Introduction to Data Science

Ethan Meyers

2025-05-22

In September 2021, a significant jump in seismic activity on the island of La Palma (Canary Islands, Spain) signaled the start of a volcanic crisis that still continues at the time of writing. Earthquake data is continually collected and published by the Instituto Geográphico Nacional (IGN). …

Table of contents

1 La Palma Earthquakes

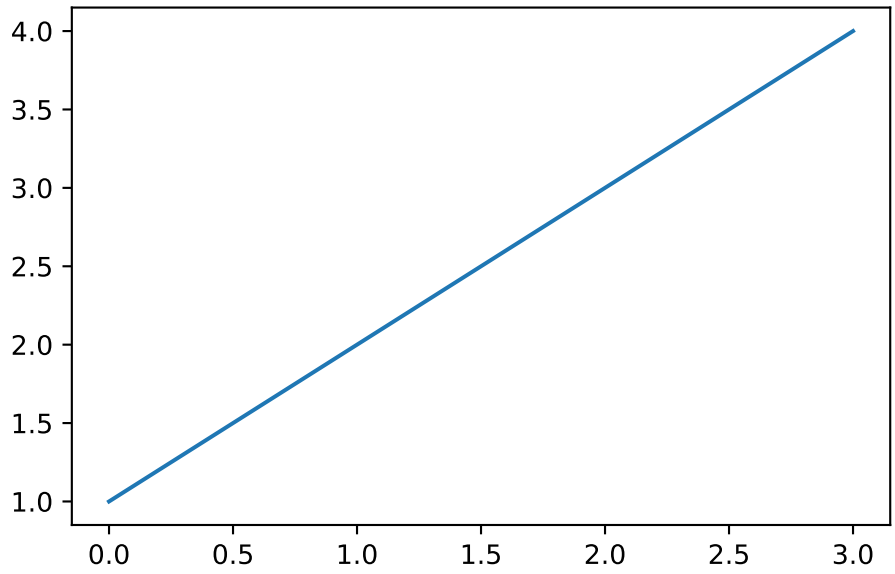
In September 2021, a significant jump in seismic activity on the island of La Palma (Canary Islands, Spain) signaled the start of a volcanic crisis that still continues at the time of writing. Earthquake data is continually collected and published by the Instituto Geográphico Nacional (IGN). …

2 Testing!!! May 22 - update 1

2.1 Introduction

```
import matplotlib.pyplot as plt
import numpy as np
eruptions = [1492, 1585, 1646, 1677, 1712, 1949, 1971, 2021]

plt.plot([1, 2, 3, 4])
```




```
plt.figure(figsize=(6, 1))
plt.eventplot(eruptions, lineoffsets=0, linelengths=0.1, color='black')
plt.gca().axes.get_yaxis().set_visible(False)
plt.ylabel('')
plt.show()
```

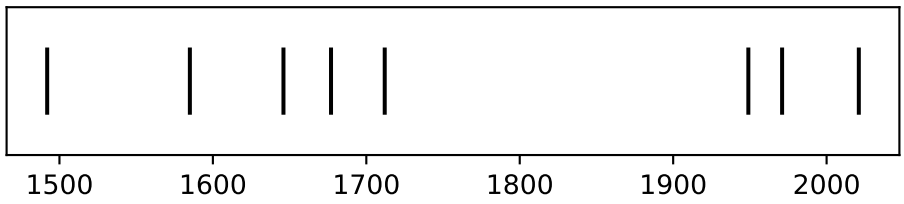


Figure 2.1: Timeline of recent earthquakes on La Palma

```
avg_years_between_eruptions = np.mean(np.diff(eruptions[:-1]))
avg_years_between_eruptions
```

```
np.float64(79.83333333333333)
```

Based on data up to and including 1971, eruptions on La Palma happen every 79.8 years on average.

