Visualization
Prof. Bernhard Schmitzer, Uni Göttingen, summer term 2024

Problem sheet 1

- *Submission by **2025-05-14** 18:00 via StudIP **as a single PDF/ZIP**. Please combine all results into one PDF or archive. If you work in another format (markdown, jupyter notebooks), add a PDF converted version to your submission.*
- *Use Python 3 for the programming tasks as shown in the lecture. If you cannot install Python on your system, the GWDG jupyter server at https://jupyter-cloud.gwdg.de/ might help. Your submission should contain the final images as well as the code that was used to generate them.*
- *Work in groups of up to three. Clearly indicate names and enrollment numbers of all group members at the beginning of the submission.*

Exercise 1.1: basic transformations and visualizations.
The file `mpg-data.csv` contains the `mpg` example dataset from the `ggplot2` library (https://ggplot2.tidyverse.org/reference/mpg.html) which contains information about the fuel efficiency of various car models.

1. Import the dataset into Python via pandas and briefly specify the dtype of each column (consult the documentation).
2. Split the dataset into different car classes. For each class, perform a linear regression on the dependency of `hwy` on `displ`.
3. Give a scatter plot of `hwy` against `displ`. Make sure that the class of each car can be determined from the plot. Add straight lines showing the regression lines for each class. Make sure that your plot has appropriate axes labels and legends.
4. Group the data by `class` and `year` and compute the median of `hwy` for each group. Present the resulting dataset as a table.

Exercise 1.2: algorithm runtimes.
The dataset stored in `runtimes.csv` contains information on the runtime of two algorithms (a serial version and a distributed version) on test problems of different size (measured in pixels), and for various numbers of worker threads (for the distributed version).

1. Inspect the file and import the dataset into Python as a pandas dataframe. Note that one variable (the number of threads) is encoded in column names. Bring this dataset into *tidy* form, as described in https://r4ds.hadley.nz/data-tidy.html using functions such as `pandas.melt` (and some post-processing with auxiliary functions). Make sure that columns in the resulting dataframe have meaningful names and dtypes.
2. Create a chart that examines how fast runtime increases with problem size, for the single and distributed versions, and for different numbers of threads for the latter.
3. Create a chart that examines how the runtime depends on the number of threads for the distributed version. What would be the ideal dependency? The chart should allow to compare the ideal case with the actual observations.

Exercise 1.3: hue rotation.
Import the photo of parrots used in the lecture (available at https://en.wikipedia.org/wiki/File:BlueAndYellowMacaw_AraArarauna.jpg) in Python as shown in the lecture. Then for a given angle $\varphi \in [0, 2\pi)$, implement a function that converts the image to HSV space, applies a rotation by angle φ to the hue channel (where $\varphi = 2\pi$ would correspond to a whole rotation and is thus equivalent to $\varphi = 0$), and transforms the resulting image back to RGB space. Apply this function to the image for $\varphi \in \{\frac{k}{2\pi} | k \in \{0, 1, 2, 3, 4\}\}$ and visualize the obtained 'rotated' images.