Introduction to Data Visualization

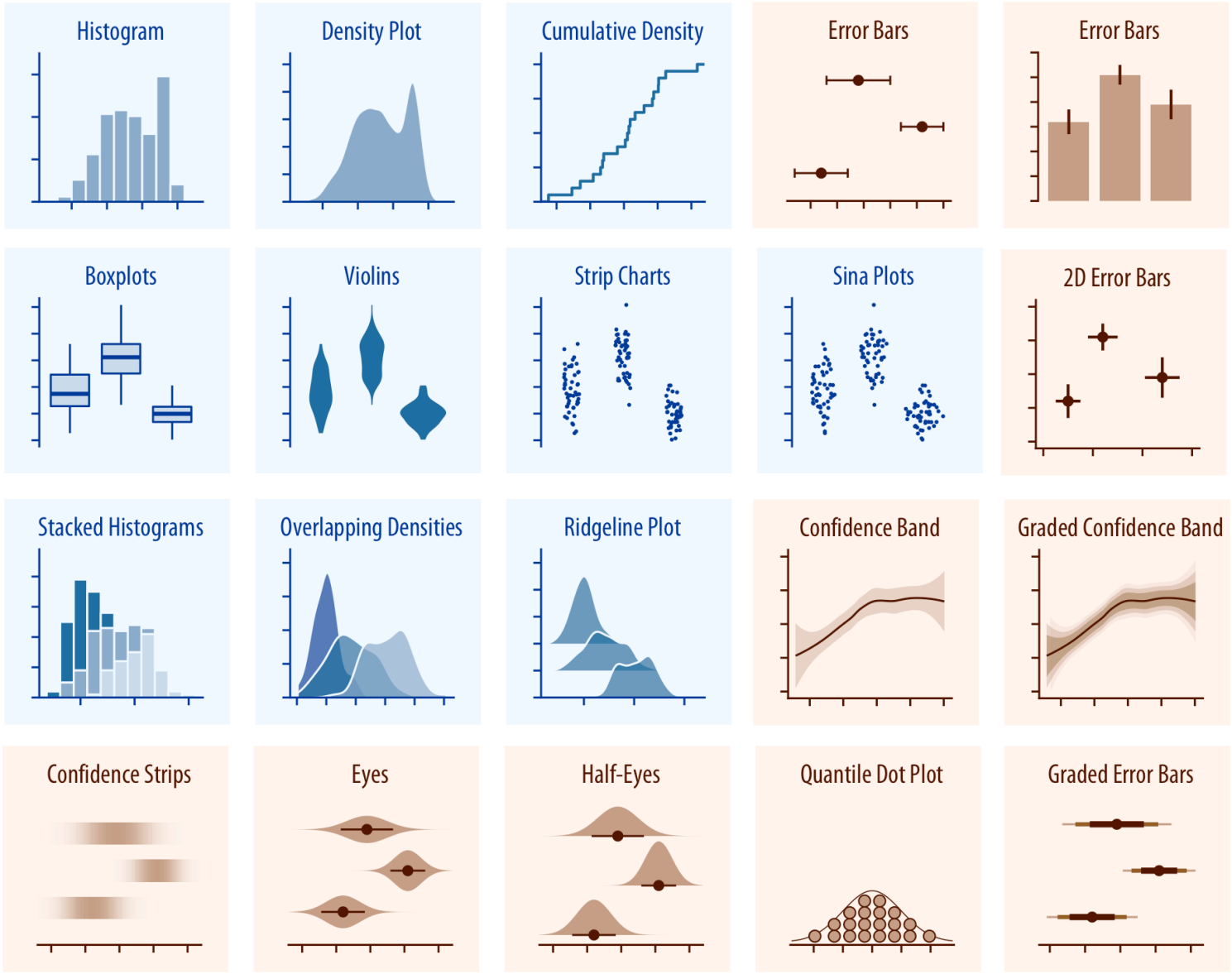
QB Bootcamp, Day 3
Friday, 11 September 2020
2:00pm - 2:30pm

There are many ways to visualize data

visualizing amounts or proportions

There are many ways to visualize data

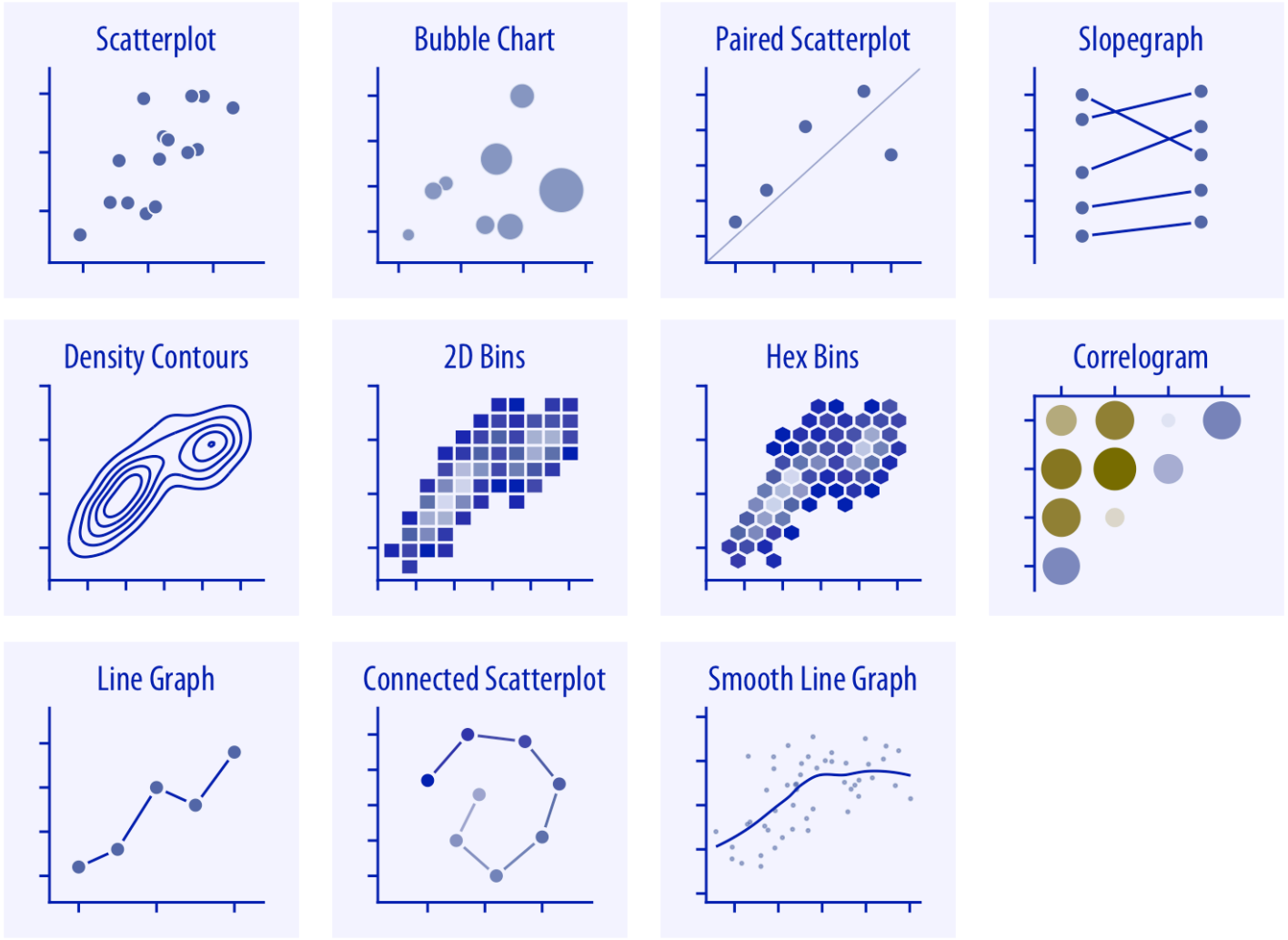
visualizing distributions or uncertainties



