Introduction to NumPy

This part of the book, along with Part 3, outlines techniques for effectively **loading, storing, and manipulating in-memory data** in Python.

Datasets can come from a **wide range of sources** and in a **wide range of formats,** including:

- collections of documents,
- collections of images,
- collections of sound clips,
- collections of numerical measurements,
- or nearly anything else.

Despite this **apparent heterogeneity,** many datasets can be represented fundamentally as **arrays of numbers.**

- For example, images—particularly digital images—can be thought of as simply two-dimensional arrays of numbers representing pixel brightness across the area.
- Sound clips can be thought of as one-dimensional arrays of intensity versus time.
- Text can be converted in various ways into numerical representations, such as binary digits representing the frequency of certain words or pairs of words.

No matter what the data is, the **first step** in making it **analyzable** will be to **transform** it into **arrays of numbers.**

For this reason, **efficient storage and manipulation of numerical arrays** is absolutely fundamental to the process of doing **data science.**

We'll now take a look at the **specialized tools** that Python has for handling such numerical arrays: the **NumPy** package and the **Pandas**

package.

In some ways, **NumPy arrays** are like Python's **built-in `list` type,** but NumPy arrays provide **much more efficient storage and data operations** as the arrays grow **larger in size**.

NumPy arrays form the **core** of nearly the **entire ecosystem of data science** tools in Python.

so time spent **learning to use NumPy effectively** will be valuable no matter what aspect of data science interests you.

You can **import** NumPy and double-check the **version:**

In [ ]:
```
import numpy
numpy.__version__
```

Out[ ]:
```
'1.21.2'
```

You'll find that most people in the SciPy/PyData world will import NumPy using `np` as an alias:

In [ ]:
```
import numpy as np
```

Throughout this chapter, and indeed the rest of the book, you'll find that this is the way we will import and use NumPy.

Reminder About Built-in Documentation

As you read through this part of the book:

- don't forget that **IPython** gives you the ability to quickly **explore the contents** of a package (by using the tab completion feature),
- as well as the **documentation of various functions** (using the `?` character). For a refresher on these, refer back to Help and Documentation in IPython.

For example, to display all the contents of the NumPy namespace, you can type this:

```
In [3]: np.<TAB>
```

And to display NumPy's **built-in documentation,** you can use this:

```
In [4]: np?
```

More detailed documentation, along with tutorials and other resources, can be found at http://www.numpy.org.