Financial Signal Processing Coursework

Prof. Danilo P. Mandic
TAs: Bruno Scalzo Dees, Giuseppe G Calvi, Shengxi Li, Zeyang Yu

March 26, 2019

Contents

Guidelines

The coursework yields 90% of the final mark. The remaining 10% accounts for presentation and organisation. Students are allowed to discuss the coursework but must code their own Python scripts, produce their own figures and tables, and provide their own discussion of the coursework assignments.

General directions and notation:

- The simulations should be coded in Python 3.
- The report should be clear, well-presented, and include the answers to the assignments in this handout with appropriate numbering.
- The report should document the results and the analysis in the assignments, in the form of figures (plots) and tables, and not by listing Python code as a proof of implementation.
- We adopt the following notation: boldface lowercase letters (e.g. $\mathbf{x}$) for vectors, lowercase letters with a (time) argument ($x[n]$) for scalar realisations of random variables and elements of a vector, and uppercase letters (X) for random variables. Column vectors will be assumed unless otherwise stated, that is, $\mathbf{x} \in \mathbf{R}^{N\times 1}$.
- The typewriter font, e.g. `mean`, is used for Python functions.

Presentation:

- Students are expected to submit **two files**. One "read-only" Jupyter Notebook (either in HTML, or PDF format), and one "executable" notebook (**in .ipynb format**), for the examiners to reproduce results. There is no length restriction, but the document should read as a stand-alone report, and each question should be numbered according to its corresponding section (e.g. Question 2 in Section 1.2 should be numbered 1.2.2).
- The final mark also considers the presentation of the report, this includes: legible and correct figures, tables, and captions, appropriate titles, table of contents, and front cover with student information.
- The figures and code snippets (only if necessary) included in the report must be carefully chosen, for clarity and to meet the page limit.
- For figures, (i) decide which type of plot is the most appropriate for each signal (e.g. solid line, non-connected points, stems), (ii) export figures in a correct format: without grey borders and with legible captions and lines.
- Avoid terms like *good estimate, is (very) close, somewhat similar*, etc - use formal language and quantify your statements.

Honour code:

Students are strictly required to adhere to the College policies on students rights and responsibilities.The College has zero tolerance to plagiarism. Any suspected plagiarism or cheating (or prohibited collaboration on the coursework, see above) will lead to a formal academic dishonesty investigation. Being found responsible for an academic dishonesty violation results in a discipline file for the student and penalties, ranging from severe reduction in marks to expulsion from College.

1 Regression Methods

1.1 Processing stock price data in Python

For the following question, initialise your Python session by importing the following modules:

```
import numpy as np                    # scientific library
import pandas as pd                   # data structure library
import matplotlib.pyplot as plt       # plotting library
%matplotlib inline
```

1. Import

   `http://www.commsp.ee.ic.ac.uk/~mandic/FSPML_Course/priceData.csv`

   using the following commands:

   ```
   px = pd.read_csv("priceData.csv")
   ```

   and perform the natural-log transform of the price using

   ```
   logpx = np.log(px)
   ```

   Plot the time-series using `logpx.plot()`. [1]

2. Using a sliding window of 252 days, compute and store the evolution of the "sliding-window-based" first and second-order statistics (mean and variance) of the price and log-price time series, using 1-day increments, and plot these as a function of time on separate figures. Comment on the stationarity of price time-series with reference to your figures. [2]

   ```
   # sliding mean
   plt.figure()
   logpx.rolling(252).mean().plot()
   plt.show()

   # sliding standard deviation
   plt.figure()
   logpx.rolling(252).std().plot()
   plt.show()
   ```

3. Compute the simple and log return time-series from the price data using the script: [2]

   ```
   # log return
   logret = logpx.diff()

   # simple return
   simpret = px.pct_change()
   ```

   Based on the Python commands in Question 1.1.1, produce figures of the "sliding" statistics of the obtained time-series and comment on the stationarity of these returns in comparison to the figures obtained in Question 1.1.1.

