**Python for Finance**

**Paul Lecubin**

Table of Contents

[Introduction: 2](#_Toc151534821)

[1.The code: 2](#_Toc151534822)

[The core: 2](#_Toc151534823)

[Excel file: 3](#_Toc151534824)

[2.Analysis of our model: 3](#_Toc151534825)

# Introduction:

For this Project I have chosen to implement the CAPM model in python to be able to analyze the Amazon stock.

In this project, we'll apply the CAPM asset pricing model to see how the Amazon stock performs. CAPM is a tool that helps evaluate the performance of a stock or investment, taking into account risk and general market conditions. In our case, we're going to use Amazon's historical data from 2022-01-01 to 2023-01-01 to study the risk and return we can expect from the stock.

Now let's analyze how we created the code to perform this analysis.

# 1.The code:

Preliminary step:

Before writing a line of code, we first need to import a library. We've imported sys, os, matplotlib, numpy, pandas, yfinance and openpyxl, which are needed to manipulate, calculate and visualize the financial data we've imported using an API from Yahoo Finance.

## The core:

We then proceed to calculate the CAPM by defining the function calculate\_capm

A ticker is a series of letters representing the shares of a listed company. We implement this function to correctly identify the company. We do the same for the index, and we also implement the risk-free rate. Using the stock data and market data functions, we download all the data required for our analysis.

We want to get the stock returns and market\_returns to be able to apply our formula, then we calculate our Beta, expected return and required return using numpy's polynomial adjustment.

.pct\_change() is a function that calculates the variation for the period we've chosen. .dropna() cleans up and deletes all non-data values. NAN is also an element that can appear if there is no number.

np.polyfit() is a Numpy function, used in the code to perform a polynomial fit, the degree in this model is 1 (linear polynomial). Market returns is considered in this model as the independent variable, and it is this variable that influences in some way the dependent variable stock return - risk free rate.

We now calculate the expected return = rf + beta \* (mr mean - rf)

Then the required return: rf + beta \* (mr max -rf)

We have decided that in this model the risk-free rate will be 1%.

## Excel file:

Having calculated our model, we now want to have an Excel file that summarizes the information we've calculated. So, we use the openpyxl library.

We also wanted to visually represent the model by comparing the Amazon share price with the stock price. We used the matplotlib function to do this.

We also performed additional calculations that could help with the analysis, such as the average share price.

We also created a scatterplot to show all the data.

# 2.Analysis of our model:

In our model, the beta is 1.6, indicating that Amazone's volatility is higher than the market. A beta higher than 1 indicates greater volatility and therefore higher risk.

The expected return in our model is equal to -0.0077 which means that the stock of amazon is not very good to get return l'action à un rendement négatif. So, on average, an investor can expect to lose money by owning an Amazon share.

The required return is 0.0845, which is the minimum return. 8.45% is the minimum return an investor could expect for the risk involved in owning an Amazon share.

A visual comparison, however, shows that the market is generally more volatile than the Amazon share.