**R code for Assignment 1**

d\_15stocks=read.csv("d\_15stocks.csv")

d\_indexes=read.csv("d\_indexes.csv")

Permno\_list=unique(d\_15stocks[,1])

Ticker\_list=unique(d\_15stocks[,3])

stock=array(0,c(3323,15))

for(i in 1:15)

{

stock[,i]=d\_15stocks[which(d\_15stocks[,1]==Permno\_list[i]),4]

}

colnames(stock)=as.vector(Ticker\_list)

vw=d\_indexes[,2]

ew=d\_indexes[,3]

#mean,sd and first order autocorrelation for 15 stocks for sample period

firstAR<-function(x)

{

n=length(x)

x1=x[1:(n-1)]

x2=x[2:n]

f.ar=cor(x1,x2)

return(f.ar)

}

mean.stock=apply(stock,2,mean)

sd.stock=apply(stock,2,sd)

f.ar.stock=apply(stock,2,firstAR)

mean.vw=mean(vw)

sd.vw=sd(vw)

f.ar.vw=firstAR(vw)

mean.ew=mean(ew)

sd.ew=sd(ew)

f.ar.ew=firstAR(ew)

#m=c(mean.stock,mean.vw,mean.ew)

# f=c(f.ar.stock,f.ar.vw,f.ar.ew)

# s=c(sd.stock,sd.vw,sd.ew)

#for subperiod

myfun<-function(x)

{

myfun=c(mean(x),sd(x),firstAR(x))

return(myfun)

}

s.t1=831

s.t2=1662

s.t3=2493

s.t4=3323

stock.sub1=apply(stock[1:s.t1,],2,myfun)

stock.sub2=apply(stock[(s.t1+1):s.t2,],2,myfun)

stock.sub3=apply(stock[(s.t2+1):s.t3,],2,myfun)

stock.sub4=apply(stock[(s.t3+1):s.t4,],2,myfun)

mean.stock.sub=cbind(stock.sub1[1,],stock.sub2[1,],stock.sub3[1,],stock.sub4[1,])

sd.stock.sub=cbind(stock.sub1[2,],stock.sub2[2,],stock.sub3[2,],stock.sub4[2,])

f.ar.stock.sub=cbind(stock.sub1[3,],stock.sub2[3,],stock.sub3[3,],stock.sub4[3,])

##plots

lends=as.vector(Ticker\_list)

matplot(t(mean.stock.sub),type="b",pch=20,xlab="Subperiods",ylab="Stock returns",xlim=c(1,5))

legend(locator(1),lends,lty=1:5,col=1:6,text.width=0.5)

matplot(t(sd.stock.sub),type="b",pch=20,xlab="Subperiods",ylab="Standard Deviance of stock returns",xlim=c(1,5))

legend(locator(1),lends,lty=1:5,col=1:6,text.width=0.5)

matplot(t(f.ar.stock.sub),type="b",pch=20,xlab="Subperiods",ylab="First\_order AR of stock returns",xlim=c(1,5))

legend(locator(1),lends,lty=1:5,col=1:6,text.width=0.5)

i.t1=832

i.t2=1664

i.t3=2496

i.t4=3329

ind=cbind(vw,ew)

ind.sub1=apply(ind[1:i.t1,],2,myfun)

ind.sub2=apply(ind[(i.t1+1):i.t2,],2,myfun)

ind.sub3=apply(ind[(i.t2+1):i.t3,],2,myfun)

ind.sub4=apply(ind[(i.t3+1):i.t4,],2,myfun)

mean.ind.sub=cbind(ind.sub1[1,],ind.sub2[1,],ind.sub3[1,],ind.sub4[1,])

sd.ind.sub=cbind(ind.sub1[2,],ind.sub2[2,],ind.sub3[2,],ind.sub4[2,])

f.ar.ind.sub=cbind(ind.sub1[3,],ind.sub2[3,],ind.sub3[3,],ind.sub4[3,])

lends=c("VWRETD","EWRETD")

