Introduction to Quantitative Finance

Pricing theory and practice in Python

2016

In the world of finance, assets are bought and sold every second. In this course, you will learn the general principles behind the pricing of such financial assets and focus in particular on options – which form an important class of assets traded on exchanges. You will start with the general concept of arbitrage free pricing and see how it leads to the Black-Scholes formula. You will also learn and implement some well known numerical pricing techniques such as the binomial and trinomial tree model.

This course was developed in close partnership with IMC, a technology-driven trading firm with offices in Amsterdam, Chicago and Sydney. `http://imc.com`

Author	Thibaut Lienart (Cambridge Spark)
Special thanks	Dr. Heiko Schaefer (IMC)
Editor	Raphaël Proust (Cambridge Spark)
Acknowledgements	Sam Berry (UBS), Maxime Lienart (BNY Mellon), Dr. Alexander Vervuurt (OxAM)

Contents

1. Preliminaries

1.1. Setting up

Please go to `http://gitlab.cambridgespark.com/pub/introQF` and follow the instructions there to get started. At the end of the setup you should have an `introQF` folder on your computer containing

- a file `introQF_library.py`
- a file `introQF_pythonWarmup.ipynb`
- a skeleton notebook `introQF.ipynb` (on the day)

and you should be able to

- open a terminal in the directory `introQF`,
- open Jupyter using the command `jupyter notebook`,
- open the warmup notebook.

1.2. Crash course in basic probability

There is essentially one notion of probability theory that you will need to understand to follow this course: *expected values*. It is presented below.

Random variables

In finance, unknown values are modelled as *random variables*. For example, the evolution of the price of a commodity, such as oil, can be modelled as a sequence of random variables (each corresponding to the price at a different time). A random variable has a *range of possible values*: the values that the random variables can take.

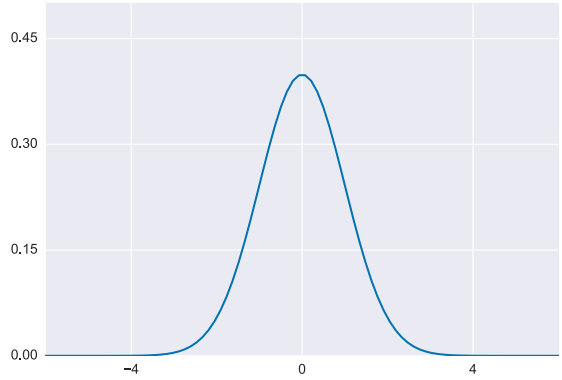


Figure 1.1.: Pdf of a standard Normal distribution.

A *weight* (probability) is associated to each of those possible values; it represents how likely the value is to be taken. These weights cannot be negative and are *normalised* so that they sum up to 1 (or, for a continuous range, they integrate to 1).

The *probability distribution function* (pdf) of a random variable is the function that associates, to each possible value, its associated weight. It is denoted as p. A classical example is the outcome of a fair coin flip. The range of possible values is $\{H, T\}$ each with a weight of 0.5 (in other words: a 50% chance). The pdf of the outcome of the fair coin flip is

Outcome	Probability
H	0.5
T	0.5

A random variable can also have a continuous range of possible values in which case the pdf is a continuous function. A well known example is the standard *Normal distribution* illustrated as follows:

In many situations, however, you cannot know what the true probability distribution of a random variable is. In such a case, you can model it based on observations. This is called *fitting* or *adjusting* a statistical model. You will use fitting to model the evolution of the price of financial assets.

Expected value

Quiz

Say I flip a coin and if it lands head I give you £10, but if it lands tails I give you nothing. What do you expect to gain on average?
Answer: intuitively, your *expected* gain is £5

To compute the expected gain of a bet you average the different possible values weighted by how likely each outcome is.

Definition

The *expected value* (EV) of any function f of a random variable X with a probability distribution function p is defined by

$$\mathbb{E}_p[f(X)] = \sum_x f(x)p(x)$$

where the x ranges over all possible outcomes for the random variable. Note that, if the range of possible values is continuous, this sum becomes an integral.

Let's apply that equation to the bet above. The function f associates £10 to the outcome H (Head) and £0 to the outcome T (Tails). The pdf p associates 0.5 to each outcome. The expected gain is computed as follows:

$$\begin{array}{rcl} \text{expected gain} & = & \sum_{x\in\{H,T\}} f(x)p(x) \\ & = & f(H)p(H) + f(T)p(T) \\ & = & 10 \times 0.5 + 0 \times 0.5 \;=\; 5 \end{array}$$

Example

Say I flip two coins in a row and, if both land heads I give you £10, if one lands heads I give you £5, if none land heads I give you nothing. What is your expected gain?
Answer: There are four possible values: $\{HH, HT, TH, TT\}$ each with a $1/4$ probability. Since there are so few values (and the outcomes are so simple), it is easy to compute the result by hand. So that you learn something new, let's do it in Python instead.
Numpy's `dot` function provides exactly the right functionality: it multiplies elements of two vectors point-wise, and then sums the results.

```
#outcomes:       {HH    HT    TH    TT  }
probs = np.array([1./4, 1./4, 1./4, 1./4])
gains = np.array([10  , 5   , 5   , 0   ])

print("Expected gains: {} GBP".format(np.dot(probs,gains)))
```

Linearity of the expected value

One important property of the expected value is that it is *linear*. Consider arbitrary constants a, b and two functions f, g, then

$$\mathbb{E}_p[a \times f(X) + g(X) + b] = a \times \mathbb{E}_p[f(X)] + \mathbb{E}_p[g(X)] + b.$$

In other words, if you multiply the outcome ($a \times f(X)$) the expected value is multiplied by the same number ($a \times \mathbb{E}_p[f(X)]$). And similarly, if you add to the outcome, it adds the same amount to the expected value.

You will see in the course that, to price an asset, you just need to compute a specific expected value. However, you will often want to consider combinations of assets. Linearity is useful in this case.

Proof

The proof starts with the definition of the expected value (assuming a discrete range of values for simplicity):

$$\begin{aligned}\mathbb{E}_p[a \times f(X) + g(X) + b] &= \textstyle\sum_x (a \times f(x) + g(x) + b)p(x) \\ &= a \times \textstyle\sum_x f(x)p(x) + \sum_x g(x)p(x) \\ &\quad\; + b \times \textstyle\sum_x p(x) \\ &= a \times \mathbb{E}_p[f(X)] + \mathbb{E}_p[g(X)] + b\end{aligned}$$

For the last step, remember that pdfs are normalised: $\sum_x p(x) = 1$.

Computing expected values in general

Note that it is often difficult to compute expected values exactly. It is even harder when the range of possible values is continuous. However, this is a well studied problem and for most variables you will encounter you can assume that someone (or some program) can estimate it for you.

For example, consider the following game: you have a random variable corresponding to a draw from a *standard normal distribution*. If the outcome is less than 2 then you get £10, otherwise you get nothing. What is the expected value of the game?

In this case, the outcome function is an *indicator function*:

$$f(x) = \begin{cases} 10 & \text{if } x < 2 \\ 0 & \text{otherwise} \end{cases}$$

The pdf of a standard normal distribution is given by:

$$p(x) = \exp(-x^2/2)/\sqrt{2\pi}$$

Following the definition, the expected value of the game is:

$$\begin{aligned}\mathbb{E}_p[f(X)] &= 10 \times \textstyle\int_{-\infty}^{2} \frac{\exp(-x^2/2)}{\sqrt{2\pi}} \mathrm{d}x \\ &= 10 \times \Phi(2)\end{aligned}$$

This integral, known as the *cumulative distribution function* of a standard normal random variable, is usually denoted $\Phi(a)$ if the upper bound is a (in the example above, $a = 2$). You will encounter this symbol again in the course.

There is no simple form for these integrals. However, since they appear often, there are readily available libraries that can approximate it.

2. Introduction

2.1. Setting

A *financial market* is a place where *agents* (for example you or a fund) can exchange or invest in *financial assets*. They can also invest in (or borrow from) a bank account with an interest rate which, in this course, is always fixed. The set of assets that an investor holds at a given point is called a *portfolio*. Here are some simple assets:

- **stocks**: shares of a company's ownership,
- **bonds**: shares of a company's or of a government's debt,
- **commodities**: tangible goods like gold or corn.

2.2. Your first plot

You can fetch the price of a share of a company from the internet using the function `getPrices` provided in `cca_imc_library.py`. The function returns the dates and prices associated to a symbol (AAPL for Apple, GOOG for Google, etc.). Use it as follows:

```
aapl_d, aapl = getPrices('AAPL', 2005, 2013)
goog_d, goog = getPrices('GOOG', 2005, 2013)
```

and visualise the results:

```
plt.figure(figsize=(8, 6))
plt.plot_date(aapl_d, aapl, label="AAPL", ls="-", marker='None')
plt.plot_date(goog_d, goog, label="GOOG", ls="-", marker='None')
plt.xlabel("Trading day [2005-2013]", fontsize=12)
plt.ylabel("Prices", fontsize=12)
plt.legend()
```

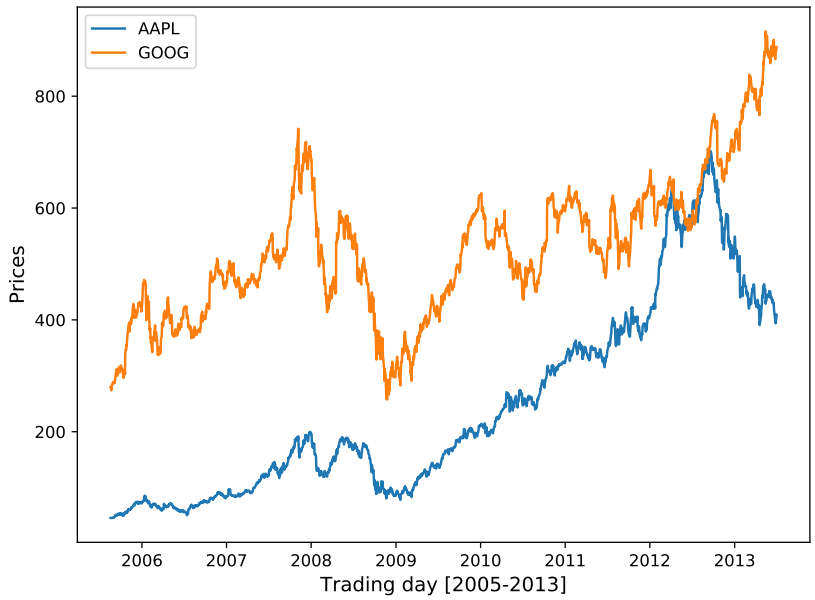


Figure 2.1.: Share prices for Apple (AAPL) and Google (GOOG) between 2005 and 2013.

2.3. Financial derivatives

On the market, agents can also agree on transactions that will take place in the future: a *forward contract*.

Example

A forward contract is an agreement between two agents to exchange an asset at a specified future time (*maturity* or *expiry* T) and at a specified price (*strike price* K). For example:

- At $t = 0$, Sam agrees to sell Laura 100 barrels of oil in $T = 2$ months for $K =$ £10k,
- At $t = T$, the contract *reaches maturity*: Laura pays Sam £10k and gets the oil.

Sam is said to have a *short position* (he agrees to *sell*) and Laura a *long position* (she agrees to *buy*).

The forward contract is an instance of a *financial derivative*. More broadly, financial derivatives are assets whose values depend on other, *underlying* assets. In the simple example above, the underlying asset is the oil barrel.

Motivation

Forward contracts raise two questions:

- what should the strike price be?
- what is the value/price of the contract at point $0 < t < T$?

The second question arises if Laura wants to exchange the contract with a third agent Tom.

Both questions will be answered in this course with the main focus on the second one: how to price derivatives. You will see how this can be done simply, for any derivative, provided the market is *arbitrage-free*.

2.4. Arbitrage-free markets

Definition

An *arbitrage* is an investment opportunity such that, starting from £0,

- you have almost no chance to lose money,
- you have a non-negligible chance to make money.

The first condition means you have probability 0 of losing money. This is different from "never" although the distinction is not too important for this course.

In other words, an arbitrage is an opportunity to make money without risk. Typically, it is assumed that the market is arbitrage-free which leads to a generic way of pricing financial assets as you will see.

Example

Consider a situation where an asset (e.g, a commodity like a barrel of oil) is traded at different prices, say £100 and £110, on different markets. You could do the following:

a. borrow £100 from the bank,
b. buy the asset for £100,
c. immediately sell the asset for £110, and
d. return £101 (the £100 principal and £1 of interests) to the bank.

Following this strategy, you would gain £9, having started with £0 and having undertook no risks.

As the example above suggests, in an arbitrage-free market, two identical assets must have the same price at any given time. This is called the *law of one price*.

Note that, in practice, there are small and short lived arbitrage opportunities in a market (e.g., price inconsistencies) but fast agents immediately exploit them until they disappear (e.g., *high frequency trading*). So, from the perspective of a standard agent, the assumption of an arbitrage-free market is reasonable.

This notion of arbitrage-free market is crucial for the rest of the course and will appear again many times. It is also quite a subtle notion. Its formalisation requires advanced mathematical tools. In this document, only the intuitive notion is used.

2.5. What you will learn

In this course you will focus on the specific problem of pricing derivatives. You will take the perspective of a *quantitative analyst* (or simply *quant*): you analyse and model the market to provide quantitative information to a *trader*.

You will start with a statistical analysis of the prices of simple assets. To this end, you will use the *LogNormal model*, one of the pillars of asset pricing. You will then be introduced to the basics of *pricing theory* and learn the generic way to price *any* asset following the so-called *risk-neutral pricing*. You will then move on to meet some standard derivatives like the *call option* and their properties.

In a second part about *computational finance*, you will apply the theory and tools introduced in the previous part on specific examples of derivatives. There, you will meet the *Black-Scholes* analytical formula to price a call option as well as some of the numerical schemes like the *binomial* and *trinomial tree*.

Simplifying assumptions

This course aims to provide intuition about the tools developed and used in the world of quantitative finance – not an exhaustive exploration of the field. In order to make this introductory course more accessible, it makes a number of (standard) simplifying assumptions:

- **Fixed interest rate**: In practice there may be a range of accessible rates that varies over time. We assume there is a single, known, fixed rate (a deterministic or "risk-free" rate).
- **No transaction fees**: In practice, every time an asset is bought or sold, the operation may (and generally does) incur a transaction fee. We assume that there is none.

- **No bid-ask spread**: The bid-ask spread is the difference between the selling price and the buying price of an asset on the market. For example a supermarket buys its apples at a lower price than it sells them: this middle-man cut is a *bid-ask spread*. We assume there is a single, identical, price.
- **No dividends**: In practice a company may pay dividends to its shareholders which affects the price of shares. We assume there are none.
- **No cost of carry**: In practice, holding an asset can incur a cost (e.g., the maintenance of an oil storage facility). We assume there are no such costs.

All those assumptions make the pricing model simpler without too much loss of generality: breaking these assumptions typically only requires adding a few terms in the mathematical models.

Remark

The no bid-ask spread assumption is very unrealistic but it helps to reason about a single price which, intuitively, you could see as the *mid* price or the price between the bid and the ask.

2.6. Summary

- A *financial market* is a place where agents (for example you or a fund) can exchange or invest in *financial assets*. They can also invest in (or borrow from) a bank account with an interest rate which, in this course, is always fixed (and positive). The set of assets that an investor holds at a given point is called a *portfolio*. Some simple assets may include *stocks*, *bonds* and *commodities*.
- *Financial derivatives* are assets whose values depend on other *underlying* assets.
- A *forward contract* is an agreement between two agents to exchange an asset (*underlying*) at a specified future time (*maturity* or *expiry*) and at a specified price (*strike price*). It is a simple derivative.
- In this document, markets will be considered *arbitrage free* – i.e., there is no possibility to make money without risk.
- In an arbitrage-free market, two identical assets must have the same price at any given time. This is known as the *law of one price*.

3. Observing prices

3.1. Log-returns

In finance, it is common to consider the *returns* of an asset: the ratio of subsequent prices of the asset. Considering a series of observation for an asset price S_i, the returns of the asset is the series S_i/S_{i-1}. A more interesting series is the *log-returns*:

$$r_i = \log \frac{S_i}{S_{i-1}}$$

These log-returns form a *stochastic process*, a series of random variables indexed by time (here the $r_1, r_2, \ldots$). In this section, you will learn to characterise the evolution of these processes.

The reason for looking at the ratio of prices is that it takes the scale out. In other words, the returns of different stocks can be compared even if their prices are very different: relative variations are more important than absolute prices.

There are theoretical reasons for considering the logarithm of these ratios which we won't cover here. The main advantage is that the log-returns can be reasonably well approximated by a Normal distribution.

