Advanced Portfolio - Midterm

Prof. Papa Momar Ndiaye

Due: 11:55pm, Wednesday, April 25th, 2018.

Instructions

- **Read these instructions and follow them precisely.**
- **Independence:** All students must work independently.
- **Submission:** Submit your answer document online
- **Answer Document:** Your answer document MUST be in the form of a single pdf file that contains all of your answers including code printouts and graphs. Do not submit your answer document in any format other than pdf. Any answer document that does not comprise a single pdf file complete with all answers will receive a grade of zero.
- **Cover Sheet:** Your answer document must include a cover sheet that states the course name, the homework number, the date, and your name.
- **Legibility and Logical Presentation:** Answer documents that are not easily legible, or not logically presented, or have a non-professional appearance will not be graded.
- **Source Code Requirement:** Your submission should also contain a separate set of source code files for all of your solutions. I may run your source code to ensure that it provides the results that you claim.
- **Permissible Computer Languages:** You can use any matrix-oriented computer programming language (R, Matlab or Python with Pandas, for example), but do not use any spreadsheets. Problems solved with spreadsheets will receive no credit.

The supplied `data.zip` file contains 30 space-delimited text files that contain price and volume data for 30 companies. Each row of each file contains date, opening price, closing price, high price, low price, volume, and adjusted price.

Question 1

Write a program called `processdata.m` or `processdata.R` to:

1. Read all daily price files;
2. Create a price matrix `P` by aligning the data's dates and placing the adjusted closing prices side-by-side in columns;
3. From the `P matrix`, create a matrix of simple (not logarithmic) daily returns `R`;
4. Compute the vector of average daily returns mu for the companies using the `mean` function in **MATLAB** or the `colMeans` function in **R** (do not use loops);
5. Compute the covariance matrix `Q` from the return matrix using the `cov` function; and
6. Save the return vector `mu` and covariance matrix `Q` in the native format for your programming language. For example, if you use **MATLAB**, store the data in .mat format in a file called `inputs.mat`. If you use **R**, store the data in `.Rdata` format in a file called `inputs.rData`.

Question 2

Write a function called `port` that uses standard quadratic programming libraries that will:

- Take the set of input parameters `mu` (mean vector μ), `Q` (covariance matrix Q), and `tau` (risk tolerance τ) and return the vector h that maximizes the following utility function U defined by

  $$U(h) = -\frac{1}{2}h^T Q h + \tau h^T \mu$$

  subject to the constraints

  $$0 \le h_i \le 0.1 \text{ for all } i\text{, and}$$

  $$\sum_{i=1}^{n} h_i = h^T e = 1$$

  where n is the number of securities in the portfolio.

Question 3

Write a program called `frontier.m` or `frontier.R` that will:

1. Load the data in the `inputs.mat` or `inputs.rData` file;
2. Create a sequence `TAU` containing numbers from zero to 0.5 in steps of 0.001;
3. Run through a loop for each value of your `TAU` sequence to
4. Find the optimum portfolio with the given `mu`, `Q`, and `tau` selected from `TAU`;
5. Compute the optimum portfolio's expected return and standard deviation of return; and - Store the portfolio return and standard deviation.
6. After completing the loop, plot the efficient frontier.

The supplied `Midtermdata.zip` file contains an **R** data file `data.rda` and a tab-delimited text file `data.tsv` containing the Dow-Jones Industrial Index and the closing prices for the 30 companies in that index for 250 trading days.

Question 4

Write a program to:

1. read in this Dow-Jones data
2. convert the matrix of daily prices to daily simple returns (not logarithmic returns),
3. annualize the returns by multiplying them by 252 (the typical number of trading days in a year),
4. move the index column out of the matrix and into a separate vector,
5. compute a covariance matrix `Qts` based on the time-series of returns, and
6. print out the first five rows and five columns of the covariance matrix.

Clearly describe all steps in your program with comments. List your program in your answer document. No points will be awarded unless the steps associated with each part of the question are clearly distinguished. Also submit the source code file.

Question 5

Write a program utilizing the Dow-Jones data that:

1. uses a loop to regress each company's returns onto the index returns,
2. prints a table of intercepts, slopes (β_i), and idiosyncratic standard deviations σ_{R_i} for all companies $i = 1, ..., 30$,
3. computes and prints the variance of the index's return
4. computes the single-index approximation to the covariance matrix `Qsi` using your computed σ_M^2, β_i and σ_{R_i} for all i, and
5. prints the first five rows and columns of this covariance matrix. Remember that the diagonal entries of Ω are $\sigma_{R_i}^2$, not σ_{R_i})

Clearly describe all steps in your program with comments. List your program in your answer document. No points will be awarded unless the steps associated with each part of the question are clearly distinguished. Also submit the source code file.

Question 6

With your function `port` from the Question 2:

1. Use the set of input parameters `mu` (mean vector μ), `Qts` (covariance matrix Q computed with a time series approximation), and `tau` (risk tolerance τ) to compute the efficient frontier corresponding to the maximization of the utility function U defined by

$$U(h) = -\frac{1}{2}h^T Q h + \tau h^T \mu$$

   subject to the constraints

$$0 \le h_i \le 0.1 \text{ for all } i, \text{ and}$$

$$\sum_{i=1}^{n} h_i = h^T e = 1$$

   where n is the number of securities in the portfolio. Be sure to include all constraints; both equality and inequality constraints.

2. Repeat the computation of the efficient frontier but now use the approximate covariance matrix `Qsi` computed from the single index model instead of the time-series model `Qts`.

3. Plot the efficient frontier using the time-series approximation to `Qts` in blue, and superimpose the efficient frontier computed with the single index covariance model `Qsi` in red. Do you think the results are similar enough that the single-index covariance model is valid?

   Clearly describe all steps in your program with comments. List your program in your answer document. No points will be awarded unless the steps associated with each part of the question are clearly distinguished. Also submit the source code file.