
Using Simulation Models for Financial Engineering

Juan F.
Mendoza M.

Simulation (May, 2022)

Using Simulation Models for Financial Engineering

- The explosive entry of systems based on **artificial intelligence and machine learning** has opened up new scenarios for the financial sector.
- These methods can bring benefits such as **user rights protections**, as well as **macroeconomic benefits**.
- **Monte Carlo methods** find a natural application in finance for the numerical resolution of **pricing** and problems in covered call options.

Overview

- Essentially, these methods consist of simulating a given process or phenomenon using a given **mathematical law** and a sufficiently **large set of data, created randomly** from distributions that adequately represent real variables.
- The idea is that, **if an analytical study is not possible**, or adequate **experimental sampling is not possible or convenient**, the **numerical simulation** of the phenomenon is used.
- In this chapter, we will look at practical cases of using simulation methods in a financial context.
- You will learn how to use **Monte Carlo methods to predict stock prices** and how to **assess the risk** associated with a portfolio of shares.

-
- Understanding the geometric Brownian motion model
 - Using Monte Carlo methods for stock price prediction
 - Studying risk models for portfolio management

Juan F.
Mendoza M.

Agenda

-
- Understanding the geometric Brownian motion model
 - Using Monte Carlo methods for stock price prediction
 - Studying risk models for portfolio management

Juan F.
Mendoza M.

Agenda

Understanding the geometric Brownian motion model

- The name Brownian comes from the Scottish botanist **Robert Brown** who, in 1827, observed, under the microscope, how **pollen particles suspended in water moved continuously in a random** and unpredictable way.
- In 1905, it was **Einstein** who gave a molecular **interpretation of the phenomenon of movement** observed by Brown.
- He suggested that the motion of the particles was **mathematically describable**, assuming that the various jumps were due to the **random collisions of pollen particles with water molecules**.

Mathematical models

- Today, **Brownian motion** is, above all, a **mathematical tool** in the context of **probability theory**.
- This **mathematical theory** has been used to describe an ever-widening set of phenomena, studied by disciplines that are very different from physics.
- For instance, the prices of financial securities, the spread of heat, animal populations, bacteria, illness, sound, and light are **modeled using the same instrument**.

Brownian motion

- Brownian motion is a phenomenon that consists of the uninterrupted and irregular movement made by small particles or grains of colloidal size, that is, particles that are far too small to be observed with the naked eye but are significantly larger than atoms when immersed in a fluid.

-
- Understanding the geometric Brownian motion model
 - Defining a standard Brownian motion
 - Addressing the Wiener process as random walk
 - Implementing a standard Brownian motion

Juan F.
Mendoza M.

Agenda

-
- Understanding the geometric Brownian motion model
 - Defining a standard Brownian motion
 - Addressing the Wiener process as random walk
 - Implementing a standard Brownian motion

Juan F.
Mendoza M.

Agenda

Defining a standard Brownian motion

There are various ways of constructing a Brownian motion model and various equivalent definitions of Brownian motion.

Let's start with the definition of a standard Brownian motion (the **Wiener process**).

The essential **properties of a standard Brownian motion** include the following:

- The standard Brownian motion starts from zero.
- The standard Brownian motion takes a continuous path.
- The increases suffered by the Brownian process are independent.
- The increases suffered by the Brownian process in the time interval, dt , indicate a Gaussian distribution, with an average that is equal to zero and a variance that is equal to the time interval, dt .

Sum of a large number of small increments

- Based on these properties, we can consider the process as **the sum of a large number of extremely small increments.**
- After choosing two instants, t and s , the random variable, $Y(s) - Y(t)$, follows a normal distribution, with a mean of $\mu(s-t)$ and variance of $\sigma^2 (s-t)$, which we can represent using the following equation:

$$Y(s) - Y(t) \sim N(\mu(s - t), \sigma^2 (s - t))$$

The hypothesis of normality

- The hypothesis of normality is very important in the context of linear transformations.
- In fact, the standard Brownian motion takes its name from the type of distribution that is a **standard normal distribution**, with parameters of $\mu = 0$ and $\sigma^2 = 1$.
- Therefore, it can be said that the Brownian motion, $Y(t)$, with a unit mean and variance can be represented as a linear transformation of a standard Brownian motion, according to the following equation:

The risk of the negativity

$$Y(t) = Y(0) + \mu * t + \sigma * Z(t)$$

- In the previous equation, we can observe the following:
- $Z(t)$ is the standard Brownian motion.
- The weak point of this equation lies in the fact that the probability that $Y(t)$ **assuming a negative value is positive**; in fact, since $Z(t)$ is characterized by independent increments, which can assume a negative sign, **the risk of the negativity** of $Y(t)$ is not zero.
- Now, consider the Brownian motion (the Wiener process) for **sufficiently small time intervals**.

Infinitesimal increment

- An infinitesimal increment of this process is obtained in the following form:

$$Z_{(t+dt)} - Z_{(t)} = \delta Z_t = N * \sqrt{dt}$$

- The previous equation can be rewritten as follows:

$$\frac{Z_{(t+dt)} - Z_{(t)}}{dt} = \frac{N}{\sqrt{dt}}$$

- This process is not limited in variation, and, therefore, cannot be differentiated in the context of classical analysis.
- In fact, the previous one tends to **infinity** for the interval, dt .

-
- Understanding the geometric Brownian motion model
 - Defining a standard Brownian motion
 - Addressing the Wiener process as random walk
 - Implementing a standard Brownian motion

Juan F.
Mendoza M.

Agenda

Addressing the Wiener process as random walk

- A Wiener process can be considered a **borderline case of random walk**.
- We have seen that the position of a particle at instant n will be represented by the following equation:

$$Y_n = Y_{n-1} + Z_n; n = 1, 2, \dots$$

- In the previous formula, we can observe the following:
 - Y_n is the next value in the walk.
 - Y_{n-1} is the observation in the previous time phase.
 - Z_n is the random fluctuation in that step.

Central limit theorem

- If the n random numbers, Z_n , have a **mean equal to zero and a variance equal to 1**, then, for each value of n , we can define a **stochastic process** using the following equation:

$$Y_n(t) = \frac{1}{\sqrt{n}} * \sum_k Z_k$$

- The preceding formula can be used in an iterative process.
- For very large values of n , we can write the following:

$$Y_n(s) - Y_n(t) \sim N(0, (s - t))$$

- The previous formula is due to the **central limit theorem** that we covered in Monte Carlo Simulations.

-
- Understanding the geometric Brownian motion model
 - Defining a standard Brownian motion
 - Addressing the Wiener process as random walk
 - Implementing a standard Brownian motion

Juan F.
Mendoza M.

