stats242 HW2 code

Michael Downs

July 24, 2015

setup

```
library(highfrequency)
library(timeDate)
library(TTR)
library(tseries)
library(xts)
library(quantmod)
library(PerformanceAnalytics)
library(caret)
library(fUnitRoots)
library(urca)
library(lubridate)
library(mondate)
library(MTS)
library(timeSeries)
library(car)

setwd("~/Documents/Pers/Ed/Courses/stats242/homework")
price=read.csv("Bankdata.csv",colClasses=c("character","numeric","numeric","numeric"))
price[nchar(price[,1])==7,1]=paste0("0",price[nchar(price[,1])==7,1])
price[,1]=as.Date(price[,1],format="%m%d%Y")
price=as.xts(price[,-1],price[,1])
rtrn=diff(log(price))

# intialize global variables
rf=0.03
rf_t=0.0001173 # based on 252 NYSE trading days / year i.e., 1.0001173^252=1.03
bp=0.0001
```

evaluate stationarity

```
par(mfrow=c(3,2))
pacf(price$JPM);pacf(price$WFC)
pacf(diff(price$JPM)[-1]);pacf(diff(price$WFC)[-1])
pacf(diff(log(price$JPM))[-1]);pacf(diff(log(price$WFC))[-1])
```

```
test_stationarity=function(x,axis=NA){
     library(fUnitRoots)
     library(urca)
     result=as.data.frame(matrix(NA,length(x),7))
     names(result)=c("period","test type","adf pval","adf result",
                     "test type","kpss pval","kpss result")
     j=1
     for(i in 1:length(x)){
          result[j,1]=axis[i]
          result[j,2]="root"
          result[j,3]=adfTest(as.numeric(x[[i]]),lags=12,type="ct")@test$p.value
          result[j,4]=if(result[j,3]<0.1){"stationary"}else{"not"}
          result[j,5]="trend"
          tmp=ur.kpss(as.numeric(x[[i]]),lags="long",type="tau")
          result[j,6]=tmp@teststat
          result[j,7]=if(result[j,6]<tmp@cval[2]){"stationary"}else{"not"}
          j=j+1
     }
     return(result)
}

test_stationarity(list(as.numeric(price$JPM),as.numeric(price$WFC)),c("JPM","WFC"))
test_stationarity(list(diff(price$JPM),diff(price$WFC)),c("JPM","WFC"))
test_stationarity(list(diff(log(price$JPM)),diff(log(price$WFC))),c("JPM","WFC"))
```

establish buy/hold benchmark

```
calc_bh=function(x,amt,type="diff(log)"){
     z=as.numeric(x)
     if(type=="diff(log)"){z=diff(log(z))}
     else if(type=="diff()"){z=diff(z)}
     rslt=rep(0,dim(x)[1]);rslt[1]=amt
     for(i in 1:length(z)){
          rslt[i+1]=rslt[i]*(1+z[i])
     }
     rslt=reclass(rslt,x)
     return(rslt)
}

wfc.t1.bh=calc_bh(price$WFC,1000,"diff(log)")
jpm.t1.bh=calc_bh(price$JPM,1000,"diff(log)")
port.t1.bh=calc_bh(price$WFC,500,"diff(log)")+calc_bh(price$JPM,500,"diff(log)")

par(mfrow=c(1,1))
plot(port.t1.bh,main="buy and hold",ylim=range(rbind(port.t1.bh,jpm.t1.bh,wfc.t1.bh)))
lines(jpm.t1.bh,col="blue")
lines(wfc.t1.bh,col="darkgreen")
legend("topleft",legend=c("wfc","50/50","jpm"),lwd=2,cex=1,col=c("darkgreen","black","blue"))
```

calculate investment ratios

```
calc_ratios=function(ra,rb,rf){
     result=rbind(SharpeRatio(ra,rf,p=0.95,FUN="StdDev",weights=NULL,annualize=FALSE),
                  TreynorRatio(ra,rb,rf,scale=252,modified = FALSE),
                  SortinoRatio(ra,rf,weights=NULL),
                  CalmarRatio(ra,scale=252),
                  InformationRatio(ra,rb,scale=252))
     rownames(result)=c("sharpe","treynor","sortino","calmar","information")
     return(result)
}

calc_ratios(diff(log(price[,2:3]))[-1],diff(log(price[,1]))[-1],rf_t)
```

