Portfolio optimization

by: Ekapope Viriyakovithya , James Ponce, Deborah KEWON , Saeed Mirzaee, Usman Muhammad

Introduction

In this study, we aimed to create a model using AMPL to propose investors to manage their stock portfolio effectively and invested on the most profitable stocks. Having said that, we collected data from IEX Group (https://iextrading.com/developer/docs/#getting-started) and tried to optimize asset allocation strategies to make recommendations for investors.

Data description

Firstly, we randomly retrieved 500 stocks from the total 8,718 stocks in the U.S. market with the period of 5 years (1258 trading days). Each row expressed the closing price of stock for a certain trading date. However, using student version of AMPL software, we are allowed to compute with maximum 300 variables for non-linear problem (500 for linear problem). Therefore, we omitted 200 stocks in the second step. We aggregated stock prices by mean and grouped by month. Then calculated average percentage changes, covariance, and used as an input for AMPL.

Model description

In this study, we used two models; minimum covariance and Young's minimax which are described below.

Minimum covariance model

A minimum covariance portfolio indicates a well-diversified portfolio that consists of individually risky assets, which are hedged when traded together, resulting in the lowest possible risk for the rate of expected return and it can be calculated based on the below:

Objective function:

$$\sum_{i=1}^{N} c_{ij} w_i w_j$$

Subject to:

$$\sum_{i=1}^{N} r_i w_i \geq \propto$$

$$\sum_{i=1}^{N} w_i = 1$$

$$w_i \geq 0$$

$$w_i \leq u$$

Young's minimax model

The essence of this model lies in the minimax formulation of game theory, so the objective function is to maximize the minimum returns of the portfolio subject to constraints.

Objective function:

$$\max_{M_p, w} M_p$$

Subject to:

$$\sum_{j=1}^{N} w_j y_{jt} - M_p \geq 0, \quad t = 1...T$$

$$\sum_{j=1}^{N} w_j \bar{y}_j \geq G$$

$$\sum_{i=1}^{N} w_j \leq W$$

$$0 \leq w_j \leq u, \quad j = 1...N$$

The target of the objective function is the maximization of the portfolio's minimum returns (*Mp*).

w_j : Optimal allocation

M_p : Portfolio's minimum returns

y_{jt} : Historic monthly returns of the shares

$\bar{y}_j$: expected returns of the assets

W : Investor's budget

G : Target return

N : Number of the assets

u : Upper bound of optimal allocations

T : Time period e.g. months

Implementation of AMPL

Mean and Covariance

For the implementation of the model in AMPL, this standard mean-covariance model, includes three constraint. The first one is regarding to return of investment, the second one refers to the budget and the last one is the upper and lower bound constraint. The objective of this model is to minimize the variance. The information required to implement the model is as follows.

Parameters:

- Number of stocks (N).
- Covariance matrix (c).
- Matrix of Mean return (r).
- Upper limit for investing in a single share (u).
- Return of investment (alpha).

Variables:

- Matrix of Weight of the stocks (w).

Constraints:

- Return of investment
- Budget constraint
- Upper and lower bound constraint.

PortOpt.mod

```
reset;
option solver minos;

param N;
param c {1..N, 1..N};
param r {1..N};
param u; # Upper limit for investing in a single share
# required return that we want
param alpha;

var w {1..N} >= 0;

# Objective function

minimize variance:
sum{i in 1..N, j in 1..N} c[i,j]*w[i]*w[j];

# Return of investment constraint
subject to RequiredReturn:
sum{i in 1..N} r[i]*w[i] >= alpha;

# Budget constraint
subject to Budget:
sum{i in 1..N} w[i] = 1;


# Upper and lower bound constrains

subject to lowerbounds {j in 1..N}:
w[j] >= 0;

subject to upperbounds {j in 1..N}:
w[j] <= u;
```

Results

We run the model with return of investment (alpha) equal to 0.03, 0.04 and 0.05, the results obtained are as follow.

