Risk Management

Rob Leonard (robleonard@tamu.edu (mailto:robleonard@tamu.edu))

Question 1: Parametric Estimation with Portfolio

Initial setup.

```
library(MASS); options(warn = -1)
load("HW06.Rdata")
ls()
```

```
## [1] "rt"  "syb" "wt"  "yt"
```

```
# setup skewness and kurtosis functions from HW 02
Sk.fun <- function(x) { ## function to compute skewness from HO 3
  mean((x-mean(x))^3/sd(x)^3)
}
Kur.fun <- function(x){ ## function to compute kurtosis
  mean((x-mean(x))^4/sd(x)^4)
}

stat           = matrix(ncol = dim(rt)[2], nrow=4)
rownames(stat) = c("skew","kurt","nu","se.nu")
colnames(stat) = colnames(rt)
for (i in 1:dim(rt)[2]){
  stat[1,i] = Sk.fun(rt[,i])
  stat[2,i] = Kur.fun(rt[,i])
  fit       = fitdistr(rt[,i], "t")
  stat[3,i] = fit$estimate[3]
  stat[4,i] = fit$sd[3]
}
stat
```

```
##               AMD        JNJ         KO        MCD       ORCL         PG
## skew   0.3492630 -0.4638567 -0.8399397 -0.1780772 -0.1877236  0.1191919
## kurt  14.1347994 14.1751790 14.0162658 39.6019951 18.7905731 17.0227981
## nu     3.0518712  3.0342409  3.0057943  2.9364931  2.9278199  3.0479331
## se.nu  0.2087198  0.2049450  0.1982409  0.1869666  0.1943795  0.1992369
##              TSLA        UNH
## skew  0.006428436 -0.4100226
## kurt  9.146944492 16.3806758
## nu    2.943646511  3.0183253
## se.nu 0.197061830  0.2042457
```

1a) Fit a multivariate-t model to the returns in rt. Show all the MLE's.

```
library(mnormt)
df         = seq(3.5,5.5, 0.01)
loglik_p   = c()
for (i in 1:length(df)) {
  fit = cov.trob(rt, nu = df[i])
  loglik_p[i] = sum(dmt(rt, mean = fit$center, S = fit$cov, df = df[i], log = T))
}
nu       = df[which.max(loglik_p)]
cat("The MLE of degrees of freedom:", paste(nu), "\n" )
```

```
## The MLE of degrees of freedom: 4.1
```

```
est = cov.trob(rt, nu = nu, cor = T)
cat("The MLE of the mean:\n")
```

```
## The MLE of the mean:
```

```
round(est$center,5) ## MLE of mean vector
```

```
##     AMD     JNJ      KO     MCD    ORCL      PG    TSLA     UNH
## 0.09446 0.06291 0.05370 0.08158 0.07317 0.05561 0.19251 0.12006
```

```
cat("\n","The MLE of the scale matrix lambda:\n")
```

```
##
##  The MLE of the scale matrix lambda:
```

```
round(est$cov,5)
```

```
##          AMD     JNJ      KO     MCD    ORCL      PG    TSLA     UNH
## AMD  7.34139 0.45019 0.40558 0.46485 1.08550 0.34600 1.83184 0.74619
## JNJ  0.45019 0.57718 0.28317 0.24992 0.37318 0.29518 0.32579 0.39769
## KO   0.40558 0.28317 0.58897 0.27525 0.33581 0.33616 0.33902 0.31227
## MCD  0.46485 0.24992 0.27525 0.61171 0.32391 0.23646 0.36817 0.31177
## ORCL 1.08550 0.37318 0.33581 0.32391 1.21971 0.32066 0.78439 0.52666
## PG   0.34600 0.29518 0.33616 0.23646 0.32066 0.56736 0.27347 0.29502
## TSLA 1.83184 0.32579 0.33902 0.36817 0.78439 0.27347 6.66620 0.57787
## UNH  0.74619 0.39769 0.31227 0.31177 0.52666 0.29502 0.57787 1.32153
```

```
cat("\n","The MLE of Cov:\n")
```

```
##
##  The MLE of Cov:
```

```
round(est$cov*nu/(nu-2),5)
```

```
##           AMD     JNJ      KO     MCD    ORCL      PG     TSLA     UNH
## AMD  14.33319 0.87894 0.79185 0.90757 2.11932 0.67553  3.57645 1.45685
## JNJ   0.87894 1.12687 0.55285 0.48794 0.72858 0.57630  0.63607 0.77644
## KO    0.79185 0.55285 1.14990 0.53740 0.65562 0.65631  0.66189 0.60967
## MCD   0.90757 0.48794 0.53740 1.19428 0.63240 0.46167  0.71880 0.60869
## ORCL  2.11932 0.72858 0.65562 0.63240 2.38134 0.62605  1.53143 1.02824
## PG    0.67553 0.57630 0.65631 0.46167 0.62605 1.10771  0.53392 0.57599
## TSLA  3.57645 0.63607 0.66189 0.71880 1.53143 0.53392 13.01497 1.12822
## UNH   1.45685 0.77644 0.60967 0.60869 1.02824 0.57599  1.12822 2.58012
```