Wilke, 2019

There are many ways to visualize data

visualizing pairwise relationships



Wilke, 2019

Data visualization resources

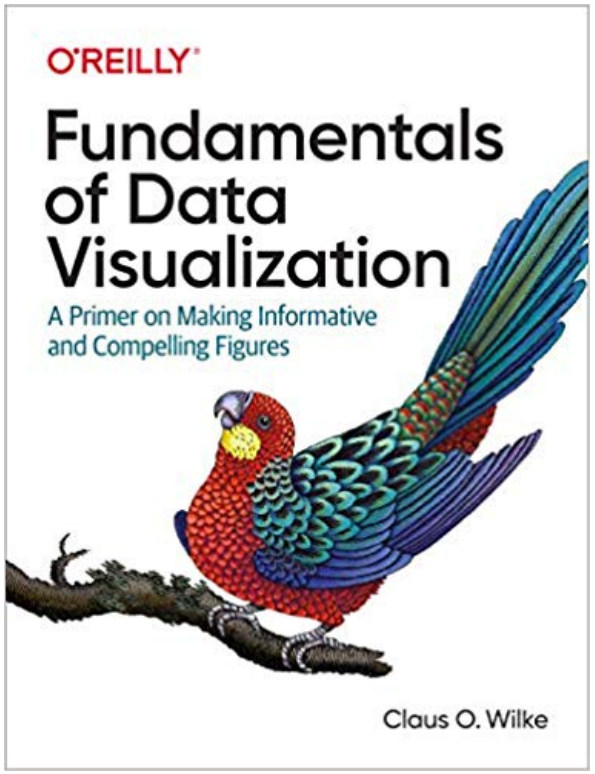
Claus Wilke gives excellent guidance on data visualization do's and don'ts

Molly Hammell will discuss data visualization later on in SEE.

I will briefly discuss 2 issues:

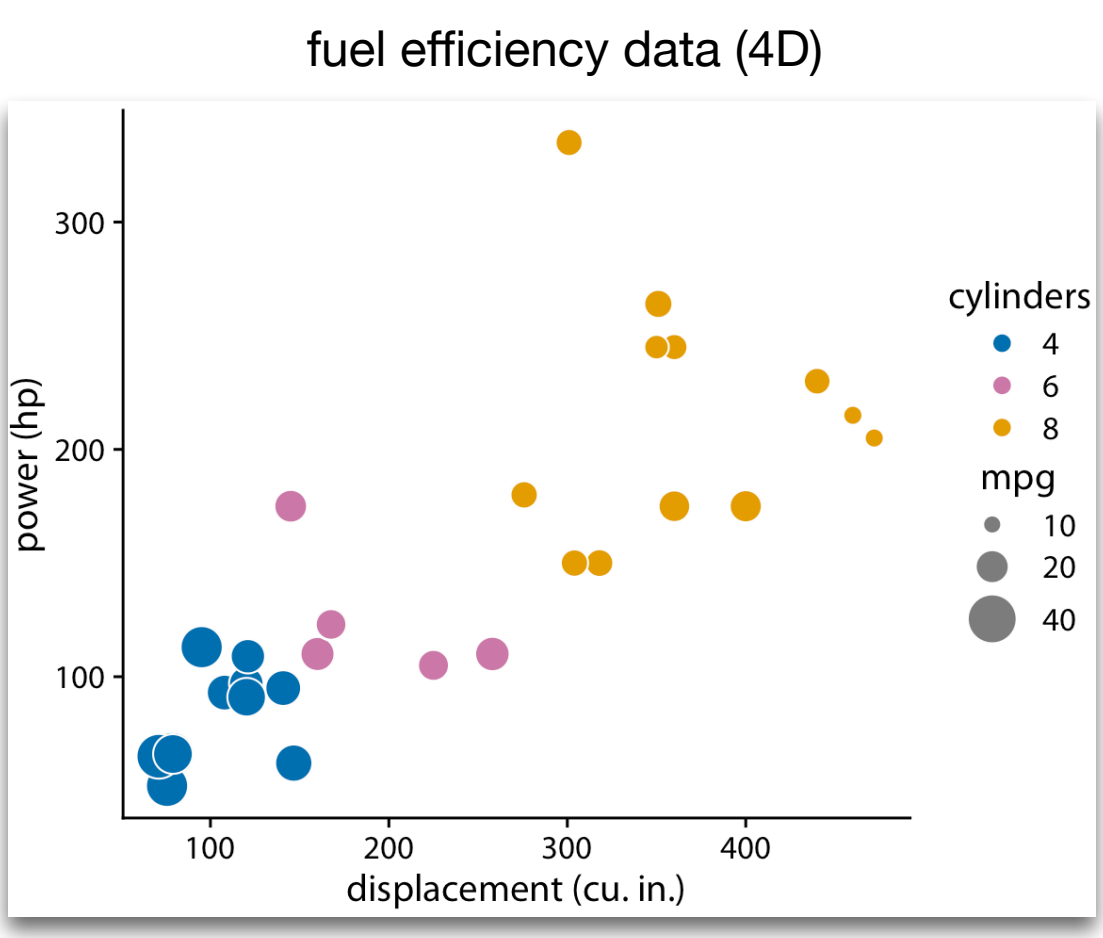
- **Aesthetics**
- **3D plots**

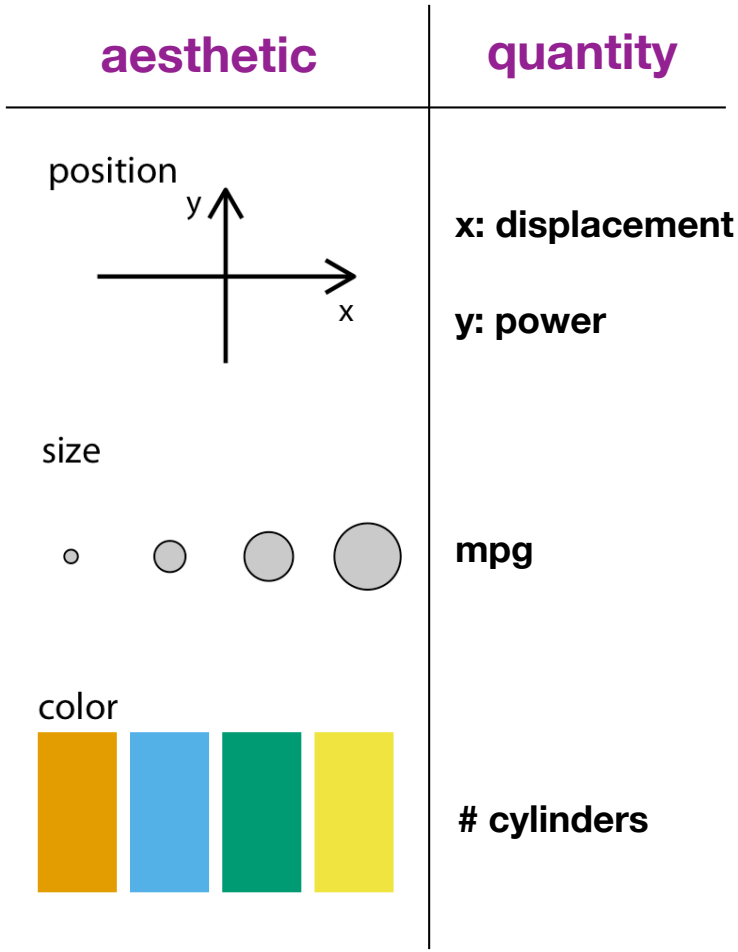
We will then dive into Matplotlib and Seaborn



