{\sigma^4}} \end{pmatrix}$$

So that the inverse of our fisher matrix gives us our variance-covariance matrix. And this gives

$$\mathbf{I}(\theta_0)^{-1} = \begin{pmatrix} \frac{\hat{\sigma^2}}{n} & 0 \\ 0 & \frac{2\hat{\sigma^2}}{n} \end{pmatrix} \tag{1}$$

Now, if we consider a taylor expansion of the score $l'(\hat{\theta})$ function around θ_0 :

$$l'(\hat{\theta}) \approx l'(\theta_0) + (\hat{\theta} - \theta_0)l''(\theta_0) + ...$$

Now $l'(\hat{\theta}) = 0$ since $\hat{\theta}$ is the MLE, so that

$$(\hat{\theta} - \theta_0) \approx -\frac{l'(\theta_0)}{l''(\theta_0)}$$

Multiplying the above cleverly by 1 we obtain

$$(\hat{\theta} - \theta_0) \approx -\frac{n^{-1/2}l'(\theta_0)}{n^{1/2}n^{-1}(l''(\theta_0))}$$

so that

$$\sqrt{n}(\hat{\theta}-\theta_0) \approx -\frac{n^{-1/2}l'(\theta_0)}{n^{-1}(l''(\theta_0))} = \frac{n^{-1/2}l'(\theta_0)}{\mathbf{I}(\theta_0)} \tag{2}$$

As shown in Stewart.et al[3], $E[\sqrt{n}(\hat{\theta}-\theta_0)] = 0$ and

$$\begin{aligned} var[\sqrt{n}(\hat{\theta}-\theta_0)] &= \frac{1}{[\mathbf{I}(\theta_0)]^2} var[n^{-1/2}l'(\theta_0)] \\ &= \frac{1}{[\mathbf{I}(\theta_0)]^2}\mathbf{I}(\theta_0) \\ &= \frac{1}{\mathbf{I}(\theta_0)} \end{aligned}$$

Given all this we note from (2), that the denominator is constant while the numerator is given by:

$$n^{-1/2}l'(\theta_0) = n^{-1/2}\sum_{i=1}^{n}\left[\frac{\partial}{\partial\theta}\ln f((X_i|\theta_0))\right]$$

which represents the sum of n i.i.d random variables and hence from the Central Limit Theorem (CLT) converges in distribution to a Normal distribution as $n \to \infty$. Hence we have

$$\sqrt{n}(\hat{\theta}-\theta_0) \longrightarrow N(0, \mathbf{I}(\theta_0)^{-1})$$

Now going back to(1), we have for a single return (n = 1),

$$\mathbf{I}(\theta_0)^{-1} = \begin{pmatrix} \hat{\sigma^2} & 0 \\ 0 & 2\hat{\sigma^4} \end{pmatrix}$$

Hence, putting all the above together we finally get (as $n \to \infty$)

$$\sqrt{n}(\hat{\theta}-\theta_0) \longrightarrow N(0, \mathbf{V}_\theta), \mathbf{V}_\theta = \begin{pmatrix} \hat{\sigma^2} & 0 \\ 0 & 2\hat{\sigma^4} \end{pmatrix}$$

or

$$\hat{\theta} \to N(\theta_0, (n\mathbf{I}(\theta_0)^{-1}) \tag{3}$$

This is the derivation seen in the appendix of the paper by Andrew Lo[2]. As described by Lo, the Sharpe ratio estimator can be written as a function

$g(\hat{\theta})$ of $\hat{\theta}$, its asymptotic distribution follows directly from the delta method:

$$\sqrt{T}(\hat{SR} - SR) \sim N(0, \mathbf{V}_g), \mathbf{V}_g = \left[\frac{\partial g}{\partial \mu}\right]^2 \sigma^2 + \left[\frac{\partial g}{\partial \sigma^2}\right]^2 2\sigma^4$$

with $g(\hat{\theta}) = \frac{(\mu - R_f)\sqrt{q}}{\sigma}$ for our purposes. Taking partial derivatives of the Sharpe ratio function g with respect to μ and σ^2, $\mathbf{V}_g$ reduces to

$$\begin{aligned}\mathbf{V}_g &= \left[\frac{\sqrt{q}}{\sigma^2}\right]\sigma^2 + \left[-\frac{\sqrt{q}(\mu - R_f)}{2\sigma^3}\right]^2 2\sigma^4, \quad \frac{\partial g}{\partial \sigma^2} = -\frac{(\mu - R_f)}{2\sigma^3} \quad \frac{\partial g}{\partial \mu} = \frac{1}{\sigma^2} \\ &= q + \frac{(\mu - R_f)^2 q}{2\sigma^2} \\ &= q + \frac{1}{2}\left[\frac{(\mu - R_f)}{\sigma}\right]^2 \sqrt{q}\sqrt{q} \\ &= q + \frac{1}{2}SR^2\sqrt{q} \\ &= 1 + \frac{SR^2}{2q}\end{aligned}$$

Hence we have,

$$\sqrt{T}(\hat{SR} - SR) \sim N\left(0, 1 + \frac{SR^2}{2q}\right)$$

Therefore from(3) we get that the distribution of the estimated annualized Sharpe ratio $\hat{SR}$ converges asymptotically to

$$\hat{SR} \sim N\left(SR, 1 + \frac{\frac{SR^2}{2q}}{n}\right)$$

for n → ∞, where n is the number data points (here years) used to estimate the statistic.

2.2 The Maximum of the Sample

Theorem1.1. Given a sample of N IID Normal random variables $X_n : n = 1, 2, ..., N$ where Z is the CDF for the standard normal distribution. The expected maximum of the sample:

$$E[max_N] := E[max\{X_n\}].$$

can be approximated as:

$$E[max_N] \approx ((1-\gamma)Z^{-1}\left[1-\frac{1}{N}\right] + \gamma Z^{-1}\left[1-\frac{1}{N}e^{-1}\right]$$

for some constant γ.

