## Quantum and Oscillator Strategy and other Strategies

## Summary and Implement

## 2018-12-08

# Moving Average Convergence/Divergence

Here I used yahoo\_Finance as my data source. When a user want to use this program, he can just change the date. Here, I use XOM for example. Load data:

library("quantmod")   
#reset the date and ticker  
start <-as.Date("2017-12-11")  
end <-as.Date("2018-12-07")  
ticker <- "XOM"  
#  
getSymbols(ticker,src = "yahoo", from = start,to = end)

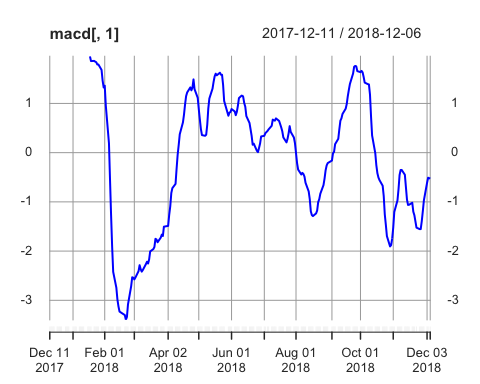
## [1] "XOM"

head(XOM)

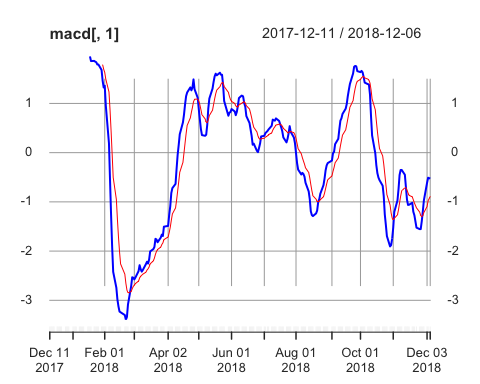
## XOM.Open XOM.High XOM.Low XOM.Close XOM.Volume XOM.Adjusted  
## 2017-12-11 83.04 83.25 82.74 83.03 8760700 78.85057  
## 2017-12-12 82.89 83.38 82.70 82.76 11307000 78.59415  
## 2017-12-13 82.61 83.30 82.48 83.12 10172700 78.93604  
## 2017-12-14 83.03 83.33 82.89 82.90 9492400 78.72712  
## 2017-12-15 83.16 83.28 82.87 83.03 26582300 78.85057  
## 2017-12-18 83.18 83.67 82.91 82.94 9363500 78.76509

Reset the parameter of MACD. In my example, the fast EMA is 12, slow EMA is 26, signal line is 9.

#reset the parameter  
macd <- MACD(XOM[, "XOM.Close"], 12, 26, 9, maType = "EMA")   
# draw the graph  
plot(macd[, 1], type = "l", col = "blue")



lines(macd[, 2], type = "l", col = "red")



#legend(201, 1.5, c("MACD", "Singal\_line"), lty = c(1,1), col = c("blue", "red"), bty = "n")

Begin to build the strategy. Here, I use some trick to spot the cross point, where the trader should implement his trading.

n <- sum(is.na(macd[,2]))  
Cp <- c()  
Cp <- rep(0, n)  
for(i in (n+1) : nrow(macd)){  
 if(macd[i, 1] > macd[i, 2]){  
 Cp[i] <- 1}   
 else{  
 Cp[i] <- -1}  
}  
# xomsignal is the trasaction signal  
xomsignal <- rep(0,nrow(macd))  
for(i in (n+1) : nrow(macd)){  
 if((Cp[i] - Cp[i - 1]) == 2) {  
 xomsignal[i] <- 'buy'}  
 else if((Cp[i] - Cp[i - 1]) == -2){  
 xomsignal[i] <- 'sell'}  
}  
XOM2<- cbind(as.data.frame(XOM), xomsignal)  
signals <- subset(XOM2,xomsignal!=0)  
signals

