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.

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:

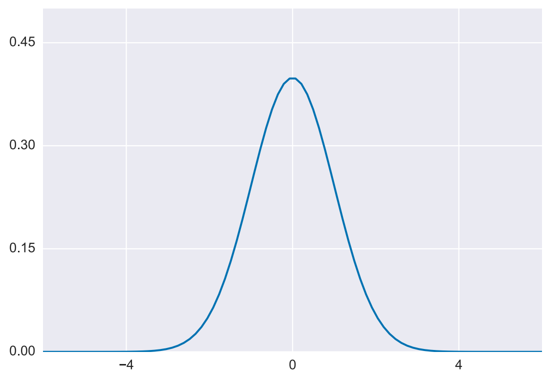

Figure 1.1.: Pdf of a standard Normal distribution

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:

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

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):

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

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:

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

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.