Final Project

Wusi Fan

2/21/2020

# 1) set up  
## set working directory  
getwd()  
setwd("/Users/wusifan/Documents/Harrisburg/Courses/ANLY 515 Risk Modeling and Assessment/Project")  
  
## Upload all of the following packages:   
library(FRAPO)  
library(timeSeries)  
library(QRM)  
library(fGarch)  
library(copula)  
library(ghyp)  
library(fBasics)  
  
# 2) read data  
  
## 8 target stocks  
AdjClose <- read\_csv("AdjClose.csv")  
head(AdjClose)  
summary(AdjClose) ### confirmed there is no NA values in the data  
dim(AdjClose) # 1258\*9  
  
## SP500 for benchmarking  
SP500 <- read\_csv("SP500.csv")  
SP500 <- SP500[,c(1,6)] ## only keep adjusted close price and date  
head(SP500)  
summary(SP500) ### confirmed there is no NA values in the data  
  
  
## for creating portfolio: split the data, use the first 1000 rows to create portfolio, and use the last 258 rows to test the performance of the portfolois  
AdjClose.train <- AdjClose[1:1000,]   
SP500.train <- SP500[1:1000,]   
  
AdjClose.test <- AdjClose[1000:1258,]   
SP500.test <- SP500[1000:1258,]   
  
  
  
# 3) create daily return series (using all data here)  
## create time series object  
date <- AdjClose$Dates  
AdjClosets <- timeSeries(AdjClose[,-1], charvec = date)  
head(AdjClosets)  
  
SP500ts <- timeSeries(SP500[,-1], charvec = date)  
head(SP500ts)  
  
## create Daily Return series  
Rets <- returnseries(AdjClosets,method = "discrete", trim = TRUE, percentage = T) ## return as percentage  
head(Rets)  
  
SP500Rets <- returnseries(SP500ts,method = "discrete", trim = TRUE, percentage = T) ## return as percentage  
head(SP500Rets)  
  
## daily return time series plot  
par(mfrow=c(2,2))  
seriesPlot(Rets)  
## comparing daily return distribution box plot  
par(mfrow=c(1,1))  
boxPlot(Rets)  
summary(Rets)  
  
## acf and pacf plots  
for(i in 1:length(names(Rets))){  
 print(i)  
 par(mfrow=c(2,2))  
 acf(Rets[,i], lag.max = 30, na.action = na.omit, main = names(Rets)[i])  
 pacf(Rets[,i], lag.max = 30, na.action = na.omit, main = names(Rets)[i])  
 acf(abs(Rets[,i]), lag.max = 30, na.action = na.omit, main = paste("abs",names(Rets)[i]))  
 pacf(abs(Rets[,i]), lag.max = 30, na.action = na.omit, main = paste("abs",names(Rets)[i]))  
}  
  
# 4) comparing independent stocks (average return and risk)  
## average return  
AvgRet <- sapply(Rets,mean)  
AvgRet  
barplot(AvgRet, main="Average Return",  
 xlab="Stocks")  
max(AvgRet) ## Expo  
min(AvgRet) ## HURN  
  
## risk  
  
## define a function to calcualte risk----------------------  
CalRisk <- function(RS,p){  
 ## find the dsitribution with minimal AIC to use  
 AIC <- stepAIC.ghyp(RS, control = list(maxit = 1000))  
 print(AIC$fit.table)  
   
 MinAICDist <- AIC$fit.table[AIC$fit.table["aic"]==min(AIC$fit.table["aic"])][1] ## type of distribution  
 Symm <- AIC$fit.table[AIC$fit.table["aic"]==min(AIC$fit.table["aic"])][2] ## symmatric or not  
 if(Symm == "FALSE"){  
 Sym = F  
 } else{  
 Sym = T  
 }  
  
 print(MinAICDist)  
 print(Symm)  
 print(Sym)  
  
 ## fit data using the best distribution  
 if (MinAICDist == "NIG"){  
 print("NIG")  
 Mfit <- fit.NIGuv(RS, symmetric = Sym, control = list(maxit = 1000), na.rm = TRUE)  
 } else if(MinAICDist == "hyp"){  
 print("hyp")  
 Mfit <- fit.hypuv(RS, symmetric = Sym, control = list(maxit = 1000), na.rm = TRUE)  
 } else{  
 Mfit <- fit.ghypuv(RS, symmetric = Sym, control = list(maxit = 1000), na.rm = TRUE)   
 }  
   