generate return graphics

```
gen_returns=function(price,all.plot,start.date,end.date){

     # format graphics
     dir.joint.pnl=(all.plot$p1pnl+all.plot$p2pnl)[-c(1,dim(all.plot)[1])]
     par(mfcol=c(3,1))
     plot((all.plot$p1states+all.plot$p2states),main="States (100=long) (-100=short)",type="l",col="blu
     plot((all.plot$p1cash+all.plot$p2cash)[-c(1,dim(all.plot)[1])],main="Cash",type="l",col="blue")

     a=length(price$WFC[paste0(start.date,"/",end.date)])
     bh=as.numeric(calc_bh(price$WFC[paste0(start.date,"/",end.date)][-c(1,a)],500,"diff(log)")+
                        calc_bh(price$JPM[paste0(start.date,"/",end.date)][-c(1,a)],500,"diff(log)"))
     plot(dir.joint.pnl,main="PNL",type="l",col="blue",ylim=range(cbind(bh,dir.joint.pnl)))
     lines(bh,col="red")
     legend("topleft",legend=c("buy-hold 50/50","portfolio"),lwd=2,cex=1,col=c("red","blue"))

     # format performance stats
     rtrn.5050=diff(log(price$WFC[paste0(start.date,"/",end.date)][-c(1:2,a)]))+
          diff(log(price$JPM[paste0(start.date,"/",end.date)][-c(1:2,a)]))
     rtrn.spy=diff(log(price$SPY[paste0(start.date,"/",end.date)][-c(1:2,a)]))
     rtrn.port=diff(log(dir.joint.pnl))
     rtrn.port=sapply(rtrn.port, function(x) replace(x, is.infinite(x),NA))
     rtrn.port=as.xts(rtrn.port,index(price[paste0(start.date,"/",end.date)][-c(1:2,a)]))
     merged=merge(rtrn.port,rtrn.5050,rtrn.spy)[-1];names(merged)=c("portfolio","buy-hold","spy")

     # pnl summary
     #print("pnl summary:")
     port.val=round(dir.joint.pnl[length(dir.joint.pnl)])
     port.v.bh=round(dir.joint.pnl[length(dir.joint.pnl)]-bh[length(bh)])
     port.ratios=calc_ratios(merged[,1],merged[,3],rf_t)
     bh.ratios=calc_ratios(merged[,2],merged[,3],rf_t)
     rslt=list(port.val=port.val,port.v.bh=port.v.bh,port.ratios=port.ratios,bh.ratios=bh.ratios)
     return(rslt)
}
```

set date

```
start.date=as.Date("2010-07-01")
end.date=as.Date("2015-02-28")
```

problem 1

a. moving avergae trading rules

```
gen_ma_w=function(m,l=NULL,w=NULL,h=NULL,rslt="pnl"){
     if(is.null(l)){l=seq(2,10,1)}
     if(is.null(w)){w=seq(0.95,1.1,0.02)}
     if(is.null(h)){h=seq(1,2,1)}
     amt=100

     for(i in 1:length(l)){
          for(j in 1:length(w)){
               for(k in 1:length(h)){
                    gl=ma_w(m[,1],l[i],w[j],h[k],amt,rslt)
                    rslt=cbind(l[i],w[j],h[k],gl)
                    if(exists("mstr")){mstr=rbind(mstr,rslt)}else{mstr=rslt}
               }
          }
     }
     return(mstr)
}

ma_w=function(ARseries,l,w,h,amount,rslt){
     rtrn=as.data.frame(coredata(ARseries))
     rtrn$states=0
     rtrn$PNL=0
     rtrn$cash=0
     rtrn$trades=0

    isAroundMean <- 1
    for (i in l:dim(rtrn)[1]){
         mu=mean(ARseries[(i-(l-1)):i])

         score=0
         if(ARseries[i]>(mu*w)) score=-1
         else if(ARseries[i]<((1/w)*mu)) score=1

        #3 conditions of whether to trade or not
        if (i > l){rtrn$states[i]=rtrn$states[i-1]}
        if (isAroundMean && (i==l||rtrn$states[i-1]==0)) {
            if (abs(score) == 1) rtrn$trades[i]=1
            if (score == 1) rtrn$states[i]=amount
            else if (score == -1) rtrn$states[i]=-amount
          }
        else if (rtrn$states[i-h]!=0){
              if(abs(score)!=1){
                   rtrn$trades[i]=1
                   rtrn$states[i]=0
              }
         }
        #update cash and PNL
        if (i==l) rtrn$cash[i] <- 500 - ARseries[i] * rtrn$states[i]
        else rtrn$cash[i] <- rtrn$cash[i-1] - ARseries[i] * (rtrn$states[i] - rtrn$states[i-1])
        rtrn$PNL[i] <- rtrn$cash[i] + rtrn$states[i] * ARseries[i]
        if (abs(score) <= 1) isAroundMean <- 1
        else isAroundMean <- 0
    }
    if(rslt=="plot"){
         return(rtrn)
         #list(ARseries = ARseries, trades = trades,
         #     states = states, cash = cash, PNL = PNL)
    }
    else{
         return(rtrn$PNL[i])
    }
}
```