Studies of the magma systems feeding the volcano, such as Marrero et al. (2019), have proposed that there are two main magma reservoirs feeding the Cumbre Vieja volcano; one in the mantle (30-40km depth) which charges and in turn feeds a shallower crustal reservoir (10-20km depth).

Eight eruptions have been recorded since the late 1400s (Figure 2.1).

Data and methods are discussed in Section 2.2.

Let x denote the number of eruptions in a year. Then, x can be modeled by a Poisson distribution

$$p(x) = \frac{e^{-\lambda}\lambda^x}{x!} \tag{2.1}$$

where λ is the rate of eruptions per year. Using Equation 2.1, the probability of an eruption in the next t years can be calculated.

Table 2.1: Recent historic eruptions on La Palma

Name	Year
Current	2021
Teneguía	1971
Nambroque	1949
El Charco	1712
Volcán San Antonio	1677

Name	Year
Volcán San Martin	1646
Tajuya near El Paso	1585
Montaña Quemada	1492

Table 2.1 summarises the eruptions recorded since the colonization of the islands by Europeans in the late 1400s.

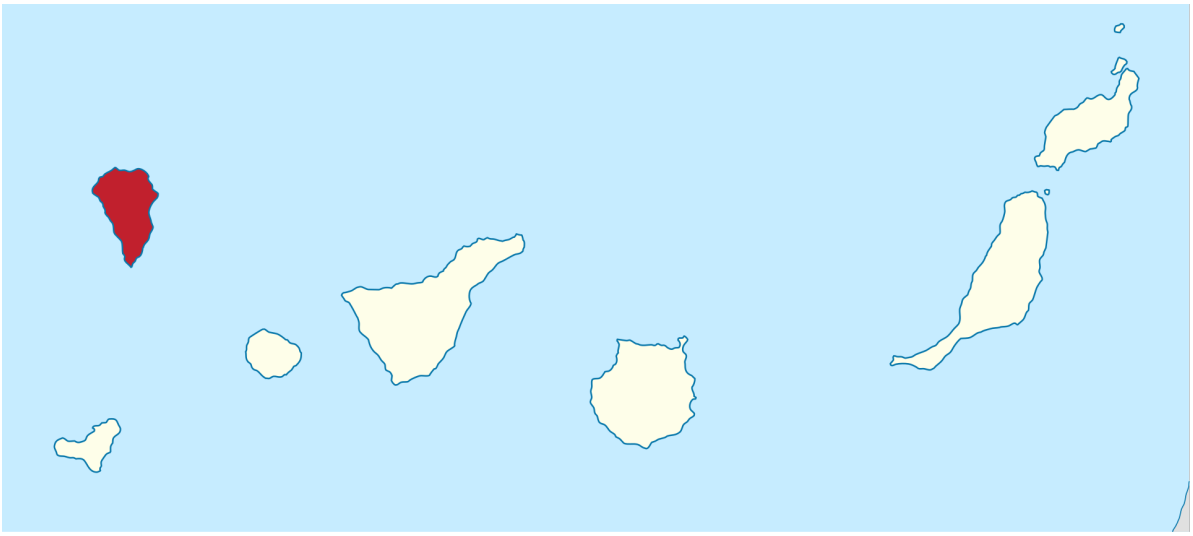


Figure 2.2: Map of La Palma

La Palma is one of the west most islands in the Volcanic Archipelago of the Canary Islands (Figure 2.2).

Figure 2.3 shows the location of recent Earthquakes on La Palma.

2.2 Data & Methods

2.3 Conclusion

References

Marrero, José, Alicia García, Manuel Berrocoso, Ángeles Llinares, Antonio Rodríguez-Losada, and R. Ortiz. 2019. "Strategies for the Development of Volcanic Hazard Maps in Monogenetic Volcanic Fields: The Example of La Palma (Canary Islands)." *Journal of Applied Volcanology* 8 (July). https://doi.org/10.1186/s13617-019-0085-5.

