DESTEC PhD — Python course — 10/10/2024

What we did in lesson 3

1. Exercise: focus on the syntax

   As a recap, we wrote the code for simple programs focusing on the syntax. The programs were already written, line by line, but with instructions in plain english. We took turns translating those instructions into Python syntax. The original exercise is in the Jupyter notebook called `exercise_syntax.ipynb`, and the solution we did in class in `exercise_syntax_solved.ipynb`.

2. Exercise: focus on the problem

   This exercise was supposed to centre on the thinking, disregarding the actual code writing. It didn't go very well, my fault! I'll rethink it and we'll try again.

3. Looking for code

   We talked a bit about where to look when you have issues with your code. Specifically, the official documentation of packages, Stack Overflow, AI chatbots.

4. Guidelines for the project

   See below.

Homework

No homework this time. Please keep exercising on writing simple code (taking problems from the github page linked in the last readings)[1], and go over the documentation for the numpy library. For those of you who already have ideas for a project, start to study the format of your data, and to explore the libraries you think you'll need.

Guidelines for personal projects

Don't worry if you still haven't an idea for a project, or if you have one but don't know where to start. I'm putting a couple of suggestions and requests here, for the future. Some (many) may be a bit trivial.

- Focus on a clear objective

  Think about what you want to achieve in practical terms. Thinking: "I want to solve this problem" is not going to help you too much. It's better to picture exactly what

[1] Somebody pointed out that the solutions linked in that page are still using Python 2 syntax. I didn't notice, sorry about that. Still, the problem formulations are useful, the logic behind the solutions can be understood, and if you are in doubt you can always ask me or an AI chatbot to translate any code into Python 3 syntax.

kind of data you want to obtain from the program. For example: a table with exactly *this* data, with *this* number of columns and rows; or *this* plot, with this data on x and this data on y.

- Think about what data you have
  The kind of data you want to start with is going to influence the choice of tools. Is it a table of numbers only? Maybe `numpy` is the right choice. A more complex set of numerical and categorical/qualitative data? Then maybe `pandas` (which we still haven't seen) is going to be better.

- Small objectives and small steps
  Some of you already proposed very ambitious projects. This is great! But maybe it's better to start small. Think about the practical steps you'll need to transform your original data into the result, and start to work on the easiest ones. For most of you, this will probably mean learning how to read data (sometimes from many files all at once) and how to work with tables. Remember: projects are just a way to force you to exercise, while also minimising useless efforts. I don't care if your projects seem super simple: especially if you never programmed before, they will still be great achievements!

- Comment your code (and document functions)
  Please please please.

- Write your code as a series of functions
  It's easier to test code as a simple series of instructions, especially on Jupyter notebooks: you can see right away if you got the intended result. But when you build your program, try to encapsulate those instructions into functions, and then to call those functions one by one. This has a practical benefit, because it will allow you to change the inner workings of any function without having to change all the remaining code (remember, variable names inside a function *stay* inside a function). I hope that it will also help you to organise your thoughts, because it will force you to think exactly at what inputs and what outputs you will need and obtain.

- If you want, send me your code
  I'll be more than happy to review your code. Any code, even exercises. Remember to send me either Python scripts (*.py) or Jupyter notebooks (*.ipynb), and to also attach example datasets if you are using them.