rule 1: time / weights

```
gen_ma_sl=function(m,lng=NULL,shrt=NULL,rslt="pnl"){
     if(is.null(lng)){lng=seq(15,30,1)}
     if(is.null(shrt)){shrt=seq(2,10,1)}
     amt=100

     for(i in 1:length(lng)){
          for(j in 1:length(shrt)){
                    gl=ma_sl(m[,1],lng[i],shrt[j],amt,rslt)
                    rslt=cbind(lng[i],shrt[j],gl)
                    if(exists("mstr")){mstr=rbind(mstr,rslt)}else{mstr=rslt}
          }
     }
     return(mstr)
}

ma_sl=function(ARseries,lng,shrt,amount,rslt="pnl"){

     lngth=length(ARseries)
     relation=rep(0,lngth)
     trades=0
     states=rep(0,lngth)
     PNL=rep(0,lngth)
     cash=rep(0,lngth)

    isAroundMean <- 1
    for (i in 1:lngth){
         if(i<lng){lng.mu=mean(ARseries[1:i])}
         else{lng.mu=mean(ARseries[(i-(lng-1)):i])}

         if(i<shrt){shrt.mu=mean(ARseries[1:i])}
         else{shrt.mu=mean(ARseries[(i-(shrt-1)):i])}

         if(shrt.mu>=lng.mu){relation[i]=-1}else{relation[i]=1}
         score=0
         if(i>1){
              if(relation[i]+relation[i-1]==0){
                   if(relation[i]==-1){score=-1}
                   if(relation[i]==1){score=1}
               }
         }

        #3 conditions of whether to trade or not
        if (i > 1){states[i]=states[i-1]}
        if (isAroundMean && (i==1||states[i-1]==0)) {
            if (abs(score) == 1) trades=trades+1
            if (score == 1) states[i]=amount
            else if (score == -1) states[i]=-amount
          }
         else if (states[i-1]!=0){
              trades=trades+1
              states[i]=0
         }

        #update cash and PNL
        if (i==1) cash[i] <- 500 - ARseries[i] * states[i]
        else cash[i] <- cash[i-1] - ARseries[i] * (states[i] - states[i-1])
        PNL[i] <- cash[i] + states[i] * ARseries[i]
        if (abs(score) <= 1) isAroundMean <- 1
        else isAroundMean <- 0
    }
    if(rslt=="plot"){
         list(ARseries = ARseries, trades = trades,
              states = states, cash = cash, PNL = PNL)
    }
    else{
         return(PNL[i])
    }
}
```

rule 2: momentum

```
library(TTR)

gen_ma_bb=function(m){
     l=seq(15,30,1) # lookback
     h=seq(2,5,1) # holding period
     amt=100
     for(i in 1:length(l)){
          for(j in 1:length(h)){
               gl=ma_bb(m[,1],l[i],h[j],amt,"pnl")
               rslt=cbind(l[i],h[j],gl)
               if(exists("mstr")){mstr=rbind(mstr,rslt)}else{mstr=rslt}
          }
     }
     return(mstr)
}

ma_bb=function(ARseries,l,h,amount,rslt="pnl"){
     length=length(ARseries)
     #price=ARseries[l:length,]
    trades=0
    states=rep(0, length)
    PNL=rep(0, length)
    cash=rep(0, length)

    isAroundMean <- 1
    for (i in 1:length){
         if(i<l){p.bar=mean(ARseries[1:i])}
         else{p.bar=mean(ARseries[(i-(l-1)):i])}

         if(i<l){
              a=ARseries[1:i]
              b=(seq(i,1,-1))^-1
              p.hat=sum((a*b)/sum(b))
              sigma=1
          }else{
               a=ARseries[(i-(l-1)):i]
               b=(seq(l,1,-1))^-1
               p.hat=sum((a*b)/sum(b))
               sigma=(1/(l-1)*(sum((ARseries[(i-(l-1)):i]-p.bar)^2)))^(1/2)
          }
         band.upper=p.hat+2*sigma
         band.lower=p.hat-2*sigma

         score=0
         if(ARseries[i]>band.upper) score=-1
         else if(ARseries[i]<band.lower) score=1

        #3 conditions of whether to trade or not
        if (i > 1){states[i]=states[i-1]}
        if (isAroundMean && (i==1||states[i-1]==0)) {
            if (abs(score) == 1) trades=trades+1
            if (score == 1) states[i]=amount
            else if (score == -1) states[i]=-amount
          }
         else if (states[i-h]!=0){
              if(abs(score)!=1){
                   trades=trades+1
                   states[i]=0
              }
         }

        #update cash and PNL
        if (i==1) cash[i] <- 500 - ARseries[i] * states[i]
        else cash[i] <- cash[i-1] - ARseries[i] * (states[i] - states[i-1])
        PNL[i] <- cash[i] + states[i] * ARseries[i]
        if (abs(score) <= 1) isAroundMean <- 1
        else isAroundMean <- 0
    }
    if(rslt=="plot"){
         list(ARseries = ARseries, trades = trades, states = states, cash = cash, PNL = PNL)
    }
    else{
         return(PNL[i])
    }
}
```