For target return (alpha) equal 0.03

- The minimum variance obtained is 0.0011, this value recommends to allocate the investment as follows: **24% in stock number 29**, 9% in stock number 76, 0.9% in stock number 84, 0.7% in stock number 95, 2% in stock number 101, 3% in stock number 135, **12% in stock number 187**, 1.5% in stock number 198, 0.7% in stock number 240, 5.4% in stock number 242, 7.5% in stock number 243,**17% in stock number 267**, 8% in stock number 279, 5% in stock number 283.
- Notice that in this case the model allocated the investment in **14 stocks**.

```
AMPL
ampl: model PortOpt.mod;
ampl: data PortOpt.dat;
MINOS 5.51: optimal solution found.
34 iterations, objective 0.001131495822
Nonlin evals: obj = 74, grad = 73.
```

```
AMPL
ampl: model PortOpt.mod;
ampl: data PortOpt.dat;
MINOS 5.51: optimal solution found.
34 iterations, objective 0.001131495822
Nonlin evals: obj = 74, grad = 73.
```

For target return (alpha) equal 0.04

- The minimum variance obtained is 0.006, this value recommends to allocate the investment as follows: 9% in stock number 135, **24% in stock number 187**, 5% in stock number 198, 3% in stock number 242, **20% in stock number 243**, **25% in stock number 279** and 10% in stock number 283.
- Notice that in this case the model allocated the investment in **7 stocks**.

```
AMPL
ampl: model PortOpt.mod;
ampl: data PortOpt.dat;
MINOS 5.51: optimal solution found.
16 iterations, objective 0.006113098675
Nonlin evals: obj = 31, grad = 30.
```

For target return (alpha) equal 0.05

- The minimum variance obtained is 0.054, this value recommends to allocate the investment as follows: **33% in stock number 135**, **27% in stock number 243**, **24% in stock number 279** and 14.8% in stock number 283.
- Notice that in this case the model allocated the investment in **4 stocks**.

```
AMPL
ampl: model PortOpt.mod;
ampl: data PortOpt.dat;
MINOS 5.51: optimal solution found.
10 iterations, objective 0.0546555031
Nonlin evals: obj = 13, grad = 12.
```

Based on the obtained results, we can clearly see the negative relationship between the risk (variance), and the diversification of the portfolio which is represented by the number of stocks where the model allocates the investment. The other words, the more diversity in portfolio the less risk, it works in the opposite way as well.

In the other hand, we notice the negative relationship among the target return and the diversity of the portfolio. In other words, the more target return expected the less diversity in the portfolio.

Minimax

In addition to Markowitz's mean-covariance analysis for the best combination of expected return and risk, Young introduced the Minimax model, which is based on the minimax decision rule in the game theory. The objective of this model is to maximize the minimum returns subject to constraints.

Our model file contains the definition of the parameters, variables and formulations:

n = number of stocks

T = number of months

W = budget

RetMat = past monthly returns for 300 selected shares

u = Upper limit for investing in a single share

G = target return of the portfolio

Mp = minimum portfolio

ExpRet = expected returns of stocks

stdv = standard deviation of stocks

Our objective function, which maximizes the minimum return is as below.

It is the result of subtraction between standard deviation of stock[j] and expected return of stock[j].

```
# objective function
maximize MinimumReturn:
sum {j in 1..n} w[j]*Mp[j];
```

In order to calculate the objective function, MP (minimum portfolio) should be first defined. Mp can be calculated using the following functions:

```
let {j in 1..n} ExpRet[j] := sum{i in 1..T} RetMat[i,j]/T;
let {j in 1..n} stdv[j] := sqrt((sum{i in 1..T} (RetMat[i,j] -ExpRet[j])^2)/T);
let {j in 1..n} Mp[j] := ExpRet[j]-stdv[j];
```

Expected return is the average of the past returns for 60 months.

We applied 4 constraints 1) the sum of expected return should be equal to or greater than our target return 2) investment(asset) allocation cannot be greater than our budget 3) lower bound of allocation is equal to or greater than 0 4) upper bound of allocation is equal to or less than limit for investing in a single share.

As for the budget, we used the percentage because percentage well reflects the change in a single stock even taking the scale into consideration. Using an actual amount for the budget can be very confusing as we do not know the actual value of each stock. The prices of certain stocks can be a lot higher than the prices of others. Therefore, we defined budget =1 (100%), so in that way we can easily compare the allocation %.

Results

1. With the 3% target return, we obtained -0.05 minimum return, which means the lowest % of return we can get is -0.048. Based on our input, we are advised to allocate 50% for stock 187, 46.7% for stock 101, and 3.3% for stock 279 (this allocation % is by month).
2. With the 4% target return, we obtained -0.13 minimum return. We are advised to allocate 50% for stock 243, 8.5% for stock 135 and 41.5% for stock 279.
3. In order to test the limit of target return, we also set the target return equal to 5%. With this input, we obtained -0.27 minimum return. We are advised to invest 50% of our budget for stock 243, 33.2% for stock 135 and 16.8% for stock 279.