4. Theoretically justify the suitability of log returns over simple returns for signal processing purposes. Next, perform the "Jarque-Bera" test for Gaussianity on the data, and comment on the results in light of your theoretical answer (hint: use the Python commands provided below). [3]

   ```
   from scipy import stats

   # return the Jarque-Bera test p-value for a time-series x
   stats.jarque_bera(x)[1]
   ```

5. You purchase a stock for £1. The next day its value goes up to £2 and the following day back to £1. What are the simple and logarithmic returns over this period and what can you conclude about logarithmic returns on the basis of this example? [1]

6. Under what circumstances should you not use log returns over simple returns? [1]

1.2 ARMA vs. ARIMA Models for Financial Applications

For this question you will need to import the following Python packages via:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from statsmodels.tsa.arima_model import ARIMA
from statsmodels.tsa.ar_model import AR
import copy
```

An autoregressive-moving-average process, ARMA(p, q), is a stochastic process $\{x[t]\}$, composed of both

- An AR part that regresses the variable $x[t]$ on its own lagged values, $x[t-1], \ldots, x[t-p]$.
- An MA part which models the error term as a linear combination of error terms at various times in the past, $y[t-1], \ldots, y[t-q]$.

In other words, the ARMA(p, q) model takes the form of

$$x[t] = \sum_{i=1}^{p} a_i x[t-i] + \sum_{i=1}^{q} b_i \eta[t-i] + \eta[t] \tag{1}$$

The ARMA models are widely used in finance, as

- The AR(p), i.e. the autoregressive component aims to explain the momentum and mean reversion effects often observed in trading markets. These can be thought of as the effects due to the participants.
- The MA(q) component, i.e. the moving average, attempts to capture, in signal processing terms, the shock effects observed as white noise. In finance, these shock effects can be though of as unexpected events which affect the observation process, e.g. wars, news, etc.

All ARMA models assume stationarity of data. This means that when sources of non-stationarity are present (i.e. a trend), the ARMA model in its original form may not be particularly suitable for analysis.

1. Import

   `http://www.commsp.ee.ic.ac.uk/~mandic/FSPML_Course/snp_500_2015_2019.csv`

   containing closing prices of the S&P 500 over the last 4 years and take the log of the data using the following script: [10]

   ```
   snp = pd.read_csv(‘snp_500_2015_2019.csv’)
   snp.set_index([‘Date’], inplace=True)
   snp_close = snp[‘Close’].to_frame().apply(np.log)
   ```

   Plot the S&P 500 time-series. Following the process in Question 1.1.1, comment on whether an ARMA or ARIMA model would be more appropriate.

2. Fit an ARMA$(1, 0)$ model using the commands below: [10]

   ```
   snp_arma = copy.deepcopy(snp_close)
   snp_arma.columns = [‘True’]
   snp_arma[‘Res’] = ARIMA(snp_arma, order=(1,0,0)).fit().resid
   snp_arma[‘Prediction’] = snp_arma[‘True’] - snp_arma[‘Res’]
   ```

