STATS 240 HW2
Nov 11, 2019

2.9. Let $X_1, \ldots, X_n$ be n observations for which the joint density function $f_{\boldsymbol{\theta}}(x_1, \ldots, x_n)$ depends on an unknown parameter vector $\boldsymbol{\theta}$. Assuming that f is a smooth function of $\boldsymbol{\theta}$, show that:
(a) $E\big(\boldsymbol{\nabla} \log f_{\boldsymbol{\theta}}(X_1, \ldots, X_n)\big) = \mathbf{0}$;
(b) $E\Big(-\boldsymbol{\nabla}^2 l(\boldsymbol{\theta})\Big) = \mathrm{Cov}\big(\boldsymbol{\nabla} \log f_{\boldsymbol{\theta}}(X_1, \ldots, X_n)\big)$, where $l(\boldsymbol{\theta})$ denotes the log-likelihood function.

3.5. The file `m_ret_10stocks.txt` contains the monthly returns of ten stocks from January 1994 to December 2006. The ten stocks include Apple Computer, Adobe Systems, Automatic Data Processing, Advanced Micro Devices, Dell, Gateway, Hewlett-Packard Company, International Business Machines Corp., Microsoft Corp., and Oracle Corp. Consider portfolios that consist of these ten stocks.
(a) Compute the sample mean $\widehat{\boldsymbol{\mu}}$ and the sample covariance matrix $\widehat{\boldsymbol{\Sigma}}$ of the log returns.
(b) Assume that the monthly target return is 0.3% and that short selling is allowed. Estimate the optimal portfolio weights by replacing $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ in Markowitz's theory by $\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}}$.
(c) Do the same as in (b) for Michaud's resampled weights (3.38) using $B = 500$ bootstrap samples.

3.6. The file `m_sp500ret_3mtcm.txt` contains three columns. The second column gives the monthly returns of the S&P 500 index from January 1994 to December 2006. The third column gives the monthly rates of the 3-month U. S. Treasury bill in the secondary market, which is obtained from the Federal Reserve Bank of St. Louis and used as the risk-free asset here. Consider the ten monthly returns in the file `m_ret_10stocks.txt`.
(b) Use the bootstrap procedure in Section 3.5 to estimate the standard errors of the point estimates of α, β, and the Sharpe and Treynor indices.
(c) Test for each stock the null hypothesis $\alpha = 0$.
(d) Use the regression model (3.24) to test for the ten stocks the null hypothesis $\boldsymbol{\alpha} = \mathbf{0}$.

$X_1 \dots X_n$

joint density function $f_\theta(x_1 \dots x_n)$

f is smooth, θ is unknown parameter

2.9 (a) WTS $\mathbb{E}(\underbrace{\nabla \log f_\theta(X_1 \dots X_n)}_{\text{use gradient of log likelihood function to access goodness of estimation } \theta,}) = \vec{0}$

$$\mathbb{E}[\nabla \log f_\theta(X_1, \dots, X_n)]$$
$$= \int \nabla \log f(x|\theta) f(x|\theta)\, dx$$
$$= \int \frac{\nabla f(x|\theta)}{f(x|\theta)} f(x|\theta)\, dx$$
$$= \int \nabla f(x|\theta)\, dx$$
$$= \nabla \int f(x|\theta)\, dx$$
$$= \nabla 1$$
$$= \vec{0} \quad \blacksquare$$

2.9 (b) WTS $\mathbb{E}(-\nabla^2 l(\theta)) = \operatorname{cov}(\nabla \log f_\theta(X_1, \dots, X_n))$

