Project 3: Storytelling with Code and Data

This project only has to be completed by students enrolled in INFOB2PWD & INFOMCTH

In this group project you will work with Jupyter Notebooks and create a computational narrative, that is, tell a “story” with code and data. The project is designed to let you gain experience with loading, cleaning, filtering, analyzing and visualizing data, and other data science practices.

Project 3 needs to be submitted through Blackboard by **March 28, 2025, 11:00**. Make sure to include in the submission the documentation, all your code (as **.ipynb** files), and if applicable any other relevant project files, combined into **one .zip file**.

Documentation

Include a .txt or .pdf document with the following information:

1. Group members: write the name of all the members of your group and the tasks done for each one. Please also include members who did not participate in the project and why (for example, they dropped out of the course). **INFOMCTH students only**: indicate your study programme (opleiding).
2. Project status: write all the project requirements and describe for each requirement if it is completed or not. For example, you can use a numerical scale (0 to 100) or a categorical scale (not done, partially completed, completed, expectations exceeded).
3. Running instructions: include the instructions or steps that are necessary to run the project with Python. This will help us grade your project, so please be very clear and specific. Include any library that is used and is not part of the Python Standard Library.

Assignment

Computational narratives combine text, data and code to tell (often interactive) stories about various topics. Some even regard them as “papers of the future”, as they improve on transparency and reproducibility by design, which is important for Open Science. Here are some examples:

- An open RNA-Seq data analysis pipeline tutorial with an example of reprocessing data from a recent Zika virus study (If you get "Service Unavailable", access the notebook directly here)
- An exploratory statistical analysis of the 2014 World Cup Final (If you get "Service Unavailable", access the notebook directly here)
- An open science approach to a recent false-positive between solar activity and the Indian monsoon (If you get "Service Unavailable", access the notebook directly: here)
- Analysis and visualization of a public OKCupid profile dataset using python and pandas
- Particle-In-Cell Plasma Sim (If you get "Service Unavailable", access the notebook directly: here)
- Visualization: Mapping Global Earthquake Activity (If you get "Service Unavailable", access the notebook directly: here)

Currently, Jupyter Notebooks are the probably most popular format for computational narratives, so we will use them, too. In addition to Python, you will need to “speak” Markdown for nice formatting of the text cells. This cheat sheet summarizes the most important Markdown bits. You might also want to have a look at these coding standards for Jupyter Notebooks.

Your task for this project is to develop a computational narrative around a dataset from the Dutch central statistics office (Centraal Bureau voor de Statistiek, CBS). Start by going to the CBS Open data StatLine portal and choose a dataset. Download the relevant CSV files to work with. (There is also a Python library for accessing CBS open data, but be aware that downloading the data every time you run a script can be slow and cause unnecessary traffic, and of course it requires a stable internet connection.) Then brainstorm about possible analyses that you could do on these data. It helps to think about these in terms of research questions and (statistics) methods that can be used to answer them.

For example, we might have chosen the "ICT knowledge and skills" data set. The data set contains information on the self-reported overall ICT skill levels (no, little, basic, advanced) and the presence of specific ICT-related skills (such as installing software, working with spreadsheet software, or writing computer programs) for different groups of people (such as men/women, age groups, educational levels). The data have been collected annually since 2015. Some possible questions and methods:

- *How do the reported ICT skill levels differ between the groups? Possible methods: (Clustered) bar charts per skill with clusters of four bars (for no/low/medium/high skill levels) for each of the considered groups on the x-axis, and the percentage of the reported skill levels on the y-axis.*
- *How have the reported skill levels changed over time? Possible methods: Line graphs per skill and group with the years on the x-axis, the percentage of the reported skill levels on the y-axis and different colors for the lines to represent the different skill levels.*

Choose at least as many analyses for your narrative as you have members in the group. As this course focuses on how to program things with Python rather than on the data analyses methods themselves, please use methods that you know or understand. If in doubt, keep it simple. The expectation for this project is to use standard descriptive statistics methods. To read up on these, David Lane et al.'s free online book "Introduction to Statistics" is a great reference. In particular, Chapter 2 (Graphing Distributions) and Chapter 3 (Summarizing Distributions) might be helpful to find suitable descriptive statistics methods (and the correct English terms for them). If you are familiar with more advanced methods, for example from your study program, you can also use them. In any case, pay attention that the chosen methods are applicable to the kind of data you have.

