Pair Trading Strategy

Part 1: Static analysis

Pair Selection

```
library(xts)
da <- read.csv("Original data set.csv",sep = "")
data <- xts(da[-1], order.by = as.Date(da$Date))
n <-  ncol(data)
```

The universe of stock contains 62 stocks samples. Our choice of pair is CHEVRON and EXXON.MOBILE. The two stocks have relatively high correlation and both come from financial sector. Companies in this sector usually have more homogeneous operations and earnings, thus we may capture trading opportunities. Also, according to APT, to stay sector neutral means that they are more likely to be exposed to similar risk factors. We prefer log price due to its nice mathematical property such as additive when calculating the return. Additionally, our group program a code to aid in stock selection. The selection rule is simple: the main principal is that the two stock should pass the co-integration test. Secondly, the smaller the half time of mean reverting , the better the choice. Finally, betas of CHEVRON and for EXXON.MOBILE are similar (1 and 1.32, respectively), which shows it would be easy for us to stay market neutral.

The code of choosing appropriate pair stock from stock universe is as follows:choosing pair using condition that half life <50 and co-integration satisfies:

```
#Pair Selection code
DFtest <- function(res){
  diff.res <- diff(res)
  lm <- lm(diff.res~res[1:(length(res)-1)])
  std.error <- summary(lm)[["coefficients"]][2,2]
  df <- lm$coefficients[2]/std.error
  if (df< -3.43){
    return(1)
  }
  else{
    return(0)
  }
}
  #Data input
  da <- read.csv("Original data set.csv",sep = "")
  data <- exp(da[-1])
  data <- cbind(Date=as.Date(da$Date),data)
  library(egcm)
  library(urca)
  halfLife <- matrix(0,nrow=63,ncol=63)
  rho =1
  for (i in (2:63)){
    for (j in (i:63)){
      rho <- egcm(data[,i],data[,j])$rho
      halfLife[i,j] <- abs(log(2)/log(rho))
    }
  }
  result <-  matrix(0,nrow=63,ncol=63) #test the existence of cointegration
  for (i in (2:63)){
```

```r
    for (j in (i:63)){
      res <- lm(data[,i]~data[,j])$residuals
      if(DFtest(data[,i]) == 0 & DFtest(data[,j]) == 0){result[i,j]= DFtest(res)}
    }
  }
  for (i in (2:63) ){
    result[i,i]=0
    halfLife[i,i]=0
  }
  pairSelect <- matrix(0,nrow = 63,ncol=63)
  for (i in (2:63)){
    for (j in (i:63)){
      if (result[i,j] == 1 & halfLife[i,j]< 50){
        pairSelect[i,j] = 1
      }
    }
  }
  write.csv(pairSelect,"pair selection result.csv")
```

Test for unit root

```r
asst1 <- data$CHEVRON
asst2 <- data$EXXON.MOBIL

library(urca)
af_CHEVRON <- ur.df(asst1,type = "drift",lags = 0)
adf_CHEVRON <- ur.df(asst1,type = "drift",lags = 4)
af_CHEVRON
```

```
##
## ###############################################################
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ###############################################################
##
## The value of the test statistic is: -1.8813 2.4499
```

```r
adf_CHEVRON
```

```
##
## ###############################################################
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ###############################################################
##
## The value of the test statistic is: -1.7212 2.4806
```

