SCIENTIFIC COMPUTING

Prof. Sebastián Roldán Vasco

MASTER IN AUTOMATION AND INDUSTRIAL CONTROL - 2024-2

DEADLINE: 5th June 2025, 8 a.m.

Dear student, read the following instructions carefully. Your grade will depend on their fulfillment.

GRADED WORK:

• Written report in GIT-related platform, i.e. GitLab or GitHub

• Data, scripts and functions would be self-contained.

The work must be written entirely in English

Goal:

This lab aims to formalize the research-related codes in a pipeline to get a structured project. Structuring code for a research project in scientific programming involves organizing your files and directories in a way that is logical, maintainable, and conducive to collaboration and reproducibility.

Materials Needed:

- Computer with Python and a text editor/IDE installed.
- Git installed for version control.
- Access to a terminal or command prompt.

Activity:

Using the results of the previous homeworks:

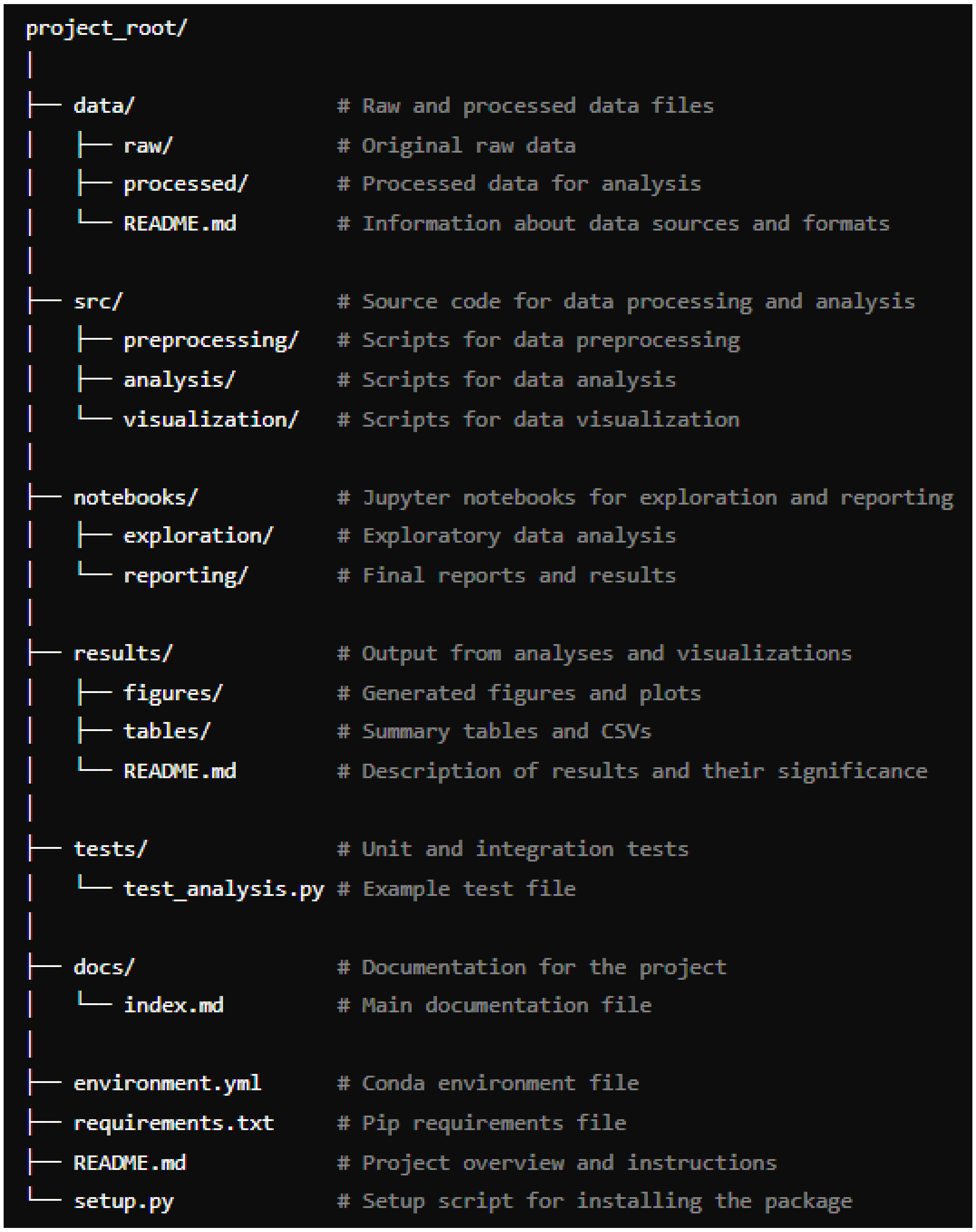
1. (10%) Create a well-organized directory structure with all the files you need to run your data and code. Figure 1 shows an example of a directory.
2. (30%) Divide your codes into three main logical modules and scripts:
   a. Preprocessing: scripts for cleaning, transforming, and preparing data.
   b. Analysis: core analysis scripts, including statistical models, machine learning algorithms, main formulae, etc.
   c. Visualization: scripts for generating plots, charts, and other visual representations of the data

