AFRICA DATA SCHOOL

Matplotlib: Visualization with Python

Matplotlib is a comprehensive library for creating ***static***, ***animated***, and ***interactive visualizations*** in Python.

AFRICA DATA SCHOOL

Installation

Anaconda

- conda install matplotlib

Conda-forge

- conda install -c conda-forge matplotlib

Pip

- python -m pip install -U matplotlib

AFRICA DATA SCHOOL

Basics of Matplotlib

- Matplotlib graphs your data on *Figures*.
- Each of which can contain one or more Axes.
- To create a figure with axes we use pyplot.subplots.
- Use Axes.plot to draw some data on the axes.

AFRICA DATA SCHOOL

Loading libraries

import matplotlib library

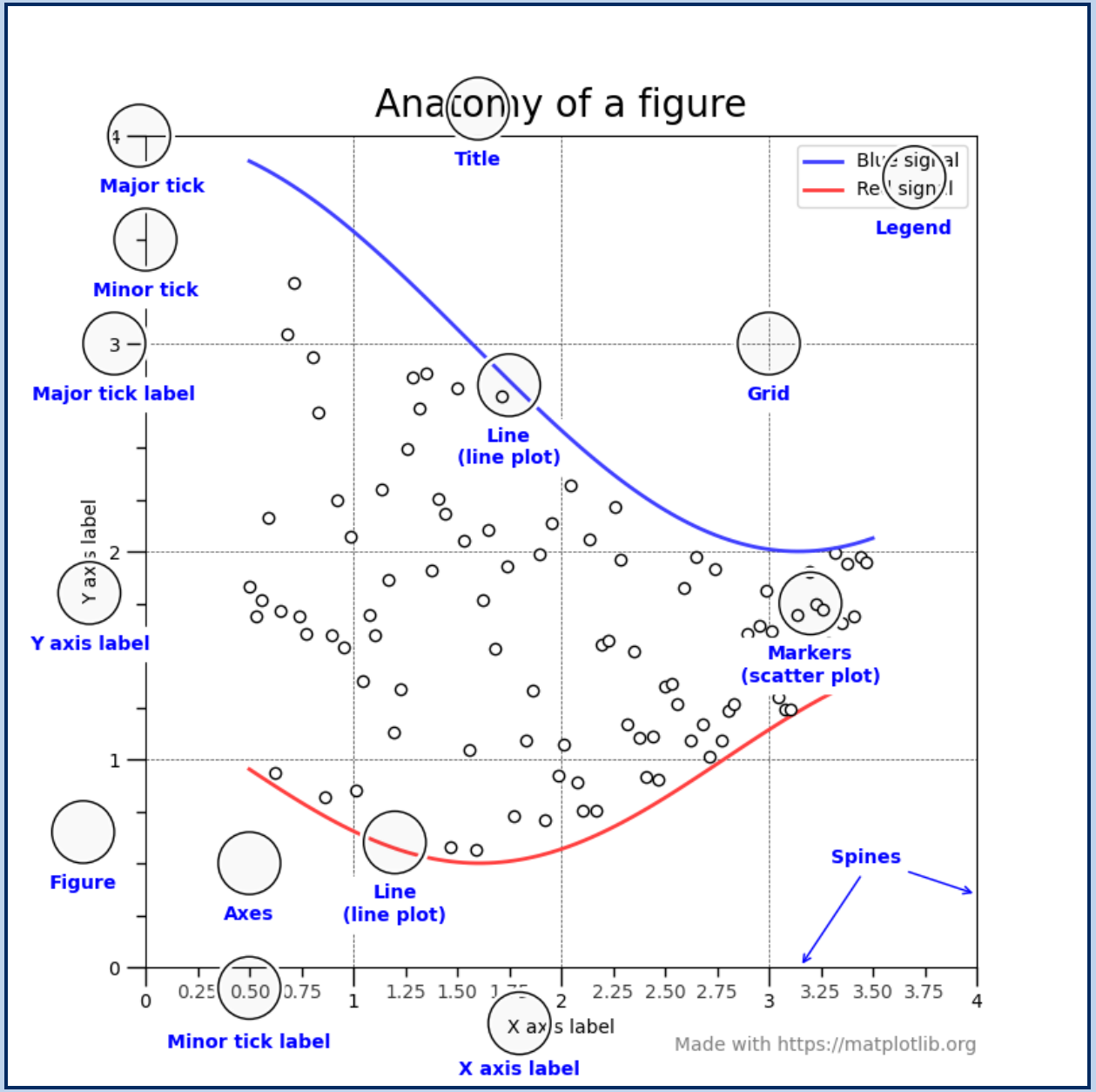
- **import** matplotlib.pyplot as **plt**
- %matplotlib inline

Alternatively

- **import** matplotlib
- matplotlib.use('qt5agg')

AFRICA DATA SCHOOL

Anatomy of a figure.



Matplotlib figure

Figure

- A canvas which contains .
- Whole figure which may contain one or more than one axes (plots)

Axis

- Number line like objects that take care of generating the graph limits.