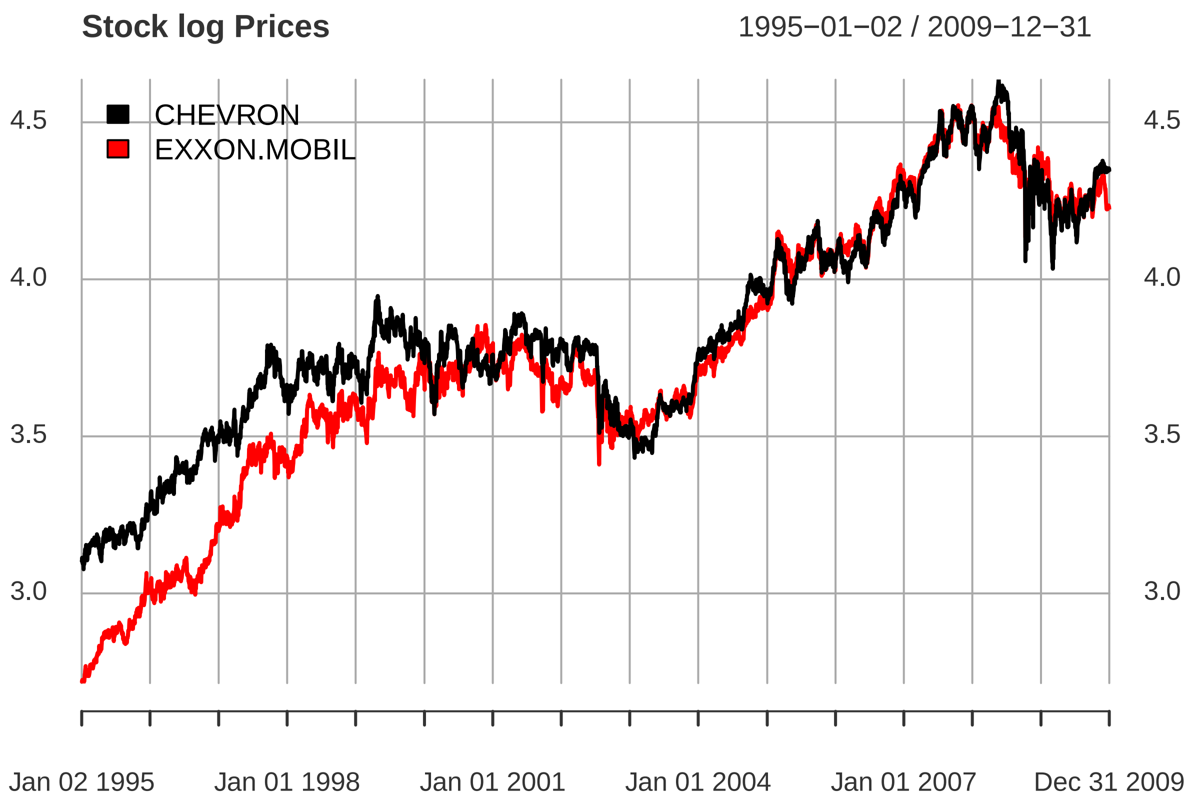
```r
af_EXXON.MOBIL <- ur.df(asst2,type = "drift",lags = 0)
adf_EXXON.MOBIL <- ur.df(asst2,type = "drift",lags = 4)
af_EXXON.MOBIL
```

```
##
## ###############################################################
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ###############################################################
##
## The value of the test statistic is: -2.07 3.1946
```

```r
adf_EXXON.MOBIL
```