matplot(t(mean.ind.sub),type="b",pch=20,xlab="Subperiods",ylab="Means of indexes returns",xlim=c(1,5))

legend(locator(1),lends,lty=1:5,col=1:6,text.width=0.5)

matplot(t(sd.ind.sub),type="b",pch=20,xlab="Subperiods",ylab="Standard Deviance of indexes returns",xlim=c(1,5))

legend(locator(1),lends,lty=1:5,col=1:6,text.width=0.5)

matplot(t(f.ar.ind.sub),type="b",pch=20,xlab="Subperiods",ylab="First\_order AR of indexes returns",xlim=c(1,5))

legend(locator(1),lends,lty=1:5,col=1:6,text.width=0.5)

#######

#1.b

#######

norm\_vw=rnorm(length(vw), mean(vw),sd(vw))

par(mfrow=c(1,2))

hist(vw,prob=T,nclass=30,xlab="VWRETD",main="")

lines(density(vw))

lines(density(norm\_vw),col="red")

hist(norm\_vw,prob=T,nclass=30,xlab="Normal",main="")

lines(density(norm\_vw))

norm\_ew=rnorm(length(ew), mean(ew),sd(ew))

par(mfrow=c(1,2))

hist(ew,prob=T,nclass=30,xlab="EWRETD",main="")

lines(density(ew))

lines(density(norm\_ew),col="red")

hist(norm\_ew,prob=T,nclass=30,xlab="Normal",main="")

lines(density(norm\_ew))

#######

#1.c

#######

CIfun<-function(x)

{

n=length(x)

xbar=mean(x,na.rm=T)

me=qt(0.995,df=n-1)\*sd(x,na.rm=T)/sqrt(n)

CI=c(xbar-me,xbar+me)

return(CI)

}

CI.stock=apply(stock,2,CIfun)

CI.index=apply(ind,2,CIfun)

#cbind(CI.stock,CI.index)

stock.sub1=apply(stock[1:s.t1,],2,CIfun)

stock.sub2=apply(stock[(s.t1+1):s.t2,],2,CIfun)

stock.sub3=apply(stock[(s.t2+1):s.t3,],2,CIfun)

stock.sub4=apply(stock[(s.t3+1):s.t4,],2,CIfun)

CI.stock.sub=rbind(stock.sub1,stock.sub2,stock.sub3,stock.sub4)

time=c(rep(c(1,2,3),each=831),rep(4,830))

stock\_t=cbind(stock,time)

lends=c("Subperiod 1","Subperiod 2","Subperiod 3","Subperiod 4")

error.bars.by(stock\_t[,1:15],time,alpha=0.01,bars=F,ylim=c(-0.008,0.008),main="",xlab="",ylab="Confidence Intervals",lty=1:5,col=1:6)

legend(locator(1),lends,lty=1:5,col=1:6)

a=cbind(c(vw[1:i.t1],NA),c(vw[(i.t1+1):i.t2],NA),c(vw[(i.t2+1):i.t3],NA),vw[(i.t3+1):i.t4])

error.bars(a,alpha=0.01,main="",xlab="VWRETD")

b=cbind(c(ew[1:i.t1],NA),c(ew[(i.t1+1):i.t2],NA),c(ew[(i.t2+1):i.t3],NA),ew[(i.t3+1):i.t4])

error.bars(b,alpha=0.01,main="",xlab="EWRETD")

#######

#1.(d)

#######

library(fBasics)

SR<-function(x)

{

sr=(max(x)-min(x))/sd(x)

return(sr)

}

myfun2<-function(x)

{

output=c(skewness(x),kurtosis(x), SR(x))

return(output)

