ORIE 4630: Spring Term 2019
Homework #5
Due: Tuesday, March 12, 2019

Students are required to work independently on homework. You should not give or receive help from other students. You should also not receive help from students or former students who took this course in previous years and who may have solutions to similar problems. The solutions you submit should be your own work and not copied from elsewhere.

Homework is due at the end of lecture (12:55pm) on the due date. You will usually have one week to do the assignments. Please don't wait until the homework is nearly due to start. Late homework is not accepted. Also, homework is not accepted by email. You can submit your assignment in lecture or in the drop box in Rhodes Hall.

Please print your name on the front of your homework so that it is legible.

Include your `R` code, output, graphs, and other work with your homework. This will allow the grader to find any errors you make and to give partial credit.

This assignment uses net returns of certain stocks and indexes from Jan 4, 2006 to Aug 18, 2017. The log gross returns for the stocks are contained in a csv file named `returns.csv`. You should download this file from the course Blackboard site and put it in your `R` or `Rstudio` working directory. This file has 35 columns: the first column shows the date (Date); the next 30 columns are for the stocks that are the components of the Dow Jones Industrial average (DOW); the final four columns are for the Dow Jones Industrial average index (DOW), the NASDAQ composite index (NASD), the NASDAQ 100 index (NASD100), and the S&P 500 index (SP500). There are 2927 days of log gross returns. The log gross returns are calculated from the adjusted closing prices downloaded from Yahoo.

Start `R` or `Rstudio` and run the following lines:

```
1  Returns=read.csv("returns.csv")
2  names(Returns)
3  stock.names=c("AAPL", "MCD", "PFE")
4  R=Returns[, stock.names]
5  head(R, 10)
6  R=exp(R)-1
7  head(R, 10)
8  mu_R=colMeans(R)
9  mu_R
10 cov_R=cov(R)
11 cov_R
```

Line 1 reads the log gross returns into a data frame named `Returns`; line 2 outputs the names of the stocks and indexes to which the log gross returns pertain. If `returns.csv` is not in your working directory, then you need to give a complete path to that file in line 1. In line 3, the symbols for the three stocks of interest, Apple (AAPL), McDonald's (MCD), and Pfizer (PFE), are stored in the object `stock.names`, and in line 4, the log gross returns for the stocks of interest are stored in the data frame `R`. Line 5 produces output of the first ten log gross returns in the data frame `R`. The log gross returns in the data frame `R` are converted to net returns in line 6. Recall that if the net return of an asset is R, then the log gross return is $r = \log(1+R)$; thus, $r = e^R - 1$. Furthermore, recall that if R is small, then $r \sim R$. Line 7 produces output of the first ten net returns from the data frame R. Note that the net returns are very similar to the log gross returns. The sample mean of the net returns is computed for each of the three stocks of interest in line 8, and the means are output in line 9. The sample covariances of the net returns for the three stocks of interest are computed in line 10, and the covariances are output in line 11.

The function `solve.QP()` is contained in the quadratic programming package `quadprog`. To install and load the package `quadprog`, run the following lines:

```
12 install.packages("quadprog")
13 library("quadprog")
```

Consider now a portfolio consisting of the three stocks Apple (AAPL), McDonald's (MCD), and Pfizer (PFE). Let the net return of the portfolio be denoted by R_p. The expected return of the portfolio is $\mu_P = w^\top \mu_R$, where w is a column vector of weights that sum to 1 and μ_R is a column vector consisting of the mean net returns for the three stocks of interest. The variance of the return of the portfolio is $\sigma_p^2 = w^\top \Omega_R w$, where Ω_R is the covariance matrix of the the returns for the three stocks of interest. While the population quantities μ_R and Ω_R are unknown to us, they have been estimated from data in lines 8 and 10, respectively.