Fundamentals of Data Visualization
Wilke, 2019

All data visualization uses “aesthetics” to encode quantitative information

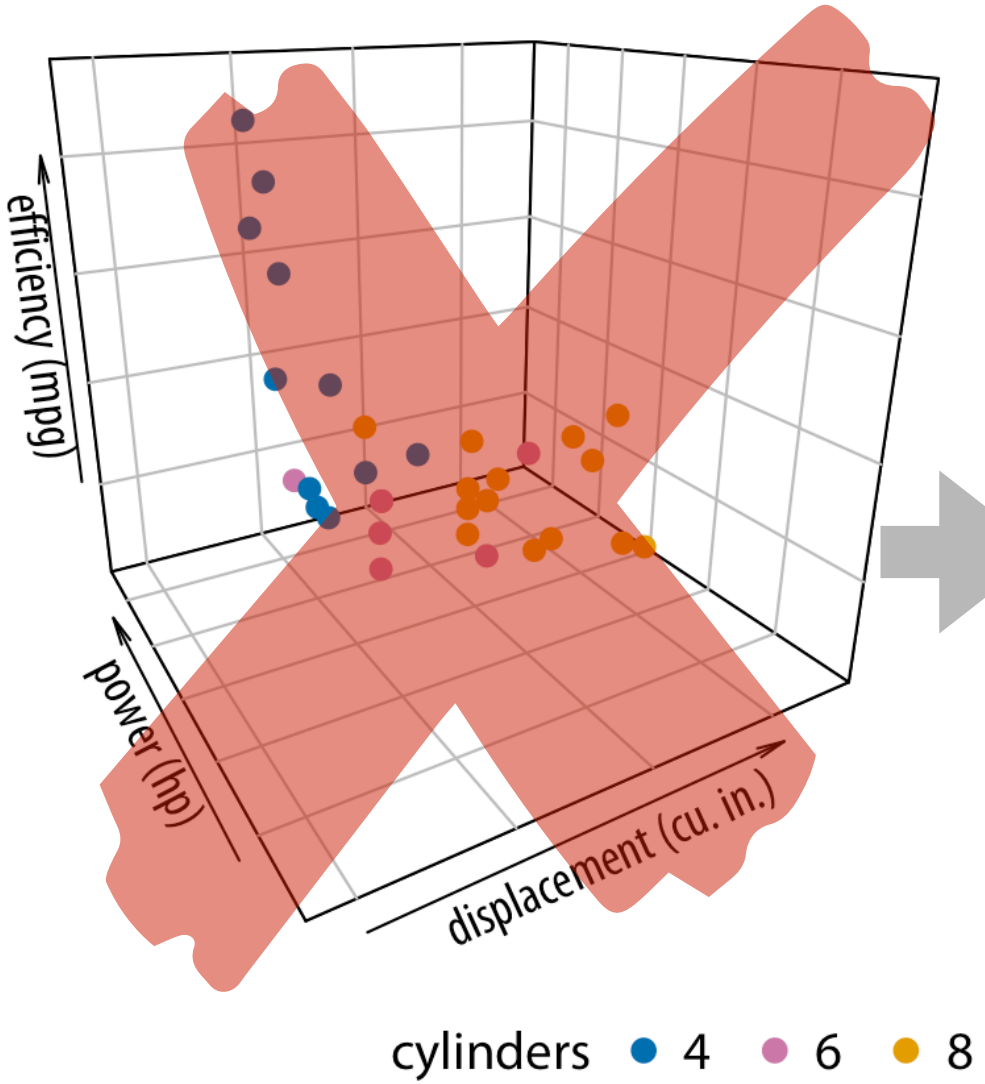




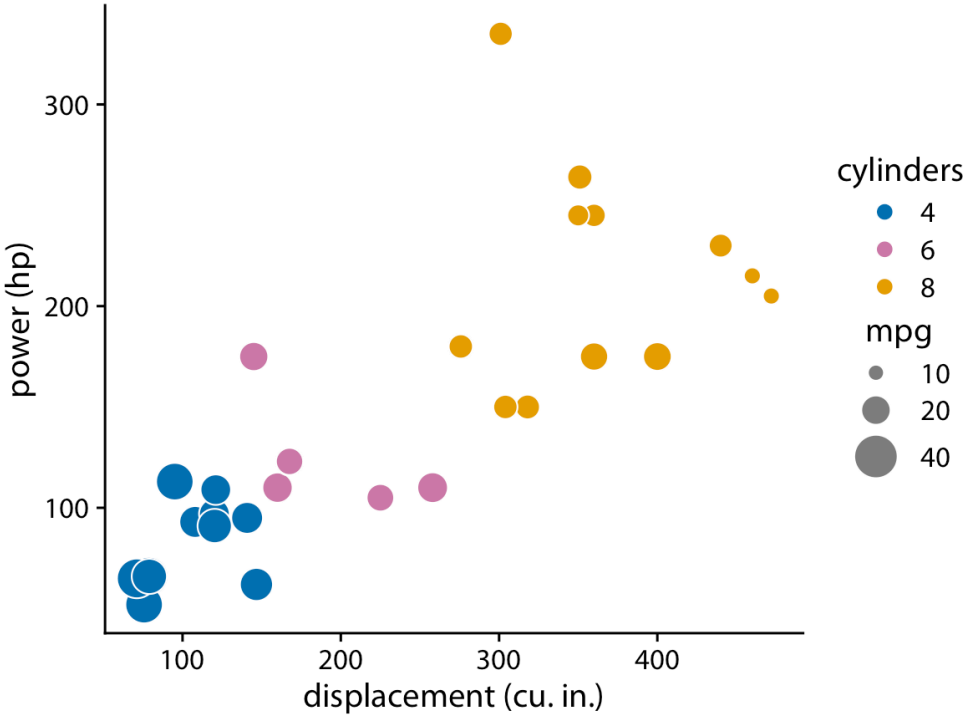
Wilke, 2019

Please don't plot data in 3D

In a 3D figure, it's virtually impossible to figure out the coordinates of each point by eye.