Agenda

Implementing a standard Brownian motion - 1

```
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(4)
n = 1000
SQN = 1 / np.math.sqrt(n)
ZValues = np.random.randn(n)
Yk = 0
SBMotion=list()
```

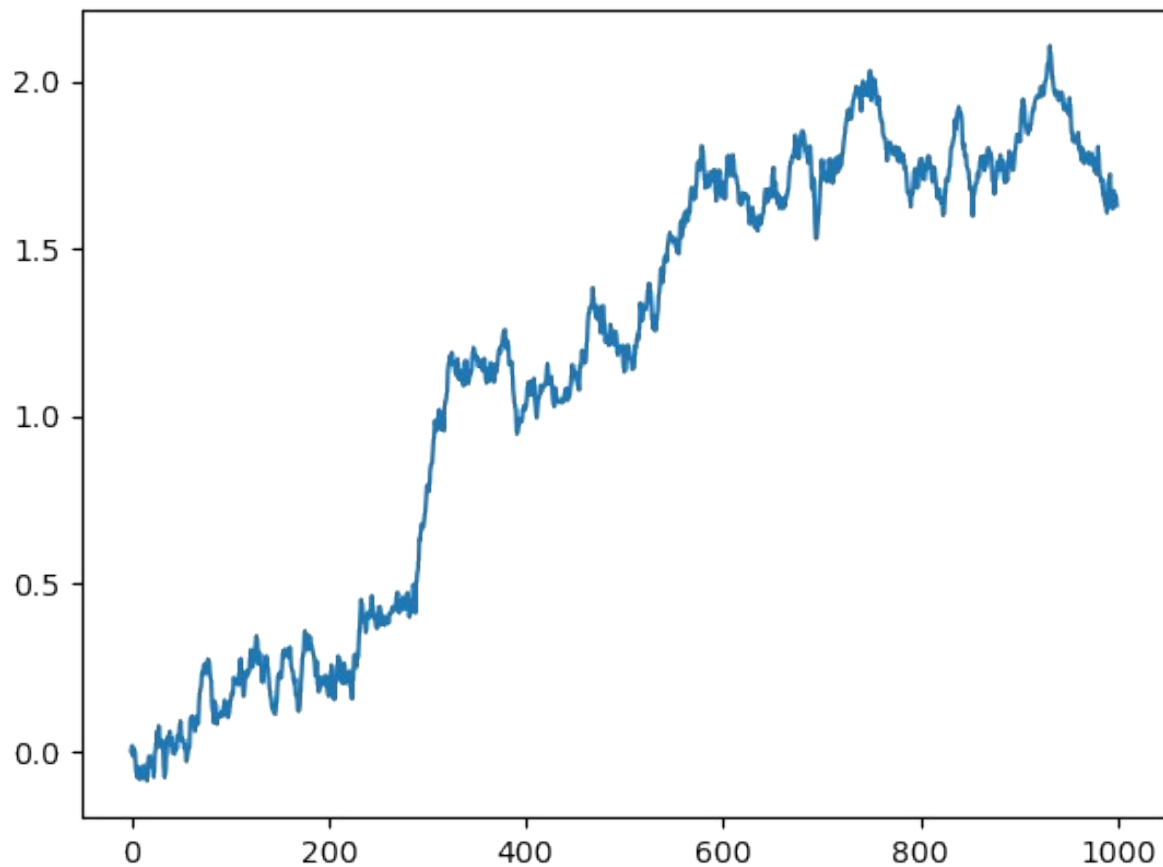
$$Y_n(t) = \frac{1}{\sqrt{n}} * \sum_k Z_k$$

Implementing a standard Brownian motion - 2

```
for k in range(n):  
    Yk = Yk + SQN * ZValues[k]  
    SBMotion.append(Yk)  
  
plt.plot(SBMotion)  
plt.show()
```

$$Y_n(t) = \frac{1}{\sqrt{n}} * \sum_k Z_k$$

Brownian motion graph



-
- Understanding the geometric Brownian motion model
 - Using Monte Carlo methods for stock price prediction
 - Studying risk models for portfolio management

Juan F.
Mendoza M.

Agenda

Using Monte Carlo methods for stock price prediction

- As we explored in Monte Carlo Simulations chapter, Monte Carlo methods simulate different **evolutions** of the process under examination using different probabilities that the event may occur under certain conditions.
- These simulations **explore the entire parameter space** of the phenomenon and return a **representative sample**.
- For each sample obtained, **measures of the quantities** of interest are carried out to **evaluate** their **performance**.
- A correct simulation means that the **average value of the result of the process converges to the expected value**.

-
- Using Monte Carlo methods for stock price prediction
 - Exploring the Amazon stock price trend
 - Handling the stock price trend as time series
 - Introducing the Black-Scholes model
 - Applying Monte Carlo simulation

Juan F.
Mendoza M.

Agenda

-
- Using Monte Carlo methods for stock price prediction
 - Exploring the Amazon stock price trend
 - Handling the stock price trend as time series
 - Introducing the Black-Scholes model
 - Applying Monte Carlo simulation

Juan F.
Mendoza M.

Agenda

Exploring the Amazon stock price trend

- The **stock market** provides an opportunity to quickly earn large amounts of money, that is, in the eyes of an inexperienced user at least.
- Exchanges on the stock market can cause **large fluctuations in price** attracting the attention of speculators from all over the world.
- In order to obtain revenues from investments in the stock market, it is necessary to have **solid knowledge** obtained from years of in-depth study of the phenomenon.
- In this context, the possibility of having a **tool to predict stock market securities** represents a need felt by all.

NASDAQ GS

- Let's demonstrate how to develop a simulation model of the stock of one of the most famous companies in the world.
- **Amazon** was founded by Jeff Bezos in the 1990s, and it was one of the first companies in the world to sell products via the internet.
- **Amazon stock** has been listed on the stock exchange since 1997 under the symbol **AMZN**.
- The historical values of AMZN stock can be obtained from various **internet sites** that have been **dealing with the stock market** over the past 10 years.
- We will refer to the performance of AMZN stock on the **NASDAQ GS stock** quote from 2010-05-19 to 2022-05-19.
- In order to get the data from 2010-05-19, we need to select 2022-05-19 on the Yahoo website as the end date.

Finance Yahoo

- Data can be downloaded in .csv format from the Yahoo Finance website at <https://finance.yahoo.com/quote/AMZN/history/>.
- In the following screenshot, you can see the Yahoo Finance section for AMZN stock with a highlighted button to download the data:

Amazon data on Yahoo Finance

Amazon.com, Inc. (AMZN)

NasdaqGS - NasdaqGS Real Time Price. Currency in USD

☆ Add to watchlist

2,142.25 -165.12 (-7.16%) **2,120.06** -22.19 (-1.04%)

At close: May 18 04:00PM EDT

Pre-Market: 8:57AM EDT

Start Trading >>

Plus500 CFD Service. Your capital is a

Summary

Chart

Conversations

Statistics

Historical Data

Profile

Financials

Analysis

Options

Holders

Sustainability

Time Period: May 19, 2021 - May 19, 2022 ▾

Show: Historical Prices ▾

Frequency: Daily ▾

Apply

Currency in USD

↓ Download

Date	Open	High	Low	Close*	Adj Close**	Volume
May 18, 2022	2,228.79	2,257.06	2,124.98	2,142.25	2,142.25	5,404,100
May 17, 2022	2,265.50	2,316.00	2,225.53	2,307.37	2,307.37	3,822,400

Selecting dataset features

- The downloaded AMZN.csv file contains a lot of features, but we will only use two of them, as follows:
 - **Date:** Date of quote
 - **Close:** Close price
- We will analyze the code, line by line, to fully understand the whole process, which will lead us to the simulation of a series of predicting scenarios of the performance of the Amazon stock price:

Predicting scenarios of the performance of the Amazon stock price - 1

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import norm
from pandas.plotting import
register_matplotlib_converters

register_matplotlib_converters()
```


Predicting scenarios of the performance of the Amazon stock price - 2

