570 Market Microstructure and trading

Wan Li

December 8, 2015

Trading Strategy Project-Empirical and Back test Report

Introduction

In the project, three major strategies based on RSI and MACD indicators are used: buy and sell, short and buy to close, and mix of the buy and short to test the return performance. Two stocks are chosen for the sample. One (Ticker: PTR) is going downward during the five years (roughly 50% decline, from 120 to 60 USD/Share) while another (Ticker: ARM) is going upward (roughly 100% increase, from 500 to 1000 GBP/Share). The purpose of doing so is to briefly check whether the strategy will differ on different fundamental unrelated stocks.

Results (Detail performance refer to excel file)

- Performance on Bootstrap back test work better than the empirical sample; Performance on sample works around 3-7% while empirical works only a little positive return nearly zero, both get a relatively high winning percent and much better than the max drawdown rate, so all of which at least demonstrates the strategy can work in some degree.
- RSI indicator works better with respect to MACD indicator; RSI Indicator work a positive return apparently than the MACD. MACD work with almost zero or negative return.
- Short strategy alone works better than buy strategy or the mix. The average is 11-16% compared to the 3-7% of buy or mix.

- The strategy works better on stock with upward trend, more volatile, good fundamentals, around 5% higher than the downward one.

The drawback:

- It is found that for the RSI, only limited trading times can match and work with the signal, the average round trading times are around 4 in the project and it's not quite suitable for short term investment. It is also not practical for short strategy.
- It is also found that for the MACD indicator, it always yield small negative return or zero. And after a few trails of changing the conditions—limit the stop loss even to 1% and strengthen the condition for sequentially >0 and <0 to ensure the apparent signals, it does not work very well even though the wining percent is greatly improved.
- So the possible improved way may be have the MACD indicator worked with other signals together to trade, like RSI and Bollinger bands etc.,. And the most importantly, it should have volume covered which is a great information signal. And also the trading strategy developed also needs to be optimized.

Performance Overview

PTR (RSI<30, >65)	Period	Trade Times	Tot Return	Wining Percent	Max down
Mix of buy and short strategy					
Empirical	5 year	16	1.98×10^{-24}	75.00%	-74.64%
Bootstrap Avg.	5 year	4	3.36%	90.69%	-106.55%
Buy strategy					
Empirical	5 year	16	1.98×10^{-24}	75.00%	-74.64%
Bootstrap Avg.	5 year	4	3.54%	95.84%	-106.55%
Short strategy					
Empirical	5 year	16	-3.43×10^{-20}	68.75%	-74.64%
Bootstrap Avg.	5 year	4	11.95%	82.79%	-106.55%

ARM(RSI<20, >80)	Period	Trade Times	Total Return	Wining Percent	Max down
Mix of buy an short strategy					
Empirical	5 year	10	-8..42×10-^{12}	90.00%	-115.58%
Bootstrap Avg.	5 year	4	3.16%	82.91%	-192.76%
Buy strategy					
Empirical	5 year	10	-8..42×10-^{12}	90.00%	-115.58%
Bootstrap Avg.	5 year	4	7.49%	94.20%	-192.76%
Short strategy					
Empirical	5 year	10	-8..46×10-^{09}	66.67%	-115.58%
Bootstrap Avg.	5 year	4	16.71%	82.17%	-192.76%

PTR (MACD)	Period	Trade Times	Total Return	Wining Percent	Max down
Mix of buy an short strategy					
Empirical	5 year	22	1.04×10^{-36}	81.82%	-74.64%
Bootstrap Avg.	5 year	16	4.65×10^{-10}	76.77%	-106.55%
Buy strategy					
Empirical	5 year	22	1.04×10^{-36}	81.82%	-74.64%
Bootstrap Avg.	5 year	16	-1.23×10^{-12}	83.01%	-106.55%
Short strategy					
Empirical	5 year	21	-4.2×10^{-26}	57.14%	-74.64%
Bootstrap Avg.	5 year	15	-2.2×10^{-7}	55.46%	-106.55%
ARM (CDMA)	Period	Trade Times	Total Return	Wining Percent	Max down
Mix of buy an short strategy					
Empirical	5 year	23	-2.8×10^{-36}	95.65%	-115.58%
Bootstrap Avg.	5 year	17	7.94×10^{-12}	82.91%	-192.76%
Buy strategy					
Empirical	5 year	23	-2.8×10^{-36}	95.65%	-115.58%
Bootstrap Avg.	5 year	17	6.15×10^{-15}	92.18%	-192.76%
Short strategy					
Empirical	5 year	22	-2.36×10^{-27}	59.09%	-115.58%
Bootstrap Avg.	5 year	16	-2.36×10^{-8}	60.34%	-192.76%