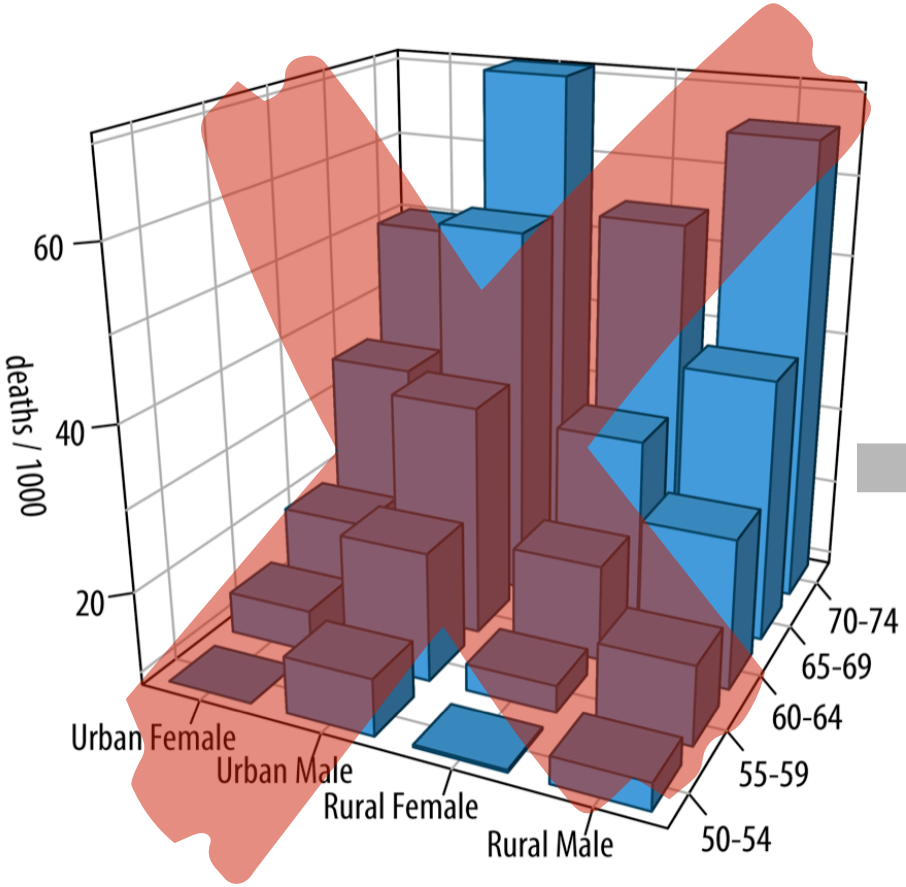


Plot in 2D using multiple aesthetics instead

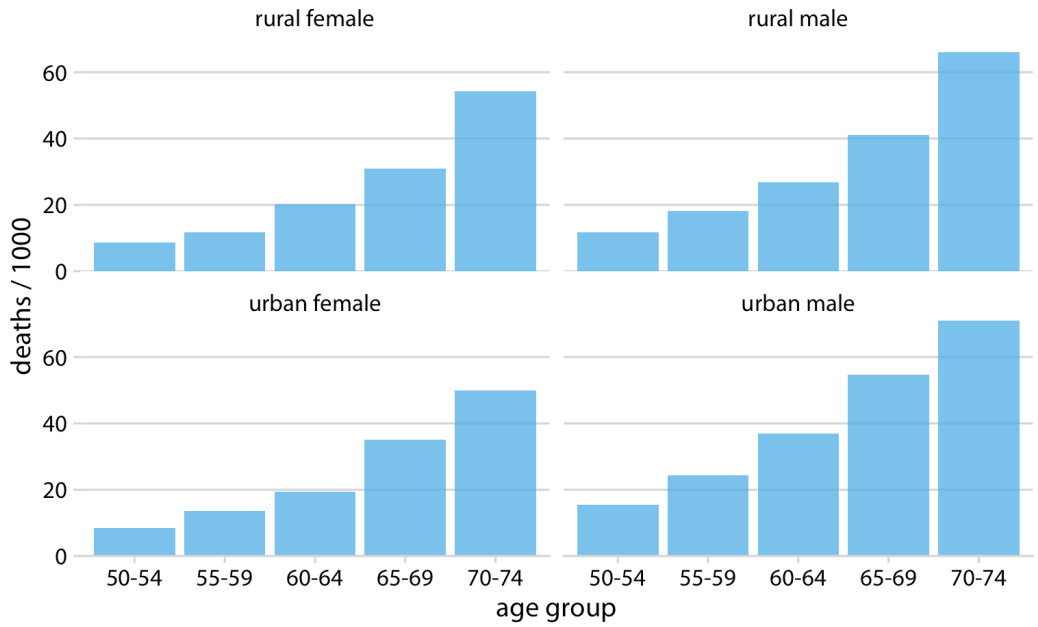


Wilke, 2019

Please don’t plot data in 3D



Alternatively, use “small multiples” of 2D plots

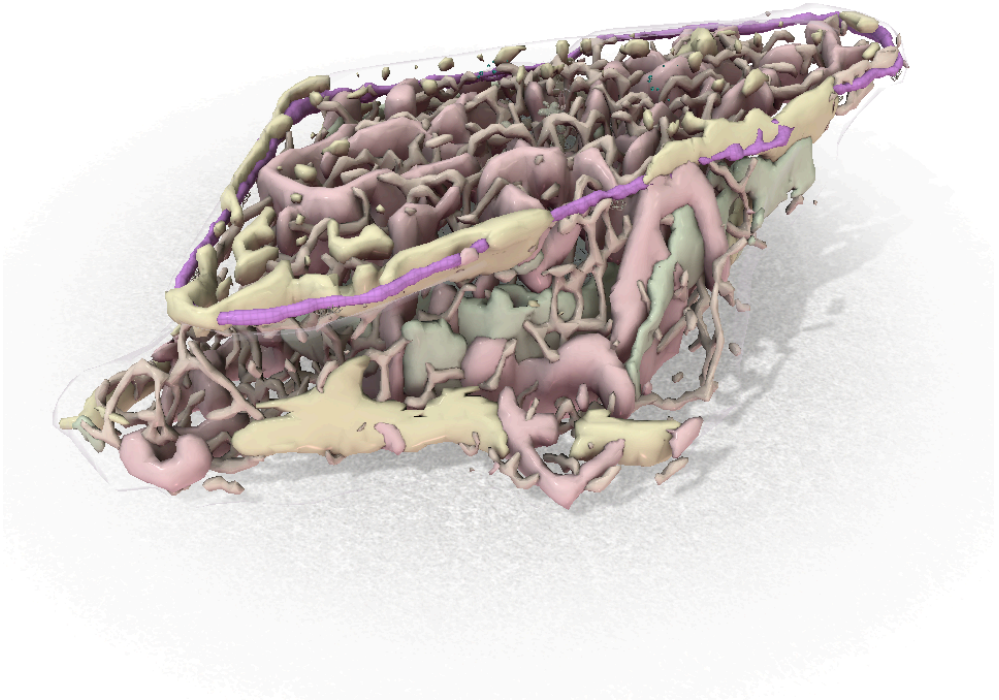


Wilke, 2019

Only use 3D if you're illustrating some inherently 3D object

Structure of GFP
(PDB:1EMA)

cellular structures
(Allen Cell Explorer)



https://www.rcsb.org/structure/1ema

https://www.allencell.org/visual-guide-to-human-cells.html

Matplotlib is the foundation for most graphics in Python

Matplotlib provides basic graphing infrastructure.
It is functional, but the capabilities are rather basic and it takes effort to make professional-looking plots.

matplotlib
Version 3.1.1

Fork me on GitHub

home | examples | tutorials | API | contents » modules | index

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.

Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the `pyplot` module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Installation

Visit the Matplotlib installation instructions.

Documentation

This is the documentation for Matplotlib version 3.1.1.

To get started, read the User's Guide.

Other versions are available:

- 3.1.1 Stable version.
- 2.2.4 LTS LTS version.
- 3.2.x Latest git master (unstable)
- 3.0.3 Older stable version.

Trying to learn how to do a particular kind of plot? Check out the examples gallery or the list of plotting commands.

Other learning resources

There are many external learning resources available including printed material, videos and tutorials.