   Plot, in the same figure, both the prediction and the true signal. Inspect the model parameters (`model.params`). Comment on the results. Are these findings useful in practice?

An ARIMA(p, d, q) model is essentially the same as ARMA, with the exception that it applies an initial differencing on the time-series in hand to remove sources of non-stationarity, where d is the differencing order.

3. Repeat Question 1.2.2, this time by fitting an ARIMA$(1, 1, 0)$ model. Comment on the results. Compare your results with those in Question 1.2.2. Which analysis is more physically meaningful? [10]

4. Comment on the necessity of taking the log of the prices for the ARIMA analysis. [15]

1.3 Vector Autoregressive (VAR) Models

Now, consider a multivariate extension of the AR processes, the so called VAR(p) process given by

$$\mathbf{y}_t = \mathbf{c} + \mathbf{A}_1\mathbf{y}_{t-1} + \mathbf{A}_2\mathbf{y}_{t-2} + \cdots + \mathbf{A}_p\mathbf{y}_{t-p} + \mathbf{e}_t \tag{2}$$

or, in an expanded matrix notation

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \\ \vdots \\ y_{k,t} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{bmatrix} + \begin{bmatrix} a^1_{1,1} & a^1_{1,2} & \cdots & a^1_{1,k} \\ a^1_{2,1} & a^1_{2,2} & \cdots & a^1_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ a^1_{k,1} & a^1_{k,2} & \cdots & a^1_{k,k} \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \\ \vdots \\ y_{k,t-1} \end{bmatrix} + \cdots + \begin{bmatrix} a^p_{1,1} & a^p_{1,2} & \cdots & a^p_{1,k} \\ a^p_{2,1} & a^p_{2,2} & \cdots & a^p_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ a^p_{k,1} & a^p_{k,2} & \cdots & a^p_{k,k} \end{bmatrix} \begin{bmatrix} y_{1,t-p} \\ y_{2,t-p} \\ \vdots \\ y_{k,t-p} \end{bmatrix} + \begin{bmatrix} e_{1,t} \\ e_{2,t} \\ \vdots \\ e_{k,t} \end{bmatrix} \tag{3}$$

1. Show how Equations (2)-(3) can be represented in a concise matrix form as [10]

   $$\mathbf{Y} = \mathbf{B}\mathbf{Z} + \mathbf{U} \tag{4}$$

   where $\mathbf{Y} \in \mathbb{R}^{K\times T}$, $\mathbf{B} \in \mathbb{R}^{K\times(KP+1)}$, $\mathbf{Z} \in \mathbb{R}^{(KP+1)\times T}$ and $\mathbf{U} \in \mathbb{R}^{K\times T}$. (Hint: let $\mathbf{B} = \begin{bmatrix}\mathbf{c} & \mathbf{A}_1 & \mathbf{A}_2 \cdots \mathbf{A}_p\end{bmatrix}$).

2. Hence, show that the optimal set of coefficients $\mathbf{B}$, denoted by $\mathbf{B}_{\text{opt}}$, is obtained via [5]

   $$\mathbf{B}_{\text{opt}} = \mathbf{Y}\mathbf{Z}^T(\mathbf{Z}\mathbf{Z}^T)^{-1} \tag{5}$$

   Show the whole derivation for full marks.

3. Now, consider a VAR(1) process, that is [10]

   $$\mathbf{y}_t = \mathbf{A}\mathbf{y}_{t-1} + \mathbf{e}_t \tag{6}$$

   For the previous time instant, the above can be rewritten as

   $$\mathbf{y}_{t-1} = \mathbf{A}\mathbf{y}_{t-2} + \mathbf{e}_{t-1} \tag{7}$$

   Using Equations (6)-(7), elaborate on how, for stability, all the eigenvalues of the matrix $\mathbf{A}$ must be less than 1 in absolute value.

In the following you will investigate how VAR models can aid in the construction of portfolios. You will need to import the following Python packages:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from statsmodels.tsa.api import VAR
```

