# 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

# 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
  - ...
- 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 FILES

# Image Data

- Image data is often recorded as a 2D or 3D array of pixels

# 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

# 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

# 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

# matplotlib

- In Python, **matplotlib** is the go-to plotting tool

# matplotlib

- In Python, **matplotlib** is the go-to plotting tool
- Vast range of functions, documentation sometimes lacking, differences between versions

# 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.

# 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/backends.html

# matplotlib: Figures and Axes (1)

- matplotlib works with **figures** and **axes**

# matplotlib: Figures and Axes (1)

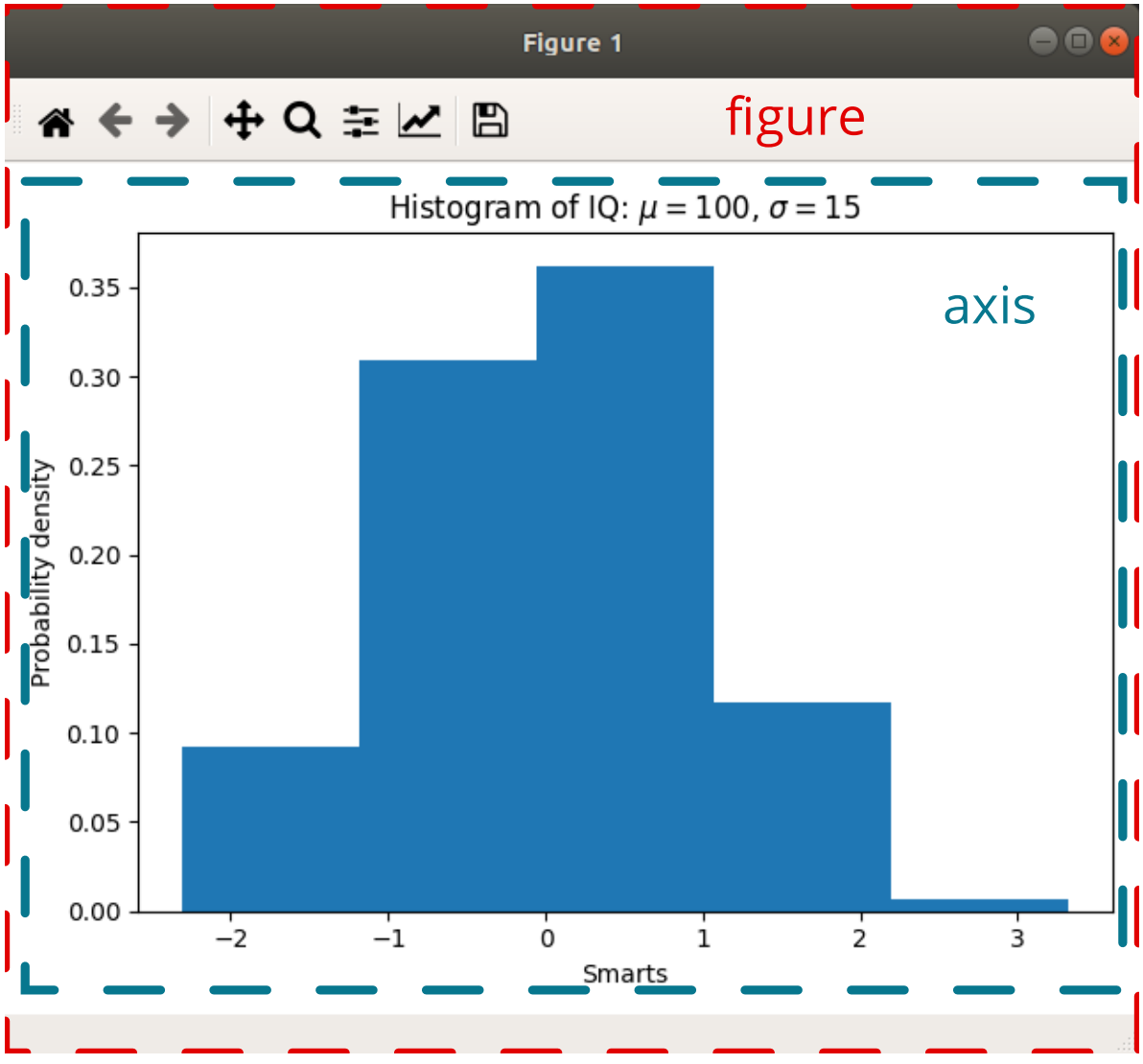
- matplotlib works with **figures** and **axes**
- Figure:
  - The window you are plotting in
  - Comes with tools for user interaction
  - Can be saved to image files