Quick search
Go

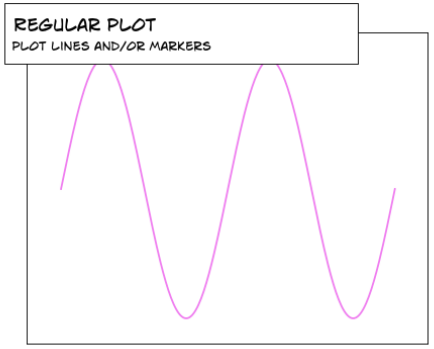
Matplotlib 3.0 is Python 3 only.

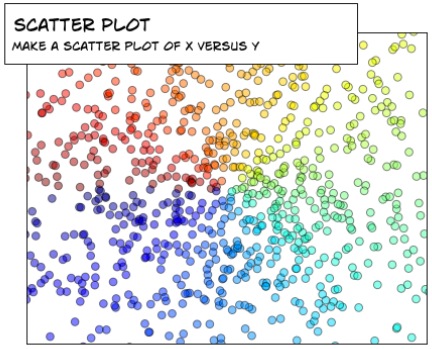
For Python 2 support, Matplotlib 2.2.x will be continued as a LTS release and updated with bugfixes until January 1, 2020.

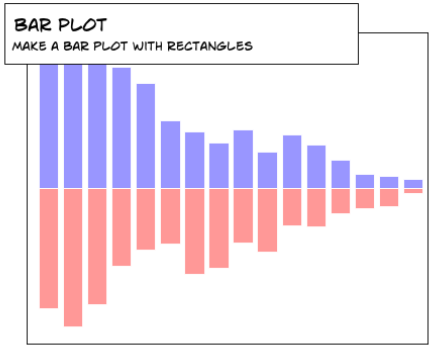
Support Matplotlib

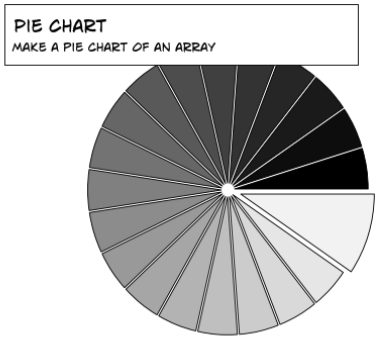
matplotlib.org

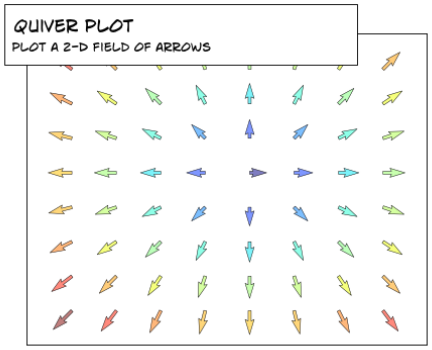
There are many, many types of plots that one can make in Matplotlib