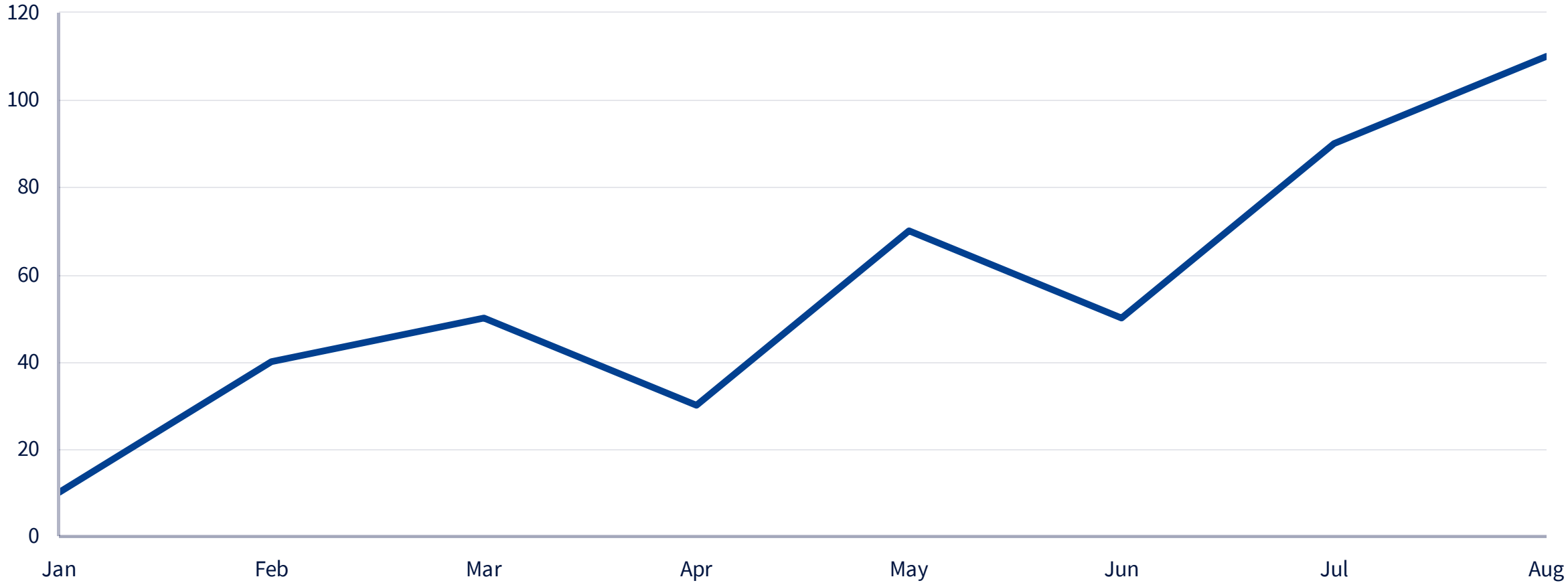
Axes

- What we generally think of as a plot.
- Each Axes has a title, an x-label and a y-label.

Artist

- Everything which one can see on the figure is an artist like Text objects, Line2D objects, collection objects.

Simple graph



AFRICA DATA SCHOOL

Note

- A given figure can contain many Axes, but a given Axes object can only be in one Figure.
- The Axes contains two (or three in the case of 3D) Axis objects
- Axes has a title (set via set_title()), an x-label (set via set_xlabel()), and a y-label set via set_ylabel()).

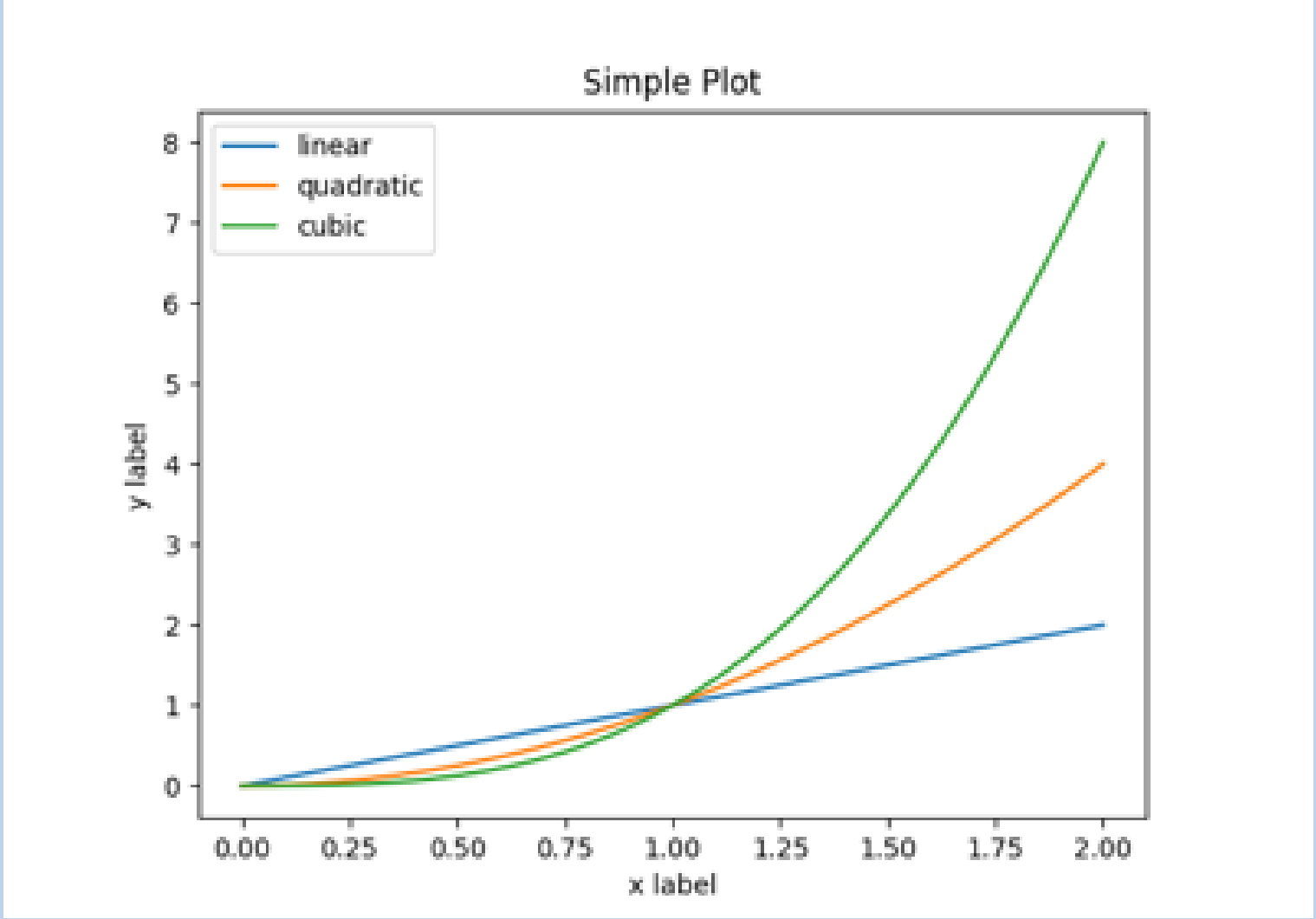
Object-oriented and pyplot interfaces

OO-style

- Explicitly create figures and axes, and call methods on them (the “object-oriented (OO) style”).
- Rely on pyplot to automatically create and manage the figures and axes, and use pyplot functions for plotting

AFRICA DATA SCHOOL

Simple plot



AFRICA DATA SCHOOL

MATPLOTLIB RECOMMENDATION

In general, we suggest to restrict pyplot to *interactive plotting* (Jupyter notebook), and to prefer the OO-style for *non-interactive plotting* (in functions and scripts that are intended to be reused as part of a larger project).

pyplot

Each pyplot function makes some change to a figure. e.g,

- Creates a figure.
- Creates a plotting area in a figure.
- Plots some lines in a plotting area.
- Decorates the plot with labels, etc

