STAT 631 Homework 6

Jack Cunningham (jgavc@tamu.edu)

10/14/24

```r
load("HW05.Rdata")
```

3)

Computing the mean and variances of the return data:

```r
y.mu = apply(y,2,mean);y.mu
```

```
     AMZN        KO       NKE       PFE      TSLA       UNH       URI         V
0.4206254 0.1692590 0.2216723 0.1501095 0.6957931 0.4128102 0.5011812 0.3970618
```

```r
y.S = var(y);y.S
```

```
          AMZN       KO       NKE      PFE      TSLA       UNH       URI
AMZN 17.892637 2.431855  5.576128 2.564539 11.972386  2.977977  8.972939
KO    2.431855 6.238607  3.820411 2.972533  4.151542  3.972146  5.440775
NKE   5.576128 3.820411 14.638695 2.812862  9.371497  4.268385 10.709588
PFE   2.564539 2.972533  2.812862 8.787569  3.860339  4.452199  5.824990
TSLA 11.972386 4.151542  9.371497 3.860339 57.338308  6.231686 16.666210
UNH   2.977977 3.972146  4.268385 4.452199  6.231686 12.173168  7.763183
URI   8.972939 5.440775 10.709588 5.824990 16.666210  7.763183 39.914885
V     5.837551 3.829269  5.747779 3.145227  7.124705  4.292491  8.992063
            V
AMZN 5.837551
KO   3.829269
NKE  5.747779
PFE  3.145227
TSLA 7.124705
UNH  4.292491
URI  8.992063
V    9.207900
```

Defining various vectors for convenience:

```
N <- dim(y)[2]
ones <- rep(1,N)
zeros <- rep(0,N)
```

Given the constraints $-0.2 \leq w_i \leq 0.4$ we can set up our optimization for the solveLP function per the below:

$$\text{cvec} = \begin{bmatrix} \mu \\ -\mu \end{bmatrix}, \text{Amat} = \begin{bmatrix} I_n & 0 \\ 0 & I_n \\ 1^T & -1^T \end{bmatrix}, \text{bvec} = \begin{bmatrix} (0.4)1 \\ (0.2)1 \\ 1 \end{bmatrix}$$

First we obtain the feasible range of returns, by minimizing and maximizing our linear programming problem:

```
## Feasible range of m.R with b1 = 0.4 and b2 = 0.2
library(linprog)
```

```
Loading required package: lpSolve
```

```
b1 = 0.4; b2 = 0.2
cvec = c(y.mu, -y.mu)
Amat.lp = rbind(diag(2*N), c(ones, -ones))
bvec.lp = c(rep(b1,N), rep(b2,N),1)
inequ = c(rep("<=", 2*N), "=")
min.lp = solveLP(cvec = cvec, bvec = bvec.lp, Amat = Amat.lp,
                 lpSolve=T, const.dir = inequ, maximum = F)
max.lp = solveLP(cvec = cvec, bvec = bvec.lp, Amat = Amat.lp,
                 lpSolve=T, const.dir = inequ, maximum = T)
mu.lim = c(lower = min.lp$opt, upper = max.lp$opt); mu.lim;
```

```
    lower     upper
0.0517211 0.7039558
```

Since in part c we need to plot the efficient frontier we find efficient portfolios for the whole range of feasible returns:

```r
library(quadprog)
m.R = seq(round(mu.lim[1] + .0005, 3), round(mu.lim[2]-.0005, 3), 0.001)
sd.R = c();
wmat = matrix(nrow = length(m.R), ncol = N); colnames(wmat) = syb;
Amat = cbind(y.mu, ones, -diag(N), diag(N));
for(i in 1:length(m.R)){
  bvec = c(m.R[i],1, rep(-b1, N), rep(-b2,N))
  out = solve.QP(y.S, dvec = zeros, Amat=Amat, bvec = bvec, meq = 2);
  sd.R[i] = sqrt(2*out$value)
  wmat[i,] = out$solution
}
```

a)

In our previous step we found all the portfolios on the efficient frontier. The minimum variance is the portfolio where risk is minimized, we find this portfolio below:

```r
i.min = which.min(sd.R); ## index with smallest sd
w.min = wmat[i.min,];mu.min = m.R[i.min]; sd.min = sd.R[i.min];
cat("Minimum variance portfolio:"); w.min
```

```
Minimum variance portfolio:

       AMZN          KO         NKE         PFE        TSLA         UNH
 0.11294283  0.40000000  0.08538630  0.26937110 -0.01240684  0.07900288
        URI           V
-0.07339837  0.13910210
```

```r
c(return = mu.min, risk = sd.min)
```

```
 return    risk
0.21700 2.08737
```

b)

The tangency portfolio is the portfolio on the efficient frontier that maximizes the Sharpe ratio $\frac{E(R_p)-\mu_f}{\sigma_{R_p}}$. We find this portfolio below:

```
mu.f = 4.37/52

i.T = which.max((m.R - mu.f)/sd.R);
w.T = wmat[i.T,]; mu.T = m.R[i.T]; sd.T = sd.R[i.T]
cat("Tangency portfolio:"); w.T
```

```
Tangency portfolio:

       AMZN          KO         NKE         PFE        TSLA         UNH
 0.28752152 -0.02951931 -0.15926386 -0.11364370  0.15294867  0.40000000
        URI           V
 0.06195668  0.40000000
```

```
c(return = mu.T, risk = sd.T)
```