```
AmznData = pd.read_csv('AMZN.csv', header=0, usecols=['Date',  
'Close'], parse_dates=True, index_col='Date')  
print(AmznData.info())  
print(AmznData.head())  
print(AmznData.tail())  
print(AmznData.describe())  
  
plt.figure(figsize=(10,5))  
plt.plot(AmznData)  
plt.show()
```

Predicting scenarios of the performance of the Amazon stock price – Dataset info

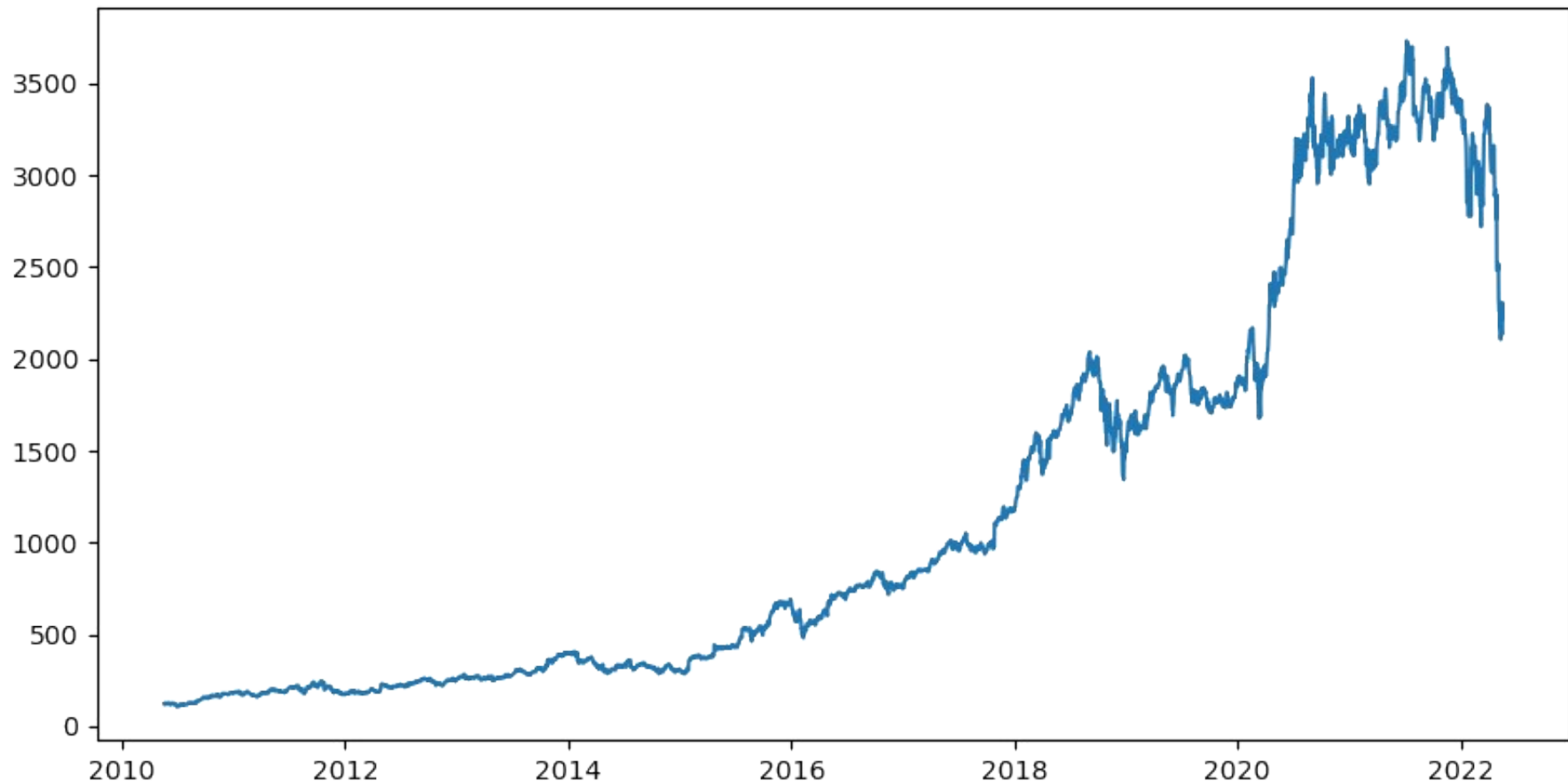
```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 3022 entries, 2010-05-19 to 2022-05-18
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype  
---  -
 0   Close   3022 non-null    float64
dtypes: float64(1)
memory usage: 47.2 KB
None
```

Date	Close
2010-05-19	124.589996
2010-05-20	119.709999
2010-05-21	122.720001
2010-05-24	122.120003
2010-05-25	124.860001

	Date	Close
	2022-05-12	2138.610107
	2022-05-13	2261.100098
	2022-05-16	2216.209961
	2022-05-17	2307.370117
	2022-05-18	2142.250000

	Close
count	3022.000000
mean	1155.464722
std	1080.197841
min	108.610001
25%	272.174995
50%	702.934998
75%	1807.959991
max	3731.409912

Amazon shares graph



-
- Using Monte Carlo methods for stock price prediction
 - Exploring the Amazon stock price trend
 - Handling the stock price trend as time series
 - Introducing the Black-Scholes model
 - Applying Monte Carlo simulation

Juan F.
Mendoza M.

Agenda

Handling the stock price trend as time series

- The trend over time of the Amazon stock price, represented in the previous graph, is configured as a **sequence of ordered data**.
- This type of data can be conveniently handled as a **time series**.
- Let's consider a simple definition: a time series contains a **chronological sequence of experimental observations of a variable**.
- This variable can relate to data of **different origins**.
- Very often, **it concerns financial data** such as unemployment rates, spreads, stock market indices, and stock price trends.

Management of future scenarios

- Dealing with the problem as a time series will allow us to extract useful information from the data in order to develop predictive models for the **management of future scenarios**.
- It may be useful to compare the trend of stock prices in the **same periods** for different years or, more simply, between **contiguous periods**.

Time series

- Let $Y_1, \dots, Y_t, \dots, Y_n$ be the elements of a time series.
- Let's start by comparing the data for two different times indicated with t and $t + 1$.
- It is, therefore, two contiguous periods; we are interested in evaluating the variation undergone by the phenomenon under observation, which can be defined by the following ratio:

$$\frac{Y_{t+1} - Y_t}{Y_t} * 100$$

Percentage change

- This percentage ratio is called a percentage change.
- It can be defined as the percentage change rate of Y of time $t + 1$ compared to the previous time, t .
- This descriptor returns information about how the data underwent a change over a period.
- The percentage change allows you to **monitor both the stock prices and the market indices**, not just comparing currencies from different countries.

Predicting scenarios of the performance of the Amazon stock price - 3

```
AmznDataPctChange = AmznData.pct_change()  
  
AmznLogReturns = np.log(1 + AmznDataPctChange)  
print(AmznLogReturns.tail(10))  
  
