FE 630
Fall 2018
Homework 4
Vivek Sathyanarayana
CWID:10442999

Introduction:

The Black-Litterman asset allocation model, created by Fischer Black and Robert Litterman of Goldman, Sachs & Company, is a sophisticated method used to overcome the problem of unintuitive, highly-concentrated, input-sensitive portfolios. Input sensitivity is a well-documented problem with mean- variance optimization and is the most likely reason that more portfolio managers do not use the Markowitz paradigm, in which return is maximized for a given level of risk. The Black-Litterman Model uses a Bayesian approach to combine the subjective views of an investor regarding the expected returns of one or more assets with the market equilibrium vector (the prior distribution) of expected returns to form a new, mixed estimate of expected returns.

The paper was replicated through code on an R program and the results were compared to the data provided in the research paper.

Program:

The data assumed to be provided was the following:

- Table 1
- Market Capitalization Weight
- Covariance Matrix
- P Matrix
- Q Matrix
- Tau
- LC
- Delta
- Risk Free Rate

Table 1 consists of Estimates of Expected Total Return Vectors.

The first part of the code uploads all the data as tables/ matrices from the csv file. It is important to remember to set the Working Directory as the location containing the program, as well as all the csv files containing the relevant data. The code snippet is provided below:

```r
#Upload available information as CSV files
table1 <- read.csv(file = "Table1.csv", header = TRUE)
covmatrix <- data.frame(read.csv(file = "Covariance.csv", header = TRUE))
row.names(covmatrix) <- covmatrix[,1] #Set rownames from first column
covmatrix <- covmatrix[,-1] #Delete first column containing stock names
covmatrix <- as.matrix(covmatrix)
Pmatrix <- data.matrix(read.csv(file = "PMatrix.csv", header = FALSE))
MarketCapWeights <- data.matrix(read.csv(file = "MarketCapWeights.csv", header = FALSE)) #value*100 for percentage

delta =2.25 #Risk aversion parameter provided in Appendix B(2)
riskfree= 0.05 #Risk free rate of 5% provided
```

The Implied Equilibrium Return is then computed using the formula: $\boldsymbol{\Pi = \delta \Sigma w}$

The code and output for computing Implied Equilibrium Return Values is given below, along with its percentage difference with respect to the values provided in the research paper:

```
#Compute Implied Equilibrium Returns to plug into E[R]
ImpliedEqReturn = delta*(covmatrix%*%MarketCapWeights)
ImpliedEqReturn = (ImpliedEqReturn + riskfree)*100 #Factoring in risk free rate and converting to percent
#Compare Computed Implied Equilibrium Returns with Data provided in research paper
ImpliedEqAccuracy <- abs(ImpliedEqReturn-table1[,4])/((ImpliedEqReturn+table1[,4])/2)
colnames(ImpliedEqAccuracy) <- c("Percentage Difference (%)")
#Round off small percentages to 0
ImpliedEqAccuracy <- round(ImpliedEqAccuracy,4)
```

```
> ImpliedEqReturn
            V1
aa   13.811630
ge   13.584088
jnj   9.766805
msft 20.404243
axp  14.923580
gm   12.807612
jpm  16.462535
pg    7.557463
ba   11.782445
hd   12.513673
ko   10.897925
sbc   8.789180
c    16.970540
hon  14.513203
mcd  10.439893
t    10.775097
cat  10.930775
hwp  14.439088
mmm   8.658297
utx  15.481220
dd   10.970263
ibm  14.658035
mo    6.853280
wmt  12.781580
dis  12.409070
intc 18.700700
mrk   9.203202
xom   7.857298
ek   10.616090
ip   12.948552
```

```
> ImpliedEqAccuracy
     Percentage Difference (%)
aa                      0.0001
ge                      0.0010
jnj                     0.0017
msft                    0.0003
axp                     0.0011
gm                      0.0017
jpm                     0.0002
pg                      0.0003
ba                      0.0023
hd                      0.0005
ko                      0.0020
sbc                     0.0001
c                       0.0000
hon                     0.0009
mcd                     0.0000
t                       0.0033
cat                     0.0010
hwp                     0.0008
mmm                     0.0002
utx                     0.0007
dd                      0.0009
ibm                     0.0001
mo                      0.0010
wmt                     0.0009
dis                     0.0001
intc                    0.0000
mrk                     0.0018
xom                     0.0029
ek                      0.0006
ip                      0.0022
```

As we can see, the program results were accurate, since the percentage difference is very low.

