PROGRAMMING IN PYTHON I

Unit 09: Plotting in Python

Michael Widrich
Institute for Machine Learning

PLOTTING IN PYTHON

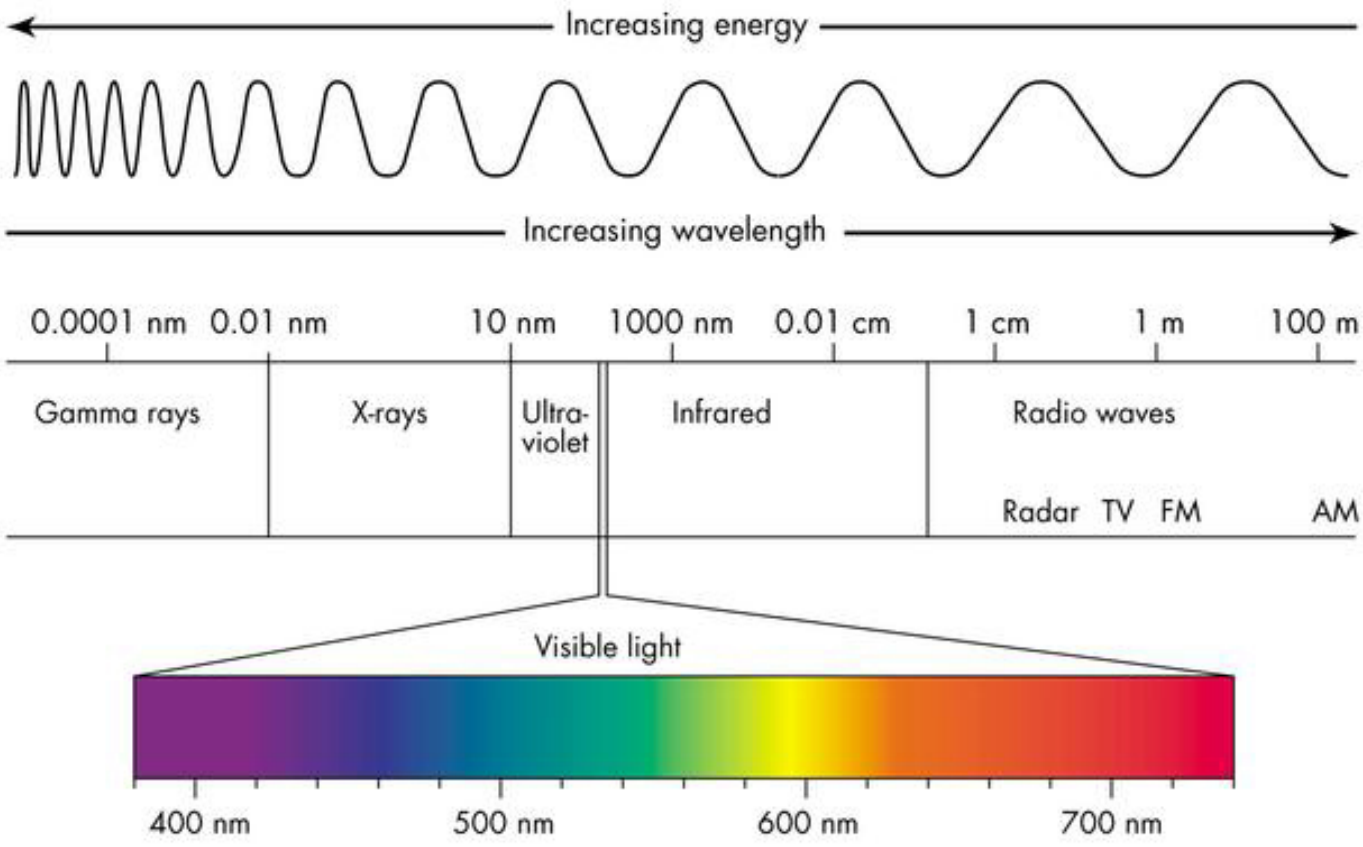
Motivation

- 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
  - Create frames and combine them to an animation/video
  - ...
- 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

VISUAL PERCEPTION

Human limitations and biases (1)