## XOM.Open XOM.High XOM.Low XOM.Close XOM.Volume XOM.Adjusted  
## 2018-02-26 77.74 78.93 77.56 78.84 16940400 75.63708  
## 2018-05-01 77.26 77.26 75.98 76.95 16231000 73.82388  
## 2018-05-10 80.64 81.79 80.50 81.72 17710300 78.40009  
## 2018-05-24 81.35 81.36 79.95 80.27 13360500 77.78955  
## 2018-06-07 82.73 83.29 82.56 82.88 13502500 80.31891  
## 2018-06-13 82.44 82.60 81.47 81.51 12049300 78.99125  
## 2018-06-29 82.45 83.54 82.29 82.73 17323200 80.17355  
## 2018-07-18 81.80 82.32 81.49 82.22 8711900 79.67930  
## 2018-07-26 83.83 84.40 83.37 84.24 13210100 81.63688  
## 2018-07-27 80.97 82.38 80.81 81.92 18220800 79.38857  
## 2018-08-23 79.56 79.63 78.90 79.08 7203900 77.42679  
## 2018-10-04 85.50 86.08 85.25 85.58 10204600 83.79091  
## 2018-11-02 81.76 82.45 80.23 81.95 19350400 80.23679  
## 2018-11-14 78.86 79.09 76.75 77.39 18552100 76.54024  
## 2018-11-29 78.22 79.53 78.19 79.06 10255200 78.19190  
## xomsignal  
## 2018-02-26 buy  
## 2018-05-01 sell  
## 2018-05-10 buy  
## 2018-05-24 sell  
## 2018-06-07 buy  
## 2018-06-13 sell  
## 2018-06-29 buy  
## 2018-07-18 sell  
## 2018-07-26 buy  
## 2018-07-27 sell  
## 2018-08-23 buy  
## 2018-10-04 sell  
## 2018-11-02 buy  
## 2018-11-14 sell  
## 2018-11-29 buy

Set a the initial capital and transaction fee. For example: initial capital = 10000 transaction fee = 5 (including sell and buy)

# set the initial capital and transaction fee  
cap<-10000  
fee <- 5  
# calculate the odd or even  
transaction.times <- nrow(signals) %/% 2 \* 2  
if(nrow(signals)%%2 == 0){  
 parity <- "even"  
}else if(nrow(signals)%%2 == 1){  
 parity <- "odd"  
}  
# calculate the number of shares   
shares<-rep(0,transaction.times)  
for(i in 1:(transaction.times/2)) {  
 shares[2\*i-1]=cap/(signals$XOM.Close[2\*i-1])  
 }  
shares<-floor(shares)  
# calculate the profits   
if (parity == "odd"){  
 PL<-cbind(signals[-nrow(signals),],shares)  
}else{  
 PL<-cbind(signals,shares)  
}  
  
profit<-rep(0,transaction.times)  
for(i in 1:(transaction.times/2)){  
 if(PL$xomsignal[1] == "buy"){  
 profit[2\*i] <- PL$shares[2\*i-1]\*(PL$XOM.Close[2\*i]-PL$XOM.Close[2\*i-1])-fee }  
 if(PL$xomsignal[1] == "sell"){  
 profit[2\*i] <- PL$shares[2\*i-1]\*(PL$XOM.Close[2\*i]-PL$XOM.Close[2\*i-1])-fee }  
 }  
final<-cbind(PL,profit)  
final[,5:9]

## XOM.Volume XOM.Adjusted xomsignal shares profit  
## 2018-02-26 16940400 75.63708 buy 126 0.00000  
## 2018-05-01 16231000 73.82388 sell 0 -243.13987  
## 2018-05-10 17710300 78.40009 buy 122 0.00000  
## 2018-05-24 13360500 77.78955 sell 0 -181.90049  
## 2018-06-07 13502500 80.31891 buy 120 0.00000  
## 2018-06-13 12049300 78.99125 sell 0 -169.39940  
## 2018-06-29 17323200 80.17355 buy 120 0.00000  
## 2018-07-18 8711900 79.67930 sell 0 -66.20024  
## 2018-07-26 13210100 81.63688 buy 118 0.00000  
## 2018-07-27 18220800 79.38857 sell 0 -278.76000  
## 2018-08-23 7203900 77.42679 buy 126 0.00000  
## 2018-10-04 10204600 83.79091 sell 0 814.00000  
## 2018-11-02 19350400 80.23679 buy 122 0.00000  
## 2018-11-14 18552100 76.54024 sell 0 -561.31976

Use the data to calculate return rate, sharpe ratio and so on. For example: risk-free rate is 0.511.