Method

Sample:
Time period: 5 year - 2010-11-30 to 2015- 11-30 (1259 trading days)

Ticker: PTR (PetroChina Co.Ltd - NYSE), ARMA(2,2) model
ARM.L (ARM Holding plc -LSE), GARCH(1,1) model

Resample method: Bootstrap residuals (500 sample)

Strategy: per RSI / MACD indicator
1)buy-sell 2) short-buy to close 3)mixed of buy and short strategy

Implement Algorithm
Step 1: fit Model of time series with respect to stock return empirically
Step 2: Bootstrap residual from model to resample price
Step 3: Figure out TA indicators
Step 4: Execute strategy on empirical and bootstrap samples
Step 5: Calculate Performance

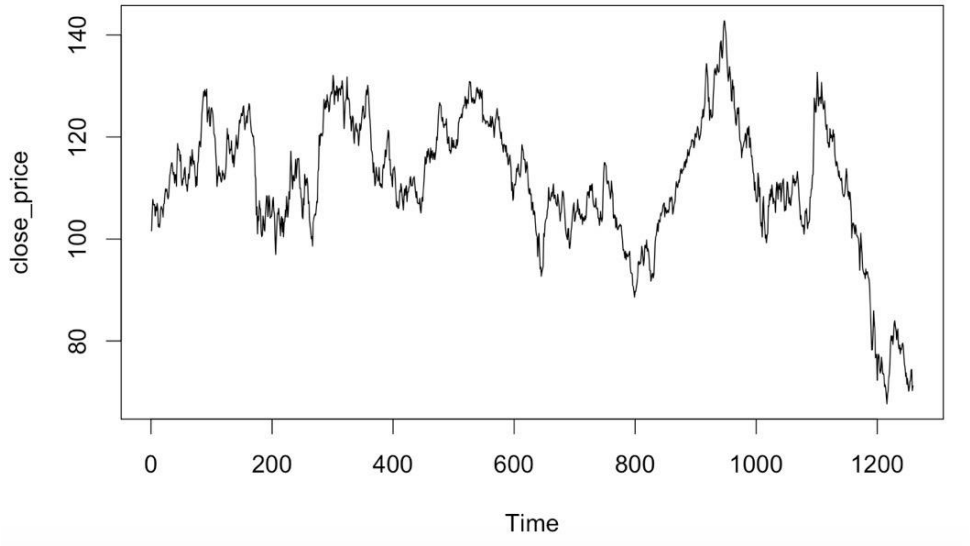
Code (Note: based on R, major code are listed below, details refer to the code files**)**

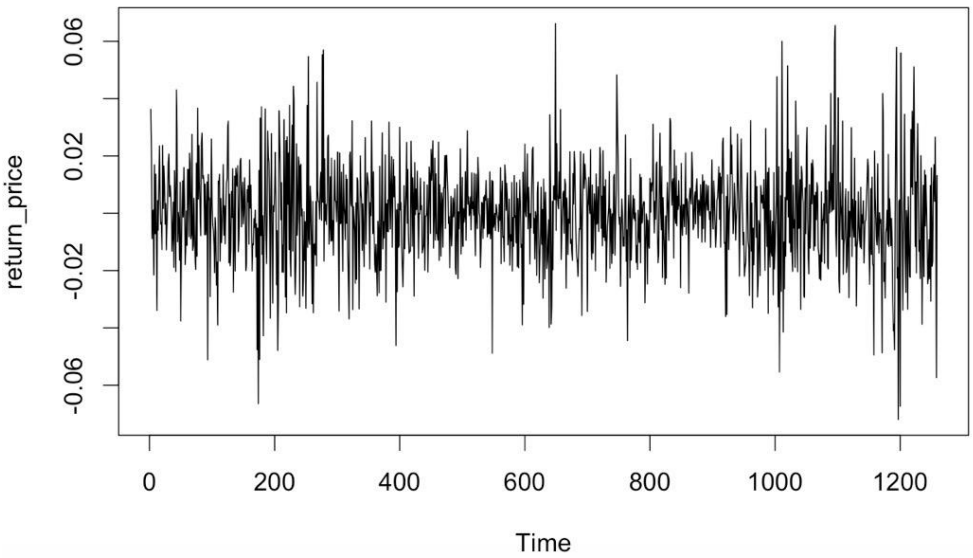
Step1: Model fit for ticker PTR , ARMA (2,2)

```
#########################   Model Built for Return  ARMA(2,2)  ################################

getSymbols("PTR",scr="yahoo",from=as.Date("2010-11-30"),to=as.Date("2015-11-30"))

close_price<-PTR[,"PTR.Adjusted",drop=FALSE]
close_price #Get a rough plot of the price
plot.ts(close_price)
acf(close_price)
pacf(close_price)

return_price<-diff(log(close_price))#return
re_price<-na.omit(return_price)
plot.ts(return_price)# remove the null one
# 1258 sample for return

y<-arima(x=re_price,order=c(2,0,2))
y
confint(y) # test for the model coefficients
error<-residuals(y)
Box.test(error,lag=20,type='Ljung') # test for the auto correlation of the residuals
```

```
# Coefficients:
#          ar1      ar2     ma1     ma2             intercept
#      -0.3051  -0.9154  0.3110  0.9523             -4e-04
#s.e.   0.0396   0.0602  0.0294  0.0505              5e-04

# so y[t]=-0.3051*y[t-1]-0.9154*y[t-2]+0.3110*e[t-1]+0.9523*e[t-2]
# drift is tested to be zero

#sigma^2 estimated as 0.0002985:  log likelihood = 3320.34,  aic = -6628.67
```