plt.figure(figsize=(10,5))  
plt.plot(AmznLogReturns)  
plt.show()
```

Evaluating percentage change

- To evaluate this useful descriptor, we will use the **pct_change()** function contained in the pandas library.
- This function returns the percentage change between the **current element and a previous element**.
- By default, the function calculates the percentage change from the immediately preceding row.

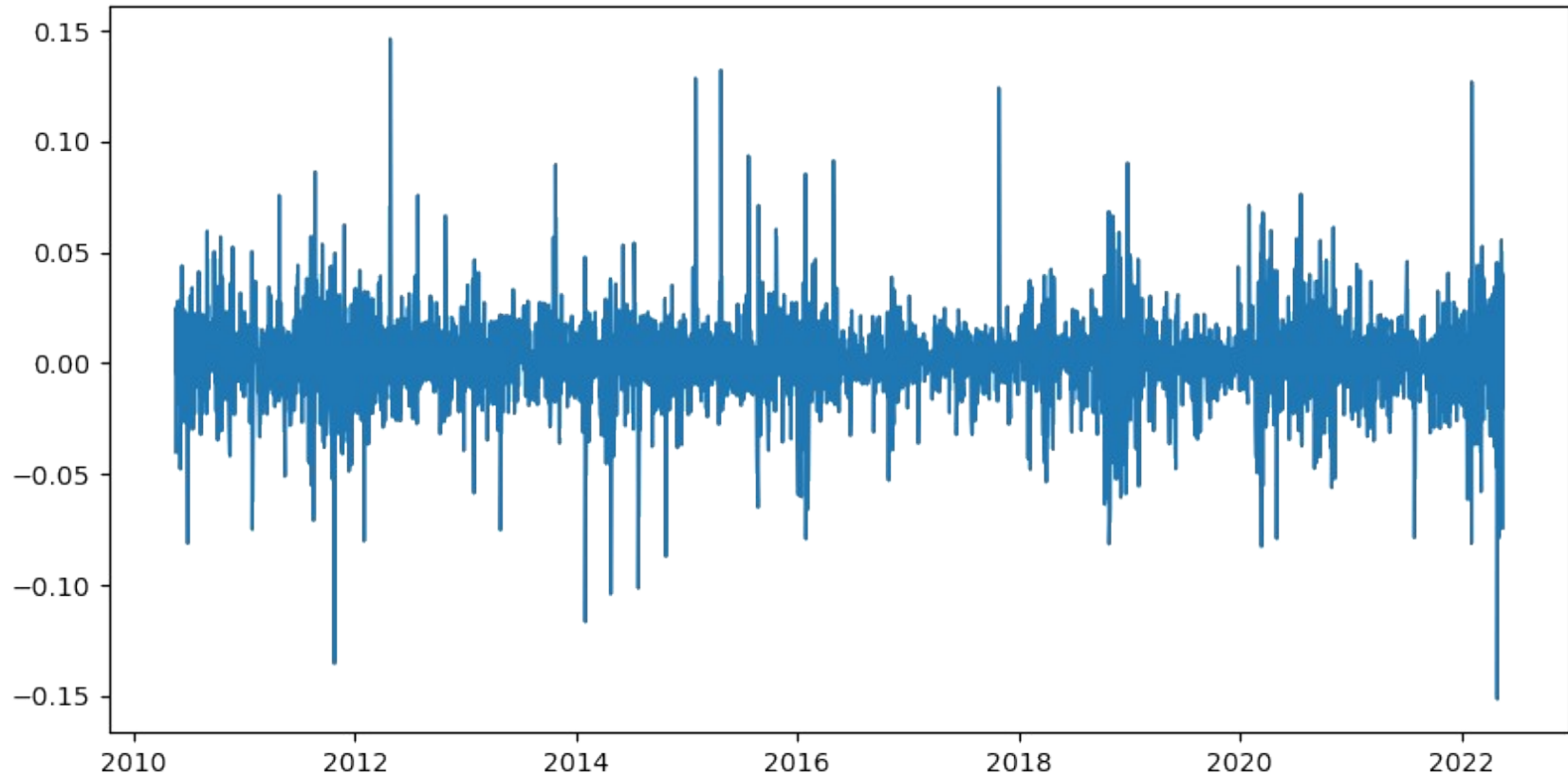
Percentage variation

- The concept of the **percentage variation** of a time series is linked to the concept of the return of a stock price.
- The **returns-based approach** allows the normalization of data, which is an operation of fundamental importance when evaluating the relationships between variables characterized by different metrics.
- We will deal with the return on a **logarithmic scale** as this choice will give us several advantages: normally distributed results; values returned (logarithm of the return) very close to the initial one (the return), at least for very small values; and additive results over time.

Logarithmic scale

	Close
Date	
2022-05-05	-0.078622
2022-05-06	-0.014141
2022-05-09	-0.053542
2022-05-10	0.000643
2022-05-11	-0.032557
2022-05-12	0.014682
2022-05-13	0.055695
2022-05-16	-0.020053
2022-05-17	0.040310
2022-05-18	-0.074252

Logarithmic values of the returns



-
- Using Monte Carlo methods for stock price prediction
 - Exploring the Amazon stock price trend
 - Handling the stock price trend as time series
 - Introducing the Black-Scholes model
 - Applying Monte Carlo simulation

Juan F.
Mendoza M.

Agenda

Introducing the Black-Scholes model

- The Black-Scholes (BS) model certainly represents the most important and revolutionary work in the history of **quantitative finance**.
- In traditional financial literature, it is assumed that almost all financial asset prices (stocks, currencies, and interest rates) are driven by a **Brownian drift motion**.
- This model assumes that the expected return of an **asset is equal to the non-risky interest rate, r** .
- This approach is capable of simulating returns on a **logarithmic scale of an asset**.

Calculating the return

- Suppose we observe an asset in the instants: $t(0), t(1), \dots, t(n)$.
- We note, using $s(i) = S(t_i)$, the value of an asset at $t(i)$.
- Based on these hypotheses, we can calculate the return using the following equation:

$$y(i) = \frac{[s(i) - s(i - 1)]}{s(i - 1)}, i = 1, 2, \dots, n$$

- Suppose we observe an asset in the instants: $t(0), t(1), \dots, t(n)$.
- We note, using $s(i) = S(t_i)$, the value of an asset at $t(i)$.
- Based on these hypotheses, we can calculate the return using the following equation:

$$x(i) = \ln s(i) - \ln s(i - 1), i = 1, 2, \dots, n$$

Applying the BS approach to Brownian geometric motion

- By applying the BS approach to Brownian geometric motion, the stock price will satisfy the following stochastic differential equation:

$$dS(t) = \mu * S(t) * dt + \sigma * S(t) * dB(t)$$

- In the previous equation, $dB(t)$ is a standard Brownian motion and μ and σ are real constants.
- The previous equation is valid where the hypothesis that $s(i) - s(i - 1)$ is small, and this happens when the stock prices undergo slight variations.
- This is because $\ln(1 + z)$ is roughly equal to z if z is small.
- The analytical solution of the previous equation is the following equation:

$$S(t) = S(0) * e(\alpha(t) + \sigma * B(t))$$

Passing the equation on a logarithmic scale

- By passing the previous equation on a logarithmic scale, we obtain the following equation:

$$\ln \frac{S(t)}{S(0)} = \alpha(t) + \sigma * B(t)$$

- In the previous equation, we can observe the following:
 - α is the drift.
 - $B(t)$ is a standard Brownian motion.
 - σ is the standard deviation.

Drift

- We introduced the concept of **drift, which represents the trend of a long-term asset in the stock market.**
- To understand drift, we will use the concept of river currents.
- If we pour liquid color into a river, it will spread by following the direction imposed by the river's current.
- Similarly, drift represents the tendency of a stock to follow the trend of a long-term asset.

-
- Using Monte Carlo methods for stock price prediction
 - Exploring the Amazon stock price trend
 - Handling the stock price trend as time series
 - Introducing the Black-Scholes model
 - Applying Monte Carlo simulation

Juan F.
Mendoza M.

Agenda

Applying Monte Carlo simulation

- Using the BS model discussed in the previous section, we can evaluate the daily price of an asset starting from that of the previous day multiplied by an exponential contribution based on a coefficient, r .
- This coefficient is a periodic rate of return.
- It translates into the following equation:

$$StockPrice(t) = StockPrice(t - 1) * e^r$$

Daily return

- The second term in the previous equation, e^r , is called the **daily return**, and, according to the BS model, it is given by the following formula:

$$e^{(\alpha(t) + \sigma * B(t))}$$

- There is no way to predict the rate of return of an asset.
- The only way to represent it is to consider it as a random number.
- So, to predict the price trend of an asset, we can use a model based on random movement such as that represented by BS equations.

The fixed drift rate and the random stochastic variable

- The BS model assumes that changes in the stock price depend on the expected return over time.
- The daily return has two terms: the fixed drift rate and the random stochastic variable.
- The two terms provide for the certainty of movement and uncertainty caused by volatility.
- To calculate the drift, we will use the expected rate of return, which is the most likely rate to occur, using the historical average of the log returns and variance, as follows:

$$\text{drift} = \text{mean}(\log(\text{returns})) - 0.5 * \text{variance}(\log(\text{returns}))$$

Calculating the drift for the return of the Amazon

- According to the previous equation, the daily change rate of the asset is the mean of the returns, which are less than half of the variance over time.
- We continue our work, calculating the drift for the return of the Amazon security calculated in the Handling the stock price trend as time series section.

Calculating the drift

- To evaluate the **drift**, we need the **mean and variance** of the returns.
- Since we also calculate the standard deviation, we will need the calculation of the **daily return**.
- Three numpy functions were used:
 - **mean()**: This computes the arithmetic mean along the specified axis and returns the average of the array elements.
 - **var()**: This computes the variance along the specified axis. It returns the variance of the array elements, which is a measure of the spread of a distribution.
 - **std()**: This computes the standard deviation along the specified axis.
- Now we can calculate the drift.

Predicting scenarios of the performance of the Amazon stock price - 4

