```
#########################
# R code for homework 2
#Yin Teng A0068164B, DSAP, NUS
#########################
#Question 1
#################
#################
#1(a)
#################
data_pf=read.csv("FF_Data_ForGRSTest.csv")
PF6=data_pf[,2:7]
Mkt_rf=data_pf$MktMinusRF
RF=data_pf$RF

myfun1<-function(x)
{
 excess.r=x-RF[1:length(x)]
 res.lm=summary(lm(excess.r~Mkt_rf[1:length(x)]))$coefficients
 return(res.lm)
}
res=apply(PF6,2,myfun1)

myfun2<-function(x)
{
 excess.r=x-RF[1:length(x)]
 res.lm=lm(excess.r~Mkt_rf[1:length(x)])
 resid=residuals(res.lm)
 fit_y=fitted(res.lm)
 res=summary(lm(abs(resid)~fit_y))
}
apply(PF6,2,myfun2)

res=vector("list",6)
for(i in 1:6)
{
 excess.r=PF6[,i]-RF
 res[[i]]=lm(excess.r~RF)
}
for( i in 1:6)
{
par(mfrow=c(2,2))
plot(res[[i]], which=c(1:3,5))
}
#################
##1.(b)
#################
```

```
PF6_1=PF6[1:518,]
PF6_2=PF6[519:1036,]
res=apply(PF6_1,2, myfun1)
b=c()
for ( i in 1:6)
{
 a=cbind(res[c(1,2),i],res[c(3,4),i],res[c(5,6),i],res[c(7,8),i])
 b=rbind(b,a)
}
apply(PF6_1,2,myfun2)
res=vector("list",6)
for(i in 1:6)
{
 excess.r=PF6_1[,i]-RF[1:518]
 res[[i]]=lm(excess.r~RF[1:518])
}
for( i in 1:6)
{
par(mfrow=c(2,2))
plot(res[[i]], which=c(1:3,5))
}


myfun3<-function(x)
{
 excess.r=x-RF[519:1036]
 res.lm=summary(lm(excess.r~Mkt_rf[519:1036]))$coefficients
 return(res.lm)
}
res=apply(PF6_2,2,myfun3)

myfun4<-function(x)
{
 excess.r=x-RF[519:1036]
 res.lm=lm(excess.r~Mkt_rf[519:1036])
 resid=residuals(res.lm)
 fit_y=fitted(res.lm)
 res=summary(lm(abs(resid)~fit_y))
}
apply(PF6_2,2,myfun4)

res=vector("list",6)
for(i in 1:6)
{
 excess.r=PF6_2[,i]-RF[519:1036]
 res[[i]]=lm(excess.r~RF[519:1036])
}
for( i in 1:6)
{
par(mfrow=c(2,2))
plot(res[[i]], which=c(1:3,5))
}
#################
#1.c
#################
SMB=data_pf$SMB
HML=data_pf$HML
myfun5<-function(x)
{
excess.r=x-RF[1:length(x)]
 res.lm=summary(lm(excess.r~Mkt_rf[1:length(x)]+SMB[1:length(x)]+HML[1:length(x)
 ]))$coefficients
 return(res.lm)
}
res=apply(PF6,2,myfun5)
b=c()
for ( i in 1:6)
{
 a=cbind(res[c(1:4),i],res[c(5:8),i],res[c(9:12),i],res[c(13:16),i])
 b=rbind(b,a)
}
myfun6<-function(x)
{
 excess.r=x-RF[1:length(x)]
 res.lm=lm(excess.r~Mkt_rf[1:length(x)]+SMB[1:length(x)]+HML[1:length(x)])
 resid=residuals(res.lm)
 fit_y=fitted(res.lm)
 res=summary(lm(abs(resid)~fit_y))
}
apply(PF6,2,myfun6)

res=vector("list",6)
for(i in 1:6)
{
 excess.r=PF6[,i]-RF
 res[[i]]=lm(excess.r~Mkt_rf+SMB+HML)
}
for( i in 1:6)
{
par(mfrow=c(2,2))
plot(res[[i]], which=c(1:3,5))
}
```

```
apply(PF6_1,2,myfun5)
apply(PF6_1,2,myfun6)

res=vector("list",6)
for(i in 1:6)
{
 excess.r=PF6_1[,i]-RF[1:518]
 res[[i]]=lm(excess.r~Mkt_rf[1:518]+SMB[1:518]+HML[1:518])
}
for( i in 1:6)
{
par(mfrow=c(2,2))
plot(res[[i]], which=c(1:3,5))
}

myfun7<-function(x)
{
excess.r=x-RF[519:1036]
 res.lm=summary(lm(excess.r~Mkt_rf[519:1036]+SMB[519:1036]+HML[519:1036]))$c
 oefficients
 return(res.lm)
}
apply(PF6_2,2,myfun7)

myfun8<-function(x)
{
 excess.r=x-RF[519:1036]
 res.lm=lm(excess.r~Mkt_rf[519:1036]+SMB[519:1036]+HML[519:1036])
 resid=residuals(res.lm)
 fit_y=fitted(res.lm)
 res=summary(lm(abs(resid)~fit_y))
}
apply(PF6_2,2,myfun8)

res=vector("list",6)
for(i in 1:6)
{
 excess.r=PF6_2[,i]-RF[519:1036]
 res[[i]]=lm(excess.r~Mkt_rf[519:1036]+SMB[519:1036]+HML[519:1036])
}
for( i in 1:6)
{
par(mfrow=c(2,2))
plot(res[[i]], which=c(1:3,5))
}
```

```
##################
####1.(d)Hotelling's T
##################
pf.manova=manova(as.matrix(PF6)~Mkt_rf+SMB+HML)
summary(pf.manova,test="Hotelling",intercept=T)


pf.manova1=manova(as.matrix(PF6_1)~Mkt_rf[1:518]+SMB[1:518]+HML[1:518])
summary(pf.manova1,test="Hotelling",intercept=T)

pf.manova2=manova(as.matrix(PF6_2)~Mkt_rf[519:1036]+SMB[519:1036]+HML[519:
 1036])
summary(pf.manova2,test="Hotelling",intercept=T)
##################
#####1.e
##################
library(urca)
summary(ca.jo(as.matrix(PF6),type="eigen",K=2))
##################
##Question 2(a)
##################
##################
###MA without transaction cost
##################
#########
#USD
#########
ma.sharpe.ratio_mat=PNL_total_mat=c()
for(L in seq(20,300,10) )
{
Rates=read.csv("Rates.csv")
INR_by_USD=1/Rates$USD_by_INR
n=dim(Rates)[1]
computeMA<-function(x,L)
{
 weights <- rep(1/L,L)
 ma.series=as.vector(filter(x, weights,                        method="convolution",side=1))
 D=cbind(as.vector(x),ma.series)
 return(D)
}
maresult=computeMA(INR_by_USD,L)

buysignal<-INR_by_USD<maresult[,2]
sellsignal<-INR_by_USD>maresult[,2]
tradeindicator <- rep(0,dim(Rates)[1])
tradeindicator[buysignal]=1
```

```r
tradeindicator[sellsignal]=-1

signal2value<-function(x,signal)
{
 result=array(0,c(n,5))
 longposition=which(signal==1)
 shortposition=which(signal==-1)
 if(length(longposition)==0) return(NA)
 lastTrade = longposition[1]
 i=longposition[1]+1
 position = 1 #initial long position
 value = 1 #initial total value is 1
 #find te first long position
    while(i<=length(x)){
      if(signal[i]==position*(-1)){
            # set new last trading
            position = signal[i]#change position
            if(signal[i]==-1){#liquidate
                  value = value * (x[i]/x[lastTrade])
                  return=x[i]/x[lastTrade]-1
                  result[i,]=c(i, x[i], x[lastTrade],value,return)
                  }
                  if(signal[i]==1){#buy order
                   result[i,]=c(i, x[i], x[lastTrade],value,0)
                  }
            lastTrade = i
            }else{#no trading
                  result[i,]=c(i, x[i], x[lastTrade],value,0)}
            i = i+1
      }
      return(result)
}

INR_USD_value=signal2value(INR_by_USD,tradeindicator)
PNL_USD=INR_USD_value[dim(INR_USD_value)[1],4]
######
#GBP
######

INR_by_GBP=1/Rates$GBP_by_INR
n=dim(Rates)[1]
maresult=computeMA(INR_by_GBP,L)

buysignal<-INR_by_GBP<maresult[,2]
sellsignal<-INR_by_GBP>maresult[,2]
tradeindicator <- rep(0,dim(Rates)[1])
```

```
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1

INR_GBP_value=signal2value(INR_by_GBP,tradeindicator)
PNL_GBP=INR_GBP_value[dim(INR_GBP_value)[1],4]
#########
#####EUR
#########
INR_by_EUR=1/Rates$EUR_by_INR
n=dim(Rates)[1]
maresult=computeMA(INR_by_EUR,L)

buysignal<-INR_by_EUR<maresult[,2]
sellsignal<-INR_by_EUR>maresult[,2]
tradeindicator <- rep(0,dim(Rates)[1])
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1

INR_EUR_value=signal2value(INR_by_EUR,tradeindicator)
PNL_EUR=INR_EUR_value[dim(INR_EUR_value)[1],4]
##sharpe ratio
valueNreturn=cbind(INR_USD_value[,4:5],INR_GBP_value[,4:5],INR_EUR_value[,4:5
 ])
valueNreturn1=c()
for (i in 1:n)
{
 if(all(valueNreturn[i,c(2,4,6)]==c(0,0,0))==F)
 valueNreturn1=rbind(valueNreturn1, valueNreturn[i,])
 }
ma.weights.value=valueNreturn1[,c(1,3,5)]/apply(valueNreturn1[,c(1,3,5)],1,sum)
ma.weighted.return=apply(valueNreturn1[,c(2,4,6)]*ma.weights.value,1,sum)
