Assignment IV: APIs and Web Scraping

Data Science Project Management (DS400) | Winter Term 2024/25

Formal requirements

Submit your assignment earlier than **January 21, 2025, 12:00 noon**. The submission must include your code, results, and explanations for code and results, for both **R** and **python**.

Submit your assignment via ILIAS. Upload the `*.html`-files that were rendered from your `*.Rmd` and `*.ipynb`, or `.qmd` solutions. You can optionally upload the `*.Rmd` and `*.ipynb`, or `qmd` files as well. You must use the following naming convention: `<Lastname>_<Firstname>_Assignment##_<Language>.<extension>` (e.g., `Lovelace_Ada_Assignment04_R.html`).

You will receive up to **7.5 points** upon passing this assignment.

Code of conduct

By submitting the assignment, you are acknowledging that the submitted assignment is your own work. This implies that …

- the work you submit is your own
- the code you wrote is your own
- any comment or interpretation is in your own words
- you made clear whom you worked with

Disregarding this code of conduct will result in failure of the assignment!

Open-Meteo APIs

Open-Meteo provides access to weather data from various weather stations. In this assignment, you will use the Open-Meteo APIs to retrieve historic weather data for a given location (here: Tübingen). You will then process the data and create a temperature anomaly plot.

NOTE: Do not use the openmeteo-requests module in Python or the openmeteo package in R. Instead, use the `requests` module in python and the `httr` package in R to interact with the Open-Meteo APIs.

Part 1: The geocoding API

The Open Meteo API requires the coordinates of the location for which you want to retrieve weather data. To get the coordinates for Tübingen, use Open-Meteo's geocoding API.

1. Use the Open-Meteo geocoding API to retrieve the coordinates for Tübingen.

2. Print the coordinates (longitude and latitude).

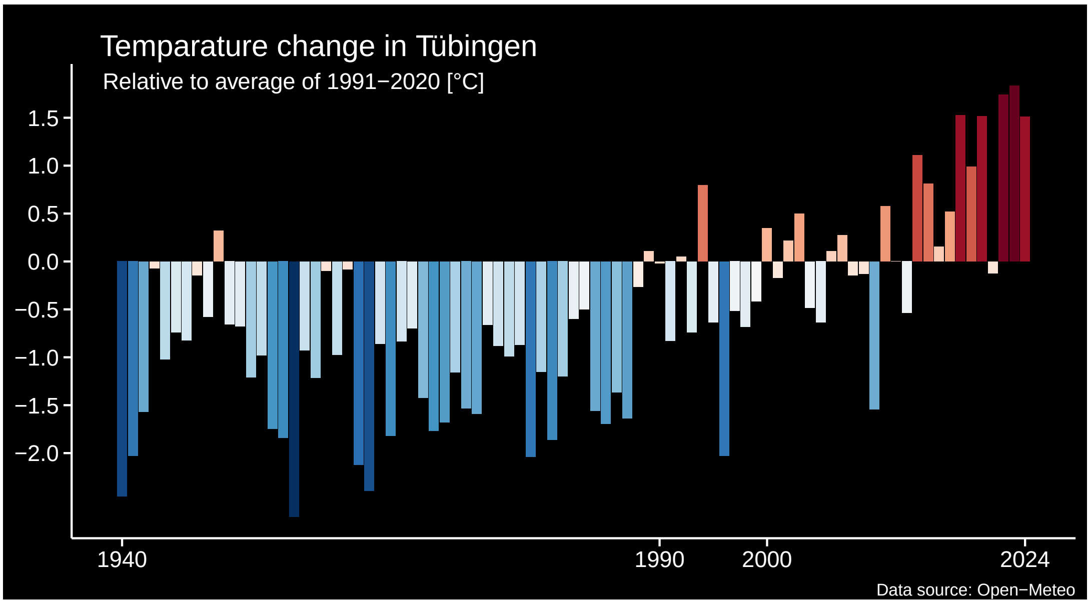
Part 2: The historic weather data API

The Open-Meteo API provides access to historic weather data for a given location in their historical weather API. You can retrieve the data for a specific date range. Your task is to retrieve the complete historic weather data for Tübingen from 1940 to 2024 and then compute yearly average temperatures.