and load the required data via:

```
df = pd.read_csv(r'snp_allstocks_2015_2019.csv')
df = df.set_index('Date')

info = pd.read_csv(r'snp_info.csv')
info.drop(columns=info.columns[0], inplace=True)
```

The file

`http://www.commsp.ee.ic.ac.uk/~mandic/FSPML_Course/snp_allstocks_2015_2019.csv`

contains daily closing prices of all the stocks composing the S&P 500 index in the period from January 1st, 2015 to January 1st, 2019, while the file

`http://www.commsp.ee.ic.ac.uk/~mandic/FSPML_Course/snp_info.csv`

contains information related to each stock (e.g. sector, headquarters location, etc.).

4. Consider the time-series of the stocks with tickers `CAG`, `MAR`, `LIN`, `HCP`, `MAT`, and detrend them using an MA(66) model (66 corresponds to 3 × 22, i.e. one quarter), that is, run [10]

   ```
   tickers = ['CAG', 'MAR', 'LIN', 'HCP', 'MAT']
   stocks = df[tickers]
   stocks_ma = stocks.rolling(window=66).mean()
   stocks_detrended = stocks.sub(stocks_ma).dropna()
   ```

Now, fit a VAR(1) model to these time-series and compute the eigenvalues of the regression matrix **A**, by:

```
model = VAR(stocks_detrended)
results = model.fit(1)
A = results.params[1:].values
eigA, _ = np.linalg.eig(A)
```

Elaborate on whether it would make sense to construct a portfolio using these stocks? Why? Comment on your results.

5. With the aid of `snp_info.csv`, repeat Question 1.3.4 but this time by selecting the tickers according to their sector, that is [10]

```
for sector in info['GICS Sector'].unique():
tickers = info.loc[info['GICS Sector']==sector]['Symbol'].tolist()
stocks = df[tickers]
.
.
.
```

Explain whether it would be more advisable, in general, to build a portfolio by grouping the stocks by sector? Comment on your results.

2 Bond Pricing

2.1 Examples of bond pricing

Give worked-our solutions to the following problems:

1. An investor receives USD 1,100 in one year in return for an investment of USD 1,000 now. Calculate the percentage return per annum with: a) Annual compounding, b) Semiannual compounding, c) Monthly compounding, d) Continuous compounding [4]

2. What rate of interest with continuous compounding is equivalent to 15% per annum with monthly compounding? [1]

3. A deposit account pays 12% per annum with continuous compounding, but interest is actually paid quarterly. How much interest will be paid each quarter on a USD 10,000 deposit? [1]

2.2 Forward rates

1. Suppose that the one–year interest rate, r_1 is 5%, and the two–year interest rate, r_2 is 7%. If you invest USD 100 for one year, your investment grows to $100 \times 1.05 =$ USD 105; if you invest for two years, it grows to $100 \times 1.07^2 =$ USD 114.49. The *extra* return that you earn for that second year is $1.07^2/1.05 - 1 = 0.090$, or 9.0 %.
   (Hint: The extra return for lending for one more year is termed the *forward rate of interest*).

   a) Would you be happy to earn that extra 9% for investing for two years rather than one? [1]

   b) Comment on the 5%, 7%, and 9% investment strategies. [1]

   c) Comment on the advantages and disadvantages of the forward rate of 9%. [1]

   d) How much would you need to go from 1y investment to 2y investment and what does it depend upon? [1]

2.3 Duration of a coupon-bearing bond

1. Duration is the weighted average of the times to each of the cash payments. The times are the future years 1, 2, 3, etc., extending to the final maturity date, which we call T. The weight for each year is the present value of the cash flow received at that time divided by the total present value of the bond.

$$\text{Duration} \;=\; \frac{1 \times PV(C_1)}{PV} + \frac{2 \times PV(C_2)}{PV} + \frac{3 \times PV(C_3)}{PV} + \cdots + \frac{T \times PV(C_T)}{PV}$$

The Table below shows how to compute duration for the 1% seven-year Treasuries, assuming annual payments. First, we value each of the coupon payments of USD 10 and the final payment of coupon plus face value of USD 1,010. The present values of these payments add up to the bond price of USD 768.55. Then, we calculate the fraction of the price accounted for by each cash flow and multiply each fraction by the year of the cash flow.

   a) Calculate the duration for the 1% bond in the Table. [1]

Year	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Total**
Payment	\$ 10	\$ 10	\$ 10	\$ 10	\$ 10	\$ 10	\$ 1010	\$ **1070.00**
PV($\mathbf{C_t}$) at 5%	\$ 9.52	\$ 9.07	\$ 8.64	\$ 8.23	\$ 7.84	\$ 7.46	\$ 717.79	**PV** = \$ **768.55**
Fraction of PV [PV($\mathbf{C_t}$)/PV]	0.0124	0.0118	0.0112	0.0107	0.0102	0.0097	0.9340	
Year × Fraction of PV [t × PV($\mathbf{C_t}$)/PV]	0.0124	0.0236	0.0337	0.0428	0.0510	0.0583	6.5377	

Table 1: Calculating the duration of the 1% 7-year bonds. The yield to maturity is 5% a year.

