Visualization
Prof. Bernhard Schmitzer, Uni Göttingen, summer term 2025

Problem sheet 1

- *Submission by* ***2025-05-28*** *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: precipitation data.
In this exercise we process precipitation data of the Deutscher Wetterdienst (DWD). The original data is available at `https://www.dwd.de/DE/leistungen/cdc/cdc_ueberblick-klimadaten.html`, but all data required for the exercise is already provided in the zip file of the problem sheet.

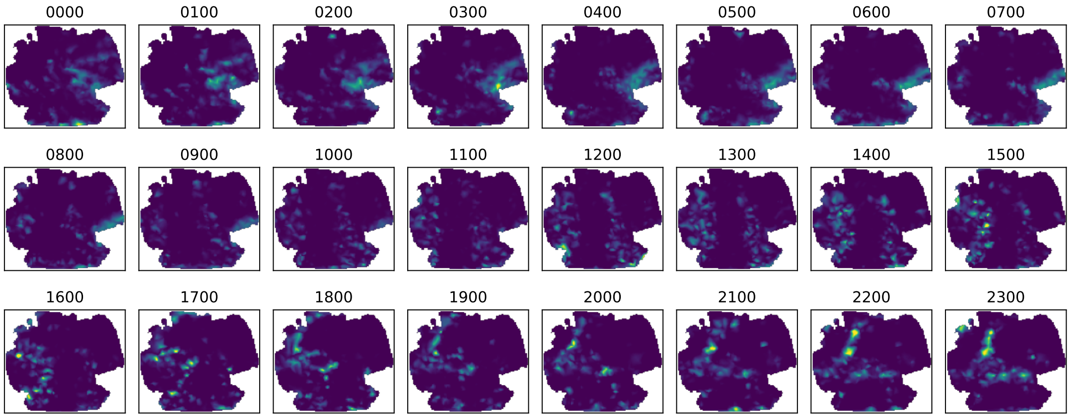
1. The file `zehn_min_rr_Beschreibung_Stationen.txt` (as available on the DWD website) contains basic information about the weather measurement stations. Its format should be self-explanatory. Convert it into reasonable CSV format. As a warmup, create a scatter plot of the geographical position of all weather stations and their elevation.

2. The file `10min_processed.csv` contains condensed precipitation data for the day 2024-04-20 in intervals of 10 minutes for (a subset of) the weather stations listed above. The column `stationid` corresponds to the column `Stations_id` in the other table. The column `date` indicates the beginning of the 10 minute interval in the format YYYYMMDDHHMM. The column `rain` encodes precipitation in this interval in millimeters, missing values are encoded as -999.

   Compute the total precipitation at each weather station within each hour of the observed day. For one (or multiple) hours, show the precipitation of all weather stations in a scatter plot. Truncate missing values to zero.

3. The file `griddata.npz` contains the arrays `geolat`, `geolong` and `ind`. The two former arrays contain latitude and longitude positions of a regular rectangular Cartesian grid covering the measurement area, the third array contains an approximate binary indicator of the region covered by the measurements (with `True` indicating coverage). Interpolate the above precipitation data from the weather stations to this grid (only to the region indicated by `ind`) and display the data as an image. This gives a smoother visual impression of the geographical distribution of precipitation. Do this for all hours of the day as a small multiple.

   The resulting figure could look approximately like this:



Exercise 1.2: meshes.

1. The following array contains the vertices of the unit cube in three dimensions:

```
points=np.array([[0,0,0],[0,1,0],[0,1,1],[0,0,1],
     [1,0,0],[1,1,0],[1,1,1],[1,0,1]],dtype=np.double)
```

   Similar as in the lecture, give a list of triangles that triangulate the surface of the unit cube as a `numpy` array. Use the mesh drawing functionality of plotly to visualize the cube based on this mesh (points + triangles).

2. Build a triangular mesh approximation for the unit disk

$$D = \{(x, y, z) \in \mathbb{R}^3 \,|\, x^2 + y^2 \leq 1,\ z = 0\}.$$

   You can do this in a way similar to the Möbius strip and donut example from the lecture, by building a rectangular mesh first and then applying a suitable coordinate transformation (*Hint: polar coordinates*). Visualize this mesh with plotly, as above.

3. Build a mesh of the surface of the cylinder

$$C = \{(x, y, z) \in \mathbb{R}^3 \,|\, x^2 + y^2 \leq 1,\ z \in [0, 1]\}$$

   and visualize it with plotly.