Motivation :

To derive an approximation for the sample maximum, max_N, we apply the Fisher-Tippett- Gnedenko theorem to the Gaussian distribution and obtain that the **normalized** maximum (max_n) converges almost surely to one of the generalized extreme value distributions $G(x)$ namely Gumbel, Frechet or Weibull. Bailey et al.[1] We obtain that

$$Pr\left\{\frac{max_N - \alpha}{\beta} \leq x\right\} = F^n(\alpha x + \beta) \longrightarrow G(x), \quad N \to \infty$$

for some non-degenerate distribution function $G(z)$.
If the above relationship holds for some non-degenerate distribution function G, then F is said to be in the maximum domain of attraction of G.

The Gaussian distribution is one such distribution known to be in the domain of attraction of the Gumbel distribution and the exponential the other. Given that our stochastic random variable $\mathbf{X_n}$ comes from a Gaussian distribution, our function G will essentially be Gumbel. Hence,

$$F^n(\alpha x + \beta) \longrightarrow G(x)$$

• where $G(x) = e^{-e^{-x}}$, is the CDF for the standard Gumbel distribution.
• $\alpha = Z^{-1}[1 - \frac{1}{N}]$
• $\beta = Z^{-1}[1 - \frac{1}{N}e^{-1}] - \alpha$
and Z^{-1} being the inverse of the CDF of the standard normal.
The normalizing constants α and β are derived in Resnick [4] and Embrechts et al.[5]

From Bailey et al[1], we have that for sufficiently large N , the mean of the sample maximum of standard normally distributed random variables can be approximated by

$$E[max_N] \approx \alpha + \beta\gamma$$

the result follows as

$$E[max_N] \approx (1-\gamma)Z^{-1}\left[1-\frac{1}{N}\right] + \gamma Z^{-1}\left[1-\frac{1}{N}e^{-1}\right]$$

2.3 Minimum Backtest Length

Theorem1.2. The minimum backtest length T_{min} needed to avoid selecting a strategy with an in-sample Sharpe Ratio as the average E[maxX_n] among N independent strategies with an out-of-sample Sharpe Ratio of zero is:

$$Tmin < \frac{2ln(N)}{E[E[max_N]]^2}$$

Proof :

We are given that $X_n : n = 1, 2, ..., N$ are IID normally distributed stochastic random variables from Theorem 1.1. From theorem 16 in Wasserman.L [7] which is proven in the appendix via jensen's inequality we have that

$$E[max_N] \leq \sigma\sqrt{2ln(N)}$$

From the paper by Bailey et al[1] we have that the minimum back test (MinBTL) is given by

$$MinBTL \approx \left(\frac{(1-\gamma)Z^{-1}[1-\frac{1}{N}] + \gamma Z^{-1}[1-\frac{1}{N}e^{-1}]}{E[E[max_N]]}\right)^2$$

We know that $E[max_N] \approx (1-\gamma)Z^{-1}[1-\frac{1}{N}] + \gamma Z^{-1}[1-\frac{1}{N}e^{-1}]$ therefore

$$
\begin{aligned}
MinBTL \approx (\frac{E[max_N]}{E[E[max_N]]})^2 &< \left(\frac{\sigma\sqrt{2ln(N)}}{E[E[max_N]]}\right)^2 \\
&= \frac{\sigma^2 2ln(N)}{E[E[max_N]]^2} \\
&= \frac{2ln(N)}{E[E[max_N]]^2}
\end{aligned}
$$

The final line above assumes that $\mathbf{X_n} \sim N(0,1)$ hence

$$T_{min} = MinBTL < \frac{2ln(N)}{E[E[max_N]]^2}$$

3 Part II

3.1 Experiment I: In Sample and Out of Sample Sharpe Ratios.

In this section we implement a buy and hold strategy to a portfolio of 10 assets consisting of the Bond index (ALBI) and Industrial indices (based on the Industrial ICB classification). A buy and hold strategy means that an investor allocates their own budget to a portfolio of assets and sticks with those assets for a period of time without buying or selling. For experiment I, we implement the buy and hold strategy on the 10 assets in the out of sample period by continuously rebalancing portfolio controls and holding the assets for a fixed period.

The in-sample data was used to calculate the out of sample initial weights/portfolio controls. These weights are the weights that give the tangency portfolio which is the maximum Sharpe ratio portfolio.

In R, using the in-sample data, a tangency porfolio was obtained. Table 1 shows the tangency portfolio obtained from the in sample data. The table indicates that 7.676225e-18 of the total available funds were invested in ALBI asset and so on. Figure 1 shows the a plot of the in-sample (IS) tangency portfolio against the efficient frontier. The maximum Sharpe ratio portfolio is also indicated for the given level of return and risk.

The in-sample portfolio returns and volatility were 0.2551269 and 0.1696472 respectively. The Sharpe ratio obtained for the whole training period was computed to be 1.001374. It is worth noting that from month to month throughout the training period, the portfolio controls remain fixed and hence no allowance for compounding. However, as will be discussed shortly, compounding is considered in the test period.

The test period consists of the same length data as the training period. The only difference in the calculations of our various statistics is that this time we allow for compounding and thus rebalance our portfolio controls. The rebalancing takes place monthly.