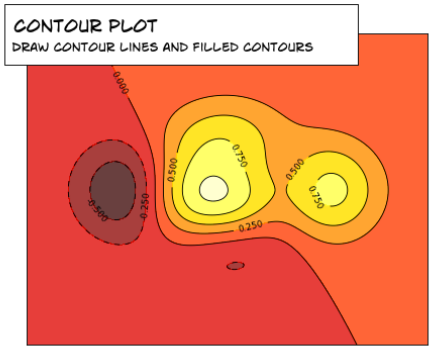


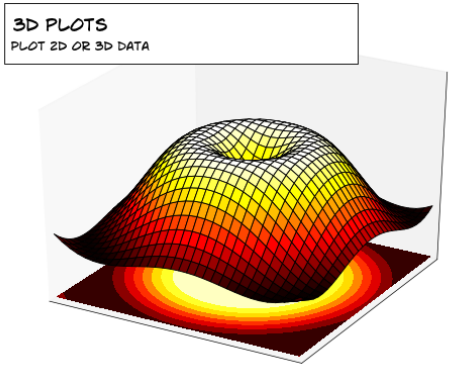


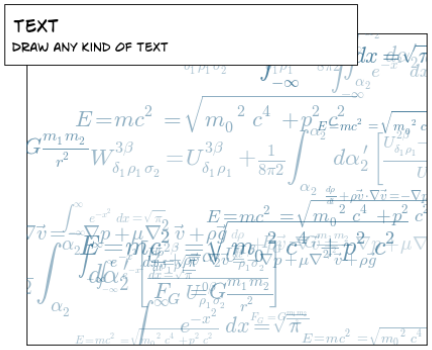


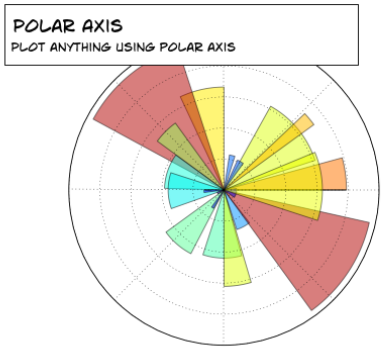








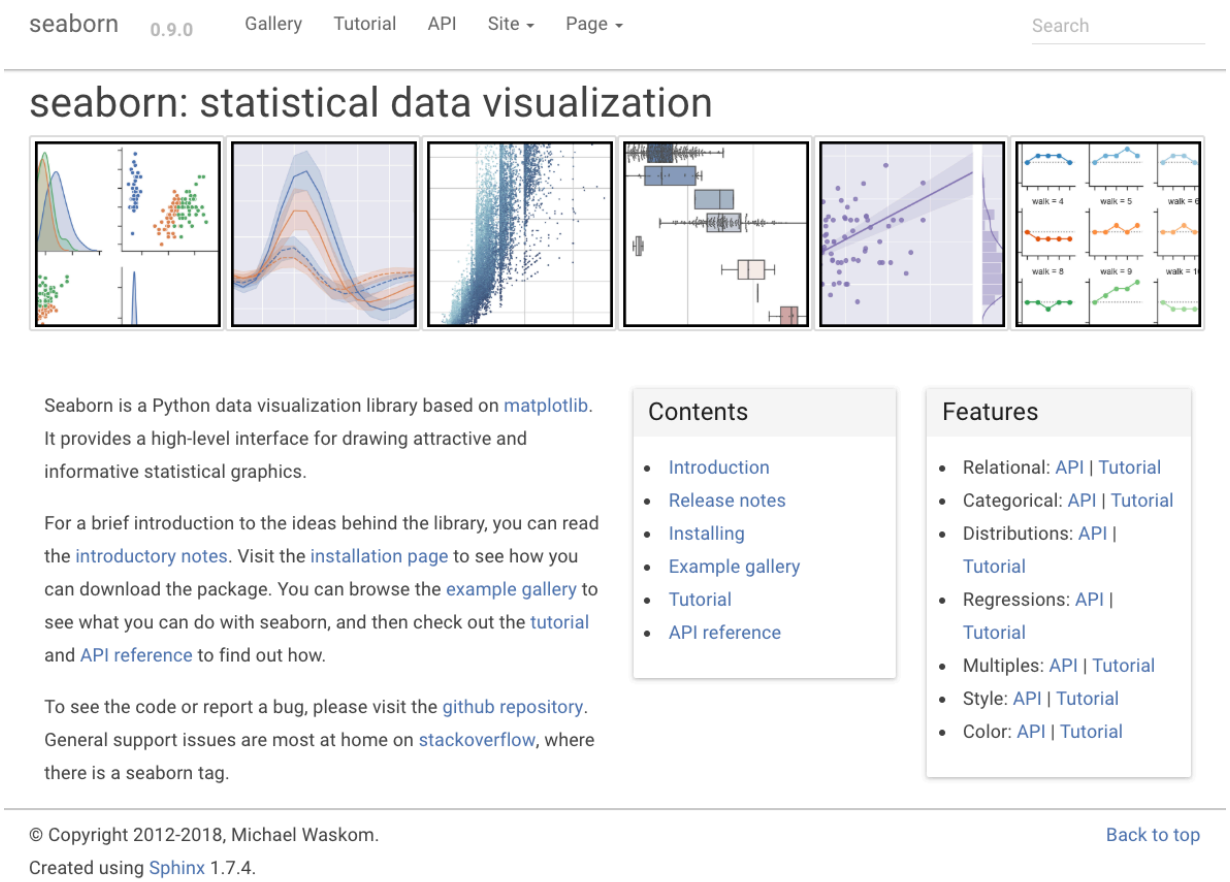




https://github.com/rougier/matplotlib-tutorial

Seaborn facilitates rapid data visualization

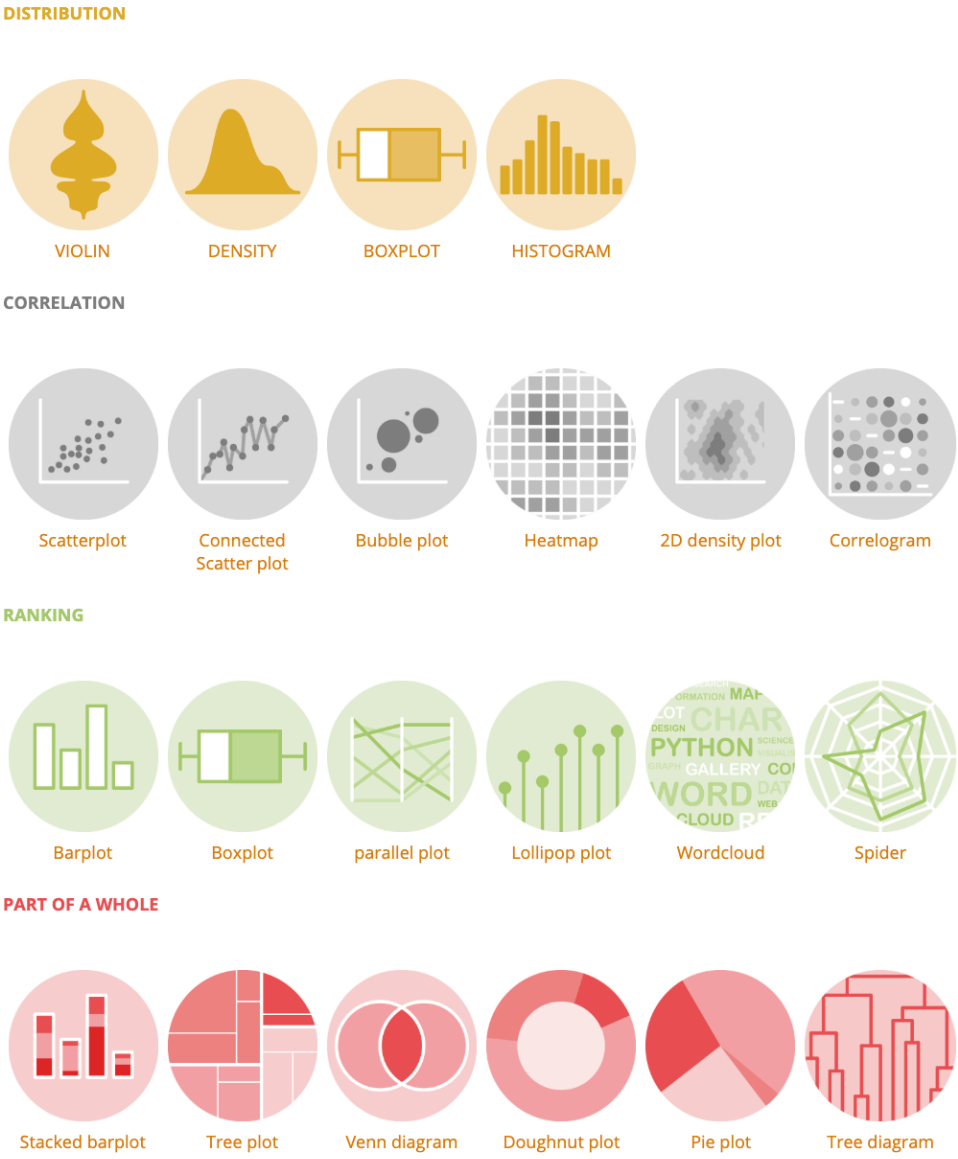
Seaborn is a *wrapper* for Matplotlib.
It provides methods for rapidly generating a wide range of decent-looking plots from data stored in Pandas dataframes.



seaborn.pydata.org

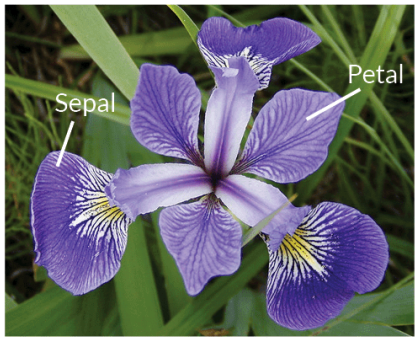
The Python Graph Gallery provides detailed plotting instructions for beginners

THE PYTHON
GRAPH GALLERY



python-graph-gallery.com

Fisher's "Iris" dataset is a famous example dataset in statistics and dataviz



Iris Versicolor

Iris Setosa

Iris Virginica

R. A. Fisher (1936).
"The use of multiple measurements in taxonomic problems".
Annals of Eugenics. **7** (2): 179–188.
(data collected by Edgar Anderson)