The weights in table 2 were computed using the formula in Appendix B(7) :

$$w = (\delta\Sigma)^{-1}\mu$$

Where delta =2.25 and Mu vector is the expected return vectors available in table 1:

Code snippet for table 2 and output is given below. The data has been rounded to 2 decimal places, as done in the research paper for easy comparison:

```r
#Table 2
table2 <- matrix(0,nrow = 30, ncol = 4)
rownames(table2) <- table1[,1]
#Compute Weights using formula in Appendix B(7): w = (δΣ)−1 μ
for (j in 1:3) {
   table2[,j] <- ((solve(delta*covmatrix)%*%(table1[,j+1]-(riskfree*100))))
}
table2[,4] <- cbind(MarketCapWeights*100)
colnames(table2) <- c("Historical Weight (%)","CAPM Weight (%)", "Implied Eq. Weight (%)", "Market Cap. Weight (%)")
#Round off data to 2 decimal places
table2 <- round(table2,2)
```

```
> table2
```

	Historical Weight (%)	CAPM Weight (%)	Implied Eq. Weight (%)	Market Cap. Weight (%)
aa	237.89	2.21	1.23	0.88
ge	-64.41	9.81	11.24	11.62
jnj	-78.25	5.50	4.89	5.29
msft	4.29	3.29	10.46	10.41
axp	-41.17	6.08	1.41	1.39
gm	5.64	3.84	1.33	0.79
jpm	-218.55	1.85	2.20	2.09
pg	92.02	-1.55	2.84	2.99
ba	-106.48	4.31	1.06	0.90
hd	308.76	-0.21	3.97	3.49
ko	-147.94	6.17	3.72	3.42
sbc	16.85	-4.46	4.16	3.84
c	315.87	4.94	7.29	7.58
hon	19.73	2.42	0.74	0.80
mcd	-53.91	1.11	1.02	0.99
t	-95.52	4.03	1.39	1.87
cat	-78.01	5.45	0.10	0.52
hwp	-171.02	6.91	0.90	1.16
mmm	49.67	5.36	1.35	1.35
utx	-38.11	4.94	0.70	0.88
dd	-135.28	0.61	1.41	1.29
ibm	24.59	5.86	6.34	6.08
mo	148.31	1.11	2.94	2.90
wmt	11.30	0.79	7.47	7.49
dis	12.05	-3.14	1.00	1.23
intc	104.94	-2.19	6.17	6.16
mrk	146.98	5.14	4.15	3.90
xom	235.86	4.23	8.35	7.85
ek	-156.42	2.32	0.32	0.25
ip	-114.51	4.80	0.27	0.57