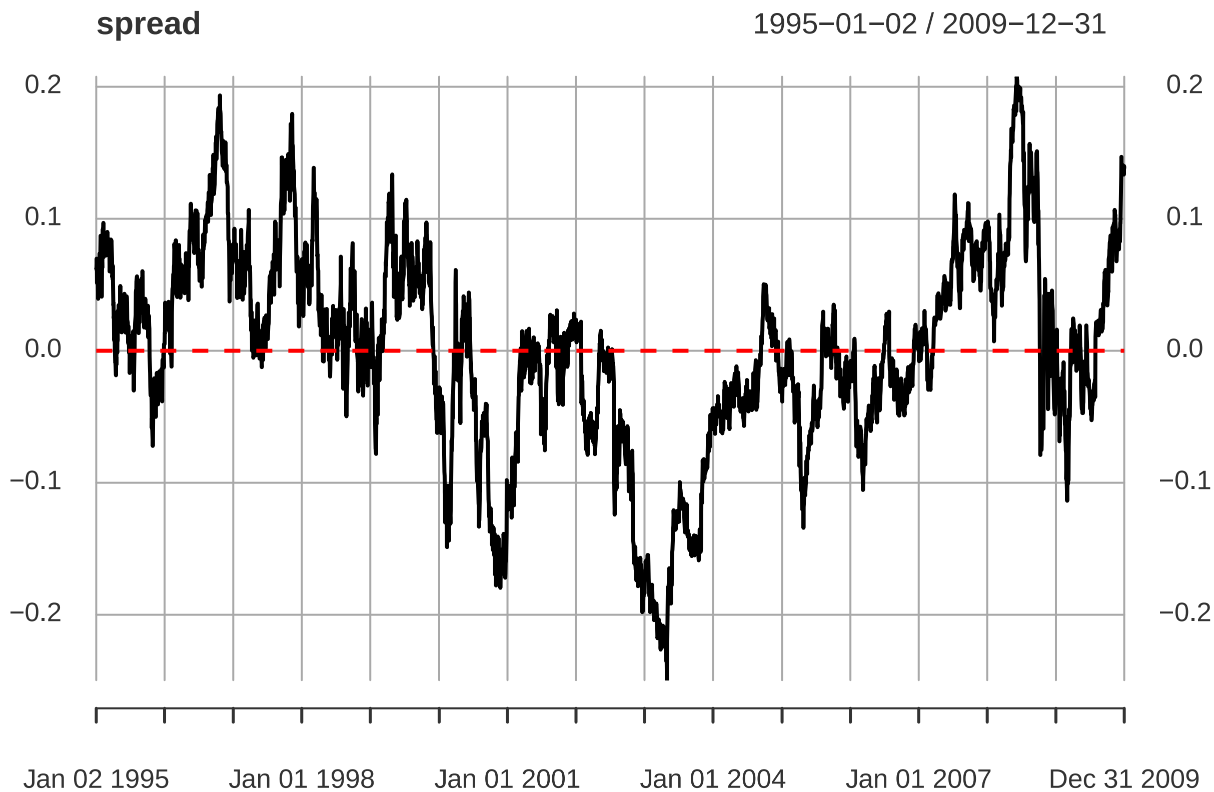
```
##
## ###############################################################
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ###############################################################
##
## The value of the test statistic is: -2.0684 3.8878
```

```r
plot(cbind(asst1, asst2), main="Stock log Prices", legend.loc ="topleft")
```



linear regression with choosend pair

```r
linearMod <- lm(asst1~asst2)
ls_coeffs <- coef(linearMod)
mu <- ls_coeffs[1]
gamma <- ls_coeffs[2]
spread <- asst1 - mu -gamma*asst2
colnames(spread) = "spread"
{plot(spread)
lines(xts(rep(0, nrow(asst1)), index(asst1)), col ="red", lwd = 2, lty = 2)}
```




```
adf_res <- ur.df(spread, type = "drift")
adf_res # see whether pass the unit root test
```

```
##
## ###############################################################
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ###############################################################
##
## The value of the test statistic is: -3.7785 7.1462
```

Performance measure

Long one spread means long one asset 1 and short beta asset 2 (long spread when spread is less than 0, i.e. the asset 1 is undervalued and asset 2 is overvalued).
The strategy is described as below:
If the spread rises higher than 1.5 standard deviation, short spread (you are betting that spread will finally move back to its long term equilibrium and revert to its zero mean). From risk management perspective, if the spread rises too much so that it exceeds the 2.5 standard deviation, the decision is to have a stop loss order. Similarly, when spreads falls below the -1.5 volatility, we start the trade – long spread. In the case that it falls under the 2.5 sigma line, we chose to close the position.

Code for generating the trading signal

```
longTermMean = linearMod$coefficients[1]
cointegrationRatio =  linearMod$coefficients[2]
sigma = sd(spread)
ndays = nrow(asst1)
```

```r
order <- rep(NA, ndays) #record trade order(long, short, or no action)
starting <- "a"
{for (i in (1:ndays)) {
    if (starting == "a" || starting == "c"){
      if (spread[i]> 1.5*sigma & spread[i]< 1.7*sigma){
        starting = "b"
        order[i] = "Short Spread"
      }
    }
    else if(starting == "b"){
      if (spread[i]> 2.5*sigma){
        starting = "c"
        order[i] = "Stop Loss Order"
      }
      else if (starting == "b"){
        if (spread[i] < 0){
        order[i] = "Close Short Position"
        starting = "a"
        }
      }
    }
  }
  starting <- "a"
  for (i in (1:ndays)) {
    if (starting == "a" || starting == "c"){
      if (spread[i]< -1.5*sigma & spread[i]> -1.7*sigma){
        starting = "b"
        order[i] = "Long Spread"
      }
    }
    else if(starting == "b"){
      if (spread[i]< -2.5*sigma){
        starting = "c"
        order[i] = "Stop Loss Order"
      }
      else if (starting == "b"){
        if (spread[i] > 0){
          order[i] = "Close Long Position"
          starting = "a"
        }
      }
    }
  }
}
```