Findings

1. Stock 135, 243 and 279 are highly recommended for investment
2. The higher the target return is, the riskier it is to invest

ex) when the target return increased from 3% to 5%, the minimum return also increased from - 0.048 to -0.27

Next steps/challenges/ recommendation/Conclusion

The proposed financial optimization model assists the investor to make the best decision for asset allocations in finance in order to both ensure his investments and have a satisfactory return of his portfolio. It is well-known that algebraic modelling languages are ideal tools for rapid prototyping and optimization model development.

Thus, our proposed portfolio optimization model utilizes the flexibility and convenience of the AMPL modelling language. A strong point of the proposed work is the variety of the state-of-the-art portfolio optimization models which aim is to advice the investor about the optimal asset allocation.

Furthermore, the code of this model can be easily extended or modified, since the majority of the operational and financial researchers are rather more familiar with mathematical modelling languages than with common programming languages.

Suggestions

For future implementation, we can include the concept of complementary stocks as a constraint. For this improvement its necessary to work previously with the dataset in order to include a new feature “the relationship between stocks” (positive or negative).

Appendix

Young’s minimax model when G (target return)=0.03

```
MINOS 5.51: optimal solution found.
14 iterations, objective -0.0479783132
ampl: display w;
w [*] :=
  1 0      61 0     121 0     181 0     241 0
  2 0      62 0     122 0     182 0     242 0
  3 0      63 0     123 0     183 0     243 0
  4 0      64 0     124 0     184 0     244 0
  5 0      65 0     125 0     185 0     245 0
  6 0      66 0     126 0     186 0     246 0
  7 0      67 0     127 0     187 0.5   247 0
  8 0      68 0     128 0     188 0     248 0
  9 0      69 0     129 0     189 0     249 0
 10 0      70 0     130 0     190 0     250 0
 11 0      71 0     131 0     191 0     251 0
 12 0      72 0     132 0     192 0     252 0
 13 0      73 0     133 0     193 0     253 0
 14 0      74 0     134 0     194 0     254 0
 15 0      75 0     135 0     195 0     255 0
 16 0      76 0     136 0     196 0     256 0
 17 0      77 0     137 0     197 0     257 0
 18 0      78 0     138 0     198 0     258 0
 19 0      79 0     139 0     199 0     259 0
 20 0      80 0     140 0     200 0     260 0
 21 0      81 0     141 0     201 0     261 0
 22 0      82 0     142 0     202 0     262 0
 23 0      83 0     143 0     203 0     263 0
 24 0      84 0     144 0     204 0     264 0
 25 0      85 0     145 0     205 0     265 0
 26 0      86 0     146 0     206 0     266 0
 27 0      87 0     147 0     207 0     267 0
 28 0      88 0     148 0     208 0     268 0
 29 0      89 0     149 0     209 0     269 0
 30 0      90 0     150 0     210 0     270 0
 31 0      91 0     151 0     211 0     271 0
 32 0      92 0     152 0     212 0     272 0
 33 0      93 0     153 0     213 0     273 0
 34 0      94 0     154 0     214 0     274 0
 35 0      95 0     155 0     215 0     275 0
 36 0      96 0     156 0     216 0     276 0
 37 0      97 0     157 0     217 0     277 0
 38 0      98 0     158 0     218 0     278 0
 39 0      99 0     159 0     219 0     279 0.0330977
 40 0     100 0     160 0     220 0     280 0
 41 0     101 0.466902  161 0 221 0     281 0
 42 0     102 0     162 0     222 0     282 0
 43 0     103 0     163 0     223 0     283 0
 44 0     104 0     164 0     224 0     284 0
 45 0     105 0     165 0     225 0     285 0
 46 0     106 0     166 0     226 0     286 0
 47 0     107 0     167 0     227 0     287 0
 48 0     108 0     168 0     228 0     288 0
 49 0     109 0     169 0     229 0     289 0
 50 0     110 0     170 0     230 0     290 0
 51 0     111 0     171 0     231 0     291 0
 52 0     112 0     172 0     232 0     292 0
 53 0     113 0     173 0     233 0     293 0
 54 0     114 0     174 0     234 0     294 0
 55 0     115 0     175 0     235 0     295 0
 56 0     116 0     176 0     236 0     296 0
 57 0     117 0     177 0     237 0     297 0
 58 0     118 0     178 0     238 0     298 0
 59 0     119 0     179 0     239 0     299 0
 60 0     120 0     180 0     240 0     300 0
;
```