Investors and financial managers track duration because it measures how bond prices change when interest rates change. For this purpose it is best to use *modified duration or volatility* which is just duration divided by one plus the yield to maturity, that is

$$\text{Modified duration} \quad = \text{volatility (\%)} \quad = \frac{\text{duration}}{1 + \text{yield}}$$

Modified duration measures the percentage change in bond price for a 1 percentage–point change in yield. In other workds, the derivative of the bond price with respect to a change in yield to maturity is $dPV/dy = -\text{duration}/(1{+}\text{y}) = -\text{modified duration}$.

b) Calculate the modified durations for the 1 % bonds in the above table, and elaborate on the differences from the calculation in Part a). [2]

c) Explain why duration (or modified duration) are convenient measures to protect the pension plan against unexpected changes in interest rates. [2]

2.4 Capital Asset Pricing Model (CAPM) and Arbitrage Pricing Theory (APT)

Consider the daily returns of 157 European companies, which you can download from

`http://www.commsp.ee.ic.ac.uk/~mandic/FSPML_Course/fsp_case_31_BSD.csv`

and import using the following command[1]:

```
import pandas as pd
df = pd.read_csv(r'fsp_case_31_BSD.csv',index_col=0,header=[0,1])
```

Throughout the assignment, the index i refers to a particular company, $i = 1, \dots, 157$, and the index t to a particular time instant (day), $t = 1, \dots, 500$.

1. Estimate the market returns per day R_{m_t}= average(company returns)). [2]

2. Estimate a rolling (sliding) beta, $\beta_{i,t}$, for every company i, with the rolling window of 22 days. [5]
   (*Hint: Estimate time series of* $\beta_{i,t}$ *wrt the market, comment on the volatility of* $\beta_{i,t}$)

3. Estimate the cap-weighted market return, R_m, where for every day

$$R_m = \text{ret(market)} = \sum_i \frac{\text{mcap}_i \times \text{ret}_i}{\sum_i \text{mcap}_i}$$

   and the weighting coefficient is $\sum_i \text{mcap}_i$. How can you interpret this weighting coefficient? [3]

4. Estimate a rolling $\beta_{m_{i,t}}$ like in Part 2 above but with the market return from Part 3. Compare the two betas in Part 2 and Part 4, that is, the equally-weighted β_m and the cap–weighted β_m . [3]
   For the rest of the exercise we will use the cap–weighted β_m (CAPM β).

5. Assume that the arbitrage pricing theory (**APT**) holds for a two–factor model, and assume that the exposure to size is $b_{s_i} = ln(size)$, that is, per company i and per every day t.
   Run for every day, t, the following cross-sectional regression

$$r_i = a + b_{m_i} R_m + b_{s_i} R_s + \varepsilon_i$$

   where ε_i is the residual of this regression (aka specific return), and r_i denotes the return per company. Effectively, we regress for a, R_m and R_s (or for R_{s_t} as this is for every day) and after this regression, you will have one a, one R_m and one R_s per day.

   a) Estimate a, R_m, and R_s. [5]

   b) Comment on the magnitude and variance of a, R_m, and R_s. [1]

   c) Now, we are moving to the temporal domain rather than the spatial (cross-sectional) domain in Part 1. Find the correlation through time for every company, that is, $< \varepsilon_i, r_i >$, where ε_i is called the *specific return.* [3]

   d) From the results of Part 5a) you have two time series per day, that is, two vectors of returns, R_m and R_s, which can be combined into the matrix [5]

$$\mathbf{R} = \begin{bmatrix} R_{m_1} & R_{s_1} \\ \vdots & \vdots \\ R_{m_{500}} & R_{s_{500}} \end{bmatrix}$$

   Calculate the covariance matrix, $cov(\mathbf{R})$ (*Hint: it is of size* 2×2) using a rolling window of 22 days. Comment on the magnitude and stability of the covariance matrix.