Table 2: DJIA Components – Portfolio Weights

Symbol	Historical Weight	CAPM Weight	Implied Equilibrium Weight	Market Capitalization Weight
aa	223.86%	2.67%	0.88%	0.88%
ge	-65.44%	9.80%	11.62%	11.62%
jnj	-70.08%	6.11%	5.29%	5.29%
msft	3.54%	3.22%	10.41%	10.41%
axp	-15.38%	5.54%	1.39%	1.39%
gm	5.76%	3.44%	0.79%	0.79%
jpm	-213.39%	1.94%	2.09%	2.09%
pg	92.00%	-1.33%	2.99%	2.99%
ba	-111.35%	4.71%	0.90%	0.90%
hd	280.01%	0.11%	3.49%	3.49%
ko	-151.58%	5.70%	3.42%	3.42%
sbc	17.11%	-4.28%	3.84%	3.84%
c	293.90%	5.11%	7.58%	7.58%
hon	15.65%	2.71%	0.80%	0.80%
mcd	-61.68%	1.32%	0.99%	0.99%
t	-86.44%	4.04%	1.87%	1.87%
cat	-70.67%	5.10%	0.52%	0.52%
hwp	-163.02%	6.60%	1.16%	1.16%
mmm	56.84%	4.73%	1.35%	1.35%
utx	-23.80%	4.38%	0.88%	0.88%
dd	-131.99%	1.03%	1.29%	1.29%
ibm	36.92%	5.57%	6.08%	6.08%
mo	136.78%	1.31%	2.90%	2.90%
wmt	21.03%	0.89%	7.49%	7.49%
dis	5.75%	-2.35%	1.23%	1.23%
intc	97.81%	-1.96%	6.16%	6.16%
mrk	144.34%	4.61%	3.90%	3.90%
xom	218.75%	4.10%	7.85%	7.85%
ek	-148.36%	2.04%	0.25%	0.25%
ip	-113.07%	4.76%	0.57%	0.57%

It was not practical to compute a percentage difference between computed weights versus weights provided in the research paper since there was no csv file with all the weights available and inputting each weight manually for comparison purposes would be time consuming. However, a general observation shows that the computed weight values are pretty accurate.

Table 3 consists of some information regarding decision making towards obtaining the relevant views. Hence, table 3 was not replicated in the program.

In order to compute the components of Table 4, we first need to compute the Expected return using the Black Litterman formula:

$$E[R] = \left[(\tau\Sigma)^{-1} + P'\Omega^{-1}P \right]^{-1} \left[(\tau\Sigma)^{-1}\Pi + P'\Omega^{-1}Q \right] \qquad (1)$$

Where:

E[R] =New (posterior) Combined Return Vector (*n x 1* column vector)
$\boldsymbol{\tau}$ = Scalar
$\boldsymbol{\Sigma}$ = Covariance Matrix of Returns (*n x n* matrix)
P = Identifies the assets involved in the views (*k x n* matrix or *1 x n* row vector in the special case of 1 view)
$\boldsymbol{\Omega}$ = Diagonal covariance matrix of error terms in expressed views representing the level of confidence in each view (*k x k* matrix)
$\boldsymbol{\Pi}$ = Implied Equilibrium Return Vector (*n x 1* column vector)
Q = View Vector (*k x 1* column vector)

(**′** indicates the transpose and $^{-1}$ indicates the inverse.)

tau= 0.873 was provided in the research paper. We directly use this value instead of computing tau, since the formula provided to compute tau results in a 3x3 matrix with no information on how to convert the matrix to a scalar value for tau. The covariance matrix was provided, the P matrix was also provided, which reflects the absolute and relative views an investor has towards the portfolio. The expressed views in the column vector Q are matched to specific assets by matrix P. Each view results in a 1xn row vector. Thus, k views result in a kxn matrix. Hence, in the three-view DJIA example, P is a 3x30 matrix:

```
> Pmatrix
     V1      V2 V3 V4 V5       V6 V7 V8 V9     V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20 V21 V22 V23     V24 V25 V26 V27      V28 V29 V30
[1,]  0 0.00000  0  0  0  0.00000  0  0  0 0.00000   0   0   0   0   0   0   0   0   0   0   0   0   0  0.0000   0   0   1  0.00000   0   0
[2,]  0 0.00000  1  0  0  0.00000  0 -1  0 0.00000   0   0   0   0   0   0   0   0   0   0   0   0   0  0.0000   0   0   0  0.00000   0   0
[3,]  0 0.76907  0  0  0 -0.04888  0  0  0 0.23092   0   0   0   0   0   0   0   0   0   0   0   0   0 -0.4644   0   0   0 -0.48672   0   0
```

In order to compute Omega, we first needed to compute LC and CF:

There were 3 investor views provided in the paper:

View 1: Merck (mrk) will have an absolute return of 10% (Confidence of View = 50%).
View 2: Johnson & Johnson (jnj) will outperform Procter & Gamble (pg) by 3% (Confidence of View = 65%).
View 3: General Electric (ge) and Home Depot (hd) will outperform General Motors (gm), Wal-Mart (wmt) and Exxon (xom) by 1.5% (Confidence of View = 30%).

