Emprical Finance - US Tech VS US Fin

LI,Shaojin Evan

2019/12/21

```r
library("quantmod")
library("magrittr")
library("dplyr")
library("BatchGetSymbols")
library("rugarch")
library("MASS")
library("fitdistrplus")
library("PerformanceAnalytics")
library("xts")
library("matrixStats")
library("GAS")
library("GARPFRM")
library("car")
library("FinTS")
library("copula")
```

```r
setwd("/Users/lishaojin/Desktop/Empirical\ Group/Stock\ List")
IYG<-read.csv(file= "IYG.csv",header=TRUE, sep = ",")
IYW<-read.csv(file= "IYW.csv",header=TRUE,sep = ",")
IYG <-IYG %>% arrange(desc(IYG$Weight)) %>% slice(1:20)
IYW <-IYW %>% arrange(desc(IYW$Weight)) %>% slice(1:20)
IYG$Ticker
IYW$Ticker
```

```r
# fin stocks
start <- as.Date("2008-12-31")
end <- as.Date("2018-12-31")
IYG.tickers <- c("BAC", "JPM","WFC","GS", "BK", "USB", "MS", "FNMA","STT","PNC" )
IYG <- BatchGetSymbols(tickers = IYG.tickers , first.date = start,last.date =end, freq.data = "daily")

IYG_dailyprice <- reshape.wide(IYG$df.tickers)
# IYG daily adjusted price
fin_price<- IYG_dailyprice$price.adjusted
```

```r
# IYW stocks
start <- as.Date("2008-12-31")
end <- as.Date("2018-12-31")
IYW.tickers <- c("MSFT", "AAPL", "CSCO", "GOOGL","IBM", "HPQ", "ORCL", "TXN", "GLW", "AABA")
IYW <- BatchGetSymbols(tickers = IYW.tickers, first.date = start,last.date =end, freq.data = "daily" )

IYW_dailyprice <- reshape.wide(IYW$df.tickers)
# IYW daily adjusted price
tech_price<- IYW_dailyprice$price.adjusted
```

##Analyzing individual performance

```r
# take out adjust price return for fin&tech
fin_ret<-as.data.frame(na.omit(IYG_dailyprice$ret.adjusted.prices))
tech_ret<-as.data.frame(na.omit(IYW_dailyprice$ret.adjusted.prices))
```

```
fin_ret_individual <- xts(fin_ret[,-1], order.by=as.Date(fin_ret$ref.date))
tech_ret_individual <- xts(tech_ret[,-1], order.by=as.Date(tech_ret$ref.date))
plot(fin_ret_individual,ylim=c(-0.5,0.5),lwd=0.2,main="Return of Financial Service Underlying Stocks")
plot(tech_ret_individual,ylim=c(-0.5,0.5),lwd=0.2,main="Return of Technology Underlying Stockse")
```

##Analyzing portfolio performance

```
# Create a portfolio thatequal weighted
N = 10
eq_weights <-rep(1/N,N)
# compute portfolio return
fin_pf <- as.data.frame(rowSums(fin_ret[,-1]*eq_weights))
tech_pf <- as.data.frame(rowSums(tech_ret[,-1]*eq_weights))
colnames(fin_pf) <- "fin_pf_ret"
colnames(tech_pf) <- "tech_pf_ret"
fin_pf_ret = xts(fin_pf, order.by=as.Date(fin_ret$ref.date))
tech_pf_ret = xts(tech_pf, order.by=as.Date(tech_ret$ref.date))

# Time series plot
plot(fin_pf_ret,ylim=c(-0.25,0.25),lwd=0.5,main="Return of US Financial Services Portfolio")
plot(tech_pf_ret,ylim=c(-0.25,0.25),lwd=0.5,main="Return of US Technology Portfolio")
```

```
# compute portfolio volatility
Vol<-function(portfolio_return) {
  gspec.ru<-ugarchspec(mean.model=list(armaOrder=c(0,0)),
                       variance.model=list(model="sGARCH",garchOrder=c(1,1)),
                       distribution.model="norm")
  gfit.ru<-ugarchfit(gspec.ru,portfolio_return)
  return (sqrt(252)*gfit.ru@fit$sigma)
}
fin_pf_vol<-as.data.frame(Vol(fin_pf_ret))
tech_pf_vol<-as.data.frame(Vol(tech_pf_ret))
colnames(fin_pf_vol) <- "fin_pf_vol"
colnames(tech_pf_vol) <- "tech_pf_vol"
fin_pf_vol = xts(fin_pf_vol, order.by=as.Date(fin_ret$ref.date))
tech_pf_vol = xts(tech_pf_vol, order.by=as.Date(tech_ret$ref.date))
# Time series plot
plot(fin_pf_vol,ylim=c(-0.5,1.5),lwd=0.5,main="Volatility of US Financial Services Portfolio")
plot(tech_pf_vol,ylim=c(-0.5,1.5),lwd=0.5,main="Volatility of US Technology Portfolio")
```