Store the trading action and strategy return in “pairResult” variable

```r
data <- as.data.frame(cbind(asst1, asst2, spread))
data <- cbind("Index" = (1: ndays), "Date" = as.Date(da$Date), data, order)
pairResult <- subset(data, !is.na(data$order))
```

Calculate return for this strategy

```r
Return <- rep(NA,nrow(pairResult))
for (i in (1:(nrow(pairResult)-1))){
```

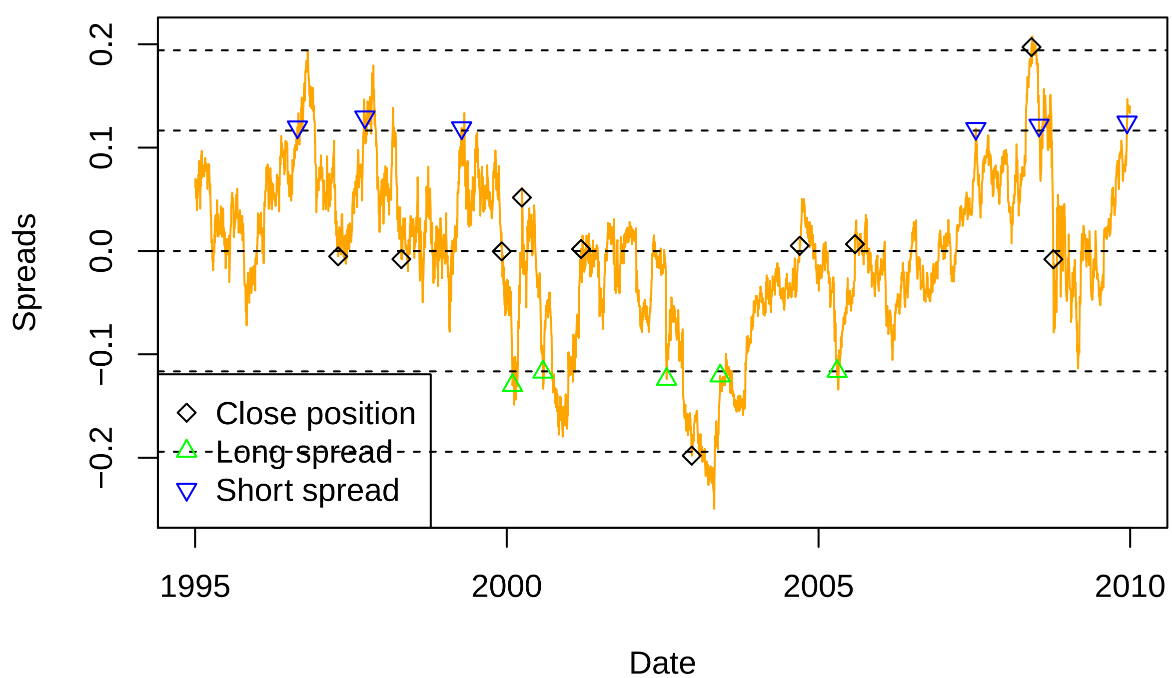
```r
  if (pairResult$order[i] == "Long Spread"||pairResult$order[i] == "Short Spread"){
    if(pairResult$order[i+1] == "Stop Loss Order"){
      Return[i+1]= -abs(diff(pairResult$spread)[i])/
        (pairResult$Index[i+1]-pairResult$Index[i])*252
    }
    else{
      Return[i+1] = abs(diff(pairResult$spread)[i])/
        (pairResult$Index[i+1]-pairResult$Index[i])*252
    }
  }
}
pairResult <- cbind(pairResult, "Annulized Return"= Return)
pairResult
```

```
##              Index       Date  CHEVRON EXXON.MOBIL        spread
## 1996-08-23   430 1996-08-23 3.399112    3.026504  0.1198672067
## 1997-04-18   600 1997-04-18 3.475454    3.286534 -0.0052484131
## 1997-09-22   711 1997-09-22 3.751340    3.468663  0.1295340359
## 1998-04-24   865 1998-04-24 3.718892    3.604138 -0.0078731015
## 1999-04-12  1116 1999-04-12 3.865979    3.630159  0.1190546862
## 1999-12-03  1285 1999-12-03 3.817024    3.721165 -0.0004072019
## 2000-02-04  1330 2000-02-04 3.670746    3.699758 -0.1300999126
## 2000-03-30  1369 2000-03-30 3.830676    3.671543  0.0516898531
## 2000-08-01  1457 2000-08-01 3.674717    3.688099 -0.1170964774
## 2001-03-12  1616 2001-03-12 3.829836    3.734808  0.0018348734
## 2002-07-26  1975 2002-07-26 3.566853    3.557916 -0.1241016991
## 2002-12-20  2080 2002-12-20 3.506308    3.575151 -0.1979992635
## 2003-06-05  2199 2003-06-05 3.614560    3.615233 -0.1208007944
## 2004-09-13  2531 2004-09-13 3.930648    3.860730  0.0050898332
## 2005-04-20  2688 2005-04-20 3.952205    4.045504 -0.1165074863
## 2005-08-03  2763 2005-08-03 4.100161    4.077537  0.0066311578
## 2007-07-12  3269 2007-07-12 4.535820    4.495579  0.1184142323
## 2008-06-02  3501 2008-06-02 4.598850    4.475175  0.1972516385
## 2008-07-16  3533 