ma.sharpe.ratio=(mean(ma.weighted.return)-0.03/252)/sd(ma.weighted.return)*sqrt(252)
ma.sharpe.ratio_mat=rbind(ma.sharpe.ratio_mat,ma.sharpe.ratio)
###PNL
PNL_total=((PNL_USD-1)*PNL_USD+(PNL_GBP-1)*PNL_GBP+(PNL_EUR-
 1)*PNL_EUR)/(PNL_USD+PNL_GBP+PNL_EUR)
PNL_total_mat=rbind(PNL_total_mat,PNL_total)
}
######################
####MA with transaction cost
######################
##########
#USD
##########
ma.sharpe.ratio_mat=PNL_total_mat=c()
for(L in seq(20,300,10) )
```

```
{
Rates=read.csv("Rates.csv")
INR_by_USD=1/Rates$USD_by_INR
n=dim(Rates)[1]
computeMA<-function(x,L)
{
 weights <- rep(1/L,L)
 ma.series=as.vector(filter(x, weights,                    method="convolution",side=1))
 D=cbind(as.vector(x),ma.series)
 return(D)
}
maresult=computeMA(INR_by_USD,L)
buysignal<-INR_by_USD<maresult[,2]
sellsignal<-INR_by_USD>maresult[,2]
tradeindicator <- rep(0,dim(Rates)[1])
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
signal2value<-function(x,signal)
{
 result=array(0,c(n,5))
 longposition=which(signal==1)
 shortposition=which(signal==-1)
 if(length(longposition)==0) return(NA)
 lastTrade = longposition[1]
 i=longposition[1]+1
 position = 1 #initial long position
 value = 1 #initial total value is 1
 #find te first long position
    while(i<=length(x)){
      if(signal[i]==position*(-1)){
            # set new last trading
            position = signal[i]#change position
            if(signal[i]==-1){#liquidate

                  value = value * (x[i]/x[lastTrade])*(1-0.0015)
                  return=x[i]/x[lastTrade]*(1-0.0015)-1
                  result[i,]=c(i, x[i], x[lastTrade],value,return)
                  }
                  if(signal[i]==1){#buy order
                   result[i,]=c(i, x[i], x[lastTrade],value,0)
                  }
            lastTrade = i
            }else{#no trading
                  result[i,]=c(i, x[i], x[lastTrade],value,0)}
            i = i+1
      }
    return(result)
}
INR_USD_value=signal2value(INR_by_USD,tradeindicator)
PNL_USD=INR_USD_value[dim(INR_USD_value)[1],4]
######
#GBP
######
INR_by_GBP=1/Rates$GBP_by_INR
n=dim(Rates)[1]
maresult=computeMA(INR_by_GBP,L)
buysignal<-INR_by_GBP<maresult[,2]
sellsignal<-INR_by_GBP>maresult[,2]
tradeindicator <- rep(0,dim(Rates)[1])
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_GBP_value=signal2value(INR_by_GBP,tradeindicator)
PNL_GBP=INR_GBP_value[dim(INR_GBP_value)[1],4]
##########
#####EUR
#########
INR_by_EUR=1/Rates$EUR_by_INR
n=dim(Rates)[1]
maresult=computeMA(INR_by_EUR,L)
buysignal<-INR_by_EUR<maresult[,2]
sellsignal<-INR_by_EUR>maresult[,2]
tradeindicator <- rep(0,dim(Rates)[1])
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_EUR_value=signal2value(INR_by_EUR,tradeindicator)
PNL_EUR=INR_EUR_value[dim(INR_EUR_value)[1],4]
##sharpe ratio
valueNreturn=cbind(INR_USD_value[,4:5],INR_GBP_value[,4:5],INR_EUR_value[,4:5
 ])
valueNreturn1=c()
for (i in 1:n)
{
 if(all(valueNreturn[i,c(2,4,6)]==c(0,0,0))==F)
 valueNreturn1=rbind(valueNreturn1, valueNreturn[i,])
 }
ma.weights.value=valueNreturn1[,c(1,3,5)]/apply(valueNreturn1[,c(1,3,5)],1,sum)
ma.weighted.return=apply(valueNreturn1[,c(2,4,6)]*ma.weights.value,1,sum)
ma.sharpe.ratio=(mean(ma.weighted.return)-0.03/252)/sd(ma.weighted.return)*sqrt(252)
ma.sharpe.ratio_mat=rbind(ma.sharpe.ratio_mat,ma.sharpe.ratio)
###PNL
PNL_total=((PNL_USD-1)*PNL_USD+(PNL_GBP-1)*PNL_GBP+(PNL_EUR-
 1)*PNL_EUR)/(PNL_USD+PNL_GBP+PNL_EUR)
```

```
PNL_total_mat=rbind(PNL_total_mat,PNL_total)
}
##################################
######Bollinger band trading rule
##################################
#################
##without transaction cost
#################
########
#USD
########
signal2value<-function(x,signal)
{
 result=array(0,c(n,5))
 longposition=which(signal==1)
 shortposition=which(signal==-1)
 if(length(longposition)==0) return(NA)
 lastTrade = longposition[1]
 i=longposition[1]+1
 position = 1 #initial long position
 value = 1 #initial total value is 1
 #find te first long position
   while(i<=length(x)){
    if(signal[i]==position*(-1)){
        # set new last trading
        position = signal[i]#change position
        if(signal[i]==-1){#liquidate
            value = value * (x[i]/x[lastTrade])
            return=x[i]/x[lastTrade]-1
            result[i,]=c(i, x[i], x[lastTrade],value,return)
            }
            if(signal[i]==1){#buy order
             result[i,]=c(i, x[i], x[lastTrade],value,0)
            }
        lastTrade = i
        }else{#no trading
            result[i,]=c(i, x[i], x[lastTrade],value,0)}
        i = i+1
    }
    return(result)
}
bb.sharpe.ratio_mat=bb.PNL_total_mat=c()
for(ndays in seq(20,300,10))
{
computeBollingerBands <- function(x, ndays, nsd=2) {
  ## create a (normalised, but that's just candy) weight vector
  weights <- rep(1/ndays,ndays)
  weights2<- rev(1:ndays)
  weights2<-weights2/sum(weights2)
  ## and apply it as a one-sided moving average calculations, see help(filter)
  bbmiddle <- as.vector(filter(x, weights,
 method="convolution",side=1))
  bbmiddle2<- as.vector(filter(x, weights2,
 method="convolution",side=1))
  ## use var(x) = E(x^2) - E(x)^2 to compute rolling variances
  v <- filter(x^2, weights, method="convolution", side=1) - bbmiddle^2
  ## from which we calculate rolling standard deviations the usual way
  bbsd <- as.vector(sqrt(v))

  bbupper <- bbmiddle2 + nsd*bbsd        # upper Bollinger band
  bblower <- bbmiddle2 - nsd*bbsd        # lowet Bollinger band
  ## now extend the data frame with a few new columns
  D <- cbind(x, bbmiddle, bbmiddle2,bbsd, bbupper, bblower)

  return(D)                              # return the augmented data frame
}
bbresult<- computeBollingerBands(INR_by_USD, ndays, nsd=2)
#using band to build trading signals
buysignal <- INR_by_USD< bbresult[,6]
sellsignal <- INR_by_USD> bbresult[,5]
tradeindicator <- rep(0,n)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_USD_value=signal2value(INR_by_USD,tradeindicator)
PNL_USD=INR_USD_value[dim(INR_USD_value)[1],4]
####GBP
bbresult<- computeBollingerBands(INR_by_GBP, ndays, nsd=2)
#using band to build trading signals
buysignal <- INR_by_GBP< bbresult[,6]
sellsignal <- INR_by_GBP> bbresult[,5]
tradeindicator <- rep(0,n)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_GBP_value=signal2value(INR_by_GBP,tradeindicator)
PNL_GBP=INR_GBP_value[dim(INR_GBP_value)[1],4]
###EUR
bbresult<- computeBollingerBands(INR_by_EUR, ndays, nsd=2)
#using band to build trading signals
buysignal <- INR_by_EUR< bbresult[,6]
sellsignal <- INR_by_EUR> bbresult[,5]
tradeindicator <- rep(0,n)
```

```
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_EUR_value=signal2value(INR_by_EUR,tradeindicator)
PNL_EUR=INR_EUR_value[dim(INR_EUR_value)[1],4]
valueNreturn=cbind(INR_USD_value[,4:5],INR_GBP_value[,4:5],INR_EUR_value[,4:5
 ])
valueNreturn1=c()
for (i in 1:n)
{