To find the weights that determine the smallest-risk portfolio for $\mu_P = 0.0008$, run the following lines:

```
14 N=length(mu_R)
15 A_eq=cbind(rep(1, N), mu_R)
16 mu_P=0.0008
17 d=rep(0, N)
18 b_eq=c(1, mu_P)
19 result=solve.QP(Dmat=2*cov_R, dvec=d, Amat=A_eq, bvec=b_eq, meq=2)
20 rbind(c("sigma_P"), round(sqrt(result$value), 7))
21 rbind(c("w_1", "w_2", "w_3"), round(result$solution, 7))
```

Recall that the general quadratic programming problem being solved by the function `solve.QP()` is to minimize the function $\frac{1}{2}x^\top Dx - d^\top x$ subject to the equality constraints $A_{\text{eq}}^\top x = b_{\text{eq}}$ and the inequality constraints $A_{\text{neq}}^\top x \geq b_{\text{neq}}$, where m_{eq} is the number of columns of A_{eq} and rows of b_{eq}, and m_{neq} is the number of columns of A_{neq} and rows of b_{neq}. In the present portfolio problem, $m_{\text{eq}} = 2$ and $m_{\text{neq}} = 0$. The standard deviation, i.e., the risk, of the smallest-risk portfolio is output in line 20. The weights corresponding to the smallest-risk portfolio are output in line 21.

To repeat this process for 5,000 values of μ_P, equally spaced between $\min(\mu_P)$ and $k * \max(\mu_P)$ with $k = 1$, to plot the efficient frontier and to find the overall minimum-variance portfolio, run the following lines:

```
22 N=length(mu_R)
23 A_eq=cbind(rep(1, N), mu_R)
24 n=5000; k=1
25 mu_P=seq(min(mu_R), k*max(mu_R), length=n)
26 d=rep(0, N)
27 weights=matrix(0, nrow=n, ncol=N)
28 sd_P=rep(0, n)
29 for (i in 1:n) {
30   b_eq=c(1, mu_P[i])
```

```
31   result=solve.QP(Dmat=2*cov_R, dvec=d, Amat=A_eq, bvec=b_eq, meq=2)
32   sd_P[i]=sqrt(result$value)
33   weights[i,]=result$solution }
34 plot(sd_P, mu_P, type="l", xlim=c(0, max(sd_P)), ylim=c(min(mu_P), max(mu_P)),
35           lwd=2, cex.axis=1.5, cex.lab=1.5)
36 ind_MV=(sd_P==min(sd_P))
37 points(sd_P[ind_MV], mu_P[ind_MV], cex=2, pch="+", col="green")
38 rbind(c("mu_MV"), round(mu_P[ind_MV], 7))
39 rbind(c("sd_MV"), round(sd_P[ind_MV], 7))
40 rbind(c("w_1_MV", "w_2_MV", "w_3_MV"), round(weights[ind_MV,], 7))
41 ind_EF=(mu_P>mu_P[ind_MV])
42 lines(sd_P[ind_EF], mu_P[ind_EF], type="l", lwd=3, col="red")
```

The matrix A_{eq} is specified in line 23. The 5,000 values of μ_P are created in line 25 and stored in the object `mu_P`. The vector d is created in line 26 and stored in the object `dvec`. The object `dvec` consists entirely of 0s. In lines 27 and 28, the objects `weights` and `sd_P` are created to consist entirely of 0s. In lines 29 to 33, the function `solve.QP()` is run for each of the 5,000 values of μ_P stored in the object `mu_P`. Note that the vector b_{eq} is created in line 30. The value of σ_P for each value of μ_P is stored in the object `sd_P` in line 32; the portfolio weights for each value of μ_P are stored as rows in the object `weights` in line 33. The values of (μ_P, σ_P) are plotted in lines 34 and 35. The index of the value μ_P in the object `mu_P` for which the risk σ_P is smallest (corresponding to the so-called minimum-variance portfolio) is determined in line 36. Line 37 marks the point $(\mu_{\text{MV}}, \sigma_{\text{MV}})$ by a green cross on the plot. The corresponding expected return μ_{MV}, minimum risk σ_{MV}, and weights w_{MV} are output by lines 38, 39, and 40. The values of μ_P in the object `mu_P` for which $\mu_P > \mu_{\text{MV}}$ are determined in line 41. These indices indicate the values of μ_P that fall on the efficient frontier. The efficient frontier is plotted in line 42.