# set risk-free rate  
rf <- 0.511  
#  
mean(final$profit)

## [1] -49.05141

var(final$profit)

## [1] 88229.37

sd(final$profit)

## [1] 297.0343

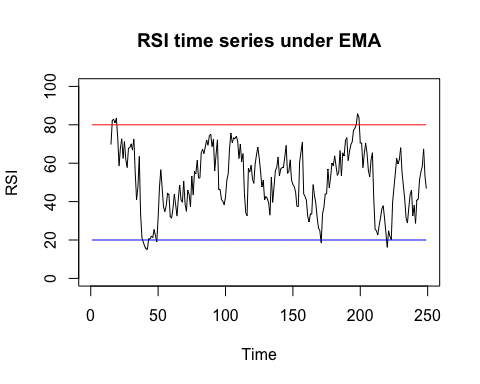
rf<-5.11/100  
ri<-rep(0,transaction.times)   
for(i in 1:(transaction.times/2)){  
 ri[2\*i]<-log(final$XOM.Close[2\*i]/final$XOM.Close[2\*i-1]) }  
ri<-ri[ri!=0]  
sharpe<-(mean(ri)-rf)/sd(ri-rf)   
sharpe

## [1] -1.444312

# RSI- Relative Strength Index

We still use the data of XOM. 1.set the time period, N =14 2.set overbought and oversold market parameter, up = 80, down=20

# set parameter  
N <- 14  
up <- 80  
down <- 20  
# EMA  
rsi<-RSI(XOM$XOM.Close,N,maType = EMA)  
ts.plot(rsi,ylim=c(0,100),main="RSI time series under EMA",ylab="RSI")   
overbought<-c(rep(up,length(rsi)))  
oversold<-c(rep(down,length(rsi)))  
lines(overbought,col="red")  
lines(oversold,col="blue")



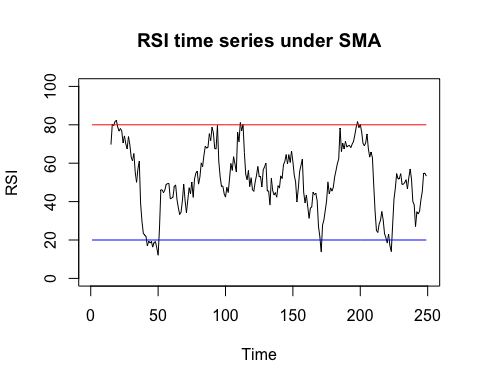
Simulate the trading process. The trick to find the point is similar to MACD.

state1<-c(rep(0,length(rsi)))  
n <- sum(is.na(rsi))  
for(i in (n+1):length(rsi)){  
 if(rsi[i] < 20){state1[i]<-1}  
 if(rsi[i] > 80){state1[i]<--1}  
}   
trade1<-c(rep(0,length(rsi)))   
for(i in 15:length(rsi)){  
 if(((state1[i]-state1[i-1])==1)&(state1[i-1]==0)) {  
 trade1[i]<-'buy'}  
 if(((state1[i]-state1[i-1])==-1)&(state1[i-1]==0)){  
 trade1[i]<-'sell' }  
 }  
trade\_set1<-cbind(as.data.frame(XOM),trade1)   
trade\_signal1<-subset(trade\_set1,trade1!=0)  
trade\_signal1

## XOM.Open XOM.High XOM.Low XOM.Close XOM.Volume XOM.Adjusted  
## 2018-01-03 85.16 86.97 84.82 86.70 13957700 82.33583  
## 2018-02-06 78.51 80.35 76.90 78.35 36262800 74.40614  
## 2018-02-21 75.82 76.44 74.87 74.89 11715000 71.84756  
## 2018-08-15 77.78 77.90 76.51 76.94 16307800 75.33153  
## 2018-09-21 85.01 85.43 84.52 85.17 26639400 83.38948  
## 2018-10-24 80.13 80.26 77.55 77.62 16085700 75.99732  
## trade1  
## 2018-01-03 sell  
## 2018-02-06 buy  
## 2018-02-21 buy  
## 2018-08-15 buy  
## 2018-09-21 sell  
## 2018-10-24 buy

SMA is similar.

rsi<-RSI(XOM$XOM.Close,N,maType = SMA)  
ts.plot(rsi,ylim=c(0,100),main="RSI time series under SMA",ylab="RSI")   
overbought<-c(rep(up,length(rsi)))  
oversold<-c(rep(down,length(rsi)))  
lines(overbought,col="red")  
lines(oversold,col="blue")



state1<-c(rep(0,length(rsi)))  
n <- sum(is.na(rsi))  
for(i in (n+1):length(rsi)){  
 if(rsi[i] < 20){state1[i]<-1}  
 if(rsi[i] > 80){state1[i]<--1}  
}   
trade1<-c(rep(0,length(rsi)))   
for(i in 15:length(rsi)){  
 if(((state1[i]-state1[i-1])==1)&(state1[i-1]==0)) {  
 trade1[i]<-'buy'}  
 if(((state1[i]-state1[i-1])==-1)&(state1[i-1]==0)){  
 trade1[i]<-'sell' }  
 }  
trade\_set1<-cbind(as.data.frame(XOM),trade1)   
trade\_signal1<-subset(trade\_set1,trade1!=0)  
trade\_signal1