2008-07-16 4.458872    4.392101  0.1216351649
## 2008-10-08  3593 2008-10-08 4.291828    4.343805 -0.0079917560
## 2009-12-14  3901 2009-12-14 4.347176    4.244057  0.1246361584
##                           order Annulized Return
## 1996-08-23         Short Spread               NA
## 1997-04-18 Close Short Position       0.18546551
## 1997-09-22         Short Spread               NA
## 1998-04-24 Close Short Position       0.22484804
## 1999-04-12         Short Spread               NA
## 1999-12-03 Close Short Position       0.17813252
## 2000-02-04          Long Spread               NA
## 2000-03-30  Close Long Position       1.17464156
## 2000-08-01          Long Spread               NA
## 2001-03-12  Close Long Position       0.18849497
## 2002-07-26          Long Spread               NA
## 2002-12-20      Stop Loss Order      -0.17735415
## 2003-06-05          Long Spread               NA
## 2004-09-13  Close Long Position       0.09555554
## 2005-04-20          Long Spread               NA
## 2005-08-03  Close Long Position       0.41374584
## 2007-07-12         Short Spread               NA
```

```
## 2008-06-02      Stop Loss Order      -0.08563373
## 2008-07-16         Short Spread               NA
## 2008-10-08 Close Short Position       0.54443307
## 2009-12-14         Short Spread               NA
```

Visualize the output

```
#set subsets for plotting
shortspread <- subset(pairResult,pairResult$order=="Short Spread")
longspread <- subset(pairResult,pairResult$order=="Long Spread")
Close <- subset(pairResult,pairResult$order!="Short Spread")
Close <- subset(Close,Close$order!="Long Spread")

plot(x = data$Date, y = data$spread, type="l", col = "orange", xlab = "Date", ylab="Spreads")
ablineseries <- function(height){
  abline(h=height,lty=2,col="black")
}
sapply(c(1.5*sigma,-1.5*sigma,2.5*sigma,-2.5*sigma,0),ablineseries)
points(shortspread$Date,shortspread$spread,pch =25, col= "blue")
points(longspread$Date,longspread$spread,pch =24, col= "green")
points(Close$Date,Close$spread,pch =23, col= "black")
legend("bottomleft", legend=c("Close position", "Long spread", "Short spread"), col=c("black", "green",
```