```
MeanLogReturns = np.array(AmznLogReturns.mean())
VarLogReturns = np.array(AmznLogReturns.var())
StdevLogReturns = np.array(AmznLogReturns.std())

Drift = MeanLogReturns - (0.5 * VarLogReturns)
print("Drift = ", Drift)
```

Drift results

- The following result is returned:

```
Drift = [0.00073783]
```

- This is the **fixed part of the Brownian motion**.
- The drift returns the **annualized change** in the expected value and **compensates for the asymmetry** in the results compared to the **straight Brownian motion**.

Standard deviations

- To evaluate the **second component of the Brownian motion**, we will use the random stochastic variable.
- This corresponds to the **distance between the mean and the events**, expressed as the number of **standard deviations**.
- Before doing this, we need to set the number of **intervals and iterations**.
- The number of **intervals will be equal to the number of observations**, which is 3,022, while the **number of iterations**, which represents the number of simulation models that we intend to develop, is 20.

Random contribution for simulations

- Before generating random values, it is recommended that you set the seed to make the experiment reproducible.
- Now, we can generate the random distribution with `norm.ppf()` function.
- A 3022 x 20 matrix is returned, containing the random contribution for the 20 simulations that we want to perform and for the 3,022 time intervals that we want to consider.
- Recall that these intervals correspond to the daily prices of the last 10 years.

Random contribution functions

- Two functions were used:
 - **norm.ppf()**: This SciPy function gives the value of the variate for which the cumulative probability has the given value.
 - **np.random.rand()**: This NumPy function computes random values in a given shape. It creates an array of the given shape and populates it with random samples from a uniform distribution over $[0, 1]$.

Daily return interpretation

- So, we will calculate the daily return.
- The **daily return** is a measure of the change that occurred in a stock's price.
- It is expressed as a **percentage of the previous day's closing price**.
- A **positive** return means the stock has **grown** in value, while a **negative** return means it has **lost** value.
- The **np.exp()** function was used to calculate the exponential of all elements in the input array.

Predicting scenarios of the performance of the Amazon stock price - 5

```
NumIntervals = 3022
Iterations = 20

np.random.seed(7)
SBMotion = norm.ppf(
    np.random.rand(NumIntervals, Iterations)
)

DailyReturns = np.exp(Drift + StdevLogReturns * SBMotion)
```


Initializing the prediction array

- After long preparation, we have arrived at a crucial moment.
- We will be able to carry out **predictions** based on the **Monte Carlo method**.
- The first thing to do is to recover the starting point of our simulation.
- Since we want to predict the trend of Amazon stock prices, we recover the first value present in the AMZN.csv file.
- The pandas **iloc()** function is used to return a pure integer using location-based indexing for selection.
- Then, we will initialize the array that will contain the predictions.
- The **numpy zeros_like()** function is used to return an array of zeros with the same shape and type as a given array.

Updating the predictions

- Now, we will set the starting value of the StockPrice array.
- To update the predictions of the Amazon stock prices, we will use a for loop that iterates for a number that is equal to the time intervals we are considering.
- For the update, we will use the **BS model** according to the following equation:

$$\begin{aligned} \text{StockPrice}(t) &= \text{StockPrice}(t - 1) * e^r \\ &= \text{StockPrice}(t - 1) * e^{(\alpha(t) + \sigma * B(t))} \end{aligned}$$

Predicting scenarios of the performance of the Amazon stock price - 6

```
StartStockPrices = AmznData.iloc[0]

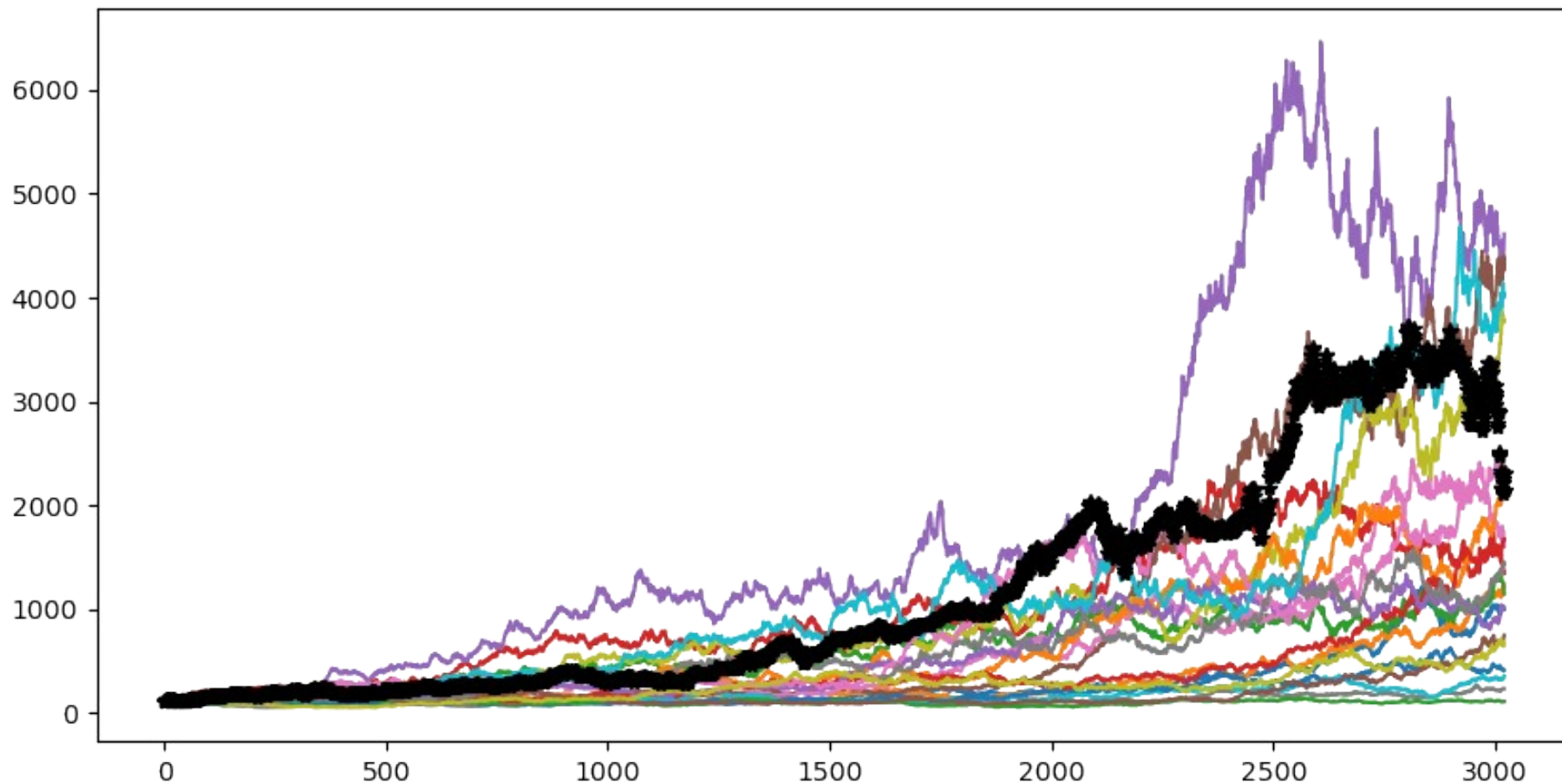
StockPrice = np.zeros_like(DailyReturns)