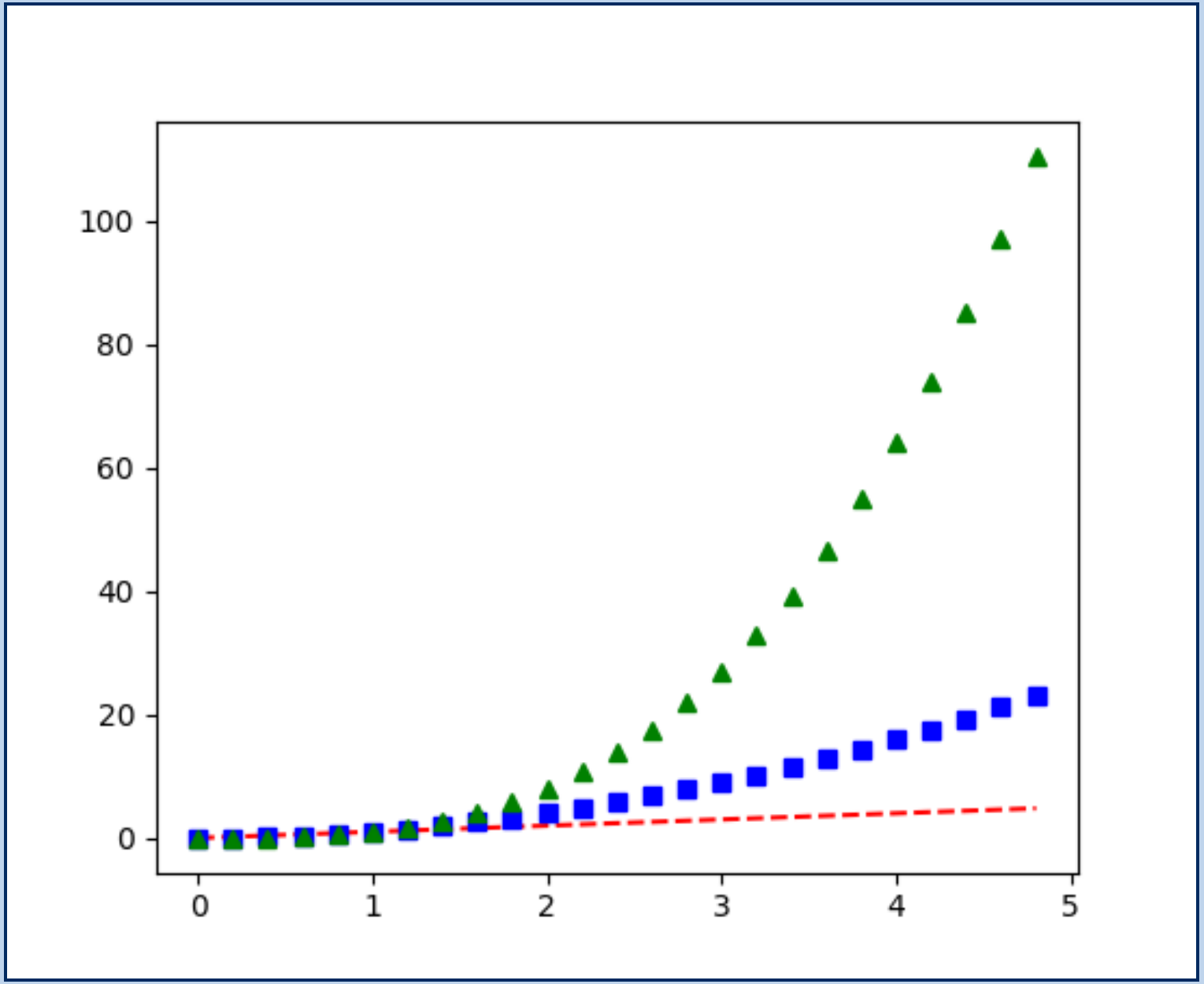
AFRICA DATA SCHOOL

pyplot properties

Property	Value Type
alpha	float
animated	[True | False]
antialiased or aa	[True | False]
clip_box	a matplotlib.transform.Bbox instance
clip_on	[True | False]
clip_path	a Path instance and a Transform instance, a Patch
color or c	any matplotlib color
contains	the hit testing function
dash_capstyle	['butt' | 'round' | 'projecting']
dash_joinstyle	['miter' | 'round' | 'bevel']
dashes	sequence of on/off ink in points
data	(np.array xdata, np.array ydata)
figure	a matplotlib.figure.Figure instance
label	any string
linestyle or ls	['-' | '--' | '-.' | ':' | 'steps' | ...]
linewidth or lw	float value in points
marker	['+' | ',' | '.' | '1' | '2' | '3' | '4']
markeredgecolor or mec	any matplotlib color
markeredgewidth or mew	float value in points
markerfacecolor or mfc	any matplotlib color
markersize or ms	float
markevery	[None | integer | (startind, stride)]
picker	used in interactive line selection
pickradius	the line pick selection radius