A matrix of asset returns from month to month throughout the test period was obtained and this depicts the relative price changes for each respective matrix from one month to the next. This matrix was then used to obtain a new portfolio control matrix through matrix multiplication with the weights/portfolio controls from the training period. The weights from the training period are considered as initial weights at the beginning of our test period and hence rebalancing the weights each month accounting for the asset price fluctuations to ensure the investor holds the

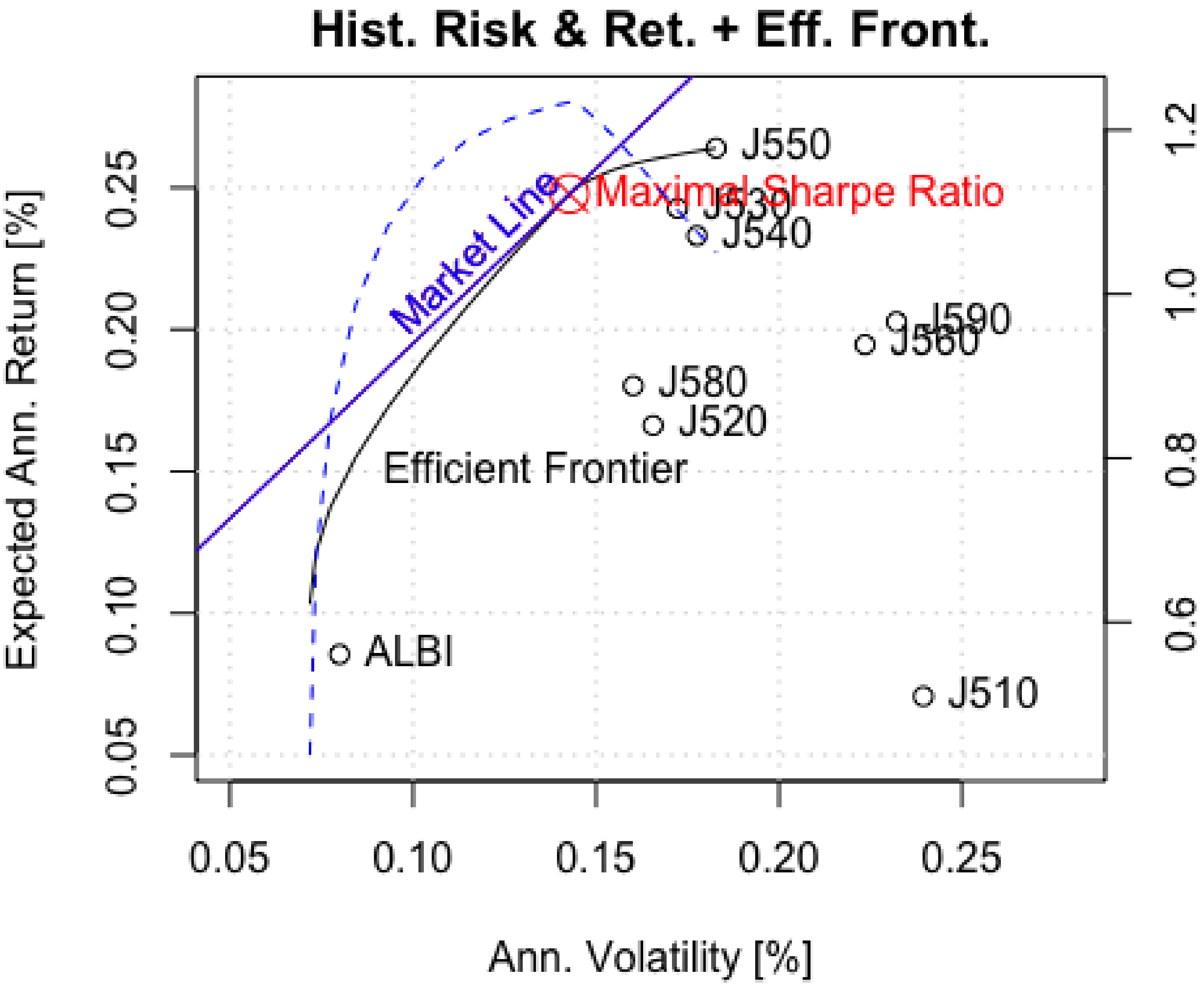

Figure 1: Plot of the IS tangency portfolio showing the maximum Sharpe ratio portfolio.

Table 1: Tangency portfolio showing combination of assets and weights

Asset	**Portfolio control**
ALBI	7.6762e-18
J500	1.0373e-25
J510	1.6156e-17
J520	8.0553e-17
J530	2.9418e-01
J540	2.2595e-01
J550	8.1242e-02
J560	3.9863e-01
J580	9.4885e-18
J590	2.5843e-16

same proportions of assets and the portfolio is still fully invested.

In comparision with the statistics obtained from the training period, the expected portfolio return for the test period was 1.220211 where as the expected portfolio return for the training period was 0.4258322. A brief reason for the difference could be the compounding effect. For the training period no rebalancing was done hence the investors return could be the returns of which he had prior views on. Whereas for the test period, the rebalancing of the portfolio could essentially shift/change an investors prior views of the portfolio. The Shape ratio of the out of sample was computed to be 0.3760202 which is relatively lower than that from the in-sample. One reason could be due to higher risk and another could be due to overfitting in the in-sample period. As asset prices are changing each month, the risk of holding more proportion of particular assets change so over the risk of the portfolio will be higher. In terms of overfiiting, the configurations or controls chosen cold only work well with in-sample data and perform poorly out-of-sample. As nicely put in Bailey[1], a researcher expects to get a Sharpe ratio above 1 in-sample despite the fact that all strategies are expected to deliver a Sharpe ratio of about zero out-of-sample (including an optimal strategy selected in-sample).

Applying constant weights found during the in-sample period on the returns in the test set, a portfolio mean 1.220211 and volatility of 0.4258322 respectively of the portfolio were obtained. The expected portfolio return is exactly the same when rebalancing was considered so is the volatility of the portfolio. Computing the performance as a performance measure, 0.3760202 was obtained. This is identical to the Sharpe ratio from rebalancing the portfolio.