Yong’s minimax model when G (target return)=0.04

```
MINOS 5.51: optimal solution found.
9 iterations, objective -0.1291259123

"option abs_boundtol 5.551115123125783e-17;"
or "option rel_boundtol 1.1102230246251565e-16;"
will change deduced dual values.

ampl: display w;
w [*] :=
  1 0            61 0            121 0            181 0            241 0
  2 0            62 0            122 0            182 0            242 0
  3 0            63 0            123 0            183 0            243 0.5
  4 0            64 0            124 0            184 0            244 0
  5 0            65 0            125 0            185 0            245 0
  6 0            66 0            126 0            186 0            246 0
  7 0            67 0            127 0            187 0            247 0
  8 0            68 0            128 0            188 0            248 0
  9 0            69 0            129 0            189 0            249 0
 10 0            70 0            130 0            190 0            250 0
 11 0            71 0            131 0            191 0            251 0
 12 0            72 0            132 0            192 0            252 0
 13 0            73 0            133 0            193 0            253 0
 14 0            74 0            134 0            194 0            254 0
 15 0            75 0            135 0.0849866    195 0            255 0
 16 0            76 0            136 0            196 0            256 0
 17 0            77 0            137 0            197 0            257 0
 18 0            78 0            138 0            198 0            258 0
 19 0            79 0            139 0            199 0            259 0
 20 0            80 0            140 0            200 0            260 0
 21 0            81 0            141 0            201 0            261 0
 22 0            82 0            142 0            202 0            262 0
 23 0            83 0            143 0            203 0            263 0
 24 0            84 0            144 0            204 0            264 0
 25 0            85 0            145 0            205 0            265 0
 26 0            86 0            146 0            206 0            266 0
 27 0            87 0            147 0            207 0            267 0
 28 0            88 0            148 0            208 0            268 0
 29 0            89 0            149 0            209 0            269 0
 30 0            90 0            150 0            210 0            270 0
 31 0            91 0            151 0            211 0            271 0
 32 0            92 0            152 0            212 0            272 0
 33 0            93 0            153 0            213 0            273 0
 34 0            94 0            154 0            214 0            274 0
 35 0            95 0            155 0            215 0            275 0
 36 0            96 0            156 0            216 0            276 0
 37 0            97 0            157 0            217 0            277 0
 38 0            98 0            158 0            218 0            278 0
 39 0            99 0            159 0            219 0            279 0.415013
 40 0           100 0            160 0            220 0            280 0
 41 0           101 0            161 0            221 0            281 0
 42 0           102 0            162 0            222 0            282 0
 43 0           103 0            163 0            223 0            283 0
 44 0           104 0            164 0            224 0            284 0
 45 0           105 0            165 0            225 0            285 0
 46 0           106 0            166 0            226 0            286 0
 47 0           107 0            167 0            227 0            287 0
 48 0           108 0            168 0            228 0            288 0
 49 0           109 0            169 0            229 0            289 0
 50 0           110 0            170 0            230 0            290 0
 51 0           111 0            171 0            231 0            291 0
 52 0           112 0            172 0            232 0            292 0
 53 0           113 0            173 0            233 0            293 0
 54 0           114 0            174 0            234 0            294 0
 55 0           115 0            175 0            235 0            295 0
 56 0           116 0            176 0            236 0            296 0
 57 0           117 0            177 0            237 0            297 0
 58 0           118 0            178 0            238 0            298 0
 59 0           119 0            179 0            239 0            299 0
 60 0           120 0            180 0            240 0            300 0
;
```