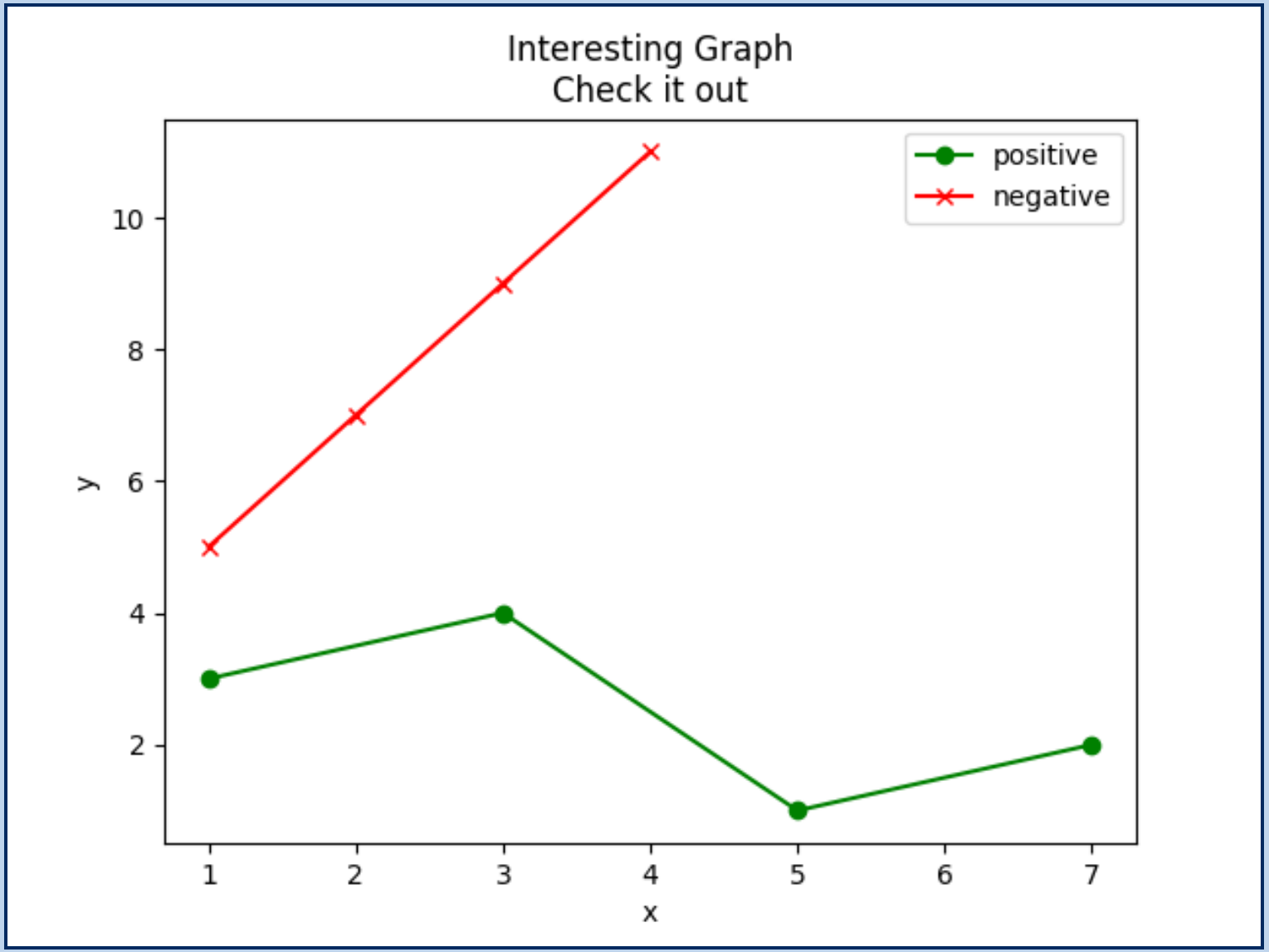
AFRICA DATA SCHOOL

Formating style of plot



AFRICA DATA SCHOOL

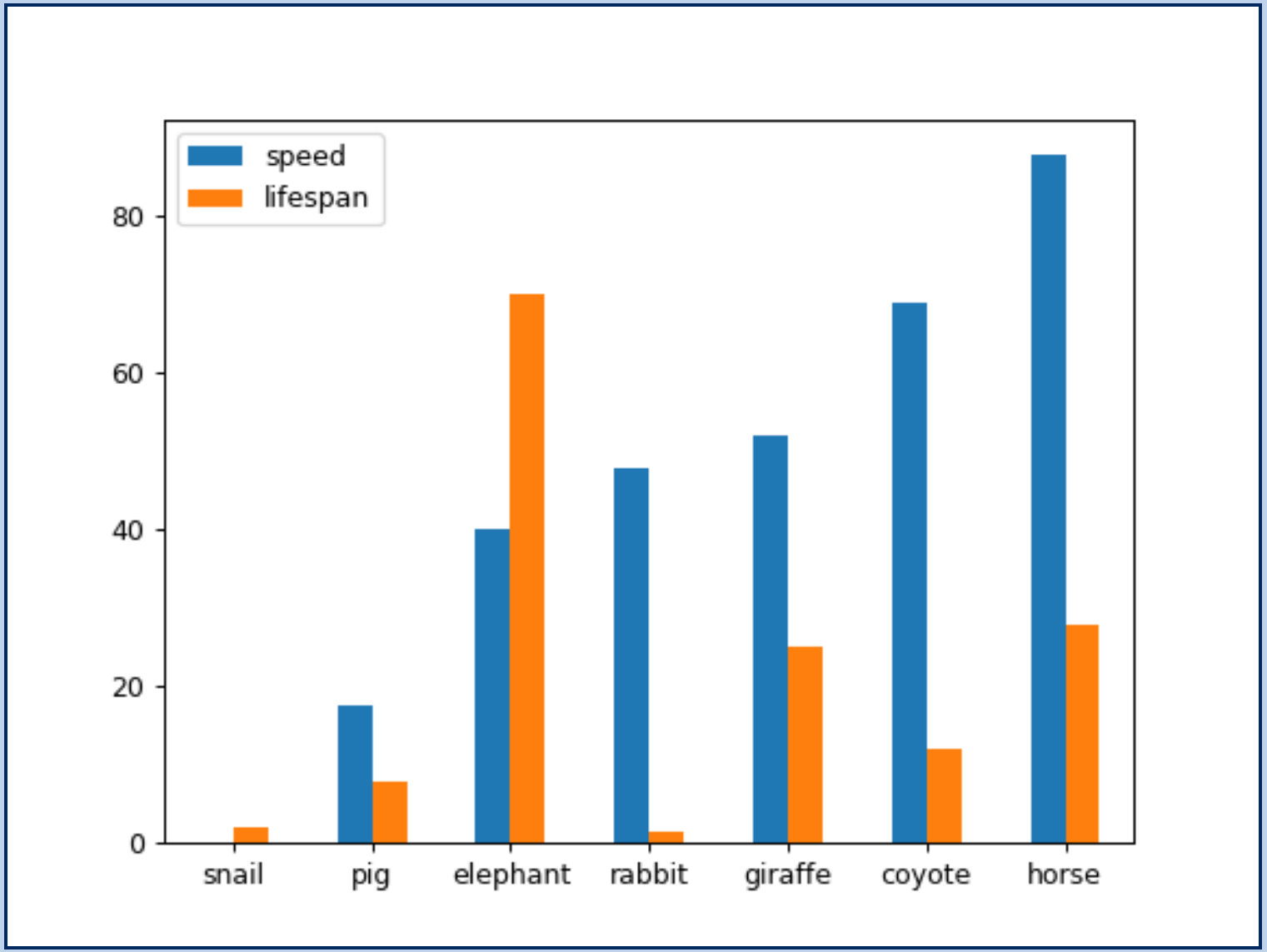
Legends, Titles, and Labels



