Financial Statistics
Homework 5

Chenghua Liu
liuch18@mails.tsinghua.edu.cn
Department of Computer Science
Tsinghua University

1 7.1 (2011.1-2021.1, without DeLL)

What is the gross exposure of the portfolio with weights

$$\boldsymbol{w} = (-0.2, 0.3, 0.4, -0.2, 0.1, 0.2, 0, 0.4)?$$

What is the risk of this portfolio invested on "Dell", "Ford", "GE", "IBM", "Johnson & Johnson", "Merck", "3-month Treasury Bill", "S&P 500 index" in the past ten years (January 1, 2005 to January 1,2015 , using daily data). Compare it with the portfolio with equal weight.

Solution:

Without considering dell, multiply other weights by a constant so that the sum of their weights is 1. So we get

$$w = (0.25000000, 0.33333333, -0.16666667, 0.08333333, 0.16666667, 0.000000000.33333333)$$

The gross exposure is $\|w\|_1 = 1.333333$

```
library(tidyquant)
library(zoo)
library(xts)
library(tidyverse)
options("getSymbols.warning4.0"=FALSE)
options("getSymbols.yahoo.warning"=FALSE)
tickers = c( "F", "GE", "IBM","JNJ","MRK", "^GSPC")
stock = tq_get(tickers, from ="2011-01-01", to="2021-01-01",
get = "stock.prices",periodicity = 'daily')
IRX = tq_get("^IRX",from = "2011-01-01", to="2021-01-01",
periodicity = 'daily')
IRX = na.omit(IRX)
F_daily_returns <- subset(stock,stock$symbol=="F") %>%tq_transmute(
select = adjusted,mutate_fun = periodReturn,
period = "daily",col_rename = "returns")
```

```
GE_daily_returns <- subset(stock,stock$symbol=="GE") %>%tq_transmute(
select = adjusted,mutate_fun = periodReturn,
period = "daily",col_rename = "returns")
IBM_daily_returns <- subset(stock,stock$symbol=="IBM") %>%tq_transmute(
select = adjusted,mutate_fun = periodReturn,
period = "daily",col_rename = "returns")
JNJ_daily_returns <- subset(stock,stock$symbol=="JNJ") %>%tq_transmute(
select = adjusted,mutate_fun = periodReturn,
period = "daily",col_rename = "returns")
MRK_daily_returns <- subset(stock,stock$symbol=="MRK") %>%tq_transmute(
select = adjusted,mutate_fun = periodReturn,
period = "daily",col_rename = "returns")
SP500_daily_returns <- subset(stock,stock$symbol=="^GSPC") %>%tq_transmute(
select = adjusted,mutate_fun = periodReturn,
period = "daily",col_rename = "returns")
TB_daily_returns<-drop_na(data.frame(date=IRX$date,returns=(1+IRX$adjusted*0.01)**(1/63)-1))
list_of_dataframes <- list(F_daily_returns,
GE_daily_returns,IBM_daily_returns,
JNJ_daily_returns,MRK_daily_returns,
SP500_daily_returns,TB_daily_returns)
for(i in list_of_dataframes){i = data.frame(i)}
returns = list_of_dataframes %>% reduce(inner_join, by = "date")
names(returns) = c("date",tickers,"TB")
returns = data.frame(returns)
returns = xts(returns[,2:8],order.by=as.Date(returns[,1]))
head(returns)
cov17 = cov(returns[,1:7])
w1 =  c(0.3,0.4,-0.2,0.1,0.2,0,0.4)*5/6
risk1 = t(as.matrix(w1))%*%cov17%*%as.matrix(w1)
w2 = as.matrix(rep(1/7,7))
risk2 = t(as.matrix(w2))%*%cov17%*%as.matrix(w2)
print(c(risk1,risk2))
#################################
[1] 1.079917e-04 9.459194e-05
```

So the risk of this portfolio is 1.079917×10^{-4} and the risk of equal weight is9.459194×10^{-5}.

■

2 7.2

Let $X_1, \cdots, X_T$ be a sequence of stationary time series with the autocovariance function $\gamma(h) = \operatorname{cov}(X_t, X_{t+h})$ and $\bar{X} = T^{-1}\sum_{t=1}^{T} X_t$. Show that

$$\operatorname{var}(\bar{X}) = T^{-2}[T\gamma(0) + 2(T-1)\gamma(1) + \cdots + 2\gamma(T-1)]$$

and

$$\lim_{T\to\infty} [T \operatorname{var}(\bar{X})] = \gamma(0) + 2\sum_{h=1}^{\infty} \gamma(h)$$

In other words,

$$\operatorname{var}(\bar{X}) \approx T^{-1}\left[\gamma(0) + 2\sum_{h=1}^{L}\gamma(h)\right]$$

for a sufficient large integer L.