Young’s minimax model when G (target return)=0.05

```
ampl: solve;
MINOS 5.51: optimal solution found.
10 iterations, objective -0.2698254011
ampl: display w;
w [*] :=
  1 0          61 0         121 0          181 0          241 0
  2 0          62 0         122 0          182 0          242 0
  3 0          63 0         123 0          183 0          243 0.5
  4 0          64 0         124 0          184 0          244 0
  5 0          65 0         125 0          185 0          245 0
  6 0          66 0         126 0          186 0          246 0
  7 0          67 0         127 0          187 0          247 0
  8 0          68 0         128 0          188 0          248 0
  9 0          69 0         129 0          189 0          249 0
 10 0          70 0         130 0          190 0          250 0
 11 0          71 0         131 0          191 0          251 0
 12 0          72 0         132 0          192 0          252 0
 13 0          73 0         133 0          193 0          253 0
 14 0          74 0         134 0          194 0          254 0
 15 0          75 0         135 0.332497   195 0          255 0
 16 0          76 0         136 0          196 0          256 0
 17 0          77 0         137 0          197 0          257 0
 18 0          78 0         138 0          198 0          258 0
 19 0          79 0         139 0          199 0          259 0
 20 0          80 0         140 0          200 0          260 0
 21 0          81 0         141 0          201 0          261 0
 22 0          82 0         142 0          202 0          262 0
 23 0          83 0         143 0          203 0          263 0
 24 0          84 0         144 0          204 0          264 0
 25 0          85 0         145 0          205 0          265 0
 26 0          86 0         146 0          206 0          266 0
 27 0          87 0         147 0          207 0          267 0
 28 0          88 0         148 0          208 0          268 0
 29 0          89 0         149 0          209 0          269 0
 30 0          90 0         150 0          210 0          270 0
 31 0          91 0         151 0          211 0          271 0
 32 0          92 0         152 0          212 0          272 0
 33 0          93 0         153 0          213 0          273 0
 34 0          94 0         154 0          214 0          274 0
 35 0          95 0         155 0          215 0          275 0
 36 0          96 0         156 0          216 0          276 0
 37 0          97 0         157 0          217 0          277 0
 38 0          98 0         158 0          218 0          278 0
 39 0          99 0         159 0          219 0          279 0.167503
 40 0         100 0         160 0          220 0          280 0
 41 0         101 0         161 0          221 0          281 0
 42 0         102 0         162 0          222 0          282 0
 43 0         103 0         163 0          223 0          283 0
 44 0         104 0         164 0          224 0          284 0
 45 0         105 0         165 0          225 0          285 0
 46 0         106 0         166 0          226 0          286 0
 47 0         107 0         167 0          227 0          287 0
 48 0         108 0         168 0          228 0          288 0
 49 0         109 0         169 0          229 0          289 0
 50 0         110 0         170 0          230 0          290 0
 51 0         111 0         171 0          231 0          291 0
 52 0         112 0         172 0          232 0          292 0
 53 0         113 0         173 0          233 0          293 0
 54 0         114 0         174 0          234 0          294 0
 55 0         115 0         175 0          235 0          295 0
 56 0         116 0         176 0          236 0          296 0
 57 0         117 0         177 0          237 0          297 0
 58 0         118 0         178 0          238 0          298 0
 59 0         119 0         179 0          239 0          299 0
 60 0         120 0         180 0          240 0          300 0
;
```

Covariance Model run for alpha = 0.03

Console

AMPL

```
ampl: model PortOpt.mod;
ampl: data PortOpt.dat;
MINOS 5.51: optimal solution found.
34 iterations, objective 0.001131495822
Nonlin evals: obj = 74, grad = 73.
w [*] :=
  1 0           76 0.0926932    151 0            226 0
  2 0           77 0            152 0            227 0
  3 0           78 0            153 0            228 0
  4 0           79 0            154 0            229 0
  5 0           80 0            155 0            230 0
  6 0           81 0            156 0            231 0
  7 0           82 0            157 0            232 0
  8 0           83 0            158 0            233 0
  9 0           84 0.00978407   159 0            234 0
 10 0           85 0            160 0            235 0
 11 0           86 0            161 0            236 0
 12 0           87 0            162 0            237 0
 13 0           88 0            163 0            238 0
 14 0           89 0            164 0            239 0
 15 0           90 0            165 0            240 0.00796595
 16 0           91 0            166 0            241 0
 17 0           92 0            167 0            242 0.0544199
 18 0           93 0            168 0            243 0.0755786
 19 0           94 0            169 0            244 0
 20 0           95 0.00768153   170 0            245 0
 21 0           96 0            171 0            246 0
 22 0           97 0            172 0            247 0
 23 0           98 0            173 0            248 0
 24 0           99 0            174 0            249 0
 25 0          100 0            175 0            250 0
 26 0          101 0.0200756    176 0            251 0
 27 0          102 0            177 0            252 0
 28 0          103 0            178 0            253 0
 29 0.249722   104 0            179 0            254 0
 30 0          105 0            180 0            255 0
 31 0          106 0            181 0            256 0
 32 0          107 0            182 0            257 0
 33 0          108 0            183 0            258 0
 34 0          109 0            184 0            259 0
 35 0          110 0            185 0            260 0
 36 0          111 0            186 0            261 0
 37 0          112 0            187 0.121237     262 0
 38 0          113 0            188 0            263 0
 39 0          114 0            189 0            264 0
 40 0          115 0            190 0            265 0
 41 0          116 0            191 0            266 0
 42 0          117 0            192 0            267 0.179745
 43 0          118 0            193 0            268 0
 44 0          119 0            194 0            269 0
 45 0          120 0            195 0            270 0
 46 0          121 0            196 0            271 0
```