 if(all(valueNreturn[i,c(2,4,6)]==c(0,0,0))==F)
 valueNreturn1=rbind(valueNreturn1, valueNreturn[i,])
 }
bb.weights.value=valueNreturn1[,c(1,3,5)]/apply(valueNreturn1[,c(1,3,5)],1,sum)
bb.weighted.return=apply(valueNreturn1[,c(2,4,6)]*bb.weights.value,1,sum)
bb.sharpe.ratio=(mean(bb.weighted.return)-0.03/252)/sd(bb.weighted.return)*sqrt(252)
bb.sharpe.ratio_mat=rbind(bb.sharpe.ratio_mat,bb.sharpe.ratio)

###PNL
PNL_total=((PNL_USD-1)*PNL_USD+(PNL_GBP-1)*PNL_GBP+(PNL_EUR-
 1)*PNL_EUR)/(PNL_USD+PNL_GBP+PNL_EUR)
bb.PNL_total_mat=rbind(bb.PNL_total_mat,PNL_total)
}
#######################################
####Bonlinger band with transcation cost
##################################
signal2value<-function(x,signal)
{
 result=array(0,c(n,5))
 longposition=which(signal==1)
 shortposition=which(signal==-1)
 if(length(longposition)==0) return(NA)
 lastTrade = longposition[1]
 i=longposition[1]+1
 position = 1 #initial long position
 value = 1 #initial total value is 1
 #find te first long position
    while(i<=length(x)){
      if(signal[i]==position*(-1)){
            # set new last trading
            position = signal[i]#change position
            if(signal[i]==-1){#liquidate

                  value = value * (x[i]/x[lastTrade])*(1-0.0015)
                  return=x[i]/x[lastTrade]*(1-0.0015)-1
                  result[i,]=c(i, x[i], x[lastTrade],value,return)
                  }
            if(signal[i]==1){#buy order
             result[i,]=c(i, x[i], x[lastTrade],value,0)
            }
        lastTrade = i
        }else{#no trading
            result[i,]=c(i, x[i], x[lastTrade],value,0)}
        i = i+1
    }
    return(result)
}

bb.sharpe.ratio_mat=bb.PNL_total_mat=c()
for(ndays in seq(20,300,10))
{
computeBollingerBands <- function(x, ndays, nsd=2) {

  ## create a (normalised, but that's just candy) weight vector
  weights <- rep(1/ndays,ndays)
  weights2<- rev(1:ndays)
  weights2<-weights2/sum(weights2)
  ## and apply it as a one-sided moving average calculations, see help(filter)
  bbmiddle <- as.vector(filter(x, weights,
 method="convolution",side=1))
  bbmiddle2<- as.vector(filter(x, weights2,
 method="convolution",side=1))
  ## use var(x) = E(x^2) - E(x)^2 to compute rolling variances
  v <- filter(x^2, weights, method="convolution", side=1) - bbmiddle^2
  ## from which we calculate rolling standard deviations the usual way
  bbsd <- as.vector(sqrt(v))

  bbupper <- bbmiddle2 + nsd*bbsd        # upper Bollinger band
  bblower <- bbmiddle2 - nsd*bbsd        # lowet Bollinger band
  ## now extend the data frame with a few new columns
  D <- cbind(x, bbmiddle, bbmiddle2,bbsd, bbupper, bblower)

  return(D)                              # return the augmented data frame
}

bbresult<- computeBollingerBands(INR_by_USD, ndays, nsd=2)
#using band to build trading signals
buysignal <- INR_by_USD< bbresult[,6]
sellsignal <- INR_by_USD> bbresult[,5]
tradeindicator <- rep(0,n)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_USD_value=signal2value(INR_by_USD,tradeindicator)
```

```
PNL_USD=INR_USD_value[dim(INR_USD_value)[1],4]
####GBP
bbresult<- computeBollingerBands(INR_by_GBP, ndays, nsd=2)
#using band to build trading signals
buysignal <- INR_by_GBP< bbresult[,6]
sellsignal <- INR_by_GBP> bbresult[,5]
tradeindicator <- rep(0,n)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_GBP_value=signal2value(INR_by_GBP,tradeindicator)
PNL_GBP=INR_GBP_value[dim(INR_GBP_value)[1],4]
###EUR
bbresult<- computeBollingerBands(INR_by_EUR, ndays, nsd=2)
#using band to build trading signals
buysignal <- INR_by_EUR< bbresult[,6]
sellsignal <- INR_by_EUR> bbresult[,5]
tradeindicator <- rep(0,n)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_EUR_value=signal2value(INR_by_EUR,tradeindicator)
PNL_EUR=INR_EUR_value[dim(INR_EUR_value)[1],4]
valueNreturn=cbind(INR_USD_value[,4:5],INR_GBP_value[,4:5],INR_EUR_value[,4:5
 ])
valueNreturn1=c()
for (i in 1:n)
{
 if(all(valueNreturn[i,c(2,4,6)]==c(0,0,0))==F)
 valueNreturn1=rbind(valueNreturn1, valueNreturn[i,])
 }
bb.weights.value=valueNreturn1[,c(1,3,5)]/apply(valueNreturn1[,c(1,3,5)],1,sum)
bb.weighted.return=apply(valueNreturn1[,c(2,4,6)]*bb.weights.value,1,sum)
bb.sharpe.ratio=(mean(bb.weighted.return)-0.03/252)/sd(bb.weighted.return)*sqrt(252)
bb.sharpe.ratio_mat=rbind(bb.sharpe.ratio_mat,bb.sharpe.ratio)

###PNL
PNL_total=((PNL_USD-1)*PNL_USD+(PNL_GBP-1)*PNL_GBP+(PNL_EUR-
 1)*PNL_EUR)/(PNL_USD+PNL_GBP+PNL_EUR)
bb.PNL_total_mat=rbind(bb.PNL_total_mat,PNL_total)
}
################
#Resistance-Support(momentum based)
#without transaction cost
################
#################
#USD
#################
```

```
RS.sharpe.ratio_mat=RS.PNL_total_mat=c()
for (L in seq(20,300,10))
{
computeRS<-function(x,L)
{
 RS.lower=RS.upper=rep(0,nrows)
 for(i in 1:nrows)
 {
     if(i<=L)
     {
          RS.upper[i]=RS.lower[i]=NA
     }else{
 RS.upper[i]=max(x[(i-L):(i-1)])
 RS.lower[i]=min(x[(i-L):(i-1)])
 D=cbind(as.vector(x),RS.upper,RS.lower)
 }
 }
 return(D)

}
signal2value<-function(x,signal)
{
 result=array(0,c(nrows,5))
 longposition=which(signal==1)
 shortposition=which(signal==-1)
 if(length(longposition)==0) return(NA)
 lastTrade = longposition[1]
 i=longposition[1]+1
 position = 1 #initial long position
 value = 1 #initial total value is 1
 #find te first long position
    while(i<=length(x)){
     if(signal[i]==position*(-1)){
          # set new last trading
          position = signal[i]#change position
          if(signal[i]==-1){#liquidate

               value = value * (x[i]/x[lastTrade])
               return=x[i]/x[lastTrade]-1
               result[i,]=c(i, x[i], x[lastTrade],value,return)
               }
               if(signal[i]==1){#buy order
                result[i,]=c(i, x[i], x[lastTrade],value,0)
               }
          lastTrade = i
          }else{#no trading
```

```
                result[i,]=c(i, x[i], x[lastTrade],value,0)}
            i = i+1
        }
        return(result)
}
RSresult<- computeRS(INR_by_USD, L)
buysignal <- INR_by_USD> RSresult[,2]
sellsignal <- INR_by_USD< RSresult[,3]
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_USD_value=signal2value(INR_by_USD,tradeindicator)
PNL_USD=INR_USD_value[dim(INR_USD_value)[1],4]
##################
###GBP
##################
RSresult<- computeRS(INR_by_GBP, L)
buysignal <- INR_by_GBP> RSresult[,2]
sellsignal <- INR_by_GBP< RSresult[,3]
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_GBP_value=signal2value(INR_by_GBP,tradeindicator)
##################
###EUR
##################
RSresult<- computeRS(INR_by_EUR, L)
buysignal <- INR_by_EUR> RSresult[,2]
sellsignal <- INR_by_EUR< RSresult[,3]
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_EUR_value=signal2value(INR_by_EUR,tradeindicator)

valueNreturn=cbind(INR_USD_value[,4:5],INR_GBP_value[,4:5],INR_EUR_value[,4:5
 ])
valueNreturn1=c()
for (i in 1:nrows)