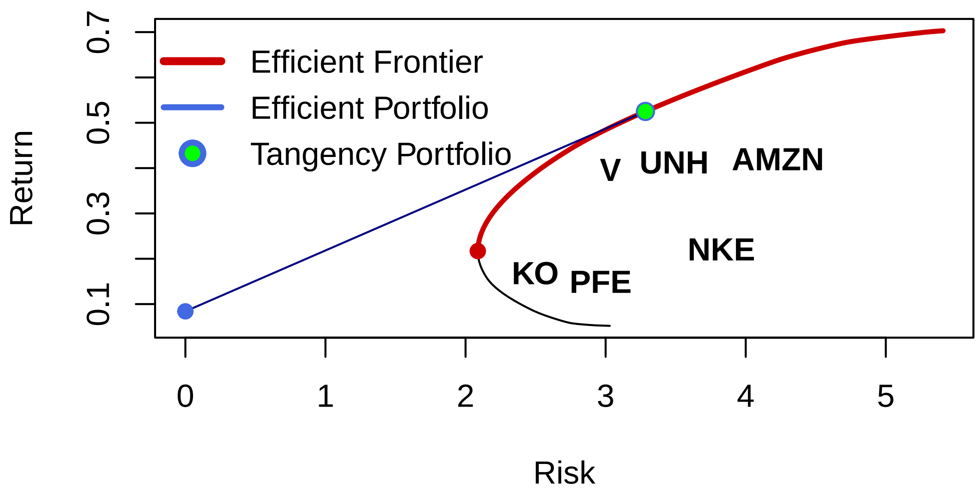
```
  return     risk
0.525000 3.284055
```

c)

```
s.min = sd.min
s.T = sd.T
y.sd = sqrt(diag(y.S))

plot(sd.R, m.R, type = "l", xlim = c(0,max(sd.R)), xlab = "Risk", ylab = "Return")
lines(sd.R[m.R > mu.min], m.R[m.R > mu.min], lwd = 2.5, col = "red3");  ## efficient frontie
lines(c(0,s.T), c(mu.f, mu.T), col = "navy")
points(0,mu.f, pch = 19, col = "royalblue") ## risk-free asset return
points(s.T,mu.T, pch = 21, col ="royalblue", bg = "green", cex = 1.2) ## tangency portfolio
points(s.min, mu.min, pch = 19, col = "red3") ## minimum variance portfolio
for(i in 1:N){ ## individual stock
    text(y.sd[i], y.mu[i], syb[i], font = 2)
}

legend("topleft",c("Efficient Frontier", "Efficient Portfolio","Tangency Portfolio"),lty=c(1
col=c("red3","royalblue","royalblue"), pt.bg=c("","","green"), pt.cex = c(NA,NA, 1.5), y.inte
```



d)

We are looking for an efficient portfolio with allowed risk of 2.5%. The risk of the tangency portfolio is 3.2840546%, since this is higher than our allowed risk we will have a portfolio with weight w_t in the tangency portfolio and w_f in the risk free asset with $w_t + w_f = 1$.

We find the appropriate weights by using the following formula for the risk of our desired portfolio, σ_p:

$$\sigma_p = w_t \sigma_t$$

Since we know the allowed risk of 2.5% and the risk of the tangency portfolio we can solve for w_t:

$$w_t = \frac{\sigma_p}{\sigma_t}$$

From this we can find the weight in the risk free asset through:

$$w_f = 1 - w_t$$

And the expected return of the portfolio, $E(R_p)$, through:

$$E(R_p) = w_f\mu_f + w_t\mu_t$$

We compute each below:

```
sd_p_d <- 2.5
w_t_d <- sd_p_d/sd.T
w_f_d <- 1 - w_t_d
mu_p_d <- w_f_d*mu.f + w_t_d*mu.T
w_p_d <- c(w_f_d, w_t_d*w.T);names(w_p_d) = c("Risk Free", syb)
cat("Portfolio:");w_p_d
```

```
Portfolio:
```

```
  Risk Free        AMZN          KO         NKE         PFE        TSLA
 0.23874591  0.21887693 -0.02247169 -0.12124026 -0.08651173  0.11643280
        UNH         URI           V
 0.30450163  0.04716477  0.30450163
```

```
c(return = mu_p_d, risk = sd_p_d)
```

```
   return      risk
0.4197222 2.5000000
```

e)

We are looking for an efficient portfolio with a target return of 0.55%. The expected return of the tangency portfolio is 0.525%, since the target return is greater than what is achieved by the tangency portfolio we need to find a portfolio on the efficient frontier with an expected return of 0.55%. Earlier we computed efficient portfolios for varying target returns, including 0.55%. It is below:

```
i.e <- which(m.R == 0.55)
w.e <- wmat[i.e,]
mu.e <- 0.55
sd.e <- sd.R[i.e]
cat("Portfolio:");w.e
```

```
Portfolio:
```

```
      AMZN          KO         NKE         PFE        TSLA         UNH 
 0.31596640 -0.05301547 -0.17949265 -0.14112158  0.17448726  0.40000000 
        URI           V 
 0.08317605  0.40000000 
```

```r
c(return = mu.e, risk = sd.e)
```

```
  return     risk 
0.550000 3.475885 
```

f)

We are looking for an efficient portfolio with target return of 0.85%. Recall that the feasible range of expected returns given our constraints is (0.0517211 , 0.7039558). Since the target return of 0.85% lies outside that range there is no efficient portfolio for this target return without loosening our constraints.