StockPrice[0] = StartStockPrices

for t in range(1, NumIntervals):
    StockPrice[t] = StockPrice[t - 1] * DailyReturns[t]

plt.figure(figsize=(10,5))
plt.plot(StockPrice)
```

Amazon trend graph



Amazon trend graph interpretation

- In the previous graph, the curve highlighted in black represents the trend of the Amazon stock prices in the last 10 years.
- The other curves are our simulations.
- We can see that some curves move away from the expected curve, while others appear much closer to the actual trend.

-
- Understanding the geometric Brownian motion model
 - Using Monte Carlo methods for stock price prediction
 - Studying risk models for portfolio management

Juan F.
Mendoza M.

Agenda

Studying risk models for portfolio management

- Having a **good risk measure** is of fundamental importance in finance, as it is one of the main tools for evaluating financial assets.
- This is because it allows you to **monitor securities and provides a criterion for the construction of portfolios.**
- One measure, more than any other, that has been widely used over the years is **variance**.

-
- Studying risk models for portfolio management
 - Using variance as a risk measure
 - Introducing the value-at-risk metric
 - Estimating the VaR for some NASDAQ assets

Juan F.
Mendoza M.

Agenda

-
- Studying risk models for portfolio management
 - Using variance as a risk measure
 - Introducing the value-at-risk metric
 - Estimating the VaR for some NASDAQ assets

Juan F.
Mendoza M.

Agenda

Using variance as a risk measure

- The advantage of a diversified portfolio in terms of risk and the expected value allows us to find the right allocation for the **securities**.
- Our aim is to obtain the **highest expected value** at the same risk or to **minimize the risk** of obtaining the same expected value.
- To achieve this, it is necessary to trace the concept of risk back to a **measurable quantity**, which is generally referred to as the **variance**.
- Therefore, by maximizing the expected value of the portfolio returns for each level of variance, it is possible to reconstruct a curve called the **efficient frontier**, which determines the maximum expected value that can be obtained with the securities available for the construction of the portfolio for each level of risk.

Mean and variance

- The minimum variance portfolio represents the portfolio with the **lowest possible variance value** regardless of the expected value.
- This parameter has the purpose of **optimizing the risk** represented by the variance of the portfolio.
- Tracing the risk exclusively to the measure of variance is optimal only if the distribution of returns is **normal**.
- In fact, the normal distribution enjoys some properties that make the variance a measure that is enough to represent the risk.
- It is completely determinable through only two parameters (**mean and variance**).
- It is, therefore, enough to know the mean and the variance to determine any other point of the distribution.

-
- Studying risk models for portfolio management
 - Using variance as a risk measure
 - Introducing the value-at-risk metric
 - Estimating the VaR for some NASDAQ assets

Juan F.
Mendoza M.

Agenda

Introducing the value-at-risk metric

- Consider the variance as the only risk measure in the case of non-normal and limiting values.
- A risk measure that has been widely used for over two decades is **Value at Risk (VaR)**.
- The birth of VaR was linked to the growing need for financial institutions to manage risk and, therefore, be able to measure it.
- This is due to the increasingly **complex** structure of financial markets.

Risk as a financial assets

- Actually, this measure was not introduced to stem the limits of variance as a risk measure since an approach to calculate the VaR value starts precisely from the assumptions of **normality**.
- However, to make it easier to understand, let's enclose the overall risk of a security into a single number or a portfolio of financial assets by adopting a single metric for different types of risk.

VaR

- In the financial context, the **VaR** is an estimate, given a confidence interval, of how high the losses of a security or portfolio may be in each time horizon.
- The VaR, therefore, focuses on the left tail of the distribution of returns, where events with a low probability of realization are located.
- Indicating the **losses** and not the dispersion of the returns around their expected value makes it a measure closer to the common idea of risk than variance.

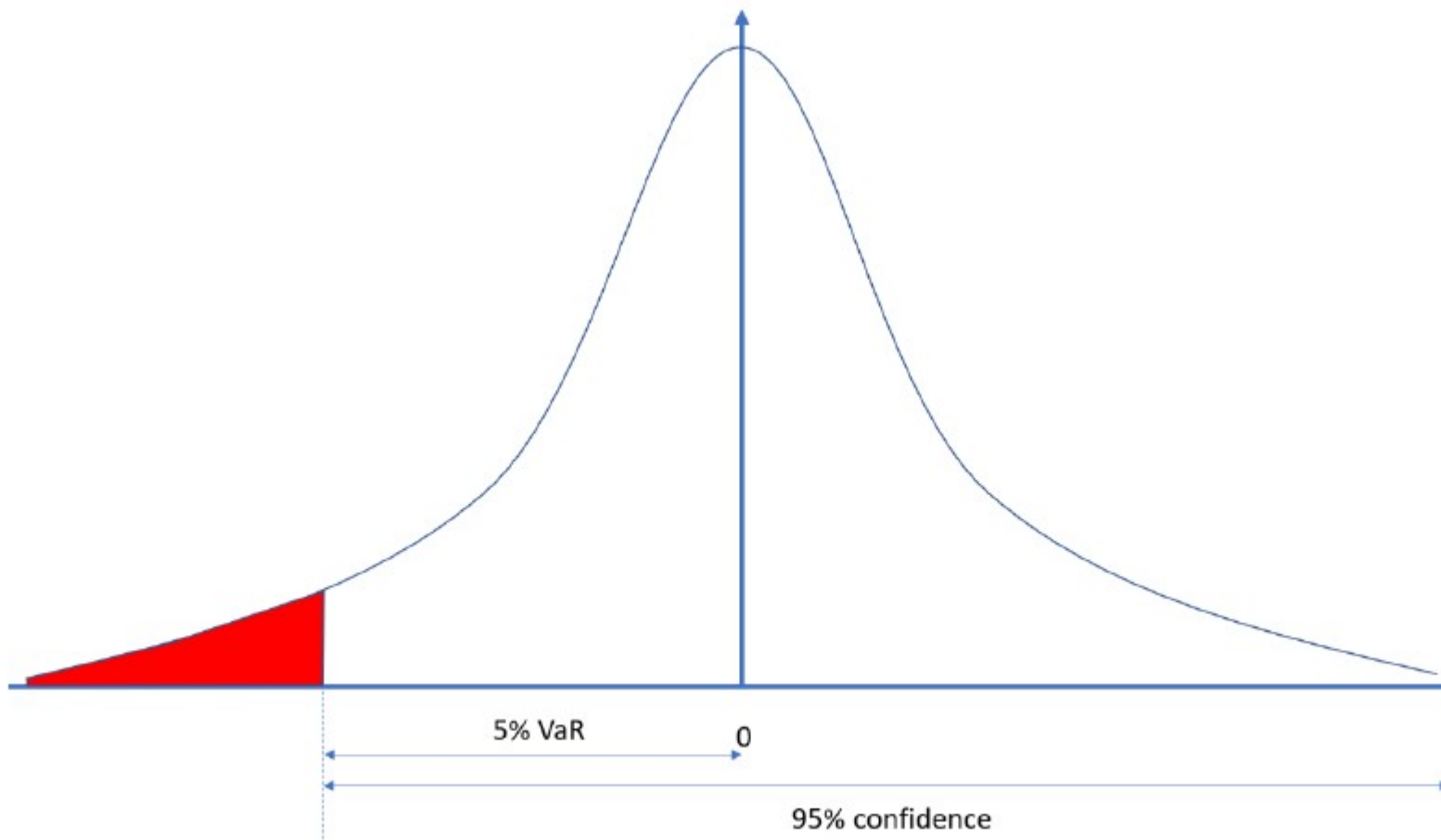
J.P. Morgan

- J.P. Morgan is credited as the bank that made VaR a widespread measure.
- In 1990, the president of J.P. Morgan, Dennis Weatherstone, was dissatisfied with the lengthy risk analysis reports he received every day.
- He wanted a simple report that summarized the bank's total exposure across its entire trading portfolio.
- After calculating the VaR, we can say that, with a probability given by the confidence interval, we will not lose more than the VaR of the portfolio in the next N days.
- VaR is the level of loss that will not be exceeded with a probability given by the confidence interval.

VaR example

- For example, a VaR of €1 million over a year with a 95% confidence level means that the maximum loss for the portfolio for the next year will be €1 million in 95% of cases.
- Nothing tells us what happens to the remaining 5% of cases.
- The following graph shows the probability distribution of portfolio returns with the indication of the value of the VaR:

Probability distribution of the portfolio returns



VaR characteristics

- VaR is a function of the following two parameters:
 - Time horizon
 - Level of confidence
- Some characteristics of VaR must be specified:
 - VaR does not describe the worst loss.
 - VaR says nothing about the distribution of losses in the left tail.
 - VaR is subject to sampling errors.
- The sampling error tells us how much the sampled value deviates from the real population value.
- This deviation is because the sample is not representative of the population or has distortions.

How bad can financial investments go?

- VaR is a widely used risk measure that summarizes, in a single number, important aspects of the **risk** of a portfolio of financial instruments.
- It has the same unit of measurement as the returns of the portfolio on which it is calculated, and it is simple to understand, answering the simple question: How bad can financial investments go?
- Let's now examine a practical case of calculating the VaR.

-
- Studying risk models for portfolio management
 - Using variance as a risk measure
 - Introducing the value-at-risk metric
 - Estimating the VaR for some NASDAQ assets

Juan F.
Mendoza M.

Agenda

Estimating the VaR for some NASDAQ assets

- **NASDAQ** is one of the most famous stock market indices in the world.
- Its name is an acronym for the National Association of Securities Dealers Quotation.
- This is the index that represents the stocks of the technology sector in the US.
- Thinking of NASDAQ in the investor's mind, the brands of the main technological and social houses of the US can easily emerge.
- Just think of companies such as Google, Amazon, Facebook, and many others; they are all covered by the NASDAQ listing.

Estimating the risk associated - 1