{
 if(all(valueNreturn[i,c(2,4,6)]==c(0,0,0))==F)
 valueNreturn1=rbind(valueNreturn1, valueNreturn[i,])
 }
 if(is.null(valueNreturn1)){RS.weights.value=valueNreturn1[,c(1,3,5)]/sum(valueNreturn
 1[,c(1,3,5)])
RS.weighted.return=sum(valueNreturn1[,c(2,4,6)]*RS.weights.value)}else{
if(nrow(valueNreturn1)==1){RS.weights.value=valueNreturn1[,c(1,3,5)]/sum(valueNretu
 rn1[,c(1,3,5)])
RS.weighted.return=sum(valueNreturn1[,c(2,4,6)]*RS.weights.value)}else{
RS.weights.value=valueNreturn1[,c(1,3,5)]/apply(valueNreturn1[,c(1,3,5)],1,sum)
RS.weighted.return=apply(valueNreturn1[,c(2,4,6)]*RS.weights.value,1,sum)
}}
RS.sharpe.ratio=(mean(RS.weighted.return)-0.03/252)/sd(RS.weighted.return)*sqrt(252)
RS.sharpe.ratio_mat=rbind(RS.sharpe.ratio_mat,RS.sharpe.ratio)

###PNL
PNL_total=((PNL_USD-1)*PNL_USD+(PNL_GBP-1)*PNL_GBP+(PNL_EUR-
 1)*PNL_EUR)/(PNL_USD+PNL_GBP+PNL_EUR)
RS.PNL_total_mat=rbind(RS.PNL_total_mat,PNL_total)
}
################
#Resistance-Support(momentum based)
#with transaction costs
################
#################
#USD
#################
RS.sharpe.ratio_mat=RS.PNL_total_mat=c()
for (L in seq(20,300,10))
{
computeRS<-function(x,L)
{
 RS.lower=RS.upper=rep(0,nrows)
 for(i in 1:nrows)
 {
     if(i<=L)
     {
          RS.upper[i]=RS.lower[i]=NA
     }else{
 RS.upper[i]=max(x[(i-L):(i-1)])
 RS.lower[i]=min(x[(i-L):(i-1)])
 D=cbind(as.vector(x),RS.upper,RS.lower)
 }
 }
 return(D)

}
signal2value<-function(x,signal)
{
 result=array(0,c(nrows,5))
 longposition=which(signal==1)
 shortposition=which(signal==-1)
 if(length(longposition)==0) return(NA)
 lastTrade = longposition[1]
 i=longposition[1]+1
 position = 1 #initial long position
 value = 1 #initial total value is 1
 #find te first long position
    while(i<=length(x)){
     if(signal[i]==position*(-1)){
          # set new last trading
          position = signal[i]#change position
          if(signal[i]==-1){#liquidate

               value = value * (x[i]/x[lastTrade])*(1-0.0015)
               return=x[i]/x[lastTrade]*(1-0.0015)-1
               result[i,]=c(i, x[i], x[lastTrade],value,return)
               }
               if(signal[i]==1){#buy order
                result[i,]=c(i, x[i], x[lastTrade],value,0)
               }
          lastTrade = i
          }else{#no trading
               result[i,]=c(i, x[i], x[lastTrade],value,0)}
          i = i+1
     }
     return(result)
}
RSresult<- computeRS(INR_by_USD, L)
buysignal <- INR_by_USD> RSresult[,2]
sellsignal <- INR_by_USD< RSresult[,3]
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_USD_value=signal2value(INR_by_USD,tradeindicator)
PNL_USD=INR_USD_value[dim(INR_USD_value)[1],4]
#################
###GBP
#################
RSresult<- computeRS(INR_by_GBP, L)
buysignal <- INR_by_GBP> RSresult[,2]
sellsignal <- INR_by_GBP< RSresult[,3]
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_GBP_value=signal2value(INR_by_GBP,tradeindicator)
#################
###EUR
#################
```

```
RSresult<- computeRS(INR_by_EUR, L)
buysignal <- INR_by_EUR> RSresult[,2]
sellsignal <- INR_by_EUR< RSresult[,3]
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_EUR_value=signal2value(INR_by_EUR,tradeindicator)

valueNreturn=cbind(INR_USD_value[,4:5],INR_GBP_value[,4:5],INR_EUR_value[,4:5
 ])
valueNreturn1=c()
for (i in 1:nrows)
{
 if(all(valueNreturn[i,c(2,4,6)]==c(0,0,0))==F)
 valueNreturn1=rbind(valueNreturn1, valueNreturn[i,])
 }
 if(is.null(valueNreturn1)){RS.weights.value=valueNreturn1[,c(1,3,5)]/sum(valueNreturn
 1[,c(1,3,5)])
RS.weighted.return=sum(valueNreturn1[,c(2,4,6)]*RS.weights.value)}else{
if(nrow(valueNreturn1)==1){RS.weights.value=valueNreturn1[,c(1,3,5)]/sum(valueNretu
 rn1[,c(1,3,5)])
RS.weighted.return=sum(valueNreturn1[,c(2,4,6)]*RS.weights.value)}else{
RS.weights.value=valueNreturn1[,c(1,3,5)]/apply(valueNreturn1[,c(1,3,5)],1,sum)
RS.weighted.return=apply(valueNreturn1[,c(2,4,6)]*RS.weights.value,1,sum)
}}
RS.sharpe.ratio=(mean(RS.weighted.return)-0.03/252)/sd(RS.weighted.return)*sqrt(252)
RS.sharpe.ratio_mat=rbind(RS.sharpe.ratio_mat,RS.sharpe.ratio)

###PNL
PNL_total=((PNL_USD-1)*PNL_USD+(PNL_GBP-1)*PNL_GBP+(PNL_EUR-
 1)*PNL_EUR)/(PNL_USD+PNL_GBP+PNL_EUR)
RS.PNL_total_mat=rbind(RS.PNL_total_mat,PNL_total)
}
################
#Resistance-Support(contrarian)
#without transaction cost
################
#################
#USD
#################
RS.sharpe.ratio_mat=RS.PNL_total_mat=c()
for (L in seq(20,300,10))
{
computeRS<-function(x,L)
{
 RS.lower=RS.upper=rep(0,nrows)
 for(i in 1:nrows)
 {
     if(i<=L)
     {
          RS.upper[i]=RS.lower[i]=NA
     }else{
 RS.upper[i]=max(x[(i-L):(i-1)])
 RS.lower[i]=min(x[(i-L):(i-1)])
 D=cbind(as.vector(x),RS.upper,RS.lower)
 }
 }
 return(D)

}
signal2value<-function(x,signal)
{
 result=array(0,c(nrows,5))
 longposition=which(signal==1)
 shortposition=which(signal==-1)
 if(length(longposition)==0) return(NA)
 lastTrade = longposition[1]
 i=longposition[1]+1
 position = 1 #initial long position
 value = 1 #initial total value is 1
 #find te first long position
   while(i<=length(x)){
     if(signal[i]==position*(-1)){
          # set new last trading
          position = signal[i]#change position
          if(signal[i]==-1){#liquidate

               value = value * (x[i]/x[lastTrade])
               return=x[i]/x[lastTrade]-1
               result[i,]=c(i, x[i], x[lastTrade],value,return)
               }
               if(signal[i]==1){#buy order
                result[i,]=c(i, x[i], x[lastTrade],value,0)
               }
          lastTrade = i
          }else{#no trading
               result[i,]=c(i, x[i], x[lastTrade],value,0)}
          i = i+1
     }
     return(result)
}
RSresult<- computeRS(INR_by_USD, L)
```

```
sellsignal <- INR_by_USD> RSresult[,2]
buysignal <- INR_by_USD< RSresult[,3]
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_USD_value=signal2value(INR_by_USD,tradeindicator)
PNL_USD=INR_USD_value[dim(INR_USD_value)[1],4]
##################
###GBP
##################
RSresult<- computeRS(INR_by_GBP, L)
sellsignal <- INR_by_GBP> RSresult[,2]
buysignal <- INR_by_GBP< RSresult[,3]
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_GBP_value=signal2value(INR_by_GBP,tradeindicator)
##################
###EUR
##################
RSresult<- computeRS(INR_by_EUR, L)
sellsignal <- INR_by_EUR> RSresult[,2]
buysignal <- INR_by_EUR< RSresult[,3]
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_EUR_value=signal2value(INR_by_EUR,tradeindicator)

valueNreturn=cbind(INR_USD_value[,4:5],INR_GBP_value[,4:5],INR_EUR_value[,4:5
 ])
valueNreturn1=c()
for (i in 1:nrows)
{
 if(all(valueNreturn[i,c(2,4,6)]==c(0,0,0))==F)
 valueNreturn1=rbind(valueNreturn1, valueNreturn[i,])
 }
 if(is.null(valueNreturn1)){RS.weights.value=valueNreturn1[,c(1,3,5)]/sum(valueNreturn
 1[,c(1,3,5)])
RS.weighted.return=sum(valueNreturn1[,c(2,4,6)]*RS.weights.value)}else{
