The Buffet Bet - Basket of Funds vs. All Passive

Stats 107 Final Project

David Wihl

December 6, 2016

Abstract

Ten years ago, Warren Buffett made a famous : the ultimate passive investment, the Vanguard S&P 500 Index Fund Admiral (VFIAX), vs. a basket of funds determined by a team of hedge fund experts. Whoever had a lower return after ten years would donate $1 million to the charity of choice of the winner. After eight years, it looks increasingly likely that Buffett will win the bet.

Type: Simulating a trading strategy.

Introduction

Yahoo only has data on VFIAX starting 2000-11-13.

Quantmod does not use adjusted price to calculate returns. So, basically, it is wrong. I found this going through the quantmod sources and then confirmed it at http://stackoverflow.com/questions/34772616/r-function-periodreturn-not-computing-returns-using-adjusted-closing-prices

Must use OHLC numbers only

Methods

Importing List of Vanguard Mutual Funds

Summary Statistics

Table 1: Best Sharpe Ratio

Ticker	Fund.Name	Expenses	CAGR	Sharpe
VSCSX	Vanguard Short Term Corp Bd Index Adm	0.10	0.03	0.11
VFSTX	Vanguard Short-Term Investment Grade Inv	0.20	0.04	0.11
VTABX	Vanguard Total Intl Bond Index Adm	0.14	0.04	0.11
VSGBX	Vanguard Short-Term Federal Inv	0.20	0.04	0.10
VTIBX	Vanguard Total Intl Bond Index Inv	0.17	0.04	0.10
VFISX	Vanguard Short-Term Treasury Inv	0.20	0.03	0.10

Table 2: Best CAGR

Ticker	Fund.Name	Expenses	CAGR	Sharpe
VSIAX	Vanguard Small-Cap Value Index Adm	0.08	0.16	0.03
VMVAX	Vanguard Mid-Cap Value Index Adm	0.08	0.16	0.07
VSGAX	Vanguard Small-Cap Growth Index Adm	0.08	0.13	0.05
VMGMX	Vanguard Mid-Cap Growth Index Adm	0.08	0.13	0.05
VEVFX	Vanguard Explorer Value Inv	0.65	0.12	0.04

Ticker	Fund.Name	Expenses	CAGR	Sharpe
VGSIX	Vanguard REIT Index Inv	0.26	0.11	0.03
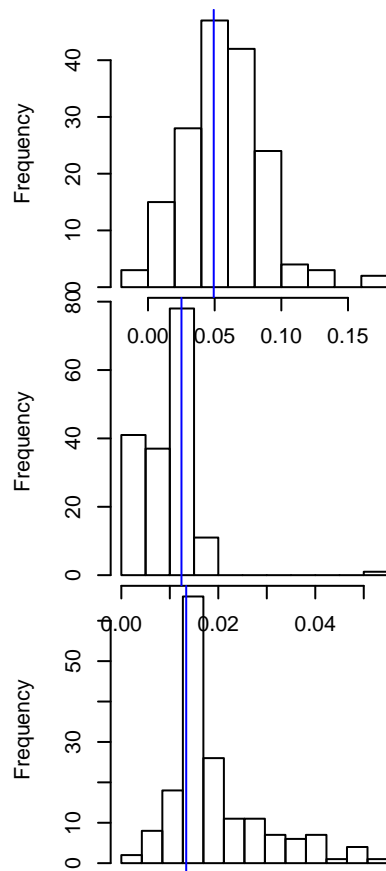
Results

Conclusions and Discussions

References

This entire project code including source data can be found on GitHub (https://github.com/wihl/stats107-project)