Measure Performance

1. Halflife of mean reverting process

the smaller the half-life, the more profitable or better performance of the pair trading strategy. In our pair stocks, assuming the spread follows AR(1) process, the corresponding half life is

$$ln(2)/ln(\rho) = ln(2)/ln(0.9930) = 98.37$$

2. Portfolio annualized return

This performance measure is defined as the difference of two spreads of different date multiply 252 and divided by number of days between these two dates The average annualized return between 1995 to 2009 is 0.2742

Part2: Dynamic analysis

Define trading signal

In this part, our group chose two different volatility estimation. One is historic volatility using 5yrs data, the other is using Garch(1,1) model to estimate volatility. Both Trigger are defined as $\pm 1.5 * volatility$.

Remark: in our example, we didn't assume a trade can be opened if the previous trade is close as described in assignment. Also, we set 60 trading days rather than 30 days because in our pair case, the half-life of mean reverting is about 90 days.

historical volatility

```
U = 0
L = 0
annRetHis = 0
dataset <- subset(da, select =c("Date","CHEVRON","EXXON.MOBIL"))
cointegrationRatio = 0
intercept = 0
x = 1
newSpread = 0
for (x in (1:2653)){
  trnSet <- dataset[x:(252*5+x),]
  ##test for integration(this section waited to be improved)
  # af_asst1 <- ur.df(asst1,type = "drift",lags = 0)
  # adf_asst1 <- ur.df(trnSet$CHEVRON,type = "drift")
  # adf_asst2 <- ur.df(trnSet$EXXON.MOBIL,type = "drift")
  ##long-run relationship
  linearModule <- lm(trnSet$CHEVRON~trnSet$EXXON.MOBIL)
  cointegrationRatio[x] <- linearModule$coefficients[2]
  intercept[x] <- linearModule$coefficients[1]
  spread <- linearModule$residuals

  outsampleSet <- dataset[(252*5+x+1):(252*5+x+200),]
  futureSpread <- outsampleSet$CHEVRON - cointegrationRatio[x] *
    outsampleSet$EXXON.MOBIL- intercept[x]
  newSpread[x] <- dataset$CHEVRON[252*5+x+1] - cointegrationRatio[x] *
    dataset$EXXON.MOBIL[252*5+x+1]- intercept[x]

  sigma <- sd(spread)
  U[x] <- 1.5* sigma
  L[x] <-  -1.5 * sigma

  #calculate annualized return
  tradingDays= 60
```

```
  if (futureSpread[1] > U[x]){annRetHis[x]=(futureSpread[1]-futureSpread[tradingDays+1])*252/tradingDays
  }else if (futureSpread[1] < L[x]){annRetHis[x]=(futureSpread[tradingDays+1]-futureSpread[1])*252/tradi
  }else {annRetHis[x]= 0}
}
```