if(nrow(valueNreturn1)==1){RS.weights.value=valueNreturn1[,c(1,3,5)]/sum(valueNretu
 rn1[,c(1,3,5)])
RS.weighted.return=sum(valueNreturn1[,c(2,4,6)]*RS.weights.value)}else{
RS.weights.value=valueNreturn1[,c(1,3,5)]/apply(valueNreturn1[,c(1,3,5)],1,sum)
RS.weighted.return=apply(valueNreturn1[,c(2,4,6)]*RS.weights.value,1,sum)
}}
RS.sharpe.ratio=(mean(RS.weighted.return)-0.03/252)/sd(RS.weighted.return)*sqrt(252)
RS.sharpe.ratio_mat=rbind(RS.sharpe.ratio_mat,RS.sharpe.ratio)

###PNL
PNL_total=((PNL_USD-1)*PNL_USD+(PNL_GBP-1)*PNL_GBP+(PNL_EUR-
 1)*PNL_EUR)/(PNL_USD+PNL_GBP+PNL_EUR)
RS.PNL_total_mat=rbind(RS.PNL_total_mat,PNL_total)
}

################
#Resistance-Support(contrarian)
#with transaction cost
################
#################
#USD
#################
RS.sharpe.ratio_mat=RS.PNL_total_mat=c()
for (L in seq(20,300,10))
{
computeRS<-function(x,L)
{
 RS.lower=RS.upper=rep(0,nrows)
 for(i in 1:nrows)
 {
     if(i<=L)
     {
            RS.upper[i]=RS.lower[i]=NA
     }else{
 RS.upper[i]=max(x[(i-L):(i-1)])
 RS.lower[i]=min(x[(i-L):(i-1)])
 D=cbind(as.vector(x),RS.upper,RS.lower)
 }
 }
 return(D)

}
signal2value<-function(x,signal)
{
 result=array(0,c(nrows,5))
 longposition=which(signal==1)
 shortposition=which(signal==-1)
 if(length(longposition)==0) return(NA)
 lastTrade = longposition[1]
 i=longposition[1]+1
 position = 1 #initial long position
 value = 1 #initial total value is 1
 #find te first long position
```

```
   while(i<=length(x)){
    if(signal[i]==position*(-1)){
        # set new last trading
        position = signal[i]#change position
        if(signal[i]==-1){#liquidate

            value = value * (x[i]/x[lastTrade])*(1-0.0015)
            return=x[i]/x[lastTrade]*(1-0.0015)-1
            result[i,]=c(i, x[i], x[lastTrade],value,return)
            }
            if(signal[i]==1){#buy order
             result[i,]=c(i, x[i], x[lastTrade],value,0)
            }
        lastTrade = i
        }else{#no trading
            result[i,]=c(i, x[i], x[lastTrade],value,0)}
        i = i+1
    }
    return(result)
}
RSresult<- computeRS(INR_by_USD, L)
sellsignal <- INR_by_USD> RSresult[,2]
buysignal <- INR_by_USD< RSresult[,3]
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_USD_value=signal2value(INR_by_USD,tradeindicator)
PNL_USD=INR_USD_value[dim(INR_USD_value)[1],4]
#################
###GBP
#################
RSresult<- computeRS(INR_by_GBP, L)
sellsignal <- INR_by_GBP> RSresult[,2]
buysignal <- INR_by_GBP< RSresult[,3]
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_GBP_value=signal2value(INR_by_GBP,tradeindicator)
#################
###EUR
#################
RSresult<- computeRS(INR_by_EUR, L)
sellsignal <- INR_by_EUR> RSresult[,2]
buysignal <- INR_by_EUR< RSresult[,3]
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
```

```
tradeindicator[sellsignal]=-1
INR_EUR_value=signal2value(INR_by_EUR,tradeindicator)

valueNreturn=cbind(INR_USD_value[,4:5],INR_GBP_value[,4:5],INR_EUR_value[,4:5
 ])
valueNreturn1=c()
for (i in 1:nrows)
{
 if(all(valueNreturn[i,c(2,4,6)]==c(0,0,0))==F)
 valueNreturn1=rbind(valueNreturn1, valueNreturn[i,])
 }
 if(is.null(valueNreturn1)){RS.weights.value=valueNreturn1[,c(1,3,5)]/sum(valueNreturn
 1[,c(1,3,5)])
RS.weighted.return=sum(valueNreturn1[,c(2,4,6)]*RS.weights.value)}else{
if(nrow(valueNreturn1)==1){RS.weights.value=valueNreturn1[,c(1,3,5)]/sum(valueNretu
 rn1[,c(1,3,5)])
RS.weighted.return=sum(valueNreturn1[,c(2,4,6)]*RS.weights.value)}else{
RS.weights.value=valueNreturn1[,c(1,3,5)]/apply(valueNreturn1[,c(1,3,5)],1,sum)
RS.weighted.return=apply(valueNreturn1[,c(2,4,6)]*RS.weights.value,1,sum)
}}
RS.sharpe.ratio=(mean(RS.weighted.return)-0.03/252)/sd(RS.weighted.return)*sqrt(252)
RS.sharpe.ratio_mat=rbind(RS.sharpe.ratio_mat,RS.sharpe.ratio)

###PNL
PNL_total=((PNL_USD-1)*PNL_USD+(PNL_GBP-1)*PNL_GBP+(PNL_EUR-
 1)*PNL_EUR)/(PNL_USD+PNL_GBP+PNL_EUR)
RS.PNL_total_mat=rbind(RS.PNL_total_mat,PNL_total)
}


#############
##Momentum
#############
#################
#USD
#################
mom.sharpe.ratio_mat=mom.PNL_total_mat=c()
nrows=dim(Rates)[1]
for(n in seq(50,300,50))
{
 for(m in seq(20,(n-1),10))
 {
computeMA.mom<-function(x,m,n)
{
 weights.short <- rep(1/m,m)
 weights.long<-rep(1/n,n)
 ma.series.short=as.vector(filter(x, weights.short,
 method="convolution",side=1))
 ma.series.long=as.vector(filter(x, weights.long,
 method="convolution",side=1))
 D=cbind(as.vector(x),ma.series.short,ma.series.long)

 return(D)
}


res=computeMA.mom(INR_by_USD,m,n)
diff.ma=res[,3]-res[,2]#long term-short term
buysignal<-diff.ma<0#short term moving average crosses longterm from above
sellsignal<-diff.ma>0#short term moving average crosses longterm from above
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1

signal2value<-function(x,signal)
{
 result=array(0,c(nrows,5))
 longposition=which(signal==1)
 shortposition=which(signal==-1)
 if(length(longposition)==0) return(NA)
 lastTrade = longposition[1]
 i=longposition[1]+1
 position = 1 #initial long position
 value = 1 #initial total value is 1
 #find te first long position
   while(i<=length(x)){
    if(signal[i]==position*(-1)){
        # set new last trading
        position = signal[i]#change position
        if(signal[i]==-1){#liquidate

            value = value * (x[i]/x[lastTrade])
            return=x[i]/x[lastTrade]-1
            result[i,]=c(i, x[i], x[lastTrade],value,return)
            }
            if(signal[i]==1){#buy order
             result[i,]=c(i, x[i], x[lastTrade],value,0)
            }
        lastTrade = i
        }else{#no trading
            result[i,]=c(i, x[i], x[lastTrade],value,0)}
        i = i+1
    }
    return(result)
}

INR_USD_value=signal2value(INR_by_USD,tradeindicator)
PNL_USD=INR_USD_value[dim(INR_USD_value)[1],4]
#################
####GBP
#################
res=computeMA.mom(INR_by_GBP,m,n)
diff.ma=res[,3]-res[,2]#long term-short term
buysignal<-diff.ma<0#short term moving average crosses longterm from above
sellsignal<-diff.ma>0#short term moving average crosses longterm from above
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_GBP_value=signal2value(INR_by_GBP,tradeindicator)
PNL_GBP=INR_GBP_value[dim(INR_GBP_value)[1],4]
#################
#####EUR
#################
res=computeMA.mom(INR_by_EUR,m,n)
diff.ma=res[,3]-res[,2]#long term-short term
buysignal<-diff.ma<0#short term moving average crosses longterm from above
sellsignal<-diff.ma>0#short term moving average crosses longterm from above
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_EUR_value=signal2value(INR_by_EUR,tradeindicator)
PNL_EUR=INR_EUR_value[dim(INR_EUR_value)[1],4]

valueNreturn=cbind(INR_USD_value[,4:5],INR_GBP_value[,4:5],INR_EUR_value[,4:5
 ])
valueNreturn1=c()
for (i in 1:nrows)
{
 if(all(valueNreturn[i,c(2,4,6)]==c(0,0,0))==F)
 valueNreturn1=rbind(valueNreturn1, valueNreturn[i,])
