

Data Visualization in Python

Before jumping into making your own plots, it is imperative to understand that how you present your data can greatly influence what conclusions viewers draw from it. Thus, it is incredibly important that you do not present your data in a way that is misleading. To avoid such pitfalls, you need to be aware of some common ways in which graphs can be misleading as well as some good practices.

- **!! Misleading graphs**

Familiarizing yourself with the tools listed below (I highly recommend going in the order presented) will enable you to generate quality graphics to visualize your data.

1. [matplotlib](#): The most used plotting package in Python. Bonus: It's almost identical to plotting in MATLAB. This is probably the most customizable plotting package out there, so you can get your plots to look exactly how you want them. The downside is that the default plots don't look all that nice, and customizing them requires a lot of code 😞
2. [seaborn](#): Shortcuts for creating nice looking matplotlib plots from pandas dataframes (i.e., tables).
3. [hvplot](#): Easily create nice looking interactive plots from pandas dataframes (i.e., tables).
4. [napari](#): Awesome drag-n-drop app for N-dimensional image viewing and analysis. *I recommend creating a new environment in which to install this as it has a lot of dependencies.*

Other resources of interest:

- [plotly](#): Interactive plots. Note that hvplot can be configured to use plotly if you want.
- [bokeh](#): Interactive plots. Note that hvplot uses bokeh by default.
- [panel](#): Create dashboard apps in your browser with interactive graphs. I don't have any experience with this, but it looks like it could be a nice option.
- [vispy](#): OpenGL accelerated plotting. This is pretty awesome, but currently requires a lot of low level coding. Thus, it is probably not very appropriate unless you are comfortable digging through a large code base with sparse documentation.
- [PySide/PyQt](#): Create graphical user interface (GUI) apps. I personally love Qt, but be warned that this may require a large investment to become reasonably familiar with.