Solution:

$$\begin{aligned}
\operatorname{var}(\bar{X}) &= \frac{\operatorname{var}\left(X_1 + X_1 + \ldots + X_T\right)}{T^2} \\
&= \frac{\sum_{t=1}^{T}\operatorname{var}\left(X_t\right) + 2\sum_{t=1}^{T-1}\operatorname{cov}\left(X_t, X_{t+1}\right) + 2\sum_{t=1}^{T-2}\operatorname{cov}\left(X_t, X_{t+2}\right) + \ldots + 2\operatorname{cov}\left(X_1, X_T\right)}{T^2} \\
&= T^{-2}[T\gamma(0) + 2(T-1)\gamma(1) + \ldots + 2\gamma(T-1)]
\end{aligned}$$

and

$$\lim_{T\to\infty}\left[T\operatorname{Var}(\bar{X})\right] = \lim_{T\to\infty}\left[\gamma(0) + \frac{2(T-1)}{T}\gamma(1) + \ldots + \frac{2}{T}\gamma(T-1)\right] = \gamma(0) + 2\sum_{h=1}^{\infty}\gamma(h)$$

■

3 7.11

Suppose that we have 100 investable stocks, labeled as 1 through 100 and classified as ”Consumer Non-durables”, ”Consumer durables”, ”Manufacturing”, ”Energy”, ”Business equipment”, ”Telecommunications”, ”Shops”, ”Health”, ”Utilities”, and ”Others”. Let $w_1, \cdots, w_{100}$ be the portfolio weights. If the first 10 stocks are labeled as ”Consumer Non-durables”, the second 10 stocks are in ”Consumer durables” and so on, write down the constraints of the portfolios:

(a) the ”health stocks” are no more than 15% and ”energy stocks” are no more than 30%

(b) no exposure to ”Telecommunications”;

(c) exposure to ”Consumer durables”, but next exposure to ”Consumer durables” is zero.

Solution:

(a)

$$\begin{gathered}
\textstyle\sum_{i=71}^{80} w_i \leqslant 15\% \\
\textstyle\sum_{i=31}^{40} w_i \leqslant 30\%
\end{gathered}$$

(b)

$$\sum_{i=51}^{60} |w_i| = 0$$

(c)

$$\sum_{i=11}^{20} |w_i| > 0$$
$$\sum_{i=1}^{20} w_i = 0$$

■

4 7.12((2001.1 to 2021.1, without DELL)

Let the study period be January 2001 to January 2015. Apply the sample covariance matrix, the Fama-French 3 factor model, and the RiskMetrics with $\lambda = 0.94$ to obtain the time-varying covariance matrix for 'Dell", "Ford", "GE", "IBM", "Intel", "Johnson & Johnson", "Merck", "3-month Treasury Bill" and "S&P 500 index" at the beginning of each month (defined as every 21 days after the initial 252 days). Optimize the portfolio and holds for the next 21 days. Compute the risk of such a portfolio and compare it with the equally weighted portfolio.

Solution:

Covariance matrix

$$\boldsymbol{R}_t = \mathbf{a} + \boldsymbol{B}\boldsymbol{f}_t + \boldsymbol{\varepsilon}_t$$
$$\operatorname{var}(\boldsymbol{R}) = \boldsymbol{B}\operatorname{var}(\boldsymbol{f})\boldsymbol{B}^T + \operatorname{var}(\varepsilon)$$

Fama-French

$$r = R_f + \beta_3 (R_m - R_f) + b_s \cdot SMB + b_v \cdot HML + \alpha$$

Risk Metrics

$$\hat{\Sigma}_t = \lambda \hat{\Sigma}_{t-1} + (1-\lambda) \boldsymbol{R}_t \boldsymbol{R}_t^{\mathrm{T}}$$