LC is the Level of Confidence factor towards each view, which was provided as a percentage. Hence, LC is a 3x1 matrix with the percentage values divided by 100:

```
> LC
     [,1]
[1,] 0.50
[2,] 0.65
[3,] 0.30
```

CF is the calibration factor, which was computed using the formula:

General Case:

$$CF = \frac{P\Sigma P'}{\frac{1}{50\%}}$$

Thus, CF was found to be 0.1407, which is pretty close to the value 0.1403 that was observed in the research paper.

Omega was then computed using the formula given below:

$$\boldsymbol{1/LC * CF}.$$

such that each diagonal element of Omega corresponds to each value in the LC matrix. Thus, Omega matrix takes the form:

DJIA Example:

$$\Omega = \begin{bmatrix} \left(\frac{1}{.50} * CF\right) & 0 & 0 \\ 0 & \left(\frac{1}{.65} * CF\right) & 0 \\ 0 & 0 & \left(\frac{1}{.30} * CF\right) \end{bmatrix}$$

Hence, the Omega matrix was found to be:

```
> Omega
          [,1]      [,2]      [,3]
[1,] 0.2814142 0.0000000 0.0000000
[2,] 0.0000000 0.2164725 0.0000000
[3,] 0.0000000 0.0000000 0.4690237
```

The Q matrix was provided as a 3x1 matrix since there were 3 views. The views take up the form:

DJIA Example:

$$Q+\varepsilon=\begin{bmatrix}10\\3\\1.5\end{bmatrix}+\begin{bmatrix}0\\0\\0\end{bmatrix}$$

General Case:

$$Q+\varepsilon=\begin{bmatrix}Q_1\\\vdots\\Q_k\end{bmatrix}+\begin{bmatrix}\varepsilon_1\\\vdots\\\varepsilon_k\end{bmatrix}$$

Thus, plugging these into the Black Litterman Formula, we obtain the New Combined Return Vector E[R]:

```
> NewExpRet
            V1
aa   13.782542
ge   13.529906
jnj   9.937632
msft 20.391101
axp  14.968191
gm   12.775147
jpm  16.428067
pg    7.473624
ba   11.837531
hd   12.412601
ko   10.994786
sbc   8.889863
c    17.002871
hon  14.496227
mcd  10.514998
t    10.722532
cat  11.028851
hwp  14.295229
mmm   8.674056
utx  15.495166
dd   10.969218
ibm  14.645614
mo    6.903925
wmt  12.836610
dis  12.440635
intc 18.713933
mrk   9.428966
xom   7.893497
ek   10.573639
ip   12.999574
```

The second column in table 4 is the Equilibrium Return Vector that was earlier computed. The third column was calculated as the difference between the New Combined Return Vector and the Implied Equilibrium Return Vector.

The fourth column is the new weight, which was computed using the same formula used to compute table 2:

$$w = (\delta\Sigma)^{-1}\mu$$

Where, the Mu vector was the New Combined Return Vector.