 VaR <- qghyp(p, Mfit)  
 ES <- ESghyp(p, Mfit)  
 comb <- c(VaR, ES)  
 print("Results (Var,ES)----------------------")  
   
 return(comb)  
}  
##----------------------------------------------------------  
  
## calculate the riks for all stocks, at 5% probability  
  
riskdf <- data.frame(matrix(ncol=3,nrow=0, dimnames=list(NULL, c("Stock", "VaR", "ES")))) ## initilize df to store result  
  
## loop thru 8 stocks and calculate their risks  
for(i in 1:length(names(Rets))){  
 risks<- CalRisk(Rets[,i],0.05)  
 riskdf[i,1] <- names(Rets)[i]  
 riskdf[i,2] <- risks[1]  
 riskdf[i,3] <- risks[2]  
}  
  
riskdf  
  
par(mfrow=c(1,1))  
## compare value at risk  
barplot(riskdf$VaR, main="VaR",  
 xlab="Stocks",names.arg=riskdf$Stock)  
## compare expected shortfall  
barplot(riskdf$ES, main="ES",  
 xlab="Stocks",names.arg=riskdf$Stock)  
  
  
  
## portfolio###############################  
# 5)   
  
### 5.1) create 2 pre-defined portfolios  
  
## create time series object for training dataset  
date <- AdjClose.train$Dates  
AdjClose.traints <- timeSeries(AdjClose.train[,-1], charvec = date)  
head(AdjClose.traints)  
  
SP500.traints <- timeSeries(SP500.train[,-1], charvec = date)  
head(SP500.traints)  
  
## create Daily Return series  
Rets.train <- returnseries(AdjClose.traints,method = "discrete", trim = TRUE, percentage = T) ## return as percentage  
head(Rets.train)  
  
SP500Rets.train <- returnseries(SP500.traints,method = "discrete", trim = TRUE, percentage = T) ## return as percentage  
head(SP500Rets.train)  
  
  
## split Rets.train to create two portfolios  
RetsP1 <- Rets.train[,c(1:4)] # P1  
RetsP2 <- Rets.train[,c(5:8)] # P2  
  
  
  
# 5.2) calculate weight of each asset in their portfolios  
## 5.2.1) method 1: global minimum variance portfolio;  
V<-cov(RetsP1, use="pairwise.complete.obs")  
ERC<-PGMV(V)  
ERC  
P1W1<-Weights(ERC)/100  
P1W1  
  
V<-cov(RetsP2, use="pairwise.complete.obs")  
ERC<-PGMV(V)  
ERC  
P2W1<-Weights(ERC)/100  
P2W1  
  
## 5.2.2) method 2: Using SP500 as market benchmark, create, calculate weight using the minimum tail-dependent method  
  
# Step 1. By using apply function compute the value of Tau for each currency.  
# Tau is a Kendall rank correlation coefficient, between   
# two measured quantities(one of a asset and one of Market Index).  
Tau <- apply(RetsP1, 2, function(x) cor(x, SP500Rets.train, method = "kendall"))  
Tau  
# Step 2. By using Kendal rank correlation coeffients "Tau", estimate the  
# value of Clayton (Archimedean family) copula parameter "Theta"  
ThetaC <- copClayton@iTau(Tau) # copula parameter Theta  
ThetaC  
  
# Step 3. Use Theta to extact lower tail dependence coefficients "Lambda".  
# Lambda represents the interdependence between each asset  
# and Market Index at the lower tail of the distributions  
LambdaL <- copClayton@lambdaL(ThetaC) # lower tail dependence coefficients   
LambdaL  
  
# Step 4. Create a variable "WTD" which represents inverse log-weighted   
# and scaled portfolio weights of each asset based on   
# low tail dependency selction criteria.  
WTD <- -1 \* log(LambdaL)  
P1W2 <- WTD / sum(WTD)   
P1W2  
  