When you have chosen a dataset and collected analysis ideas, start sketching your computational narrative and fill the notebook with life. Make sure that your project meets the following requirements:

1. The computational narrative is presented in one Jupyter Notebook. This means, only one Jupyter Notebook per group.
2. Longer pieces of code, especially when not directly meaningful for the narrative (for example details of the data pre-processing when loading the data from the CSV files), are stored in separate .py modules and imported to the notebook.
3. The notebook has different sections, including introductory text with a description of the data set, data loading, the different analyses with explanations, a summary/conclusion, and possible references.
4. **There should be at least as many distinct data analyses as there are group members**. This means each member must contribute one analysis, including a research question, methods, and a description of the results.
5. Ensure that the documentation includes running instructions and specifies any additional libraries used for visualizations in the Jupyter Notebook.

6. **INFOMCTH students only**: You must select a dataset related to your study programme. If your group includes students from different programs, choose just one for the project. Preferably, get the dataset from a repository linked to your field of study (e.g., https://enermaps.openaire.eu/ datasets for Energy Research). Additionally, describe in the notebook how your narrative links to your study (e.g., connect the narrative to a specific project that you have done, to your master thesis, or to an area that you want to work on after your graduation).

Feel free to be creative and add **additional features**. For example, integrate additional datasets in the analysis, have a look into geovisualization or make analyses interactive. Of course, you can easily challenge yourselves more in this project by trying more sophisticated analyses. The Python ecosystem is full of interesting data science libraries, browse the web for inspiration.

Referencing

In this course, we cover basic data analysis techniques and fundamental plots. However, you may need to explore additional types of plots and analyses beyond what was introduced. To ensure transparency and academic integrity, you must properly reference all external sources used for any plot or analysis included in your project.

For each plot, indicate the source that inspired it. Use a standard referencing format (e.g., APA) and provide a brief explanation of how you adapted or modified the code for your specific dataset and research question. The reference must be directly related to the plot or analysis—if a simple example was significantly expanded or modified, provide a justification for how you developed it further.

Valid sources:

- Course Jupyter Notebooks
- Books
- Reputable websites (e.g., documentation pages, academic sources, blogs with clear explanations)
- GenAI tools (e.g., ChatGPT), but only for generating code related to one of the three minimum required plots.

GenAI Usage

You are allowed to use Generative AI (e.g., ChatGPT) only for inspiration on the types of analyses or for generating the code of **one plot** (out of the minimum three required). If you use GenAI, document it explicitly as follows:

- Provide the exact prompt you used
- Include the response (or a summarized version)
- Indicate any modifications you made to adapt the code to your dataset
- Ensure that the interpretation of the results is entirely your own

Failure to properly reference sources may impact your project evaluation.

Grading

Check the project grading rubric. This will give you a clearer idea of what is important to focus on.

A typical submission graded with a 7 / 7.5 will be:

- Fully comply with all the basic requirements of the assignment.
- One **clear** research question per student and **3 basic metrics** (descriptive statistics) for each research question (e.g., averages, means, max).
- All plots/data analyses must contain a valid reference. This includes using GenAI for a maximum of only one plot.

Implementing additional functionalities (e.g., integrate additional datasets in the analysis), an extensive data analysis, demonstrating creativity, a particularly interesting and challenging research question, or analyses beyond descriptive statistics (e.g., regression, correlations, etc.) contribute to a higher grade.

Tips

Ask your tutors for help and feedback early and regularly. Don't wait until you are really stuck somewhere. Often it is difficult to fix programming problems that are caused by poor design decisions made early in the project.

You can share your project files through Blackboard. At some point you might also want to work together on your code during a video call. Four out of several options to do that:

- One of you shares their screen with Spyder and does the editing, while you discuss it together.
- You use a collaborative coding environment like https://repl.it/site/multiplayer, and put the code back to Spyder/Blackboard when you are done there.
- Microsoft's Visual Studio also supports live sharing and collaborative editing of code, see https://visualstudio.microsoft.com/de/services/live-share/ for more information.
- CoCalc https://cocalc.com/doc/jupyter-notebook.html offers services specifically for the collaborative editing of Jupyter notebooks. They offer a free plan, which has some limitations, but is probably sufficient for the project.