Implementation

```
setwd("/Users/liuchenghua/Downloads/data")
library(foreach)
assets = c("FORD","GE","IBM","JNJ","MRK","^IRX","SP500")
FORD = read.csv("FORD.csv",stringsAsFactors = F)
IRX = read.csv("^IRX.csv",stringsAsFactors = F)
IRX$Open = as.numeric(IRX$Open)
IRX$Close = as.numeric(IRX$Close)
#intersection of available days
billdates = IRX[complete.cases(IRX),1]
transact.day = intersect(FORD$Date,billdates)
transact.day = as.Date(transact.day)
```

```r
transact.days = transact.day[transact.day >= as.Date("2011-01-01")]

#calculate daily returns
cal.return<-function(name){
        print(name)
        data = read.csv(paste(name,".csv",sep = ""),header = T,stringsAsFactors = F)
        if(name == "SP500"){
                data$Date = as.Date(data$Date,"%m/%d/%y")
                data = data[order(data$Date),]
        }else{
                data$Date = as.Date(data$Date)
        }
        data = data[data$Date %in% transact.days,]
        data$Close = as.numeric(data$Close)
        return = diff(data$Close)/data$Close[-length(data$Close)]
        # return = diff(log(data$Close)) you may also use log return
        return(return)
}

assets = c("FORD","GE","IBM","INTC","JNJ","MRK","^IRX","SP500")
assets.return = foreach(i=1:8,.combine = 'cbind')%do%cal.return(assets[i],transact.day)

# The problem ask you to optimize the weights. You may refer to Chapter 6 May11 P24
library(quadprog)
alc.opt = function(risk, ret.ge = 5e-4,gross.le = 5) {
        n = ncol(risk$S)
        Dmat = risk$S
        dvec = rep(1/n, n)
        Amat = t(rbind(rep(1, n),
        risk$mu,
        expand.grid(rep(list(c(-1, 1)), n))))
        bvec = c(1, ret.ge, rep(-gross.le, 2^n))

        solve.QP(Dmat, dvec,Amat,bvec, meq = 1)$solution
}

# Three methods to estimate the sample covariance matrix

# Sample covariance
risk.cov = function(t1,t2, D, ...) {

        R = D[t1:t2,]
        list(mu = colMeans(R), S = cov(R))
}

FF = read.table('F-F_Research_Data_Factors_daily.txt',
header = T, row.names = 'Date',
check.names = F)
FF$Date = rownames(FF)
FF$Date  = as.Date(FF$Date,"%Y%m%d")
FF = dplyr::filter(FF, Date %in% transact.day)
```

```
# Fama-French
risk.FF = function(t1, t2, D, FF,...) {
        Y = D[t1:t2,] - FF[t1:t2, 'RF']
        y0 = mean(FF[t1:t2, 'RF'])
        X = as.matrix(FF[t1:t2, c('Mkt-RF', 'SMB', 'HML')])
        x = as.data.frame(t(colMeans(X)))
        mu = c()
        E = list()
        fm = lm(Y ~ X)
        B = coef(fm)[-1,]
        Sigmaf = cov(X)
        Sigmae = cov(fm$residuals)
        S = t(B)%*%Sigmaf%*%B+Sigmae
        mu = apply(fm$fitted.values,2, mean) + y0
        list(mu = mu, S = S)
}

# Risk Metric/ Exponential Smoothing
risk.RM = function(t1,t2, D, risk0, lambda = 0.94) {
        risk1 = risk.cov(t1, t2, D)
        list(mu = lambda * risk1$mu +
        (1 - lambda) * risk0$mu,
        S = lambda * risk1$S + (1 - lambda)*risk0$S)
}

# To compare the risk of each method,
# we provide three types of covariance matrix estimators, so that we will derive
# more concrete conclusions.

risk.eval <-function(t1, t2, D, w){
        risk = rep(0,3)
        SamCov =  risk.cov(i, i+20, assets.return)
        risk[1] = sqrt(t(w)%*%SamCov$S%*%w)
        SamCov =  risk.RM(i, i+20, assets.return,
        list(mu = rep(0,8),
        S =  matrix(0, nrow = 8, ncol = 8)))
        risk[2] = sqrt(t(w)%*%SamCov$S%*%w)
        SamCov =  risk.FF(i, i+20, assets.return,FF)
        risk[3] = sqrt(t(w)%*%SamCov$S%*%w)
        return(risk)
}
```

Equally weighted

```
trial.days = seq(1, 5010-42, by=21)
trial.n = length(trial.days)
Risk0 = matrix(0, trial.n,3)
Ret0 = rep(0, trial.n)
for (j in 1:length(trial.days)){
        # sample covariance
        i = trial.days[j]
        # estimation of covariance matrix
```

```
        # optimize the porforlio
        w_cov = rep(1, 8)/8
        # performance in the next 21 days
        Risk0[j,] = risk.eval(i, i+20, assets.return,w_cov)          # risk
        Ret0[j] = prod(assets.return[(i+21):(i+40),] %*% w_cov + 1)−1 # return
}
matplot(Risk0,type=”l”,lty=1)
legend(”topleft”,col=1:3, lty=1,legend = c(”Sample”,”Exponential”,”FF”))
```

Sample Covariance Matrix

```
Risk1 = matrix(0, trial.n,3)
Ret1 = rep(0, trial.n)

trial.days = seq(1, 5010−42,by=21)
for (j in 1:length(trial.days)){
        #sample covariance
        i = trial.days[j]
        #estimation of covariance matrix
        SamCov1 = risk.cov(i,i+20, assets.return)
        #optimize the porforlio
        w_cov = alc.opt(SamCov1)
        #performance in the next 21 days
        Risk1[j,] = risk.eval(i, i+20, assets.return,w_cov)          # risk
        Ret1[j] = prod(assets.return[(i+21):(i+40),] %*% w_cov + 1)−1 # return
}
matplot(Risk1,type=”l”)
legend(”topleft”,col=1:3, lty=1,legend = c(”Sample”,”Exponential”,”FF”))
```