rule 3: bollinger bands

1b oscillator rule

```
gen_osc=function(m){
     l=seq(3,12,1) # lookback
     h=seq(1,2,1) # holding period
     amt=100
     for(i in 1:length(l)){
          for(j in 1:length(h)){
               gl=osc(m[,1],l[i],h[j],amt,"pnl")
               rslt=cbind(l[i],h[j],gl)
               if(exists("mstr")){mstr=rbind(mstr,rslt)}else{mstr=rslt}
          }
     }
     return(mstr)
}

osc=function(ARseries,l,h,amount,rslt="pnl"){
     length=length(ARseries)
    trades=0
    states=rep(0, length)
    PNL=rep(0, length)
    cash=rep(0, length)

    isAroundMean <- 1
    for (i in 1:length){
         if(i<l){rsi=50}
         else{
              up=sum((as.numeric(ARseries[(i-(l-1)):i])-
                           as.numeric(ARseries[(i-l):(i-1)]))>0)/l

              dn=sum((as.numeric(ARseries[(i-(l-1)):i])-
                           as.numeric(ARseries[(i-l):(i-1)]))<0)/l
              rsi=100*(up/(up+dn))
          }

         score=0
         if(rsi>70){score=-1}
         else if(rsi<30){score=1}

        #3 conditions of whether to trade or not
        if (i > 1){states[i]=states[i-1]}
        if (isAroundMean && (i==1||states[i-1]==0)) {
            if (abs(score) == 1) trades=trades+1
            if (score == 1) states[i]=amount
            else if (score == -1) states[i]=-amount
          }
         else if (states[i-h]!=0){
              if(abs(score)!=1){
                   trades=trades+1
                   states[i]=0
              }
         }

        #update cash and PNL
        if (i==1) cash[i] <- 500 - ARseries[i] * states[i]
        else cash[i] <- cash[i-1] - ARseries[i] * (states[i] - states[i-1])
        PNL[i] <- cash[i] + states[i] * ARseries[i]
        if (abs(score) <= 1) isAroundMean <- 1
        else isAroundMean <- 0
    }
    if(rslt=="plot"){
         list(ARseries = ARseries, trades = trades, states = states, cash = cash, PNL = PNL)
    }
    else{
         return(PNL[i])
    }
}
```

problem 2

```
gen_direct=function(m){
     cst=c(0.001) # cost
     lag=seq(2,5,1) # lag
     thr=seq(1.8,2.1,0.1) # threshold
     h=seq(1,1,1) # holding period
     amt=100
     for(i in 1:length(lag)){
          for(j in 1:length(thr)){
               for(k in 1:length(h)){
                    gl=direct(m[,1],lag[i],thr[j],h[k],amt,"pnl",cst)
                    rslt=cbind(lag[i],thr[j],h[j],cst,gl)
                    if(exists("mstr")){mstr=rbind(mstr,rslt)}else{mstr=rslt}
               }
          }
     }
     return(mstr)
}

direct=function(x,l,thr,h,amt,rslt="pnl",cst=0){

    # format x and y
    x$diffs=as.numeric(diff(x));x=x[-1]
    x$costs=x[,1]*cst
    x$class=0;x$class[(x$diffs-x$costs)>0]=1
    lags=as.data.frame(matrix(0,dim(x)[1],l));lag.ids=NULL
    for(i in 1:l){
         lag.ids[i]=sprintf("lag%i",i)
         lags[(i+1):dim(lags)[1],i]=x[1:(dim(x)[1]-i),1]
    }
    names(lags)=lag.ids
    lags=as.xts(lags, index(x))
    x=cbind(x,lags);x=as.data.frame(coredata(x))

    trades=0;start=max(h,2,l+1);x=coredata(x)
    x$upProb=0;x$dnProb=0;x$pred=NA;x$score=0
    x$states=0;x$cash=0;x$PNL=0;x$trades=0
    for(i in 1:dim(x)[2]){x[,i]=as.numeric(x[,i])}

    for (i in start:(dim(x)[1]-1)){
         glm.fit=glm(as.factor(x$class[1:(i-1)])~.,
                     data=x[1:(i-1),5:(5+(l-1))],family="binomial")
         x$upProb[i]=predict(glm.fit,x[i,5:(5+(l-1))],type="response")

         if(i<23){incr=0.5}else{incr=thr*sd(x$upProb[(i-22):(i-1)])}
         if(x$upProb[i]>(0.5+incr)){x$score[i]=1;x$pred[i]=1}
         else if(x$upProb[i]<(0.5-incr)){x$score[i]=-1;x$pred[i]=0}

        #3 conditions of whether to trade or not
        if (i==start){x$states[i]=x$states[i-1]}
        else if (x$states[i-1]==0) {
            if (abs(x$score[i])==1){x$trades[i]=1}
            if (x$score[i]==1){x$states[i]=-amt}
            else if (x$score[i]==-1){x$states[i]=amt}
          }
        else if (x$states[i-1]!=0){
              x$trades[i]=1
              if(x$score[i]==1&&x$score[i-1]==1){x$states[i]=x$states[i-1]}
              else if(x$score[i]==-1&&x$score[i-1]==-1){x$states[i]=x$states[i-1]}
              else{x$states[i]=0}
         }
        #update cash and PNL
        if(i==start){x$cash[i]=500}
        else{x$cash[i]=x$cash[i-1]-((1-cst)*(x[i,1]*(x$states[i]-x$states[i-1])))}
        x$PNL[i]=x$cash[i]+x$states[i]*x[i,1]
    }
    if(rslt=="full"){
         return(x)
    }
    else{
         return(x$PNL[i])
    }
}
```

```r
par(mfrow=c(3,1))
plot(jpm.plot$upProb)
hist(jpm.plot$upProb[1:100],xlim=c(0,1))
abline(v=mean(jpm.plot$upProb[1:100]),col="red",lwd=3)
hist(jpm.plot$upProb[500:600],xlim=c(0,1))
abline(v=mean(jpm.plot$upProb[500:600]),col="red",lwd=3)
```

