What we did in lesson 5

Suggestions to make good graphics (MANGO!)

1. Meaningful and sincere
   Your images should convey a specific message without hiding information or tricking the reader. So choose the type of plot carefully and be honest when presenting less-than-optimal data.

   → *choices*: type of plot, selection of data to show

2. Accessible
   Not everyone shares the same visual spectrum, but you want everybody to be able to see your graphics. If colors convey meaning in your plots, choose them so that they can be distinguished in grayscale.

   → *choices*: color palettes, line and marker styles

3. No wasted ink
   Clutter makes graphics less readable and obscures your message. Make plots that you like, but avoid useless decorations as much as possible. If it doesn't convey meaning, it can usually be discarded.

   → *choices:* what to keep in your plot, legend or text on plot

4. Good resolution or vector
   Most computer-generated plots are born vectors and should remain that way until publication. If you need a raster image, please ensure that its resolution is high enough. Most journals require at least 300 dpi for raster images and 500 dpi if they contain text or lines.

   → *choices*: image file format and resolution

5. Only if needed
   Do you really need a plot? Wouldn't a table be better?

Exercise: make these plots better

We looked at existing plots (both from scientific papers and not), criticised them and proposed ways to make them better. They are in the folder called `notsogood_plots`.

Plotting with Python: the Matplotlib library

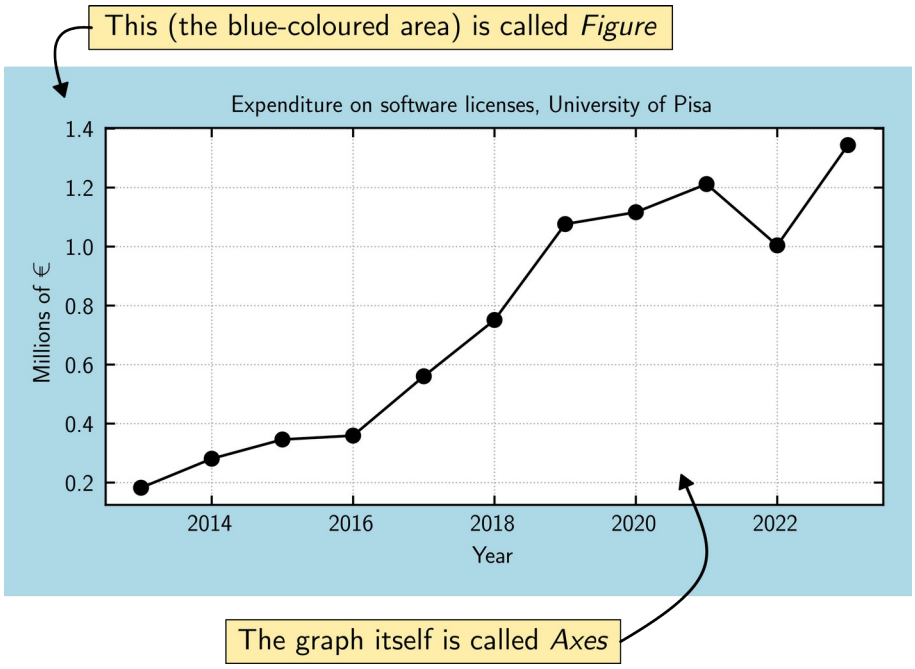
As you can tell by the graph here, there are *a lot* of plotting libraries for Python. Maybe too many. The most important distinction is between libraries that are especially good for *interactive* plots and those that are especially good for *static* plots. If you are exploring your data or making stuff for online viewing, you may want one of the first kind. A good library for interactive plots is Plotly.

But we are interested in making plots for publication, which for now usually means static images. For this purpose, the mother of all Python plotting libraries is Matplotlib. Matplotlib is commonly the one working behind the scenes even when you are using other plotting libraries.

Using Matplotlib can be a bit unnerving at first, because you have to get into a very specific mindset and accept some new naming conventions. But it allows you to customise your plots to the finest detail, and to "easily" make them consistent.

Instead of trying to learn *all* the syntax of the library, as always, we focused on understanding a couple of simple (but hopefully useful) examples. Anything more complicated stems from here.

The basic important concept is this:



A more complete version of this figure with the names of other objects is in the official documentation for Matplotlib.

Everything else we talked about should be in the Jupyter notebook called `lesson5_matplotlib.ipynb`.

After the plot: using Inkscape to finetune vector graphics

We didn't have time to do it during the lesson, but you may be interested in a free and open-source software that manages vector graphics called Inkscape. It's best to do most of the graphic customization directly in the code, so that you can easily replot if the data has changed. But sometimes you may get frustrated trying to change some minor detail in Python without succeding. In this case, it's best to save your plot in a vector format (`.svg` is the best in this case), and tweak it in Inkscape.

Some useful stuff

Matplotlib

- The official quick start guide to Matplotlib.
- A gallery of basic plot types you can do with Matplotlib.
- The Matplotlib cheatsheets, that you should also have received with this file.
- A list of named colors, one of colormaps, one of line styles and one of markers.

Accessible graphics

- Blogs with nice-looking and accessible palettes, by Paul Tol and David Nichols.
- Colorbrewer, an interactive online tool to choose an accessible color palette.
- Coblis, a color blindness simulator for your pictures, all online.
- Color Oracle, a free program that you can install on your PC to simulate a specific kind of color blindness on the whole screen.

General plotting and quantitative graphics

- Data-to-viz, a website that tries to help you find the right plot for your data, and also has specific examples with Matplotlib code.
- A useful and beautiful book on graphics called The Visual Display of Quantitative Information, by Edward R. Tufte (it's also very expensive, unfortunately, but you can get it from the University library).