Figure 2 shows plots of the cumulative geometric returns for when re-balancing and no re-balancing is carried out on the test sample as discussed above. The portfolio controls for no re-balancing were those computed in the training period. As can be seen the cumulative geometric returns follow an almost identical pattern to one another.

Hence from this we conclude that there is no major difference between the two methods of carrying out the buy and hold strategy in this case (rebalacing the portfolio controls versus keeping the portfolio controls constant). However, dynamically rebalancing of the portfolio helps an investor manage volatility which may not be picked up if the investor opts to not rebalance.

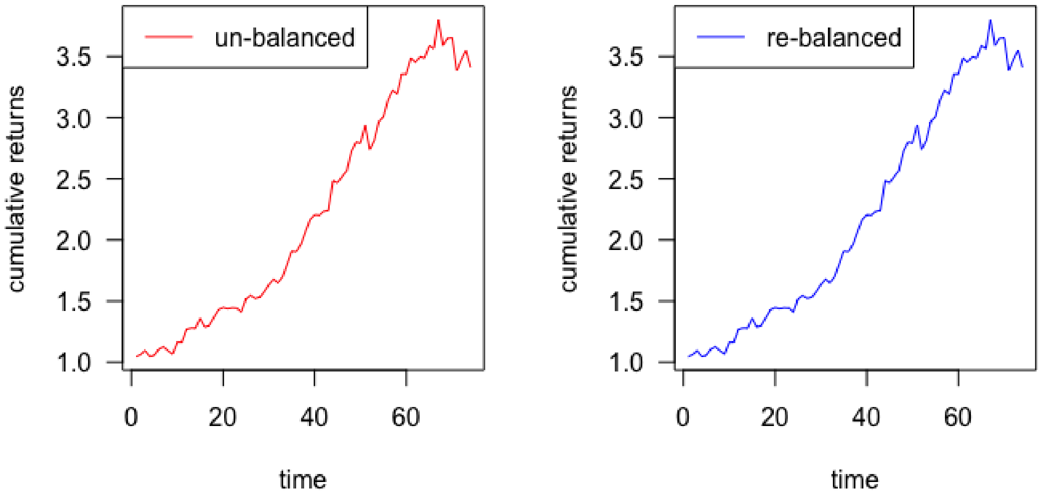

Figure 2: Cumulative return plots considering constant weights and rebalanced weights

3.2 Experiment II: Out-Of-Sample Backtesting using a Rolling Window.

This experiment also implements a buy and hold strategy except this time the strategy is carried out using rolling windows. A rolling window of a month is initially chosen and in each window a Sharpe ratio maximizing portfolio is obtained. In total 74 Sharpe ratio maximizing portfolios were obtained as a total of 74 window periods was considered. Figure 3 shows a time series plot of the monthly portfolio returns.

In the plot we can see that we have a stationary time series as we have a constant mean averaging around 1.03 that is time invariant. The time series shows no apparent trend but it has some noise present mainly due to the volatility of the portfolio. Hence, the time series can be used to forecast portfolio returns for future time periods.

Table 2 gives a summary of the portfolio mean, variance and Sharpe ratio values obtained from the out-of-sample period through implementing monthly windows.

From this we can see that the portfolio mean increase in comparison to the out-of-sample (OOS) portfolio mean from experiment I. The volatility of the portfolio was

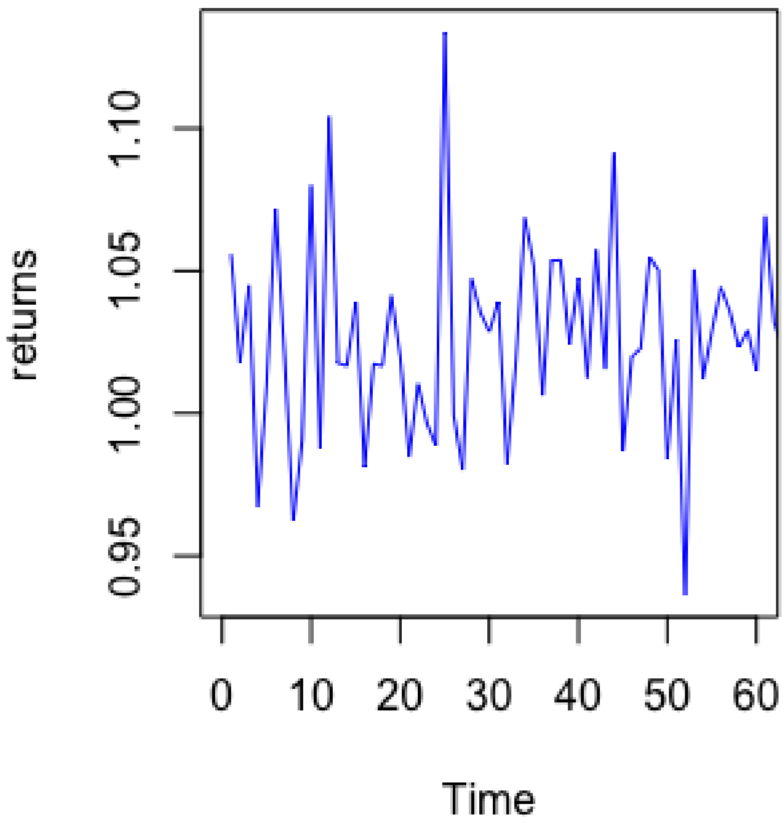

Figure 3: Time Series plot of returns

Table 2: Out-of-sample Portfolio statistics

Portfolio returns	**Portfolio volatility**	**Riskfree asset mean**	**Sharpe ratio**
1.3379	0.6543	1.0612	0.4281

0.4280565 whereas the OOS volatility from experiment I was 0.2551269. This could be due to spikes in volatility experienced in some of the monthly windows. The OOS Sharpe ratio's from both experiments are also different mainly due to the volatility picked up when rolling windows are carried out.

