Visualization
Prof. Bernhard Schmitzer, Uni Göttingen, summer term 2023

Problem sheet 2

- *Submission by 2023-05-23 18:00 via StudIP* ***as a single PDF****. Please combine all results into one PDF. If you work in another format (markdown, jupyter notebooks), convert them to PDF before 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 two to three. Clearly indicate names and matrikelnr of all group members at the beginning of the submission.*

Exercise 2.1: customer survey.
A local shop has performed a customer survey where participants answered four questions on a rating scale from 'excellent' to 'appalling'. Not all submitted questionnaires were necessarily complete. The questions and the obtained numbers of answers for each rating are given below:

Question	number of answers				
	excellent	good	average	poor	appalling
Quality of products	191	184	133	82	25
Do you like our shop?	448	16	12	146	0
Customer service	63	127	171	191	58
Overall experience	40	100	126	52	26

You are tasked with visualizing the data. Keep the following criteria in mind:

- The plot should reveal *cumulative* information about the survey. For instance, what fraction of customers rates the service at least average or better?
- Both the relative distribution of ratings as well as the absolute number of answers for each question should be represented visually, not necessarily within a single plot.
- Though not strictly necessary, think about how color can be used to make the plot more intuitive.

Exercise 2.2: visualizing the exponential function.
The exponential function takes complex numbers to complex numbers, $\mathbb{C} \ni z \mapsto e^z \in \mathbb{C}$. Identifying $\mathbb{C}$ with $\mathbb{R}^2$ in the usual way, $z = x + i \cdot y$ for $z \in \mathbb{C}$, $(x, y) \in \mathbb{R}^2$ and i being the imaginary unit, the exponential function can be interpreted as a map from $\mathbb{R}^2$ to $\mathbb{R}^2$. In this representation, it can be written as

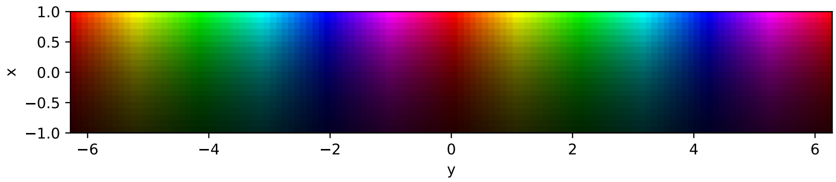
$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} e^x \cdot \cos(y) \\ e^x \cdot \sin(y) \end{pmatrix}.$$

This exercise aims at visualizing this function by using the HSV color space.

1. Create a Cartesian grid of points over $(x, y) \in [-1, 1] \times [-2\pi, 2\pi]$ as shown in the lecture. Evaluate the exponential function for all points on the grid. Translate the result to polar coordinates (as shown in the lecture).

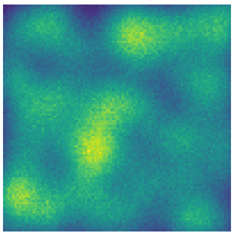
2. As in the lecture, create an empty array for the HSV image over the grid. Set the hue values according to the angle of the polar coordinates, set the saturation to 1, set the value according to the radius of the polar coordinates. Convert the image to RGB and display the result.

The outcome should look similar to this (rotated for better formatting):



Exercise 2.3: particle colocalization.
We will perform a simple toy example for colocalization analysis of particle species.

1. Download the file `problem-sheet-02_proteins.npz` from StudIP and import it with `np.load` into Python. It contains three grey level images, stored as keys `"A"`, `"B"`, and `"C"` in the imported object. Display them. Image `"A"` should look as follows:

   Each image represents the observed distribution of a corresponding particle species (e.g. proteins in super-resolution microscopy experiments). We want to find out if the locations of species `"A"`, `"B"`, and `"C"` are correlated or not.

2. One way to study visually whether two intensity images are correlated is by superimposing them, e.g. as different channels of a multi-color image. Choose a concrete way to do this and describe how can one qualitatively distinguish correlated and uncorrelated images. Apply this to the pairs (`"A"`,`"B"`) and (`"A"`,`"C"`) and form a conjecture about which two images are correlated.

3. Finally, visualize directly the pixel-wise correlation, i.e. show the relation between the pixel-wise intensities of image `"A"` versus those of `"B"` and `"C"`. This should give a more quantitative impression.