# 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
- Axis:[1]
  - The object we can use to plot on
  - A figure can have multiple axes
  - We can draw to an axis 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)



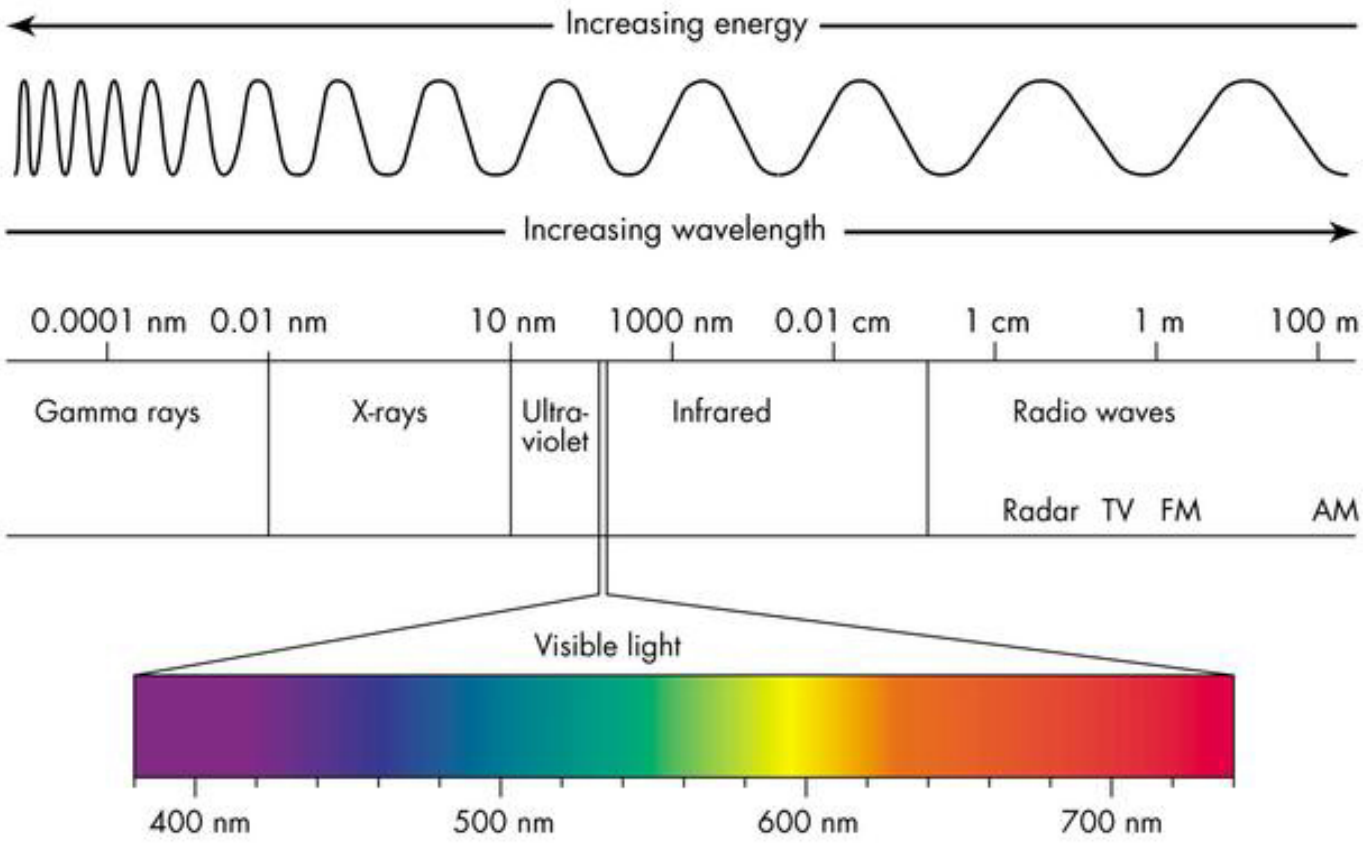
# ADDITIONAL INFORMATION: VISUAL PERCEPTION