Graph of Ticker PTR Price and Return

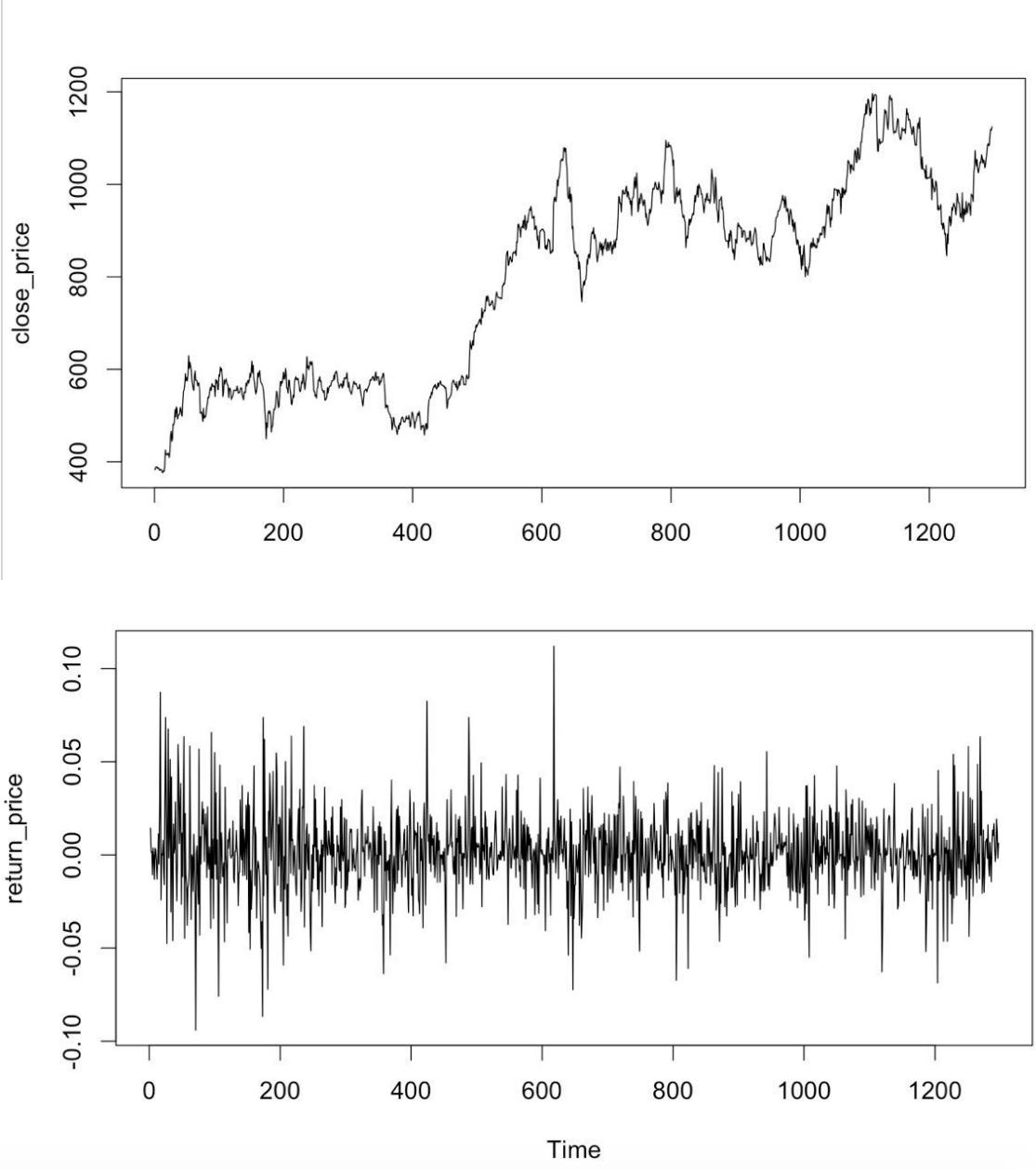




Model fit for ticker ARM.L , GARCH (1,1)

```
getSymbols("ARM.L",scr="yahoo",from=as.Date("2010-11-30"),to=as.Date("2015-11-30"))

close_price<-ARM.L[,"ARM.L.Adjusted",drop=FALSE]
close_price #Get a rough plot of the price
plot.ts(close_price)
acf(close_price)
pacf(close_price)

return_price<-diff(log(close_price))#return
re_price<-na.omit(return_price)
plot.ts(return_price)# remove the null one
# 1258 sample for return


#y<-arima(x=re_price,order=c(1,0,1))
y<-garchFit(~garch(1,1),data=re_price,trace=FALSE)
summary(y)
error<-residuals(y)
error[1:10]


#Coefficient(s):
#  mu       omega      alpha1       beta1
#1.0036e-03  1.5396e-05  3.6236e-02  9.2746e-01

#Std. Errors:
#based on Hessian

#Error Analysis:
#Estimate  Std. Error  t value Pr(>|t|)
#mu     1.004e-03   5.560e-04    1.805  0.07110 .
#omega  1.540e-05   7.780e-06    1.979  0.04783 *
#alpha1 3.624e-02   1.336e-02    2.713  0.00667 **
#beta1  9.275e-01   2.910e-02   31.877  < 2e-16 ***

#Standardised Residuals Tests:
#Statistic p-Value
#Jarque-Bera Test   R    Chi^2  418.9212  0
#Shapiro-Wilk Test  R    W      0.9705736 1.391907e-1
#jung-Box Test     R    Q(10)  5.506453  0.8548866
#Ljung-Box Test     R    Q(15)  9.138897  0.8701436
#Ljung-Box Test     R    Q(20)  10.65898  0.9545913
#Ljung-Box Test     R^2  Q(10)  7.263557  0.7003512
#Ljung-Box Test     R^2  Q(15)  10.72817  0.7716151