Figure 4 shows the performance of the out-of-sample buy and hold strategies from the two experiments. As can be seen, the performance of the two experiments is almost similar during the first 20 months. After this point, buy and hold strategy implemented with rolling windows (experiment II) out performs the strategy implemented in experiment I. Again, this is confirmed through the Sharpe ratio which indicates that implementing rolling windows on the buy and hold strategy provides higher excess returns for the extra volatility endured for holding the assets throughout the period.

4 Conclusion

A buy and hold strategy was implemented for 10 assets over the same period using two different approaches. From the analysis given above we can see that implementing the strategy using rolling windows provides higher excess returns for level of given risk. This is due to the investor accounting for volatility frequently and hence always rebalancing his portfolio accordingly and thus being rewarded higher excess returns. A Major take away is that the two methods given in experiment I yield the same results whereas experiment II gives another method which which gives a better portfolio performance. Three different trials/configurations (N=3) were used during this backtesting exercise and the trial in experiment II gave the best performance. However, IS and OOS Sharpe ratios were different for this optimal configuration thus favouring arguments seen in Proposition 1 in Bailey[1]. That is, as the researcher tries a growing number of strategy configurations, there will be a non-null probability of selecting IS a strategy with null expected performance OOS.

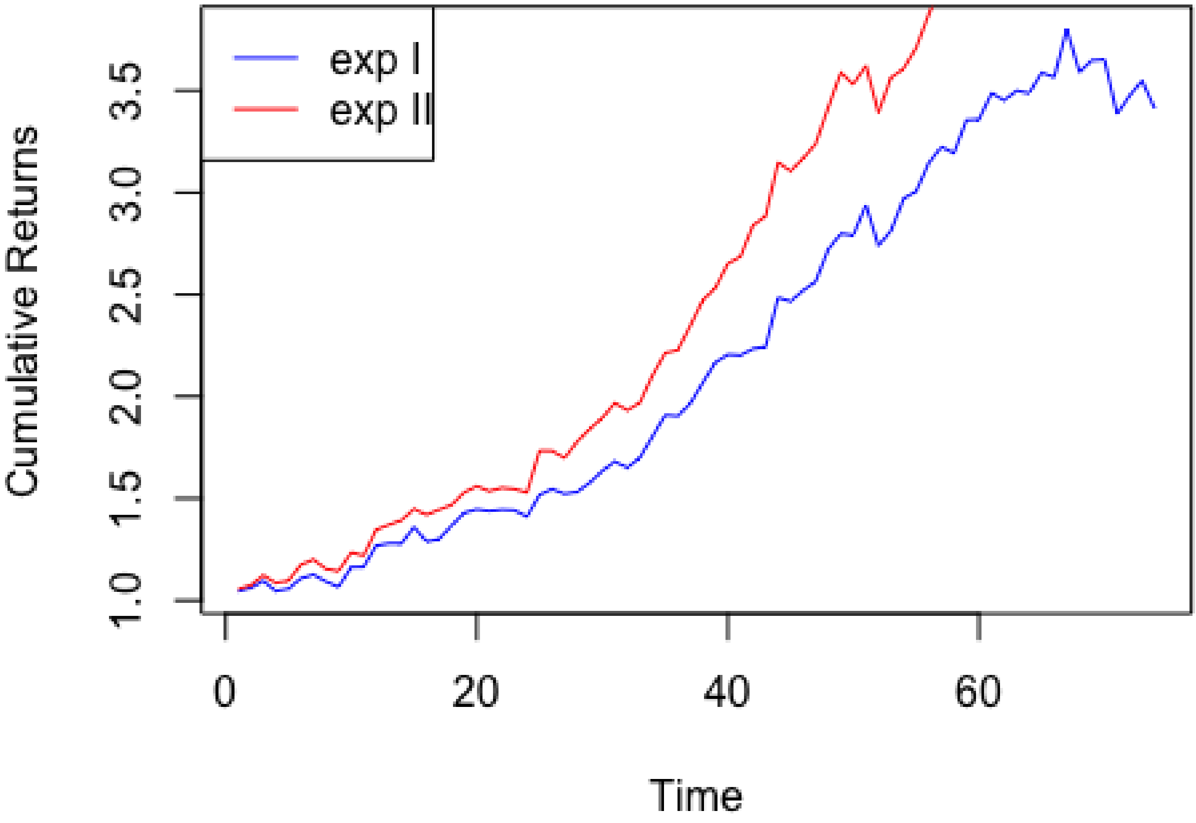

Figure 4: OOS experiment I and experiment II performance

5 References

[1] Bailey. D, Borwein. J, Prado. M and Zhu.Q, Pseudo-Mathematics and financial charlatanism: The effects of backtest overfitting on the out-of-sampe performance. Available at http://www.ams.org/journals/notices/201405/rnoti-p458.pdf
[2] A.Lo, The statistics of Sharpe ratios, Financial Analysts Journal **58** 4 (Jul/Aug, 2002). Available at http://ssrn.com/abstract=377260.
[3] Stewart. T, Gumedze. F and Thiart. C, Probability theory and statisticl inference (STA2004F). (sept, 2012).
[4] S. Resnick, Extreme Values, Regular Variation and Point Processes, Springer, 1987
[5] P. Embrechts, C. Klueppelberg and T. Mikosch, Modelling Extremal Events, Springer-Verlag, New York,2003.
[6] Myriam Charras-Garrido, Pascal Lezaud. Extreme Value Analysis : an Introduction. Journal de la Societe Française de Statistique, Societe Française de Statistique et Societe Mathematique de France, 2013, 154 (2), pp 66-97. ¡hal-00917995¿
[7] Wasserman. L , All of statistics: a coincise course in statistical inference, 2004.