using the Garch to estimate dynamic volatility

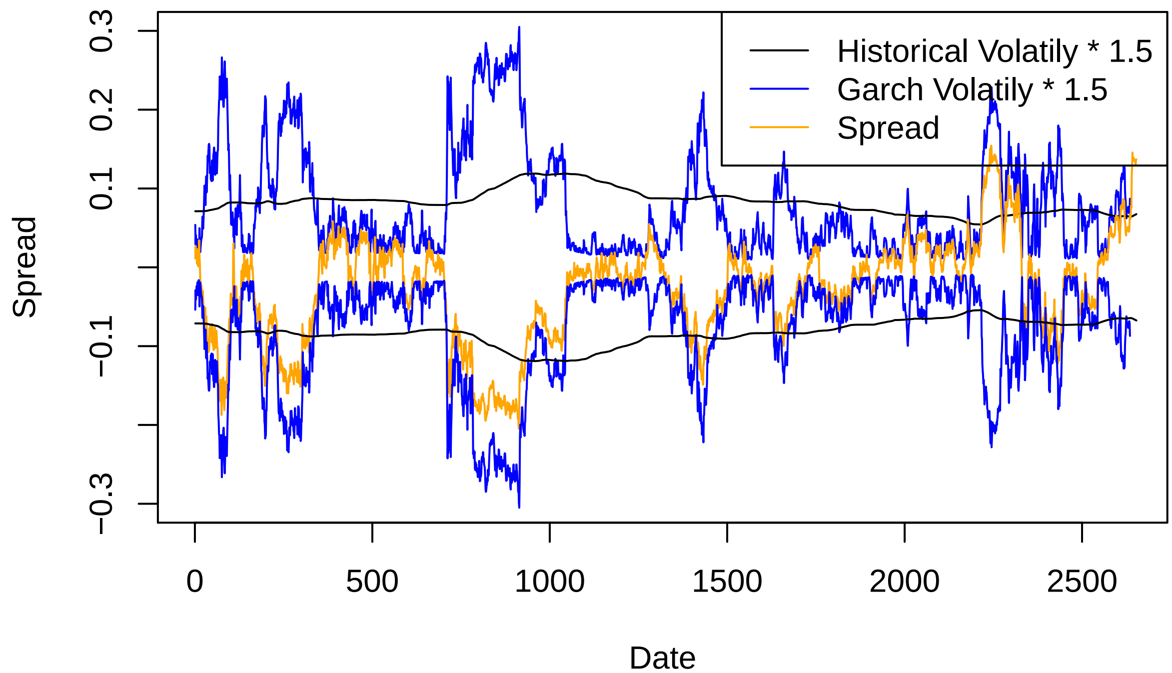
```
library(fGarch)
dataset <- subset(da, select =c("Date","CHEVRON","EXXON.MOBIL"))
UGarch=0
LGarch=0
annRetGarch=0
x=1 #daily rebalance
for (x in (1:2635)){
  trnSet <- dataset[x:(252*5+x),]
  ##test for integration
  #af_asst1 <- ur.df(asst1,type = "drift",lags = 0)
  # adf_asst1 <- ur.df(trnSet$CHEVRON,type = "drift")
  # adf_asst2 <- ur.df(trnSet$EXXON.MOBIL,type = "drift")
  ##long-run relationship
  linearModule <- lm(trnSet$CHEVRON~trnSet$EXXON.MOBIL)
  cointegrationRatio[x] <- linearModule$coefficients[2]
  intercept[x] <- linearModule$coefficients[1]
  spread <- linearModule$residuals

  outsampleSet <- dataset[(252*5+x+1):(252*5+x+200),]
  futureSpread <- outsampleSet$CHEVRON - cointegrationRatio[x] *
    outsampleSet$EXXON.MOBIL- intercept[x]
  #excuting the following garch estimate model requires few minutes, please be patient
  modelGarch <- garchFit(~garch(1,1),data=spread)
  forecastedVolatility <- predict(modelGarch,1)$standardDeviation
  UGarch[x] <- 1.5* forecastedVolatility
  LGarch[x] <-  -1.5 * forecastedVolatility

  #calculate annualized return
  tradingDays= 60
  if (futureSpread[1] > UGarch[x]){annRetGarch[x]=(futureSpread[1]-futureSpread[tradingDays+1])*252/trad
  }else if (futureSpread[1] <LGarch[x]){annRetGarch[x]=(futureSpread[tradingDays+1]-futureSpread[1])*252
  }else {annRetGarch[x]= 0}
}
```