#Ljung-Box Test     R^2  Q(20)  14.68727  0.7940136
#LM Arch Test       R    TR^2   7.966914  0.7877106
```

Graph of Ticker ARM.L



Step2: Bootstrap for resample
ARMA(2, 2)

```
N<-length(re_price)
pricesim<-matrix(nrow=(N+1),ncol=500)  #Create simulated price in matrix named pricesim
re_pricesim<-matrix(nrow=N,ncol=500)  #Create simulated return in matrix named re_pricesim
maerror<-matrix(nrow=N,ncol=500)   #Create ma error term in matrix named maerror
se<-matrix(nrow=N,ncol=500)
er<-matrix(nrow=N,ncol=500)#Create standard error term in matrix named se
bootstrap_error<-matrix(nrow=N,ncol=500) #Create boostraping error in matrix named boostrap_error
```

```
##################################### Model For ARMA(2, 2) ###################################
ar1<-0.3051;ar2<--0.9154;ma1<-0.3110;ma2<-0.9523; sigma<-sqrt(0.0002985)
pricesim[1,]<-close_price[1]
pricesim[2,]<-close_price[2]
pricesim[3,]<-close_price[3]
re_pricesim[1,]<-re_price[1]
re_pricesim[2,]<-re_price[2]


for(j in 1:500)
{
  maerror[,j]<-rnorm(N,sd=sigma)  #create N error term for ma with mean 0 and sigma
  bootstrap_error[,j]<-sample(error,N,replace=TRUE) # boostrap the residuals from empirical mode

  for ( i in 3:N)
  {
    re_pricesim[i,j]<-ar1*re_pricesim[i-1,j]+ar2*re_pricesim[i-2,j]
    +ma1*maerror[i-1,j]+ma2*maerror[i-2,j]+bootstrap_error[i,j] #ARMA(2,2)model for return
    pricesim[i+1,j]<-pricesim[i,j]*exp(re_pricesim[i,j])  # simluate price
  }

}
```