[1] The missing values in the file should be replaced by 0s.

e) From Part 5a), you have for every company, i, and for every day, t, the specific return $\varepsilon_{i,t}$. We can therefore form the following matrix

$$\mathbf{E} = \begin{bmatrix} \varepsilon_{1,t=0} & \cdots & \varepsilon_{157,t=0} \\ \varepsilon_{1,t=1} & \cdots & \varepsilon_{157,t=1} \\ \vdots & \ddots & \vdots \\ \varepsilon_{1,t=500} & \cdots & \varepsilon_{157,t=500} \end{bmatrix}_{500\times 157}$$

Estimate the covariance matrix of these specific returns, $cov(\mathbf{E})$ (*Hint: the size of the matrix is* 157×157).
Perform PCA on the covariance matrix from Part e), find the percentage of the variance explained by the first principal component, and comment on the result. [8]

3 Portfolio Optimization

3.1 Adaptive minimum-variance portfolio optimization

Consider a portfolio of M assets where the price of an asset m at time t is denoted by $p_m[t]$. The return of each asset $r_m[t]$ is then given by

$$r_m[t] = \frac{p_m[t] - p_m[t-1]}{p_m[t-1]} \tag{8}$$

It is convenient to collect the returns together into a vector, $\mathbf{r}[t] = [r_1[t], ..., r_M[t]]^T$, where each asset m is characterised by the mean and variance of its returns, $\mu_m = E\{r_m[t]\}$ and $\sigma_m^2 = E\left\{(r_m[t] - \mu_m)^2\right\}$. The correlation between the returns of two assets m and l is given by the correlation coefficient $\rho_{ml} \in [-1, 1]$, while the covariance matrix of all assets is given by

$$\mathbf{C} = E\left\{(\mathbf{r}[t] - \boldsymbol{\mu})(\mathbf{r}[t] - \boldsymbol{\mu})^T\right\} \in \mathbb{R}^{M\times M} \tag{9}$$

$$\boldsymbol{\mu} = E\{\mathbf{r}[t]\} \in \mathbb{R}^M \tag{10}$$

and summarizes the "risk" structure of the system, where the individual covariances between assets m and l are given by $[\mathbf{C}]_{ml} = \sigma_{ml} = \rho_{ml}\sigma_m\sigma_l$.

It is important to realise that the holdings of a portfolio can be represented by a set of weights, $\mathbf{w} = [w_1, ..., w_M]^T$, that we apply to our assets. In other words, the weight w_m represents the percentage of our capital which is allocated to asset m. The return of the portfolio at time t is therefore given by $\bar{r}[t] = \mathbf{w}^T\mathbf{r}[t]$, so that the expected return and variance of the portfolio are respectively given by

$$\text{Expected return: } \bar{\mu} = E\{\bar{r}[t]\} = \mathbf{w}^T\boldsymbol{\mu} \tag{11}$$

$$\text{Variance: } \bar{\sigma}^2 = E\left\{(\bar{r}[t] - \bar{\mu})^2\right\} = \mathbf{w}^T\mathbf{C}\mathbf{w} \tag{12}$$

Finding the set of weights $\mathbf{w}$ for a portfolio represents a large portion of the task of asset managers. One of the simplest criteria used in portfolio management is to minimize the risk (measured by variance) of the portfolio, given by (12). Therefore, the minimum variance portfolio is derived based on finding the set of optimal weights, $\mathbf{w}$, which minimizes the variance, $\bar{\sigma}^2$, subject to the portfolio weights summing up to unity, i.e. $\mathbf{w}^T\mathbf{1} = 1$.