Fin&Tech Portfolio Comparison

```
# Using US treasury 30 day bill as risk free rate
getSymbols("DGS3MO",src = "FRED",from ="2008-12-31", to ="2018-12-31",periodicity="daily")
rf <-DGS3MO[7045:9650]

tech_SR <- (tech_pf_ret- rf)/(tech_pf_vol)
fin_SR <- (fin_pf_ret- rf)/(fin_pf_vol)
colnames(tech_SR) <- "tech_SR"
colnames(fin_SR) <- "fin_SR"

cor(tech_pf_ret,fin_pf_ret)
cor(tech_pf_vol,fin_pf_vol)
```

```
table.tech <- cbind.data.frame(tech_pf_ret,tech_pf_vol,tech_SR)
table.fin <- cbind.data.frame(fin_pf_ret,fin_pf_vol,fin_SR)
head(table.tech)
```

Comparison with different criteria

```
#count for Fin - ret over 3%
finret1 <- slice(table.fin[(!table.fin$fin_pf_ret<0.03)|(!table.fin$fin_pf_ret>-0.03),]) %>%tally(name=

#count for Fin - ret over 5%
finret2 <- slice(table.fin[(!table.fin$fin_pf_ret<0.05)|(!table.fin$fin_pf_ret>-0.05),]) %>%tally(name=


#count for Tech - ret over 3%
techret1 <- slice(table.tech[(!table.tech$tech_pf_ret<0.03)|(!table.tech$tech_pf_ret>-0.03),]) %>%  tal

#count for Tech - ret over 5%
techret2 <- slice(table.tech[(!table.tech$tech_pf_ret<0.05)|(!table.tech$tech_pf_ret>-0.05),]) %>%  tal

comp_ret1 <- merge(finret1,techret1,all=TRUE)
comp_ret2 <- merge(finret2,techret2,all=TRUE)
print(c(comp_ret1,comp_ret2))

#count for Fin - ret lower than -3%
finnret1 <- slice(table.fin[(!table.fin$fin_pf_ret>-0.03)|(!table.fin$fin_pf_ret>-0.03),]) %>% tally(na

#count for Fin - ret lower than -5%
finnret2 <- slice(table.fin[(!table.fin$fin_pf_ret>-0.05)|(!table.fin$fin_pf_ret>-0.05),]) %>% tally(na

#count for Tech - ret lower than -3%
technret1 <- slice(table.tech[(!table.tech$tech_pf_ret>-0.03)|(!table.tech$tech_pf_ret>-0.03),]) %>% ta

#count for Tech - ret lower than -5%
technret2 <- slice(table.tech[(!table.tech$tech_pf_ret>-0.05)|(!table.tech$tech_pf_ret>-0.05),]) %>% ta

comp_nret1 <- merge(finnret1,technret1,all=TRUE)
comp_nret2 <- merge(finnret2,technret2,all=TRUE)

#count for Fin - vol over 30%
finvol1 <-slice(table.fin[!table.fin$fin_pf_vol<30,]) %>% tally(name="Financial Service")

#count for Fin - vol over 50%
finvol2 <-slice(table.fin[!table.fin$fin_pf_vol<50,]) %>% tally(name="Financial Service")

#count for Tech - vol over 30%
techvol1 <- slice(table.tech[!table.tech$tech_pf_vol<30,]) %>%  tally(name="Technology")

#count for Tech - vol over 50%
techvol2 <- slice(table.tech[!table.tech$tech_pf_vol<50,]) %>%  tally(name="Technology")

comp_vol1 <- merge(finvol1,techvol1,all=TRUE)
comp_vol2 <- merge(finvol2,techvol2,all=TRUE)
```

```r
#count for Fin - SR over 0.3
finSR <- slice(table.fin[!table.fin$fin_SR<0.3,]) %>% tally(name="Financial Service")


#count for Tech - SR over 0.3
techSR <- slice(table.tech[!table.tech$tech_SR<0.03,]) %>% tally(name="Technology")

comp_SR <- merge(finSR,techSR,all=TRUE)
```