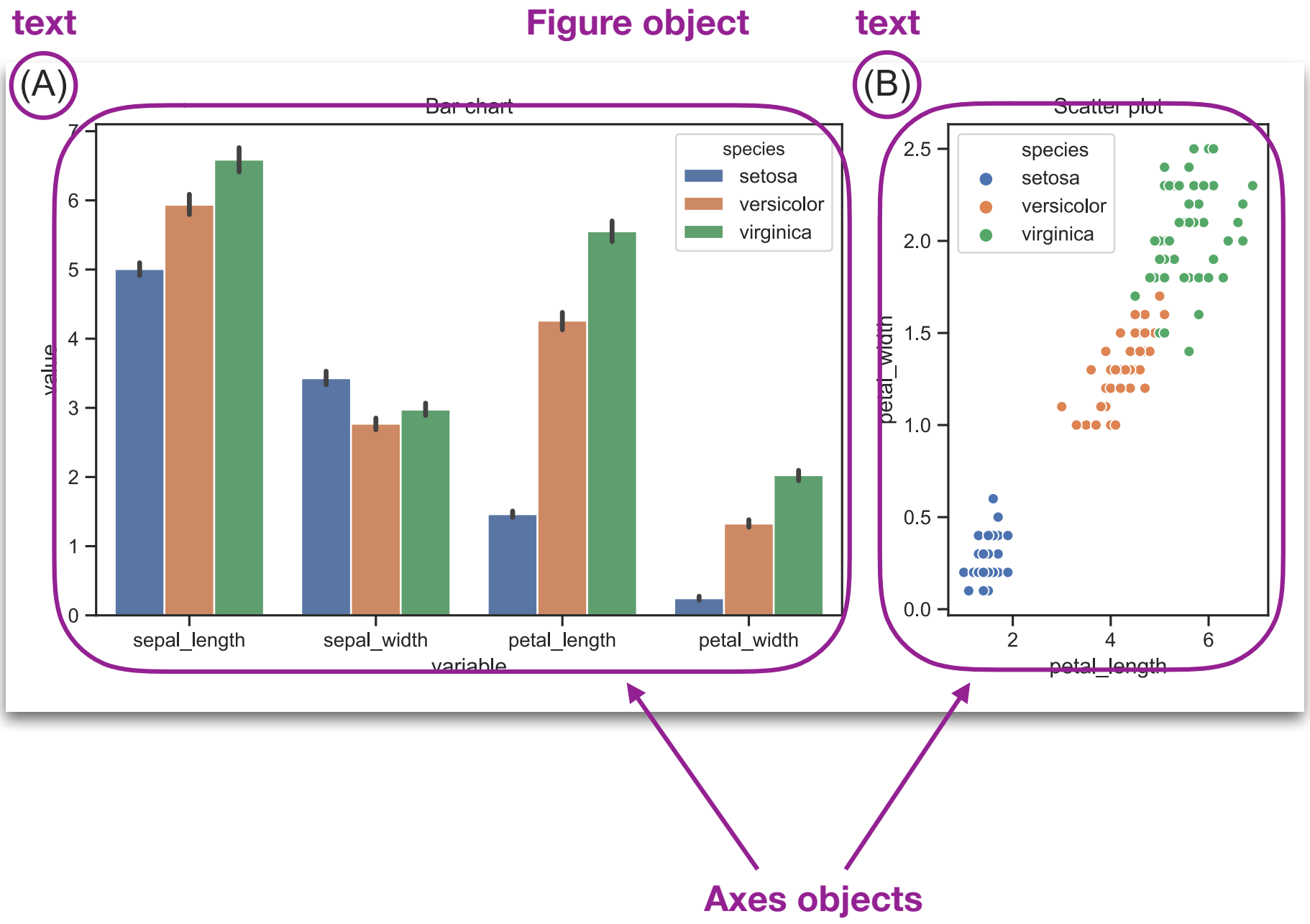
150 rows
(50 per species)

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

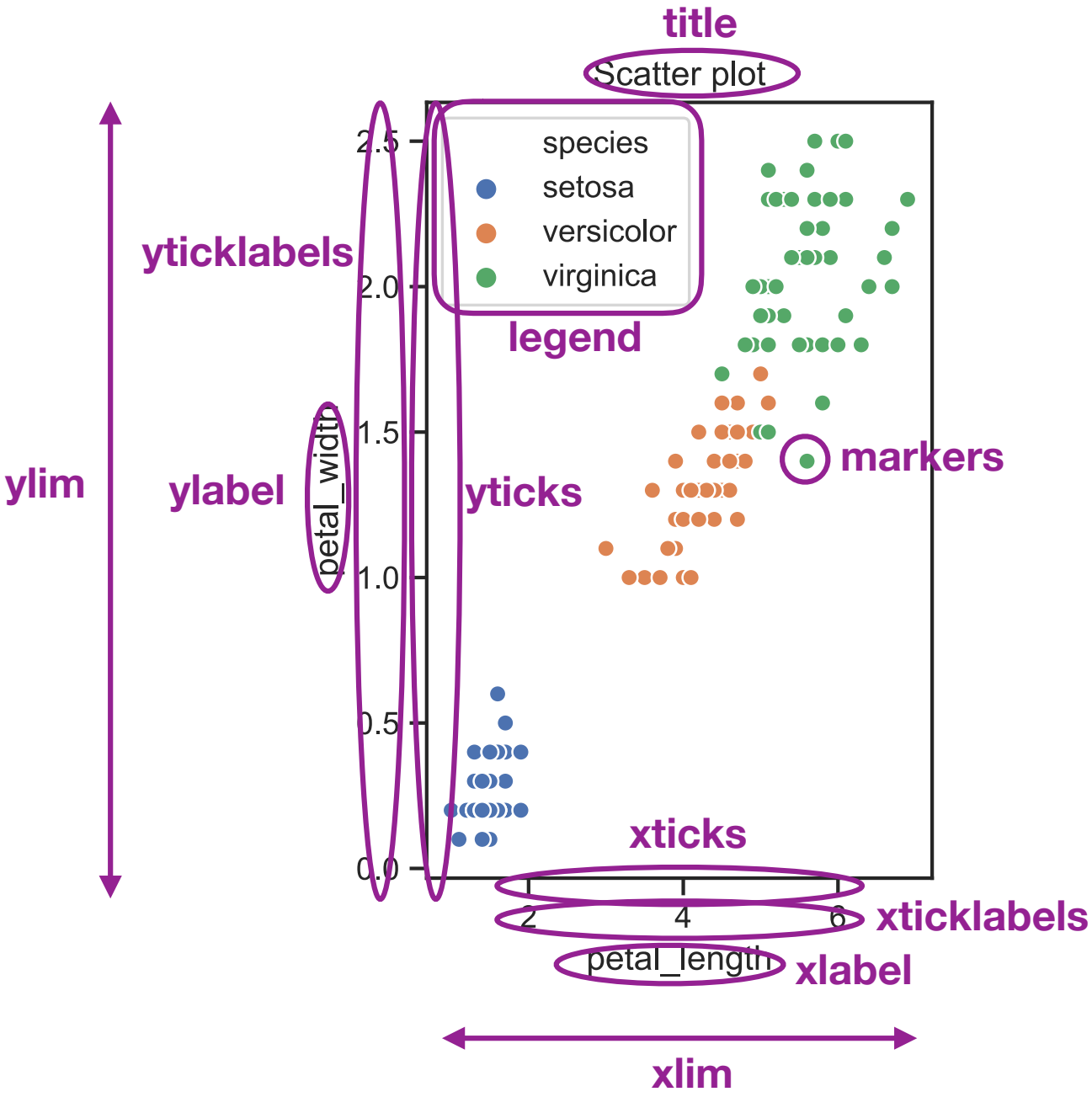
⋮

https://en.wikipedia.org/wiki/Iris_flower_data_set
https://www.datacamp.com/community/tutorials/machine-learning-in-r

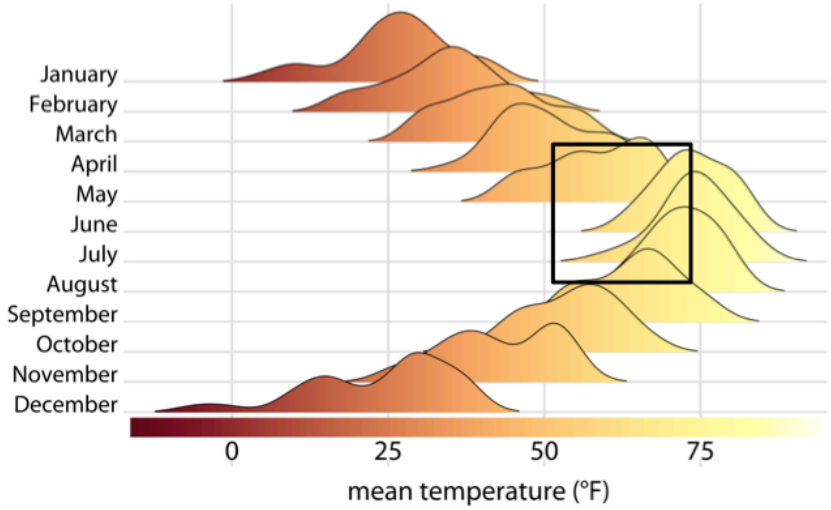
Matplotlib figures contain one or more Axes objets