47 0	122 0	197 0	272 0
48 0	123 0	198 0.0159892	273 0
49 0	124 0	199 0	274 0
50 0	125 0	200 0	275 0
51 0	126 0	201 0	276 0
52 0	127 0	202 0	277 0
53 0	128 0	203 0	278 0
54 0	129 0	204 0	279 0.0834729
55 0	130 0	205 0	280 0
56 0	131 0	206 0	281 0
57 0	132 0	207 0	282 0
58 0	133 0	208 0	283 0.0511497
59 0	134 0	209 0	284 0
60 0	135 0.0304855	210 0	285 0
61 0	136 0	211 0	286 0
62 0	137 0	212 0	287 0
63 0	138 0	213 0	288 0
64 0	139 0	214 0	289 0
65 0	140 0	215 0	290 0
66 0	141 0	216 0	291 0
67 0	142 0	217 0	292 0
68 0	143 0	218 0	293 0
69 0	144 0	219 0	294 0
70 0	145 0	220 0	295 0
71 0	146 0	221 0	296 0
72 0	147 0	222 0	297 0
73 0	148 0	223 0	298 0
74 0	149 0	224 0	299 0
75 0	150 0	225 0	300 0

Covariance Model run for alpha = 0.04

Console

AMPL

```
ampl: model PortOpt.mod;
ampl: data PortOpt.dat;
MINOS 5.51: optimal solution found.
16 iterations, objective 0.006113098675
Nonlin evals: obj = 31, grad = 30.
w [*] :=
  1 0          61 0          121 0          181 0          241 0
  2 0          62 0          122 0          182 0          242 0.0383905
  3 0          63 0          123 0          183 0          243 0.207177
  4 0          64 0          124 0          184 0          244 0
  5 0          65 0          125 0          185 0          245 0
  6 0          66 0          126 0          186 0          246 0
  7 0          67 0          127 0          187 0.246647   247 0
  8 0          68 0          128 0          188 0          248 0
  9 0          69 0          129 0          189 0          249 0
 10 0          70 0          130 0          190 0          250 0
 11 0          71 0          131 0          191 0          251 0
 12 0          72 0          132 0          192 0          252 0
 13 0          73 0          133 0          193 0          253 0
 14 0          74 0          134 0          194 0          254 0
 15 0          75 0          135 0.0965431  195 0          255 0
 16 0          76 0          136 0          196 0          256 0
 17 0          77 0          137 0          197 0          257 0
 18 0          78 0          138 0          198 0.0493634  258 0
 19 0          79 0          139 0          199 0          259 0
 20 0          80 0          140 0          200 0          260 0
 21 0          81 0          141 0          201 0          261 0
 22 0          82 0          142 0          202 0          262 0
 23 0          83 0          143 0          203 0          263 0
 24 0          84 0          144 0          204 0          264 0
 25 0          85 0          145 0          205 0          265 0
 26 0          86 0          146 0          206 0          266 0
 27 0          87 0          147 0          207 0          267 0
 28 0          88 0          148 0          208 0          268 0
 29 0          89 0          149 0          209 0          269 0
 30 0          90 0          150 0          210 0          270 0
 31 0          91 0          151 0          211 0          271 0
 32 0          92 0          152 0          212 0          272 0
 33 0          93 0          153 0          213 0          273 0
 34 0          94 0          154 0          214 0          274 0
 35 0          95 0          155 0          215 0          275 0
 36 0          96 0          156 0          216 0          276 0
 37 0          97 0          157 0          217 0          277 0
 38 0          98 0          158 0          218 0          278 0
 39 0          99 0          159 0          219 0          279 0.256693
 40 0         100 0          160 0          220 0          280 0
 41 0         101 0          161 0          221 0          281 0
 42 0         102 0          162 0          222 0          282 0
 43 0         103 0          163 0          223 0          283 0.105186
 44 0         104 0          164 0          224 0          284 0
 45 0         105 0          165 0          225 0          285 0
```