6 Appendix

Source : Wasserman.L [7]

Theorem16 : Let $X_1, ..., X_n$ be random variables. Suppose there exists $\sigma > 0$ such that $E(e^{tX_i}) \leq e^{t^2\sigma^2/2}$ for all positive t. Then

$$E(\max X_n) \leq \sigma\sqrt{2log(n)}$$

Proof :
By Jensen's inequality, for $1 \leq i \leq n$

$$\begin{aligned}\exp\left\{tE(\max\{X_i\})\right\} &\leq E\left(\exp\left\{t\max\{X_i\}\right\}\right)\\ &= E(\max\exp\{tX_i\}) \leq \sum_{i=1}^{n} E(\exp\{tX_i\}) \leq ne^{t^2\sigma^2/2}\end{aligned}$$

Thus,

$$E(\max X_i) \leq \frac{log(n)}{t} + \frac{t\sigma^2}{2}$$

The result follows by setting t = $\sqrt{2log(n)}/\sigma$.

```r
## Load the Tactical Asset Allocation data from EXCEL
# Filename: JSE-IND-S10-RCPP.R (Demonstrates use of C++ for big-data)
# See Also: JSE-IND-S10-RJAVA.R (Demonstrates use of JVM for big-data)
#
# 1. ICB Industrial Level Indices
# 2. ALBI (All Bond Index (ALBI) Total Return Index (TRI) Data)
# 3. Money Market Data:  JIBAR and STEFI TRI
# 4. Various Indices: JSE Growth, JSE Value, JSE ALSI, JSE SRI
#
# Situation: Load data from *.csv file and convert into timeSeries object data
#
# Big-Data Issues to consider:
#
#  A: xlsx using rJava (set the Heap size correctly and tune GC)
#  B: openxlsx using rcpp [this file]
#
# This script file address and demonstrates option B

# Author : T Gebbie 2017

## 0. Clearr environemnt and remove all plots
rm(list=ls()) # clear environment
# if(!is.null(dev.list())) dev.off() # remove all RStudio plots

## 2. Packages
# 2.1.  install.packages("<name of the package>")
# 2.2.  library("<name of the package>")
# 2.3.  any(grepl("<name of your package>",
#       installed.packages()))
## Use the openxlsx package (Rcpp)
install.packages("openxlsx")
install.packages("Rcpp")
install.packages("timeSeries")
library(openxlsx)
library(Rcpp)
library(zoo)
library(xts)
library(timeSeries)
library(rbenchmark)
library(nloptr) # for SQP
library(quadprog) # for QP
## 3. Paths
# 3.1.  setwd("<location of your dataset>")
rootp0 <- getwd()
setwd("..") # move up one level in the directory tree
rootp <- getwd() #set root path
setwd(path.expand('~'))
filen <- "PT-TAA-JSE-Daily-1994-2017.xlsx"
fpath <- "/Bongz/Documents/Portfolio theory/"
ffilen  <- paste(rootp,fpath,filen,sep="")

## 4. load the dataset by sheet
#importing data in excel sheet with 4 spread sheets
```

```
dfS <- list()
for (i in 1:4){
  dfS[[i]] <- read.xlsx(ffilen, sheet = i,detectDates = TRUE)
}
dim(dfS[[3]]) #setting into a dataframe
# if you need to convert column `A` to date then use
# df$A <- as.POSIXct(df$A,format="%H:%M:%S")

## 5. Keep only the specified list of Tickers
Entities = c('X1','STEFI','ALBI','J203','J500',sprintf("J5%d",seq(10,90,by = 10)))
Items    = c('Date','TRI','Stefi')
# find Tickers in colnames, and
# TRI at the attribute type and reference and join
for (i in c(1,2,3,4)){
  # logical FALSE vector
  tI0 <- logical(length = length(colnames(dfS[[i]]))) #Vector of falses
  tI1 <- tI0 #Vector of falses
  # find the Entities in the data frame
  for (j in 1:length(Entities)){
    tI0 <- tI0 | grepl(Entities[j],colnames(dfS[[i]]))
  }
  # find the Items in the data frame
  for (k in 1:length(Items)){
    tI1 <- tI1 | grepl(Items[[k]],dfS[[i]][2,])
  }
  # combined the logical indices
  tI <- tI0 & tI1
  # remove the columns not required
  dfS[[i]] <- dfS[[i]][,tI]
  # remove the first two rows (as they are not dates)
  dfS[[i]] <- dfS[[i]][-c(1,2),]
  # rename the first column name to Dates
  names(dfS[[i]])[1] <- "Date"
  # clean up the remaining column names
  newColNames <- strsplit(colnames(dfS[[i]]),":")
  for (m in 2:length(newColNames)){
    names(dfS[[i]])[m] <- newColNames[[m]][1]
  }
}
# get the dimensions of the remaining data-frame list elements
for (i in 1:length(dfS)){
  print(dim(dfS[[i]])) # throw to console
}

## 6. Clean and convert into a single timeSeries object
# 6.1. Initialise the timeSeries object with first data frame
iN <- 1
tsTAA <- timeSeries(dfS[[iN]][,2:ncol(dfS[[iN]])],as.Date(dfS[[iN]][,1]))
print(dim(tsTAA)) # print dimensions to console
#  correct the column names
# 6.3. Concatenate additional timeSeries columns on to the object
for (i in c(2,3,4)){
  # consider iterative merging using inherited zoo properties
  # the first column is the Date column the rest are features we do this
  # to ensure that the dates are correctly aligned when time series are merged
  tsTAA <- cbind(tsTAA,timeSeries(dfS[[i]][,2:ncol(dfS[[i]])],as.Date(dfS[[i]][,1])))
  print(dim(tsTAA))
  print(colnames(tsTAA))
}
# 6.4 Set the units to TRI
setFinCenter(tsTAA) <- "Johannesburg"
# 6.5 Fix the colname errors introduce during the cbind
names(tsTAA)[grep("TS.1.1",names(tsTAA))] <- "ALBI"
names(tsTAA)[grep("TS.1.2",names(tsTAA))] <- "STEFI"