## XOM.Open XOM.High XOM.Low XOM.Close XOM.Volume XOM.Adjusted  
## 2018-01-03 85.16 86.97 84.82 86.70 13957700 82.33583  
## 2018-01-05 86.75 86.88 85.71 86.75 11047600 82.38332  
## 2018-02-09 76.25 76.48 73.90 75.78 29491600 72.70140  
## 2018-05-21 81.73 82.35 81.52 82.28 8822400 79.73744  
## 2018-05-23 80.95 82.23 80.57 82.15 15140900 79.61146  
## 2018-08-15 77.78 77.90 76.51 76.94 16307800 75.33153  
## 2018-09-24 85.79 87.09 85.72 86.60 13549500 84.78958  
## 2018-09-26 86.02 86.50 85.69 85.78 10275500 83.98673  
## 2018-10-24 80.13 80.26 77.55 77.62 16085700 75.99732  
## 2018-10-26 77.87 78.41 76.96 77.53 18160100 75.90919  
## trade1  
## 2018-01-03 sell  
## 2018-01-05 sell  
## 2018-02-09 buy  
## 2018-05-21 sell  
## 2018-05-23 sell  
## 2018-08-15 buy  
## 2018-09-24 sell  
## 2018-09-26 sell  
## 2018-10-24 buy  
## 2018-10-26 buy

# Pair trading strategy

Here we use XOM and CVS to implement pair trading strategy. ticker1 = XOM ticker2 = CVS

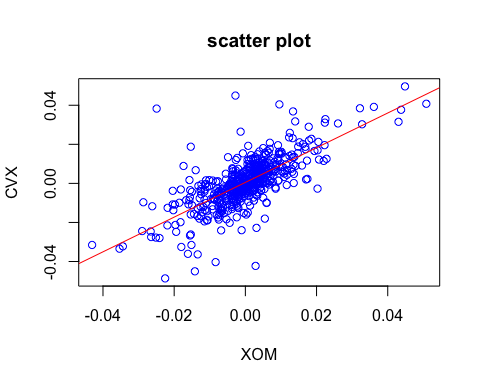
library(tseries)  
if(FALSE){  
# set the tickers and time period.  
 start <- as.Date("2015-11-30")  
 end <- as.Date("2017-11-30")  
 ticker1 = "XOM"  
ticker2 = "CVS"  
getSymbols(ticker1,src = "yahoo", from = start,to = end)  
getSymbols(ticker2,src = "yahoo", from = start,to = end)  
# get the graphs  
close1 <- as.matrix(XOM$XOM.Close)  
close2 <- as.matrix(CVS$CVS.Close)  
logr1 <- diff(log(close1))  
logr2 <- diff(log(close2))  
}  
  
setwd("/Users/yifuhe/Desktop")  
CVS <-read.csv("CVX.csv")   
XOM <-read.csv("XOM.csv")   
x <-diff(log(XOM$Close))  
y <-diff(log(CVS$Close))  
fit2<-lm(y~x)  
summary(fit2)

##   
## Call:  
## lm(formula = y ~ x)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.045283 -0.004507 0.000057 0.004919 0.059951   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.0004814 0.0003980 1.21 0.227   
## x 0.8882734 0.0374877 23.70 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.008918 on 500 degrees of freedom  
## Multiple R-squared: 0.5289, Adjusted R-squared: 0.528   
## F-statistic: 561.5 on 1 and 500 DF, p-value: < 2.2e-16

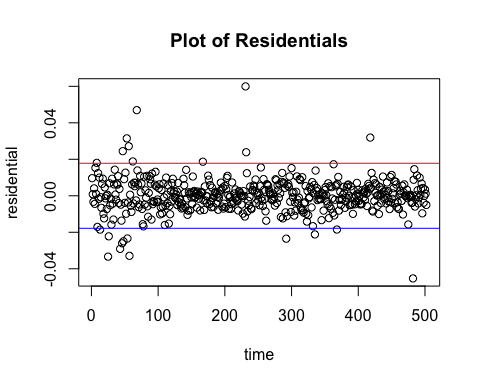
z<-c(NA)

build the strategy

# Estimate the co-integrating relation  
if(FALSE){  
plot(x,y,main = "scatter plot of the log returns")  
fit <- lm(logr2~logr1)  
summary(fit)  
intercept <- coef(fit)[1]  
slope <- coef(fit)[2]  
abline(fit,cex = 1.3,pch = 16, col = "red" )  
res<-fit$residuals  
plot(fit,which=1)  
# adf test  
adf.test(res)  
}  
  