Comparasion between two different volatility estimate method

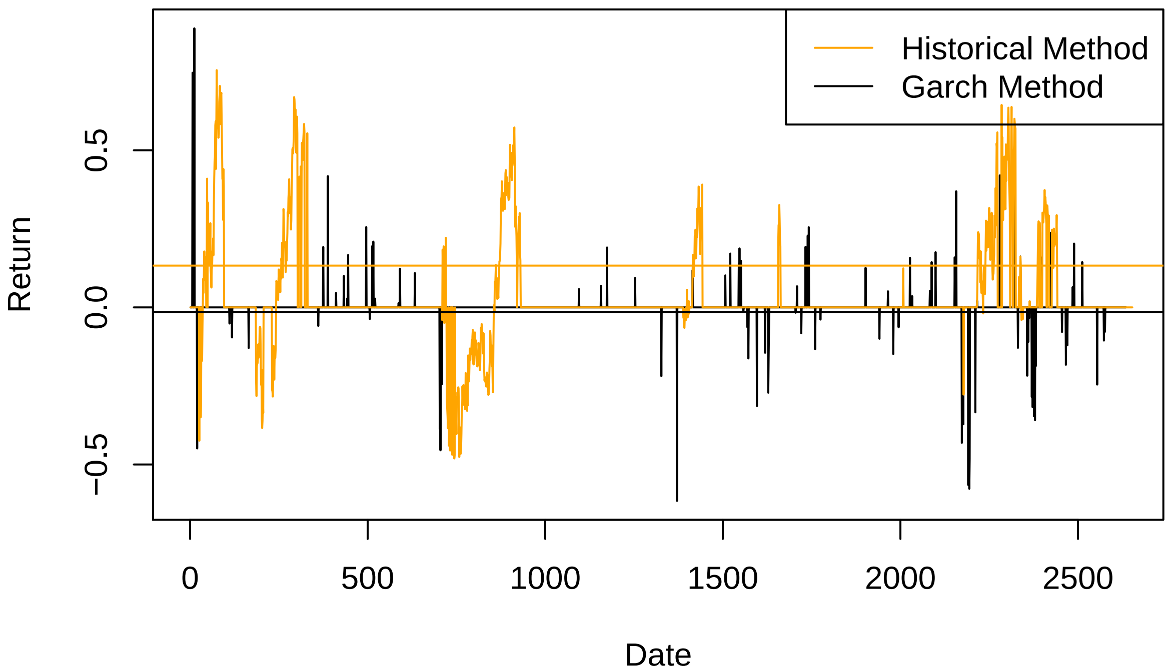
visulazation

```
  plot(UGarch, type="l",ylim=c(-0.3,0.3),col ="blue", ylab= "Spread",xlab = "Date")
  points(U,type ="l")
  points(newSpread,type ="l", col ="orange")
  points(-UGarch,type="l",ylim=c(-0.3,0.3),col ="blue")
  points(-U,type ="l")
  legend("topright",
         legend = c("Historical Volatily * 1.5","Garch Volatily * 1.5","Spread"),
         col = c("black","blue","orange"),
```

```
        lty =1)
```




```
annRetHis[is.na(annRetHis) ] <- 0
annRetGarch[is.na(annRetGarch)] <- 0
plot(annRetGarch,type="l", ylab="Return",xlab = "Date")
points(annRetHis,type="l",col = "orange")
annRetHisMean <-   mean(annRetHis[annRetHis != 0])
annRetGarchMean <-  mean(annRetGarch[annRetGarch != 0])
abline(h=annRetHisMean,col="orange")
abline(h=annRetGarchMean)
legend("topright",
       legend = c("Historical Method","Garch Method"),
       col = c("orange","black"),
       lty =1
)
```



Result

Historical Method Return : 0.1096838
Garch Method Return : −0.01476873
Static Return (form Part 1) : 0.274

Conclusion

Out sample result is not as good as in sample result, or even not desirable (negative return) in Garch model case. Reasons for the difference are as follows:

1. In static case, we always bet on the spread will finally move back to zero and we wait and close trade until that time; that means we never lose. But in dynamic out sample test, we only trade for 60 days and close the position no matter the spread is move back or move away although it is more possible to move back to mean.
2. Defects in code: as shown in Figure 6 and 7, the programming doesn't perform well in selecting a good point of trading start, which leads to a lot of negative return. If these points of negative return could be excluded, the total profile of return would be much better.
3. The choice of pair stock is not that good enough. If we chose a pair of lower half life (around 30 rather than 90 in our case), the result would be better. The half life of co-integrated pair choice is listed in appendix file.
4. We didn't know well in using ECM model. We only use the long term relationship in ECM form. But as for short run dynamics term, we have no idea to use it, for example, how to use short run dynamics in selecting triggers.