problem 3

```r
monnb=function(d){lt=as.POSIXlt(as.Date(d,origin="1900-01-01"));lt$year*12+lt$mon}
# compute a month difference as a difference between two monnb's
mondf=function(d1,d2){monnb(d2)-monnb(d1)}
mnths=mondf(start.date,end.date)+1

gen_distance=function(vrs,m,start.date,mnths,output){

     cst=c(0.001) # cost
     thr=seq(1.7,1.7,0.2) # threshold
     reg=seq(0.5,0.5,0.2) # allowance for regression
     lag=seq(3,3,1)

     amt=50

     for(r in 1:length(reg)){
          for(k in 1:length(thr)){
               for(l in 1:length(lag)){
                    inv=process_days(vrs,m,start.date,mnths,lag[l])
                    gl=distance(inv,thr[k],reg[r],amt,output,cst) #"pnl", "full"
                    rslt=cbind(vrs,thr[k],reg[r],lag[l],cst,gl)
                    if(exists("mstr")){mstr=rbind(mstr,rslt)}else{mstr=rslt}
               }
          }
     }
     return(mstr)
}

process_days=function(vrs,m,start.date,mnths,lag){

     for(i in 1:mnths){
          cur.date=start.date; month(cur.date)=month(cur.date)+(i-1)
          lb.date=cur.date;month(lb.date)=month(lb.date)-lag
          inv.date=cur.date;month(inv.date)=month(inv.date)+1
          lb=m[paste0(lb.date,"/",cur.date-1)]
          inv=m[paste0(cur.date,"/",inv.date-1)]

          inv=as.data.frame(coredata(inv))

          if(vrs=="a"){
               p.m.1.a=prod(1*(1+diff(log(lb[,1]))[-1]))
               p.m.2.a=prod(1*(1+diff(log(lb[,2]))[-1]))
               p.dist.a=sum((p.m.1.a-p.m.2.a)^2)/dim(lb)[1]
               p.sd.a=sqrt(1/(dim(lb)[1]-1)*sum(((p.m.1.a-p.m.2.a)^2-p.dist.a)^2))
               p.1=p.2=NULL
               for(j in 2:dim(lb)[1]){
                    p.1[j]=prod(1*(1+(diff(log(lb[((j-1):j),1]))[2]))
                    p.2[j]=prod(1*(1+(diff(log(lb[((j-1):j),2]))[2]))
               }
               p.1=p.1[-1];p.2=p.2[-1]

               inv$adjP1=inv$adjP2=0
              for(k in 2:dim(inv)[1]){
                    inv$adjP1[k]=prod(1*(1+(diff(log(inv[((i-1):k),1])))))
                    inv$adjP2[k]=prod(1*(1+(diff(log(inv[((i-1):k),2])))))
               }
               inv$p12Dist=(inv$adjP1-inv$adjP2)^2
               inv$p12Sd=inv$p12Dist/p.sd.a
               inv=inv[-1,]
          }else if (vrs=="b"){
               mns=colMeans(lb)
               tgt.p=sum(mns)/2
               cum.factor.1=mns[1]/tgt.p;cum.factor.2=mns[2]/tgt.p
               lb.parity=(mns[1]/cum.factor.1)/(mns[2]/cum.factor.2)
               lb$adjP1=lb[,1]/cum.factor.1;lb$adjP2=lb[,2]/cum.factor.2
               lb$curPrty=lb$adjP1/lb$adjP2
               lb$dev=lb$curPrty-lb.parity
               p.sd.b=sd(lb$dev)# this is sd of price diff

               # calc deviations
               inv$adjP1=inv[,1]/cum.factor.1;inv$adjP2=inv[,2]/cum.factor.2
               inv$curPrty=inv$adjP1/inv$adjP2
               inv$dev=inv$curPrty-lb.parity
               inv$p12sd=abs(inv$dev)/p.sd.b
          }
          inv$p1states=0;inv$p2states=0;inv$p1cash=0;inv$p2cash=0;
          inv$p1pnl=0;inv$p2pnl=0;inv$trades=0
          for(l in 1:dim(inv)[2]){inv[,l]=as.numeric(inv[,l])}
          start=2;max.days=20

          if(exists("mstr")){mstr=rbind(mstr,inv)}else{mstr=inv}
     }
     plot_series(cbind(mstr$adjP1,mstr$adjP2),0,0,"adjusted price",
                      mstr$dev,0,0,"parity deviation")
     return(mstr)
}

distance=function(inv,thr,reg,amt,rslt="pnl",cst=0){
     start=2;max.days=20

     for (i in start:(dim(inv)[1]-1)){
          if (i==start){inv$p1states[i]=inv$p1states[i-1]}
          else if(inv$p1states[i-1]==0){
               if(inv$p12sd[i]>thr){
                    inv$trades[i]=2
                    #if((inv$adjP1[i]-inv$adjP1[i-2])>(inv$adjP2[i]-inv$adjP2[i-2])){
                    if(inv$adjP1[i]>inv$adjP2[i]){
                         # if time, adjust these amounts by the price of the stock.
                         # but given relative performance, cut losses.
                         inv$p1states[i]=-amt
                         inv$p2states[i]=amt
                    } else{
                         inv$p1states[i]=amt
                         inv$p2states[i]=-amt
                    }
                    days.ctr=1
               }
          }
          else if(inv$p1states[i-1]!=0 && days.ctr<max.days){
               if(inv$p12sd[i]>(reg*thr)){
                    # note question stem says sell if they cross, presumably that's the threshold (vs p
                    inv$p1states[i]=inv$p1states[i-1]
                    inv$p2states[i]=inv$p2states[i-1]
                    days.ctr=days.ctr+1
               } else{
                    inv$trades[i]=2
                    inv$p1states[i]=0
                    inv$p2states[i]=0
                    days.ctr=0
               }
          }
          else if(days.ctr==max.days+1){
               inv$trades=2
               inv$p1states[i]=0
               inv$p2states[i]=0
               days.ctr=0
          }
          #update cash and PNL
         if(i==start){inv$p1cash[i]=inv$p2cash[i]=500}
         else{
               inv$p1cash[i]=inv$p1cash[i-1]-((1-cst)*(inv[i,1]*(inv$p1states[i]-inv$p1states[i-1])))
               inv$p2cash[i]=inv$p2cash[i-1]-((1-cst)*(inv[i,2]*(inv$p2states[i]-inv$p2states[i-1])))
          }
          inv$p1pnl[i]=inv$p1cash[i]+inv$p1states[i]*inv[i,1]
          inv$p2pnl[i]=inv$p2cash[i]+inv$p2states[i]*inv[i,2]
     }
    # return results
    if(rslt=="full"){
         return(inv)
    }
    else{
         return(sum(inv$p1pnl[i],inv$p2pnl[i]))
    }
}
```