```
import datetime as dt
import numpy as np
import pandas_datareader.data as wb
import matplotlib.pyplot as plt
from scipy.stats import norm
```

Estimating the risk associated - 2

```
StockList =  
['ADBE', 'CSCO', 'IBM', 'NVDA', 'MSFT', 'HPQ']  
StartDay = dt.datetime(2021, 1, 1)  
EndDay = dt.datetime(2021, 12, 31)
```


Tickers

- Six tickers have been included in a DataFrame.
- A ticker is an abbreviation used to uniquely identify the shares listed on the stock exchange of a particular security on a specific stock market.
- It is made up of letters, numbers, or a combination of both.
- The tickers are used to refer to six leading companies in the global technology sector:

Tech tickers

- ADBE: Adobe Systems Inc. – one of the largest and most differentiated software companies in the world.
- CSCO: Cisco Systems Inc. – the production of Internet Protocol (IP)-based networking and other products for communications and information technology.
- IBM: International Business Machines – the production and consultancy of information technology-related products.
- NVDA: Nvidia Corp. – visual computing technologies. This is the company that invented the GPU.
- MSFT: Microsoft Corp. – this is one of the most important companies in the sector, as well as one of the largest software producers in the world by turnover.
- HPQ: HP Inc. – the leading global provider of products, technologies, software, solutions, and services to individual consumers and large enterprises.

Estimating the risk associated - 3

```
StockData = wb.DataReader(StockList, 'yahoo',  
StartDay, EndDay)  
StockClose = StockData["Adj Close"]  
print(StockClose.describe())
```

Statistics of the portfolios

Symbols	ADBE	CSCO	IBM	NVDA	MSFT	HPQ
count	252.000000	252.000000	252.000000	252.000000	252.000000	252.000000
mean	560.613652	52.095236	123.035105	195.125132	273.740511	29.639194
std	76.180505	4.703257	8.437241	58.725245	37.288644	3.272012
min	421.200012	42.510708	105.371773	115.811600	209.563217	23.323399
25%	488.294998	50.025710	115.873182	143.043198	241.360935	27.649185
50%	569.324982	52.532499	126.282703	191.977417	274.920120	29.136092
75%	632.419998	55.213575	130.241081	222.354874	299.970924	31.324860
max	688.369995	63.152836	137.747864	333.662292	341.606354	37.832893

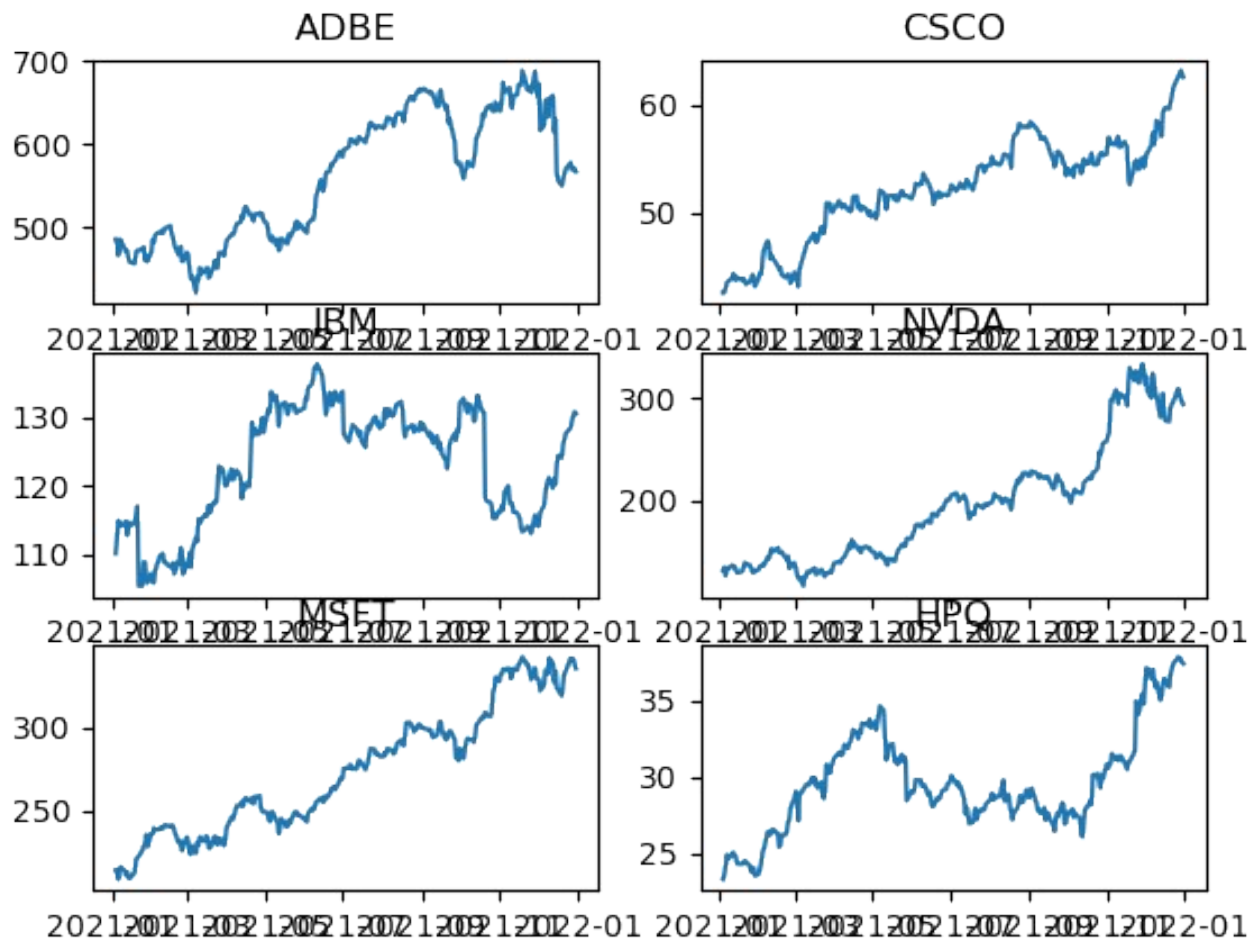
Estimating the risk associated - 4