Log-returns for AAPL and GOOG

You can compute and plot the log-returns of stock as illustrated below. Try the code yourself.

```
lr_aapl = np.log(aapl[1:]/aapl[0:-1])
lr_goog = np.log(goog[1:]/goog[0:-1])

plt.figure(figsize=(8, 6))
plt.plot_date(aapl_d[1:], lr_aapl, label="AAPL", ls="-", marker="None")
```

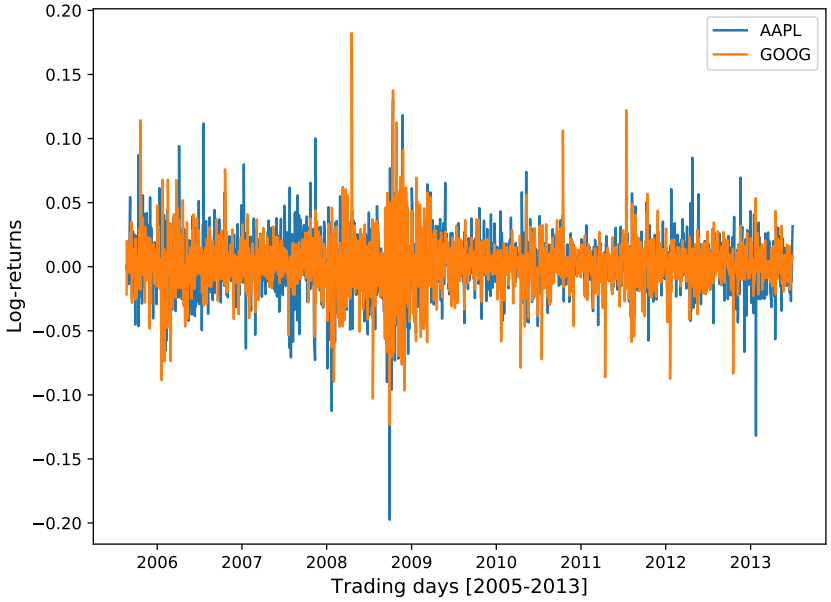


Figure 3.1.: Log-returns for Apple (blue) and Google (green) between 2005 and 2013.

```
plt.plot_date(goog_d[1:], lr_goog, label="GOOG", ls="-", marker="None")
plt.xlabel('Trading days [2005-2013]', fontsize=12)
plt.ylabel('Log-returns', fontsize=12)
plt.legend()
```

Observe the increase in *volatility* (financial term for variability) around 2008-2009, the financial crisis. Note also that the regions of higher variability are clustered together which is a well known phenomenon in finance.

Although it is a bit hard to make a precise statement just yet, it looks like the random processes at hand behind each of the two companies are similar. The similarity becomes more apparent when looking at the histograms of the log-returns. You can compute and plot them as follows:

```
sns.distplot(lr_aapl, kde=False, norm_hist=True, label="AAPL")
sns.distplot(lr_goog, kde=False, norm_hist=True, label="GOOG")
plt.legend()
```

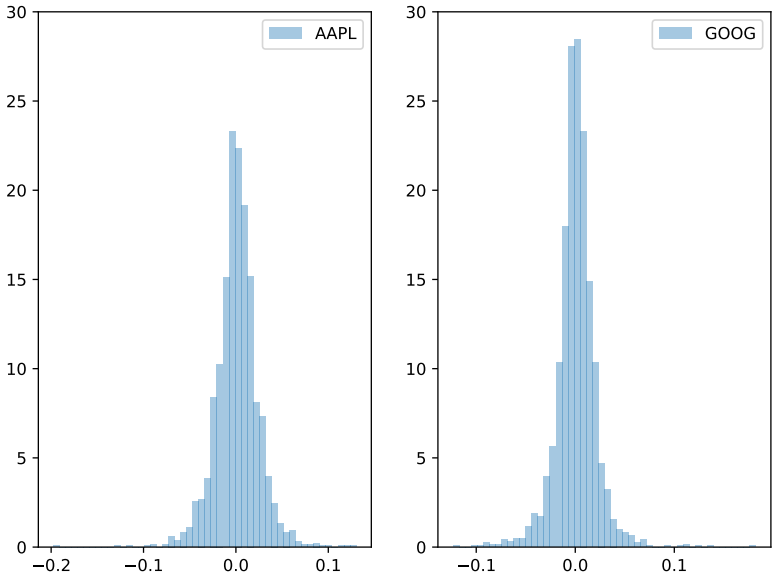


Figure 3.2.: Histograms of the log-returns for Apple (left) and Google (right) between 2005 and 2013. The shape of the two histograms is roughly the same.

```
plt.figure(figsize=(8, 6))
plt.subplot(1, 2, 1)
sns.distplot(lr_aapl, kde=False, norm_hist=True, label="AAPL")
plt.ylim([0, 30])
plt.legend()

plt.subplot(1, 2, 2)
sns.distplot(lr_goog, kde=False, norm_hist=True, label="GOOG")
plt.legend()
plt.ylim([0, 30])
```

Fitting a normal distribution

The values of the log-returns seem distributed in a similar way to the Normal distribution: they roughly have a bell shape. You can *fit* a normal dis-

tribution to these histograms. Fitting, in this context, is finding the mean μ and variance σ^2 of the Normal distribution $\mathcal{N}(\mu, \sigma^2)$ that best matches the histograms. The `scipy` library provides the `norm.fit` function for that purpose:

```
x = np.linspace(-.15, .15, 500)

plt.figure(figsize=(8, 6))
plt.subplot(1, 2, 1)

lr        = lr_aapl
mu, sigma = norm.fit(lr)
sns.distplot(lr, kde=False, norm_hist=True, label="Hist-AAPL")
plt.plot(x, norm.pdf(x, mu, sigma), label="Nfit-AAPL")
plt.legend(loc="upper left")

plt.subplot(1, 2, 2)

lr        = lr_goog
mu, sigma = norm.fit(lr)
sns.distplot(lr, kde=False, norm_hist=True, label="Hist-GOOG")
plt.plot(x, norm.pdf(x, mu, sigma), label="Nfit-GOOG")
plt.legend()
```

The fit is acceptable, but far from perfect. A different distribution might fit better but the Normal distribution is easier to work with (its expected values are easy to compute), is a first approximation (its shortcomings can be taken into account as you will see later), and leads to the *Black-Scholes model* which is as famous in finance as the Standard Model in physics or the Turing Machine in computer science.

Note that the shortcomings of the simplicity of the Normal model are taken into account in practice as you will see later.

Notation: Normal Distribution

When a random variable X is normally distributed with mean μ and variance ν we use the notation

$$X \sim \mathcal{N}(\mu, \nu)$$

Note that $\sqrt{\nu}$ is the standard deviation and a measure for the width of the distribution.

Modelling the log-returns as Normal random variables amounts to mod-

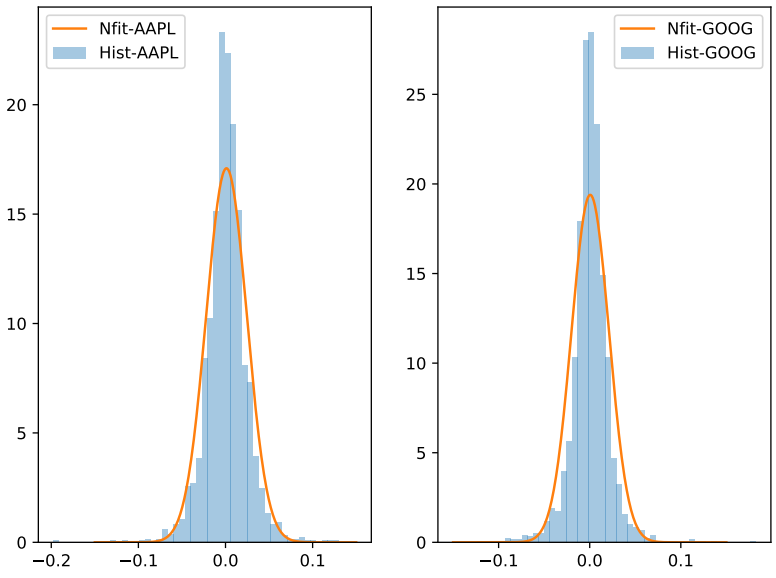


Figure 3.3.: Histograms of the log-returns with fitted Normal distributions.

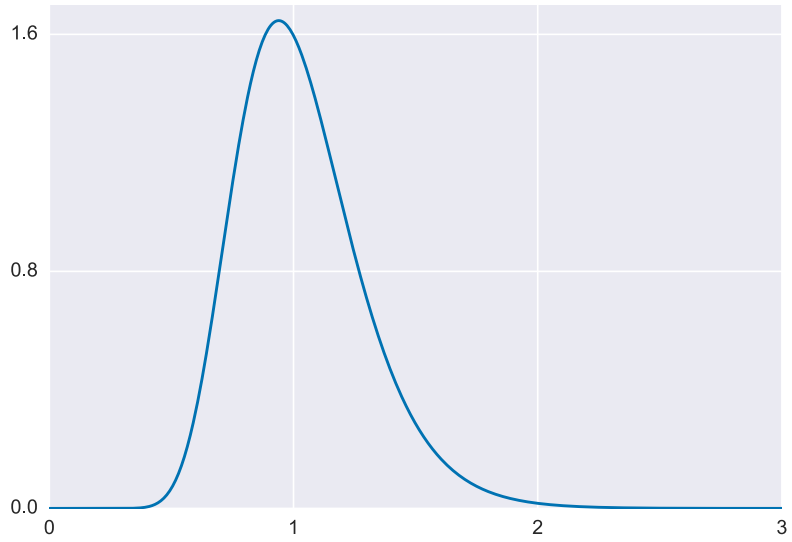


Figure 3.4.: Pdf of a LogNormal distribution.

elling the returns themselves as *LogNormal* random variables. A consequence is that the stock prices are also modelled as LogNormal.

Note that random variables modelled as LogNormal cannot be negative.

Bonus: expected value of a LogNormal

If X is modelled as a LogNormal random variable, it means that $\log X$ is modelled as a Normal random variable. Assuming the parameters of that Normal distribution are denoted by $\mathcal{N}(\mu, \sigma^2)$ then the expected value of X is given by

$$\mathbb{E}[X] = \exp\left(\mu + \frac{\sigma^2}{2}\right)$$

which is useful to know for the development of the Black-Scholes formula (which you will meet later).

Bonus: fitting a Student-t distribution

The Student-t distribution is commonly used to model the distribution of small samples of values that follow the Normal distribution. You can fit it to the sample data from AAPL and GOOG (or another stock) as follows:

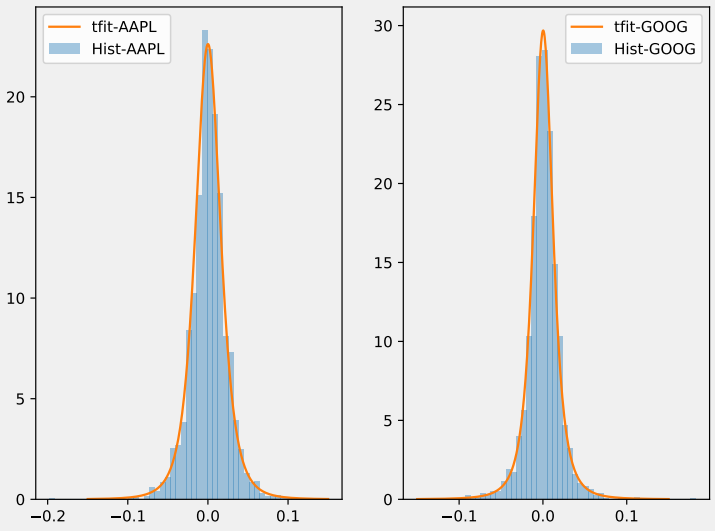
```
plt.figure(figsize=(8, 6))

lr             = lr_aapl
df, loc, scale = t.fit(lr)

plt.subplot(1, 2, 1)
sns.distplot(lr, kde=False, norm_hist=True, label="Hist-AAPL")
plt.plot(x, t.pdf(x, df, loc, scale), label="tfit-AAPL")
plt.legend()

lr             = lr_goog
df, loc, scale = t.fit(lr)

plt.subplot(1, 2, 2)
sns.distplot(lr, kde=False, norm_hist=True, label="Hist-GOOG")
plt.plot(x, t.pdf(x, df, loc, scale), label="tfit-GOOG")
plt.legend(loc="upper right")
```



As you can observe, the fit from the Student-t is far superior. This is something to keep in mind when considering more advanced models in finance although we will not use it in what follows.

Lagged log-returns

So far you have analysed the log-returns over two subsequent prices, but the notion of "subsequent" is arbitrary: you used daily quotes and so the subsequent prices where 24h apart, you could use weekly quotes, or monthly quotes. In general you can consider log-returns over ℓ days:

$$r_i^\ell = \log \frac{S_i}{S_{i-\ell}} \quad \text{where} \quad \ell = 1, 2, \dots$$

When $\ell = 1$, the definition is trivially identical to the previous one. You can explore what changes when increasing the lag; simply change the value of the `lag` variable in the code below.

```
# inline function to compute the lagged-log returns for an arbitrary lag
llr = lambda prices, lag: np.log(prices[range(0+lag,len(prices),    lag)] /
                                 prices[range(0,    len(prices)-lag,lag)])

# compute the log-returns corresponding to ratios over 5 quotes
# play with the lag afterwards to see what changes in the analysis
lag      = 5
llr_aapl = llr(aapl, lag)
llr_goog = llr(goog, lag)

plt.figure(figsize=(8, 6))

plt.subplot(1, 2, 1)
sns.distplot(llr_aapl, kde=False, norm_hist=True, label="AAPL")
plt.xlim([x.min(), x.max()])
plt.legend()

plt.subplot(1, 2, 2)
sns.distplot(llr_goog, kde=False, norm_hist=True, label="GOOG")
plt.xlim([x.min(), x.max()])
plt.legend()

mu, sigma = norm.fit(llr_aapl)

plt.subplot(1, 2, 1)
plt.plot(x, norm.pdf(x, mu, sigma))

mu, sigma = norm.fit(llr_goog)

plt.subplot(1, 2, 2)
plt.plot(x, norm.pdf(x, mu, sigma))
```

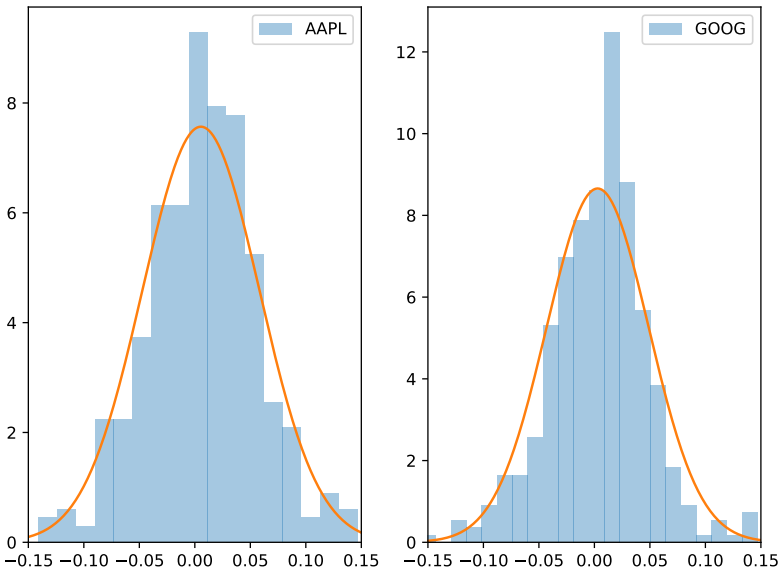


Figure 3.5.: Histogram of the log-returns for a lag of 5 days.

As you can observe, the form of the histogram stays the same. However, the horizontal spread of the histogram (i.e., the variance) increases: the uncertainty is related to time. The mean also changes, but it is less obvious. We can try to establish these links by estimating the mean and the variance for several lags:

```
lags = np.array([1, 3, 5, 7, 10, 15, 20])
sig_aapl = np.zeros(len(lags))
mu_aapl  = np.zeros(len(lags))
sig_goog = np.zeros(len(lags))
mu_goog  = np.zeros(len(lags))

i = 0
for lag in lags:
    mu_aapl[i], sig_aapl[i] = norm.fit(llr(aapl, lag))
    mu_goog[i], sig_goog[i] = norm.fit(llr(goog, lag))
    i+=1
plt.figure(figsize=(8, 6))
plt.plot(lags, mu_aapl, label="AAPL")
plt.plot(lags, mu_goog, label="GOOG")
plt.xlabel("Lag [days]", fontsize=12)
plt.ylabel("Fitted means", fontsize=12)
plt.legend()
```

The LogNormal model

From the rough analysis above, it seems reasonable to model both the mean and the variance of the log-returns as growing *linearly* with the lag. This forms the *LogNormal model* which, mathematically, can be written as

$$\log \frac{S_{t+\tau}}{S_t} \sim \mathcal{N}(\mu\tau, \sigma^2\tau)$$

where $\tau = (T - t)$ is the time span considered. The parameters μ and σ (the *volatility*) are adjusted so that they correspond best to the observed log-returns and therefore characterise the asset.

This model is too simple to accurately represent the financial reality. But it is still widely used as it is easy to interpret and a number of corrections can be applied to account for its shortcomings. We will discuss this in more details towards the end of the course.