## repeat for P2  
Tau <- apply(RetsP2, 2, function(x) cor(x, SP500Rets.train, method = "kendall"))  
ThetaC <- copClayton@iTau(Tau)  
LambdaL <- copClayton@lambdaL(ThetaC)  
WTD <- -1 \* log(LambdaL)  
P2W2 <- WTD / sum(WTD)   
P2W2  
  
P1Ws <- rbind(P1W1,P1W2)  
P2Ws <- rbind(P2W1,P2W2)  
  
# 6) create a portfolio using minimum tail-dependent method to pick stocks  
  
# Step 1. By using apply function compute the value of Tau for each currency.  
# Tau is a Kendall rank correlation coefficient, between   
# two measured quantities(one of a asset and one of Market Index).  
Tau3 <- apply(Rets.train, 2, function(x) cor(x, SP500Rets.train, method = "kendall"))  
Tau3  
# Step 2. By using Kendal rank correlation coeffients "Tau", estimate the  
# value of Clayton (Archimedean family) copula parameter "Theta"  
ThetaC3 <- copClayton@iTau(Tau3) # copula parameter Theta  
ThetaC3  
  
# Step 3. Use Theta to extact lower tail dependence coefficients "Lambda".  
# Lambda represents the interdependence between each asset  
# and Market Index at the lower tail of the distributions  
LambdaL3 <- copClayton@lambdaL(ThetaC3) # lower tail dependence coefficients   
LambdaL3  
  
# Step 4. Select assets with Lambdas below   
# the median value of Lambda, and save the results as "IdxTD"  
# Which currencies would you select?   
IdxTD3 <- LambdaL3 < median(LambdaL3)  
IdxTD3  
  
# 17) Create a variable "WTD" which represents inverse log-weighted   
# and scaled portfolio weights of each selected currency based on   
# low tale dependency selction criteria.  
WTD3 <- -1 \* log(LambdaL3[IdxTD3])  
P3W <- WTD3 / sum(WTD3)  
P3W  
  
P3 = Rets.train[, IdxTD3]  
  
## interesting, it's includes all mid-size consulting firms  
  
  
# 7) calculated portfolio risk with calcualted weights  
  
## ------------------------------------------------------------------------  
## Define function using GARCH-copula approach to calcualte portfolio risk  
CalPortRisk <- function(PortRS,weights,pc){  
 ## GARCH - calculated portfolio risk  
 # Estimate GARCH model  
 # Step 1  
 gfit<-lapply(PortRS,garchFit,formula=~arma(0,0)+garch(1,1), cond.dist="std",trace=FALSE)  
 gfit  
 ## get SDs for 4 assets  
 gprog<-unlist(lapply(gfit,function(x) predict(x,n.ahead = 1)[3]))  
 ## get degrees-of-freedom parameters (shapes)  
 gshape<-unlist(lapply(gfit, function(x) x@fit$coef[5]))  
 # take a look at all paramaters of the GARCH model  
 gcoef<-unlist(lapply(gfit, function(x) x@fit$coef))  
   
 # Step 2  
 ## residuals for all 4 assets  
 gresid<-as.matrix(data.frame(lapply(gfit,function(x) x@residuals / sqrt(x@h.t))))  
 head(gresid)  
 #QQ plots of the standardized residuals of all 4 assets  
 par(mfrow=c(2,2))  
 unlist(lapply(gfit, function(x) plot(x, which=13)))  
   
 #ACF of the squared residuals  
 #par(mfrow=c(1,1))  
 #unlist(lapply(gfit, function(x) plot(x, which=11)))  
   
   
 # Step 3  
 U <- sapply(1:4, function(y) pt(gresid[, y], df = gshape[y]))  
 head(U)  
 hist(U)  
   
 # Step 4  
 ##Kendall's rank correlations.   
 cop <- fit.tcopula(Udata = U, method = "Kendall")  
   
   
 # Step 5  
 # 100,000 random return simulated for each asset  
 rcop <- rcopula.t(100000, df = cop$nu, Sigma = cop$P)  
 head(rcop)  
 #hist(rcop[,1], breaks=100)  
   
   
 #Step 6  
 # Compute the quantiles for these Monte Carlo draws.  
 qcop <- sapply(1:4, function(x) qstd(rcop[, x], nu = gshape[x]))  
 head(qcop)  
 hist(qcop[,2], breaks = 100)  
   
