Exercise 3

Machine Learning in Finance with Python (ECON5130)

Richard Foltyn
University of Glasgow

Deadline: November 24, 12:00

1 Daily returns of stocks in the Dow Jones Industrial Average (DJIA)

In this exercise, we examine the interdependency of daily returns of the 30 stocks in the Dow Jones Industrial Average index using principal component analysis.

This exercise can be downloaded as a Jupyter notebook from GitHub. You can use this notebook to get started when typing up your solution.

1.1 Load index components

Import the comma-separated data stored in `data/DJIA_components.csv` in the GitHub repository into a pandas `DataFrame`. This data set contains five columns:

1. Company name
2. Exchange on which the company is traded
3. The ticker symbol
4. The index weighting
5. The market capitalisation in billion USD (from early October 2022)

Sort the `DataFrame` by market capitalisation in descending order so that the most valuable companies are at the top and tabulate the five most valuable companies.

1.2 Download daily price data for the five largest companies

Download the daily price data for the five largest companies for the first half of 2022, i.e., for the period from 2022-01-01 to 2022-06-30 using the `pandas-datareader` package we discussed in the lecture together with the Yahoo! Finance backend. Keep only the column labelled `Close` for each company and discard the remaining data.

Hint: You may need to install `pandas-datareader` first by running the following code:

[5]:
```
# Uncomment to install pandas-datareader, in particular on Google Colab
# ! pip install pandas-datareader
```

1.3 Compute and plot daily returns

Using the daily prices at close, compute the daily returns for each of the five largest companies. Drop all rows containing `NaN`. Create a figure of 5-by-5 panels where each panel contains the pairwise scatter plot of daily returns of a company pair. Compute the correlation between each pair and add it as text to each panel.

Hint: You can add text to a Matplotlib axes object using the `text()` method.

1.4 Principal component analysis for sample of five largest companies

Use `scikit-learn`'s `PCA` to perform a principal component analysis of the daily return data. Use the maximum number of principal components for this sample (comment on what the maximum number is).

Plot the fraction of the total variance captured by each principal component as a bar chart. How many components are required to explain at least 90% of the variance in the return data?

1.5 Compute and plot loadings for each principal component

For each of the principal components you found above, compute the loadings with respect to each of the five stocks. Plot these as bar charts in a figure with one row per principal component, the individual stocks on the x-axis and the loadings on the y-axis. Which stocks does the first principal component load onto most?

1.6 PCA of all stocks in DJIA

1.6.1 Download data for remaining stocks and compute daily returns

Use `pandas-datareader` to download the daily price data for the remaining bottom 25 companies in the DJIA by market capitalisation (again only keep the `Close` column and discard the rest). Compute the daily returns for all 25 stocks in the same way you did earlier and merge them with the daily returns for the five largest stocks.

Hint: You can append additional columns to a `DataFrame` in many ways, but the easiest is by calling `pd.concat(..., axis=1)`.

1.6.2 Determine number of components by sample size

Consider the pooled sample of daily returns for all 30 companies you created in the previous sub-question. Proceed as follows:

For each $i = 1, 2, \ldots, 30$,

1. Select the daily returns of the i largest companies by market capitalisation.
2. Perform the PCA on this subset of daily returns.
3. Store the number of principal components required to explain at least 90% of the variance.

Create a plot with the number of companies included in the sample on the x-axis and the number of principal components required to explain 90% of the variance on the y-axis.