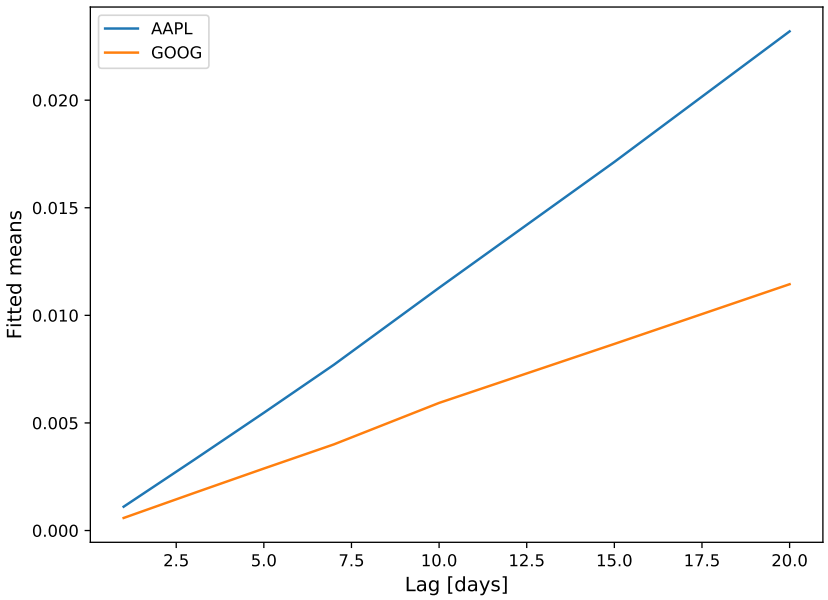


Figure 3.6.: Evolution of the mean with the lag.

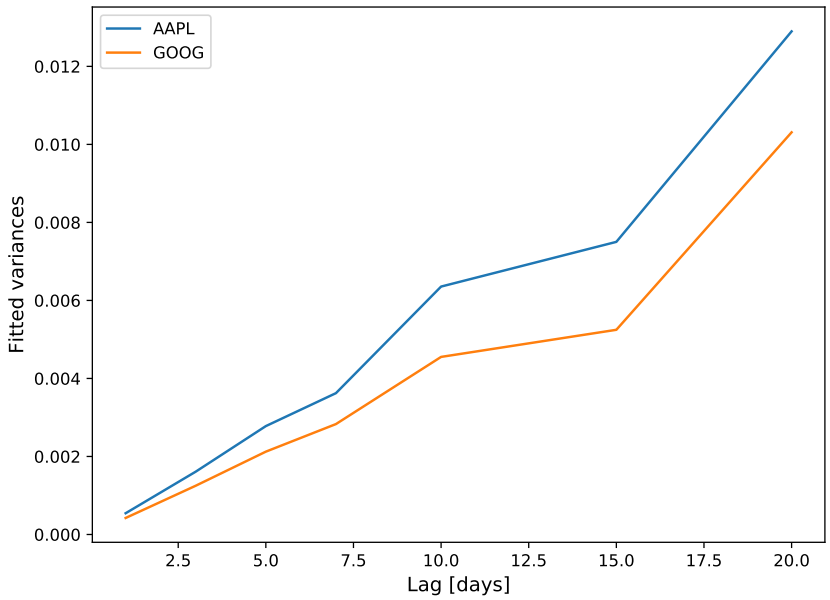


Figure 3.7.: Evolution of the variance with the lag.

3.2. Summary

In this section you analysed the evolution of the *log-returns* of assets and observed that:

- looking at the returns allows to get rid of the price-scale of different assets,
- the log-returns can be modelled approximately with a Normal distribution,
- the mean and variance of the log-returns can be modelled to grow linearly with the lag.

These observations led to the LogNormal model for the random process associated with the log-returns which can be written

$$\log(S_{t+\tau}/S_t) \sim \mathcal{N}(\mu\tau, \sigma^2\tau)$$

4. Pricing Theory

4.1. Goal of this section

By the end of this section, you will understand the following magical sentence:

> The price of a financial asset is equal to its discounted expected payoff under the risk-neutral probability.

It is expressed by the following elegant (if somewhat cryptic) mathematical formulation:

$$V_t = \Phi_r(t, T)\mathbb{E}_t^\star[V_T]$$

This is known as the *risk-neutral* or *arbitrage-free* price. Once you understand this formula, you will know how to price *any* financial asset.

The section is structured as follows:

- First, you will learn about *discounting* which is the way to account for the time span considered in the price. It explains the $\Phi_r(t, T)$ factor.
- Then, you will learn about the *risk-neutral probability distribution* which is the distribution used to compute the expected value of an asset. It explains the $\mathbb{E}_t^\star[V_T]$ and why we mark it with *.
- Finally, you will use these two concepts together to price derivatives.

4.2. Discounting

Remember, from the introduction, that you can always decide to invest some amount X in a bank account with an interest rate r. If you leave it in the bank from time t to $T > t$, then at time T it is worth $X\Psi_r(t, T)$ where

Ψ_r denotes the multiplicative factor gained at rate r over a period of time (*capitalisation factor*).

It is more conventional to work with the *discounting factor*, defined as the inverse of the multiplicative factor:

$$\Phi_r(t,T) = \frac{1}{\Psi_r(t,T)}$$

In order to follow notational conventions, we will use the discounting factor Φ_r exclusively. You can form intuition about the discounting factor as follows: if you invest $X\Phi_r(t,T)$ from time t to $T > t$ then you end up with X.

The precise form of this Φ_r depends on how the "rate" is defined in the context and does not really matter; we can take it as given. In mathematical finance the factor is often modelled as a *continuously compounded interest rate* with the form $\Phi_r(t,T) = \exp(-r(T-t))$. In what follows, and as we did for the LogNormal model, we write τ for the time span $T - t$.

More importantly, we assume positive interest rates. Specifically, we assume that $0 < \Phi_r(t,T) < 1$. It justifies the saying:

> "A pound today is worth more than a pound tomorrow.

Indeed, if you have £1 $\times$ $\Phi_r(t,T)$ (which is less than £1) at time t (today) and you invest in a bank account up until time T (tomorrow), you will end up with £1. And similarly, if you have £1 today and invest it, you will get (slightly) more than £1 tomorrow whence the saying.

This principle is referred to as the *time value of money*. And it means that at time $t = 0$, if you want to compare two future deterministic (non-random) trades incurring some transfer of value (*cash flows*), say, X_1 at time T_1 and X_2 at time T_2, you need to *discount* them; that is, you should compare

$$X_1\Phi_r(0,T_1), \quad \text{and} \quad X_2\Phi_r(0,T_2)$$

In other words, you should compare the amounts that you need to invest in the bank *today* in order to get the same future cash-flows.

Definition

Discounting allows to compute the value today (*present value*) of a future cash-flow thereby allowing the comparison of different investment plans.

Example

You want to sell your house and your friend offers to pay you £200k today or £208k next year. The annual rate at your local bank is 5% – thus, the multiplicative factor is $\Psi_r(0, 1\mathrm{Y}) = 1.05$. What should you do?
Comparing the two cases, you have:

1. the present value of taking the £200k now is £200k,
2. the present value of getting £208k in 1 year is $\text{£}208\text{k}/1.05 = \text{£}198.1\text{k}$.

It is therefore clearly in your advantage to take the money today.
Remark: Here obviously you could also see that placing £200k at 5% would get you £210k in a year which is greater than what your friend offered. But if you imagine comparing more than 2 cash-flows, it becomes clear that comparing the present value makes more sense.

4.3. The risk-neutral world

Motivation

Consider a coin-flip game: you win £10 if it lands heads and £0 if it lands tails.
What is the highest price you would accept to pay to play the game?

To answer the question, you should take two things into account: your *expected gain* (here £5) and your *aversion to risk*.

The aversion to risk can be classified in three categories:

- **Casino player**: You like risk and you are happy to pay more than your expected gain (say, up to £6) even though, on average, you will lose money.

- **Risk-neutral player**: You like a "fair game" and you are happy to match your expected game by paying up to £5. On average you don't gain or lose anything.
- **Risk-averse player**: You don't like the risk of losing money and expect a reward to compensate this risk-taking. You only accept to pay less than your expected gains (say, £3).

Let's consider a simple, *arbitrage-free market* where one agent, the *seller*, is selling access to the game described above; the other agents are potential players. Because the market is abitrage-free, and according to the law of one price, there must be a unique price for participating in the game. From the seller's perspective, the price should be as high as possible with many people willing to play.

- If all players are risk-neutral then the seller should set the price to the expected gain: £5.
- In practice however, most players are risk-averse and hence, in order to attract many players, the seller sets the price lower at, say, £3.

Risk-neutral probability distribution

The difference between the actual price and the expected gain (here £2) can be interpreted as a *reward* to players for undertaking risk. You can view this reward in two equivalent ways:

- as the expected gain minus a "price of risk",
- as the expected gain *if* the odds of landing heads *were* 0.3 (30%).

To clarify the second point, let us define $p^\star$, the alternate pdf for this game with $p^\star(H) = 3/10$ and $p^\star(T) = 7/10$. The expected gain under $p^\star$ is:

$$\text{expected gain} = £10 \times \frac{3}{10} + £0 \times \frac{7}{10} = £3$$

Note that the pdf $p^\star$ leads to an *interpretation* of the price. Specifically, it is a fictional pdf under which the actual price corresponds to the amount risk-neutral players would pay.

Definition

The *risk-neutral probability distribution* (denoted $p^\star$) is a fictive probability distribution associating *different weights* to possible events in such a way that the expected gain under that probability distribution is equal to that implied by the price on the market.
We write $\mathbb{E}^\star$ for the expected value under $p^\star$.

Use of the risk-neutral probability distribution

Why is the risk-neutral probability distribution useful? Why is this interpretation of price important? Because it lets you price any other game or asset that depends on the same source of randomness in a a way that guarantees *there is no arbitrage*. To compute the price of these new games, simply compute their expected gain under the known, risk-neutral pdf. This is a fundamental principle in finance; the general proof is a bit technical but if you are curious, you can refer to *Follmer and Schied* (see references).

Example

Let us assume that the game above is still on the market but that I am now also offering another game based on the same coin flips where the rewards are

- £6 if head,
- £15 if tail.

What is the *arbitrage-free, risk-neutral* price of that game?
The source of randomness is the same than that of the previous game (the same coin flips) and we had already described it with $p^\star = 3/10$.
Hence, the arbitrage-free, risk-neutral price is computed as follows:

$$P^\star_{\text{game}_2} = \mathbb{E}^\star[\text{game}_2] = £6 \times \frac{3}{10} + £15 \times \frac{7}{10} = £12.3$$

Bonus: another price implies arbitrage

> Can you show that if the second game above was priced any differently than $P^\star$, there would be an arbitrage opportunity?

To carry this proof, first consider the case where the price is outside the range £6 to £15. You can find an arbitrage opportunity by either selling or buying the second game.

If the price is within the £6 to £15 range, put yourself in the position of a seller. You sell $\alpha > 0$ times the second game for the price P and 1 time the first game for the price 3. Your cash flows in £ are

- *In case of Head*: $\alpha(P-6)+(3-10)$ where the first term is the price you receive (P) minus the money you pay out to the players (6) and the second term is the price you receive (3) minus the money you pay out to the players (10).
- *In case of Tail*: $\alpha(P-15)+(3-0)$ where the terms are similar but with the outcomes for tails.

If the price P is such that both cash flows can be positive simultaneously, then you have an arbitrage opportunity. Remember that the price is within the £6 to £15 range or, in other words: $(P-6)>0$ and $(P-15)<0$. Thus, finding an opportunity for arbitrage reduces to finding a pair (P,α) such that

$$\begin{cases} \alpha & > & 7/(P-6) \\ \alpha & < & -3/(P-15) \end{cases}$$

Similarly, if you consider buying as opposed to selling, the conditions are just reversed.

This is illustrated in the figure below. You can observe that, indeed, if the price is different than $P^\star = £12.3$, there exists arbitrage opportunities.

The region highlighted on the right corresponds to the case where you should *sell* the games. The other region corresponds to the flipped case where you should *buy* the games. Any point in those regions forms an arbitrage.

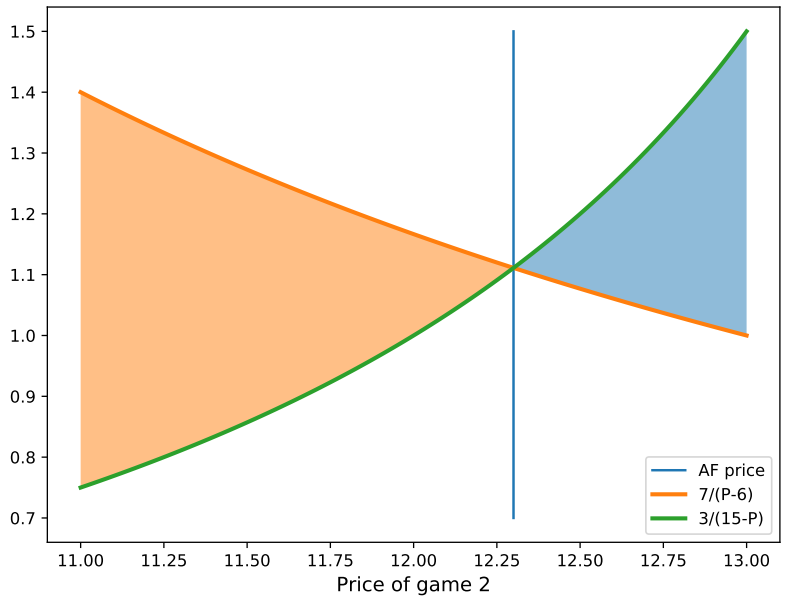


Figure 4.1.: Boundary curves for arbitrage opportunities.

4.4. Gathering the pieces

You have now learned the two fundamental building blocks of asset pricing: *discounting*, which values today a future cash-flow, and the *arbitrage-free price* of an asset which sets the price of any instantaneous asset. Putting both together, we can price any derivative as advertised at the beginning of the section:

Pricing formula

The *arbitrage-free* (*risk-neutral*) price/value today V_t of an asset with future payoff V_T is given by the discounted expected payoff under the current risk-neutral measure:

$$V_t = \Phi_r(t, T)\mathbb{E}_t^\star[V_T]$$

In the formula, the discounting factor is (usually) trivial. The difficult part to compute is the expected value under the current risk-free probability distribution. As you saw with the coin-flip example, the procedure for computing the current risk-free probability distribution is:

- observe an existing price on the market,
- compute a pdf $p^\star$ that matches it.

For the coin-flip example, $p^\star$ is easy to compute because it is only really determined by a single value.

In the general case however, it is harder to compute $p^\star$ but you can use the prices of simple assets like shares that are already on the market since they also have to respect the pricing formula. You will need to remember this when we consider the *calibration* of standard models like the Binomial tree model or the Black-Scholes model.

4.5. From coin-flips to a simple market model

The example above illustrated the notion of a risk-neutral world in a very simple market model: the only two assets derive from a coin-flip. We will now do the same thing in a more realistic market model and in a way that will introduce you to the concept of *replication*.

Assume there are three assets:

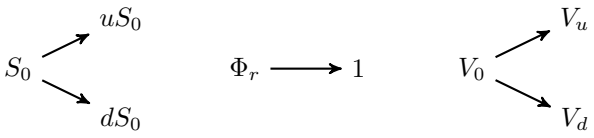


Figure 4.2.: Schematic representation of the evolution over one time step of the value of the three assets considered.

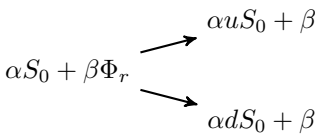


Figure 4.3.: Schematic representation of the evolution over one time step of the value of the portfolio considered.

- one stock with current quoted price S_0,
- one bank account with fixed interest rate r,
- one derivative based on the stock with current value V_0.

Now, consider one "time step" (e.g., between today and tomorrow), and model the evolution of the different assets in the following simple way:

Note the use of Φ_r as a shorthand for $\Phi_r(0, 1)$ here.

In this market, we consider two possibilities of evolution for the price of the stock with two fixed factors u and d. The derivative takes values V_u or V_d following the situation. (Note the similarities with the coin flip example).

Consider now the following scenario: you build a portfolio with the stock and the bank account in such a way that the payoff of your portfolio replicates that of the derivative. Then, by the law of one price in an arbitrage-free market, the initial value of that portfolio should be set to the initial value of the derivative. This approach is called *pricing by replication*. Visually the evolution of the portfolio looks like:

Note here that both α and β can be positive (invest in the stock, invest in the bank account) or negative (sell the stock, borrow from the bank account). Since you want to replicate the evolution of the derivative, you need to determine $(\alpha^\star, \beta^\star)$ such that

$$\begin{cases} \alpha^\star u S_0 + \beta^\star &= V_u \\ \alpha^\star d S_0 + \beta^\star &= V_d \end{cases}$$

and, once you have determined those, the law of one price (LOP) implies that

$$V_0 = \alpha^\star S_0 + \beta^\star \Phi_r$$

With a little bit of simple maths (see the Bonus section below), the pricing by replication can be shown to be equivalent to the risk-neutral pricing. But what matters most here is for you to understand the principle of replication and the use of the law of one price.

Another important concept unveiled by the concept of replication is that of *hedging*: trying to reduce risks by also trading a position which compensates the possible downside of a derivative.