46 0	106 0	166 0	226 0	286 0
47 0	107 0	167 0	227 0	287 0
48 0	108 0	168 0	228 0	288 0
49 0	109 0	169 0	229 0	289 0
50 0	110 0	170 0	230 0	290 0
51 0	111 0	171 0	231 0	291 0
52 0	112 0	172 0	232 0	292 0
53 0	113 0	173 0	233 0	293 0
54 0	114 0	174 0	234 0	294 0
55 0	115 0	175 0	235 0	295 0
56 0	116 0	176 0	236 0	296 0
57 0	117 0	177 0	237 0	297 0
58 0	118 0	178 0	238 0	298 0
59 0	119 0	179 0	239 0	299 0
60 0	120 0	180 0	240 0	300 0

Covariance Model run for alpha = 0.05

Console

AMPL

```
ampl: model PortOpt.mod;
ampl: data PortOpt.dat;
MINOS 5.51: optimal solution found.
10 iterations, objective 0.0546555031
Nonlin evals: obj = 13, grad = 12.
w [*] :=
  1 0           61 0          121 0          181 0          241 0
  2 0           62 0          122 0          182 0          242 0
  3 0           63 0          123 0          183 0          243 0.269262
  4 0           64 0          124 0          184 0          244 0
  5 0           65 0          125 0          185 0          245 0
  6 0           66 0          126 0          186 0          246 0
  7 0           67 0          127 0          187 0          247 0
  8 0           68 0          128 0          188 0          248 0
  9 0           69 0          129 0          189 0          249 0
 10 0           70 0          130 0          190 0          250 0
 11 0           71 0          131 0          191 0          251 0
 12 0           72 0          132 0          192 0          252 0
 13 0           73 0          133 0          193 0          253 0
 14 0           74 0          134 0          194 0          254 0
 15 0           75 0          135 0.333691   195 0          255 0
 16 0           76 0          136 0          196 0          256 0
 17 0           77 0          137 0          197 0          257 0
 18 0           78 0          138 0          198 0          258 0
 19 0           79 0          139 0          199 0          259 0
 20 0           80 0          140 0          200 0          260 0
 21 0           81 0          141 0          201 0          261 0
 22 0           82 0          142 0          202 0          262 0
 23 0           83 0          143 0          203 0          263 0
 24 0           84 0          144 0          204 0          264 0
 25 0           85 0          145 0          205 0          265 0
 26 0           86 0          146 0          206 0          266 0
 27 0           87 0          147 0          207 0          267 0
 28 0           88 0          148 0          208 0          268 0
 29 0           89 0          149 0          209 0          269 0
 30 0           90 0          150 0          210 0          270 0
 31 0           91 0          151 0          211 0          271 0
 32 0           92 0          152 0          212 0          272 0
 33 0           93 0          153 0          213 0          273 0
 34 0           94 0          154 0          214 0          274 0
 35 0           95 0          155 0          215 0          275 0
 36 0           96 0          156 0          216 0          276 0
 37 0           97 0          157 0          217 0          277 0
 38 0           98 0          158 0          218 0          278 0
 39 0           99 0          159 0          219 0          279 0.24888
 40 0          100 0          160 0          220 0          280 0
 41 0          101 0          161 0          221 0          281 0
 42 0          102 0          162 0          222 0          282 0
 43 0          103 0          163 0          223 0          283 0.148166
 44 0          104 0          164 0          224 0          284 0
 45 0          105 0          165 0          225 0          285 0
 46 0          106 0          166 0          226 0          286 0
```

```
47 0        107 0       167 0       227 0       287 0
48 0        108 0       168 0       228 0       288 0
49 0        109 0       169 0       229 0       289 0
50 0        110 0       170 0       230 0       290 0
51 0        111 0       171 0       231 0       291 0
52 0        112 0       172 0       232 0       292 0
53 0        113 0       173 0       233 0       293 0
54 0        114 0       174 0       234 0       294 0
55 0        115 0       175 0       235 0       295 0
56 0        116 0       176 0       236 0       296 0
57 0        117 0       177 0       237 0       297 0
58 0        118 0       178 0       238 0       298 0
59 0        119 0       179 0       239 0       299 0
60 0        120 0       180 0       240 0       300 0
```

Index and Symbol Table (It looks in 4 columns in Word desktop version)