To plot the weights for the portfolios on the efficient frontier as a function of μ_P, run the following lines:

```
43 plot(mu_P[ind_EF], weights[ind_EF,1], type="l", lwd=2, ylim=c(min(weights),
44           max(weights)), lty=1, col="red", xlab="mu_P", ylab="weight")
45 lines(c(min(mu_P[ind_EF]), max(mu_P[ind_EF])), c(0,0))
46 lines(mu_P[ind_EF], weights[ind_EF,2], type="l", lwd=2, lty=2, col="blue")
47 lines(mu_P[ind_EF], weights[ind_EF,3], type="l", lwd=2, lty=3, col="brown")
48 legend("bottomleft", legend=c("w_1", "w_2", "w_3"), lty=c(1,2,3), lwd=c(2,2,2),
49           col=c("red","blue","brown"))
```

It turns out, for example, that the smallest-risk portfolio for $\mu_P = 0.0009$ has a negative weight. If shorting is not permitted, then the goal is to find the smallest-risk portfolio having non-negative weights. To find the smallest-risk portfolio having non-negative weights for given expected return $\mu_P = 0.0009$, run the following lines:

```
50 N=length(mu_R)
51 A_eq=cbind(rep(1, N), mu_R)
52 A_neq=diag(1, nrow=N)
53 mu_P=0.0009
54 d=rep(0, N)
55 b_eq=c(1, mu_P)
56 b_neq=rep(0, N)
57 result=solve.QP(Dmat=2*cov_R, dvec=d, Amat=cbind(A_eq, A_neq),
58           bvec=c(b_eq, b_neq), meq=2)
```

```
59 rbind(c("sigma_P"), round(sqrt(result$value), 7))
60 rbind(c("w_1", "w_2", "w_3"), round(result$solution, 7))
```

The matrix A_{neq} for the inequality contraints is specified in line 52; the corresponding vector of constants b_{neq} for the inequality constraints is specified in line 56. Line 59 outputs the standard deviation σ_{P} of the smallest-risk portfolio having non-negative weights, and line 60 outputs the weights of the portfolio.

Under the condition of non-negative weights, to plot the efficient frontier, and to find the minimum-variance portfolio, based on 5,000 values of μ_P, equally spaced between $\min(\mu_P)$ and $\max(\mu_P)$, run the following lines:

```
61 N=length(mu_R)
62 A_eq=cbind(rep(1, N), mu_R)
63 A_neq=diag(1, nrow=N)
64 n=5000
65 mu_P=seq(min(mu_R), max(mu_R), length=n)
66 d=rep(0,N)
67 weights=matrix(0, nrow=n, ncol=N)
68 sd_P=rep(0, n)
69 for (i in 1:n) {
70   b_eq=c(1, mu_P[i])
71   b_neq=rep(0, 3)
72   result=solve.QP(Dmat=2*cov_R, dvec=d, Amat=cbind(A_eq, A_neq),
73           bvec=c(b_eq, b_neq), meq=2)
74   sd_P[i]=sqrt(result$value)
75   weights[i,]=result$solution }
76 plot(sd_P, mu_P, type="l", xlim=c(0 ,max(sd_P)), ylim=c(min(mu_P), max(mu_P)),
77           lwd=2, cex.axis=1.5, cex.lab=1.5 )
78 ind_MV=(sd_P==min(sd_P))
79 points(sd_P[ind_MV], mu_P[ind_MV], cex=2, pch="+", col="green")
80 rbind(c("mu_MV"), round(mu_P[ind_MV], 7))
81 rbind(c("sd_MV"), round(sd_P[ind_MV], 7))
82 rbind(c("w_1_MV", "w_2_MV", "w_3_MV"), round(weights[ind_MV,], 7))
83 ind_EF=(mu_P>mu_P[ind_MV])
84 lines(sd_P[ind_EF], mu_P[ind_EF], type="l", lwd=3, col="red")
```

Line 80 outputs μ_{MV}, the expected return of the minimum-variance portfolio; line 81 outputs σ_{MV}, the standard deviation of the minimum-variance portfolio; and line 82 outputs the weights $w_{1,MV}, w_{2,MV}, w_{3,MV}$ for the minimum-variance portfolio.

To plot the non-negative weights for the portfolios on the efficient frontier as a function of μ_P, run the lines 43 to 49 again.