Bonus: Equivalence between replication and risk-neutral pricing

It is fairly straightforward to verify that the solution to the Replication equation above is

$$\alpha^\star = \frac{V_u - V_d}{(u-d)S_0} \qquad \beta^\star = \frac{uV_d - dV_u}{u-d}$$

Using the law of one price (see LOP equation above), the price of the derivative is therefore:

$$\begin{aligned} V_0 &= \alpha^\star S_0 + \beta^\star \Phi_r \\ &= \Phi_r \left(\underbrace{\frac{\Phi_r^{-1} - d}{u-d}}_{p^\star} V_u + \underbrace{\frac{u - \Phi_r^{-1}}{u-d}}_{1-p^\star} V_d \right) \end{aligned}$$

The second line is obtained after a little bit of simple algebra; the form in which it is written is meant to remind you of the general pricing formula. Indeed, assuming for simplicity that $u > \Phi_r^{-1} > d$, the $p^\star$ above is between 0 and 1 and is the risk-neutral probability with:

$$V_0 = \Phi_r(p^\star V_u + (1-p^\star)V_d) = \Phi_r \mathbb{E}^\star[V]$$

You have now shown that replication pricing can be put in the same form as risk-neutral pricing. The one thing left to verify is that both consider the same risk-neutral probability $p^\star$.

For this, note that, in an arbitrage-free markets, *all assets* must be priced according the same general pricing formula, and, in particular, the stock price must verify:

$$S_0 = \Phi_r \mathbb{E}^\star[S] = \Phi_r(p^\star u S_0 + (1-p^\star) d S_0)$$

Solving this for $p^\star$ leads to

$$p^\star = \frac{\Phi_r^{-1} - d}{u - d}$$

which is the exact same one as was obtained via replication-pricing. This finishes the proof that *risk-neutral pricing* and *replication pricing* are here equivalent.

Remark

Above, we assumed that $u > \Phi_r^{-1} = \Psi_r > d$. This ensures that the risk-neutral probability is between 0 and 1. However, observe that economically, it makes sense:

- if $\Psi_r \geq u$, then no agent would have an interest in buying the asset and they should just invest in the bank account,
- if $\Psi_r \leq d$, there is an obvious arbitrage opportunity.

4.6. Summary

In this section we have introduced the following concepts:

- The *risk-neutral world* is an imaginary world where investors are assumed to require no extra return on average for bearing risks or, equivalently, where all assets return the risk free rate.
- *Discounting* is a way to compute the present-value of future cash flows in order to compare them,
- *Replication* imitates the cash flows of a derivative using simple assets and deducing the price of the derivative using the law of one price.

5. Derivatives

5.1. Introduction

You have already encountered a very simple kind of derivative: the *forward contract* where one party (in the *long* position) agrees to *buy* an asset at a specified time in the future at a specified price and the other party (in the *short* position) agrees to *sell* the asset at that time at that price.

It's time to learn about more, well known derivatives and their characteristics.

5.2. Payoff curves

The payoff curve of a derivative represents the value of the exchange, at maturity, for all possible outcomes of the underlying.

In the case of a forward contract, the payoff is the difference between the strike price and the market price of the derivative at maturity. The agent in the long position gets a positive payoff if the strike price is higher than the market price of the derivative at maturity: they get more money from the forward contract than they would have had selling the underlying at market price at maturity. For the agent in the short position, the opposite is true.

More formally, the payoff for the agent in the long position is $V_T^{\text{fwd}} = (S_T - K)$ where S_T is the market price of the asset at maturity (time T).

Several comments apply:

1. The curves intersect at 0 when $S_T = K$: no one makes or loses money if the underlying is exactly worth the strike price,
2. The curves are mirrored which is due to the mirrored buyer-seller relation,

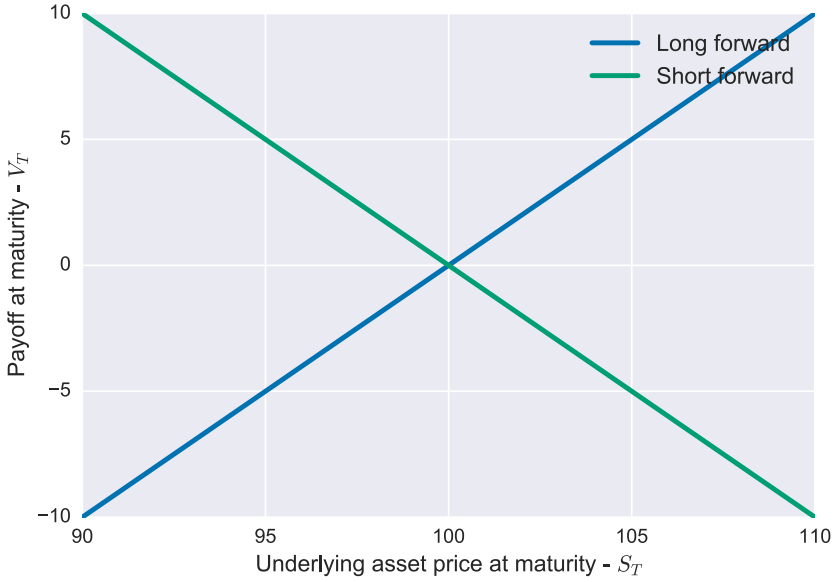


Figure 5.1.: Payoff curves for the long and the short forward contracts when the strike price is $K = 100$.

3. The payoffs can be big, which means that either agents can loose a lot of money! You will learn later about some derivatives that are designed to avoid losing more than a fixed amount.

In financial lingo, the key features of the long forward contract are summarised as:

- *Unlimited upside*: You can make an arbitrarily large profit.
- *Limited downside*: You can lose a lot of money but not arbitrarily so because the price of the underlying cannot go below 0.
- *Bullish* or *optimistic*: You expect the price of the underlying to go up.

The opposite term for bullish is *Bearish* (or *pessimistic*): You expect the price of the underlying to go down.

In the rest of this section, you will discover other derivatives, their payoff curves and key features. We focus on long positions since the short positions are always mere mirrors of the long ones. We will explain some of their properties but not in much details since the focus of this course is more the pricing than the financial engineering.

Note that the payoff is what you need in order to price the derivative. In the example above you can do it as follows (and you will learn how to do this with more generality in the next section):

$$\begin{aligned} V_t^{\text{fwd}} &= \Phi_r(t,T) \times \mathbb{E}_t^\star[V_T^{\text{fwd}}] \\ &= \Phi_r(t,T) \times (\mathbb{E}_t^\star[S_T] - K) \\ &= S_t - \Phi_r(t,T) \times K \end{aligned}$$

where, at the last line, you have to use the fact that the price of the underlying must obey the risk-neutral pricing: $S_t = \Phi_r(t,T)\mathbb{E}_t^\star[S_T]$.

5.3. Options

Options are derivatives centred around the concept of having the option (the *right* without the *obligation*) to buy, or sell, an asset at a specified price – the *strike* price – and at a specified time – *maturity* time.

Options are financial instruments that transfer risk from a person unwilling to take it to a person willing to bear it in return for a fee (price of the

option). This price can be interpreted as that of an insurance policy and is also known as a *premium*.

Note that you will first consider the case where the contract can only be exercised at the maturity date (these contracts are called *European options*) and you will later learn about alternatives (*American options*, *Asian options*, etc.). The origin of the names of options ("European", "American", "Asian") is unclear and has nothing to do with geography; these options are traded everywhere.

Call option

The European (long) call gives you the option to buy an asset at a fixed strike-price and at a fixed future date (maturity). Choosing to buy is referred to as *exercising the call*.

Example

You buy a call on an ounce (roughly 30g) of gold expiring in 2 months with a strike price of £1k. Consider the following two scenarios at the maturity date:

1. The price of an ounce of gold is now £1.2k which is larger than the strike price. You choose to exercise the call (buy the ounce for £1k) and immediately sell the asset for the current price with a resulting payoff of £0.2k.
2. The price of the ounce of gold is now £0.9k which is not larger than the strike price. You choose not to exercise the call with a resulting payoff of £0.

It is important to note here that the payoff is different than the profit you will make at maturity. Indeed, you have to take into account that you initially paid a small amount to buy the contract.
For our purpose (pricing derivatives) the payoff is the relevant measure.

Following the example above, the payoff value of a (long) call at maturity can be simply computed as:

$$V_T^{\text{call}} = \max\{S_T - K, 0\}$$

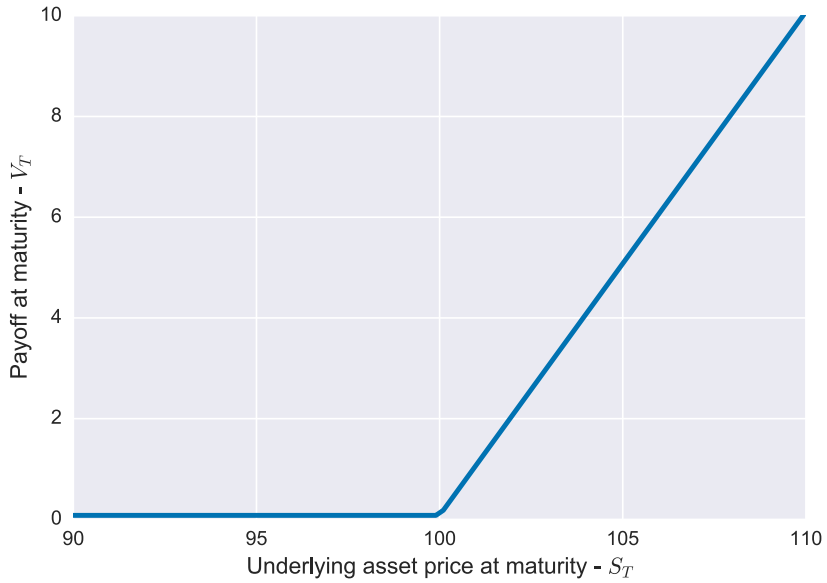


Figure 5.2.: Payoff curve of a long call with strike $K = £100$.

the figure below illustrates the *hockey stick* shape of the corresponding payoff curve:

The key features of a (long) call are:

- *Unlimited upside*
- *No downside* (the only risk is to lose the initial price or *premium*)
- *Bullish*
- *Leverage* (see below)

Put option

The European (long) put gives you the option to sell an asset at a fixed strike-price and at a fixed future date (maturity).

The payoff of a put is, by contrast to the call:

$$V_T^{\text{put}} = \max\{K - S_T, 0\}$$

Figure 5.3.: Payoff of a long put with strike $K = £100$.

the figure below illustrates the hockey stick shape of the corresponding curve.

Observe the symmetry between the call and the put. You will exploit this symmetry to price calls and puts.

The key features of a (long) put are:

- *Limited upside* (possibility to make large gains if the stock crashes but the price of the underlying is bounded by 0)
- *No downside* (the only risk is to lose the initial price or *premium*)
- *Bearish*
- *Leverage* (see below)

Options and leverage

Buying a call over an underlying is cheaper than buying the underlying directly. Additionally, with options you can, starting from the same capital, make more profit than if you invest directly in the underlying. This is

called *leverage*.

Definition

Leverage is a multiplicative effect attained by a derivative over investing simply in the underlying.

Example

Let us consider a call on an ounce of gold again. Suppose that

- the price of the call is £15,
- the current price (*spot* price) of the underlying (an ounce of gold) is £0.95k,
- you have a budget of £10k,
- the price of the underlying rises to £1k by the maturity date T,
- the strike price in the call is at £0.98k.

Consider these two options:

a. You invest in the **underlying**: With your budget you initially buy 10.5 ounces of gold which are worth £10.5k at maturity T. Your gross growth is 5%.
b. You invest in the **call**: With your budget you can buy (more than) 650 call options. At maturity T, you exercise the calls (i.e., you buy 650 ounces of gold at £0.98k per ounce). You sell the gold immediately at £1k per ounce thereby making 650 $\times$£0.02k $=$ £13k. Your gross growth is 30%.

Question: What happens, in each situation, if the price of the underlying drops to £0.9k (instead of rising) by the maturity date T?

The example above is artificial. But the principle remains valid: with a call you can buy far more of the underlying than you could have otherwise, only effectively exchanging money for the asset at the maturity date if it is in your favour to do so. In that case you can borrow money from the bank in order to effectively buy the asset, immediately sell for a higher price and reimburse the bank quickly.

Figure 5.4.: Payoff curve of a long straddle with strike $K = 100$.

Combinations

In your portfolio, you can combine several derivatives. For example you can combine a call and a put with the same characteristics (same strike and maturity date). This is called a *straddle*.

Below is Python code that produces the payoff curve for a straddle.

```
K = 100 # Strike Price
S = np.linspace(90,110,100)
Vcall = np.maximum((S-K),0.)
Vput  = np.maximum((K-S),0.)
Vstraddle = Vcall+Vput
plt.plot(S, Vstraddle, label="Long straddle")
plt.xlabel("Underlying asset price at maturity - $S_T$")
plt.ylabel("Payoff at maturity - $V_T$")
plt.xticks(np.arange(90, 115, 5))
plt.yticks(np.arange(0, 15, 5))
```

Each specific combination has a purpose. When investing in a straddle, for example, you expect a change in the price of the underlying but you do not

Figure 5.5.: Illustration of the put-call parity. observe how combining the put and the forward gives the call.

know in which direction. For example, if a company is due to announce its earnings and it's unclear whether it's going to be good or bad news.

5.4. Put-Call Parity

Remember that the payoff curves for puts and calls are symmetrical. Mathematically, the relation goes as follows:

$$V_T^{\text{call}} - V_T^{\text{put}} = (S_T - K)$$

This can be seen by overlaying the two payoff curves:

Now, remember that the price of anything on the market can be obtained by computing the expected payoff in the risk-neutral world and discounting it. Hence, the price of the put (V_t^{put}) and that of the call (V_t^{call}) at a time $t < T$ are directly related:

$$\begin{aligned} V_t^{\text{put}} &= \Phi_r(t,T)\mathbb{E}_t^\star[V_T^{\text{put}}] \\ &= \Phi_r(t,T)\mathbb{E}_t^\star[K - S_T + V_T^{\text{call}}] \\ &= \Phi_r(t,T)K - S_t + V_t^{\text{call}} \end{aligned}$$

using the same reasoning as the one we used when pricing the forward contract. This is known as the *put-call parity* principle.

This principle only holds for European options (i.e., where the exercise is only possible at the maturity date).

5.5. Other derivatives

There are many more types of derivatives (options and other) used in finance. Below are a few other types of options.

American options

American options are similar to the European options that you saw before except that the contract can be exercised at any time before the maturity date (*early exercise*), if it is in your advantage to do so. You will see how to price an American put when covering the binomial tree.

Bonus: More on early exercise

You may wonder when it is in your advantage to exercise early. Obviously, it is only in your interest to exercise the option if it is *in the money* (if it has a positive payoff). Looking at an American call the moment we exercise, the payoff is $(S_t - K)$ and for an American put, $(K - S_t)$.

Let us start with the American put. The lower the price of the underlying is, the more valuable the put becomes. In contrast, the corresponding call loses more and more value. So if we take the limit of the put-call parity and let the price of the underlying go to zero:

$$\lim_{S_t \to 0}(V_t^{\text{call}} - V_t^{\text{put}}) = -\lim_{S_t \to 0} V_t^{\text{put}}$$

but the put-call parity tells us that $(V_t^{\text{call}} - V_t^{\text{put}})$ is equal to $(S_t - K\exp(-r\tau))$ so that, in the limit where the price of the underlying is very low,

$$V_t^{\text{put}} \approx K\exp(-r\tau) - S_t$$

Since we assume that $r > 0$, $K\exp(-r\tau) < K$ so that, in the limit above,

$$V_t^{\text{put}} < K - S_t$$

In other words, with a positive interest rate, early exercise the American put will always be in your favour as long as the spot price is small enough.

Exercise

Follow a similar reasoning for an American call.
Note that doing so, you can show that it is never optimal to exercise an American call early under these conditions.

Remark

Note that we have ignored dividends and the cost of holding the stock. If we include those factors, it can be optimal to exercise an American call early.

Exotic options

There are many other types of options with more complex properties than that of the European or American options. Even though it relies on the same formula, their pricing requires a bit more effort.

For example, some options are *path dependent*: their payoff depends on the evolution of the underlying's price between the instantiation point and the exercise point. An *Asian option* for example, looks at the average price of the underlying over a pre-fixed period of time. The payoff of a *barrier option* depends upon whether a fixed price (*barrier*) is attained by the underlying.

We do not cover those in depth here but it is useful to know that these exist since they justify the use and development of specific methods in order to compute their price.

5.6. Summary

In this section we have covered the following concepts with regards to derivatives:

- In any *forward contract,* one party is in a *long* position (agrees to *buy* an asset at a specified time in the future at a specified price) whereas the other party is in a *short* position (agrees to *sell* the asset at that time and price).
- The payoff curve of a derivative represents the value of the contract at maturity.
- Options are derivatives that grant their holders the *right* or *option*, but not the *obligation,* to buy or sell an asset at a specified price (the *strike* price) and at a specified time (*maturity*) time.
  - The parties to an option are its buyer (*long party*) and its seller (*short party*)
  - The European *long call* gives you the option to buy an asset at a fixed strike-price and at a fixed future date (maturity). The European *long put* gives you the option to sell an asset at a fixed strike-price and at a fixed future date (maturity).
  - European options may be exercised only at the maturity date, whereas American options may be exercised at any time before the maturity date (*early exercise*).
  - You can create combinations of derivatives in your portfolio; for example, the *straddle* is a combination of a *call* and a *put* option with the same strike price and maturity date.
  - Besides the common *vanilla* European or American options, there are options with features more complex known as *exotic* options, the pricing of which requires a bit more effort.
- Options offer leverage: they are much cheaper than the underlying asset and hence many can be bought. When exercised this can lead to greater profits than simply investing in the underlying.