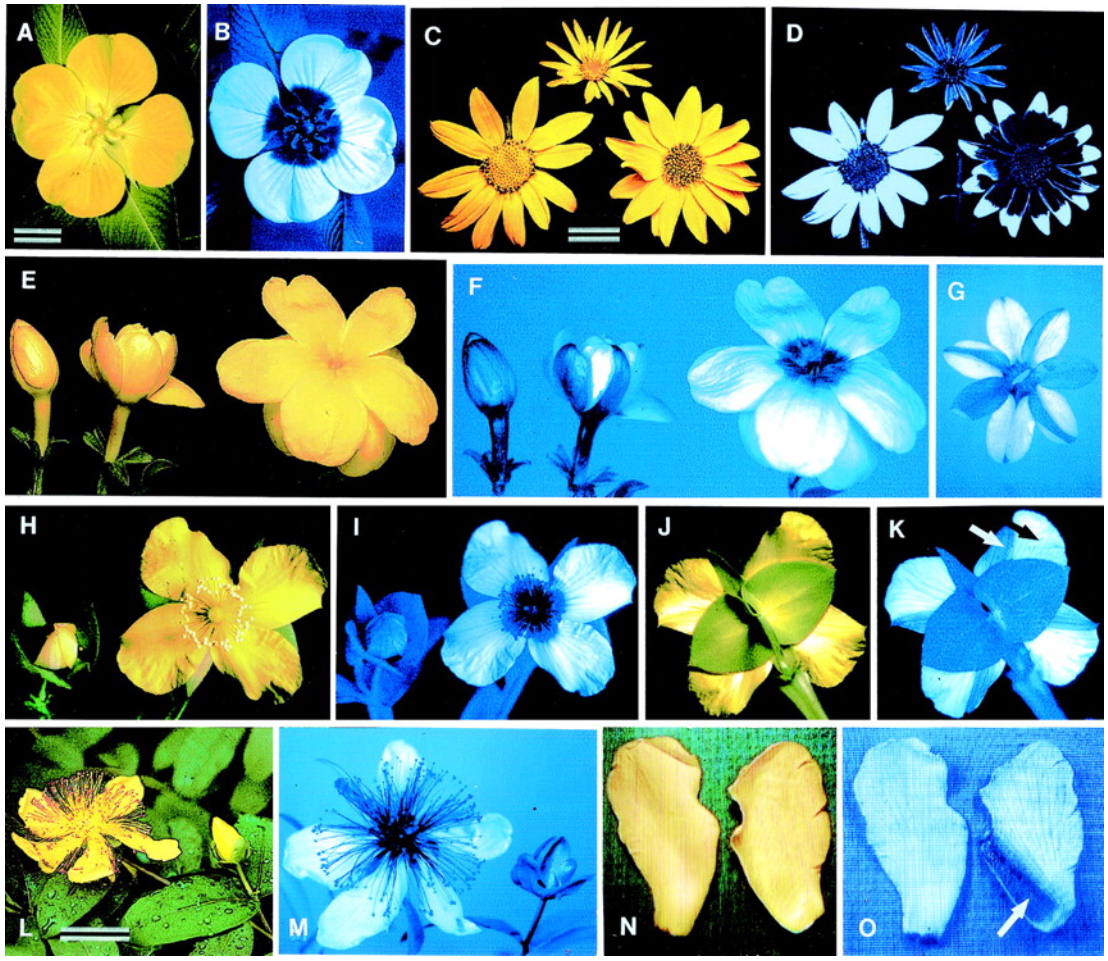
- Human eyes only perceive a very small fraction of wavelengths at a limited frame-rate



[Source: https://www.cyberphysics.co.uk/topics/radioact/Radio/EMSpectrumcolor.jpg]

Human limitations and biases (2)