GARCH (1,1)

```
######################################## Model For GARCH(1, 1) ########
omega<-1.5396e-05; alpha1<-3.6236e-02; beta1<-9.2746e-01;
sigmasquare<-matrix(nrow=N,ncol=500)
sigmasquare[1,]<-((re_price[1]-error[1])/rnorm(1))^2
er[1,]<-(re_price[1]-error[1])
pricesim[1,]<-close_price[1]
pricesim[2,]<-close_price[2]
re_pricesim[1,]<-re_price[1]
```

```
for(j in 1:500)
{

  se[,j]<-rnorm(N)#create standard normal error (se)with mean 0 and 1
  bootstrap_error[,j]<-sample(error,N,replace=TRUE) # boostrap the residuals from empirical model

  for ( i in 2:N)
  {

    sigmasquare[i,j]<-omega+alpha1*er[1,j]+beta1*sigmasquare[i-1,j]
    #calculate sigma square at time t based on sigma square at t-1 and error term square(sigma*se at t-1)
    er[i,j]<-sqrt(sigmasquare[i,j])* se[i,j] # calculate error term at time t based on sigma*se
    re_pricesim[i,j]<-er[i,j]+bootstrap_error[i,j] #GARCH(1,1) model, calculate return by error term add boostrap
    pricesim[i+1,j]<-pricesim[i,j]*exp(re_pricesim[i,j])  # simluate price

  }

}
```

Step 3: Figure out RSI Indicator

(here set buy if RSI<30, SELL RSI>65 for PTR and (RSI<20,RSI>80 for ARM with more volatility)

```
 ###################################   RSI Raw Indicator Created  ##########################
RSIindicator<-RSI(pricesample[,1],n=14,maType="EMA")

for (k in 2:501)
{
  temp<-RSI(pricesample[,k],n=14,maType="EMA")
  RSIindicator<-cbind(RSIindicator,temp) # Aggregrate all the MACD and signals
}

str(RSIindicator)
RSIindicator[1:100,1:5]

#################################### RSI Trade Price Signal Call ################
flag<-0
temp_L <- temp_I<-1 #initialize back to 1
trade2<-matrix(nrow=1259,ncol=501)
min0<-rep(40,501)
RSIindicator[1:100,1:10]
```

```
for (j in 1:501 ){
  for (i in 15:1259)
  {
    if(flag%%2 == 0)
    {
      if (!is.na(RSIindicator_mirror[i,j]) && RSIindicator_mirror[i,j]<30)
      {
        trade2[i,j]<-pricesample_mirror[i,j]
        temp_L <- j # save the colum index when the point of price calls
        temp_I <- i  # save the row index when the point of price calls
        flag <- (flag+1) # when buy finishes, switch to sell for round trip;
      }
    }
    else if(flag%%2 == 1)
    {
      if(!is.na(RSIindicator_mirror[i,j])&& RSIindicator_mirror[i,j]>65)
      {
        trade2[i,j] <- pricesample_mirror[i,j]*(-1)  # quote the price for short position (negative
        flag <- (flag+1) # when sell finishes, switch to buy for round trip;

        if(  pricesample_mirror[i,j]/pricesample_mirror[temp_I,temp_L] <= 0.95 )
        {  # loss limit control, when sell price/buy price<=0.9,
          # quoted price turn back to NA to give away the buy/sell signals
          trade2[temp_I,temp_L] <- NA
          trade2[i,j] <- NA
        }
      }
    }
  }
}
```

Step 4 and 5:Excecute strategy and Calculate the performance

Mix of Buy and short Strategy

```
#########################  cal_re4 begin with mix of buy-sell/ short-buy2close strategy #

cal4_re<-rep(1,501)  #initialize the calcuated total return value to 1 for iterating
cal4_rate<-rep(0.0,501) # calculate the wining percent
pos_rate <- all_rate <- 1 #initialize the positive return times and all the reutrn times
all_trade<-rep(0,501)
log_tradeprice2 <- 1
flag <- TRUE
m <- n <- 1
```

```
for (m in 1:501 )
{
  if(tradeprice2[1,m] > 0 && !is.na(tradeprice2[1,m]))
    flag <- FALSE

   for (n in seq(1,1258,2))  # for round trips
  {
    if(!is.na(tradeprice2[n,m]) && !is.na(tradeprice2[n+1,m]))
    {
      if(flag)
        log_tradeprice2 <- log(tradeprice2[n,m]/tradeprice2[n+1,m]*(-1)) # for single round return calculation,
      # negative price(short) as numerator, positive price(buy) as denominator
      else
        log_tradeprice2 <- log(tradeprice2[n+1,m]/tradeprice2[n,m]*(-1))
        cal4_re[m]= cal4_re[m] * log_tradeprice2 # calculate total return for one sample
      if(log_tradeprice2>0) # to calculate the positive return times
        {
        pos_rate=pos_rate+1
        }
        all_rate= all_rate+1 # to calculate the total trading times
    }
  }
  cal4_rate[m] <- pos_rate/all_rate #calculate the wining percent
  all_trade[m] <-all_rate #calculate the trading times
  pos_rate <- 1 # reset 1 to calculate for next strategy
  all_rate <- 1 #reset 1 to calculate for next strategy
}

cal4_re[1]
cal4_rate[1]
all_trade[1]

mean(cal4_re[2:501])
mean(cal4_rate[2:501])
mean(all_trade[2:501])
```