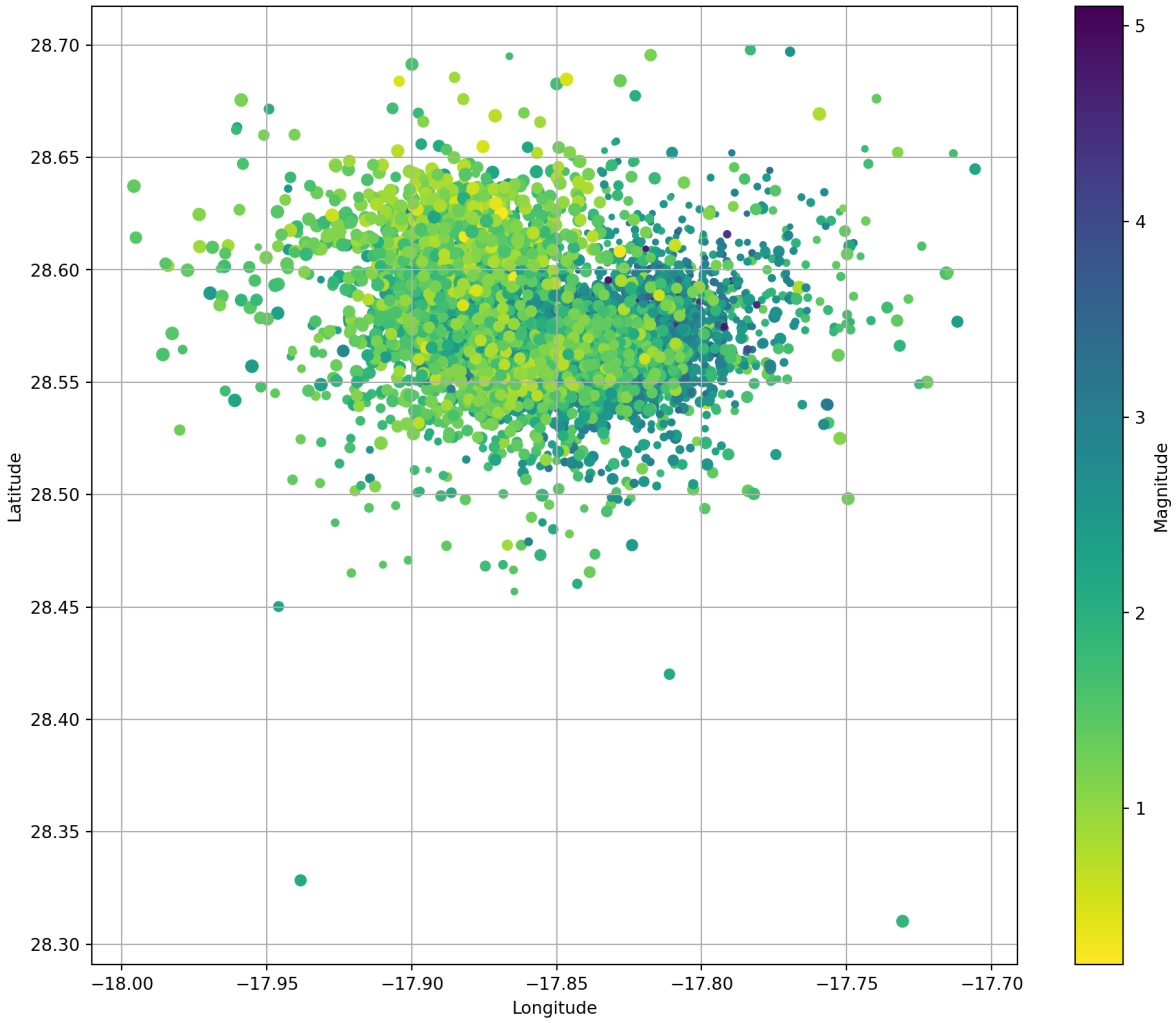


Figure 2.3: Locations of earthquakes on La Palma since 2017.