a. distance

```
gen_coint=function(vrs,m,start.date,mnths,output){ #vrs: "joh","ols","avg"

     cst=c(0.001) # cost
     thr=seq(2,2,0.1) # threshold
     reg=seq(0.5,0.5,0.1) # allowance for regression
     lag=seq(2,2,1) # lookback period

     amt=50

     # need lookback window adjustment
     for(r in 1:length(reg)){
          for(k in 1:length(thr)){
               for(l in 1:length(lag)){
                    inv=calc_parity(vrs,m,start.date,mnths,lag[l])
                    gl=coint(inv,thr[k],reg[r],amt,output,cst) #"pnl" or "full"
                    rslt=cbind(vrs,thr[k],reg[r],lag[l],cst,gl)
                    if(exists("mstr")){mstr=rbind(mstr,rslt)}else{mstr=rslt}
               }
          }
     }
     return(mstr)
}
# if time, adjust lookback period. also, apply lookback logic to #2. just change interval to weeks
calc_parity=function(vrs,m,start.date,mnths,lag){
     for(i in 1:mnths){

          cur.date=update(start.date,day=1)+months(i-1)
          lb.date=update(cur.date,day=1)-months(lag)
          inv.date=update(cur.date,day=1)+months(1)
          lb=m[paste0(lb.date,"/",cur.date-1)]
          inv=m[paste0(cur.date,"/",inv.date-1)]

          inv=as.data.frame(coredata(inv))

          # calc lookback weight
          p2.coef.ols=abs(summary(lm(lb[,1]~lb[,2]))$coefficients[2,1]) # ols p2 coef.
          var.ord=try(VARorder(lb,output=F)$aicor,silent=TRUE) # vector autoregressive lag
          if(!inherits(var.ord,"try-error")){
               if(!is.na(var.ord) && var.ord>1){ #&& var.ord<6
                    #smry=ca.jo(lb,K=var.ord,ecdet=c("none"))
                    smry=ca.jo(lb)
                    p2.coef.joh=abs(cajorls(smry,r=1)$beta[2,1])
                    #p2.coef.joh=abs(summary(ca.jo(lb,K=var.ord,ecdet=c("none")))@V[2,1]) # Johansen
               } else{p2.coef.joh=p2.coef.ols}
          } else{p2.coef.joh=p2.coef.ols}
          p2.coef.avg=sum(p2.coef.joh,p2.coef.ols)/2
          lb$adjP1=log(lb[,1]);lb$adjP2=log(lb[,2])
          if(vrs=="joh"){lb$wt=lb$adjP1-(lb$adjP2*p2.coef.joh)}
          else if(vrs=="ols"){lb$wt=lb$adjP1-(lb$adjP2*p2.coef.ols)}
          else if(vrs=="avg"){lb$wt=lb$adjP1-(lb$adjP2*p2.coef.avg)}
          lb.wtMean=colMeans(lb$wt)
          lb.wtSd=sd(lb$wt)
          # calc current deviations
          inv$adjP1=log(inv[,1]);inv$adjP2=log(inv[,2])
          if(vrs=="joh"){inv$curWt=inv$adjP1-(inv$adjP2*p2.coef.joh);inv$coef=abs(p2.coef.joh)}
          else if(vrs=="ols"){inv$curWt=inv$adjP1-(inv$adjP2*p2.coef.ols);inv$coef=abs(p2.coef.ols)}
          else if(vrs=="avg"){inv$curWt=inv$adjP1-(inv$adjP2*p2.coef.avg);inv$coef=abs(p2.coef.avg)}
          inv$lbWt=lb.wtMean
          inv$dev=inv$curWt-lb.wtMean
          inv$p12sd=abs(inv$dev)/lb.wtSd

          inv$p1states=0;inv$p2states=0;inv$p1cash=0;inv$p2cash=0;
          inv$p1pnl=0;inv$p2pnl=0;inv$trades=0
          for(l in 1:dim(inv)[2]){inv[,l]=as.numeric(inv[,l])};start=2;max.days=20

          if(exists("mstr")){mstr=rbind(mstr,inv)}else{mstr=inv}
     }
     plot_series(cbind(mstr$adjP1,mstr$adjP2),0,0,"adjusted price",
                      mstr$dev,0,0,"weight deviation")
     return(mstr)
}

coint=function(inv,thr,reg,amt,rslt="pnl",cst=0){
     start=2;max.days=20

     for (i in start:(dim(inv)[1]-1)){
          if (i==start){inv$p1states[i]=inv$p1states[i-1]}
          else if(inv$p1states[i-1]==0){
               if(inv$p12sd[i]>thr){
                    inv$trades[i]=2
                    p1.amt=100*(1*amt)/((1+inv$coef[i])*amt)
                    p2.amt=100*(inv$coef[i]*amt)/((1+inv$coef[i])*amt)
                    #p1.amt=amt;p2.amt=amt
                    if((inv$adjP1[i])>(inv$adjP2[i])){
                         inv$p1states[i]=-p1.amt
                         inv$p2states[i]=p2.amt
                    } else{
                         inv$p1states[i]=p1.amt
                         inv$p2states[i]=-p2.amt
                    }
                    days.ctr=1
               }
          }
          else if(inv$p1states[i-1]!=0 && days.ctr<max.days){
               if(inv$p12sd[i]>(reg*thr)){
                    inv$p1states[i]=inv$p1states[i-1]
                    inv$p2states[i]=inv$p2states[i-1]
                    days.ctr=days.ctr+1
               } else{
                    inv$trades[i]=2
                    inv$p1states[i]=0
                    inv$p2states[i]=0
                    days.ctr=0
               }
          }
          else if(days.ctr==max.days+1){
               inv$trades=2
               inv$p1states[i]=0
               inv$p2states[i]=0
               days.ctr=0
          }
          #update cash and PNL
         if(i==start){inv$p1cash[i]=inv$p2cash[i]=500}
         else{
               inv$p1cash[i]=inv$p1cash[i-1]-((1-cst)*(inv[i,1]*(inv$p1states[i]-inv$p1states[i-1])))
               inv$p2cash[i]=inv$p2cash[i-1]-((1-cst)*(inv[i,2]*(inv$p2states[i]-inv$p2states[i-1])))
          }
          inv$p1pnl[i]=inv$p1cash[i]+inv$p1states[i]*inv[i,1]
          inv$p2pnl[i]=inv$p2cash[i]+inv$p2states[i]*inv[i,2]
     }
    # return results
    if(rslt=="full"){
         return(inv)
    }
    else{
         return(sum(inv$p1pnl[i],inv$p2pnl[i]))
    }
}
```