Visualizations

Bar Charts

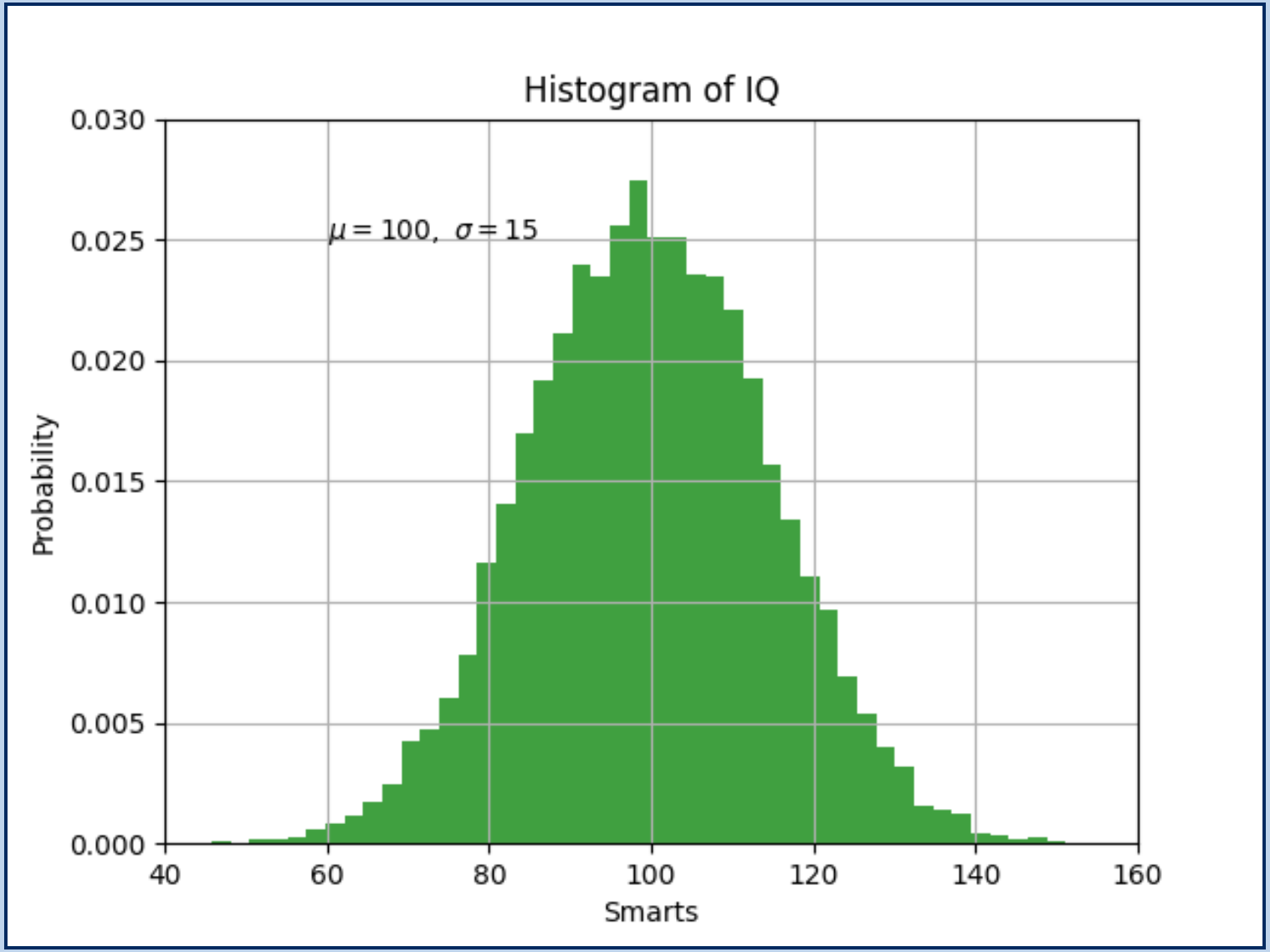
plt.bar()



AFRICA DATA SCHOOL

Histograms

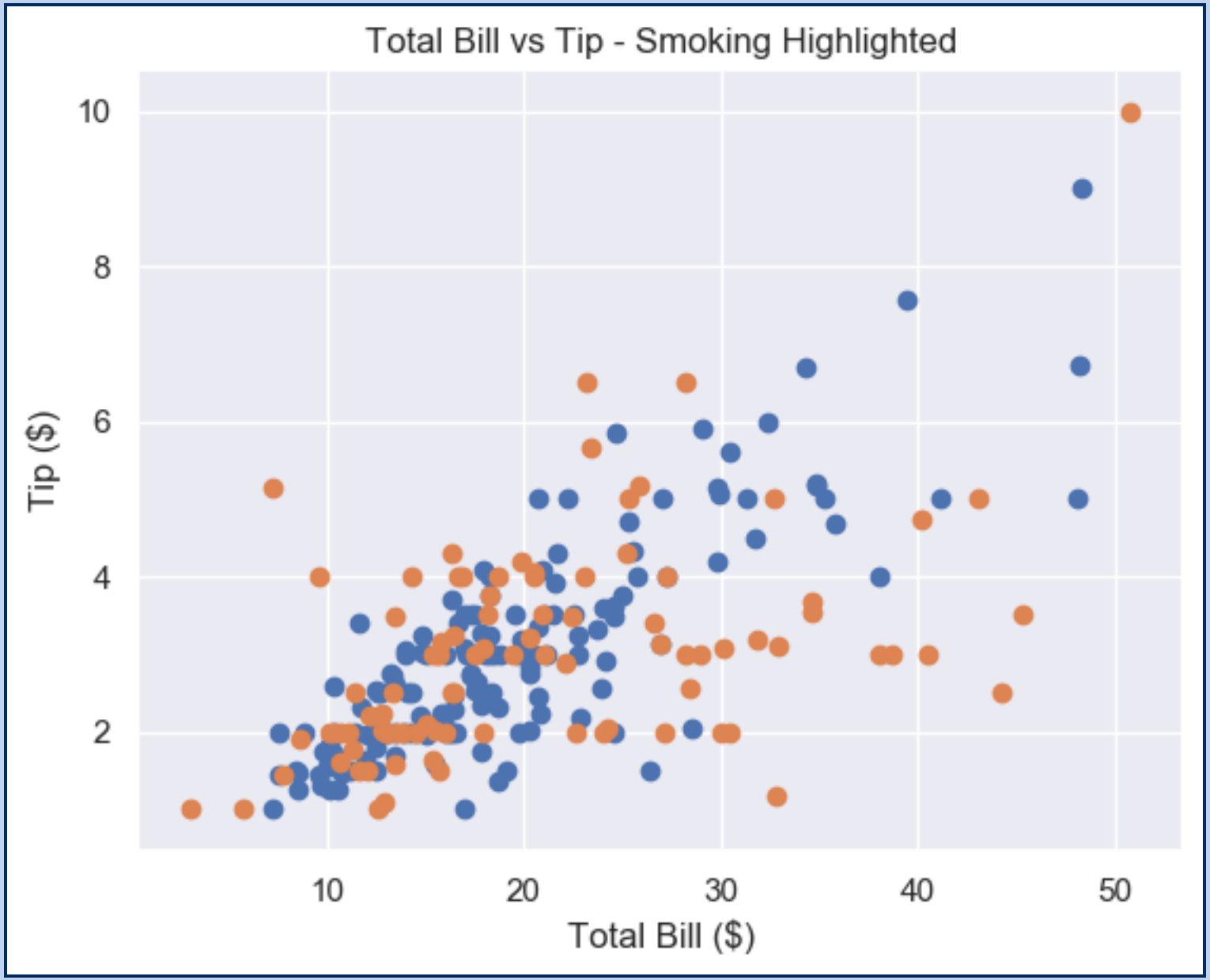
plt.hist()