1b) What is the distribution of the return of this portfolio? Please include all the parameter estimates.

```
mu.f = 3/253   # risk free daily return
wtadj = c(wt[1],wt[9],wt[7],wt[5],wt[3],wt[2],wt[8],wt[4],wt[6])  # wts in different order than
 rt
wtadj
```

```
## risk_free       AMD       JNJ        KO       MCD      ORCL        PG      TSLA
##    0.1000   -0.0224    0.0764   -0.0768    0.4367   -0.0437    0.0865    0.1128
##       UNH
##    0.3305
```

```
# sum(wt[2:9])  # risky weights only sum to .9, which is what we invest in the risky asset portf
olio
mu.P = .1*mu.f+sum(wtadj[2:9]*est$center)   # 10% risk free, 90% risky (already 90% in risky)
lambda.P = sqrt((wtadj[2:9])%*%est$cov%*%(wtadj[2:9]))  # weights already only sum to 90%
sig.P = lambda.P*sqrt(nu/(nu-2))
cat("\n Parametric estimates of portfolio:\n")
```

```
##
##  Parametric estimates of portfolio:
```

```
c(mean = mu.P, risk = sig.P,scale = lambda.P, DF = nu)
```

```
##       mean       risk      scale         DF
## 0.09838346 0.99188516 0.70987047 4.10000000
```

The distribution is a t distribution with estimated portfolio mean daily return of 0.098, estimated scale or lambda of 0.7098, risk of 0.9918 and estimated degrees of freedom of 4.1.

1c)

```
alpha = seq(.01,.05, .01)
S = 500000/100
q.t = qt(alpha, df = nu)
VaR.P = c()   # set up results vector storage
ES.P = c()    # set up results vector storage
for (i in 1:length(alpha)){
  VaR.P[i] = -S*(mu.P + lambda.P*q.t[i])
  ES.P[i] = S*(-mu.P+lambda.P*dt(q.t[i],nu)/alpha[i]*(nu+q.t[i]^2)/(nu-1))
}
cat("one-day risk estimates for: \n")
```

```
## one-day risk estimates for:
```

```
rbind(alpha,VaR.P,ES.P)
```

```
##           [,1]     [,2]      [,3]      [,4]      [,5]
## alpha     0.01     0.02     0.030     0.040     0.050
## VaR.P 12629.87 10039.88  8657.492  7723.962  7022.042
## ES.P  17669.18 14414.74 12708.174 11572.495 10729.698
```

Question 2: Single Stock

2a) Compute the estimated one-day VaR and ES using nonparametric estimation.

```
q = quantile(yt, alpha)
S = 200000/100
VaR.np = c()
ES.np = c()
for (i in 1:length(alpha)){
  VaR.np[i] = -S*q[i]
  ES.np[i] = -S*mean(yt[yt < q[i]])
}
cat("Nonparametric estimates:\n ")
```

```
## Nonparametric estimates:
##
```

```
rbind(q, VaR.np,ES.np)
```

```
##                  1%           2%           3%          4%          5%
## q         -4.827747    -4.007599    -3.481648   -3.107848   -2.759937
## VaR.np  9655.493470  8015.198171  6963.296175 6215.695213 5519.874038
## ES.np  14955.997170 11973.728764 10477.507573 9509.685503 8789.406210
```