Buy Strategy

```
################################### cal5_re begin with buy sell strategy ##########
cal5_re<-rep(1,501)
cal5_rate<-rep(0.0,501)
all_trade<-rep(0,501)
pos_rate <- all_rate <- 1
log_tradeprice5 <- 1
flag <- TRUE
m <- n <- begin <- 1
```

```
for (m in 1:501 )
{
  for(n in 1:1258){
    if(!is.na(tradeprice2[n,m]) && tradeprice2[n,m]>0){  #Find the first signal price >0 for buy sell
      begin <- n
      flag <- tradeprice2[n,m]<0
      break
    }
  }
  n <- 1
  for (n in seq(begin,1258,2))
  {
    if(!is.na(tradeprice2[n,m]) && !is.na(tradeprice2[n+1,m]))
    {
      if(flag)
        log_tradeprice5 <- log(tradeprice2[n,m]/tradeprice2[n+1,m]*(-1)) #Calculate the round return
      else
        log_tradeprice5 <- log(tradeprice2[n+1,m]/tradeprice2[n,m]*(-1))
        cal5_re[m]= cal5_re[m] * log_tradeprice5 # calculate total return for one sample
      if(log_tradeprice5>0)
        {
        pos_rate=pos_rate+1
        }# to calculate the positive return times
        all_rate = all_rate+1 # to calculate the total trading times
    }
  }
  cal5_rate[m] <- pos_rate/all_rate #calculate the wining percent
  all_trade[m] <-all_rate #calculate the trading times
  pos_rate <- 1
  all_rate <- 1
}

cal5_re[1]
cal5_rate[1]
all_trade[1]

mean(cal5_re[2:501])
mean(cal5_rate[2:501])
mean(all_trade[2:501])
```

Short and Buy2close Strategy

```
################################## cal6_re begin with short, buy2close strategy ################

cal6_re<-rep(1,501)
cal6_rate<-rep(0.0,501)
pos_rate <- all_rate <- 1
log_tradeprice6 <- 1
flag <- TRUE
m <- n <- begin <- 1

for (m in 1:501 )
{
  for(n in 1:1258){
    if(!is.na(tradeprice2[n,m]) && tradeprice2[n,m]<0){  #Find the first signal price <0 for shor
      begin <- n
      flag <- tradeprice2[n,m]<0
      break
    }
  }
  n <- 1
  for (n in seq(begin,1258,2))
  {
    if(!is.na(tradeprice2[n,m]) && !is.na(tradeprice2[n+1,m]))
    {
      if(flag)
        log_tradeprice6 <- log(tradeprice2[n,m]/tradeprice2[n+1,m]*(-1)) #Calculate the round ret
      else
        log_tradeprice6 <- log(tradeprice2[n+1,m]/tradeprice2[n,m]*(-1))
      cal6_re[m]= cal6_re[m] * log_tradeprice6 # for return calculation
      if(log_tradeprice6>0)
        pos_rate=pos_rate+1 # to calculate the positive return times
      all_rate = all_rate+1 # to calculate the total trading times
    }
  }
  cal6_rate[m] <- pos_rate/all_rate
  pos_rate <- 1
  all_rate <- 1
}
cal6_re[1]
cal6_rate[1]
all_trade[1]

mean(cal6_re[2:501])
mean(cal6_rate[2:501])
mean(all_trade[2:501])
```

```
######Maxdown measure#################################
maxdown<-rep(0,501)
maxprice<-rep(0,501)
minprice<-rep(0,501)

for(j in 1:501){

  maxprice[j]<-max(pricesample[,j])
  minprice[j]<-min(pricesample[,j])
  maxdown[j]<-log(minprice[j]/maxprice[j])

}

maxdown[1]
mean(maxdown[2:501])
```