Additionally, it is important to remember that:

- The *payoff* is different than the *profit* you will make at maturity.
- The *put-call parity principle* only holds for European options.

6. Analytical pricing: the Black-Scholes formula

6.1. The bare minimum

You have learned how to price assets using the discounted expected value under the risk-neutral probability distribution. When assuming log-normality of the returns, this leads to the *Black-Scholes model* and, in particular, to explicit prices for European options.

Getting to the Black-Scholes model requires a fair bit of maths. The key concept is to express the distribution corresponding to the LogNormal model, find the corresponding risk-neutral distribution and then compute expected values under that model. Depending on your level of interest for the underlying maths, you can look at the appendix for the full development, at the overview below for the main points, or just look at the (standard) formula available below it.

1. First, model the log-returns $\log(S_T/S_t)$ as $\mathcal{N}(\mu\tau, \sigma^2\tau)$ in agreement with our previous observations.
2. Then, remember that, in the risk neutral world, all assets are expected to return the bank account rate r. Hence, the distribution must be modified in the following way: $\mu \leftarrow \mu^\star = r - \sigma^2/2$ so that the pricing formula holds for the asset.
3. The previous two points imply a specific model for the distribution of S_T.
4. Computing the expected values under that distribution model leads to pricing of European derivatives.

In particular, for the European long call, the expected value leads to following standard pricing formula:

$$V_t^{\text{call}} = S_t\Phi(d_1) - K\exp(-r\tau)\Phi(d_2)$$

where

$$\Phi(a) = \int_{-\infty}^{a} \frac{\exp(-x^2/2)}{\sqrt{2\pi}} \mathrm{d}x$$

is the *cumulative normal distribution* which can be computed using the `norm.cdf` function from `scipy.stats`.

> **Warning**
>
> Do not confuse the symbol $\Phi(a)$ (the integral above) and $\Phi_r(t, T)$ (the discounting factor $\exp(-r(T-t))$). For this course, you will only use the former in the context of the Black-Scholes formula so there should not be any ambiguity. Some textbooks use the notation $\mathrm{CND}(a)$ for $\Phi(a)$.

The constants d_1 and d_2 are given by

$$d_1 = \frac{\log(S_t/K) + (r + \sigma^2/2)\tau}{\sigma\sqrt{\tau}} \qquad d_2 = \frac{\log(S_t/K) + (r - \sigma^2/2)\tau}{\sigma\sqrt{\tau}}$$

Which is expressed in Python as follows:

```
def callPriceBS(St, K, tau, r, v):
    d1 = ( (np.log(St/K)) + (r+np.power(v, 2)/2)*tau ) / (v*np.sqrt(tau))
    d2 = d1-v*np.sqrt(tau)
    return St*norm.cdf(d1)-K*np.exp(-r*tau)*norm.cdf(d2)
```

So for example the price of a call which expires in 2 years, with a strike price of £100, a current asset price of £105, an annual interest rate of 5% and a volatility of 15% is

```
print("{0:.4f}".format(callPriceBS(105, 100, 2, 0.05, 0.15)))
```

which is approximately £17.4.

6.2. Greeks

When considering adding specific derivatives to their portfolio, traders will often consider specific numerical quantities which help characterise these derivatives and, by extension, their portfolio. These numerical

quantities are known as *the Greeks*: a set of symbols (most of which Greek letters) representing the sensitivity of the current value (price) of the derivative with respect to small changes of parameters.

It should be stressed here that the Greeks form an essential set of tools for the traders, in a way even more so than the price itself. They tell them what their risks are subject to variations in the market when investing in a particular derivative.

The essentials: Delta, Vega, Theta

Delta

The Delta (Δ) of a derivative quantifies the sensitivity of the value of the derivative considered to a small change in the price of the underlying:

$$\Delta := \frac{\partial V}{\partial S} \approx \frac{V(S_t + \Delta S) - V(S_t)}{\Delta S}$$

where ΔS is a small change in the underlying (which may be positive or negative). In words, this greek quantifies how much the value of the derivative changes ($\Delta V = V(S_t + \Delta S) - V(S_t)$) when the value of the underlying changes a little bit (ΔS).

As an aside, you may recognise the first term which is the *partial derivative* of the price of the derivative with respect to that of the underlying. It is defined as the limit of the right-hand side when the perturbation ΔS goes to zero.

Note also that this greek is computed at a specific time t and will therefore vary with time.

In the case of the Call, you have seen before an explicit formula for the price and you can compute the Delta exactly (analytically) provided you are comfortable with differentiation (it actually all boils down to $\Phi(d_1)$). If not or, in general, if you don't have an explicit formula for the price, you can also estimate the Delta numerically:

```
def callDeltaBS(St, K, tau, r, v):
    d1 = ( (np.log(St/K)) + (r+np.power(v, 2)/2)*tau ) / (v*np.sqrt(tau))
    return norm.cdf(d1)

def callDeltaNumerical(St, K, tau, r, v):
```

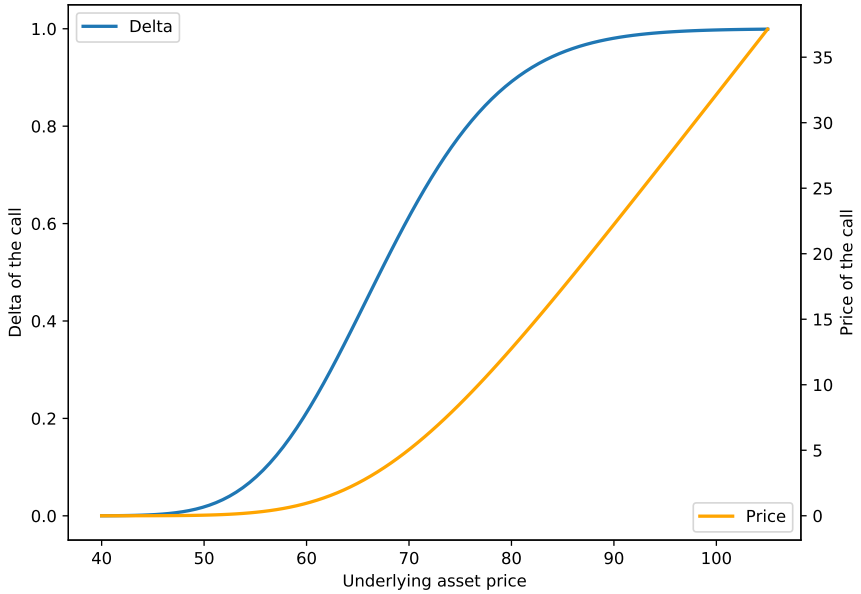


Figure 6.1.: Illustration of the Delta of a call for a range of asset prices (here $K = 75$, $\sigma = 0.1$, $\tau = 2$, $r = 0.05$).

```
    deltaS = St/1000
    Vplus  = callPriceBS(St+deltaS, K, tau, r, v)
    V      = callPriceBS(St, K, tau, r, v)
    return (Vplus-V)/deltaS
```

Trying both out with realistic values you can see that the numerical approximation can be perfectly acceptable (here within half a percent):

```
dAnalytical = callDeltaBS(60, 75, 2, 0.05, 0.1)
dNumerical  = callDeltaNumerical(60, 75, 2, 0.05, 0.1)
pctDiff     = np.abs(dAnalytical-dNumerical)/dAnalytical*100

print("Delta (Analytical): {0:.5f}".format(dAnalytical))
print("Delta (Numerical) : {0:.5f}".format(dNumerical))
print("Percentage diff   : {0:.1f}".format(pctDiff))
```

The curve corresponding to the Δ associated to a range of possible prices is illustrated below with the arameters $K = 75$, $\sigma = 0.1$, $\tau = 2$, $r = 0.05$:

Question

Can you think of why the Delta of a Call looks like the curve above? Can you think of the two limits where either the price of the underlying is much higher or much lower than the strike?

Vega

The Vega (ν, which is actually "nu" but known as "Vega", probably because it looks like a "V", maybe also because it rhymes with other greeks...) quantifies the sensitivity to the volatility σ of the underlying:

$$\nu = \frac{\partial V}{\partial \sigma} \approx \frac{V(\sigma_t + \Delta\sigma) - V(\sigma_t)}{\Delta\sigma}$$

where σ_t denotes the current estimate of the volatility.

Exercise

Use either a numerical or analytical approach to plot the evolution of the Vega of a call against a range of possible volatilities. You will have to choose arbitrary values for the strike price and other constants. Can you interpret the result?

Another useful illustration is the Vega against the price of the underlying which looks like this:

Observe also that the value of the call tends to the price of the underlying when the price of the underlying is far higher than the strike. On the other hand, when the price is much lower, then the value of the call tends to zero.

Theta

Finally, the Theta (Θ) quantifies the sensitivity to the lifetime $\tau = (T - t)$ of the contract:

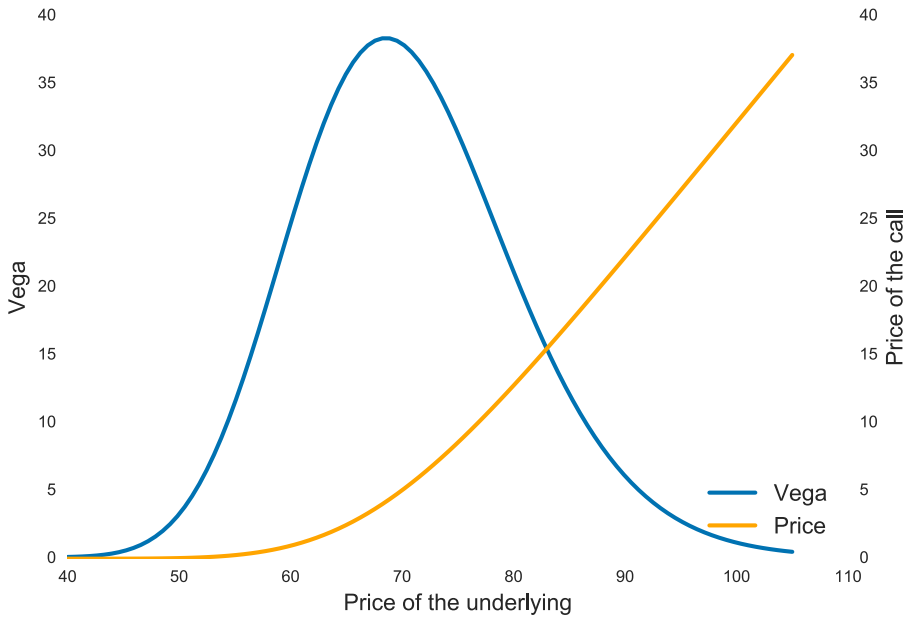


Figure 6.2.: Vega of a call for a range of prices of the underlying (with $K = 75$, $\tau = 2$, $r = 0.05$ and $\sigma = 0.1$)

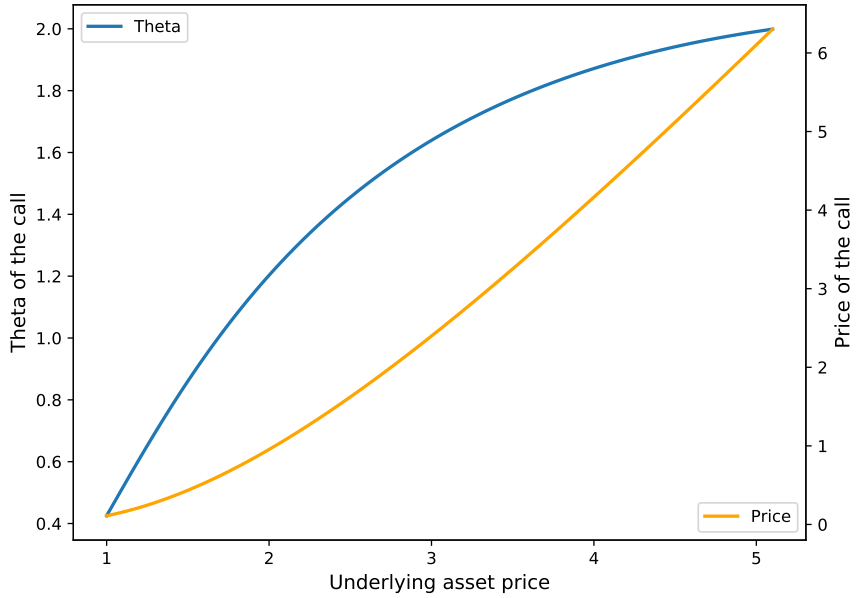


Figure 6.3.: Theta of a call for a range of volatilities (here $S = 60$, $K = 75$, $r = 0.05$, $\sigma = 0.1$).

$$\Theta = \frac{\partial V}{\partial \tau} \approx \frac{V(\tau + \Delta\tau) - V(\tau)}{\Delta\tau}$$

Note that the Theta of a call is always negative: the call loses value as time goes by. This is because it becomes less and less likely for the price to move significantly away from the current observed spot-price. This observation actually holds for any options and is known as *theta erosion*.

Summary

So far, you have learned the three most important Greeks.

Name	definition	Intuition
Delta, Δ	$\frac{\partial V}{\partial S}$	sensitivity to the price of the underlying

Name	definition	Intuition
Vega, ν	$\frac{\partial V}{\partial \sigma}$	sensitivity to the volatility of the underlying
Theta, Θ	$\frac{\partial V}{\partial \tau}$	sensitivity to the lifespan of the contract

Note that, if you can compute these Greeks for some derivative, you can compute them for a portfolio of these derivatives. This is because the Greeks are linear: the Delta of a portfolio is the sum of the Deltas of its assets, and similarly for Vega and Theta. For example, if you have two identical calls and one put in your portfolio, then the Delta of the portfolio is $\Delta^{\text{portf.}} = 2\Delta^{\text{call}} + \Delta^{\text{put}}$

Bonus: Other greeks

There are a few other greeks that are of interest to the traders like the Rho (sensitivity to interest rate). Note that Rho, Delta, Vega and Theta are *first order* greeks: they relate the variation of the price of a derivative to the variation of a parameter.

Traders also use *higher order* greeks which quantify the sensitivity of another (first order) greek to a parameter. An important one is the Gamma which measures the sensitivity of the Delta to the price of the underlying or:

$$\Gamma = \frac{\partial \Delta}{\partial S} = \frac{\partial^2 V}{\partial S^2} \approx \frac{(V(S_t + \Delta S) - V(S_t)) - (V(S_t) - V(S_t - \Delta S))}{\Delta S^2}$$

Bonus: Hedging with the Greeks

Although this course does not aim to teach trading, it is helpful to understand some of the ways traders use the Greeks to build their portfolio and therefore why being able to compute them accurately is important.

Greeks, as you have seen, fundamentally represent risks attached to a specific asset. A trader may seek to reduce those risks by having a portfolio with one or several Greeks at or around zero. For example, traders may consider *Delta-neutral* portfolios (with a Δ equal or close to zero) and seek to, at any given time, adjust (balance) their portfolio such that this holds.

Figure 6.4.: Implied volatility pipeline.

Holding such *hedges* perfectly is usually impossible and costly. Indeed, at every time step, it requires buying and selling assets such that the portfolio meets the requirements but, typically, these operations incur transaction fees. This makes repeated rebalancing of a portfolio costly. A trader therefore has to make a trade-off between frequent rebalancing and sticking to a strategy.

6.3. The volatility smile

Let us come back now to the Black-Scholes formula proper. The only non-observable parameter of the formula in the current context is the volatility which we currently assume is modelled based on observations:

However, on the market, we can assume that there already exists lots of calls and puts being sold at some price. We can therefore define the concept of *implied volatility* by reversing the second arrow (going from Prices to Volatility).

Definition

The *implied volatility* of a call/put is the volatility such that, when used in the Black-Scholes formula, we recover the same price as the one observed on the market.

Could we also use the implied volatility to calibrate our model? Intuitively, it would be nice if both the volatility modelled on the log-returns and the implied volatility matched. However, as you will see, this is not the case. This section is dedicated to understanding why and explaining how to work around it.

Observing the implied volatility

On markets with numerous trades, there are many options available with all sorts of underlyings, strikes and maturities. If we consider calls and puts on AAPL with a fixed maturity for example (data courtesy of IMC):

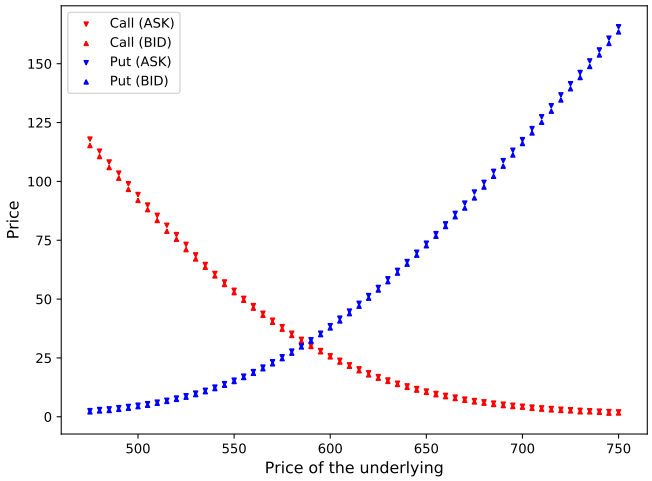


Figure 6.5.: Prices for AAPL options with a fixed maturity and a range of strikes (data courtesy of IMC).