```
fig, axs = plt.subplots(3, 2)
axs[0, 0].plot(StockClose['ADBE'])
axs[0, 0].set_title('ADBE')
axs[0, 1].plot(StockClose['CSCO'])
axs[0, 1].set_title('CSCO')
axs[1, 0].plot(StockClose['IBM'])
axs[1, 0].set_title('IBM')
axs[1, 1].plot(StockClose['NVDA'])
axs[1, 1].set_title('NVDA')
axs[2, 0].plot(StockClose['MSFT'])
axs[2, 0].set_title('MSFT')
axs[2, 1].plot(StockClose['HPQ'])
axs[2, 1].set_title('HPQ')
```

Estimating the risk associated - 5

```
plt.figure(figsize=(10,5))  
plt.plot(StockClose)  
plt.show()
```

Graphs of the statistics



Estimating the risk associated - 6

```
StockReturns = StockClose.pct_change()  
print(StockReturns.tail(15))
```


The stock returns DataFrame

Symbols	ADBE	CSCO	IBM	NVDA	MSFT	HPQ
Date						
2021-12-10	0.034589	0.029540	0.004208	-0.009577	0.028340	0.007745
2021-12-13	0.005883	-0.010802	-0.012169	-0.067455	-0.009167	-0.031567
2021-12-14	-0.065988	-0.014332	0.009626	0.006250	-0.032587	0.009070
2021-12-15	0.025160	0.037390	-0.005252	0.074884	0.019218	0.018539
2021-12-16	-0.101915	0.006341	0.022906	-0.068026	-0.029135	0.012135
2021-12-17	-0.016693	0.002487	0.011673	-0.020643	-0.003386	-0.003542
2021-12-20	-0.012342	-0.001654	-0.002669	-0.002950	-0.012014	-0.012032
2021-12-21	0.014097	0.011100	0.015032	0.048919	0.023069	0.024633
2021-12-22	0.011587	0.008193	0.006048	0.011178	0.018057	0.007023
2021-12-23	0.010000	0.012189	0.006782	0.008163	0.004472	0.009925
2021-12-27	0.014150	0.018304	0.007579	0.044028	0.023186	0.011952
2021-12-28	-0.014402	0.001734	0.007674	-0.020133	-0.003504	-0.003937
2021-12-29	-0.000123	0.006768	0.005429	-0.010586	0.002051	0.000790
2021-12-30	0.002178	-0.005316	0.004200	-0.013833	-0.007691	-0.006056
2021-12-31	-0.006082	-0.003930	-0.001867	-0.005915	-0.008841	-0.002119

Estimating the risk associated - 7

```
PortfolioValue = 10000000000.00  
ConfidenceValue = 0.95  
MeanStockRet = np.mean(StockReturns)  
StdStockRet = np.std(StockReturns)
```

The mean and standard deviation of returns

- To start, we set the value of our portfolio; it is a **billion dollars**.
- These figures should not frighten you.
- For a bank that manages numerous investors, achieving this investment value is not difficult.
- So, we set the **confidence interval**.
- Previously, we said that VaR is based on this value.
- Subsequently, we started to calculate some fundamental quantities for the VaR calculation.
- I am referring to the mean and standard deviation of returns.
- To do this, we used the related numpy functions: `np.mean ()` and `np.std`.

Estimating the risk associated - 8

```
WorkingDays2021 = 252.  
AnnualizedMeanStockRet = MeanStockRet /  
WorkingDays2021  
AnnualizedStdStockRet = StdStockRet /  
np.sqrt(WorkingDays2021)
```

Working days

- Previously, we saw that the data extracted from the finance section of the Yahoo website contained **252 records**.
- This is the number of **working days** of the stock exchange in 2021, so we set this value.
- So, let's move on to **annualizing** the mean and the standard deviation just calculated.
- This is because we want to calculate the **annual risk index of the stocks**.
- For the annualization of the average, it is enough to **divide by the number of working days**, while for the standard deviation, we must **divide by the square root of the number of working days**.

Estimating the risk associated - 9

```
INPD = norm.ppf(1 - ConfidenceValue,  
AnnualizedMeanStockRet, AnnualizedStdStockRet)  
VaR = PortfolioValue * INPD
```

Calculating the inverse normal probability distribution

- To start, we calculate the inverse normal probability distribution with a risk level of 1 for the **confidence, mean, and standard deviation**.
- This technique involves the construction of a probability distribution starting from the three parameters we have mentioned.
- In this case, we work backward, starting from some distribution statistics, and try to reconstruct the starting distribution.
- To do this, we use the **norm.ppf()** function of the SciPy library.

PPF

- The `norm()` function returns a normal continuous random variable.
- The acronym, `ppf`, stands for **percentage point function**, which is another name for the **quantile function**.
- The quantile function, associated with a probability distribution of a random variable, specifies the value of the random variable so that the probability that the variable is less than or equal to that value is equal to the given probability.

Calculating VaR

- At this point, the **VaR** is calculated by multiplying the inverse normal probability distribution obtained by the value of the portfolio.
- To make the value obtained more readable, it was rounded to the first two decimal places.
- Finally, the results obtained were printed, one for each row, to make the comparison simple.

Estimating the risk associated - 10

```
RoundVaR = np.round_(VaR, 2)

for i in range(len(StockList)):
    print("Value-at-Risk for", StockList[i], "is  
equal to ", RoundVaR[i])
```

Estimating the risk associated - results

```
Value-at-Risk for ADBE is equal to -1901394.1  
Value-at-Risk for CSC0 is equal to -1244064.07  
Value-at-Risk for IBM is equal to -1485930.72  
Value-at-Risk for NVDA is equal to -2922034.66  
Value-at-Risk for MSFT is equal to -1358218.17  
Value-at-Risk for HPQ is equal to -2020465.58
```

Result interpretation

- The stocks that returned the **highest risk** were HP and IBM, while the one that returned the **lowest risk** was the Microsoft stock.

References

- Giuseppe Ciaburro. *Hands-On Simulation Modeling with Python*. Packtpub, 2020