 }
mom.weights.value=valueNreturn1[,c(1,3,5)]/apply(valueNreturn1[,c(1,3,5)],1,sum)
mom.weighted.return=apply(valueNreturn1[,c(2,4,6)]*mom.weights.value,1,sum)
mom.sharpe.ratio=c(n,m,(mean(mom.weighted.return)-
 0.03/252)/sd(mom.weighted.return)*sqrt(252))
mom.sharpe.ratio_mat=rbind(mom.sharpe.ratio_mat,mom.sharpe.ratio)

###PNL
```

```
PNL_total=c(n,m,((PNL_USD-1)*PNL_USD+(PNL_GBP-1)*PNL_GBP+(PNL_EUR-1)*PNL_EUR)/(PNL_USD+PNL_GBP+PNL_EUR))
mom.PNL_total_mat=rbind(mom.PNL_total_mat,PNL_total)
}
}


#############
##Momentum
##with transaction cost
#############
#################
#USD
#################
mom.sharpe.ratio_mat=mom.PNL_total_mat=c()
nrows=dim(Rates)[1]
for(n in seq(50,300,50))
{
 for(m in seq(20,(n-1),10))
  {

res=computeMA.mom(INR_by_USD,m,n)
diff.ma=res[,3]-res[,2]#long term-short term
buysignal<-diff.ma<0#short term moving average crosses longterm from above
sellsignal<-diff.ma>0#short term moving average crosses longterm from above
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1

signal2value<-function(x,signal)
{
 result=array(0,c(nrows,5))
 longposition=which(signal==1)
 shortposition=which(signal==-1)
 if(length(longposition)==0) return(NA)
 lastTrade = longposition[1]
 i=longposition[1]+1
 position = 1 #initial long position
 value = 1 #initial total value is 1
 #find te first long position
    while(i<=length(x)){
      if(signal[i]==position*(-1)){
            # set new last trading
            position = signal[i]#change position
            if(signal[i]==-1){#liquidate
            value = value * (x[i]/x[lastTrade])*(1-0.0015)
            return=x[i]/x[lastTrade]*(1-0.0015)-1
            result[i,]=c(i, x[i], x[lastTrade],value,return)
            }
            if(signal[i]==1){#buy order
             result[i,]=c(i, x[i], x[lastTrade],value,0)
            }
        lastTrade = i
        }else{#no trading
            result[i,]=c(i, x[i], x[lastTrade],value,0)}
        i = i+1
    }
    return(result)
}

INR_USD_value=signal2value(INR_by_USD,tradeindicator)
PNL_USD=INR_USD_value[dim(INR_USD_value)[1],4]
##################
####GBP
##################
res=computeMA.mom(INR_by_GBP,m,n)
diff.ma=res[,3]-res[,2]#long term-short term
buysignal<-diff.ma<0#short term moving average crosses longterm from above
sellsignal<-diff.ma>0#short term moving average crosses longterm from above
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_GBP_value=signal2value(INR_by_GBP,tradeindicator)
PNL_GBP=INR_GBP_value[dim(INR_GBP_value)[1],4]
##################
#####EUR
##################
res=computeMA.mom(INR_by_EUR,m,n)
diff.ma=res[,3]-res[,2]#long term-short term
buysignal<-diff.ma<0#short term moving average crosses longterm from above
sellsignal<-diff.ma>0#short term moving average crosses longterm from above
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_EUR_value=signal2value(INR_by_EUR,tradeindicator)
PNL_EUR=INR_EUR_value[dim(INR_EUR_value)[1],4]

valueNreturn=cbind(INR_USD_value[,4:5],INR_GBP_value[,4:5],INR_EUR_value[,4:5
 ])
valueNreturn1=c()
for (i in 1:nrows)
```

```
{
 if(all(valueNreturn[i,c(2,4,6)]==c(0,0,0))==F)
 valueNreturn1=rbind(valueNreturn1, valueNreturn[i,])
 }
mom.weights.value=valueNreturn1[,c(1,3,5)]/apply(valueNreturn1[,c(1,3,5)],1,sum)
mom.weighted.return=apply(valueNreturn1[,c(2,4,6)]*mom.weights.value,1,sum)
mom.sharpe.ratio=c(n,m,(mean(mom.weighted.return)-
 0.03/252)/sd(mom.weighted.return)*sqrt(252))
mom.sharpe.ratio_mat=rbind(mom.sharpe.ratio_mat,mom.sharpe.ratio)

###PNL
PNL_total=c(n,m,((PNL_USD-1)*PNL_USD+(PNL_GBP-1)*PNL_GBP+(PNL_EUR-
 1)*PNL_EUR)/(PNL_USD+PNL_GBP+PNL_EUR))
mom.PNL_total_mat=rbind(mom.PNL_total_mat,PNL_total)
}
}

################
###Question 2(b)
###Oscillator Rule without transaction cost
################

rsi.sharpe.ratio_mat=rsi.PNL_total_mat=c()
for(L in seq(20,300,10))
{
RSI<-function(x,L)
{
 s=c(NA, diff(x))
up.series=s*ifelse(s>0,yes=1,no=0)
down.series=s*ifelse(s<0,yes=1,no=0)
weights <- rep(1,L)
 U=as.vector(filter(up.series, weights,                    method="convolution",side=1))
 D=as.vector(filter(down.series, weights,
 method="convolution",side=1))
 rsi=100*(U/(U+D))
 res=cbind(as.vector(x),U,D,rsi)
}

signal2value<-function(x,signal)
{
 result=array(0,c(nrows,5))
 longposition=which(signal==1)
 shortposition=which(signal==-1)
 if(length(longposition)==0) return(NA)
 lastTrade = longposition[1]
 i=longposition[1]+1
 position = 1 #initial long position
 value = 1 #initial total value is 1
 #find te first long position
    while(i<=length(x)){
      if(signal[i]==position*(-1)){
            # set new last trading
            position = signal[i]#change position
            if(signal[i]==-1){#liquidate

                  value = value * (x[i]/x[lastTrade])
                  return=x[i]/x[lastTrade]-1
                  result[i,]=c(i, x[i], x[lastTrade],value,return)
                  }
                  if(signal[i]==1){#buy order
                   result[i,]=c(i, x[i], x[lastTrade],value,0)
                  }
            lastTrade = i
            }else{#no trading
                  result[i,]=c(i, x[i], x[lastTrade],value,0)}
            i = i+1
      }
      return(result)
}

RSIresult=RSI(INR_by_USD,L)
buysignal<-RSIresult[,4]<30
sellsignal<-RSIresult[,4]>70#short term moving average crosses longterm from above
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_USD_value=signal2value(INR_by_USD,tradeindicator)
PNL_USD=INR_USD_value[dim(INR_USD_value)[1],4]
####GBP

RSIresult=RSI(INR_by_GBP,L)
buysignal<-RSIresult[,4]<30
sellsignal<-RSIresult[,4]>70#short term moving average crosses longterm from above
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_GBP_value=signal2value(INR_by_GBP,tradeindicator)
PNL_GBP=INR_GBP_value[dim(INR_GBP_value)[1],4]

###EUR
RSIresult=RSI(INR_by_EUR,L)
buysignal<-RSIresult[,4]<30
```

```
sellsignal<-RSIresult[,4]>70#short term moving average crosses longterm from above
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_EUR_value=signal2value(INR_by_EUR,tradeindicator)
PNL_EUR=INR_EUR_value[dim(INR_EUR_value)[1],4]

valueNreturn=cbind(INR_USD_value[,4:5],INR_GBP_value[,4:5],INR_EUR_value[,4:5
 ])
valueNreturn1=c()
for (i in 1:nrows)
{
 if(all(valueNreturn[i,c(2,4,6)]==c(0,0,0))==F)
 valueNreturn1=rbind(valueNreturn1, valueNreturn[i,])
 }
rsi.weights.value=valueNreturn1[,c(1,3,5)]/apply(valueNreturn1[,c(1,3,5)],1,sum)
rsi.weighted.return=apply(valueNreturn1[,c(2,4,6)]*rsi.weights.value,1,sum)
rsi.sharpe.ratio=(mean(rsi.weighted.return)-0.03/252)/sd(rsi.weighted.return)*sqrt(252)
rsi.sharpe.ratio_mat=rbind(rsi.sharpe.ratio_mat,rsi.sharpe.ratio)
###PNL
PNL_total=((PNL_USD-1)*PNL_USD+(PNL_GBP-1)*PNL_GBP+(PNL_EUR-
 1)*PNL_EUR)/(PNL_USD+PNL_GBP+PNL_EUR)
rsi.PNL_total_mat=rbind(rsi.PNL_total_mat,PNL_total)
}


################
###Question 2(b)