where $l(\theta)$ is log-likelihood

$$\mathbb{E}(-\nabla^2 l(\theta)) = \mathbb{E}(-\nabla^2 \log f_\theta(X_1 \dots X_n)) = \mathbb{E}\left(-\nabla\left(\frac{\nabla f_\theta(X_1 \dots X_n)}{f_\theta(X_1 \dots X_n)}\right)\right)$$
$$= \mathbb{E}\left(\frac{-f_\theta(X_1 \dots X_n)\nabla^2 f_\theta(X_1 \dots X_n) + \nabla f_\theta(X_1 \dots X_n)(\nabla f_\theta(X_1 \dots X_n))^T}{f_\theta^2(X_1 \dots X_n)}\right)$$
$$= \mathbb{E}\left(\frac{-\nabla^2 f_\theta(X_1 \dots X_n)}{f_\theta(X_1 \dots X_n)}\right) + \mathbb{E}\left[(\nabla \log f_\theta(X_1 \dots X_n))(\nabla \log f_\theta(X_1 \dots X_n))^T\right]$$
$$= \nabla^2\left[\int f_\theta(X_1 \dots X_n)\, dX_1 \dots X_n\right] + \operatorname{cov}(\nabla \log f_\theta(X_1 \dots X_n))$$
$$= \underbrace{\nabla^2(1)}_{=0} + \operatorname{cov}(\nabla \log f_\theta(X_1 \dots X_n))$$
$$= \operatorname{cov}(\nabla \log f_\theta(X_1 \dots X_n)) \quad \blacksquare$$

Author: Chih-Hsuan (Carolyn) Kao; Email: chkao831@stanford.edu; Date: Nov 10th, 2019; Title: STATS 240 HW2 Autumn19-20;

Problem 3.5 part (a) Compute the sample mean mu and the sample covariance matrix of the log returns

```r
stock = read.table("m_logret_10stocks.txt", header = TRUE)
stock <- stock[2:length(stock)] #strip out date since unnecessary
stock <- stock[-c(157,158,159), ] #strip out blank last rows

sample_mu = colMeans(stock)
sample_mu
```

```
##         AAPL         ADBE          ADP          AMD         DELL
##  0.006894302  0.007686409  0.003922326  0.002308663  0.011893799
##          GTW           HP          IBM         MSFT         ORCL
## -0.002486445  0.005012223  0.005677223  0.007293060  0.006934118
```

```r
sample_cov = cov(stock)
sample_cov
```

```
##              AAPL         ADBE          ADP          AMD         DELL
## AAPL 0.0045492771 0.0013279090 0.0001289013 0.0026123284 0.0019279099
## ADBE 0.0013279090 0.0044543592 0.0005155629 0.0017548782 0.0011450512
## ADP  0.0001289013 0.0005155629 0.0007882798 0.0005010847 0.0003765201
## AMD  0.0026123284 0.0017548782 0.0005010847 0.0071438193 0.0019955112
## DELL 0.0019279099 0.0011450512 0.0003765201 0.0019955112 0.0035220840
## GTW  0.0022417112 0.0019385825 0.0005769367 0.0030814814 0.0025456103
## HP   0.0007299668 0.0006935191 0.0002949267 0.0009015779 0.0005459061
## IBM  0.0009500523 0.0004466988 0.0004151517 0.0015804227 0.0010751672
## MSFT 0.0008986926 0.0006104442 0.0003242627 0.0011862503 0.0016045961
## ORCL 0.0013409915 0.0011763324 0.0003809339 0.0015399549 0.0011940995
##               GTW           HP          IBM         MSFT         ORCL
## AAPL 0.0022417112 7.299668e-04 0.0009500523 8.986926e-04 0.0013409915
## ADBE 0.0019385825 6.935191e-04 0.0004466988 6.104442e-04 0.0011763324
## ADP  0.0005769367 2.949267e-04 0.0004151517 3.242627e-04 0.0003809339
## AMD  0.0030814814 9.015779e-04 0.0015804227 1.186250e-03 0.0015399549
## DELL 0.0025456103 5.459061e-04 0.0010751672 1.604596e-03 0.0011940995
## GTW  0.0065056598 4.673048e-04 0.0010237907 1.440775e-03 0.0011096197
## HP   0.0004673048 2.304971e-03 0.0004827528 8.859382e-05 0.0003080524
## IBM  0.0010237907 4.827528e-04 0.0014830329 9.009031e-04 0.0008346107
## MSFT 0.0014407745 8.859382e-05 0.0009009031 2.018664e-03 0.0009180812
## ORCL 0.0011096197 3.080524e-04 0.0008346107 9.180812e-04 0.0038527785
```

```r
one_vec = matrix(1, ncol(stock), 1)
```

Problem 3.5 part (b) Assume monthly target return is 0.3% and short selling is allowed (weight can be negative), estimate the optimal portfolio weights