AFRICA DATA SCHOOL

Scatter Plots

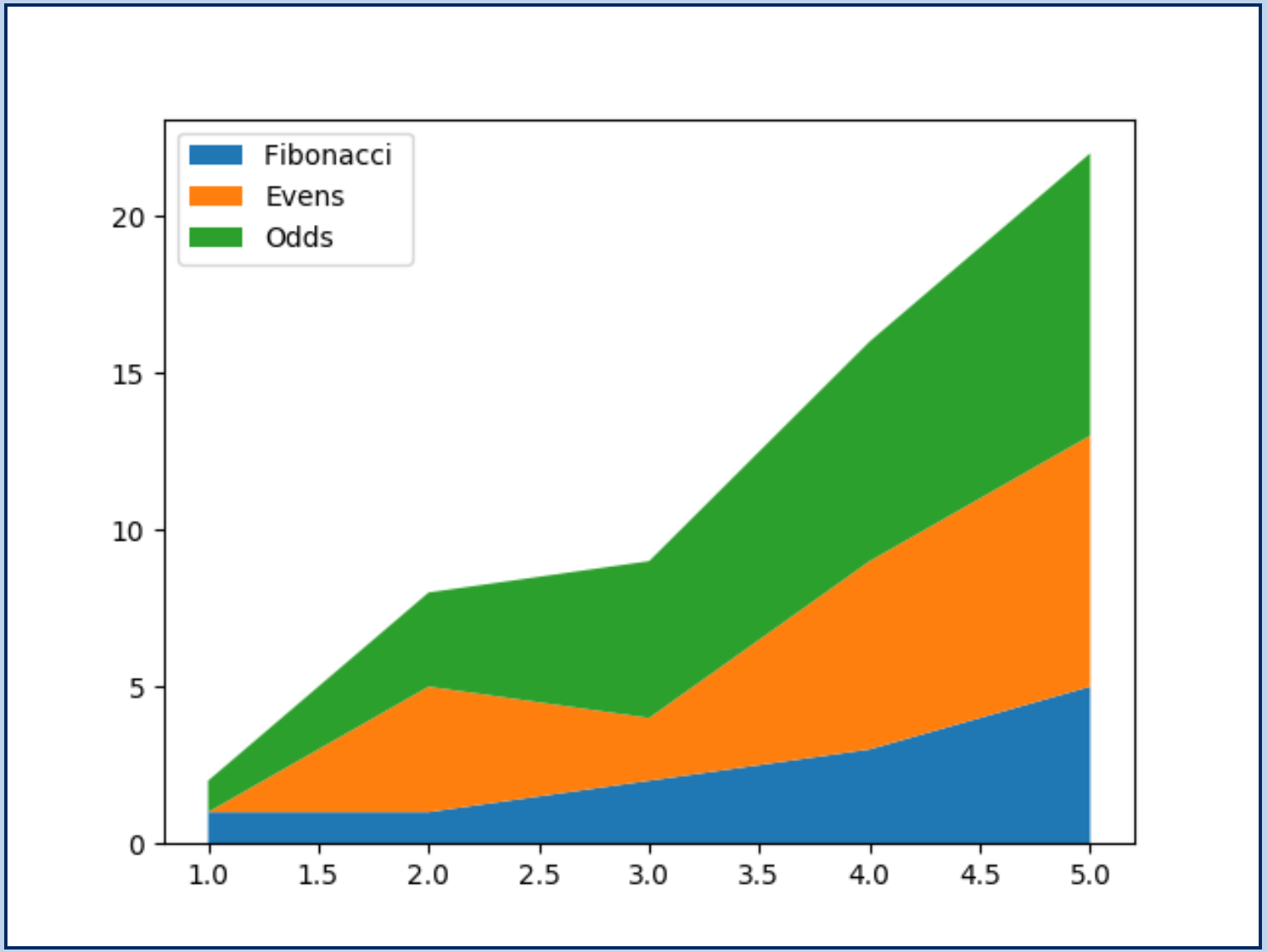
plt.scatter()



AFRICA DATA SCHOOL

Stack Plots

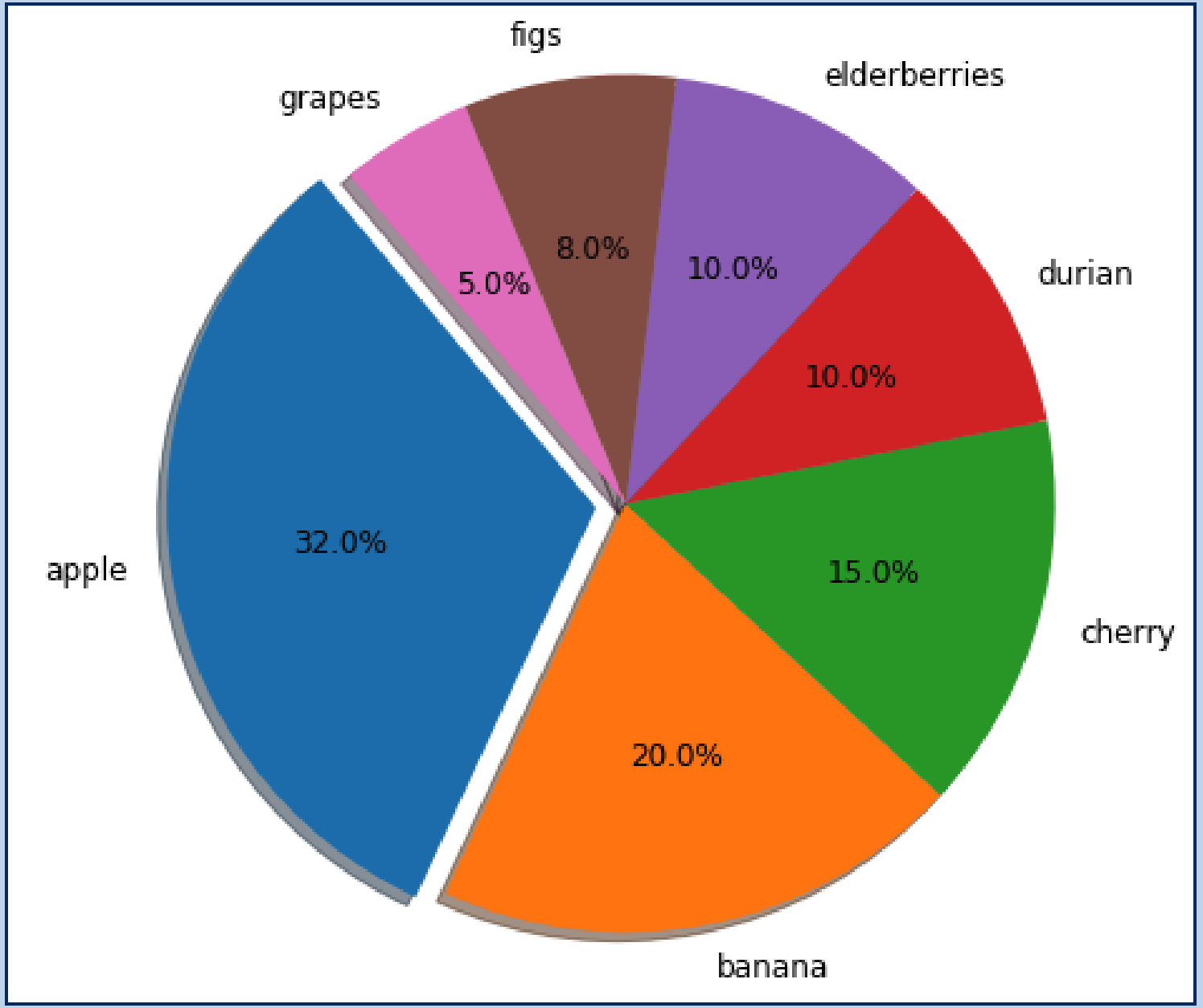
plt.stackplot()



