Tutorial VII - NumPy and Pandas for Scientific Computing

Programming with Python

Analyzing Climate Data

Imagine you're a climate scientist working on a project to analyze temperature data from weather stations across the country. You've been given a large dataset, and you need to use NumPy to process and analyze this data efficiently.

```python
import numpy as np

# First, we simulate loading data from 100 weather stations over 365 days. Each row
↪   represents a station, each column a day.
temp_data = np.random.randint(0, 40, size=(100, 365))

# TODO: a) Calculate the average temperature for each station and the overall average
↪   temperature.
# Your code here

# TODO: b) Find the highest and lowest temperature recorded and print an message with the
↪   corresponding stations.
# Your code here

# TODO: c) Identify heat waves. A heat wave is defined as 5 consecutive days with
↪   temperatures above 30°C. Print a message counting the number of heat waves.
# Your code here

# TODO: d) Calculate the temperature anomaly for each day (difference from each indidvual
↪   station's average temperature).
# Your code here

# TODO: e) Find the hottest and coldest stations and determine the index of the station
↪   with the highest average temperature and the station with the lowest average
↪   temperature.
# Your code here
```

NASA GISTEMP Climate Change Analysis

In this exercise, you'll use Pandas to analyze real global temperature anomaly data from NASA, helping to understand trends in climate change over time.

The dataset is provided by the GISS Team, 2024: GISS Surface Temperature Analysis (GISTEMP), version 4. NASA Goddard Institute for Space Studies. Dataset at https://data.giss.nasa.gov/gistemp/.

```python
import pandas as pd
import matplotlib.pyplot as plt

# First, we load the NASA GISTEMP dataset for global temperature anomalies.
url = "https://data.giss.nasa.gov/gistemp/tabledata_v4/GLB.Ts+dSST.csv"
temp_anomaly_data = pd.read_csv(url, skiprows=1) # skiprows=1 ensures that the first
↪  column is not read as a row index

# TODO: a) Display the first 5 rows to learn basic information about the DataFrame. For
↪  your work, you only need the 'Year' and and the data from all months. Drop the rest
↪  of the columns.
# Your code here

# TODO: b) Calculate and print the average temperature anomaly for each year. To do so,
↪  you first need to pd.melt()` the DataFrame to convert months to a single column.
# Your code here

# TODO: c) Find the year with the highest temperature anomaly and the year with the
↪  lowest.
# Your code here

# TODO: d) Create a new column 'Anomaly_Category' with values 'Cool', 'Neutral', or
↪  'Warm' based on the anomaly value. All values below -0.2 should be 'Cool', all values
↪  above 0.2 should be 'Warm' and all values around 0 should be 'Neutral'.
# Your code here

# TODO: e) Calculate the percentage of 'Warm' months for each decade, not for each year!
# Your code here

# TODO: f) Save the DataFrame with the Anomaly_Category column to a Excel file with the
↪  name 'temp_anomaly_data.xlsx' for next lecture.
# Your code here
```

That's it!

You can find the solutions to these exercises online in the associated GitHub repository, but we will also quickly go over them in next week's tutorial. To access the solutions, click on the Github button on the lower right and search for the folder with today's lecture and tutorial. Alternatively, you can ask ChatGPT or Claude to explain them to you. Remember, the goal is not just to complete the exercises, but to understand the concepts and improve your programming abilities.