The fifth column of table 4 is the Market Capitalization weights, which were provided to us. The sixth column of table 4 is computed as the difference between the new weight and the market capitalization weight. The computed data was modified such that the data output displayed two decimal places, as done in the research paper. The code and output towards table 4 is provided below:

```r
#Table 4
NewExpRet <- solve(solve(tau*covmatrix)+t(Pmatrix)%*%solve(Omega)%*%Pmatrix)%*%(solve(tau*covmatrix) %*%ImpliedEqReturn+t(Pmatrix)%*%solve(Omega)%*%Q)
ReturnDiff <- NewExpRet - ImpliedEqReturn
OmegaHat <- ((solve(delta*covmatrix)%*%(NewExpRet-(riskfree*100))))

#Compute Weights Difference
WeightDiff <- OmegaHat - (MarketCapWeights*100)

table4 <- cbind(NewExpRet, ImpliedEqReturn, ReturnDiff, OmegaHat, MarketCapWeights*100, WeightDiff)
colnames(table4) <- (c("New Return Vector E[R]", "Implied Eq. Vector", "Return Diff.", "New Weight", "Market Cap. Weight", "Weight Diff."))
table4 <- round(table4,2)
```

```
> table4
     New Return Vector E[R] Implied Eq. Vector Return Diff. New Weight Market Cap. Weight Weight Diff.
aa                    13.78              13.81        -0.03       0.88               0.88         0.00
ge                    13.53              13.58        -0.05      10.76              11.62        -0.86
jnj                    9.94               9.77         0.17       6.25               5.29         0.96
msft                  20.39              20.40        -0.01      10.41              10.41         0.00
axp                   14.97              14.92         0.04       1.39               1.39         0.00
gm                    12.78              12.81        -0.03       0.84               0.79         0.05
jpm                   16.43              16.46        -0.03       2.09               2.09         0.00
pg                     7.47               7.56        -0.08       2.03               2.99        -0.96
ba                    11.84              11.78         0.06       0.90               0.90         0.00
hd                    12.41              12.51        -0.10       3.23               3.49        -0.26
ko                    10.99              10.90         0.10       3.42               3.42         0.00
sbc                    8.89               8.79         0.10       3.84               3.84         0.00
c                     17.00              16.97         0.03       7.58               7.58         0.00
hon                   14.50              14.51        -0.02       0.80               0.80         0.00
mcd                   10.51              10.44         0.08       0.99               0.99         0.00
t                     10.72              10.78        -0.05       1.87               1.87         0.00
cat                   11.03              10.93         0.10       0.52               0.52         0.00
hwp                   14.30              14.44        -0.14       1.16               1.16         0.00
mmm                    8.67               8.66         0.02       1.35               1.35         0.00
utx                   15.50              15.48         0.01       0.88               0.88         0.00
dd                    10.97              10.97         0.00       1.29               1.29         0.00
ibm                   14.65              14.66        -0.01       6.08               6.08         0.00
mo                     6.90               6.85         0.05       2.90               2.90         0.00
wmt                   12.84              12.78         0.06       8.01               7.49         0.52
dis                   12.44              12.41         0.03       1.23               1.23         0.00
intc                  18.71              18.70         0.01       6.16               6.16         0.00
mrk                    9.43               9.20         0.23       4.69               3.90         0.79
xom                    7.89               7.86         0.04       8.39               7.85         0.54
ek                    10.57              10.62        -0.04       0.25               0.25         0.00
ip                    13.00              12.95         0.05       0.57               0.57         0.00
```

Comparing the computed data with the data provided in the research paper, we can see that the program computed the vectors accurately, with minor difference.

