

```

> # Markowitz Model : Minimum Variance Portfolio and Minimum Variance
Line
> #install.packages("readxl")
> #install.packages("matlib")
> #Select CRAN - UK(London)

> #Libraries used
> library(readxl)
> library(matlib)

> #Reading Excel files and converting into matrix
> A1 <-
read_excel("C:/Users/priya/Desktop/Coding/Project/Portfolio/stockdata.xls
x")
> A<-as.matrix(A1)
> T = nrow(A) #Number of weeks
> N = ncol(A) #Number of assets

> #Creating Unit Vector u
> u = matrix(1,1,N)

> #Creating Expectation Vector m
> m = matrix(0,1,N)
> for(j in 1:N){
+ sum=0
+ for(i in 1:T){
+ sum = sum+A[i,j]
+ }
+ m[1,j] = sum/T
+ }

> #Creating Covariance Matrix c
> c = matrix(0,N,N)
> for(i in 1:N){
+ for(j in 1:N){
+ c[i,j] = cov(A[,i],A[,j])
+ }
+ }

> #precalculating for faster runtime
> ic = inv(c)
> p = m%*%ic%*%t(m)
> q = u%*%ic%*%t(m)
> r = m%*%ic%*%t(u)
> s = u%*%ic%*%t(u)
> M = matrix(c(p,q,r,s),2,2)
> d = det(M)
> iM = inv(M)

> #Calculating Weight Vector w
> w = u%*%ic/(s)[1,1]
> t(w) #Showing weight vector in column form
      [,1]
[1,] 0.0243566620

```

[2,] -0.0044270332
[3,] -0.0335702868
[4,] -0.0303228009
[5,] -0.0441272608
[6,] 0.0065446941
[7,] 0.0105525793
[8,] 0.0012463628
[9,] 0.0318452979
[10,] -0.0121803050
[11,] -0.0286437661
[12,] -0.0332489669
[13,] 0.0320382760
[14,] -0.0264416111
[15,] 0.0120607412
[16,] 0.0083295225
[17,] 0.0179366697
[18,] -0.0652738850
[19,] -0.0645395856
[20,] -0.0091513382
[21,] 0.1306528883
[22,] 0.0585871614
[23,] 0.1148090302
[24,] -0.0032177814
[25,] 0.0534614238
[26,] -0.0142151370
[27,] 0.0242424399
[28,] 0.0187299613
[29,] -0.0054426368
[30,] -0.0926940834
[31,] 0.0061514431
[32,] 0.0265009623
[33,] 0.1862419081
[34,] 0.0027285102
[35,] 0.0138284847
[36,] -0.0888200544
[37,] 0.0108172157
[38,] -0.0067138674
[39,] 0.0379099595
[40,] -0.0055923978
[41,] 0.0611113954
[42,] -0.0140887970
[43,] 0.0117090110
[44,] 0.0502736106
[45,] -0.1033725415
[46,] -0.0233200002
[47,] 0.0911662336
[48,] 0.0520427161
[49,] -0.0207778495
[50,] 0.0511867102
[51,] 0.0231554688
[52,] -0.0169294702
[53,] 0.0629559571
[54,] -0.0310305925
[55,] 0.0777064520
[56,] 0.0012725254
[57,] -0.0056713789
[58,] -0.0017550681
[59,] 0.0003563647
[60,] 0.0200071184
[61,] 0.0651279670

```

[62,] 0.0072060278
[63,] 0.0514505271
[64,] 0.0490919485
[65,] -0.0123118731
[66,] 0.1305779988
[67,] 0.1781652578
[68,] 0.0351650349
[69,] -0.0253761158
[70,] 0.0161198800
[71,] -0.0364316859
[72,] -0.0494487846
[73,] -0.0070731233
[74,] 0.0379258958
[75,] -0.0327905165
[76,] 0.0159593142
[77,] 0.0065318997
[78,] 0.0217801318
[79,] 0.0114202592
[80,] -0.0166455624
[81,] -0.0137823807
[82,] 0.0024704668
[83,] -0.0088650553
[84,] 0.0093935111
[85,] -0.0174558021
[86,] 0.0132988001
[87,] 0.0215487176
> rowSums(w) #checking if weight sums are 1
[1] 1

> #Associated Risk and Returns
> returns = w%*%t(m)
> risk = w%*%c%*%t(w)
> returns
      [,1]
[1,] 0.0009568303
> risk
      [,1]
[1,] 8.921528e-05

> #For a given level of returns mu, calculating weights on MVL
> mu = 0.10
> wmv1 = mu*((s[1,1]*m%*%ic-r[1,1]*u%*%ic)/d) + (p[1,1]*u%*%ic-
q[1,1]*m%*%ic)/d
> t(wmv1)#Showing weight vector in column form
      [,1]
[1,] -1.4470984228
[2,] 1.0899476018
[3,] 0.1287137968
[4,] 0.0295020476
[5,] -0.3115786633
[6,] -0.2456921191
[7,] 0.1453006846
[8,] -0.4841253669
[9,] -0.4959702664
[10,] -0.5200209230
[11,] 0.7979398892
[12,] 0.6031095945
[13,] -0.2791914258