3. Retrieve the historic weather data for Tübingen from 1940 to 2024 and store the data in a data frame.

4. Compute the yearly average temperatures. Print the first five rows of the data frame.

5. Create a "temperature anomaly" plot. It should look as close as possible like the one shown below. The temperature anomaly is the difference between the yearly average temperature and the average temperature over a specific period. Here, we choose the period of 1991–2020 (similar to the one from Our World in Data). The plot should show the temperature anomalies for Tübingen from 1940 to 2024 (with the average temperature of 1991–2020 as the reference).



NOTE: The colormap is used in the plot is RdBu (a diverging colormap suitable for showing temperature anomalies). In Python, this colormap can be accessed as **`RdBu_r`** in **`matplotlib`**. In R, you can use the **`RdBu`** colormap from the **`RColorBrewer`** package.

Web Scraping

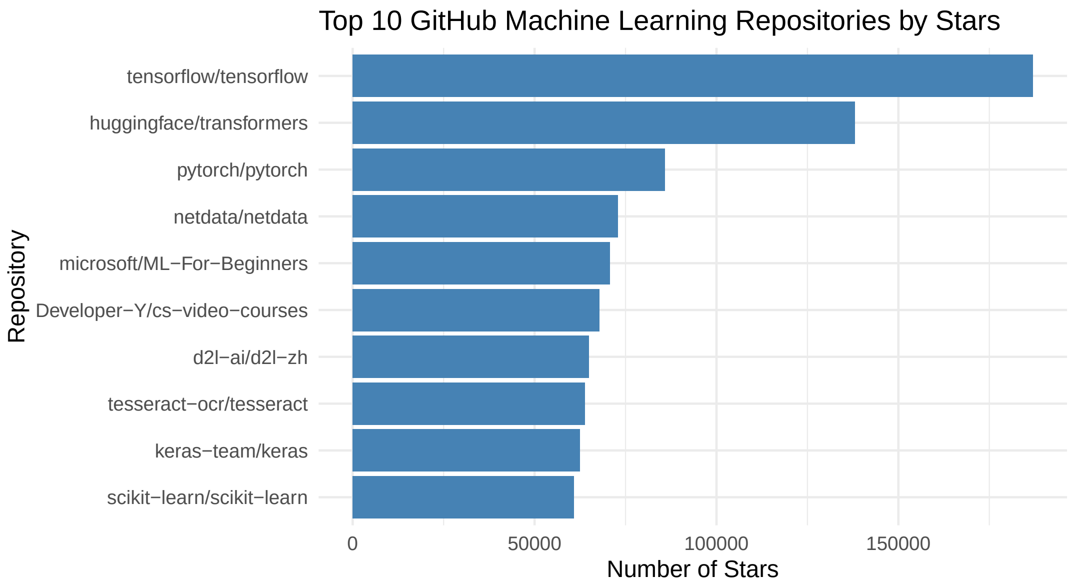
In this part of the assignment, you will scrape data from the GitHub machine-learning topic page. The page contains a list of repositories related to machine learning. Your task is to scrape some information about these repositories.

NOTE: The page uses JavaScript to dynamically load more repositories. It is sufficient to scrape the inital list of repositories (i.e., the first 20 repositories).

Part 3: Scraping the GitHub machine-learning topic page

6. Scrape the repository name, the repository URL, the repository's short description, the number of stars, and the last updated date for the first 20 repositories on the GitHub machine-learning topic page. Put the scraped data into a data frame. Print the entire data frame.

7. Visualize the popularity of these repositories by creating a bar plot showing the number of stars of the top 10 repositories. The plot should look similar to the one shown below.


Top 10 GitHub Machine Learning Repositories by Stars
Repository
tensorflow/tensorflow
huggingface/transformers
pytorch/pytorch
netdata/netdata
microsoft/ML−For−Beginners
Developer−Y/cs−video−courses
d2l−ai/d2l−zh
tesseract−ocr/tesseract
keras−team/keras
scikit−learn/scikit−learn
0
50000
100000
150000
Number of Stars