Textbook p.70

$$W_{eff} = \arg\min_{w} w^T \Sigma w \quad s.t. \quad w^T\mu = 0.003, \; w^T\vec{1} = 1$$

$$w_{eff} = \{ B\Sigma^{-1}\vec{1} - A\Sigma^{-1}\mu + 0.003(C\Sigma^{-1}\mu - A\Sigma^{-1}\vec{1}) \} / D$$

where $A = \mu^T\Sigma^{-1}\vec{1}$

$B = \mu^T\Sigma^{-1}\mu$

$C = \vec{1}^T\Sigma^{-1}\vec{1}$

$D = BC - A^2$

```r
#reference: Textbook page 70 formula
library(MASS)

A = t(sample_mu)%*%ginv(sample_cov)%*%one_vec
B = t(sample_mu)%*%ginv(sample_cov)%*%sample_mu
C = t(one_vec)%*%ginv(sample_cov)%*%one_vec
D = B%*%C - A%*%A

a = A[1,1]
b = B[1,1]
c = C[1,1]
d = D[1,1]

eff_weight = ( b*ginv(sample_cov)%*%one_vec - a*ginv(sample_cov)%*%sample_mu + 0.003*(c*ginv(sample_cov)%*%samp
le_mu - a*ginv(sample_cov)%*%one_vec) )/d
eff_weight #this is the optimal portfolio weight
```

```
##              [,1]
##   [1,]  0.07136165
##   [2,] -0.02964316
##   [3,]  0.64982619
##   [4,] -0.02366900
##   [5,] -0.19107218
##   [6,]  0.07856725
##   [7,]  0.16168683
##   [8,]  0.09650606
##   [9,]  0.16306989
##  [10,]  0.02336648
```

Problem 3.5 part (c) Using B=500 bootstrap samples for resampled weights from part (b)

```r
library(MASS)

calculate_weight <- function(data,term){

  dat <- data[term,]
  one_vec <- matrix(1, ncol(dat), 1)
  sample_mu = colMeans(dat)
  sample_cov = cov(dat)
  A = t(sample_mu)%*%ginv(sample_cov)%*%one_vec
  B = t(sample_mu)%*%ginv(sample_cov)%*%sample_mu
  C = t(one_vec)%*%ginv(sample_cov)%*%one_vec
  D = B%*%C - A%*%A

  a = A[1,1]
  b = B[1,1]
  c = C[1,1]
  d = D[1,1]

  eff_weight = ( b*ginv(sample_cov)%*%one_vec - a*ginv(sample_cov)%*%sample_mu + 0.003*(c*ginv(sample_cov)%*%sa
mple_mu - a*ginv(sample_cov)%*%one_vec) )/d
  return(eff_weight)
}

library(boot)
#boot(object, f=function(object), R=Number of bootstrap samples)
boot_weight = boot(stock,calculate_weight,R=500)
#boot_weight
colMeans(boot_weight$t)
```

```
##  [1]  0.06835340 -0.01203646  0.61560219 -0.03578639 -0.12741643
##  [6]  0.03291295  0.15880286  0.09694007  0.18086460  0.02176322
```