3 Python basics

This chapter introduces the fundamental concepts and basic syntax of the Python programming language. All the code covered here is part of the standard Python language and does not require any additional packages or libraries. Thus the Python discussed here forms the foundation for not only analyzing data in Python, but also for writing any kind of Python code.

While the chapter covers several different key concepts and syntax of Python, we focus on a subset of features that are most central for data analysis, rather than covering the full range of Python's capabilities. Becoming proficient in the basic Python covered in this chapter will be important as a basis for writing code in subsequent chapters, so make sure to practice and understand these concepts thoroughly.

By the end of this chapter, you should be comfortable with writing basic Python code, performing simple calculations, and understanding how Python represents and manipulates different types of data. These foundational skills will prepare you for more advanced topics in data analysis that are covered in the rest of the book.

3.1 Expressions

A **Python expression** is **any piece of code that produces a value.**. For example, the following is an expression that simply creates the number 21.

```
21
```

```
21
```

Similarly, an expression could be a series of mathematical operations that evaluate to number. For example, if want want to add 5 plus 2 and then multiple the result by 6 we can write:

```
6 * (5 + 2)
```

```
42
```

As mentioned above, the defining features of a *python expression* is that it produces a value. Expressions are one of the fundamental building blocks of data analysis and they will appear frequently throughout this book.

Exercise

What would happen if we remove the parenthesis from the expression we ran above and instead run `6 * 5 + 2`. See if you can predict what the result will be and then try it out in Python by running the code in a code cell and see if you get the result you predicted.

Solution

```
6 * 5 + 2
```

```
32
```

The result is 32, which makes sense because in the standard order of mathematical operations, multiplication occurs before addition so we multiple 6 * 5 and get 30, and then we add 2 to get 32.

3.1.1 Mathematical expressions

The expressions shown above were all "mathematical expressions" because they involve calculating numeric quantities. We can also write statements that will do operations on text and other types of data which we will describe more below. But first, let's explore mathematical expressions a bit more. Below is a table of some of the mathematical operations that are part of

Table 3.1: Python mathematical operators

Operation	Symbol	Example	Result
Addition	+	5 + 3	8
Subtraction	-	10 - 4	6
Multiplication	*	7 * 2	14
Division	/	12 / 5	2.4
Exponentiation	**	3 ** 2	9
Remainder	%	10 % 3	1

Exercise

What is the remainder from dividing 365 by 7? Please write some Python code that produces the answer.

Solution

```
365 % 7
```

```
1
```

3.2 Syntax

Syntax is the set of rules that defines how Python code **must** be written. One that think of syntax as the grammar of the Python programming language. In order for Python to be able to run your code, it **must** use the correct syntax.

To illustrate this, let's calculate the value of 8 squared (8^2) which hopefully you remember is equal to the value of 64. As shown Table 3.1, if we want to take a value `x` to the power `y` (i.e., to calculate x^y) we use the syntax `x**y`. So, if we wanted to calculate 8^2 we would write the following Python code:

```
8**2
```

```
64
```

Since we have written the correct syntax, the code runs and the result of 64 is calculated as expected.

However, if we accidentially put an extra space between the two `*` symbols, Python will not know how to interpret the expression and we will get a syntax error as shown below:

```
"hey"
```

```
'hey'
```

ERROR NUMBER 1: If python produces an error message with a ^ this will make it so that Quarto document will not be able to render to a pdf :(

For example, if I use a string without a closing quote this will produce this type of error

```
"hey
```

Likewise if I have a cell with the following it will produce this type of error:

```
"8 * * 2
```

When there is a syntax error, Python will print out `SyntaxError` and give you an indication where the syntax error has occurred using a ^ symbol.[1] As we can see here, Python is trying to show that the syntax error has occurred due to the extra space between the * symbols.

[1] The reason this is a syntax error is because Python inteprets a single `*` symbol as a multiplicaiton symbol. Thus it is trying to multiple 8 by another multiplication symbol `*`, which gives an error since one can only multiply two numbers together.