Now consider the problem of combining a portfolio on the efficient frontier with a risk-free asset. First, consider the case where no restrictions are placed on the weights; in particular, shorting is allowed. To find the minimum-variance portfolio and the tangency portfolio, and to plot these points on the efficient frontier along with the risk-return line for the optimal portfolios in the case that the risk-free rate is $\mu_f = 0.00015$, run the following lines:

```
85 mu_F=0.00015
86 N=length(mu_R)
87 A_eq=cbind(rep(1, N), mu_R)
88 n=5000; k=1
89 mu_P=seq(min(mu_R), k*max(mu_R), length=n)
```

```
90 d=rep(0, N)
91 weights=matrix(0, nrow=n, ncol=N)
92 sd_P=rep(0, n)
93 for (i in 1:n) {
94   b_eq=c(1, mu_P[i])
95   result=solve.QP(Dmat=2*cov_R, dvec=d, Amat=A_eq, bvec=b_eq, meq=2)
96   sd_P[i]=sqrt(result$value)
97   weights[i,]=result$solution }
98 plot(sd_P,mu_P, type="l", xlim=c(0,max(sd_P)), ylim=c(mu_F,max(mu_P)),
99           lwd=2, cex.axis=1.5, cex.lab=1.5)
100 ind_MV=(sd_P==min(sd_P))
101 points(sd_P[ind_MV], mu_P[ind_MV], cex=2, pch="+", col="green")
102 rbind(c("mu_MV"), round(mu_P[ind_MV], 7))
103 rbind(c("sd_MV"), round(sd_P[ind_MV], 7))
104 rbind(c("w_1_MV", "w_2_MV", "w_3_MV"), round(weights[ind_MV,], 7))
105 points(0, mu_F, cex=3, pch="*", col="brown")
106 ind_EF=(mu_P>mu_P[ind_MV])
107 lines(sd_P[ind_EF], mu_P[ind_EF], type="l", lwd=3, col="red")
108 sharpe=(mu_P-mu_F)/sd_P
109 ind_T=(sharpe==max(sharpe))
110 rbind(c("mu_T"), round(mu_P[ind_T], 7))
111 rbind(c("sd_T"), round(sd_P[ind_T], 7))
112 rbind(c("w_1_T", "w_2_T", "w_3_T"), round(weights[ind_T, ], 7))
113 points(sd_P[ind_T], mu_P[ind_T], cex=3, pch="*", col="brown")
114 lines(c(0, max(sd_P[ind_EF])), mu_F+c(0, max(sd_P[ind_EF]))*sharpe[ind_T],
115           lwd=3, col="blue")
```

Line 102 outputs μ_{MV}, the expected return of the minimum-variance portfolio; line 103 outputs σ_{MV}, the standard-deviation of the minimum variance portfolio; and line 104 outputs the weights $w_{1,MV}, w_{2,MV}, w_{3,MV}$ for the minimum-variance portfolio. Line 110 outputs μ_T, the expected return of the tangency portfolio; line 111 outputs σ_T, the standard deviation of the tangency portfolio; and line 112 outputs the weights $w_{1,T}, w_{2,T}, w_{3,T}$ for the tangency portfolio.

To find the optimal portfolio, i.e., the portfolio that is a combination of the risk-free asset and the tangency portfolio, for given expected return $\mu_O = 0.0008$, run the following lines:

```
116 mu_O=0.0008
117 w=(mu_O-mu_F)/(mu_P[ind_T]-mu_F)
118 sd_O=w*sd_P[ind_T]
119 rbind(c("w_F", "w_T"), round(c(1-w, w), 7))
120 rbind(c("sd_O"), round(sd_O, 7))
121 rbind(c("w_F_O", "w_1_O", "w_2_O", "w_3_O"), round(c(1-w, w*weights[ind_T,]), 7))
```

Line 119 outputs the weights w_F and w_T that determine the optimal portfolio: w_F is the weight put on the risk-free asset and w_T is the weight put on the tangency portfolio; of course, $w_F = 1 - w_T$. Line 120 outputs the standard deviation of the optimal portfolio. The weights $w_{F,O}, w_{1,O}, w_{2,O}, w_{3,O}$ assigned to the four individual assets (the risk-free asset and the three risky assets) are output by line 121. Note that, in this particular example, one of the weights is negative, so the optimal portfolio would require shorting.