For each of those prices, you could then compute the corresponding implied volatility. The computation itself requires a root finding algorithm – such as Newton-Raphson. You won't learn the details of that operation here, so we computed the implied volatility for you (data courtesy of IMC):

As you can see, the implied volatility varies significantly with the strike, even though all these puts and calls are on the same asset and with the same maturity. This differs from our model which has a single volatility σ computed from the log-returns.

Understanding the issue

The phenomenon identified above is known as the *volatility smile* and is the result of the LogNormal model itself. Remember that, to establish a model, we fitted a distribution to the log-returns. The fit was imperfect and this imperfection causes the volatility smile: with a very good fit, the implied volatility observed would be roughly constant (corresponding to σ).

The LogNormal model is a useful first approximation because it is simple

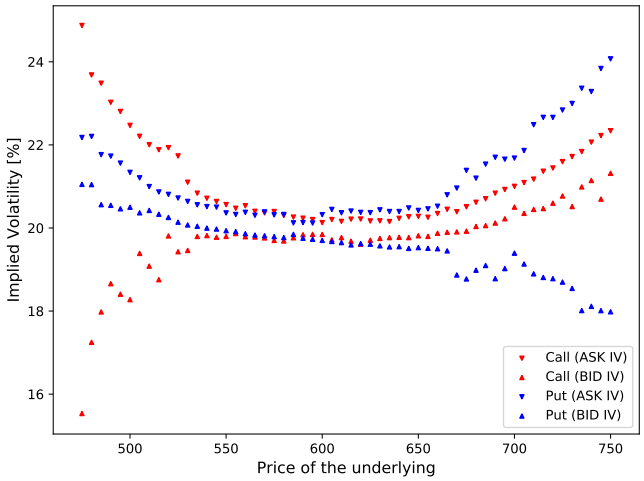


Figure 6.6.: Implied volatilities corresponding to the prices of the AAPL options (data courtesy of IMC).

to work with and yields the Black-Scholes formula. But now this model's limitations start to show.

To work around this issue, it would be natural to suggest fitting another distribution (for example the Student-t) or even using more complex models (stochastic volatility models, jump models, etc.). However, the approach most often used by practitioners is simple and uses the same framework you have learned so far and you will learn about it in the next section.

Bonus: Working with the volatility smile

So far we started from the LogNormal model and deduced the price of options implied by that model. We have now observed that the implied volatility is not constant. So we could suggest still using the same formula but with a volatility that is a function of the strike: $\sigma(K)$. Even further, the smile above also changes if the price of the underlying changes. Therefore it is common to suggest a volatility model of the form $\sigma(f(S, K))$ where a simple common choice of f is

$$f(S, K) = \log \frac{S \exp(r\tau)}{K}$$

This is a pragmatic approach which allows both to keep using the Black-Scholes formula (easy to implement, fast to compute and directly yielding Greeks), and to better fit the underlying financial reality.

In state of the art models, people consider some simple parametrized function for σ, typically a combination of polynomials, and fit it to the implied volatility smile.

6.4. Summary

In this section we have covered the following concepts with regards to the Black-Scholes formula and analytical pricing:

- the Black-Scholes formula and how it could be computed in a simple context where all parameters are given
- the Greeks, numerical quantities that help characterise derivatives (and portfolio) with the essential ones being the Delta, Vega and Theta:
- The **Delta** quantifies the sensitivity of the value of a derivative (or portfolio) to a change in price of an underlying: $\Delta = \partial V / \partial S$
- The **Vega** quantifies the sensitivity of the value of a derivative (or portfolio) to a change in the volatility of an underlying: $\nu = \partial V / \partial \sigma$
- The **Theta** quantifies the sensitivity to the lifetime of the derivative: $\Theta = \partial V / \partial \tau$
- the *implied volatility* of a derivative is the volatility such that, when used in the BS formula, the same price is recovered as the one observed on the market
- when the implied volatility is displayed against the strike price of contracts, it is not constant. Rather, there is a *volatility smile*.

7. The binomial model

7.1. From one-step to multi-step model

The binomial option pricing model is a simple approximation of returns which, when refined, converges to the analytic pricing formula for vanilla options (such as the European call).

The interest in covering it is that the binomial model is simple and useful for any kind of payoff including when early exercise is allowed (such as for an American option).

Recombining binomial tree

When covering arbitrage-free pricing, we had introduced a simple one-step model for the evolution of the price of an asset over one period of time which had the following form where the price of the asset will be either multiplied by a factor u (with probability p) or d (with probability $q = 1 - p$).

The model may seem unrealistic since it models possible prices after a period of time as taking one out of only two values. A simple extension of this model is therefore to make it multi-step, i.e.: repeat the branching over the same period of time.

Below, a two-step model is represented, the extension to N steps should then be clear. On the left, the evolution of the prices, on the right, the probability of these price evolutions.

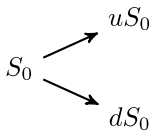


Figure 7.1.: A very simple market model.

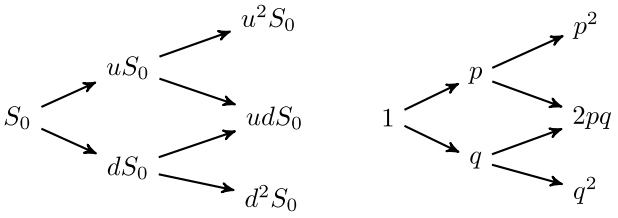


Figure 7.2.: A two-step model.

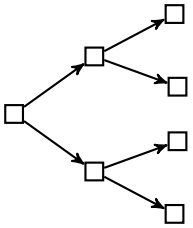


Figure 7.3.: A bushy tree.

The resulting structure is called a *recombining binomial tree*. The word "recombining" refers to the fact that the probability of going up or down at one node is independent of the node considered – it is always p and q. This means that going up at one node then down at the next node will bring you to the same state as going down first then up. A recombining tree is simple to parametrise and it also avoids exponential growth in the number of nodes. Indeed, the number of states for the consecutive steps grows slowly: 1 then 2 then 3 etc.

This distinguishes the recombining tree with the *bushy tree* where going up-then-down or down-then-up leads to two different states. Consequently, the number of nodes grows exponentially with the number of steps considered: 1 then 2 then 4 etc.

7.2. Convergence to the LogNormal distribution

When the number of steps grows, the probability of reaching a specific node (specific asset price) converges to a LogNormal distribution. To verify this, observe first that the factor in front of the probability of reaching each node is equal to the number of *paths* leading to that node. On the illustration above, for example, there are two paths leading to the fi-

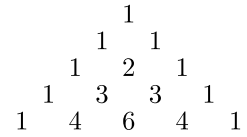


Figure 7.4.: Five first lines of Pascal's triangle.

nal middle node (up-down and down-up) whence the $2pq$ probability. It is easy to check by induction that the number of paths is given by the rows of Pascal's triangle[1] (binomial coefficients):

Explicitly, the probability to get to the kth node at the Nth step is given by

$$\left(\begin{array}{c} N \\ k \end{array} \right) p^k q^{N-k}, \qquad \text{for } k = 0, \ldots, N$$

where the first term is the N-choose-k function (also sometimes written C_k^N). The number N-choose-k is the kth entry of the Nth row of the Pascal's triangle which can be computed using:

$$\left(\begin{array}{c} N \\ k \end{array} \right) = \frac{N!}{k!(N-k)!}$$

where $N! = N \times (N-1) \times \cdots \times 1$. This function is implemented in `scipy.misc.comb`. Let's for example compute the number of paths leading to the end nodes of a 4-step model:

```
# note: there are N+1 final nodes when considering N steps
nPaths = lambda N: [comb(N, k) for k in range(0, N+1)]
print(nPaths(4))
```

You now have a direct way of computing the probability of reaching the kth node at the Nth step, and you can also observe that the return at that node is given by

$$\frac{S_T(k)}{S_0} = u^k d^{N-k}$$

For example you can verify (either by drawing the tree and using the formulas) that in a 3-step model this gives:

[1] https://en.wikipedia.org/wiki/Pascal's_triangle

$$\text{returns: } [d^3, ud^2, u^2d, u^3], \quad \text{with probability } [q^3, 3p^2q, 3pq^2, p^3]$$

where $q = (1-p)$ because the weights are normalised.

You can now check that the binomial tree distribution converges to a Log-Normal distribution. Use the code below to plot the returns and their associated probability. Modify the constants in the code (especially the number of steps) and try again.

```
endProbas  = lambda p, N:    [comb(N,k)*(p**k)*(1-p)**(N-k) for k in range(0,N+1)]
endReturns = lambda u, d, N: [(u**k)*(d)**(N-k) for k in range(0,N+1)]
# enter some arbitrary values
p = 0.6  # probability of going up
u = 1.03 # multiplicative "up factor"
d = 0.99 # multiplicative "down factor"

# number of steps (play with this number)
N = 100
returns = endReturns(u,d,N)
probas  = endProbas(p,N)
# display probability distribution function
plt.figure(figsize=(8, 6))
plt.scatter(returns, probas)
plt.xlabel("Returns", fontsize=12)
plt.ylabel("Likelihood", fontsize=12)
```

This leads to the following figure where you may recognise the shape of a LogNormal distribution:

You can show this link more clearly by looking at the log-returns which converge to a Normal distribution:

```
from numpy import trapz

logreturns = np.log(returns)

# the theory gives us the mean and variance
# at the limit when N goes to infinity
# the last line is to "normalize" so that the curves and
# the points are on the same scale
# (you can safely ignore all this)
th_mean = N*(p*np.log(u)+(1-p)*np.log(d))
th_var  = N*p*(1-p)*np.log(d/u)**2
pr_norm  = probas / abs(trapz(logreturns,probas))

# show the possible log-returns with their weights
```

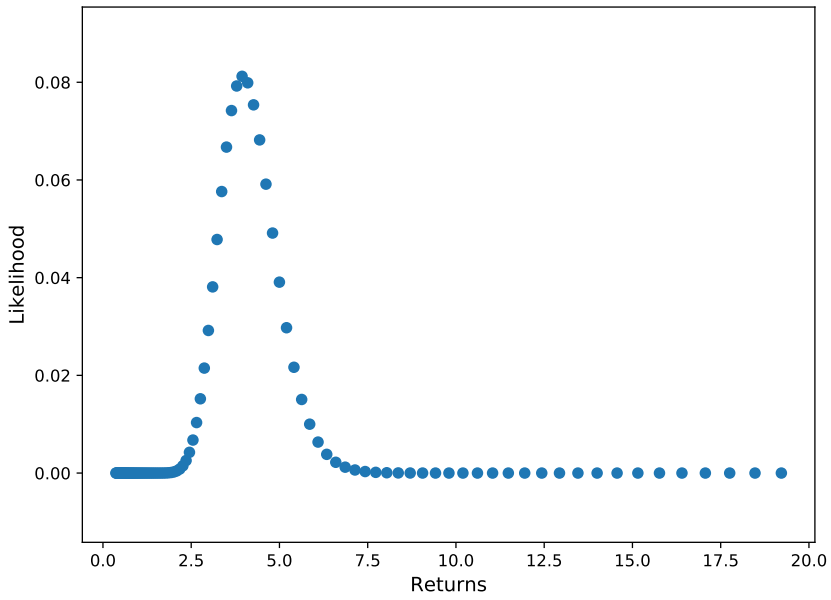


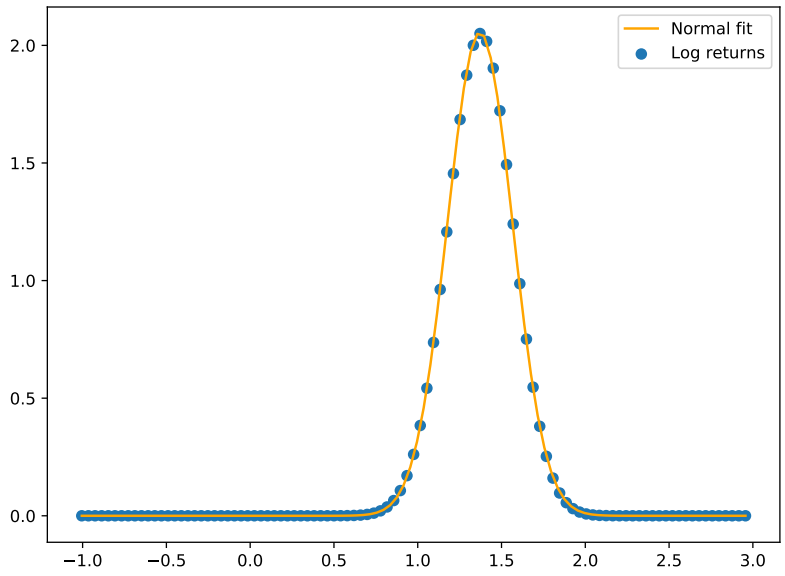
Figure 7.5.: End returns and associated probabilities for a binomial tree with $N = 100$ steps.

```
plt.figure(figsize=(8, 6))
plt.scatter(logreturns, pr_norm, label="Log returns")

# show the normal distribution with the theoretical mean and variance
# computed above
xx = np.linspace(np.min(logreturns),np.max(logreturns),100)
yy = norm.pdf(xx,th_mean,np.sqrt(th_var))

plt.plot(xx, yy, label="Normal fit", color="orange")

plt.legend()
```



You will see soon how to calibrate the parameters of the binomial tree in order to recover precisely the same LogNormal model you have already encountered. But before doing that, let's look at how to use a binomial tree in order to price a derivative which will lead more naturally to the model calibration.

7.3. Pricing with a binomial tree

Risk-neutral probability in a tree

The only probability we need to consider in the tree when pricing is the *risk-neutral probability* $p^\star$. In order to obtain it, remember first that, in the risk-neutral world, the underlying price S_t must obey

$$S_t = \Phi_r(t,T)\mathbb{E}_t^\star[S_T]$$

If we model the evolution of the price of an asset over a time τ with a tree with N steps then each branching corresponds to a time span τ/N which we denote Δt. The relation above, must hold for $t = 0$ and $T = \Delta t$ (the first step in the tree). This gives:

$$S_0 = \Phi_r(0,\Delta t)(p^\star u + (1-p^\star)d)S_0$$

Solving for $p^\star$ and using the usual form for the computation of interests, you get the risk-neutral probability of going up in a binomial tree:

$$p^\star = \frac{\exp(r\Delta t) - d}{u - d}$$

This is the same expression than the one you had obtained in the section about replication pricing. In the rest of this section, we will show how to price with a binomial tree assuming u, d are given and show, later, how they can be set in a way that makes these prices agree with the LogNormal model.

European type derivative

Consider a binomial tree modelling the evolution of the price of an asset over a time τ with N steps. Assume u and d are given (you will see how to calibrate those at the next point). Pricing any derivative depending upon that asset is rather simple:

1. Follow the tree forwards to compute the *intrinsic value* of the derivative at each one of the final nodes (corresponding to time T).

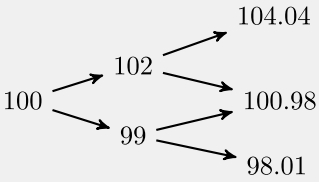
2. Follow the tree backwards to price the derivative at every nodes using the risk-neutral probability and discounting (i.e., using arbitrage-free pricing).

If you follow this process back until the root node, you get a single value which is the arbitrage-free price of the derivative at the initial time t. It's best to see this in action and to code it!

Example

Consider the following model, asset and option:

- *Model:* a recombining two-step model with $u = 1.02$ and $d = 0.99$,
- *Asset:* an asset with current price $S_0 = 100$,
- *Option:* a call with a strike price $K = 99$ maturing at $T = 1$,
- *Interest rate:* for simplicity, assume the interest rate is 0 or, in other words, $\Phi_r = 1$.



The top node comes from $100 \times 1.02 \times 1.02 = 104.04$. The risk-neutral probability is given here by: (with $\Phi_r = 1$)

$$p^\star = \frac{1-d}{u-d} = \frac{1}{3}$$

The call at the top point is worth (104.04-99)=5.04. Similarly, the middle node is worth 1.98 and the bottom node is worth 0. Merging the top and the middle node, the risk-neutral price is:

$$5.04p^\star + 1.98(1-p^\star) = 3.0$$

Merging the bottom and the middle node, the risk-neutral price is 0.66. Merging those two nodes again you get: 1.44. To visualise this:

So the model gives 1.44 as the price of the derivative at time $t = 0$.

Here is some code reproducing the results above. Can you generalise it to an arbitrary number of steps?

```
# EUROPEAN CALL
S0 = 100
K  = 99
```

```
u  = 1.02
d  = 0.99
# Go forward in the tree for two steps:
S_step1 = S0*np.array([d,u])
S_step2 = S0*np.array([d**2,u*d,u**2])

# Compute risk-neutral probability (r=0)
p_star = (1-d)/(u-d)

# Go backward in the tree with a call
C_step2 = np.array([max(S_step2[i]-K,0.) for i in range(0,3)])
C_step1 = np.array([ p_star*C_step2[1] + (1-p_star)*C_step2[0],
                     p_star*C_step2[2] + (1-p_star)*C_step2[1]])
C_step0 = p_star*C_step1[1]  + (1-p_star)*C_step1[0]

for step in ("Forward", S0, S_step1, S_step2,
             "Backward",C_step2, C_step1, C_step0):
    print(step)
```

Calibration

The tree model simplifies the evolution of the market by looking at a finite number of possible prices for the asset and assigning probabilities for each of those. The range of values is governed by the number of steps and the up and down multiplicative factors u and d.