Portfolio Return descriptive analysis

```r
## descriptive analysis of Fin & Tech

table.Stats(cbind(fin_pf_ret,tech_pf_ret))

## Hidtogram with density & normal fit
chart.Histogram(fin_pf_ret, methods = c("add.density", "add.normal"),main = "US Finance Service Portfoli
chart.Histogram(tech_pf_ret, methods = c("add.density", "add.normal"),main = "US Technology Portfolio Re
```

##use maximum likelihood method to estimate the three parameters (location, scale, and degrees of freedom) of the t distribution.

```r
# density function
dt_G<-function(x,loc,sc,df){
  dt((x-loc)/sc,df)/sc
}
#distribution function
pt_G<-function(q, mean, sd, nu){
  pt((q-mean)/sd,nu)
}
#quantile function
qt_G<-function(p, mean, sd, nu){
  qt(p,nu)*sd+mean
}

# use fitdist get the results for location, scale, and degree of freedom

fin_fit<-fitdist(as.vector(fin_pf_ret), dt_G, start=list(loc=0, sc=1, df=3))
tech_fit<-fitdist(as.vector(tech_pf_ret), dt_G, start=list(loc=0, sc=1, df=3))

plot(fin_fit,breaks=100)
plot(tech_fit,breaks=100)
```

##Beta for the Fin & Tech

```r
# get S&P 500 as Market Performance
getSymbols("^GSPC", src = "yahoo", from = start, to = end,periodicity="daily")
SP500<-GSPC$GSPC.Adjusted
SP_ret<-na.omit(diff(log(SP500)))

# regression between fin & SP500
lm_fin_SP <-lm(fin_pf_ret ~ SP_ret)
lm_fin_SP.coefs<-coef((summary(lm_fin_SP)))

# regression between tech & SP500
```

```
lm_tech_SP <-lm(tech_pf_ret ~ SP_ret)
lm_tech_SP.coefs<-coef((summary(lm_tech_SP)))

# correlation matrix
correlation.mat = matrix(0, 2, 2)
rownames(correlation.mat) = c("Beta", "Rho")
colnames(correlation.mat) = c("Fin Portfolio","Tech Portfolio")
correlation.mat["Beta", ] = c(lm_fin_SP.coefs[2],lm_tech_SP.coefs[2])
correlation.mat["Rho", ] = c(cor(fin_pf_ret,SP_ret),cor(tech_pf_ret,SP_ret))


data.frame(correlation.mat)
```

annualized performance

```
# using PerformanceAnalytics
Performance <- function(x) {
  cumRetx = Return.cumulative(x)
    annRetx = Return.annualized(x, scale=252)
    sharpex = SharpeRatio.annualized(x, scale=252)
    winpctx = length(x[x > 0])/length(x[x != 0])
    annSDx = sd.annualized(x, scale=252)

    DDs <- findDrawdowns(x)
    maxDDx = min(DDs$return)
    maxLx = max(DDs$length)

    Perf = c(cumRetx, annRetx, sharpex, winpctx, annSDx, maxDDx, maxLx)
    names(Perf) = c("Cumulative Return", "Annual Return","Annualized Sharpe Ratio",
        "Win %", "Annualized Volatility", "Maximum Drawdown", "Max Length Drawdown")
    return(Perf)
}
data.frame(cbind(fin=Performance(fin_pf_ret),tech=Performance(tech_pf_ret)))

charts.PerformanceSummary(cbind(fin_pf_ret,tech_pf_ret),main="Performance Comparison",colorset = c("bla
```