Step3 : Figure out MACD indicator

```
#################################   MACD Raw Indicator Created  ###################################

MACDindicator<-MACD(pricesample[,1],nFast=12, nSlow=26, nSig=9) # Initialize the MACD and signals

for (k in 2:501)
{
  temp<-MACD(pricesample[,k],nFast=12, nSlow=26, nSig=9)
  MACDindicator<-cbind(MACDindicator,temp) # Aggregrate all the MACD and signals
}

class(MACDindicator)
str(MACDindicator)
MACDindicator[ 1:100,1:4]
```

```
#################################  MACD Trade Price Singal Call ##########################
remove(trade)
trade<-matrix(nrow=1259,ncol=501) # Create matrix to contain buy and sell signal prices;
flag<-0 # for round trip trading guarantee
temp_L <- temp_I<-1 # Create index of price when quoting
#l<-i<-1
MACDindicator_mirror <- as.matrix(MACDindicator)
pricesample_mirror <- as.matrix(pricesample)
#MCDindicator_mirror[1:100,1:5]
#pricesample_mirror[1:100,1:5]
```

Since MACD always yield negative return, so change the condition to sequential <0 and >0 to figure out the strengthened signal and have the stop loss limited to 1% to check out the possible positive performance.

```
for (l in seq(1,1002,2) )
{
  for (i in 35:1257)
  {
    if(flag%%2 == 0)
    {
      if( MACDindicator_mirror[i-1,l] < MACDindicator_mirror[i-1,l+1] && MACDindicator_mirror[i,l]<MACDindicator_mirror[i,l+1]
          && MACDindicator_mirror[i+1,l]>MACDindicator_mirror[i+1,l+1] && MACDindicator_mirror[i+2,l] > MACDindicator_mirror[i+2,l+1] )
        # When MACD cross signal from below to above,
        #equally the difference changes from <0 to >0, then buy
      {
        trade[i+2,(l+1)/2]<-pricesample_mirror[i+2,(l+1)/2] # quote the price for long position ( positive value for buy price)
        flag<-flag+1 # when buy finishes, switch to sell for round trip;
        temp_L <- (l+1)/2 # save the colum index when the point of price calls
        temp_I <- i+2  # save the row index when the point of price calls

      }
    }
    if(flag%%2 == 1)
    {
      if(MACDindicator_mirror[i-1,l] > MACDindicator_mirror[i-1,l+1] && MACDindicator_mirror[i,l]>MACDindicator_mirror[i,l+1]
         && MACDindicator_mirror[i+1,l]<MACDindicator_mirror[i+1,l+1] && MACDindicator_mirror[i+2,l]< MACDindicator_mirror[i+2,l+1])
        # When MACD cross signal from below to above,
      { #equally the difference changes from >0 to <0, then sell
        trade[i+2,(l+1)/2]<-pricesample_mirror[i+2,(l+1)/2]*(-1) # quote the price for short position (negative value for sell or short
        flag<-flag+1 # when sell finishes, switch to buy for round trip;

        if(pricesample_mirror[i+2,(l+1)/2]/pricesample_mirror[temp_I,temp_L] <= 0.99){  # loss limit control, when sell price/buy price<
          # quoted price turn back to NA to give away the buy/sell signals
          trade[temp_I,temp_L] <- NA
          trade[i+2,(l+1)/2] <- NA
        }
      }
    }
```

```
#Re permutation for the tradeprice
remove(tradeprice)
tradeprice <- matrix( nrow=1259, ncol=501)    # rearrange the signal prices in matrix to order the buy and sell signal pr
i <- j <- m <- n <- 1
for (j in 1:501 )
{
  for (i in 1:1259)
  {
    if( !is.na(trade[i,j]) && m <= 1259 ) # filter out the NA at first
    {
      tradeprice[m,n] <- trade[i,j]
      m <- m+1
    }
  }
  m <- 1
  n <- n+1
}
```