Table 4: Return Vectors and Resulting Portfolio Weights

Symbol	New Combined Return Vector (*E[R]*)	Implied Equilibrium Return Vector (Π)	Difference (*E[R]* - Π)	New Weight ($\hat{w}$)	Market Capitalization Weight (*w*)	Difference ($\hat{w}$ - *w*)
aa	13.78	13.81	-0.03	0.88%	0.88%	--
ge	13.52	13.57	-0.05	10.77%	11.62%	-0.85%
jnj	9.92	9.75	0.17	6.28%	5.29%	0.99%
msft	20.40	20.41	-0.01	10.41%	10.41%	--
axp	14.98	14.94	0.04	1.39%	1.39%	--
gm	12.80	12.83	-0.03	0.84%	0.79%	0.05%
jpm	16.43	16.46	-0.03	2.09%	2.09%	--
pg	7.47	7.56	-0.09	2.00%	2.99%	-0.99%
ba	11.87	11.81	0.05	0.90%	0.90%	--
hd	12.42	12.52	-0.10	3.23%	3.49%	-0.25%
ko	11.02	10.92	0.09	3.42%	3.42%	--
sbc	8.89	8.79	0.10	3.84%	3.84%	--
c	17.01	16.97	0.03	7.58%	7.58%	--
hon	14.48	14.50	-0.02	0.80%	0.80%	--
mcd	10.52	10.44	0.08	0.99%	0.99%	--
t	10.69	10.74	-0.05	1.87%	1.87%	--
cat	11.02	10.92	0.10	0.52%	0.52%	--
hwp	14.31	14.45	-0.14	1.16%	1.16%	--
mmm	8.67	8.66	0.02	1.35%	1.35%	--
utx	15.49	15.47	0.02	0.88%	0.88%	--
dd	10.98	10.98	0.00	1.29%	1.29%	--
ibm	14.65	14.66	-0.01	6.08%	6.08%	--
mo	6.91	6.86	0.05	2.90%	2.90%	--
wmt	12.83	12.77	0.06	8.00%	7.49%	0.51%
dis	12.44	12.41	0.03	1.23%	1.23%	--
intc	18.72	18.70	0.02	6.16%	6.16%	--
mrk	9.44	9.22	0.22	4.68%	3.90%	0.77%
xom	7.91	7.88	0.04	8.38%	7.85%	0.53%
ek	10.57	10.61	-0.04	0.25%	0.25%	--
ip	12.97	12.92	0.05	0.57%	0.57%	--
			Sum	100.77%	100.00%	0.77%

Once again, for comparison, since we did not have the research paper data as a csv file, the accuracy of every computed column could not be compared as a percentage difference with the data provided in the research paper. However, in order to obtain an idea of the performance, the Difference in new weights that was computed was compared with the data provided in the research paper by manually inputting the information. Since, the computation of the new weights incorporates the use of the New Combined Return Vector, it is a good way to observe the accuracy of the New Combined Return Vector.

Thus, the performance of the program was observed by calculating the difference between the Computed weights difference and the weights difference provided in the research paper.

```
> ComparativeAnalysis
     Deviation from Official Percentage Weights
aa                                         0.00
ge                                         0.01
jnj                                        0.03
msft                                       0.00
axp                                        0.00
gm                                         0.00
jpm                                        0.00
pg                                        -0.03
ba                                         0.00
hd                                         0.01
ko                                         0.00
sbc                                        0.00
c                                          0.00
hon                                        0.00
mcd                                        0.00
t                                          0.00
cat                                        0.00
hwp                                        0.00
mmm                                        0.00
utx                                        0.00
dd                                         0.00
ibm                                        0.00
mo                                         0.00
wmt                                       -0.01
dis                                        0.00
intc                                       0.00
mrk                                       -0.02
xom                                       -0.01
ek                                         0.00
ip                                         0.00
```

From this result, we can see that the program has provided extremely accurate results with minor deviations up to a maximum deviation of ± 0.03. It is unclear whether the author of the research paper rounded any data prior to using it towards further computation. The R program developed did not round any data prior to computation and used rounding only for display purposes. Additionally, it is unclear whether the author of the research paper used an optimizer to compute weights or used the alternative method incorporated in this program, as described in Appendix B(7). Hence, these considerations could have attributed toward the minor difference in observed values.