##Downside risk measures

```
# downside risk for fin & tech
downsiderisk.mat = matrix(0, 3, 2)
rownames(downsiderisk.mat) = c("SemiDeviation", "VaR", "ES")
colnames(downsiderisk.mat) = c("Fin Portfolio","Tech Portfolio")
downsiderisk.mat["SemiDeviation", ] = SemiDeviation(cbind(fin_pf_ret,tech_pf_ret))
downsiderisk.mat["VaR", ] = VaR(cbind(fin_pf_ret,tech_pf_ret), p = 0.05)
downsiderisk.mat["ES", ]= PerformanceAnalytics::ES(cbind(fin_pf_ret,tech_pf_ret), p = 0.05)

data.frame(downsiderisk.mat)
```

##VaR test manually

```
# __Historical Approach__
# using formula
alpha <- 0.05
HA_VaR_fin_pf <- sort(as.vector(fin_pf_ret))[floor(length(as.vector(fin_pf_ret))*alpha)]
HA_VaR_tech_pf <- sort(as.vector(tech_pf_ret))[floor(length(as.vector(tech_pf_ret))*alpha)]
```

```
HA_VaR_fin_pf
HA_VaR_tech_pf

# __Variance-Covariance Method__
# get mean of fin_pf_ret&tech_pf_ret
mu_fin_pf <- mean(fin_pf_ret)
mu_tech_pf <- mean(tech_pf_ret)
# get standard deviation of fin_pf_ret&tech_pf_ret
sigma_fin_pf <- sd(fin_pf_ret)
sigma_tech_pf <- sd(tech_pf_ret)
# using formula
VC_VaR_fin_pf <- mu_fin_pf+qnorm(0.05)*sigma_fin_pf
VC_VaR_tech_pf <- mu_tech_pf+qnorm(0.05)*sigma_tech_pf
#compute results
VC_VaR_fin_pf
VC_VaR_tech_pf

# Monte Carlo simulation of VaR under the variance-covariance method
x<-VC_VaR_fin_pf
mu<-mu_fin_pf
sigma<-sigma_fin_pf

n<-1000000
set.seed(1234)
R<-rnorm(n,mean=mu,sd=sigma)
res<-1*(R<x)
mean(res)
c(mean(res)-sd(res)/sqrt(n)*qnorm(0.05),mean(res)+sd(res)/sqrt(n)*qnorm(0.05))
```

rolling VaR

```
# set up estimation window and testing window
observations = nrow(fin_pf_ret)

# window size
WS = 250

# out of sample forecast period
WT = observations-WS

alpha = 0.95

# loop over testing sample, compute VaR
rollingVaR <- function(x, p = 0.95) {
  # normal VaR, HS and modified HS
  normal.VaR = as.numeric(VaR(x, p=p, method="gaussian"))
  historical.VaR = as.numeric(VaR(x, p=p, method="historical"))
  ans = c(normal.VaR, historical.VaR)
  names(ans) = c("Normal", "Historical")
  return(ans)
}

# rolling 1-step ahead estimates of VaR
```

```
VaR.results.fin = rollapply(fin_pf_ret, width=WS,FUN = rollingVaR, by.column = FALSE,align = "right")

VaR.results.tech = rollapply(tech_pf_ret, width=WS,FUN = rollingVaR,by.column = FALSE,align = "right")

VaR.results.fin =lag.xts(VaR.results.fin, k=-1)
VaR.results.tech =lag.xts(VaR.results.tech, k=-1)

VaRFin<-rep(VC_VaR_fin_pf,nrow(VaR.results.fin))
VaRFin<-as.xts(VaRFin,order.by = index(VaR.results.fin))
chart.TimeSeries(merge(VaR.results.fin,VaRFin),legend.loc="topright",ylim = c(-0.05,0),main = "VaR.Fin"

VaRTech<-rep(VC_VaR_tech_pf,nrow(VaR.results.tech))
VaRTech<-as.xts(VaRTech,order.by = index(VaR.results.tech))
chart.TimeSeries(merge(VaR.results.tech,VaRTech),legend.loc="topright",ylim = c(-0.05,0),main="VaR.Tech
```