These two values must be set in such a way that the tree converges to the same LogNormal model than the one observed empirically, in the risk-neutral world. In that world, we don't care about the mean since it is already fixed by the risk-neutral pricing (here, via fixing $p^\star$), so we just have to match the variance parameter σ^2. Remember that the returns can be modelled with a LogNormal distribution where the variance grows linearly with the time span considered. It can be shown that if

$$u = \exp(\sigma\sqrt{\Delta t}) \qquad \text{and} \qquad d = 1/u$$

then, the binomial tree converges to a LogNormal distribution with the same variance as the one observed. Note that in this parametrisation, there is effectively only one degree of freedom to consider. (More details are available in the appendix if you are interested.)

Dealing with early-exercise

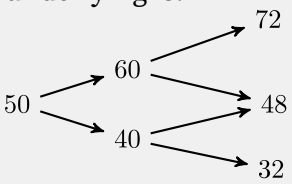
When considering derivatives with early exercise, the principle is very much the same with a slight twist:

- Follow the tree forwards to compute the intrinsic value of the derivative at each one of the final nodes (write this V^{intr}_{node}).
- Follow the tree backwards and at each node:
  - compute the value of the derivative using arbitrage-free pricing as in the binomial tree (call this V^{af}_{node})
  - the value kept at the node is $\max(V^{af}_{node}, V^{intr}_{node})$ (i.e., V^{af}_{node} if it is not in your advantage to exercise early, V^{intr}_{node} if it is).

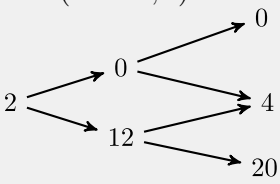
Let's try below with an American put with 2-steps. Try to reproduce the process in Python. Note that we introduce an interest rate here in this example (only if $r > 0$ can it be in your advantage to exercise early, cf. the bonus section in the module on derivatives for more information).

Example

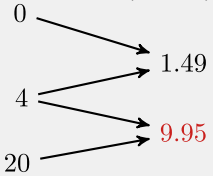
Consider an American put over a time of $\tau = 2$ years with underlying price 50 at $t = 0$, strike price 52 and say that $u = 1.2$, $d = 0.8$ and $r = 0.05$ (5%). For $N = 2$ steps, the tree for the evolution of the underlying is:



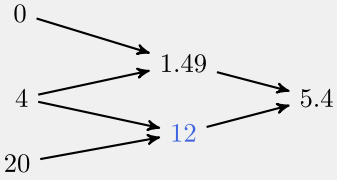
The *intrinsic values* of the put are obtained by just computing $\max(K - S, 0)$ with S the price at each node:



Now if you go backwards one step, using the same approach as before (with, here, $p^\star = (\exp(0.05) - 0.8)/(1.2 - 0.8) \approx 0.6282$)



The value in red (9.95) is less than the value of exercising early at that node (which would be 12) so we replace it by 12 and the final tree is:



note that there is no replacement at the final node since $5.4 > 2$.

Remark: had it been a European put, the method would have given you an original price of 4.6 which is less than the price here. Can you give an intuitive explanation as to why this is (always) the case?

7.4. Summary and discussion

Gathering elements from the previous points you have seen that:

- *Binomial trees* approximate the LogNormal distribution over a finite set of values,
- *Pricing using a tree* is simple and can be done at any intermediate time point allowing to price any derivatives directly. This is the main appeal for this model and its extensions.
- Binomial trees *converge to the LogNormal model* when the number of steps grows if the tree is calibrated appropriately.

Ingredients needed for coding the binomial tree

Here we summarise the key elements needed to build a binomial tree (not all are necessary depending on how you implement it):

Term	Definition	Python functions
number of steps	N	
time span, time per step	$\tau, \Delta t = \tau/N$	
up,down factors	$u, d = \exp(\pm\sigma\sqrt{\Delta t})$	`numpy.exp`, `numpy.sqrt`
risk-neutral probability of going up	$p^\star = \frac{\exp(r\Delta t)-d}{u-d}$	`numpy.exp`, `numpy.sqrt`
probability to get to the kth node at Nth step	$C_k^N p^k (1-p)^{N-k}$	`scipy.misc.comb`
price of the underlying at that node	$S_T(k) = u^k d^{N-k}$	`numpy.pow`

with these formulas you can

1. follow the tree forwards to compute the intrinsic value of the derivative at each node for N steps,
2. follow the tree backwards to price the derivative at each node until the original price.

8. Advanced Models

8.1. The trinomial tree

The trinomial tree is a simple extension of the binomial tree where, at each node, you can go up (factor u), down (factor d) or stay at the same price. In order for the tree to be *recombining*, we can pose $u = 1/d$. The tree then looks like:

The main advantage of the trinomial tree over the binomial tree is that it is a more accurate and more flexible model and can exhibit better numerical properties. In particular, there are many ways in which the trinomial tree can be parametrised which are consistent with the same LogNormal model and this can be useful when calibrating the tree with a specific purpose in mind such as the pricing of a given derivative.

Pricing in the trinomial tree

Pricing is done in much the same way as in the binomial tree, the main parameter that is modified is the risk-neutral probability. Indeed, there

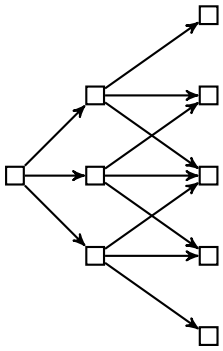


Figure 8.1.: Recombining trinomial tree.

are now three branches to consider at each node and hence three probabilities. + If you write $p_u^\star$ the probability of going up, $p_d^\star$ the probability of going down then the probability of staying at the same level must be $p_m^\star = 1 - (p_u^\star + p_d^\star)$ because the weights are normalised. This means there are two degrees of freedom (instead of one in binomial trees). As a result, you need to look at two consecutive steps in order to compute the risk-neutral probabilities $p_u^\star$ and $p_d^\star$ (as opposed to one step in binomial trees). The key is still to verify the principle of risk-neutral pricing. Over the first layer this gives:

$$S_0 = \Phi_r S_0(up_u^\star + p_m^\star + dp_d^\star)$$

where $\Phi_r = \Phi_r(0, \Delta t) = \exp(-r\Delta t)$.
Rearranging leads to

$$\Phi_r^{-1} = up_u^\star + p_m^\star + dp_d^\star$$

Following the same logic, the second layer gives:

$$\Phi_r^{-2} = u^2 p_u^{\star 2} + 2up_u^\star p_m^\star + 2p_u^\star p_d^\star + p_m^{\star 2} + 2dp_m^\star p_d^\star + d^2 p_d^{\star 2}$$

Recalling that $\Psi_r = \Phi_r^{-1}$, you can check that the solution to these two equations is

$$p_u^\star = \left(\frac{\sqrt{\Psi_r} - \sqrt{d}}{\sqrt{u} - \sqrt{d}}\right)^2 \qquad p_d^\star = \left(\frac{\sqrt{u} - \sqrt{\Psi_r}}{\sqrt{u} - \sqrt{d}}\right)^2$$

You want the trinomial model (like the binomial one) to converge to the LogNormal model. This can be achieved by setting u and d to

$$u, d = \exp(\pm\sigma\sqrt{2\Delta t})$$

Try to adapt the code written for the binomial model to the trinomial one. Compare the two when pricing a European call with respect to using the Black-Scholes formula, what do you observe? In particular, can you compare the convergence of the prices given by the two methods to the analytical price when the number of steps increases?

Other parametrisation

There are other parametrisations of the trinomial tree which are compatible with the LogNormal model each adapted to different contexts. For example, in the *Kamrad and Ritchken* parametrisation an *stretch* parameter $\lambda \geq 1$ is added and the up and down factors are set to

$$u, d = \exp(\pm\lambda\sigma\sqrt{2\Delta t})$$

The stretch λ can be chosen such that exact values are attained in the tree. This improves the accuracy of the pricing when considering barrier options. You can find details about this parametrisation in the original article by *Kamrad and Ritchken* (see references).

8.2. Towards more advanced models

A brief introduction

The aim of this section is to give you an overview of how what we have seen so far can be placed in a larger context. Remember the purpose of this course was to provide an introduction!

If you have never met Partial Differential Equations, you might want to skim through this point to just get a feel for some of the more advanced tools and techniques that are used in the world of quantitative finance. If you have met those before the following point should help you gaining a wider perspective on pricing models.

Replicating the trinomial tree

So far we have always considered that you are computing the value of a derivative at the same time as you are buying it and hence you know the initial (spot) price of the underlying asset. However, it may be interesting to pre-compute the tree in which case the price of the underlying asset when the contract is instantiated is unknown. You may also want to compute the Δ of a derivative which, as you may remember, is given by the variation in derivative price over a variation in the price of the underlying.

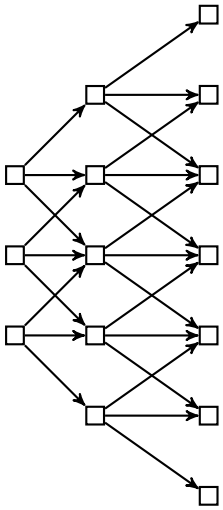


Figure 8.2.:

To simplify the presentations, say that we model the initial price of an underlying with three possible values: S_0, uS_0 and dS_0. With this convenient model choice, we do not need to compute one tree for each of those three initial prices since they interlace:

The pricing process remains the same (and so does the parametrisation), you will just get three prices for the derivative corresponding to the different initial prices of the underlying asset. In particular, this directly gives you a way to approximate the Δ of the derivative around S_0.

From tree pricing to grid pricing

The replication described above can be done over many initial prices and with many steps in the tree. When going through the grid backwards to perform pricing, you almost always deal with the following case:

where i indexes the rows and j the columns (steps). There are a small portion of the nodes where one of the corner is missing, but it is essentially a grid otherwise.

As before, $V_{i,j}$ is given by arbitrage-free pricing. Thus, writing $\Phi_r = \exp(-r\Delta t)$ and $\alpha = p_u^\star \Phi_r$, $\beta = p_m^\star \Phi_r$ and $\gamma = p_d^\star \Phi_r$, we have:

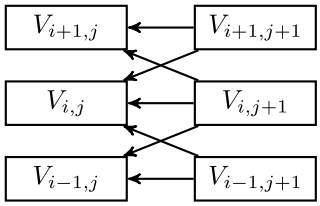


Figure 8.3.:

$$V_{i,j} = \alpha V_{i+1,j+1} + \beta V_{i,j+1} + \gamma V_{i-1,j+1}$$

This can be expressed as a matrix-vector product:

$$V_{\cdot,j} = \begin{bmatrix} \ddots & & & & \\ \alpha & \beta & \gamma & & \\ & \alpha & \beta & \gamma & \\ & & \alpha & \beta & \gamma \\ & & & & \ddots \end{bmatrix} V_{\cdot,j+1}$$

where $V_{\cdot,j}$ is the column of values $V_{i,j}$ for all i and a fixed j. This matrix-vector form allows you to compute the columns of grid from right (maturity time) to left (initial time).

Not only does the form above let you work column-wise, it is also the particular form of a *finite-difference solver*.

The Black-Scholes PDE

Finite-difference solvers are useful to approximate solutions to *partial differential equations* (PDE). The method above can be seen as a particular instance of such a solver for a particular PDE called the *Black-Scholes PDE*. It can be shown that pricing any derivative under the LogNormal model corresponds to solving that PDE. Writing V the value of the derivative, the PDE is:

$$\frac{\partial V(S,t)}{\partial t} + rS\frac{\partial V(S,t)}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V(S,t)}{\partial S^2} - rV(S,t) = 0$$

The domain is $S > 0$, $t \in [0, T]$ and the boundary conditions are:

- *Terminal condition*: $V(\cdot, T)$ must equal the payoff of the derivative.
- *Left-boundary* (for a call): $\lim_{S \to 0} V(S,t) = 0$.
- *Right-boundary* (for a call): $\lim_{S \to \infty} V(S,t) = S - K\exp(-r(T-t))$.

Approximating solutions to PDE using finite-difference solvers (and finite-element methods in general) is a well studied process with well known, efficient algorithms. This can be leveraged to come up with more advanced pricing methods.

8.3. The need for performance computing

Option trading is constituted of roughly two segments: the *over the counter* (OTC) market and electronic exchanges.

In the OTC market there are typically few transaction but with big notional value (value of the leveraged position). These transactions can contain many derivatives including exotic options, often with special features for just one customer. Valuing these trades is complex. Fortunately, because these transactions are infrequent, it is possible to spend days on their valuation.

Screen trading on the electronic exchanges is a completely different matter: highly automated, very fast trade of low volume, standardised vanilla options (mostly European or American style). In this context, in order to be competitive, you need to compute the price of options quickly. Unfortunately, the parameters for option prices can change quickly – especially the price of the underlying. As a result, you also need to recompute the price of options quickly.

Using wider trees, finite-difference schemes and interpolation helps: you spend more time upfront to compute the price, but you can quickly reprice a derivative when the value of the underlying changes. However, this is not sufficient.

Indeed, remember the volatility smile: the underlying price and its volatility changes. Let's say you solved the PDE for an underlying price $S = 100$

and volatility $\sigma_{100} = 20\%$. Later, the price moves to to $S = 101$ while the parametrised volatility moves to $\sigma_{101} = 19\%$. You need to solve the PDE again! Even though you need to solve the PDE again, you should not discard the result of the first computation: the price might drop back to 100. In general, you can avoid some computation by *caching* results: storing the result of previous computation so they can be fetched later.

Coming up with smart schemes for caching results and interpolating between them is the way to keep up with fast markets and be able to be the first to react to changing market conditions.

8.4. Discussion

In this section you saw that it is easy to extend the simple binomial model and that, based on the application of interest, a specific model such as the trinomial model may be preferred. There are many more numerical models out there, each of which with their advantages for pricing specific derivatives or for recovering specific properties.

You also saw that the problem of pricing a derivative under the LogNormal model is equivalent to that of solving a PDE (the Black-Scholes PDE). It is therefore possible to draw directly from the wealth of algorithms available for approximating solutions to such PDE.

Finally you saw that in order to stay competitive on electronic exchanges, it was crucial to have efficient computation methods and clever interpolation schemes to be both accurate and quick.

A. Some Intuition on Risk Neutral Pricing

In this section, we revisit the central equation for pricing derivatives, namely

$$V_t = \Phi_r(t,T)\mathbb{E}_t^\star[V_T].$$

We leave the formal derivation using a self financing portfolio that replicates the cash flows of derivatives to the textbooks. Nevertheless, we would like to give you an intuition about what is happening here using a simple example.

We have empirically established a model for the log returns of the underlying namely that they are distributed normally:

$$\log \frac{S_{t+\tau}}{S_t} \sim \mathcal{N}\left(\mu\tau, \sigma^2\tau\right).$$

It would seem very natural to price derivatives using the observed (real world) distribution with it's drift and variance.

Using a simple forward we can just do that. Remember that the payoff of the forward is $V_T = S_T - K$ so that we can calculate the price as the expectation

$$V_t = \Phi_r \mathbb{E}\left[S_T - K\right] = \Phi_r\left(\mathbb{E}\left[S_T\right] - K\right).$$

Note how we dropped the * in the notation: here we work in the real world measure, not the risk neutral world. The difference in the results will help us understand the concept of risk neutral measures.

In the appendix on moments we calculate the first moments using the real world distribution which we can use here:

$$V_t = \Phi_r \left(e^{\mu(T-t)+\frac{1}{2}\sigma^2(T-t)} S_t - K \right).$$

Naturally we see the risk free rate r pop up as we discount the expected future cash flow and we also see the real world growth rate of the underlying μ and it's standard deviation σ appear. But is this the correct price of a forward?

Let's remember what a forward contract is: if you buy a forward contract you agree to buy the underlying at a future time T for price K, which we fix at the time t of entering the contract. This looks like a pure bet but people in the market have a way to take chance out of the picture: they use *replication*.

Replication is a technique in which a market participant uses a portfolio of (usually) simpler financial products to exactly reproduce the behaviour of a more complex, e.g. derivative, product. We do have the law of one price and if any portfolio of simpler products behaves just the same as the derivative, both need to have the same price[1].

Back to our forward contract: we do not know the value of the forward contract $V(t)$ at time t when buyer and seller enter. But we do know it's payoff, i.e. the value of the forward contract at it's maturity T:

$$\text{Payoff of a Forward at T:} \qquad V(T) = S(T) - K.$$

There are two things happening at maturity, we exchange the stock and we exchange K in cash. So how can we achieve the same actions without buying a forward?

At time t we buy the stock. This will cost us $S(t)$ the price of the stock at time t. Holding the stock means that we will have the stock at maturity T at which moment it will be worth $S(T)$. This replicates one part of the payoff.

Now we can reproduce the cash part of the payoff by getting a loan of a bank. We plan to pay the money back at time T and in order to replicate the behaviour of the forward, we want to pay back K. So what is the value at time t of a bank loan that is paid back at time $T > t$? Given this is a risk free bank account, the rate is r so we have to take out a loan of $e^{-r(T-t)}K$ at

[1] If not I can trade the portfolio against the derivative which constitutes an arbitrage opportunity.

time t. At time T we now have an exact replication of the forward's payoff, we are long the stock and have to pay K.