Step 4 and 5:Execute strategy and Calculate the performance

Mix of Buy and short Strategy

```
##############################cal_re begin with mix of buy-sell/ short-buy2close strategy#####

cal_re<-rep(1,501)  #initialize the calcuated total return value to 1 for iterating
cal_rate<-rep(0.0,501) # calculate the wining percent
pos_rate <- all_rate <- 1 #initialize the positive return times and all the reutrn times
log_tradeprice <- 1
all_trade<-rep(0,501)
flag <- TRUE
m <- n <- 1
for (m in 1:501 )
{
  if(tradeprice[1,m]>0 && !is.na(tradeprice[1,m]))
    flag <- FALSE
  for (n in seq(1,1258,2))  # for round trips
  {

    if(!is.na(tradeprice[n,m]) && !is.na(tradeprice[n+1,m]))
    {
      if(flag)
        log_tradeprice <- log(tradeprice[n,m]/tradeprice[n+1,m]*(-1)) # for single round return
      # negative price(short) as numerator, positive price(buy) as denominator
      else
        log_tradeprice <- log(tradeprice[n+1,m]/tradeprice[n,m]*(-1))
        cal_re[m]= cal_re[m] * log_tradeprice # calculate total return
      if(log_tradeprice>0)
      {# to calculate the positive return times
        pos_rate=pos_rate+1
      }
      all_rate = all_rate+1
    }# to calculate the total trading times
  }
  cal_rate[m] <- pos_rate/all_rate #calculate the wining percent
  all_trade[m] <-all_rate #calculate the trading times
  pos_rate <- 1 # reset 1 to calculate for next strategy
  all_rate <- 1 #reset 1 to calculate for next strategy
}
cal_re[1]
cal_rate[1]
all_trade[1]

mean(cal_re[2:501])
mean(cal_rate[2:501])
mean(all_trade[2:501])
```

Buy Strategy

```
#########################################cal2_re begin with buy sell strategy####
cal2_re<-rep(1,501)
cal2_rate<-rep(0.0,501)
pos_rate <- all_rate <- 1
log_tradeprice <- 1
all_trade<-rep(0,501)
flag <- TRUE
m <- n <- begin <- 1
for (m in 1:501 )
{
  for(n in 1:1258){
    if(!is.na(tradeprice[n,m]) && tradeprice[n,m]>0){  #Find the first signal price >0 for buy sell st
      begin <- n
      flag <- tradeprice[n,m]<0
      break
    }
  }
  n <- 1
  for (n in seq(begin,500,2))
  {
    if(!is.na(tradeprice[n,m]) && !is.na(tradeprice[n+1,m]))
    {
      if(flag)
        log_tradeprice <- log(tradeprice[n,m]/tradeprice[n+1,m]*(-1)) #Calculate the round return
      else
        log_tradeprice <- log(tradeprice[n+1,m]/tradeprice[n,m]*(-1))
        cal2_re[m]= cal2_re[m] * log_tradeprice # for total return calculation
      if(log_tradeprice>0)
        {
        pos_rate=pos_rate+1
        }# to calculate the positive return times
        all_rate = all_rate+1 # to calculate the total trading times
    }
  }
  cal2_rate[m] <- pos_rate/all_rate #calculate the wining percent
  all_trade[m] <-all_rate #calculate the trading times
  pos_rate <- 1
  all_rate <- 1
}

cal2_re[1]
cal2_rate[1]
all_trade[1]

mean(cal2_re[2:501])
mean(cal2_rate[2:501])
mean(all_trade[2:501])
```

Short and Buy2close Strategy

```
#####################################cal_3 begin with short, buy2close strate
cal3_re<-rep(1,501)
cal3_rate<-rep(0.0,501)
pos_rate <- all_rate <- 1
log_tradeprice <- 1
all_trade<-rep(0,501)
flag <- TRUE
m <- n <- begin <- 1

for (m in 1:501 )
{
  for(n in 1:1258){
    if(!is.na(tradeprice[n,m]) && tradeprice[n,m]<0){ #find the first signal price<0 for short-
      begin <- n
      flag <- tradeprice[n,m]<0
      break
    }
  }
  n <- 1
  for (n in seq(begin,500,2))
  {
    if(!is.na(tradeprice[n,m]) && !is.na(tradeprice[n+1,m]))
    {
      if(flag)
        log_tradeprice <- log(tradeprice[n,m]/tradeprice[n+1,m]*(-1)) # for single round return
      else
        log_tradeprice <- log(tradeprice[n+1,m]/tradeprice[n,m]*(-1))
        cal3_re[m]= cal3_re[m] * log_tradeprice # for return calculation
      if(log_tradeprice>0)
        {
        pos_rate=pos_rate+1
        }# to calculate the positive return times
      all_rate = all_rate+1 # to calculate the total trading times
    }
  }
  cal3_rate[m] <- pos_rate/all_rate #calculate the wining percent
  all_trade[m] <-all_rate #calculate the trading times
  pos_rate <- 1
  all_rate <- 1
}
cal3_re[1]
cal3_rate[1]
all_trade[1]

mean(cal3_re[2:501])
mean(cal3_rate[2:501])
mean(all_trade[2:501])
```