Suppose now that an optimal portfolio is sought, but shorting is not permitted. In this situation, the optimal portfolio should be based on the tangency portfolio from the efficient frontier where the weights are all constrained to be non-negative. Under the requirement that all weights must be non-negative, in the case that the risk-free rate is $\mu_f = 0.00015$, to find the minimum-variance portfolio and the tangency portfolio, and to plot these points on the efficient frontier along with the risk-return line for the optimal portfolios, run the following lines:

```
122 mu_F=0.00015
123 N=length(mu_R)
124 A_eq=cbind(rep(1, N), mu_R)
125 A_neq=diag(1, nrow=N)
126 n=5000; k=1
127 mu_P=seq(min(mu_R), k*max(mu_R), length=n)
128 d=rep(0, N)
129 weights=matrix(0, nrow=n, ncol=N)
130 sd_P=rep(0, n)
131 for (i in 1:n)  {
132   b_eq=c(1, mu_P[i])
133   b_neq=rep(0, 3)
134   result=solve.QP(Dmat=2*cov_R, dvec=d, Amat=cbind(A_eq, A_neq),
135           bvec=c(b_eq, b_neq), meq=2)
136   sd_P[i]=sqrt(result$value)
137   weights[i,]=result$solution }
138 plot(sd_P, mu_P, type="l", xlim=c(0,max(sd_P)), ylim=c(mu_F, max(mu_P)),
139           lwd=2, cex.axis=1.5, cex.lab=1.5)
140 ind_MV=(sd_P==min(sd_P))
141 points(sd_P[ind_MV], mu_P[ind_MV], cex=2, pch="+", col="green")
142 rbind(c("mu_MV"), round(mu_P[ind_MV], 7))
143 rbind(c("sd_MV"), round(sd_P[ind_MV], 7))
144 rbind(c("w_1_MV", "w_2_MV", "w_3_MV"), round(weights[ind_MV,], 7))
145 points(0, mu_F, cex=3, pch="*", col="brown")
146 ind_EF=(mu_P>mu_P[ind_MV])
147 lines(sd_P[ind_EF], mu_P[ind_EF], type="l", lwd=3, col="red")
148 sharpe=(mu_P-mu_F)/sd_P
149 ind_T=(sharpe==max(sharpe))
150 rbind(c("mu_T"), round(mu_P[ind_T], 7))
151 rbind(c("sd_T"), round(sd_P[ind_T], 7))
152 rbind(c("w_1_T", "w_2_T", "w_3_T"), round(weights[ind_T, ], 7))
153 points(sd_P[ind_T], mu_P[ind_T], cex=3, pch="*", col="brown")
154 lines(c(0, max(sd_P[ind_EF])), mu_F+c(0,max(sd_P[ind_EF]))*sharpe[ind_T],
155           lwd=3, col="blue")
```

To find the optimal portfolio, i.e., the portfolio that is a combination of the risk-free asset and the tangency portfolio, for given expected return $\mu_O = 0.0008$, run lines 116 to 121.

Questions:

1. [10 points] Adapt and run lines 1 to 11 for the stocks IBM (IBM), General Electric (GE), and Nike (NKE).

i) Report the second log gross return r_5 and the second net return R_5 for General Electric (GE). Are the values for r_5 and R_5 close? Explain this phenomenon briefly.
ii) Submit the output from lines 9 and 11.
iii) Report the sample mean for Nike (NKE).

iv) Report the sample variance for IBM (IBM).
v) Report the sample covariance between IBM (IBM) and General Electric (GE).
vi) Report the sample correlation between General Electric (GE) and Nike (NKE). (Hint: if you use the function `cor()`, submit your output.)

2. [10 points] Run lines 12 and 13. Adapt and run lines 14 to 21 for the stocks IBM (IBM), General Electric (GE), and Nike (NKE) to find the smallest-risk portfolio having expected return $\mu_P = 0.0006$.

i) Submit the output from lines 20 and 21.
ii) Report the standard deviation of the smallest-risk portfolio having expected return $\mu_P = 0.0006$.
iii) Report the weight that is assigned to IBM (IBM) in the smallest-risk portfolio having expected return $\mu_P = 0.0006$.
iv) Does the smallest-risk portfolio require shorting any of the assets? Explain your answer briefly.

3. [10 points] Adapt and run lines 22 to 42 for the stocks IBM (IBM), General Electric (GE), and Nike (NKE). Use $k = 2$.

i) Submit the plot of the efficient frontier.
ii) Report the mean of the minimum-variance portfolio.
iii) Report the standard deviation of the minimum-variance portfolio.
iv) Report the weights on the assets for the minimum-variance portfolio.