b. cointegration

problem 4

```
gen_trinity=function(vrs,m,start.date,mnths,output){ #vrs: "joh","ols","avg"

     cst=c(0.001) # cost
     thr=seq(1.9,1.9,0.1) # threshold
     reg=seq(0.55,0.55,0.1) # allowance for regression
     lag=seq(1,1,1) # lookback period

     amt=50

     # need lookback window adjustment
     for(r in 1:length(reg)){
          for(k in 1:length(thr)){
               for(l in 1:length(lag)){
                    inv=calc_trinity(vrs,m,start.date,mnths,lag[l])
                    gl=trinity(inv,thr[k],reg[r],amt,output,cst) #"pnl" or "full"
                    rslt=cbind(vrs,thr[k],reg[r],lag[l],cst,gl)
                    if(exists("mstr")){mstr=rbind(mstr,rslt)}else{mstr=rslt}
               }
          }
     }
     return(mstr)
}
```

```
calc_trinity=function(vrs,m,start.date,mnths,lag){
     for(i in 1:mnths){

          cur.date=update(start.date,day=1)+months(i-1)
          lb.date=update(cur.date,day=1)-months(lag)
          inv.date=update(cur.date,day=1)+months(1)
          lb=m[paste0(lb.date,"/",cur.date-1)]
          inv=m[paste0(cur.date,"/",inv.date-1)]

          inv=as.data.frame(coredata(inv))

          # calc lookback weight
          p2.coef.ols=summary(lm(lb[,1]~lb[,2]+lb[,3]))$coefficients[2,1] # abs?
          p3.coef.ols=summary(lm(lb[,1]~lb[,2]+lb[,3]))$coefficients[3,1] # abs?

          smry=ca.jo(lb)
          p2.coef.joh=cajorls(smry,r=1)$beta[2,1]
          p3.coef.joh=cajorls(smry,r=1)$beta[3,1]

          lb$adjP1=log(lb[,1]);lb$adjP2=log(lb[,2]);lb$adjP3=log(lb[,3])
          if(vrs=="joh"){lb$wt=lb$adjP1-(lb$adjP2*p2.coef.joh)-(lb$adjP3*p3.coef.joh)}
          else{lb$wt=lb$adjP1-(lb$adjP2*p2.coef.ols)-(lb$adjP3*p3.coef.ols)}

          lb.wtMean=colMeans(lb$wt)
          lb.wtSd=sd(lb$wt)

          # calc current deviations
          inv$adjP1=log(inv[,1]);inv$adjP2=log(inv[,2]);inv$adjP3=log(inv[,3])
          if(vrs=="joh"){
               inv$curWt=inv$adjP1-(inv$adjP2*p2.coef.joh)-(inv$adjP3*p3.coef.joh)
               inv$p2coef=abs(p2.coef.joh);inv$p3coef=abs(p3.coef.joh)
          }
          else{
               inv$curWt=inv$adjP1-(inv$adjP2*p2.coef.ols)-(inv$adjP3*p3.coef.ols)
               inv$p2coef=abs(p2.coef.ols);inv$p3coef=abs(p3.coef.ols)
          }
          inv$lbWt=lb.wtMean
          inv$dev=inv$curWt-lb.wtMean
          inv$p12sd=abs(inv$dev)/lb.wtSd

          inv$p1states=0;inv$p2states=0;inv$p3states=0;
          inv$p1cash=0;inv$p2cash=0;inv$p3cash=0;
          inv$p1pnl=0;inv$p2pnl=0;inv$p3pnl=0;inv$trades=0
          for(l in 1:dim(inv)[2]){inv[,l]=as.numeric(inv[,l])};start=2;max.days=20

          if(exists("mstr")){mstr=rbind(mstr,inv)}else{mstr=inv}
     }
     par(mfrow=c(2,1))
     plot(mstr$adjP1,col="darkgreen",ylim=c(3,5.5),type="l",main="adjusted prices")