```

[14,] 0.5508494311
[15,] 0.2021002318
[16,] -0.0480791213
[17,] 1.6482077055
[18,] -1.1787429552
[19,] 0.4198838832
[20,] 0.2980894401
[21,] -0.8476883347
[22,] -1.2141961158
[23,] -1.4375228825
[24,] -0.0004151066
[25,] 0.6068040534
[26,] -0.0709488144
[27,] 0.3022893279
[28,] -0.4581018133
[29,] 0.1890578156
[30,] 1.0596000006
[31,] 0.0764900626
[32,] 0.0197454842
[33,] 1.9985944848
[34,] -0.1014329089
[35,] -0.0709344584
[36,] 0.5942171829
[37,] -0.2907703225
[38,] 1.3385628303
[39,] -0.9820779911
[40,] 0.4794740485
[41,] -0.4536910420
[42,] -0.5856285819
[43,] 0.9479929665
[44,] -0.7831984949
[45,] -1.3035712720
[46,] 0.6895881308
[47,] -0.2234463506
[48,] 1.3432032161
[49,] 0.2925246967
[50,] -0.5178500986
[51,] -1.1767590502
[52,] -0.2315874917
[53,] -0.1076124191
[54,] -0.3312724824
[55,] -0.2184513327
[56,] 0.9093253940
[57,] -0.3283227595
[58,] 1.1157241203
[59,] 0.0951282264
[60,] -0.4656106794
[61,] -0.5787740729
[62,] -0.7503952723
[63,] 1.3196828423
[64,] 0.2493405000
[65,] 0.1000762381
[66,] 0.1416668013
[67,] -0.2273525755
[68,] -0.3864029207
[69,] 1.0809322072
[70,] 0.4936652002
[71,] -0.8819471880
[72,] 0.8020897000
[73,] -0.7940234766

```

[74,] 0.1670302962
[75,] -0.2388392568
[76,] -0.4646610968
[77,] -0.1349846151
[78,] -0.5223214898
[79,] 0.2893920593
[80,] -0.0934290538
[81,] 0.1986619871
[82,] 1.2737439138
[83,] 0.2915144522
[84,] -0.3798397502
[85,] -0.3393291956
[86,] -0.4760503780
[87,] 0.1001778635
> rowSums(wmvl) #checking if weight sums are 1
[1] 1

> #Associated Risk and Returns
> returnsmvl = wmvl%*%t(m)
> riskmvl = wmvl%*%c%*%t(w)
> returnsmvl
      [,1]
[1,] 0.1
> riskmvl
      [,1]
[1,] 8.921544e-05

> #END OF FILE

```