If we replicate the forward at time T and nothing happens between the time t we construct the replication portfolio and maturity, the value of the portfolio at time t must be identical to the value of the forward at time t, $V(t)$.

At time t our replication portfolio has the value

$$V(t) = S(t) - e^{-r(T-t)}K.$$

We are long the stock and short a bank account worth $e^{-r(T-t)}K$. By the law of one price the value of the replication portfolio must be identical to the value of the forward so that the above equation gives us the value of a forward contract.

Notice how our replication portfolio is risk free: on the one hand we are short a (by definition) risk free bank account, on the other we hold a stock. The stock position looks risky but is not as we bought it at the time we entered the risk contract for the known price $S(t)$ and, as we have a binding forward agreement, we are bound to pass the stock along at maturity. It's price $S(T)$ at maturity is not concerning us.

Given the law of one price the forward price given by the replication argument is *enforceable* by arbitrage. So that really looks like the correct price of the forward. That leaves the question how we can reconcile this result with our previously calculated expectation under the real world measure.

Now let us contrast our two solutions, the one under the real world measure using the empirically estimated distribution of the log returns, the other gained by replication and the law of one price:

$$\begin{aligned}\Phi_r\mathbb{E}\left[S_T - K\right] &= e^{-r(T-t)}\left(e^{\mu(T-t)+\frac{1}{2}\sigma^2(T-t)}S_t - K\right)\\ \Phi_r\mathbb{E}^\star\left[S_T - K\right] &= S(t) - e^{-r(T-t)}K.\end{aligned}$$

Now if we rearrange the first equation we see that

$$\Phi_r\mathbb{E}\left[S_T - K\right] = e^{\left(\mu+\frac{1}{2}\sigma^2-r\right)(T-t)}S(t) - e^{-r(T-t)}K,$$

so that if we would set

$$\mu = r - \frac{1}{2}\sigma^2,$$

we would regain the replication price! By changing the drift (growth) of our stock process we changed our measure from the real world to the risk neutral world. The variance of the underlying process stays the same. In the risk neutral world our log returns are still normally distributed but now with a different mean:

$$\log \frac{S_{t+\tau}}{S_t} \sim \mathcal{N}\left((r - \sigma^2/2)\tau, \sigma^2\tau\right).$$

The replication removed all risk from the forward contract which in turn implied a different drift for our underlying. When we now look at the expectation of the underlying:

$$\mathbb{E}^\star\left[S_T\right] = S(t)e^{r(T-t)}.$$

In the risk neutral world our stock just returns the risk free rate!

The replication of the forward contract is an example of a *static* hedge: you had to construct your replication portfolio of stock and bank account only once and never needed to adjust it in any way. For more complex derivative like options, we can also build a replication portfolio. But in the case of options we need to continuously adjust our portfolio which is thus called *dynamic hedging*. The overall effect is the same, we hedge away the risk which explains why in our pricing equations we see the risk free rate r and not the mean of the historical distribution.

B. Deriving the Black-Scholes formula for the European call

The European call option has value at maturity given by

$$V^{\text{call}}(S_T) = \max\{0, (S_T - K)\},$$

and its expected value at time t under the risk-neutral distribution is:

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp(-\epsilon^2/2) \max\left[0, S_t \exp\left((r - \sigma^2/2)\tau + \sigma\sqrt{\tau}\epsilon\right) - K\right] \mathrm{d}\epsilon.$$

You may recognise the term $\exp(-\epsilon^2/2)/\sqrt{2\pi}$ corresponding to the $\mathcal{N}(0,1)$ distribution, the rest is obtained by plugging the risk-neutral form of S_T in the formula. The $\max$ is different from zero only when

$$K \leq S_t \exp\left((r - \sigma^2/2)\tau\right) \exp(\sigma\sqrt{\tau}\epsilon).$$

Rearranging a bit leads to an equivalent condition in ϵ:

$$\epsilon \geq a = \frac{\log(K/S_t) - (r - \sigma^2/2)\tau}{\sigma\sqrt{\tau}}.$$

Therefore, you can now write:

$$\begin{aligned} \mathbb{E}_t^\star[H(S_T)] \quad = \quad & S_t \exp[(r - \sigma^2/2)\tau] \int_a^\infty \tfrac{1}{\sqrt{2\pi}} \exp(-\epsilon^2/2 + \sigma\sqrt{\tau}\epsilon)\mathrm{d}\epsilon \\ & -K \int_a^\infty \tfrac{1}{\sqrt{2\pi}} \exp(-\epsilon^2/2)\mathrm{d}\epsilon. \end{aligned}$$

Both integrals can easily be related to $\Phi(x) = 1/\sqrt{2\pi} \int_{-\infty}^{x} \exp(-\epsilon^2/2)\mathrm{d}\epsilon$ which you have already met. For this note that

- $\int_a^\infty \exp(-\epsilon^2/2)\mathrm{d}\epsilon/\sqrt{2\pi} = \Phi(-a)$ (by symmetry),
- the first integral can be related to $\int_a^\infty \exp(-(\epsilon - b)^2/2)\mathrm{d}\epsilon/\sqrt{2\pi}$ and therefore to $\Phi(b - a)$.

This finally leads to the Black-Scholes formula for the price of a call:

$$\begin{aligned} V_t^{\text{call}} &= \exp(-r\tau)\mathbb{E}_t^\star[V_T^{\text{call}}] \\ &= S_t\Phi(d_1) - K\exp(-r\tau)\Phi(d_2). \end{aligned}$$

where $d_2 = -a$ and $d_1 = \sigma\tau - a$ or explicitly

$$d_{1,2} = \frac{\log\left(\frac{S_t}{K}\right) + (r \pm \sigma^2/2)\tau}{\sigma\sqrt{\tau}}.$$

C. Calibrating a binomial tree in spot space

Here we will derive all the equations for the binomial tree in full detail. For trees built in spot space this is unfortunately quite an exercise. Deriving the equations for trees built in $\log S$ is much simpler, just try it yourself.

Starting out from a single step in the binomial tree we have just two free parameters: the probability to go up p and the factor u by which we go up with the spot price. The down jump factor d is fixed by our requirement to have a recombining tree, $d = \frac{1}{u}$.

We will fit these two parameters to the first two moments of our underlying distribution in a single binomial step of given length Δt[1]:

$$\begin{aligned} \mathbb{E}[S] &= Se^{r\Delta t} \\ \mathbb{E}\left[S^2\right] &= S^2 e^{\left(2r+\sigma^2\right)\Delta t}. \end{aligned}$$

The first moment of this single step binomial tree is given by

$$\mathbb{E}[S] = Se^{r\Delta t} = puS + (1-p)\,dS.$$

We can use this equation to derive the probability p of moving up as:

$$p = \frac{\Psi - d}{u - d},$$

with $\Psi = e^{r\Delta t}$.

Now we have just one parameter left, namely u, and we use the second moment to fix it. The second moment in the one step binomial tree is given by

$$\mathbb{E}\left[S^2\right] = pu^2S^2 + (1-p)\,d^2S^2.$$

[1] Δt is yet another parameter but it is fixed by the number of steps n and the time to maturity: $\Delta t = \frac{T-t}{n}$

The second moment in the binomial tree needs to match the second moment of the distribution so that we have:

$$\begin{aligned}
\mathbb{E}\left[S^2\right] = pu^2S^2 + (1-p)\,d^2S^2 &= S^2e^{\left(2r+\sigma^2\right)\Delta t} \\
pu^2 + d^2 - pd^2 &= \Psi^2 e^{\sigma^2\Delta t} \\
p\left(u^2-d^2\right) + d^2 &= \Psi^2 e^{\sigma^2\Delta t} \\
\tfrac{\Psi-d}{u-d}\,(u-d)\,(u+d) + d^2 &= \Psi^2 e^{\sigma^2\Delta t} \\
(\Psi-d)\,(u+d) + d^2 &= \Psi^2 e^{\sigma^2\Delta t} \\
\Psi u + \Psi d - 1 - d^2 + d^2 &= \Psi^2 e^{\sigma^2\Delta t} \\
\Psi u + \Psi u^{-1} - 1 &= \Psi^2 e^{\sigma^2\Delta t} \\
\Psi u^2 + \Psi - u &= u\Psi^2 e^{\sigma^2\Delta t} \\
u^2 + 1 - u\Psi^{-1} - u\Psi e^{\sigma^2\Delta t} &= 0 \\
u^2 - u\left(\Psi^{-1} + \Psi e^{\sigma^2\Delta t}\right) + 1 &= 0 \\
u^2 - 2\beta u + 1 &= 0
\end{aligned}$$

where we have used

$$\beta = \frac{1}{2}\left(e^{-r\Delta t} + e^{r\Delta t+\sigma^2\Delta t}\right).$$

We can solve the quadratic equation to get

$$u = \beta \pm \sqrt{\beta^2 - 1}.$$

As we always have $\beta > 1$ and we want our up step u to be greater 1, so we select

$$u = \beta + \sqrt{\beta^2 - 1} > 1.$$

We can make our life easier by expanding the exponentials in β using

$$e^a = 1 + a + \frac{1}{2}a^2,$$

after dropping all quadratic terms involving Δt^2 we get

$$\beta \approx 1 + \frac{1}{2}\sigma^2\Delta t.$$

Using the approximation in our expression for u and again dropping any quadratic term we finally get

$$\begin{aligned} u &= \beta + \sqrt{\beta^2 - 1} \\ &= 1 + \tfrac{1}{2}\sigma^2\Delta t + \sqrt{\left(1 + \tfrac{1}{2}\sigma^2\Delta t\right) - 1} \\ &= 1 + \tfrac{1}{2}\sigma^2\Delta t + \sqrt{1 + \sigma^2\Delta t + \tfrac{1}{4}\sigma^4\Delta t^2 - 1} \\ &= 1 + \tfrac{1}{2}\sigma^2\Delta t + \sqrt{\sigma^2\Delta t} \\ &= 1 + \tfrac{1}{2}\sigma^2\Delta t + \sigma\sqrt{\Delta t} \\ &= e^{\sigma\sqrt{\Delta t}}. \end{aligned}$$

So by calibrating our binomial tree to the first two moments of the distribution we can fix our parameters to

$$\begin{aligned} p &= \tfrac{\Psi - d}{u - d} \\ u &= e^{\sigma\sqrt{\Delta t}} \\ d &= \tfrac{1}{u}. \end{aligned}$$

D. Calibration of the Binomial tree

When the number of steps grows to infinity we would like, in the risk-neutral world, to match the mean and variance corresponding to the Log-Normal model. Specifically $(r - \sigma^2/2)\tau$ (mean) and $\sigma^2\tau$ (variance). In the binomial tree, the possible returns after N-steps are $u^k d^{N-k}$ for $k = 0, \ldots, N$. The corresponding log-returns are

$$k \log u + (N - k)d,$$

and their expected value in the risk-free world is thus $N \log d + \mathbb{E}^\star(k) \log(u/d)$. Here, k is a binomial random variable with mean $Np^\star$ and variance $Np^\star(1 - p^\star)$. Therefore, at the limit $N \to \infty$ we must have (with $\Delta t = \tau/N$)

$$\begin{aligned} \log d + p^\star \log(u/d) &= (r - \sigma^2/2)\Delta t \\ p^\star(1 - p^\star)[\log(u/d)]^2 &= \sigma^2 \Delta t. \end{aligned}$$

These equations are simplified if we use the symmetric parametrisation $u = 1/d$, and remember that we have an explicit form for $p^\star$. It's a bit cumbersome to solve the corresponding equations by hand but the least we can do is check that everything holds for the standard parametrisation – i.e. when $u, d = \exp(\pm\sigma\sqrt{\Delta t})$

The code below illustrates how it can be easily checked.

```
# some arbitrary values (modify at will)
sigma = 0.2
r     = 0.05
T     = 3.0

psi   = lambda N:     np.exp(r*T/N)
u_th  = lambda N:     np.exp(sigma*np.sqrt(T/N))
pstar = lambda N,u,d: (psi(N)-d)/(u-d)
```

```
def tester(N):
    u  = u_th(N)
    d  = 1/u
    ps = pstar(N,u,d)
    s2 = sigma**2

    # Equations for the mean and variance matching
    # (both going ->0 when N->infinity)
    eq1 = (np.log(d)+ps*np.log(u/d)) - ((r-s2/2)*T/N)
    eq2 = (ps*(1-ps)*(np.log(u/d))**2) - (s2*T/N)

print("With {} time steps, eq1: {}, eq2: {}".format(N,eq1,eq2))
```

If you try the `tester` function with 3, 10 and 100 (for example), you will see that the results tend to zero quickly, as expected.

E. Calculating Moments of a Log Normal Distribution

Several times we used the first two moments of the underlying. This raises the question: how do we calculate moments of a log normally distributed random variable? This can be done in different ways, you can look up the moment generating function and use that or you can calculate the expectation explicitly.

Let's do the latter here. The moments are defined as the expectation of the underlying taken to the power k i.e.: $\mathbb{E}\left[S_t^k\right]$ with $k = 1, 2, \ldots$ giving you the first, second and higher moments.

Let's assume that we know the distribution of the log returns

$$X_t = \log \frac{S_t}{S_0} \sim \mathcal{N}(\mu, \sigma^2).$$

We will rewrite the expectation using X_t so that we get

$$\begin{aligned}
\mathbb{E}\left[S_t^k\right] &= \mathbb{E}\left[S_0^k e^{kX_t}\right] \\
&= S_0^k \int_{-\infty}^{\infty} e^{kX_t} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(X_t-\mu)^2}{2\sigma^2}\right] dX_t.
\end{aligned}$$

Taking the terms in the exponential together and completing the square we get

$$\begin{aligned}
kX_t - \frac{(X_t-\mu)^2}{2\sigma^2} &= \frac{2k\sigma^2 X_t - X_t^2 + 2\mu X_t - \mu^2}{2\sigma^2} \\
&= -\frac{1}{2\sigma^2}\left[X_t^2 - 2\left(k\sigma^2+\mu\right)X_t + \mu^2\right] \\
&= -\frac{1}{2\sigma^2}\left[\left(X_t - \left(k\sigma^2+\mu\right)\right)^2 - \left(k\sigma^2+\mu\right)^2 + \mu^2\right] \\
&= -\frac{(X_t-\mu')^2}{2\sigma^2} + \frac{\mu^2+\left(k\sigma^2+\mu\right)^2}{2\sigma^2} \\
&= -\frac{(X_t-\mu')^2}{2\sigma^2} + \frac{k\left(k\sigma^2+2\mu\right)}{2}
\end{aligned}$$

Putting this back into our expectation we pull part of the exponential in front of the integral to get

$$\begin{aligned}\mathbb{E}\left[S_t^k\right] &= S_0^k e^{\frac{k}{2}\left(k\sigma^2+2\mu\right)} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(X_t-\mu'\right)^2}{2\sigma^2}} dX_t \\ &= S_0^k e^{\frac{k}{2}\left(k\sigma^2+2\mu\right)}.\end{aligned}$$

Here we used the fact that the intergral over the probability density is trivially 1.

Now let's put it to the test and calculate the first two moments for the underlying process, once using historically observed (risky) distribution and once in the risk neutral world:

In the risky world we have the log returns distributed as $\mathcal{N}\left(\mu t, \sigma^2 t\right)$ so that we get

$$\begin{aligned}\mathbb{E}\left[S_t\right] &= S_0 e^{\mu t+\frac{1}{2}\sigma^2 t} \\ \mathbb{E}\left[S_t^2\right] &= S_0^2 e^{2\mu t+2\sigma^2 t}.\end{aligned}$$

In the risk neutral case we have to adjust the drift term so that we the log returns are distributed as $\mathcal{N}\left(\left(r-\frac{1}{2}\sigma^2\right)t, \sigma^2 t\right)$:

$$\begin{aligned}\mathbb{E}\left[S_t\right] &= S_0 e^{rt} \\ \mathbb{E}\left[S_t^2\right] &= S_0^2 e^{2rt+\sigma^2 t}.\end{aligned}$$

F. Some Financial Jargon

Term	Definition
Asset	Something of economic value, examples are stocks, bonds, commodities.
Bond	A bond represents part of a company's or state's debt.
Stock	A stock represents part of a company.
Commodities	Tangible assets, like oil, sugar, pork bellies.
Derivative Index	A financial contract that depends on another asset. The asset the derivative depends on is called the underlying. A measure for the performance of a market (or part of), often calculated using baskets of stocks. Examples are DAX, S&P, CAC.
Underlying	The asset on which a derivative depends. This is often a stock but can also be bonds, indices or commodities.
Short/Long	If you own some asset you are *long* that asset. If you sold asset or owe an asset to someone else, you are *short*.
Bid/Ask	Assets trading in financial markets always have two prices: one at which someone is willing to buy it, the *bid*, or to sell it, the *ask*. Naturally the bid is smaller than the ask: buy low sell high.
Spread	In our case we use spread to mean the difference between the ask and the bid price. The spread is therefore positive.

Term	Definition
Risk Free Rate	The risk free rate describes a interest rate of an asset which is without any risk. Examples would be the yield of government bonds of states which are considered impossible to default, e.g. Switzerland.
Position	A trader holding an asset is said to be in a long position. A trader owing an asset is said to be in a short position.
Hedging	The act of eliminating or at least reducing the risk associated with holding a position. This usually involves building up an offsetting position.