[14X030] Introduction to Computational Finance

Exercise series 2

March 7, 2023

General instructions

Each student is expected to upload on Moodle a **zip** file containing:

- A report in **pdf** format to answer exercise questions.
- The code used to generate the results of the report.

Deadline

Upload on Moodle due by : **March 13, 2023 at 11:59 pm**.

1 Correlation of two random variables

- Let A and B be two independent random variables such that $A \sim \mathcal{N}(\mu_A, \sigma_A)$ and $B \sim \mathcal{N}(\mu_B, \sigma_B)$, with $\mu_A = \mu_B = 0$, $\sigma_A = 1$ and $\sigma_B = 2$.
- Let X and Y be two random variables such that $X = A + 4B$ and $Y = 2A + B$.

1. Derive analytically the following quantities:
   (a) μ_X, μ_Y, σ_X and σ_Y, the expectations and standard deviations of X and Y.
   (b) The covariance between X and Y, defined as :
   $$\mathrm{Cov}(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)]$$
   (c) The correlation coefficient between X and Y, defined as :
   $$\mathrm{Cor}(X, Y) = \frac{\mathrm{Cov}(X, Y)}{\sigma_X \sigma_Y}$$
2. Using the language of your choice:
   (a) Simulate $n = 10000$ realizations of A and B and draw the graph of the corresponding realizations of X vs realizations of Y.
   (b) How does this relate to the value of Cor(X, Y)? Is the slope equal to the correlation coefficient? Comment on this last point.
   (c) Compute the empirical correlation coefficient from the n realizations and compare it to the value of $\mathrm{Cor}(X, Y)$ obtained from the above analytical derivation (1c).

Hint: In Python, one can generate series of X and Y with:

```
import numpy as np

np.random.seed(0)

n = 10000
mu_a, sig_a = 0, 1
mu_b, sig_b = 0, 2

a = np.random.normal(mu_a, sig_a, n)
b = np.random.normal(mu_b, sig_b, n)

x = a + 4 * b
y = 2 * a + b
```

2 Hidden sine

Consider the following time series:

$$\forall t \in \{1, \dots, n\}, X_t = \sin\left(\frac{2\pi t}{T}\right) + \varepsilon_t$$

With $\varepsilon_t \sim \mathcal{N}(0, 1)$, $T = 10$ and $n = 1000$.

1. Plot a realization of the time series above.
2. Compute and plot the corresponding ACF graph for lags 0 to 50.
3. Try with other values of T and comment on the impact of T on the ACF graph.

3 AR model on an empirical time series

The file `eur_usd.txt` (download it from Moodle) contains the daily EUR_USD price evolution with the following data structure:

```
timestamp price
```

where `timestamp` denotes the time (in seconds) measured starting from `01 Jan 1970 00:00:00.000`.
You can use the following snippet of Python code to load the data:

```python
from datetime import datetime
import numpy as np

data = np.loadtxt('eur_usd.txt')
price_ts = data[:, 1]

# Tip: You can convert the timestamp to a better format:
days = [datetime.fromtimestamp(x).strftime('%b %d') for x in data[:, 0]]
```

1. Preprocessing of the time series.
   (a) Plot the time series. Does it look stationary?
   (b) Compute the corresponding daily returns and plot this new time series. Does it look stationary? For the rest of the exercise, substract the mean of this series to center it.
2. Analysis of the time series of daily returns.
   (a) Compute and plot the ACF for lags 0 to 10.

(b) Using the analytical expressions seen during the course, compute the parameter ϕ_1 of the AR(1) model for this time series.

(c) Use your model to do predictions from the initial value. Plot the predictions along the time series of daily returns on the same graph. What do you think of these predictions?

3. AR(p) model with a library.

   (a) In Python, you can use the `statsmodels` library to fit an AR(p) model on a time series.

   ```
   from statsmodels.tsa.ar_model import AutoReg
   predictions = AutoReg(your_time_series, lags=p).fit().predict()
   ```

   Plot the AR(1) predictions computed with this library and compare it to the predictions of your model obtained in 2c (hint: it should be really similar).

   (b) Using this library, plot the predictions for higher-order AR models. Does the quality of the predictions improves?