Backtecting VaR for fin

```
##record hit rates
violations.mat.fin = matrix(0, 2, 5)
rownames(violations.mat.fin) = c("Normal", "Historical")
colnames(violations.mat.fin) = c("Benchmark", "Vio", "1-alpha", "Percent", "VR")
violations.mat.fin[, "Benchmark"] = (1-alpha)*WT
violations.mat.fin[, "1-alpha"] = 1 - alpha

VaR.results.fin <- na.omit(VaR.results.fin)
# Show Normal VaR violations
normalVaR.violations.fin = fin_pf_ret[index(VaR.results.fin), ] < VaR.results.fin[, "Normal"]

violation.dates.fin = index(normalVaR.violations.fin[which(normalVaR.violations.fin)])

# plot violations of fin
plot(fin_pf_ret[index(VaR.results.fin),], col="black", ylab="Return",lwd=0.3,main="Fin Violation Test")
lines(merge(VaR.results.fin[, "Normal"],VaRFin), col=c("blue","green"), lwd=1)
lines(fin_pf_ret[violation.dates.fin,], type="p", pch="+", col="red", lwd=1.5)

for(i in colnames(VaR.results.fin)) {
  VaR.violations.fin = fin_pf_ret[index(VaR.results.fin), ] < VaR.results.fin[, i]
  violations.mat.fin[i, "Vio"] = sum(VaR.violations.fin)
  violations.mat.fin[i, "Percent"] = sum(VaR.violations.fin)/WT
  violations.mat.fin[i, "VR"] = violations.mat.fin[i, "Vio"]/violations.mat.fin[i, "Benchmark"]
}

data.frame(violations.mat.fin)
```

Backtecting VaR for tech

```
##record hit rates
violations.mat.tech = matrix(0, 2, 5)
rownames(violations.mat.tech) = c("Normal", "Historical")
colnames(violations.mat.tech) = c("Benchmark", "Vio", "1-alpha", "Percent", "VR")
violations.mat.tech[, "Benchmark"] = (1-alpha)*WT
violations.mat.tech[, "1-alpha"] = 1 - alpha
```

```
# Show Normal VaR violations
normalVaR.violations.tech = tech_pf_ret[index(VaR.results.tech), ] <VaR.results.tech[, "Normal"]

violation.dates.tech = index(normalVaR.violations.tech[which(normalVaR.violations.tech)])


# plot violations of tech
plot(tech_pf_ret[index(VaR.results.tech),], col="black", ylab="Return",lwd=0.3,main="Tech Violation Tes
lines(merge(VaR.results.tech[, "Normal"],VaRTech), col=c("blue","green"), lwd=1)
lines(tech_pf_ret[violation.dates.tech,],type="p", pch="+", col="red", lwd=1.5)

VaR.results.tech <- na.omit(VaR.results.tech)
for(i in colnames(VaR.results.tech)) {
  VaR.violations.tech = tech_pf_ret[index(VaR.results.tech), ] < VaR.results.tech[, i]
  violations.mat.tech[i, "Vio"] = sum(VaR.violations.tech)
  violations.mat.tech[i, "Percent"] = sum(VaR.violations.tech)/WT
  violations.mat.tech[i, "VR"] = violations.mat.tech[i, "Vio"]/violations.mat.tech[i, "Benchmark"]
}

data.frame(violations.mat.tech)
```

##VaR test for fin

```
VaR.test = VaRTest(1-alpha, actual=coredata(fin_pf_ret[index(VaR.results.fin),]),
                   VaR=coredata(VaR.results.fin[,"Normal"]))
names(VaR.test)
# LR test for correct number of exceedances
VaR.test[1:7]

# LR tests for independence of exceedances
VaR.test[8:12]

# backtest VaR but re-fit every 5 observations
VaR.results.fin.5 = rollapply(fin_pf_ret, width=WS, by = 5,FUN = rollingVaR, by.column = FALSE,align =

chart.TimeSeries(merge(fin_pf_ret, VaR.results.fin, fill=na.locf), legend.loc="topright",lwd = 0.5,main

# expand series to match fin a

chart.TimeSeries(merge(fin_pf_ret, VaR.results.fin.1,fill=na.locf), legend.loc="topright",lwd = 0.5,mai
```