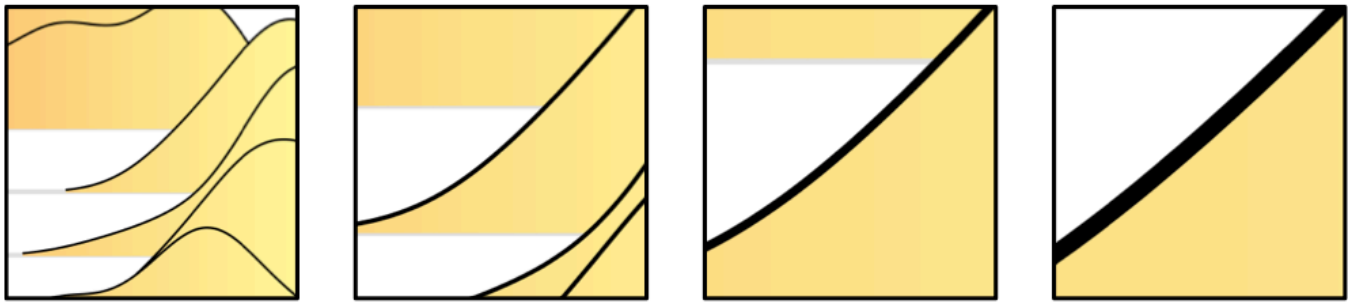
Each Axes object contains many individual elements that can be adjusted



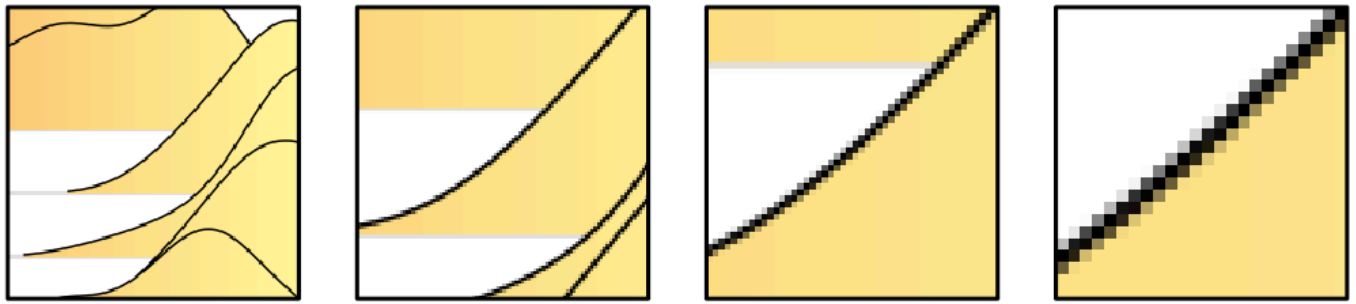
Keep in mind the difference between vector graphics and bitmaps



vector format
(PDF, SVG)

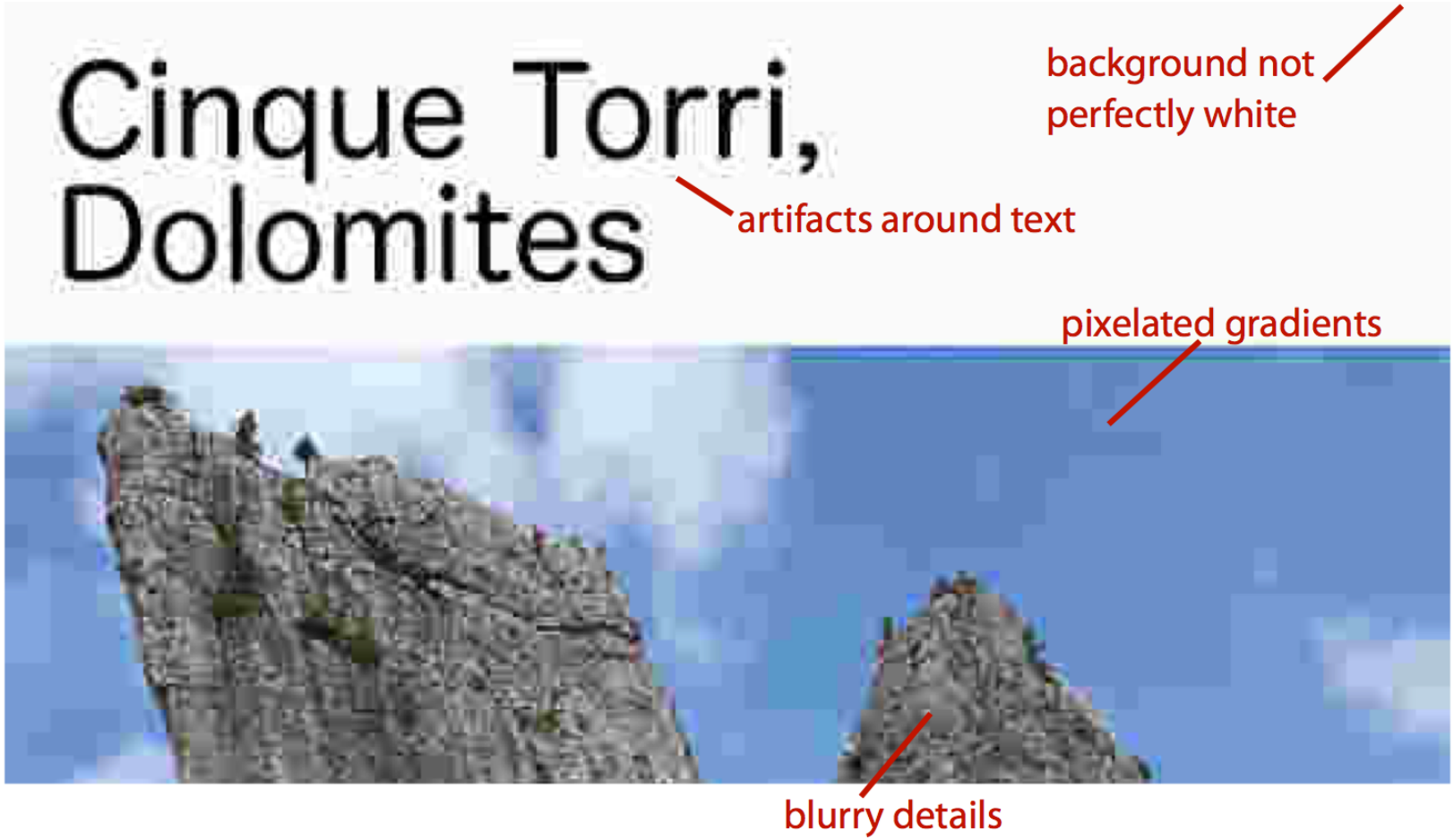


bitmap format
(TIFF, PNG, JPEG)



Wilke, 2019

JPEG is often good for compressing images, but can produce strange artifacts



Wilke, 2019

Quick guide to graphics file formats

PDF

- Standard vector graphics format
- Easily imported and edited in Illustrator
- Bad for plotting many datapoints

PNG

- Lossless bitmap compression
- Standard bitmap format for plots
- Doesn’t compress natural images well

JPEG

- Lossy image compression
- Standard bitmap format for images
- Can produce strange artifacts in plots