2b) Parametric estimation with a t distribution.

```
library(MASS)
fit.t2b = fitdistr(yt, "t")         # fit univariate t
cat("\n")
```

```
fit.t2b$est
```

```
##          m          s         df
## 0.05590765 1.19508911 3.32135297
```

```
mu2b = fit.t2b$est["m"]
lambda2b = fit.t2b$est["s"]
nu2b = fit.t2b$est["df"]
q.t2b = qt(alpha, df = nu2b)
VaR.t2b = -S*(mu2b + lambda2b*q.t2b)
ES.t2b = S*(-mu2b+lambda2b*dt(q.t2b,nu2b)/alpha*(nu2b+q.t2b^2)/(nu2b-1))
cat("\n","Parametric t estimates:\n ")
```

```
##
##  Parametric t estimates:
##
```

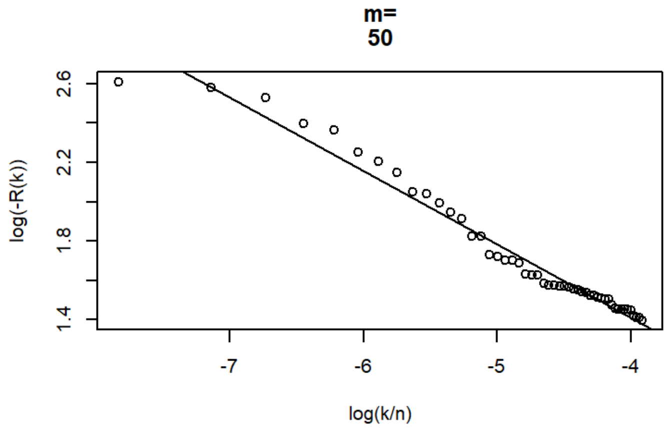
```
rbind(q.t2b, VaR.t2b,ES.t2b)
```

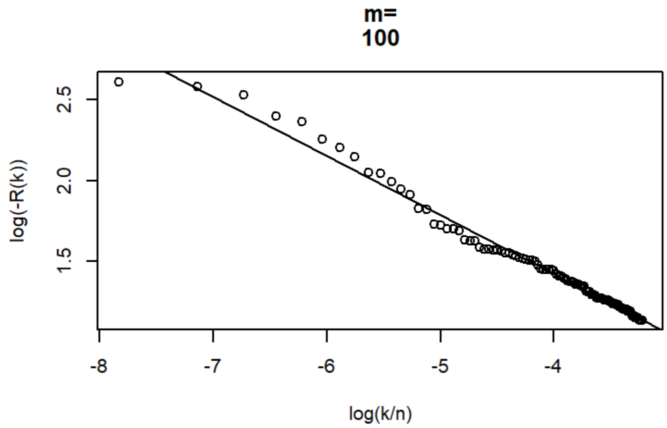
```
##                 [,1]         [,2]         [,3]        [,4]        [,5]
## q.t2b      -4.207469    -3.281817    -2.806931   -2.494196   -2.263377
## VaR.t2b  9944.785588  7732.313230  6597.248847 5849.757983 5298.058524
## ES.t2b  14764.830106 11722.664237 10187.769237 9191.320374 8465.411043
```

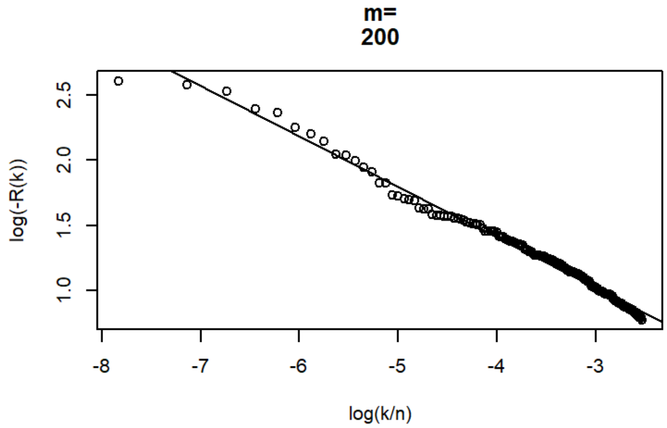
2c)