Backtesting VaR for fin refit by by every 5 obeservations

```
##record hit rates
violations.mat.fin.5 = matrix(0, 2, 5)
rownames(violations.mat.fin.5) = c("Normal", "Historical")
colnames(violations.mat.fin.5) = c("Benchmark", "Vio", "1-alpha", "Percent", "VR")
violations.mat.fin.5[, "Benchmark"] = (1-alpha)*WT
violations.mat.fin.5[, "1-alpha"] = 1 - alpha

VaR.results.fin.5 <- na.omit(VaR.results.fin.5)
for(i in colnames(VaR.results.fin.5)) {
  VaR.violations.fin.5 = tech_pf_ret[index(VaR.results.fin.5), ] < VaR.results.fin.5[, i]
  violations.mat.fin.5[i, "Vio"] = sum(VaR.violations.fin.5)
  violations.mat.fin.5[i, "Percent"] = sum(VaR.violations.fin.5)/WT
  violations.mat.fin.5[i, "VR"] = violations.mat.fin.5[i, "Vio"]/violations.mat.fin.5[i, "Benchmark"]
}

data.frame(violations.mat.fin.5)
```

##VaR test for tech

```
VaR.test = VaRTest(1-alpha, actual=coredata(tech_pf_ret[index(VaR.results.tech),]),
                   VaR=coredata(VaR.results.tech[,"Normal"]))
names(VaR.test)
# LR test for correct number of exceedances
VaR.test[1:7]

# LR tests for independence of exceedances
VaR.test[8:12]

# backtest VaR refit by every 5 observations
VaR.results.tech.5 = rollapply(tech_pf_ret, width=WS,by =5, FUN = rollingVaR, by.column = FALSE,align =

chart.TimeSeries(merge(tech_pf_ret, VaR.results.tech), legend.loc="topright",lwd = 0.5,main="Tech VaR")

# expand series to match tech and use trick to fill NA values with last values carried forward
chart.TimeSeries(merge(tech_pf_ret, VaR.results.tech.1), legend.loc="topright",lwd = 0.5,main="Tech VaR
```

Backtesting VaR for tech refit by every 5 observations

```
##record hit rates
violations.mat.tech.5 = matrix(0, 2, 5)
rownames(violations.mat.tech.5) = c("Normal", "Historical")
colnames(violations.mat.tech.5) = c("Benchmark", "Vio", "1-alpha", "Percent", "VR")
violations.mat.tech.5[, "Benchmark"] = (1-alpha)*WT
violations.mat.tech.5[, "1-alpha"] = 1 - alpha

VaR.results.tech.5 <- na.omit(VaR.results.tech.5)
for(i in colnames(VaR.results.tech.5)) {
  VaR.violations.tech.5 = tech_pf_ret[index(VaR.results.tech.5), ] < VaR.results.tech.5[, i]
  violations.mat.tech.5[i, "Vio"] = sum(VaR.violations.tech.5)
  violations.mat.tech.5[i, "Percent"] = sum(VaR.violations.tech.5)/WT
  violations.mat.tech.5[i, "VR"] = violations.mat.tech.5[i, "Vio"]/violations.mat.tech.5[i, "Benchmark"]
}

data.frame(violations.mat.tech.5)
```

##rolling GARCH(1,1) with VaR violations for Fin

```
spec = ugarchspec(distribution.model = "std")

Fin.roll = ugarchroll(spec, fin_pf_ret, n.ahead=1,
                      forecast.length = WT,
                      refit.every=5,
                      refit.window = c("recursive", "moving"),
                      window.size = WS,
                      calculate.VaR=TRUE)
plot(Fin.roll,which=4)
plot(Fin.roll,which=5)
report(Fin.roll, type="VaR",VaR.alpha=0.05)
```

##rolling GARCH(1,1) with VaR violations for Tech

```
Tech.roll = ugarchroll(spec, tech_pf_ret, n.ahead=1,
                       forecast.length = WT,
                       refit.every=5,
                       refit.window = c("recursive", "moving"),
                       window.size = WS,
                       calculate.VaR=TRUE)

plot(Tech.roll,which=4)
plot(Tech.roll,which=5)
report(Tech.roll, type="VaR",VaR.alpha=0.05)
```