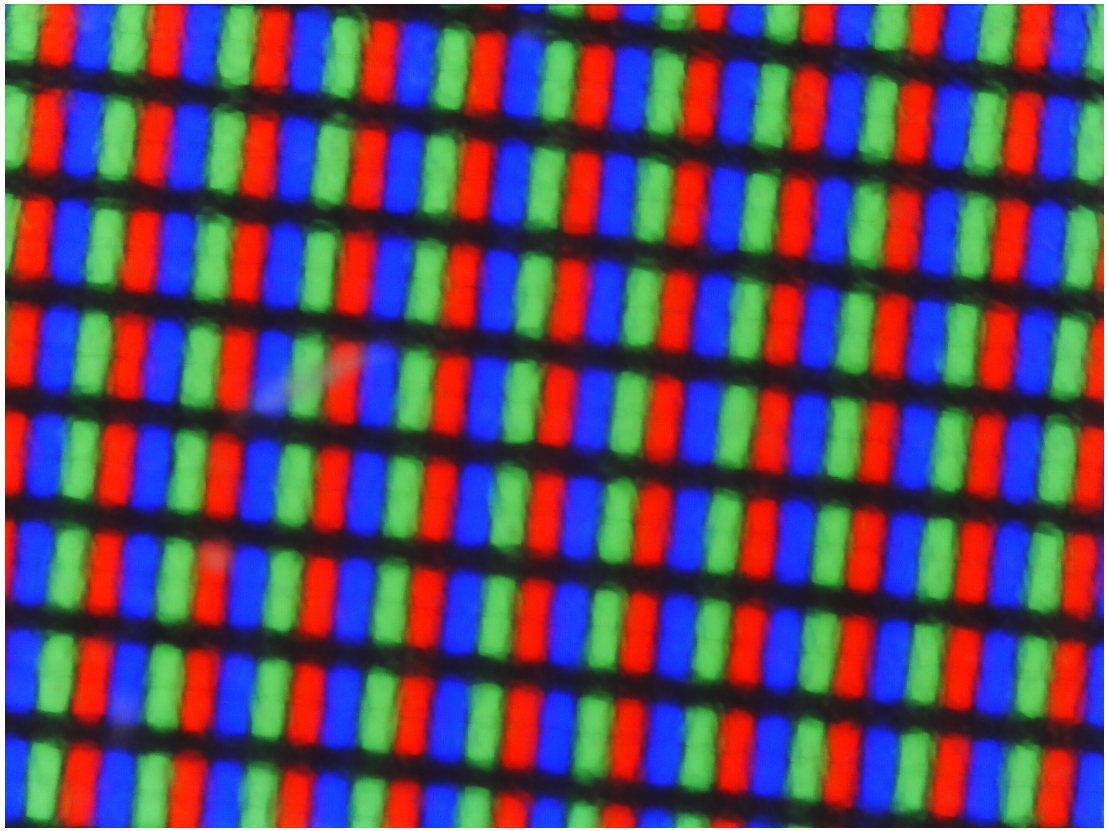
- A lot of things make more sense in spectra we cannot see
  - Flowers in UV spectrum provide signals for insects



[Source: https://www.pnas.org/content/98/24/13745]

Human limitations and biases (3)

- My PC-screen under the microscope (humans perceive this as white color)



[Image captured with `https://teleskop-austria.at/DigMic50#m`]

Human limitations and biases (4)

- Human perception is biased towards certain wave-lengths
  - Color perception differs across individuals (https://www.color-blindness.com/coblis-color-blindness-simulator/)
  - Different standards to mix RGB channels into grayscale image (`https://en.wikipedia.org/wiki/Grayscale#Colorimetric_(perceptual_luminance-preserving)_conversion_to_grayscale`)
- Especially when dealing with ML, it is important to be aware of our natural limitations and biases!

IMAGE FILES

Image data (1)

- Image data is often recorded as 2D or 3D array of pixels
- Each pixel is a point in the array, carrying a value
- Grayscale 2D image
  - 2D array (spatial dimensions)
  - Each pixel carries a brightness information
  - `https://en.wikipedia.org/wiki/Grayscale`

Image data (2)

- RGB 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)
  - `https://en.wikipedia.org/wiki/RGB_color_model`
- HSL (hue, saturation, lightness) or HSV (hue, saturation, value)
  - Like RGB but other color channels
  - `https://en.wikipedia.org/wiki/HSL_and_HSV`

JPEG

- JPEG (Joint Photographic Experts Group)
- File suffix: `.jpg` or `.jepg`
- Pixel-based (stores values of pixels in image)
- Uses lossy compression
  - Data is lost when creating the file
- Mainly used for digital images produced by photography

PNG

- PNG (Portable Network Graphics)
- File suffix: `.png`
- Pixel-based (stores values of pixels in image)
- Uses lossless compression
  - No data is lost when creating the file
- Mainly used for images where pixel values have to be kept lossless, e.g. computer games

SVG

- SVG (Scalable Vector Graphics)
- File suffix: `.svg`
- Vector-based (stores code to produce image, e.g. coordinates of lines)
  - 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 linewidth `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

matplotlib

- In Python, matplotlib is the go-to plotting tool
- Fast range of functions, documentation sometimes lacking, differences between versions
- Typical usage: Search `https://matplotlib.org/gallery` for something close to what you want to do and copy/modify code from there.
- Documentation/Tutorials: `https://matplotlib.org/`