AFRICA DATA SCHOOL

Pie Charts

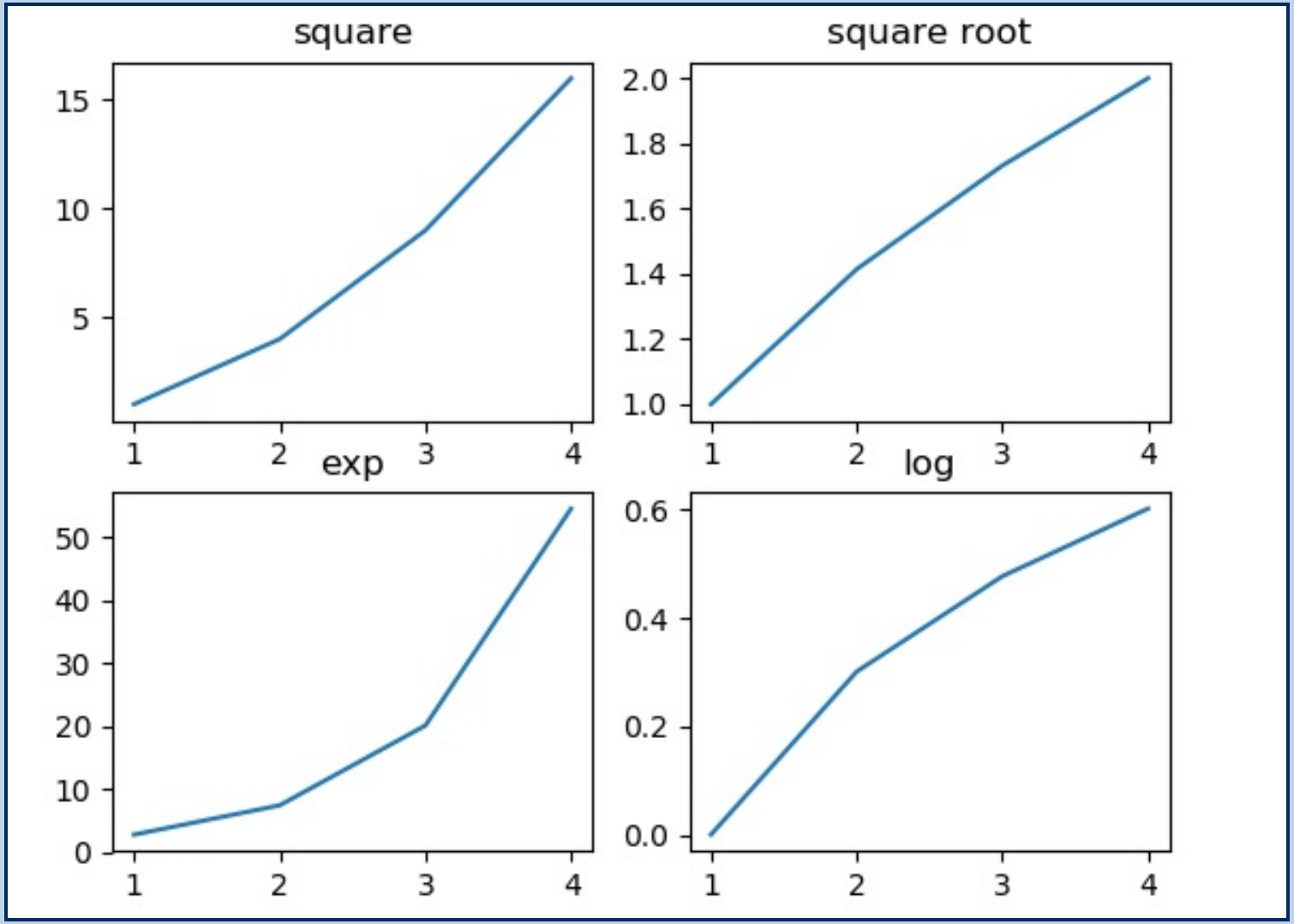
plt.pie()



AFRICA DATA SCHOOL

Matplotlib - Subplots() Function

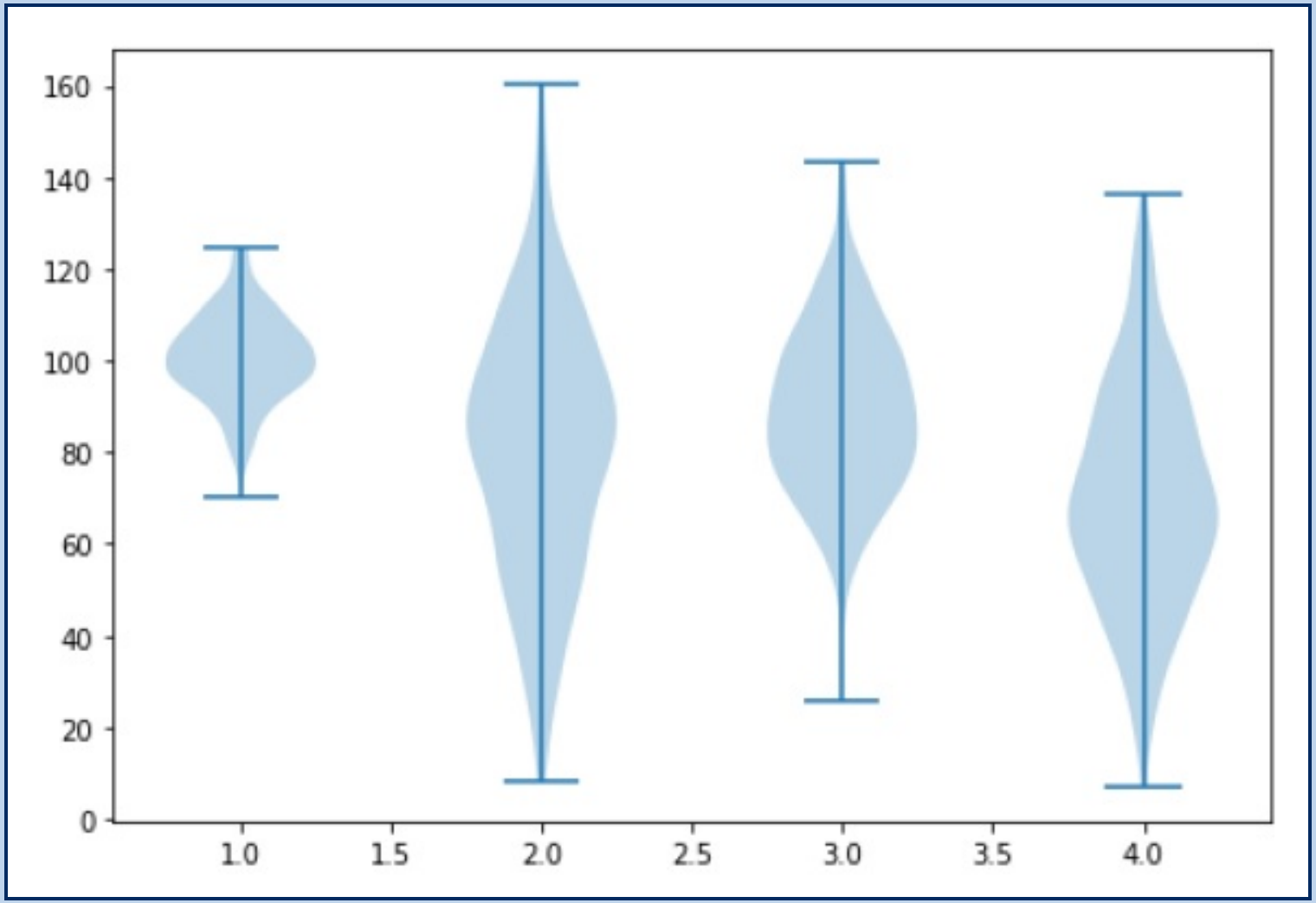
plt.subplot()