   If you must change this structure to fit your thesis purposes better, feel free to do that. However, you must justify this decision.
3. (25%) Perform an analysis of the computational complexity of all implemented algorithms. Each folder must contain a notebook with this analysis.
4. (15%) Ensure that your project is well-documented:
   a. Make a txt file with the library requirements (see *Figure 2*). If you use Matlab instead of Python, ignore this item (the grade will be computed from the previous point). This file is necessary to avoid errors and continuous exceptions for specific uninstalled libraries. The user can install all your required libraries using `pip` as follows:

   ```
   pip install -r requirements.txt
   ```

   b. README.md[1]: an overview of the project, how to set it up, and how to run the analyses. Create one for each folder and a general one. The latter must have a short theoretical background with references, a brief description of the methodology, a summary of the obtained results, a brief discussion, and conclusions.
   c. Docstrings[2]: use docstrings in your Python or Matlab code to explain what each function and class does. All classes and functions must be documented.

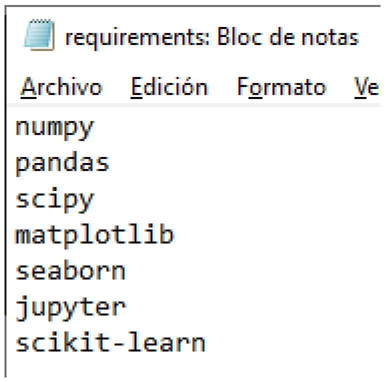
[1] Check the following link to write the file correctly: How to Write a Good README File for Your GitHub Project (freecodecamp.org)
[2] Python Docstrings - GeeksforGeeks



```
project_root/
│
├── data/               # Raw and processed data files
│   ├── raw/            # Original raw data
│   ├── processed/      # Processed data for analysis
│   └── README.md       # Information about data sources and formats
│
├── src/                # Source code for data processing and analysis
│   ├── preprocessing/  # Scripts for data preprocessing
│   ├── analysis/       # Scripts for data analysis
│   └── visualization/  # Scripts for data visualization
│
├── notebooks/          # Jupyter notebooks for exploration and reporting
│   ├── exploration/    # Exploratory data analysis
│   └── reporting/      # Final reports and results
│
├── results/            # Output from analyses and visualizations
│   ├── figures/        # Generated figures and plots
│   ├── tables/         # Summary tables and CSVs
│   └── README.md       # Description of results and their significance
│
├── tests/              # Unit and integration tests
│   └── test_analysis.py # Example test file
│
├── docs/               # Documentation for the project
│   └── index.md        # Main documentation file
│
├── environment.yml     # Conda environment file
├── requirements.txt    # Pip requirements file
├── README.md           # Project overview and instructions
└── setup.py            # Setup script for installing the package
```

Figure 1. Typical layout of a directory structure

d. Notebooks: use Jupyter notebooks or Matlab live scritps for detailed explanations and exploratory analysis. These files are intended to explain results rather than run the algorithms.



```
numpy
pandas
scipy
matplotlib
seaborn
jupyter
scikit-learn
```

Figure 2. Example of a requirement text file namely “requirements.txt”

5. (20%) Provide an example code to figure out the general behavior of your algorithms. In this code, you must show the main aspects of all implemented algorithms. It could be a notebook.