names(tsTAA)[grep("TS.1",names(tsTAA))] <- "ALSI"

## 7. Convert from Daily Sampled Data to Monthly Sampled Data
# 7.1. Decimate the daily data to monthly data
tsTAA <- daily2monthly(tsTAA)  #orders your dates chronologically
# 7.2 Visualise the data on a single plot
#   combine and visualise and prettify the y-axis
plot(tsTAA,plot.type = c("single"),
     format = "auto",
     at=pretty(tsTAA),
     ylab = "Returns",
     main = "TRI for sectors", las = 1)

## 8. Compute returns Geometric (Arithmetic)
# We need to manage the missing data NA (Not A Number)
# 8.1 remove all header NA data first
# omit NA
tsTAA <- na.omit(tsTAA) # but we will revisit this in the next section
#   remove all rows with NAs and compute the index

tsIdx <- index2wealth(tsTAA)
#   ensure that the date range is complete
#   explicitly compute the daily geometric returns

tsGRet <- diff(log(tsIdx)) #differences a time series

## 6. Plot two plots on the same figure
#   matrix of figures
par(mfrow=c(2, 1))
#   set the scaling for the first graph
par(mar = c(bottom=1.5, 5.1, top=4, 2.1))
#   plot the single set of time-series
plot(tsIdx,plot.type = c("single"),
     format = "auto",
     ylab = "Price",
     main="Monthly Price Index",
     cex.main=0.7, cex.lab=0.7, cex.axis=0.7, las = 1,xlab = "Time/Date")
#   include the legend
legend("bottomleft",names(tsIdx),cex=0.7)
#   set the scaling for the second graph
par(mar = c(bottom=4, 5.1, top=1.5, 2.1))
#   plot the time-series
```

```r
plot(tsGRet,plot.type = c("single"),
     format="%B\n%Y",
     at=pretty(tsGRet),
     ylab = "Returns",
     main = "Monthly Sampled Geometric Returns",
     cex.main=0.7, cex.lab=0.7, cex.axis=0.7)
#   include the legend
legend("bottomleft",names(tsGRet),cex=0.7)

## 11. Save the workspace and with the prepared data
save(tsGRet,tsTAA,tsIdx,file = "~/Documents/Portfolio theory/PT-TAA.RData")
save.image
unlink("PT-TAA.RData")
unlink(".RData")


#EXPERIMENT 1

##########################################################

load(file = "~/Documents/Portfolio theory/PT-TAA.RData") #ALREADY PREPROCEESED DATA

Entities <- colnames(tsGRet)
# remove the money market assets (we will compute excess returns!)
#our tickers
Entities <- Entities[-c(grep('STEFI',Entities))]
Entities <- Entities[-c(grep('ALSI',Entities))]

#splitting the data
ind <- 1:74
train.sample <- tsGRet[ind,]
test.sample <- tsGRet[-ind,]

#extract risk free rates (stefi) for the training and test period
train.RF <- train.sample[,"STEFI"] #risk free rate for training period.
test.RF <- test.sample[,"STEFI"] #risk free for test period.

#final test and training samples that include only our tickers
train.sample <- train.sample[,Entities]
test.sample <- test.sample[,Entities]

#compute geometric means
mean.returns <- colMeans(train.sample, na.rm = TRUE)
sd.returns <- colStdevs(train.sample, na.rm = TRUE)
var.returns <- var(train.sample, na.rm = TRUE)
risk.free.train <- colMeans(train.RF, na.rm = TRUE)
any(is.na(train.sample)) #checking if we still have missing values

#annualize data
mean.returns <- mean.returns*12
sd.returns <- sqrt(12)*sd.returns
var.returns <- var.returns*12
risk.free.train <- risk.free.train*12
```

```r
##############################
#plot
plot(sd.returns, mean.returns,
     ylab = "Expected Ann. Return [%]", xlab = "Ann. Volatility [%]",
     main = "Hist. Risk & Ret. + Eff. Front.", xlim = c(0,0.3))
grid()
text(sd.returns, mean.returns,labels = names(mean.returns), cex= 1, pos = 4)

##############################

#tangency portfolio
one.vec <- rep(1,length(mean.returns))
init.wts <- one.vec / length(one.vec)
IS.weights <- matrix(NA,1,length(mean.returns))


sharpe <- function(x) {
  return(-(x %*% mean.returns - risk.free.train) / sqrt(x %*% var.returns %*% x))
}

#ensuring fully invested
constraint <- function(x) {
  return(x%*%one.vec - 1)
}

#make use of sqp to solve for tangency portfolio
soln <- slsqp(init.wts, fn = sharpe, gr = NULL, # target returns
               lower = rep(0,length(init.wts)), # no short-selling
               upper = rep(1,length(init.wts)), # no leverage
               heq = constraint, # fully invested constraint function
               control = list(xtol_rel = 1e-8)) # SQP
IS.weights <- soln$par
print(IS.weights)

#compute the sharpe ratio for this period
portfolio.return <- IS.weights %*% (mean.returns)
#portfolio returns for buy and hold period
portfolio.volatiliy <- IS.weights %*% var.returns %*% IS.weights
#portfolio variance for buy and hold period
portfolio.sharp <- (portfolio.return-risk.free.train)
/ sqrt(portfolio.volatiliy)

#TEST DATA
#new period, observations 75 - 148
#exponentiate the returns from the test period.
#These will give the relative % changes from the beignning of the period to end.