The code for the entire program is given below:

```r
#Upload available information as CSV files
table1 <- read.csv(file = "Table1.csv", header = TRUE)
covmatrix <- data.frame(read.csv(file = "Covariance.csv", header = TRUE))
row.names(covmatrix) <- covmatrix[,1] #Set rownames from first column
covmatrix <- covmatrix[,-1] #Delete first column containing stock names
covmatrix <- as.matrix(covmatrix)
Pmatrix <- data.matrix(read.csv(file = "PMatrix.csv", header = FALSE))
MarketCapWeights <- data.matrix(read.csv(file = "MarketCapWeights.csv", header = FALSE)) #value*100 for percentage

delta =2.25 #Risk aversion parameter provided in Appendix B(2)
riskfree= 0.05 #Risk free rate of 5% provided

#Compute Implied Equilibrium Returns to plug into E[R]
ImpliedEqReturn = delta*(covmatrix%*%MarketCapWeights)
ImpliedEqReturn = (ImpliedEqReturn + riskfree)*100 #Factoring in risk free rate and converting to percent
#Compare Computed Implied Equilibrium Returns with Data provided in research paper
ImpliedEqAccuracy <- abs(ImpliedEqReturn-table1[,4])/((ImpliedEqReturn+table1[,4])/2)
colnames(ImpliedEqAccuracy) <- c("Percentage Difference (%)")
#Round off small percentages to 0
ImpliedEqAccuracy <- round(ImpliedEqAccuracy,4)

#Set View vector and Tau
Q <- matrix(c(10,3,1.5))
LC <- matrix(c(0.5,0.65,0.3))
CF= sum(Pmatrix%*%covmatrix%*%t(Pmatrix)*0.5) #Provided in Page 9

#Compute Omega Matrix
Omega <- matrix(0,nrow = 3, ncol = 3)
for (i in 1:3) {
Omega[i,i] <- matrix((1/LC[i,1])*CF)
}

#Compute tau: Actual formula results in 3x3 matrix
#tau = Pmatrix%*%covmatrix%*%t(Pmatrix)/sum(Omega/3)
#Professor advised to directly use tau value provided since there is no information on how to convert to scalar
tau = 0.873
```

```r

#Table 2
table2 <- matrix(0,nrow = 30, ncol = 4)
rownames(table2) <- table1[,1]
#Compute Weights using formula in Appendix B(7): w = (δΣ)−1 μ
for (j in 1:3) {
   table2[,j] <- ((solve(delta*covmatrix)%*%(table1[,j+1]-(riskfree*100))))
}
table2[,4] <- cbind(MarketCapWeights*100)
colnames(table2) <- c("Historical Weight (%)","CAPM Weight (%)", "Implied Eq. Weight (%)", "Market Cap. Weight (%)")
#Round off data to 2 decimal places
table2 <- round(table2,2)

#Table 4
NewExpRet <- solve(solve(tau*covmatrix)+t(Pmatrix)%*%solve(Omega)%*%Pmatrix)%*%(solve(tau*covmatrix) %*%ImpliedEqReturn+t(Pmatrix)%*%solve(Omega)%*%Q)
ReturnDiff <- NewExpRet - ImpliedEqReturn
OmegaHat <- ((solve(delta*covmatrix)%*%(NewExpRet-(riskfree*100))))

#Compute Weights Difference
WeightDiff <- OmegaHat - (MarketCapWeights*100)

table4 <- cbind(NewExpRet, ImpliedEqReturn, ReturnDiff, OmegaHat, MarketCapWeights*100, WeightDiff)
colnames(table4) <- (c("New Return Vector E[R]", "Implied Eq. Vector", "Return Diff.", "New Weight", "Market Cap. Weight", "Weight Diff."))
table4 <- round(table4,2)

#Comparative Analysis
OfficialWeightDiff <- c(0,-0.85,0.99,0,0,0.05,0,-0.99,0,-0.25,0,0,0,0,0,0,0,0,0,0,0,0,0,0.51,0,0,0.77,0.53,0,0)
ComparativeAnalysis <- (OfficialWeightDiff-WeightDiff)
colnames(ComparativeAnalysis) <- c("Deviation from Official Percentage Weights")
```