     lines(mstr$adjP2,col="red")
     lines(mstr$adjP3,col="blue")
     legend("topleft",legend=c("spy","wfc","jpm"),lwd=2,cex=1,col=c("darkgreen","red","blue"))
     plot(mstr$dev,main="weight deviation",type="l")
     return(mstr)
}

trinity=function(inv,thr,reg,amt,rslt="pnl",cst=0){
     start=2;max.days=20

     for (i in start:(dim(inv)[1]-1)){
          if (i==start){inv$p1states[i]=inv$p1states[i-1]}
          else if(inv$p1states[i-1]==0){
               if(inv$p12sd[i]>thr){
                    inv$trades[i]=2
                    p1.amt=100*(1*amt)/((1+inv$p2coef[i]+inv$p3coef[i])*amt)
                    p2.amt=100*(inv$p2coef[i]*amt)/((1+inv$p2coef[i]+inv$p3coef[i])*amt)
                    p3.amt=100*(inv$p3coef[i]*amt)/((1+inv$p2coef[i]+inv$p3coef[i])*amt)
                    #p1.amt=amt;p2.amt=amt
                    if(inv$adjP1[i]>inv$adjP2[i]){
                         inv$p1states[i]=p1.amt
                         inv$p2states[i]=-p2.amt
                         inv$p3states[i]=-p3.amt
                    } else{
                         inv$p1states[i]=-p1.amt
                         inv$p2states[i]=p2.amt
                         inv$p2states[i]=p3.amt
                    }
                    days.ctr=1
               }
          }
          else if(inv$p1states[i-1]!=0 && days.ctr<max.days){
               if(inv$p12sd[i]>(reg*thr)){
                    inv$p1states[i]=inv$p1states[i-1]
                    inv$p2states[i]=inv$p2states[i-1]
                    inv$p3states[i]=inv$p3states[i-1]
                    days.ctr=days.ctr+1
               } else{
                    inv$trades[i]=2
                    inv$p1states[i]=0
                    inv$p2states[i]=0
                    inv$p3states[i]=0
                    days.ctr=0
               }
          }
          else if(days.ctr==max.days+1){
               inv$trades=3
               inv$p1states[i]=0
               inv$p2states[i]=0
               inv$p3states[i]=0
               days.ctr=0
          }
          #update cash and PNL
         if(i==start){inv$p1cash[i]=inv$p2cash[i]=inv$p2cash[i]=500}
         else{
               inv$p1cash[i]=inv$p1cash[i-1]-((1-cst)*(inv[i,1]*(inv$p1states[i]-inv$p1states[i-1])))
               inv$p2cash[i]=inv$p2cash[i-1]-((1-cst)*(inv[i,2]*(inv$p2states[i]-inv$p2states[i-1])))
               inv$p3cash[i]=inv$p3cash[i-1]-((1-cst)*(inv[i,3]*(inv$p3states[i]-inv$p3states[i-1])))
          }
          inv$p1pnl[i]=inv$p1cash[i]+inv$p1states[i]*inv[i,1]
          inv$p2pnl[i]=inv$p2cash[i]+inv$p2states[i]*inv[i,2]
          inv$p3pnl[i]=inv$p3cash[i]+inv$p3states[i]*inv[i,3]
     }
    # return results
    if(rslt=="full"){
         return(inv)
    }
    else{
         return(sum(inv$p1pnl[i],inv$p2pnl[i],inv$p3pnl[i]))
    }
}
```