compound.returns <- exp(test.sample)
test.RF <- exp(test.RF)

#loop through
#get out a weight matrix
OOS.weights <- matrix(rep(NA), nrow(test.sample),length(names(test.sample))) #initialize
OOS.weights[1,] <- IS.weights #set intial, in-sample weights in the weight matrix
```

```
for(i in 2:nrow(test.sample)){
  OOS.weights[i,] <- as.numeric(compound.returns[i,] * OOS.weights[i-1,])
  / as.numeric(compound.returns[i,]) %*% OOS.weights[i-1,]

}

###########################
#cumprod(rtrns.test)
#fts <-timeSeries(rtrns.test)
#plot(ts)
############################

portfolio.returns.test <- matrix(NA,74,1)
for (i in 1:nrow(compound.returns)) {
  portfolio.returns.test[i] <- OOS.weights%*%t(compound.returns[i,])
}

portfolio.returns.test <- timeSeries(portfolio.returns.test)
plot(cumprod(portfolio.returns.test), type ="l", col = "blue",
     ylab = "Cumulative Returns", xlab ="Time")

#annualized
portfolio.returns.prod <- prod(portfolio.returns.test^(12/74))
portfolio.variance.test <- var(portfolio.returns.test*(12))
test.RF <- prod(test.RF^(12/74)) #annualized

portfolio.test.SR <-
  (portfolio.returns.prod - test.RF) / sqrt(portfolio.variance.test)

#plots
par(mfrow = c(1,2))
wts4 <- compound.returns%*%as.matrix(IS.weights)
plot(cumprod(wts4) ,col = "red",type = "l",
     ylab = "cumulative returns", las =1
     ,xlab = "time")
legend("topleft",legend = c("uncompounded"), col =c("red"), lty = 1)
plot(cumprod(portfolio.returns.test), type ="l",
     col = "blue", ylab = "cumulative returns",las =1
     ,xlab = "time")
legend("topleft",legend = c("compounded"), col =c("blue"), lty = 1)

#time series plot
plot.ts(portfolio.returns.test, ylab = "Returns", col = "red")

#constant sharpe ratio weights
wts4 <- compound.returns%*%as.matrix(IS.weights)
returns.constant <- prod(wts4^(12/74))
vol.constant <- var(wts4*12)
sr.constant <- (returns.constant-test.RF)/sqrt(vol.constant)
#GIVES A SHARPE RATIO OF 0.376

#EXPERIMENT 2
```

```
#experiment 2
# LOAD ALREADY PREPROCEESED DATA
load(file = "~/Documents/Portfolio theory/PT-TAA.RData")

Entities <- colnames(tsGRet)
# remove the money market assets (we will compute excess returns!)
#our tickers
Entities <- Entities[-c(grep('STEFI',Entities))]
Entities <- Entities[-c(grep('ALSI',Entities))]

ind <- 1:73
dat <- na.omit(tsGRet[,Entities])
dat.RF <- tsGRet[,"STEFI"]
test.sample <- dat[-ind,]
test.RF <- tsGRet[-ind,"STEFI"] #stefi for training

#IS.weights <- matrix(NA,74,10)
rolling.window <- function(x){

  newdat <- dat[x,]
  rf <- dat.RF[x,]
  newdat <- na.omit(newdat)
  mean.returns <- colMeans(newdat, na.rm = TRUE)
  sd.returns <- colStdevs(newdat, na.rm = TRUE)
  var.returns <- var(newdat, na.rm = TRUE)
  risk.free.train <- colMeans(rf, na.rm = TRUE)

  #annualize
  mean.returns <- mean.returns*(12)
  sd.returns <- sqrt(12)*sd.returns
  var.returns <- var.returns*(12)
  risk.free.train <- risk.free.train*(12)

  one.vec <- rep(1,length(mean.returns))
  init.wts <- one.vec / length(one.vec)
  IS.weights <- matrix(NA,1,length(mean.returns))

  sharpe <- function(x) {
    return(-(x %*% mean.returns - risk.free.train)
           / sqrt(x %*% var.returns %*% x))
  }

  #ensuring fully invested
  constraint <- function(x) {
    return(x%*%one.vec - 1)
  }

  #make use of sqp to solve for tangency portfolio
  soln <- slsqp(init.wts, fn = sharpe, gr = NULL, # target returns
                lower = rep(0,length(init.wts)), # no short-selling
                upper = rep(1,length(init.wts)), # no leverage
                heq = constraint, # fully invested constraint function
                control = list(xtol_rel = 1e-8)) # SQP
```

```r
  IS.weights <- soln$par
  #print(IS.weights)
  portfolio.return <- IS.weights %*% mean.returns
  #portfolio returns for buy and hold period
  portfolio.volatiliy <- IS.weights %*% var.returns %*% IS.weights
  #portfolio variance for buy and hold period
  portfolio.sharp <- (portfolio.return-risk.free.train)
  / sqrt(portfolio.volatiliy)
  #return(portfolio.sharp)
  return(IS.weights)
}

compound.returns <- exp(test.sample)
test.RF <- exp(test.RF)
IS.weights <- matrix(NA,74,10)
rtrns.vec <- c(rep(0,74))
for (i in 1:74) {
  IS.weights[i,] = rolling.window(i:(73+i))
  rtrns.vec[i] <- IS.weights[i,]%*%t(compound.returns[i,])
}



portfolio.returns.prod <- prod(rtrns.vec^(12/74))
portfolio.variance.test <- var(rtrns.vec^(12))
test.RF <- prod(test.RF^(12/74)) #annualized

plot.ts(rtrns.vec, ylab = "returns",col ="blue",xlim =c(0,60))
#time series of the performance
par(mfrow = c(1,1))

portfolio.test.SR <- (portfolio.returns.prod - test.RF)
/ sqrt(portfolio.variance.test)



lines(cumprod(rtrns.vec), col ="red")
legend("topleft", legend = c("exp I", "exp II"),
       col = c("blue","red"), lty = 1)
```