  
#png(file = "linearregression of XOM and CVX.png")  
plot(x,y,col = "blue",main = "scatter plot",xlab="XOM",ylab="CVX")  
abline(fit2,cex = 1.3,pch = 16,col = "red")



z <- fit2$residuals  
stdz<-sd(z)  
delta <-2\*stdz  
plot(z,ylab="residential",xlab="time",main="Plot of Residentials")  
abline(h=-delta,col="blue")  
abline(h=delta,col="red")



# adf test  
adf.test(z)

##   
## Augmented Dickey-Fuller Test  
##   
## data: z  
## Dickey-Fuller = -8.5793, Lag order = 7, p-value = 0.01  
## alternative hypothesis: stationary

if P-value is less than 0.01, we can rejected the null Hypothesis. For example: use delta = 2 \* std(z\_t) accuracy = 0.001 Portfolio: long — logr2 short — logr1

if(FALSE){  
# set delta  
delta<-2\*sd(res)  
accuracy = 0.001  
##  
date<-as.Date(row.names(as.data.frame(XOM))[-1])  
error <- logr2-slope\*logr1 - intercept  
data <- data.frame(date,logr1,logr2,error)  
data$signal=c(rep("none",nrow(data)))  
data[abs((data$error+delta))<=accuracy,]$signal<-"buy"  
data[abs((data$error-delta))<=accuracy,]$signal<-"sell"  
data$order <- seq(nrow(data))  
trade\_signal <- subset(data, data$signal != "none")  
trade\_signal}  
##  
  
  
slope <- coef(fit2)[2]  
intercept <- coef(fit2)[1]  
date<-as.Date(XOM$Date[-1])  
yt\_axt<-y-slope\*x  
data<-data.frame(date,x,y,yt\_axt)  
data$signal = rep("none",nrow(data))  
data[abs((data$yt\_axt+delta-intercept))<=0.001,]$signal<-"buy"   
data[abs((data$yt\_axt-delta-intercept))<=0.001,]$signal<-"sell"  
data$order = seq(nrow(data))  
trade<-subset(data,data$signal!="none")  
trade

## date x y yt\_axt signal order  
## 8 2015-12-10 0.0007930875 0.01922055 0.01851607 sell 8  
## 9 2015-12-11 -0.0179969667 -0.03255097 -0.01656474 buy 9  
## 13 2015-12-17 -0.0151489498 -0.03152779 -0.01807138 buy 13  
## 62 2016-03-01 0.0140000675 0.03172991 0.01929403 sell 62  
## 167 2016-07-29 -0.0139550134 0.00675581 0.01915168 sell 167  
## 363 2017-05-10 -0.0048714503 0.01342300 0.01775018 sell 363  
## 368 2017-05-17 -0.0068069425 -0.02406347 -0.01801704 buy 368

Calculate P/L Suppose that the transaction size is 1 contract.

if(FALSE){  
close1 <- XOM$Close[(trade\_signal$order)+1]  
close2 <- CVS$Close[(trade\_signal$order)+1]  
signal <- trade\_signal$signal  
signal[signal == "sell"] = 1  
signal[signal == "buy"] = -1  
signal <- as.numeric(signal)  
signal1 <- signal\*slope  
signal1[length(signal1)] = -2\*slope  
signal2 <- signal\*(-1)  
signal2[length(signal2)] = 2  
pnl1 <- signal1 %\*% close1  
pnl2 <- signal2 %\*% close2  
pnl <- pnl1 +pnl2  
pnl}  
  
# need to figure out why get different outcome when use same data from different source  
close1 <- XOM$Close[(trade$order)+1]  
close2 <- CVS$Close[(trade$order)+1]  
signal <- trade$signal  
signal[signal == "sell"] = 1  
signal[signal == "buy"] = -1  
signal <- as.numeric(signal)  
signal1 <- signal\*slope  
signal1[length(signal1)] = -2\*slope  
signal2 <- signal\*(-1)  
signal2[length(signal2)] = 2  
pnl1 <- signal1 %\*% close1  
pnl2 <- signal2 %\*% close2  
pnl <- pnl1 +pnl2  
pnl

## [,1]  
## [1,] 12.22956