Risk Metric

```
Risk2 = matrix(0, trial.n,3)
Ret2 = rep(0, trial.n)
trial.days = seq(1, 5010−42,by=21)
SamCov1 = list(mu = rep(0,8),
S = matrix(0, nrow = 8, ncol = 8))
for (j in 1:length(trial.days)){
        #sample covariance
        i = trial.days[j]
        #estimation of covariance matrix
        SamCov1 = risk.RM(i, i+20, assets.return,
        SamCov1)
        #optimize the porforlio
        w_cov = alc.opt(SamCov1)
        #performance in the next 21 days
        risk2 = sqrt(t(w_cov)%*%SamCov1$S%*%w_cov)
        Risk2[j,] = risk.eval(i, i+20, assets.return,w_cov)           # risk
        Ret2[j] = prod(assets.return[(i+21):(i+40),] %*% w_cov + 1)−1  # return
}
matplot(Risk2,type=”l”)
```

Fama French three-factor model

```r
Risk3 = matrix(0, trial.n,3)
Ret3 = rep(0, trial.n)

for (j in 1:length(trial.days)){
        #sample covariance
        i = trial.days[j]
        #estimation of covariance matrix
        SamCov1 = risk.FF(i, i+20, assets.return, FF)
        #optimize the porforlio
        w_cov = alc.opt(SamCov1)
        #performance in the next 21 days
        Risk3[j,] = risk.eval(i, i+20, assets.return,w_cov)          # risk
        Ret3[j] = prod(assets.return[(i+21):(i+40),] %*% w_cov + 1) - 1 # return
}
matplot(Risk3,type="l")
```

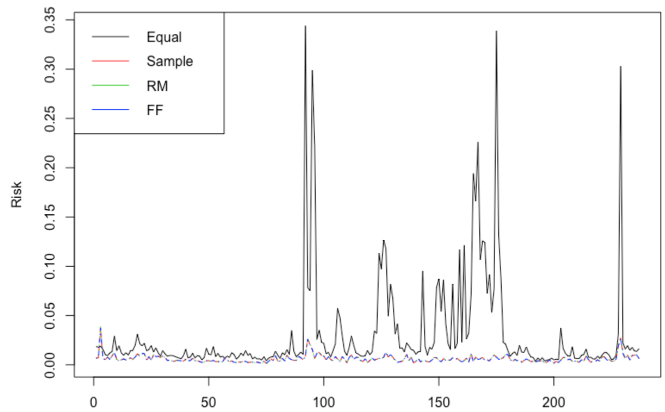
Comparison

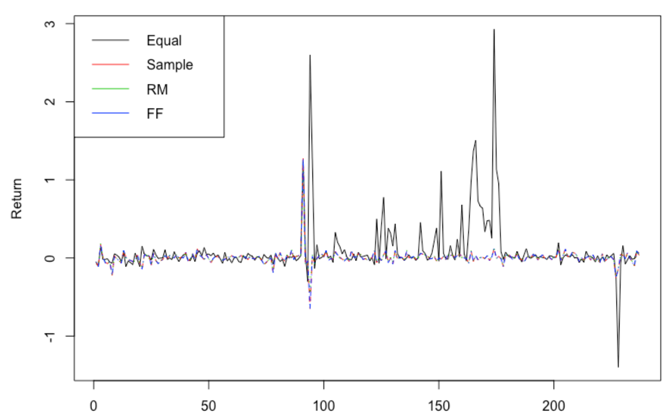
```r
Risk = cbind(Risk0[,1],  Risk1[,1], Risk2[,1],Risk3[,1])
Return = cbind(Ret0, Ret1, Ret2, Ret3)
result = data.frame(Risk =apply(Risk,2 ,mean), Return =  apply(Return,2, mean))
row.names(result) = c("Equal","Sample","RM","FF")
knitr::kable(result)

matplot(Risk,type="l",ylab = "Risk")
legend("topleft",col=1:4, lty=1,legend = c("Equal","Sample","RM","FF"))
matplot(Return,type="l",ylab = "Return")
legend("topleft",col=1:4, lty=1,legend = c("Equal","Sample","RM","FF"))
```

The results,

	Risk	Return
Equal	0.0311830	0.1046140
Sample	0.0059755	0.0045591
RM	0.0060727	0.0058939
FF	0.0059860	0.0045164


Equal
Sample
RM
FF
Risk
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0
50
100
150
200


Equal
Sample
RM
FF
Return
-1
0
1
2
3
0
50
100
150
200

■