Programming with Python (/python-novice-inflammation/index.html)

The best way to learn how to program is to do something useful, so this introduction to Python is built around a common scientific task: **data analysis**.

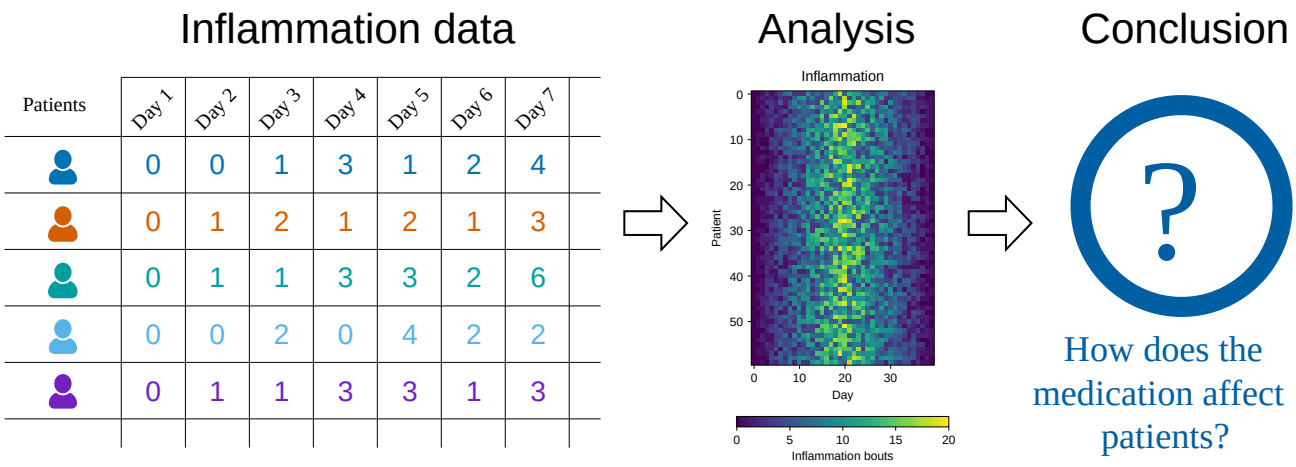
Scenario: A Miracle Arthritis Inflammation Cure

Our imaginary colleague “Dr. Maverick” has invented a new miracle drug that promises to cure arthritis inflammation flare-ups after only 3 weeks since initially taking the medication! Naturally, we wish to see the clinical trial data, and after months of asking for the data they have finally provided us with a CSV spreadsheet containing the clinical trial data.

The CSV file contains the number of inflammation flare-ups per day for the 60 patients in the initial clinical trial, with the trial lasting 40 days. Each row corresponds to a patient, and each column corresponds to a day in the trial. Once a patient has their first inflammation flare-up they take the medication and wait a few weeks for it to take effect and reduce flare-ups.

To see how effective the treatment is we would like to:

1. Calculate the average inflammation per day across all patients.
2. Plot the result to discuss and share with colleagues.



Data Format

The data sets are stored in comma-separated values (./reference.html#comma-separated-values) (CSV) format:

- each row holds information for a single patient,
- columns represent successive days.

The first three rows of our first file look like this:

Code

```
0,0,1,3,1,2,4,7,8,3,3,3,10,5,7,4,7,7,12,18,6,13,11,11,7,7,4,6,8,8,4,4,5,7,3,4,2,3,0,0
0,1,2,1,2,1,3,2,2,6,10,11,5,9,4,4,7,16,8,6,18,4,12,5,12,7,11,5,11,3,3,5,4,4,5,5,1,1,0,1
0,1,1,3,3,2,6,2,5,9,5,7,4,5,4,15,5,11,9,10,19,14,12,17,7,12,11,7,4,2,10,5,4,2,2,3,2,2,
1,1
```

Each number represents the number of inflammation bouts that a particular patient experienced on a given day.

For example, value “6” at row 3 column 7 of the data set above means that the third patient was experiencing inflammation six times on the seventh day of the clinical study.

In order to analyze this data and report to our colleagues, we’ll have to learn a little bit about programming.

Prerequisites

You need to understand the concepts of **files** and **directories** and how to start a Python interpreter before tackling this lesson. This lesson sometimes references Jupyter Notebook although you can use any Python interpreter mentioned in the Setup (/python-novice-inflammation/setup.html).

The commands in this lesson pertain to **Python 3**.

Getting Started

To get started, follow the directions on the “Setup (/python-novice-inflammation/setup.html)” page to download data and install a Python interpreter.

Schedule

	Setup (/python-novice-inflammation/setup.html)	Download files required for the lesson
00:00	1. Python Fundamentals (/python-novice-inflammation/01-intro/index.html)	What basic data types can I work with in Python? How can I create a new variable in Python? How do I use a function? Can I change the value associated with a variable after I create it?
00:30	2. Analyzing Patient Data (/python-novice-inflammation/02-numpy/index.html)	How can I process tabular data files in Python?
01:30	3. Visualizing Tabular Data (/python-novice-inflammation/03-matplotlib/index.html)	How can I visualize tabular data in Python? How can I group several plots together?
02:20	4. Storing Multiple Values in Lists (/python-novice-inflammation/04-lists/index.html)	How can I store many values together?
03:05	5. Repeating Actions with Loops (/python-novice-inflammation/05-loop/index.html)	How can I do the same operations on many different values?

03:35	6. Analyzing Data from Multiple Files (/python-novice-inflammation/06-files/index.html)	How can I do the same operations on many different files?
03:55	7. Making Choices (/python-novice-inflammation/07-cond/index.html)	How can my programs do different things based on data values?
04:25	8. Creating Functions (/python-novice-inflammation/08-func/index.html)	How can I define new functions? What's the difference between defining and calling a function? What happens when I call a function?
04:55	9. Errors and Exceptions (/python-novice-inflammation/09-errors/index.html)	How does Python report errors? How can I handle errors in Python programs?
05:25	10. Defensive Programming (/python-novice-inflammation/10-defensive/index.html)	How can I make my programs more reliable?
06:05	11. Debugging (/python-novice-inflammation/11-debugging/index.html)	How can I debug my program?
06:55	12. Command-Line Programs (/python-novice-inflammation/12-cmdline/index.html)	How can I write Python programs that will work like Unix command-line tools?
07:25	Finish	

The actual schedule may vary slightly depending on the topics and exercises chosen by the instructor.

Licensed under CC-BY 4.0 (https://creativecommons.org/licenses/by/4.0/) 2018–2022 by The Carpentries (https://carpentries.org/)
Licensed under CC-BY 4.0 (https://creativecommons.org/licenses/by/4.0/) 2016–2018 by Software Carpentry Foundation (https://software-carpentry.org)