Problem 3.6 part (b) Use bootstrap procedure to estimate standard errors of alpha, beta, Sharpe, and Treynor

```r
sp500ret = read.table("m_sp500ret_3mtcm.txt",
                   skip = 1,
                   sep="\t",
                   header = TRUE,
                   fill=FALSE,
                   strip.white=TRUE)
sp500ret$X3mTCM <- sp500ret$X3mTCM/1200 #scale the risk free rate
#compute the market risk premium (excess return of market)
sp500ret$risk_premium <- sp500ret$sp500 - sp500ret$X3mTCM

#this function returns vector contains alpha, beta, sharpe, treynor for each stock
calculate_ten <- function(data,term){

  dat <- data[term]

  Y = dat - sp500ret$X3mTCM #Y is stock premium
  CAPM = lm( Y~ sp500ret$risk_premium ) #stock premium~market premium
  alpha = CAPM$coefficients[1]
  beta = CAPM$coefficients[2]
  sharpe = mean(Y)/sd(Y)
  treynor = mean(Y)/beta

  vec = c(alpha,beta,sharpe,treynor)

  return(vec)
}

for (i in (1:10)){
  #boot(object, f=function(object), R=Number of bootstrap samples)
  boot_result = boot(stock[,i],calculate_ten,R=500)
  print(boot_result)
}
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = stock[, i], statistic = calculate_ten, R = 500)
##
## AAPL
## Bootstrap Statistics :
##        original        bias    std. error
## t1* 0.003847378 -5.805583e-05 0.005144915
## t2* 1.384639766 -1.392659e+00 0.308113272
## t3* 0.054284058  6.550327e-03 0.080502914
## t4* 0.002651239  2.902775e-02 1.025124191
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = stock[, i], statistic = calculate_ten, R = 500)
##
## ADBE
## Bootstrap Statistics :
##        original        bias    std. error
## t1* 0.004658172 -0.0003128057 0.005557615
## t2* 1.531350490 -1.5488458915 0.308158279
## t3* 0.066852814 -0.0004643335 0.084432468
## t4* 0.002914498  0.0959439905 1.624881322
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = stock[, i], statistic = calculate_ten, R = 500)
##
## ADP
## Bootstrap Statistics :
##         original        bias    std. error
## t1* 0.0008070075 -0.0001575263 0.002217006
## t2* 0.8476768625 -0.8481756702 0.121102341
## t3* 0.0250397117 -0.0016958585 0.079648817
## t4* 0.0008246488  0.0245506825 0.398464283
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = stock[, i], statistic = calculate_ten, R = 500)
##
## AMD
## Bootstrap Statistics :
##          original        bias    std. error
## t1* -0.0006186328 -0.0004941501  0.00651148
## t2*  2.3238266303 -2.3025975250  0.36644761
## t3* -0.0108132519 -0.0023720319  0.07731643
## t4* -0.0003935868 -0.0040588855  0.29366472
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = stock[, i], statistic = calculate_ten, R = 500)
##
## DELL
## Bootstrap Statistics :
##        original        bias    std. error
## t1* 0.008883858  8.399114e-05 0.004897839
## t2* 1.674991864 -1.699153e+00 0.268282608
## t3* 0.146273084  6.384468e-03 0.083969823
## t4* 0.005176448  3.176824e-02 0.716487034
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = stock[, i], statistic = calculate_ten, R = 500)
##
## GTW
## Bootstrap Statistics :
##         original        bias    std. error
## t1* -0.005425335 -0.0001193166 0.006393484
## t2*  2.232801517 -2.2310385494 0.375360850
## t3* -0.070808203  0.0030531692 0.079379380
## t4* -0.002557207 -0.0160853324 0.978199869
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = stock[, i], statistic = calculate_ten, R = 500)
##
## HP
## Bootstrap Statistics :
##        original        bias    std. error
## t1* 0.001900414 -0.0002475011 0.003918937
## t2* 0.875234288 -0.8607473472 0.210568268
## t3* 0.037255992 -0.0030129695 0.081517944
## t4* 0.002043947  0.0002245588 0.131905852
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = stock[, i], statistic = calculate_ten, R = 500)
##
## IBM
## Bootstrap Statistics :
##        original        bias    std. error
## t1* 0.002625623 -0.0001885272 0.003177288
## t2* 1.347923471 -1.3378651648 0.166954133
## t3* 0.063903599 -0.0002200483 0.083246848
## t4* 0.001820528 -0.0092710121 0.517361969
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = stock[, i], statistic = calculate_ten, R = 500)
##
## MSFT
## Bootstrap Statistics :
##        original        bias    std. error
## t1* 0.004255549 -2.454395e-05 0.003545019
## t2* 1.458538648 -1.472979e+00 0.205132957
## t3* 0.090733636  5.821146e-03 0.080850282
## t4* 0.002790306 -2.267812e-02 0.314784886
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = stock[, i], statistic = calculate_ten, R = 500)
##
## ORCL
## Bootstrap Statistics :
##        original        bias    std. error
## t1* 0.003910500 -0.0003313862 0.004895005
## t2* 1.567604154 -1.5842059378 0.281977154
## t3* 0.059855542 -0.0013396692 0.078883134
## t4* 0.002367197  0.0089001008 0.217022658
```