1. Derive the optimal weights to construct the minimum variance portfolio by solving the following optimization problem:

$$\min_{\mathbf{w}} \quad J(\mathbf{w}, \mathbf{C}) = \frac{1}{2}\mathbf{w}^T\mathbf{C}\mathbf{w} \tag{13}$$

$$\text{subject to} \quad \mathbf{w}^T\mathbf{1} = 1 \tag{14}$$

   or, equivalently, the Lagrange optimization

$$\min_{\mathbf{w},\lambda} \quad J'(\mathbf{w}, \lambda, \mathbf{C}) = \frac{1}{2}\mathbf{w}^T\mathbf{C}\mathbf{w} - \lambda\left(\mathbf{w}^T\mathbf{1} - 1\right) \tag{15}$$

   Secondly, derive the theoretical variance of your returns if we were to apply the minimum variance estimator. [15]

2. Download the daily returns of the **last** 10 stocks from

   `http://www.commsp.ee.ic.ac.uk/~mandic/FSPML_Course/fsp_case_31_BSD.csv`

   and split your data into the training ($\sim$ 50% of the data) and testing data ($\sim$ 50% of the data). Compute the minimum variance portfolio weights from the training data and compare with the performance obtained on the test data with an equally-weighted portfolio (i.e. with $w_m = 1/M$, $\forall m$). The performance of your strategy over a time-horizon of T days can be measured by the portfolio variance, $\bar{\sigma}^2$, and the cumulative return, which is given by

$$\text{Cumulative return: } R[T] = \sum_{t=1}^{T} \bar{r}[t] \tag{16}$$

   Compare the variance of your portfolio returns to the theoretical value obtain in Part 1 *Hint: plot the cumulative returns with the aid of the* `pandas.DataFrame.cumsum()` *function in Python.* [15]

3. Implement the adaptive time-varying minimum variance portfolio using the empirical estimate of the mean vector and covariance matrix, using a rolling window of length M days, given by

$$\hat{\boldsymbol{\mu}}[t] = \frac{1}{M} \sum_{\tau=t-M+1}^{t} \mathbf{r}[\tau] \tag{17}$$

$$\hat{\mathbf{C}}[t] = \frac{1}{M} \sum_{\tau=t-M+1}^{t} \left(\mathbf{r}[\tau] - \hat{\boldsymbol{\mu}}[t]\right)\left(\mathbf{r}[\tau] - \hat{\boldsymbol{\mu}}[t]\right)^T \tag{18}$$

$$\hat{\mathbf{w}}[t] = \arg\min_{\mathbf{w},\lambda} \quad J(\mathbf{w}, \lambda, \hat{\mathbf{C}}[t]) \tag{19}$$

Compare the performance of this adaptive minimum variance scheme with the solution you derived in Part 1 and the equally-weighted strategy in Part 2. Explain the effect of the recursive update of the variables involved on the performance. Elaborate on whether there exist a different method to compute the sample covariance matrix? [15]

4 Robust Statistics and Non Linear Methods

4.1 Data Import and Exploratory Data Analysis

In this assignment, you will employ different robust statistical techniques on the following financial data:

- 3 stocks (AAPL, IBM and JPM) and 1 index (DJI);
- Dates ranging from 16/03/2018 to 11/03/2019;
- Each file contains the following columns: *open, high, low, close, adj. close*[2].

You will first conduct the following exploratory data analysis:

1. Import `AAPL.csv`, `IBM.csv`, `JPM.csv` and `DJI.csv` into separate `pandas.DataFrames`, and set the date as the index column. For each stock and for each column, generate the key descriptive statistics (e.g. mean, median, stddev, etc.) that summarize the distribution of the dataset. Lastly, using the *adj. close* column for each stock, compute the 1-day returns and add them to their corresponding dataframe as a new column. [5]

2. Plot the histogram and probability density function of the *adj. close* and 1-day returns. Comment on the difference, if any, between the pdf of the the *adj. close* and the returns. [3]

3. For each stock, plot the *adj. close*, the associated **rolling mean** (using a 5-day window), and the ±1.5× **standard deviations** relative to the rolling mean[3]. In a separate figure, repeat the steps above using the **rolling median** (using a 5-day window) and ±1.5× **median absolute deviation** relative to the rolling median. Comment on the difference, if any, between the two figures. [9]

4. Introduce outlier points for the *adj. close* in the four dates {2018-05-14, 2018-09-14, 2018-12-14, 2019-01-14} with a value equal to 1.2 × the maximum value of the column. Comment on the impact of the outlier points in Part 3. [3]