Index	Symbol
1	EWL
2	SKF
3	HMY
4	TOK
5	VECO
6	ALLY.A
7	KBWB
8	TOT
9	LEE
10	IIF
11	VGM
12	IGHG
13	PAA
14	MBG
15	BGT
16	TLK
17	DZZ
18	ALL.D
19	FNK
20	UTL
21	SYNL
22	PAGP
23	NCI
24	VOYA
25	NS
26	PFLT
27	JQC
28	MCY
29	ERIE
30	BYD
31	UCFC
32	SUNW
33	ONCY
34	RJI
35	AFT
36	IGI
37	PSA.W
38	CPL
39	EXG
40	CXO
41	BVN
42	HAO
43	GXC
44	FXN
45	VKI
46	BSJJ
47	CVS
48	HST
49	E
50	SBAC
51	WAFD
52	HI
53	PXMV
54	BID
55	JAZZ
56	CMG
57	BLRX
58	PDM
59	SLAB
60	FTR
61	MLPX
62	AUY
63	USA
64	WBC
65	PTY
66	HEP
67	VCSH
68	TGB
69	PM
70	USAC
71	ORCL
72	LAND
73	MUS
74	ABB
75	HMSY
76	CNC
77	FRAN
78	JXI
79	FANH
80	BAS
81	CPTA
82	MCA
83	GFF
84	OSPN
85	AWSM
86	GPOR
87	SPDW
88	RGT
89	QRTEA
90	CATY
91	BIB
92	KYN
93	ETR
94	SPHB
95	ELMD
96	CECO
97	NTRS
98	ASYS
99	GTXI
100	HON
101	ISRG
102	ENT
103	HELE
104	RST
105	HCCI
106	TEUM
107	PZD
108	SMSI
109	FPA
110	BNDX
111	QDF
112	DVN
113	ALSK
114	REM
115	WFC.R
116	TTGT
117	HSON
118	SPTS
119	IGN
120	FLNT
121	IWX
122	ECH
123	DVAX

124	CHFS
125	ESI
126	RING
127	SCOR
128	ARR.B
129	BHR
130	JPI
131	INWK
132	STAA
133	IXP
134	LB
135	USLV
136	IUSV
137	MRVL
138	MFA
139	PSR
140	VVUS
141	DKL
142	F
143	ABUS
144	VVR
145	MSA
146	RPT
147	CHK.D
148	AOA
149	CHN
150	JPXN
151	MRC
152	GEO
153	EOD
154	CHKR
155	ATI
156	POOL
157	HFWA
158	CDMO
159	FGM
160	PLW
161	GAL
162	WCC

163	IFN
164	EGPT
165	PHK
166	GLV
167	RF.A
168	CCIH
169	ERX
170	CHT
171	HRL
172	BR
173	MEOH
174	IWP
175	IDE
176	HIX
177	TWO
178	GALT
179	VNQ
180	GTIM
181	LAD
182	CPK
183	IJS
184	PSK
185	SNPS
186	FFIN
187	CHGG
188	GHM
189	MET
190	MCF
191	SOHO
192	NVG
193	TAO
194	RVT
195	MIDD
196	EWC
197	EFG
198	HIIQ
199	STML
200	GCC
201	SPTL

202	MIDZ
203	GM
204	PLD
205	PEJ
206	SUNS
207	CYRN
208	NHC
209	BECN
210	WEC
211	ETJ
212	PCEF
213	BSCK
214	HCI
215	AGYS
216	MUC
217	NGS
218	DUC
219	IOVA
220	JWN
221	CEV
222	ODP
223	SYMC
224	VSEC
225	AHT.D
226	CZNC
227	SOXS
228	PRGX
229	TDS
230	IYW
231	FUND
232	MDSO
233	EWO
234	AIV
235	UPW
236	UIHC
237	XONE
238	IDLV
239	ATRI
240	HSII

241	NAN
242	OLED
243	AXGN
244	FRBK
245	STFC
246	GSBC
247	CPSI
248	IWR
249	IWN
250	GPX
251	BTA
252	BKN
253	PYN
254	BVSN
255	LKFN
256	HCFT
257	XOP
258	CORP
259	DPZ
260	WTR
261	VCV
262	RY
263	SHYD
264	NUV
265	SLY
266	EPAY
267	CENT
268	CGA
269	ACAD
270	RRGB
271	TDJ
272	ABC
273	AKAM
274	SNV
275	STAG
276	TRMB
277	OSBC
278	CUK
279	INGN

280	ZROZ
281	TARO
282	ARC
283	SKY
284	BBT.F
285	AGIO

286	PNR
287	QADA
288	ICLN
289	VEA
290	EVM
291	MRO

292	ALL
293	LUB
294	TSCO
295	FLXN
296	FVE
297	ROL

298	HRTX
299	TVC
300	IRM