Appendix A - Code

Preparing Report

The following is the code used to prepare this report:

```r
# Load libraries
library(ggplot2)
library(quantmod)
library(knitr)
#
# Import List of Securities
#
loadRData = function(filename) {
  # Source: http://stackoverflow.com/questions/5577221/how-can-i-load-an-object-into-a-variable-name-th
  load(filename)
  get(ls()[ls()!="filename"])
}
fromDate = "2010-11-13"
toDate   = "2016-11-13"
funds    = read.csv("data/vanguard.csv",header = T, stringsAsFactors = F)
# Load from previously cached download
stockDataEnv= loadRData("data/stockData.RData")
fundData = mget(funds$Ticker,stockDataEnv)
#
# Generate Summary Statistics
#
for (i in 1:nrow(funds)) {
  sym = funds$Ticker[i]
  data = eval(parse(text=paste("fundData$",sym,sep="")))
  if(!is.null(data)){
    # we have the stockdata; Calculate summary statistics
    funds$startDate[i] = as.Date(index(data[1]))
    funds$endDate[i] = as.Date(index(last(data[])))
    startPrice = as.numeric(Ad(data[1]))
    endPrice = as.numeric(Ad(last(data[])))
    funds$totalRet[i] = (endPrice - startPrice) / startPrice
    dailyRets = dailyReturn(Ad(data))
    funds$stdDev[i] = sd(dailyRets)
    funds$avgRet[i] = mean(dailyRets)
    funds$Sharpe[i] = funds$avgRet[i] / funds$stdDev[i]
    # CAGR
    years =  as.numeric((as.Date(funds$endDate[i]) -
                         as.Date(funds$startDate[i])))/365
    funds$CAGR[i] = ((endPrice / startPrice)^(1/years)) - 1


  }
}
#
# Generate Tables of Top Performers
#
colsToDisplay = c("Ticker","Fund.Name","Expenses","CAGR","Sharpe")
kable(head(funds[order(-funds$Sharpe),colsToDisplay]), caption="Best Sharpe Ratio",
      row.names = F,digits=2)

kable(head(funds[order(-funds$CAGR),colsToDisplay]), caption="Best CAGR",
      row.names = F,digits=2)


#
```

```r
# Show histograms with base VFIAX case
#
par(mfrow=c(3,1),pin=c(1.5,1.5))

# CAGR
hist(funds$CAGR,main="CAGR (all funds)",xlab="")
abline(v=funds[funds$Ticker=="VFIAX",]$CAGR,col="blue")

# Standard Deviation
hist(funds$stdDev,main="Standard Deviation (all funds)",xlab="")
abline(v=funds[funds$Ticker=="VFIAX",]$stdDev,col="blue")

# Sharpe Ratio
hist(funds$Sharpe,main="Sharpe Ratio (all funds)",xlab="")
abline(v=funds[funds$Ticker=="VFIAX",]$Sharpe,col="blue")
##
##
```

Caching Stock Data

Stock data was downloaded and cached with the following code:

```r
# Download and cache stock symbol data
# Inspired from http://gekkoquant.com/2012/06/01/stock-data-download-saving-r/
library(quantmod)
fromDate = "2000-11-13"
toDate   = "2016-11-13"
funds    = read.csv("data/vanguard.csv",header = T,
                    colClasses = c("character","character","numeric"))
stocksLst = funds$Ticker
savefilename <- "data/stockdata.RData" #The file to save the data in
startDate = as.Date(fromDate) #Specify what date to get the prices from
maxretryattempts <- 5 #If there is an error downloading a price how many times to retry

#Load the list of ticker symbols from a csv, each row contains a ticker
stockData <- new.env() #Make a new environment for quantmod to store data in
nrstocks = length(stocksLst) #The number of stocks to download

#Download all the stock data
for (i in 1:nrstocks){
  for(t in 1:maxretryattempts){

    tryCatch(
      {
        #This is the statement to Try
        #Check to see if the variables exists
        #NEAT TRICK ON HOW TO TURN A STRING INTO A VARIABLE
        #SEE  http://www.r-bloggers.com/converting-a-string-to-a-variable-name-on-the-fly-and-vice-vers
        if(!is.null(eval(parse(text=paste("stockData$",stocksLst[i],sep=""))))){
          #The variable exists so dont need to download data for this stock
          #So lets break out of the retry loop and process the next stock
          #cat("No need to retry")
          break
```

```r
      }

        #The stock wasnt previously downloaded so lets attempt to download it
        cat("(",i,"/",nrstocks,") ","Downloading ", stocksLst[i] , "\t\t Attempt: ", t , "/", maxretrya
        getSymbols(stocksLst[i], env = stockData, src = "yahoo",
                   from = startDate, to=toDate, adjust=T)
      }
      #Specify the catch function, and the finally function
      , error = function(e) print(e))
  }
}


#Lets save the stock data to a data file
tryCatch(
  {
    save(stockData, file=savefilename)
    cat("Sucessfully saved the stock data to %s",savefilename)
  }
  , error = function(e) print(e))
```