Homework 9: A Diversified Portfolio

36-402, Spring 2025

Due at 6 pm on Thursday, 27 March 2025

AGENDA: Learning about factor models; more practice with density estimation; still more practice with bootstrapping.

ADVICE: Read Chapter 17 of the text.

Classical financial theory suggests that the log-returns[1] of corporate stocks should be IID Gaussian random variables, but allows for the possibility that the log-returns of different stocks might be correlated with each other. In fact, theory suggests that the log-returns to any given stock should be the sum of two components: one which is specific to that firm, and one which is common to all firms. (More specifically, the common component is one which couldn't be eliminated even in a perfectly diversified portfolio.) This in turn implies that stocks' log-returns should match a one-factor model.

The `RData` file stockData.RData can be loaded using the `load` function. It contains three objects:

- `close_price`: a data frame containing the daily closing prices in 2015 for the stocks of 28 selected large US corporations. Each row is labeled by the relevant date, which can be extracted by the `rownames` function.
- `stock_info`: a small data frame containing basic information about the component stocks (courtesy of Wikipedia). The rows are in the same order as the columns of `close_price`. This is not quite necessary, but could be interesting.
- `tricky_prices`: This is explained in the last problem.

Many problems have hints for making the appropriate figures using base R graphics. You are welcome to use `ggplot` instead.

1. *Visualizing and transforming the data.*

   (a) (2) First, visualize the closing prices. Plot the closing prices for all stocks on the same graph. Use lines, not points. There is no need, here, to label the individual traces uniquely; you just want to see the shape of the data.

   (b) (2) The closing prices are on very different scales, and show clear dependence over time. It is more common to analyze the log returns, rather than the raw prices over time. Create a new data frame with the log daily returns for each stock. The log daily return at time t is defined as $\log\left(\frac{\text{price at time } t}{\text{price at time } t-1}\right)$. (This data frame will have the same number of columns, and one fewer row.) Plot the log returns over time, placing all 28 time series on the same plot. Do the log returns look more comparable than the closing prices? (You may see an even nicer plot if you add a little transparency to the lines.)

[1] See Homework 3.

2. *Exploring the distribution of log returns.*

   (a) (1) Focusing on Boeing (symbol BA), plot a histogram of the log returns. Use 30 cells instead of the default, so that you can better see the shape (`breaks=30`). Make sure you are plotting the probability density, not counts (`probability=TRUE`).

   (b) (2) The distribution of log returns is often modeled by a Gaussian distribution. Estimate the Gaussian that best fits the BA returns, using maximum likelihood. Create a new plot which shows both the histogram (from the previous question) and the estimated Gaussian pdf. (With base R graphics, `curve()` is the easiest way to do this.) How well does the Gaussian distribution appear to fit?

   (c) (5) It can be hard to see the shape and the deviations from Gaussianity very well in a histogram. Use a kernel density estimate with a Gaussian kernel to approximate the distribution. Use cross-validation to choose the appropriate kernel bandwidth. Plot the kernel density estimate, along with the best-fitting Gaussian density from the previous part. Where is the KDE notably higher/lower than the best-fit Gaussian? Is the KDE symmetric? How do the tails compare?

   (d) (3) Plot kernel density estimates for all 28 stocks on the same plot, separately cross-validating each one. Adjust the axes so that all the curves are visible. Do the other curves look similar to the BA curves, and do they seem to support your answers from the previous question (2c)?

3. *Our first factor model* Fit a one-factor model. (Using `factanal()`, as in the textbook, is recommended, but there are many other functions in R.)

   (a) (3) Make a barplot (or similar) of the vector of factor loadings. It will be easier to interpret your plot if you sort the weights prior to plotting. Also, make sure the axis labels are readable by making them perpendicular to the axis itself. (For `barplot()`, use the `las=1` or `las=2` options.) — A table will get more partial credit here than a hard-to-understand figure.

   (b) (3) Comment on any notable patterns in the loadings. (Looking up what some of the companies do may help.)

   (c) (3) Plot the factor score against the date. Comment on any notable patterns.

   (d) (3) How does the time-series plot in the previous problem compare to your earlier plot of log returns over time?

4. (10) *Bootstrapping will continue until morale improves* Use case bootstrapping (i.e., resampling of days) to find 85% confidence intervals for the factor loadings of the one-factor model. Again, report the results as a figure rather than a table.

5. *Testing an implication of the model*

   (a) (2) Find the sample covariances in the log-returns between (i) GE and Chevron; (ii) GE and Boeing; (iii) Goldman Sachs and Chevron; (iv) Goldman Sachs and Boeing.

   (b) (5) Find what those four covariances *should* be, according to the one-factor model you have previously estimated.

   (c) (5) Explain why, if the one-factor model is right, $\frac{\text{Cov}[GE,CVX]/\text{Cov}[GE,BA]}{\text{Cov}[GS,CVX]/\text{Cov}[GS,BA]} = 1$.

   (d) (5) Use case bootstrapping to give a 95% confidence interval for that ratio of covariances. Does it include 1? What can you conclude about the 1-factor model from whether or not it includes 1?

6. (10) *Second factor model* Fit a two factor model to the log-returns. Make plots of the factor loadings for both factors, and describe any patterns you see in the loadings. In particular, how has the first factor changed?

7. (5) *Changing the data* The `tricky_prices` data frame contains closing prices for the same stocks and two additional stocks. Again, convert these prices to log return values, and fit a one-factor model to all 30 stocks, as you did above.

   Look at your factor loadings, day-by-day factor scores, and closing prices. What changed when the new stocks were added? Why? Use plots and words to explain what happened.

8. (1) How long, roughly, did you spend on this assignment? How much of that time was spent on math, on coding/debugging, and on writing?

PRESENTATION RUBRIC (15): The text is laid out cleanly, with clear divisions between problems and sub-problems. The writing itself is well-organized, free of grammatical and other mechanical errors, and easy to follow. Plots are carefully labeled, with informative and legible titles, axis labels, and (if called for) sub-titles and legends; they are placed near the text of the corresponding problem. All quantitative and mathematical claims are supported by appropriate derivations, included in the text, or calculations in code. Numerical results are reported to appropriate precision. All parts of all problems are answered with actual coherent sentences, and raw computer code or output are only shown when explicitly asked for. Text from the homework assignment, including this rubric, is included only when relevant, not blindly copied.

(In Gradescope, assign *all* pages to this rubric.)

CODE RUBRIC (15): The code is logically organized and easy to read. No redundant code; no needlessly repetitive code; no unused code. Variables and functions have descriptive and appropriate names. (Loop or array indices, arguments, etc., can have short, conventional names such as `i` `x`, `df`, etc.) All functions have comments defining their purpose, their inputs, their outputs, and any dependencies on other code you wrote. Vectorization is used wherever appropriate. Allowed packages: `knitr`, `tidyverse`, `dplyr`, `ggplot2`, and those explicitly mentioned in the textbook or the assignment for implementing particular methods. (Any other packages require prior permission from the professor, which must be renewed for each assignment; record the date on which you got permission in your comments.) Code from the textbook and class examples is used wherever possible and appropriate. In particular, it should be used for tasks like bootstrapping, calibration plots, and cross-validation (*unless* the package implementing a model includes its own cross-validation functions). All plots and tables are generated by code included in the R Markdown file. Numerical results (etc.) appearing in text are neither hand-copied nor spat out with `cat()`, `print()`, `sprintf()` etc., but instead properly formatted through in-line code.

(Do not assign any pages to this rubric; instead, submit your Rmd file to the "HW *k* R Markdown File" assignment on Gradescope, for the appropriate *k*.)