5. Generate a box plot for the *adj. close* for each stock, describe the information the box plot conveys and elaborate on any other observations you may have. [5]

4.2 Robust Estimators

This section involves the implementation, analysis and assessment of the following estimators:

- Robust location estimator: median
- Robust scale estimator: IQR (Interquartile range) and MAD (Median Absolute Deviation)

1. Create a Python function for each estimator type that takes a `pandas.Series` as input, and returns the estimator value as output. [15]

2. Assess and compare the respective relative computational efficiency of the different estimators. [5]

3. Assess and compare the breakdown points associated with the different estimators. [5]

4.3 Robust and OLS regression

1. Regress each stock's 1-day returns against the 1-day returns of DJI using Ordinary Least Squares (OLS) regression. [10]

2. Regress the 1-day returns for each stock against the 1-day returns of DJI using Huber Regression (check `sklearn HuberRegressor`). [10]

3. Assess and compare the results obtained using both regression methods and the impact of outliers in both cases. [5]

[2] An *adjusted closing price* is a stock's closing price that has been amended to account for any corporate actions such as stock splits, dividends/distributions and/or rights offerings.

[3] Refer to slide: "Anomaly Detection: Z-score based methods".

4.4 Robust Trading Strategies

The *Moving Average Crossover* is a simple trading strategy based on the following rules:

a. Buy X shares of a stock when its 20-day MA $>$ 50-day MA

b. Sell X shares of the stock when its 20-day MA $<$ 50-day MA

In this section, you will implement robust variations of the *Moving Average Crossover* strategy:

1. For each stock, plot the *adj. close*, the 20-day and 50-day MAs, and highlight the crossover points and the regions to buy or sell. In a separate figure, repeat these steps using the *adj. close* values corrupted with outliers for each stock. [10]

2. Instead of using the rolling mean (MA), use the rolling median and repeat the steps in Part 1. Comment on the difference between the trading signals generated when using the mean *vs* the median for the data points corrupted with synthetic outliers. [15]

5 Graphs in Finance

The answers to these questions are expected to be “open-ended" in the sense that students are encouraged to be creative and express themselves through identifying the utility of graphs in finance. **You are welcome to work either as individuals or in pairs, within the following format.**

A. If you are working as an individual, then solve the following five given questions. The task involves applying graphs to modelling and visualizing the relationship between stocks within the S&P 500 index.

1. Load the `snp_allstocks_2015_2019.csv` and the `snp_info.csv` files, as shown in previous sections of the coursework. Consider log-returns. Choose *up to* 10 assets of your choice, explaining your motivations (e.g. they could be stocks within a given sector, or the stocks with the highest market-cap of each sector, industry, location or criterion of your choice). [10]

2. Open the link `https://python-graph-gallery.com/327-network-from-correlation-matrix/` and install the Python module `networkx`. Following the instructions on the webpage, and other related posts you may find online, construct a graph using your selected stocks based on their correlation matrix. Explain the role of the correlation matrix and show the results as clearly as possible. [15]

3. Discuss your results, especially whether the topology of your graph is dictated by the nature of the data. Would the re–ordering of graph vertices affect your results? Would the re–ordering of your time–series data affect your results? [15]

4. Adopt a distance metric other than correlation (this could be any dissimilarity measure, spectral distance, etc.). Read the list at `https://rdrr.io/cran/TSdist/man/TSdist-package.html` for illustration. Justify your choice, and repeat Q2 and Q3 above. [15]

5. Give your intuition on how Q1–Q4 would be affected if you considered raw prices. [15]

B. If you are working in pairs, then you should propose a study within *any* finance–related context, that in your opinion admits a graph representation and benefits from optimisation on graphs. Your study may be any global or local financial problem – from the stock market to the global macroeconomic system. Rigorous elaboration at a conceptual level is also welcome, as long as there is a clear hint/link to practical utility. At our discretion, we will consider bonus marks for the most original and feasible solutions. **Do not forget to state the name of your partner.** [70]