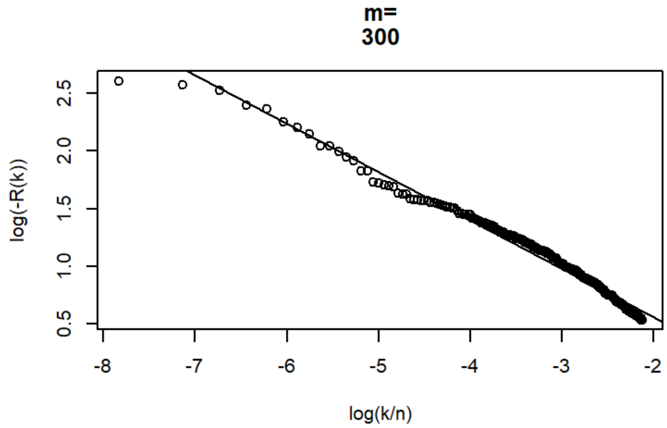
```r
par(mfrow   = c(3, 2))
yt2c = sort(as.numeric(yt)) # sort from smallest to largest
m = c(50,100,200,300,400)
n = length(yt2c)
out2c = matrix(nrow = 3, ncol = 5)
dimnames(out2c)[[1]] = c("corr", "slope","ahat")
dimnames(out2c)[[2]] = paste("m", m, sep = " = ")
for(i in 1:5){
  x = log((1:m[i])/n)
  y = log(-yt2c[1:m[i]])
  out2c[1,i] = cor(x,y)
  out2c[2,i] = lsfit(x,y)$coef[2] ## ols estimate of slope
  plot(x,y, main = c("m=",m[i]), xlab = "log(k/n)", ylab="log(-R(k))")
  abline(lsfit(x,y)$coef)
}
out2c[3,] = -1/out2c[2,] ## ahat = -1/slope
out2c
```

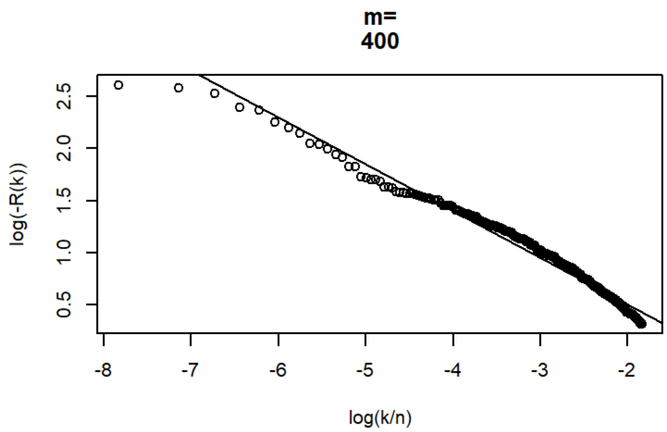
```
##           m = 50    m = 100    m = 200    m = 300    m = 400
## corr  -0.9853377 -0.9926753 -0.9945774 -0.9915708 -0.9884176
## slope -0.3725127 -0.3650425 -0.3880697 -0.4196750 -0.4490241
## ahat   2.6844725  2.7394072  2.5768565  2.3827963  2.2270517
```











Choose m = 200 as this is the most linear of the 5 plots. Calculate non parametric VaR0 piece:

```
a = out2c["ahat","m = 200"]
alpha0 = 0.1   # alpha 0 needs to be greater than any alpha in the vector (0.1 will work)
VaR0 = -S*quantile(yt2c,alpha0)
cat("Nonparametric estimates: ", paste(c("0.1-quantile","\t VaR(.1)"),
round(c(quantile(yt2c,alpha0), VaR0), 4), sep = " = "))
```

```
## Nonparametric estimates:  0.1-quantile = -1.8787      VaR(.1) = 3757.4198
```

Calculate VaR and ES by combining parametric and nonparametric pieces:

```
VaR = VaR0*(alpha0/alpha)^(1/a)
names(VaR) = paste(alpha)
ES = a/(a-1)*VaR
cat("Semiparametric Risk Estimate for alpha = ", alpha, ":", "\n")
```

```
## Semiparametric Risk Estimate for alpha =  0.01 0.02 0.03 0.04 0.05 :
```

```
rbind(VaR,ES)
```

```
##          0.01      0.02     0.03     0.04     0.05
## VaR  9182.468  7016.795 5995.192 5361.894 4917.115
## ES  15005.743 11466.658 9797.182 8762.262 8035.417
```