###Oscillator Rule with transaction cost
################

rsi.sharpe.ratio_mat=rsi.PNL_total_mat=c()
for(L in seq(20,300,10))
{

signal2value<-function(x,signal)
{
 result=array(0,c(nrows,5))
 longposition=which(signal==1)
 shortposition=which(signal==-1)
 if(length(longposition)==0) return(NA)
 lastTrade = longposition[1]
 i=longposition[1]+1
 position = 1 #initial long position
 value = 1 #initial total value is 1
 #find te first long position
```

```
    while(i<=length(x)){
     if(signal[i]==position*(-1)){
          # set new last trading
          position = signal[i]#change position
          if(signal[i]==-1){#liquidate

               value = value * (x[i]/x[lastTrade])*(1-0.0015)
               return=x[i]/x[lastTrade]*(1-0.0015)-1
               result[i,]=c(i, x[i], x[lastTrade],value,return)
               }
               if(signal[i]==1){#buy order
                result[i,]=c(i, x[i], x[lastTrade],value,0)
               }
          lastTrade = i
          }else{#no trading
               result[i,]=c(i, x[i], x[lastTrade],value,0)}
          i = i+1
     }
     return(result)
}

RSIresult=RSI(INR_by_USD,L)
buysignal<-RSIresult[,4]<30
sellsignal<-RSIresult[,4]>70#short term moving average crosses longterm from above
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_USD_value=signal2value(INR_by_USD,tradeindicator)
PNL_USD=INR_USD_value[dim(INR_USD_value)[1],4]
####GBP

RSIresult=RSI(INR_by_GBP,L)
buysignal<-RSIresult[,4]<30
sellsignal<-RSIresult[,4]>70#short term moving average crosses longterm from above
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
tradeindicator[sellsignal]=-1
INR_GBP_value=signal2value(INR_by_GBP,tradeindicator)
PNL_GBP=INR_GBP_value[dim(INR_GBP_value)[1],4]

###EUR
RSIresult=RSI(INR_by_EUR,L)
buysignal<-RSIresult[,4]<30
sellsignal<-RSIresult[,4]>70#short term moving average crosses longterm from above
tradeindicator <- rep(0,nrows)
tradeindicator[buysignal]=1
```

```
tradeindicator[sellsignal]=-1
INR_EUR_value=signal2value(INR_by_EUR,tradeindicator)
PNL_EUR=INR_EUR_value[dim(INR_EUR_value)[1],4]

valueNreturn=cbind(INR_USD_value[,4:5],INR_GBP_value[,4:5],INR_EUR_value[,4:5
 ])
valueNreturn1=c()
for (i in 1:nrows)
{
 if(all(valueNreturn[i,c(2,4,6)]==c(0,0,0))==F)
 valueNreturn1=rbind(valueNreturn1, valueNreturn[i,])
 }
rsi.weights.value=valueNreturn1[,c(1,3,5)]/apply(valueNreturn1[,c(1,3,5)],1,sum)
rsi.weighted.return=apply(valueNreturn1[,c(2,4,6)]*rsi.weights.value,1,sum)
rsi.sharpe.ratio=(mean(rsi.weighted.return)-0.03/252)/sd(rsi.weighted.return)*sqrt(252)
rsi.sharpe.ratio_mat=rbind(rsi.sharpe.ratio_mat,rsi.sharpe.ratio)
###PNL
PNL_total=((PNL_USD-1)*PNL_USD+(PNL_GBP-1)*PNL_GBP+(PNL_EUR-
 1)*PNL_EUR)/(PNL_USD+PNL_GBP+PNL_EUR)
rsi.PNL_total_mat=rbind(rsi.PNL_total_mat,PNL_total)
}

#################
###Question 3
#################
###normalized each foreign curreny
nUSD=rep(0,nrows)
nUSD[1]=1#inverst 1 Rupee
for (i in 2:nrows)
{
 nUSD[i]=nUSD[i-1]*INR_by_USD[i]/INR_by_USD[i-1]
}
nGBP=rep(0,nrows)
nGBP[1]=1#inverst 1 Rupee
for (i in 2:nrows)
{
 nGBP[i]=nGBP[i-1]*INR_by_GBP[i]/INR_by_GBP[i-1]
}
nEUR=rep(0,nrows)
nEUR[1]=1#inverst 1 Rupee
for (i in 2:nrows)
{
 nEUR[i]=nEUR[i-1]*INR_by_EUR[i]/INR_by_EUR[i-1]
}
###calcluate average square and std
month=rep(c(1:48),each=21)
```

```
Rates1=cbind(Rates,month[1:nrows])
month=Rates1[,5]
aveUSD_GBP=stdUSD_GBP=rep(0,nrows)
for( i in 1: 48)
{
 if(i<=3){aveUSD_GBP[(i*21-20):min((i*21),nrows)]=NA
      stdUSD_GBP[(i*21-20):min((i*21),nrows)]=NA}else{
      diff2=(nUSD[((i-4)*21):((i-1)*21)]-nGBP[((i-4)*21):((i-1)*21)])^2
 aveUSD_GBP[(i*21-20):min((i*21),nrows)]=mean(diff2)
 stdUSD_GBP[(i*21-20):min((i*21),nrows)]=sd(diff2)
 }
}
aveUSD_EUR=stdUSD_EUR=rep(0,nrows)
for( i in 1: 48)
{
 if(i<=3){aveUSD_EUR[(i*21-20):(i*21)]=NA
      stdUSD_EUR[(i*21-20):(i*21)]=NA}else{
      diff2=(nUSD[((i-4)*21):((i-1)*21)]-nEUR[((i-4)*21):((i-1)*21)])^2
 aveUSD_EUR[(i*21-20):min((i*21),nrows)]=mean(diff2)
 stdUSD_EUR[(i*21-20):min((i*21),nrows)]=sd(diff2)
 }
}
aveEUR_GBP=stdEUR_GBP=rep(0,nrows)
for( i in 1: 48)
{
 if(i<=3){aveEUR_GBP[(i*21-20):(i*21)]=NA
      stdEUR_GBP[(i*21-20):(i*21)]=NA}else{
      diff2=(nEUR[((i-4)*21):((i-1)*21)]-nGBP[((i-4)*21):((i-1)*21)])^2
 aveEUR_GBP[(i*21-20):min((i*21),nrows)]=mean(diff2)
 stdEUR_GBP[(i*21-20):min((i*21),nrows)]=sd(diff2)
 }
}
ave_total=cbind(aveUSD_GBP,aveUSD_EUR,aveEUR_GBP)
std_total=cbind(stdUSD_GBP,stdUSD_EUR,stdEUR_GBP)
min_ave=apply(ave_total,1,min)
no.pair=ifelse(min_ave==aveUSD_GBP,1,0)*1+ifelse(min_ave==aveUSD_EUR,1,0)*2
 +ifelse(min_ave==aveEUR_GBP,1,0)*3#1=USD_GBP;2=USD_EUR;3=EUR_GBP
threshold=2*(ifelse(min_ave==aveUSD_GBP,1,0)*stdUSD_GBP+ifelse(min_ave==ave
 USD_EUR,1,0)*stdUSD_EUR+ifelse(min_ave==aveEUR_GBP,1,0)*stdEUR_GBP)

###main pair trading function
sig1=nUSD-nGBP
sig2=nUSD-nEUR
sig3=nEUR-nGBP
sig_mat=cbind(sig1,sig2,sig3)
diff1=(nUSD-nGBP)^2
```

```
diff2=(nUSD-nEUR)^2
diff3=(nEUR-nGBP)^2
diff_mat=cbind(diff1,diff2,diff3)
trade_series=cheap_series=exp_series=cheap_no=exp_no=rep(0,nrows)
for(i in 1:nrows)
{
if(i<=63){trade_series[i]=cheap_series=exp_series=cheap_no=exp_no=NA}else{
trade_series[i]=diff_mat[i,no.pair[i]]
if(no.pair[i]==1){cheap_series[i]=min(c(nUSD[i],nGBP[i]))
 exp_series[i]=max(c(nUSD[i],nGBP[i]))
 cheap_no[i]=ifelse(nUSD[i]==cheap_series[i],1,0)*1+ifelse(nGBP[i]==cheap_series[i],
 1,0)*2
 exp_no[i]=ifelse(nUSD[i]==exp_series[i],1,0)*1+ifelse(nGBP[i]==exp_series[i],1,0)*2}
 else{
     if(no.pair[i]==2){cheap_series[i]=min(c(nUSD[i],nEUR[i]))
 exp_series[i]=max(c(nUSD[i],nEUR[i]))
 cheap_no[i]=ifelse(nUSD[i]==cheap_series[i],1,0)*1+ifelse(nEUR[i]==cheap_series[i],
 1,0)*3
 exp_no[i]=ifelse(nUSD[i]==exp_series[i],1,0)*1+ifelse(nEUR[i]==exp_series[i],1,0)*3}
 else{cheap_series[i]=min(c(nEUR[i],nGBP[i]))
 exp_series[i]=max(c(nEUR[i],nGBP[i]))
 cheap_no[i]=ifelse(nEUR[i]==cheap_series[i],1,0)*3+ifelse(nGBP[i]==cheap_series[i],
 1,0)*2
 exp_no[i]=ifelse(nEUR[i]==exp_series[i],1,0)*3+ifelse(nGBP[i]==exp_series[i],1,0)*2
}
}
}
}
for(i in 1:nrows)
{
 if(trade_series[i]>threshold[i])

}
buysignal.cheap<-trade_series>threshold
sellsignal.exp<-trade_series>threshold
sellsignal.cheap<-c(NA,((diff(cheap_no)!=0)&(diff(no.pair)==0)))
buysignal.exp<-c(NA,((diff(cheap_no)!=0)&(diff(no.pair)==0)))
tradeindicator.cheap <- rep(0,nrows)
tradeindicator.cheap[buysignal.cheap]=1
tradeindicator.cheap[sellsignal.cheap]=-1
tradeindicator.exp <- rep(0,nrows)
tradeindicator.exp[buysignal.exp]=1
tradeindicator.exp[sellsignal.exp]=-1

signal2value.cheap<-function(x,signal)
{
 result=array(0,c(nrows,6))
 converge=rep(0,nrows)
 longposition=which(signal==1)
 shortposition=which(signal==-1)
 if(length(longposition)==0) return(NA)
 lastTrade = longposition[1]
 i=longposition[1]+1
 position = 1 #initial long position
 value = 1 #initial total value is 1
 #find te first long position
   while(i<=length(x)){