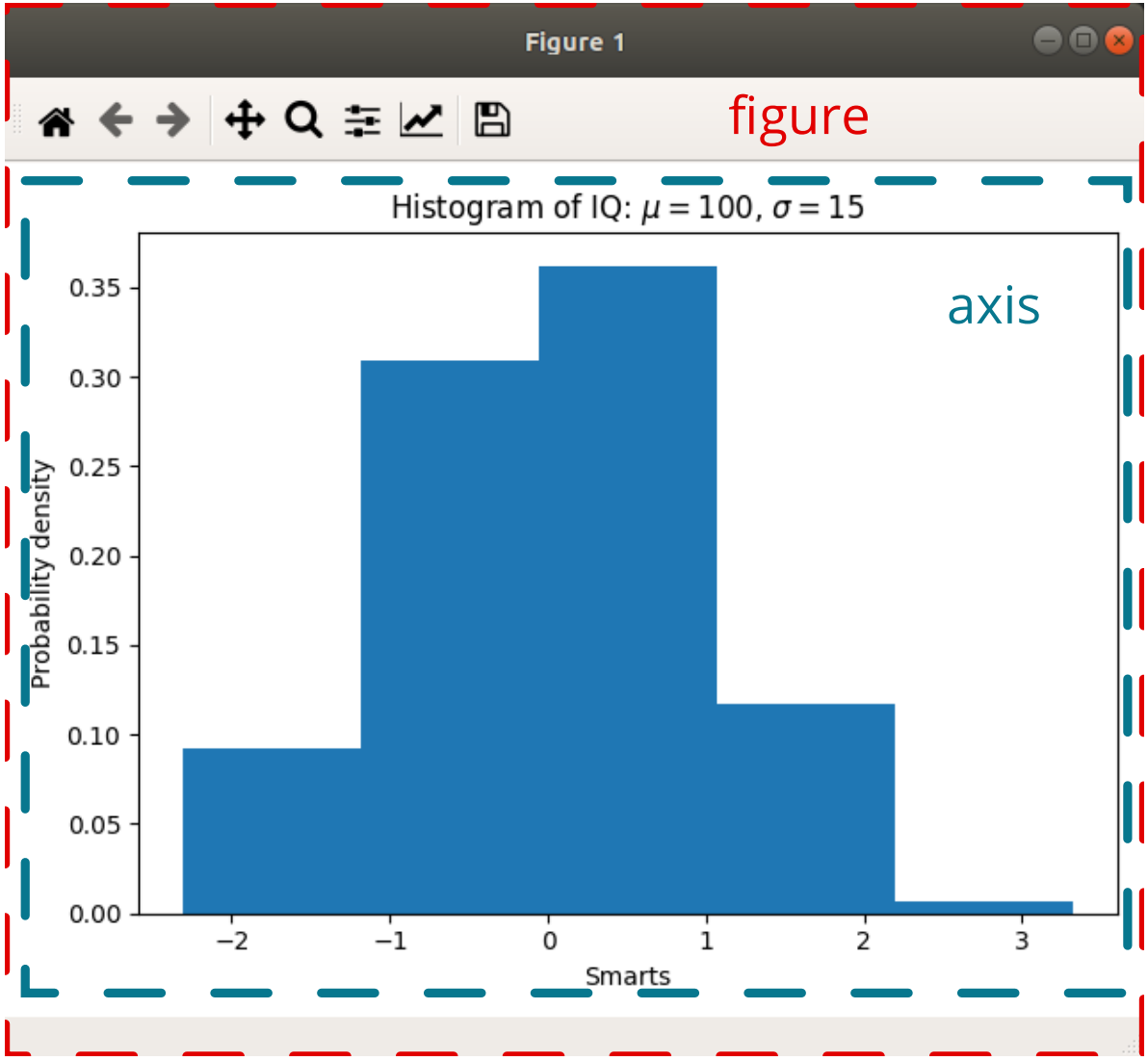
matplotlib: backends

- 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/faq/usage_faq.html#what-is-a-backend`

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 file
- axis
  - The canvas to plot on
  - A figure can have multiple axes
  - We can draw to an axis multiple times

matplotlib: figures and axes (2)



OTHER MODULES

Other modules (1)

- Pillow
  - New version of PIL (Python Imaging Library)
  - Fast image manipulation for Python
  - `https://pillow.readthedocs.io/en/stable/`
- OpenCV
  - Fast image manipulation, larger functionality than pillow but more difficult setup/usage
  - `https://docs.opencv.org/master/d6/d00/tutorial_py_root.html`

Other modules (2)

- Datashader
  - Optimized large-scale plotting in Python (scatter plots with millions of points)
  - `https://datashader.org/`
- ffmpeg
  - Go-to for videos/animations
  - Not a Python package (but wrappers are available)
  - `https://www.ffmpeg.org/`