Problem 3.6 part (c) Test for each stock the null hypothesis alpha = 0

```r
stocklist <- list("AAPL","ADBE","ADP","AMD","DELL","GTW","HP","IBM","MSFT","ORCL")

for (i in (1:10)){
  Y = as.matrix(stock[,i])-sp500ret$X3mTCM
  X = sp500ret$risk_premium
  #Use linear regression on each stock
  model <- lm(Y~X)
  print(stocklist[[i]])
  cat("p value of alpha estimate:",summary(model)$coefficients[1,4])
  cat("\n")
  #perform hypothesis analysis at 95% confidence level
  if(summary(model)$coefficients[1,4] > 0.05){
    print("fail to reject null hypothesis that alpha is zero at 95% confidence level")
  } else {
    print("reject null hypothesis that alpha is zero at 95% confidence level")
  }
  cat("\n")
}
```

```
## [1] "AAPL"
## p value of alpha estimate: 0.447607
## [1] "fail to reject null hypothesis that alpha is zero at 95% confidence level"
##
## [1] "ADBE"
## p value of alpha estimate: 0.3421587
## [1] "fail to reject null hypothesis that alpha is zero at 95% confidence level"
##
## [1] "ADP"
## p value of alpha estimate: 0.6685821
## [1] "fail to reject null hypothesis that alpha is zero at 95% confidence level"
##
## [1] "AMD"
## p value of alpha estimate: 0.9168247
## [1] "fail to reject null hypothesis that alpha is zero at 95% confidence level"
##
## [1] "DELL"
## p value of alpha estimate: 0.03203205
## [1] "reject null hypothesis that alpha is zero at 95% confidence level"
##
## [1] "GTW"
## p value of alpha estimate: 0.3361941
## [1] "fail to reject null hypothesis that alpha is zero at 95% confidence level"
##
## [1] "HP"
## p value of alpha estimate: 0.6029036
## [1] "fail to reject null hypothesis that alpha is zero at 95% confidence level"
##
## [1] "IBM"
## p value of alpha estimate: 0.2752631
## [1] "fail to reject null hypothesis that alpha is zero at 95% confidence level"
##
## [1] "MSFT"
## p value of alpha estimate: 0.1481676
## [1] "fail to reject null hypothesis that alpha is zero at 95% confidence level"
##
## [1] "ORCL"
## p value of alpha estimate: 0.3797688
## [1] "fail to reject null hypothesis that alpha is zero at 95% confidence level"
```

Problem 3.6 part (d) Use the regression model (3.24) to test null hypothesis alpha = 0

```r
Y = as.matrix(stock[,1:10]-sp500ret$X3mTCM)
X = sp500ret$risk_premium
model <- lm(Y~X)
alpha <- coef(model)[1,]
beta <- coef(model)[2,]
n <- dim(Y)[1]
q <- dim(Y)[2]
residual <- Y - alpha - X%*%t(beta)
V <- t(residual)%*%residual/n

Fval <- ((n-q-1)/q)*alpha%*%ginv(V)%*%alpha/(1+mean(X)^2/mean((X-mean(X))^2))
cat("F-value is ",Fval)
```

```
## F-value is  0.9943758
```

```r
lower_CI <- qf(0.025,q,n-q-1)
upper_CI <- qf(0.975,q,n-q-1)

cat("\n")
```

```r
cat(" lower CI is: ",lower_CI)
```

```
##  lower CI is:  0.3195226
```

```r
cat(" upper CI is: ",upper_CI)
```

```
##  upper CI is:  2.137958
```

```r
cat("\n")
```

```r
cat("F value lies between CI, so we fail to reject the null hypothesis that alpha equals zero at 95% confidence level using the regression model (3.24)")
```

```
## F value lies between CI, so we fail to reject the null hypothesis that alpha equals zero at 95% confidence l
evel using the regression model (3.24)
```

$$\hat{\alpha} = \bar{y} - \bar{x}\hat{\beta}$$

$$\hat{\beta} = \left(\sum_{t=1}^{n}(x_t - \bar{x})\vec{y}\right) / \left(\sum_{t=1}^{n}(x_t - \bar{x})^2\right)$$

$$\hat{V} = n^{-1}\sum_{t=1}^{n}(\vec{y}_t - \hat{\alpha} - \hat{\beta}x_t)(\vec{y}_t - \hat{\alpha} - \hat{\beta}x_t)^T$$

$$residual = \vec{y}_t - \hat{\alpha} - \hat{\beta}x_t$$

$$Fvalue: \frac{\left(\frac{n-q-1}{q}\right)\hat{\alpha}^T\hat{V}^{-1}\hat{\alpha}}{1 + \frac{\bar{x}^2}{n^{-1}\sum_{t=1}^{n}(x_t - \bar{x})^2}} \sim F_{q,n-q-1} \text{ under } H_0$$