PROGRAMMING IN PYTHON I

Plotting in Python

Andreas Schörgenhumer

Institute for Machine Learning

JOHANNES KEPLER
UNIVERSITY LINZ

Institute for
Machine Learning

Copyright Statement

Contact

Andreas Schörgenhumer

Institute for Machine Learning
Johannes Kepler University
Altenberger Str. 69
A-4040 Linz

E-Mail: `schoergenhumer@ml.jku.at`

Write mails only for personal questions

Institute ML Homepage

Motivation

- Often, we want to visualize data or handle images files
  - □ Visualize data and data distributions
  - □ Show/visualize image data
  - □ Create image data and save/load it from image files
  - □ …
- We will now take a look at how this is done in Python and what we have to be aware of when dealing with image data

Image Data

- Image data is often recorded as a 2D or 3D array of pixels
- Each pixel is a point in the array, carrying a value
- **Grayscale** 2D image:
  - 2D array (2 spatial dimensions)
  - Each pixel carries a brightness information
  - https://en.wikipedia.org/wiki/Grayscale
- **RGB(A)** 2D image:
  - 3D array (2 spatial dimensions + 1 dimension for color channels)
  - Each pixel carries a brightness information of a specific color channel (red, green, blue, (alpha))
  - https://en.wikipedia.org/wiki/RGB_color_model

JPEG

- **JPEG** (Joint Photographic Experts Group)
- File suffix: `.jpg` or `.jepg`
- Pixel-based (stores values of pixels in image = raster graphics)
- Uses lossy compression
  - □ Data is lost when creating the file

PNG

- **PNG** (Portable Network Graphics)
- File suffix: `.png`
- Pixel-based (stores values of pixels in image = raster graphics)
- Uses lossless compression
  - □ No data is lost when creating the file

SVG

- **SVG** (Scalable Vector Graphics)
- File suffix: `.svg`
- Vector-based (stores code to produce image, e.g., coordinates of lines = vector graphics)
  - □ Image is “drawn” based on specifications in `.svg` file
  - □ No loss of resolution when zooming into image
  - □ E.g.: Draw line from `x` to `y` with line width `w`
- Uses lossless compression
- Mainly used for images where resolution is important and vector design is feasible
  - □ Line plots, histograms, neural network architecture depictions, ...

matplotlib

- In Python, **matplotlib** is the go-to plotting tool
- Vast range of functions, documentation sometimes lacking, differences between versions
- Typical usage: Search `https://matplotlib.org/stable/gallery/index.html` for something close to what you want and continue from there.
- Documentation/Tutorials: `https://matplotlib.org/`

matplotlib: Backends

- matplotlib will use the system backends, which depend on the OS
  - Different backends for different tasks (performance, user interaction, animations, 3D plots, etc.)
  - Plots might look different on different OS due to backends
  - Functionality depends on available backends, some backends can be installed manually
- matplotlib has an interactive and non-interactive mode
  - Interactive mode will show plots immediately, non-interactive mode only when explicitly shown
- `https://matplotlib.org/stable/users/explain/figure/backends.html`

matplotlib: Figures and Axes (1)

- matplotlib works with **figures** and **axes**
- Figure:
  - The window you are plotting in
  - Comes with tools for user interaction
  - Can be saved to image files
- Axes:[1]
  - The object we can use to plot on
  - A figure can have multiple axes
  - We can draw to axes multiple times

[1] To avoid confusion with “number lines” (x-axis, y-axis,), matplotlib actually always refers to the plotting object as Ax**e**s and to these number lines as Ax**i**s.

matplotlib: Figures and Axes (2)

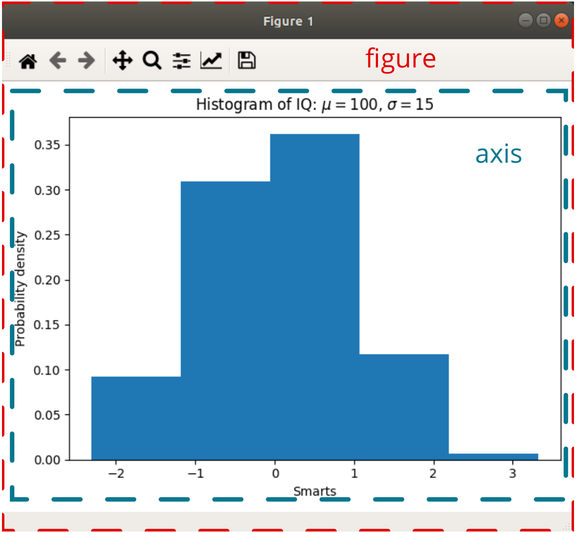
Figure 1
figure
Histogram of IQ: $\mu = 100$, $\sigma = 15$
axis
Probability density
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−2
−1
0
1
2
3
Smarts