4. [5 points] Adapt and run lines 43 to 49 for the stocks IBM (IBM), General Electric (GE), and Nike (NKE). Submit the plot of the weights as a function of the portfolio return.

5. [10 points] Adapt and run lines 50 to 60 for the stocks IBM (IBM), General Electric (GE), and Nike (NKE) to find the smallest-risk portfolio having non-negative weights and expected return $\mu_P = 0.0006$.

i) Submit the output from lines 59 and 60.
ii) Report the standard deviation of the smallest-risk portfolio having expected return $\mu_P = 0.0006$.
iii) Report the weight that is assigned to General Electric (GE) in the smallest-risk portfolio having expected return $\mu_P = 0.0006$.
iv) Does the smallest-risk portfolio require shorting any of the assets? Explain your answer briefly.

6. [10 points] Adapt and run lines 61 to 84 for the stocks IBM (IBM), General Electric (GE), and Nike (NKE) to plot the efficient frontier under the constraint of non-negative weights. Re-run lines 43 to 49.

i) Submit the plot of the efficient frontier.
ii) Submit the plot of the weights.
iii) Does the minimum-variance portfolio change when the condition of non-negative weights is imposed? Briefly explain why it does or does not change.

7. [10 points] Adapt and run lines 85 to 115 for the stocks IBM (IBM), General Electric (GE), and Nike (NKE) to plot the risk-return line of the optimal portfolios and to find the tangency portfolio in the case of a risk-free rate $\mu_f = 0.0001$. Use $k = 1.5$.

i) Submit the plot of the risk-return line.
ii) Report the expected return of the tangency portfolio.
iii) Report the standard deviation of the tangency portfolio.
iv) Report the weights on the assets for the tangency portfolio. Are any of the assets shorted in the tangency portfolio?

8. [10 points] Adapt and run lines 116 to 121 for the stocks IBM (IBM), General Electric (GE), and Nike (NKE) to find the optimal portfolio having expected return $\mu_P = 0.0006$ in the case of a risk-free rate $\mu_f = 0.0001$.

i) Submit the output from lines 119, 120, and 121.
ii) Report the standard deviation of the optimal portfolio having expected return $\mu_P = 0.0006$.
iii) Report the weight that is assigned to the risk-free asset in the optimal portfolio having expected return $\mu_P = 0.0006$.
iv) Report the weights assigned to Chevron (CVX), Cisco (CSCO), and J. P. Morgan (JPM) in the optimal portfolio.

9. [15 points] Adapt and run lines 122 to 155 for the stocks IBM (IBM), General Electric (GE), and Nike (NKE) to plot the risk-return line for the optimal portfolios and to find the tangency portfolio in the case of risk-free rate $\mu_f = 0.0001$ under the constraint of non-negative weights on the assets.

i) Submit the plot of the risk-return line.
ii) Report the expected return of the tangency portfolio.
iii) Report the standard deviation of the tangency portfolio.
iv) Report the weights on the assets for the tangency portfolio.
v) Compare the tangency portfolio for this question with that from Question 7. Explain the similarity or difference between the two tangency portfolios.
vi) Adapt and run lines 116 to 121 to find the optimal portfolio having expected return $\mu_P = 0.0006$ in the case of risk-free rate $\mu_f = 0.0001$ under the constraint of non-negative weights on the assets. Include the output from line 120. By comparison with your answer to Question 8, what is the price of imposing non-negative weights?

10. [10 points] Adapt and run lines 122 to 155 for the stocks IBM (IBM), General Electric (GE), and Nike (NKE) to plot the risk-return line for the optimal portfolios and to find the minimum variance and tangency portfolio in the case of risk-free rate $\mu_f = 0.0001$ under the simultaneous constraints that i) all weights must be at least $-.3$; and ii) the weight on IBM (IBM) is at most .4.
i) Submit the plot of the risk-return line.
ii) Submit the plot of the weights.
iii) Report the expected return of the minimum-variance portfolio.
iv) Report the standard deviation of the minimum-variance portfolio.
v) Report the expected return of the tangency portfolio.
vi) Report the standard deviation of the tangency portfolio.
vii) What is the standard deviation of the optimal portfolio having expected return $\mu_P = 0.0006$ in the case of risk-free rate $\mu_f = 0.0001$ under the present constraints on the weights on the assets.