}

stock.sub1=apply(stock[1:s.t1,],2,myfun2)

stock.sub2=apply(stock[(s.t1+1):s.t2,],2,myfun2)

stock.sub3=apply(stock[(s.t2+1):s.t3,],2,myfun2)

stock.sub4=apply(stock[(s.t3+1):s.t4,],2,myfun2)

sk.stock.sub=cbind(stock.sub1[1,],stock.sub2[1,],stock.sub3[1,],stock.sub4[1,])

ku.stock.sub=cbind(stock.sub1[2,],stock.sub2[2,],stock.sub3[2,],stock.sub4[2,])

str.stock.sub=cbind(stock.sub1[3,],stock.sub2[3,],stock.sub3[3,],stock.sub4[3,])

lends=as.vector(Ticker\_list)

matplot(t(sk.stock.sub),type="b",pch=20,xlab="Subperiods",ylab="Skewness of stock returns",xlim=c(1,5))

legend(locator(1),lends,lty=1:5,col=1:6,text.width=0.5)

matplot(t(ku.stock.sub),type="b",pch=20,xlab="Subperiods",ylab="Kurtosis of stock returns",xlim=c(1,5))

legend(locator(1),lends,lty=1:5,col=1:6,text.width=0.5)

matplot(t(str.stock.sub),type="b",pch=20,xlab="Subperiods",ylab="Studentized range of stock returns",xlim=c(1,5))

legend(locator(1),lends,lty=1:5,col=1:6,text.width=0.5)

#test for skweness (D'Agostino test) and test for kurtosis (Anscombe-Glynn test)

library(moments)

apply(stock,2,agostino.test)

apply(ind,2,agostino.test)

m\_15stocks=read.csv("m\_15stocks.csv")

m\_indexes=read.csv('m\_indexes.csv')

Permno\_list\_m=unique(m\_15stocks[,1])

Ticker\_list\_m=unique(m\_15stocks[,3])

stock\_m=array(0,c(2385,15))

for(i in 1:15)

{

stock\_m[,i]=m\_15stocks[which(m\_15stocks[,1]==Permno\_list\_m[i]),4]

}

colnames(stock\_m)=as.vector(Ticker\_list\_m)

apply(stock\_m,2,agostino.test)

apply(ind\_m,2,agostino.test)

##apply Jarque-Beta test

library(tseries)

apply(stock,2,jarque.bera.test)

apply(ind,2,jarque.bera.test)

##########

#2(a)

##########

#library(ares)

AAPL=read.csv("d\_aapl.csv")

P=rev(AAPL$Adj.Close)

n=length(P)

return=diff(P)/P[1:(n-1)]

log\_return=log(1+return)

Box.test(log\_return,lag=10,type="Ljung")

###########

#2(b)

###########

ar(log\_return,method="mle")

n=length(log\_return)

x1=c(0,log\_return[1:(n-1)])

x2=c(0,0,log\_return[1:(n-2)])

x3=c(rep(0,3),log\_return[1:(n-3)])

x4=c(rep(0,4),log\_return[1:(n-4)])

x5=c(rep(0,5),log\_return[1:(n-5)])

x6=c(rep(0,6),log\_return[1:(n-6)])

ts.lm=lm(log\_return~x1+x2+x3+x4+x5+x6)

summary(ts.lm)

#######

#2.(d)

#######

library(fUnitRoots)

urdfTest(log\_return,lag=6)

coeff=ts.lm$coefficients

ar.part=coeff[1]+coeff[5]\*x4+coeff[7]\*x6

resid=log\_return-coeff

Box.test(resid,lag=1,type="Ljung")

#####

#3.(b)

#####

library(fGarch)

garchFit(formula=~arma(6,0)+garch(1,1),data=log\_return,cond.dist="norm")

#####

#3.(c)

#####

garchFit(formula=~arma(6,0)+garch(1,1),data=log\_return,cond.dist="std")

#####

#3.(d)

#####

library(rugarch)

gspec=ugarchspec(variance.model = list(model = "iGARCH", garchOrder = c(1, 1),

submodel = NULL, external.regressors = NULL, variance.targeting = FALSE), mean.model = list(armaOrder = c(6, 0),include.mean = TRUE, archm = FALSE, archpow = 1, arfima = FALSE, external.regressors = NULL, archex = FALSE), distribution.model = "norm")

ugarchfit(gspec,log\_return)