Course
CS-E4002
Course materials
Your points
Code Vault

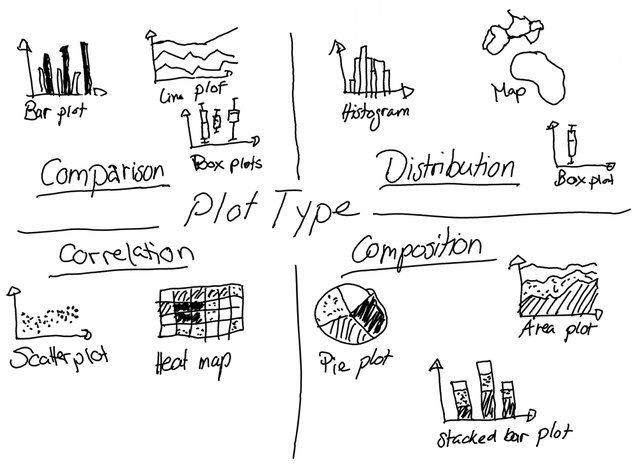
This course has already ended.

« More Matplotlib Functions Course materials What NOT to plot »
CS-E4002 / Writing and Plotting / What to plot

What to plot

In the preceding materials, we saw how to generate different plots. Now, we want to take a look at an even more important part of a visualization: How you should appropriately visualize your findings.

With visualizations we want to tell a story - think carefully what is used best to tell your story. Here is a short breakdown of different plot styles, to help you choose the right plot style for your data. You can find a nice description here: https://www.visual-design.net/post/choose-the-right-chart



Comparison

Very often we want to compare two or more series of data. If we have numerical values in the same range, a **line plot** (the traditional plot) with multiple lines is very convenient. Make sure you scale the x-axis and the y-axis so that the values are easily comparable. If you do not use a linear scale, make sure to mention it in the plot title.

If the x-axis in our data is not continuous, a **bar plot** can give us a better comparison. If we want to compare several distributions, **box plots** (with several boxes) are the way to go.

Distribution

Distribution plots are a good tool to get a rough overview of your data. A standard way of summarizing distributions is to use a **box plot**.

If you want to get a more detailed impression of the distribution of your data than a box plot, a **histogram** is the right plot style for you.

If your data has a geographical aspect, a **map** is the prefered plot style.

Correlation

In correlation plots you can see if one or more pairs of variables are related. Careful: Correlation is not equal to causation!

If you are interested in the correlation between two variables a **scatter plot** serves your needs very well. Scatter plots are often combined with a trendline, which can be used to represent a regression relationship between the variables.

A **heatmap** shows pairwise values (e.g. value counts or correlation coefficients) with different color intensities reflecting the values' magnitudes.

Composition

The plots in this section provide insights about subdivisions and proportional quantities, i.e. the comparison of certain parts compared to the whole part.

Pie charts are very famous examples of that. They give a nice visual indication of how much one part is compared to the whole pie. Without additional information, this relation is all the plot shows.

If you are interested both the absolute values as well as their subdivisions, a **stacked bar plot** is the right plot for you. These are the same as bar plots, but individual bars are subdivided to show their component parts.

If the values are continuous, a **stacked area chart** gives the best visual. This is essentially a line plot, where the area under the line is subdivided to show component amounts.

What to take into acount

- Use appropriate colours

  > Choose colours that fit the findings and try to use not aggressive colours to not hurt the eyes. Also, keep in mind that roughly 10% of the population is colorblind: choose colors that can be easily distinguished - including on a black-and-white printout. You should also make sure to follow basic conventions (green - good/positive, red - bad/negative, etc.). To represent different values on a continuous scale (and in the same data series), it is best to use different intensities/shades of the same color. To represent different values in a categorical group (or across different data series), it is best to use different colors altogether.

- Use 0/0 as the point of origin, unless there are very good reasons not to do so – even then, make sure to state it clearly!
- If you have discrete values which are not dependent on each other - do not display them in a line plot. Use a bar plot instead.
- If you are a user of LaTeX (look it up - very convenient): look up the Tikzplotlib

  > For LaTeX enthusiasts, the library tikzplotlib allows you to directly save your plots as tex files. Once the tex file is created, values, legends, colours and all the other parameters can be changed directly in the file without having to generate the plot again from Python code. This can be very useful when producing curves for a report, thesis or article. The LaTeX package pgfplots is necessary to include such a tikz plot in a tex file (the link given is not from the original documentation but more readable).