# Visual Perception

The following (optional) slides just serve as a starting point if you are interested and want to delve deeper into the topic of visualization.

# Human Limitations and Biases (1)

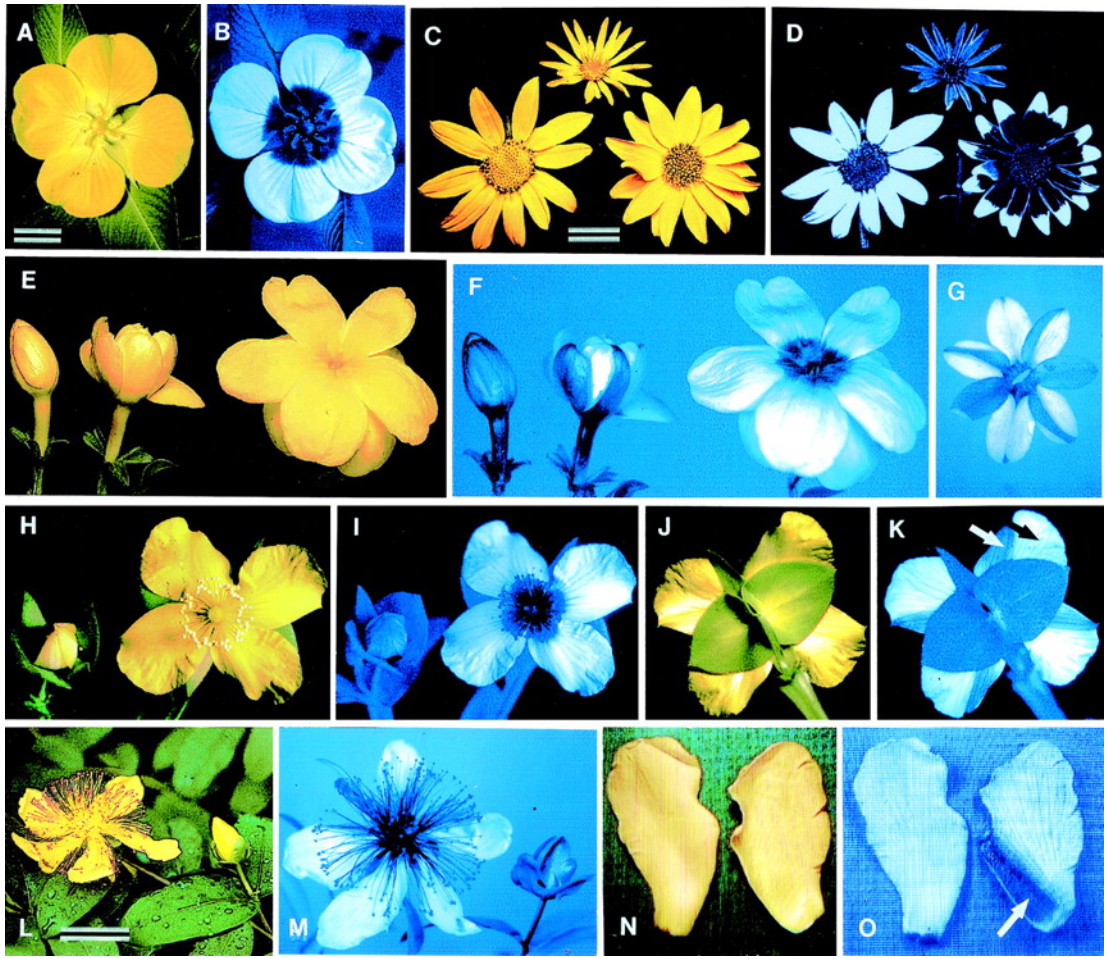
- Human eyes only **perceive a very small fraction of wavelengths** at a limited frame-rate