    if(signal[i]==position*(-1)){
        # set new last trading
        position = signal[i]#change position
        if(signal[i]==-1){#liquidate
            if((i-lastTrade)<=42){

            value = value * (x[i]/x[lastTrade])
            return=x[i]/x[lastTrade]-1
            converge[i]=1
            result[i,]=c(i, x[i], x[lastTrade],value,return,converge[i])}else{
                value = value * (x[lastTrade+42]/x[lastTrade])
            return=x[lastTrade+42]/x[lastTrade]-1
            converge[i]=0
            result[i,]=c(i, x[i], x[lastTrade],value,return,converge[i])
                }
            }
            if(signal[i]==1){#buy order
             result[i,]=c(i, x[i], x[lastTrade],value,0,NA)
            }
        lastTrade = i
        }
        else{
            #no trading
            result[i,]=c(i, x[i], x[lastTrade],value,0,NA)
            }
        i = i+1
        }
    return(result)
}


signal2value.exp<-function(x,signal)
{
 result=array(0,c(nrows,6))
 converge=rep(0,nrows)
 longposition=which(signal==1)
 shortposition=which(signal==-1)
 if(length(shortposition)==0) return(NA)
 lastTrade = shortposition[1]
 i=shortposition[1]+1
 position = -1 #initial short position
 value = 1 #initial total value is 1
 #find te first long position
    while(i<=length(x)){
     if(signal[i]==position*(-1)){
          # set new last trading
          position = signal[i]#change position
          if(signal[i]==1){#liquidate
               if((i-lastTrade)<=42){

               value = value * (x[lastTrade]/x[i])
               return=x[lastTrade]/x[i]-1
               converge[i]=1
               result[i,]=c(i, x[i], x[lastTrade],value,return,converge[i])}else{
                    value = value * (x[lastTrade]/x[lastTrade+42])
               return=x[lastTrade]/x[lastTrade+42]-1
               converge[i]=0
               result[i,]=c(i, x[i], x[lastTrade],value,return,converge[i])
                    }
               }
               if(signal[i]==-1){#sell order
                result[i,]=c(i, x[i], x[lastTrade],value,0,NA)
               }
          lastTrade = i
          }
          else{
               #no trading
               result[i,]=c(i, x[i], x[lastTrade],value,0,NA)
               }
          i = i+1
          }
     return(result)
}

res.cheap=signal2value.cheap(cheap_series,tradeindicator.cheap)
cheap_value=res.cheap[dim(res.cheap)[1],4]
res.exp=signal2value.exp(exp_series,tradeindicator.exp)
exp_value=res.exp[dim(res.exp)[1],4]

valueNreturn=cbind(res.cheap[,4:5],res.exp[,4:5])
valueNreturn1=c()
```

```
for (i in 1:nrows)
{
 if(all(valueNreturn[i,c(2,4)]==c(0,0))==F)
 valueNreturn1=rbind(valueNreturn1, valueNreturn[i,])
 }
pair.weights.value=valueNreturn1[,c(1,3)]/apply(valueNreturn1[,c(1,3)],1,sum)
pair.weighted.return=apply(valueNreturn1[,c(2,4)]*pair.weights.value,1,sum)
pair.sharpe.ratio=(mean(pair.weighted.return)-
 0.03/252)/sd(pair.weighted.return)*sqrt(252)
###PNL
PNL_pair=((cheap_value-1)*cheap_value+(exp_value-
 1)*exp_value)/sum(cheap_value+exp_value)

#####with transaction cost
signal2value.cheap<-function(x,signal)
{
 result=array(0,c(nrows,6))
 converge=rep(0,nrows)
 longposition=which(signal==1)
 shortposition=which(signal==-1)
 if(length(longposition)==0) return(NA)
 lastTrade = longposition[1]
 i=longposition[1]+1
 position = 1 #initial long position
 value = 1 #initial total value is 1
 #find te first long position
    while(i<=length(x)){
      if(signal[i]==position*(-1)){
           # set new last trading
           position = signal[i]#change position
           if(signal[i]==-1){#liquidate
                if((i-lastTrade)<=42){

                value = value * (x[i]/x[lastTrade])*(1-0.0015)
                return=x[i]/x[lastTrade]*(1-0.0015)-1
                converge[i]=1
                result[i,]=c(i, x[i], x[lastTrade],value,return,converge[i])}else{
                     value = value * (x[lastTrade+42]/x[lastTrade])*(1-0.0015)
                return=x[lastTrade+42]/x[lastTrade]*(1-0.0015)-1
                converge[i]=0
                result[i,]=c(i, x[i], x[lastTrade],value,return,converge[i])
                     }
                }
                if(signal[i]==1){#buy order
                 result[i,]=c(i, x[i], x[lastTrade],value,0,NA)
                }
```

```
          lastTrade = i
          }
          else{
              #no trading
              result[i,]=c(i, x[i], x[lastTrade],value,0,NA)
              }
          i = i+1
          }
     return(result)
}


signal2value.exp<-function(x,signal)
{
 result=array(0,c(nrows,6))
 converge=rep(0,nrows)
 longposition=which(signal==1)
 shortposition=which(signal==-1)
 if(length(shortposition)==0) return(NA)
 lastTrade = shortposition[1]
 i=shortposition[1]+1
 position = -1 #initial short position
 value = 1 #initial total value is 1
 #find te first long position
    while(i<=length(x)){
     if(signal[i]==position*(-1)){
          # set new last trading
          position = signal[i]#change position
          if(signal[i]==1){#liquidate
              if((i-lastTrade)<=42){

              value = value * (x[lastTrade]/x[i])*(1-0.0015)
              return=x[lastTrade]/x[i]*(1-0.0015)-1
              converge[i]=1
              result[i,]=c(i, x[i], x[lastTrade],value,return,converge[i])}else{
                  value = value * (x[lastTrade]/x[lastTrade+42])*(1-0.0015)
              return=x[lastTrade]/x[lastTrade+42]*(1-0.0015)-1
              converge[i]=0
              result[i,]=c(i, x[i], x[lastTrade],value,return,converge[i])
                  }
              }
              if(signal[i]==-1){#sell order
               result[i,]=c(i, x[i], x[lastTrade],value,0,NA)
              }
          lastTrade = i
          }
```

```
        else{
            #no trading
            result[i,]=c(i, x[i], x[lastTrade],value,0,NA)
            }
        i = i+1
        }
    return(result)
}

res.cheap=signal2value.cheap(cheap_series,tradeindicator.cheap)
cheap_value=res.cheap[dim(res.cheap)[1],4]
res.exp=signal2value.exp(exp_series,tradeindicator.exp)
exp_value=res.exp[dim(res.exp)[1],4]

valueNreturn=cbind(res.cheap[,4:5],res.exp[,4:5])
valueNreturn1=c()
for (i in 1:nrows)
{
 if(all(valueNreturn[i,c(2,4)]==c(0,0))==F)
 valueNreturn1=rbind(valueNreturn1, valueNreturn[i,])
 }
pair.weights.value=valueNreturn1[,c(1,3)]/apply(valueNreturn1[,c(1,3)],1,sum)
pair.weighted.return=apply(valueNreturn1[,c(2,4)]*pair.weights.value,1,sum)
pair.sharpe.ratio=(mean(pair.weighted.return)-
 0.03/252)/sd(pair.weighted.return)*sqrt(252)
###PNL
PNL_pair=((cheap_value-1)*cheap_value+(exp_value-
 1)*exp_value)/sum(cheap_value+exp_value)

####graphs
plot(trade_series,type="l",col="blue",ylim=c(0,0.06),xlab="Days",ylab="")
par(new=T)

plot(threshold,type="l",col="black",ylim=c(0,0.06),,xlab="Days",ylab="")
```