 # creating a matix of 1 period ahead predictions of standard deviations  
 ht.mat <- matrix(gprog, nrow = 100000, ncol = ncol(loss), byrow = TRUE)  
 head(ht.mat)  
 pf <- qcop \* ht.mat  
 head(pf)  
   
   
 # Step 7  
 pfall <- (qcop \* ht.mat) %\*% weights ## matrix multiplization  
 head(pfall)  
 tail(pfall)  
 hist(pfall,breaks = 100)  
 # Step 8  
 ## Estimated short fall  
 pfall.es95 <- median(head(sort(pfall), 100000\*pc))  
 pfall.es95   
 ## Value at Risk  
 pfall.var95 <- max(head(sort(pfall), 100000\*pc))  
 pfall.var95  
   
 results <- c(pfall.var95,pfall.es95)  
 print("Results (Var, ES)-------------------")  
 return(results)  
  
}  
# ----------------------------------------------------------------------------  
  
## Calculate portfolio risk  
CalPortRisk(RetsP1,P1W1,0.05) # -1.445726 -1.858080  
CalPortRisk(RetsP1,P1W2,0.05) # -1.467183 -1.905599  
  
CalPortRisk(RetsP2,P2W1,0.05) # -1.811953 -2.412306  
CalPortRisk(RetsP2,P2W2,0.05) # -1.817273 -2.375933  
  
CalPortRisk(P3,P3W,0.05) #same as P2W2  
  
  
  
# 8) Out-of-Sample Performance - test the portfolio performance using test data  
  
## create time series object for testing dataset  
date <- AdjClose.test$Dates  
AdjClose.testts <- timeSeries(AdjClose.test[,-1], charvec = date)  
head(AdjClose.testts)  
  
SP500.testts <- timeSeries(SP500.test[,-1], charvec = date)  
head(SP500.testts)  
  
## create Daily Return series  
Rets.test <- returnseries(AdjClose.testts,method = "discrete", percentage = F) +1 ## return as decimal  
head(Rets.test)  
  
SP500Rets.test <- returnseries(SP500.testts,method = "discrete", percentage = F) +1 ## return as decimal  
SP500Rets.test[1] <- 100  
head(SP500Rets.test)  
  
## calcualte equity  
SP500Equity <- cumprod(SP500Rets.test)   
SP500Equity  
  
  
## split Rets.train to create two portfolios  
RetsP1.test <- Rets.test[,c(1:4)] # P1  
RetsP1.test[1, ] <- P1W1\*100 ## using Weight 1 since it yield lower risk  
head(RetsP1.test)  
  
## calcualte equity  
P1Equity <- rowSums(apply(RetsP1.test, 2, cumprod))  
P1Equity  
  
RetsP2.test <- Rets.test[,c(5:8)] # P2  
RetsP2.test[1, ] <- P2W1\*100 ## using Weight 1 since it yield lower risk  
head(RetsP2.test)  
  
## calcualte equity  
P2Equity <- rowSums(apply(RetsP2.test, 2, cumprod))  
P2Equity  
  
  
## P3  
RetsP3.test <- Rets.test[,c(5:8)] # P2  
RetsP3.test[1, ] <- P3W\*100 ## using Weight 1 since it yield lower risk  
head(RetsP3.test)  
## calcualte equity  
P3Equity <- rowSums(apply(RetsP3.test, 2, cumprod))  
P3Equity  
  
  
### compare  
y <- cbind(SP500Equity, P1Equity, P2Equity,P3Equity)  
summary(y)  
## TDEquity method yeilds the best average equity  
  
par(mfrow=c(1,1))  
  
# Create a time series plots of equity curves for the "Out-of-Sample Periods".  
plot(SP500Equity, type = "l", ylim = range(y), ylab = "Equity Index",  
 xlab = "Out-of-Sample Periods")  
lines(P1Equity, lty = 2)  
lines(P2Equity, lty = 3)  
lines(P3Equity, lty = 5)  
legend("topleft",  
 legend = c("SP500", "Large Consulting", "Median Consulting","Lower Tail Dep."),  
 lty = c(1,2,3,5))