AFRICA DATA SCHOOL

Violin Plot

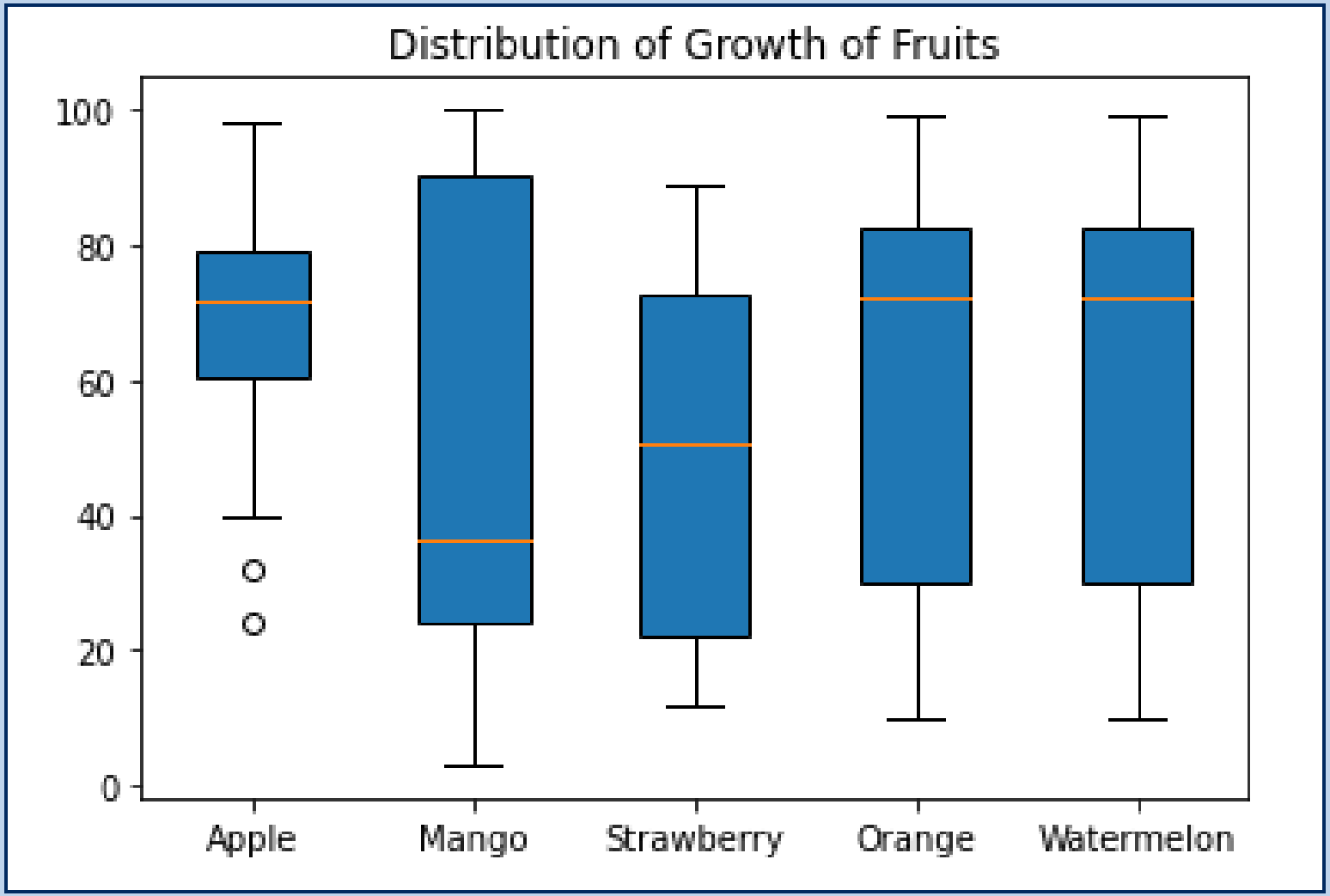
ax.violionplot()



AFRICA DATA SCHOOL

Box Plot

ax.boxplot()



AFRICA DATA SCHOOL

Backends.

What is a backend?

- To support variety of use cases, Matplotlib can target different outputs, and each of these capabilities is called a backend.
- The “frontend” is the user facing code, i.e., the plotting code, whereas the “backend” does all the hard work behind-the-scenes to make the figure.

Types of backends

User interface backends

- For use in PyQt/PySide, PyGObject, Tkinter, wxPython, or macOS/Cocoa.
- Also called, interactive backends.

Hardcopy backends

- To make image files such as PNG, SVG, PDF, PS.
- Also called, non-interactive backends.

AFRICA DATA SCHOOL

Selecting a backend

Setting rcParams[“backend”] in your matplotlibrc file

- backend : qt5agg

If your script depends on a specific backend

- import matplotlib
- matplotlib.use('qt5agg')

Setting the MPLBACKEND environment variable

> set MPLBACKEND=qt5agg

> python simple_plot.py

AFRICA DATA SCHOOL

Additional Resources

Class Notebooks

- https://github.com/Africa-Data-School/ADS_Course_Material

Matplotib Documentation

- https://matplotlib.org/stable/contents.html