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

# Human Limitations and Biases (2)

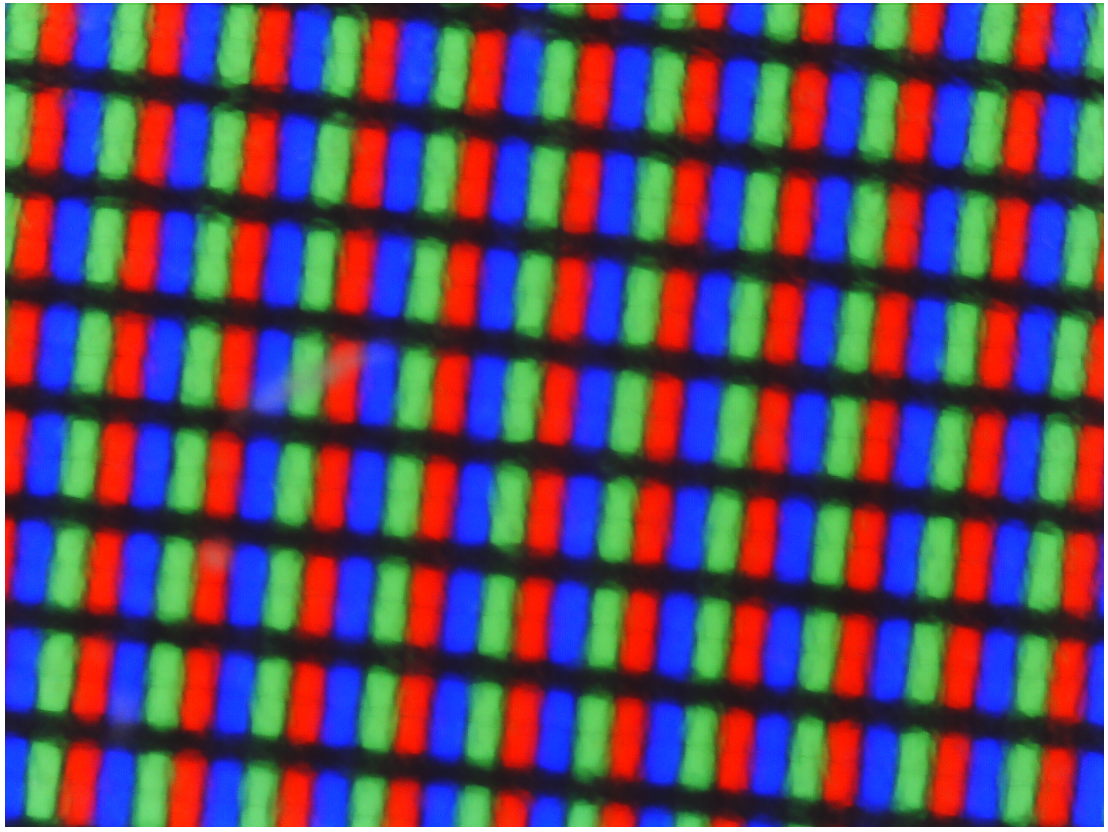
- A lot of things make more sense in spectra we cannot see
  - Flowers in UV spectrum provide signals for insects




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

# Human Limitations and Biases (3)

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



# Human Limitations and Biases (4)

- Human perception is **biased towards certain wavelengths**
  - 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!**