Graded Practice 4

Astroinformatics I

Name: José Luis Ricra Mayorca
Date: Jul 10, 2025

Task 1

To create a GitHub repository, Tutorial 8 [Hernitschek, 2025a] was used as a reference, and the following steps were performed:

```
# Create account
The first step was to create a GitHub account by visiting https://github.com
Then, the "Signup" option was selected, and a username, email, and password were entered.

# Create Personal Access Token
Go to https://github.com/settings/tokens
Click on "Generate new token"
Set expiration time to 30 days and select the scope (repo)

# Install Git on Ubuntu, in the terminal:
sudo apt update
sudo apt install git

# Verify installation
git --version

# Initial configuration
git config --global user.name "josericraastro"
git config --global user.email "jose.ricra@uamail.cl"

# A folder was created for the project
mkdir astroinformatics-practices-jose
cd astroinformatics-practices-jose
git init

# Practices 1, 2, and 3 were manually placed into this folder.

# A file named README.md was created (its content will be shown in Task 2).
nano README.md

# A commit was made, the folder was linked to the repository, and files were uploaded
git add .
git commit -m "Initial commit: uploaded graded practices 1 to 4"
git remote add origin https://github.com/josericraastro/astroinformatics-practices-jose.git
git push -u origin main
```

The link to the repository is: https://github.com/josericraastro/astroinformatics-practices-jose

Figure 1 shows the files uploaded to the GitHub repository.

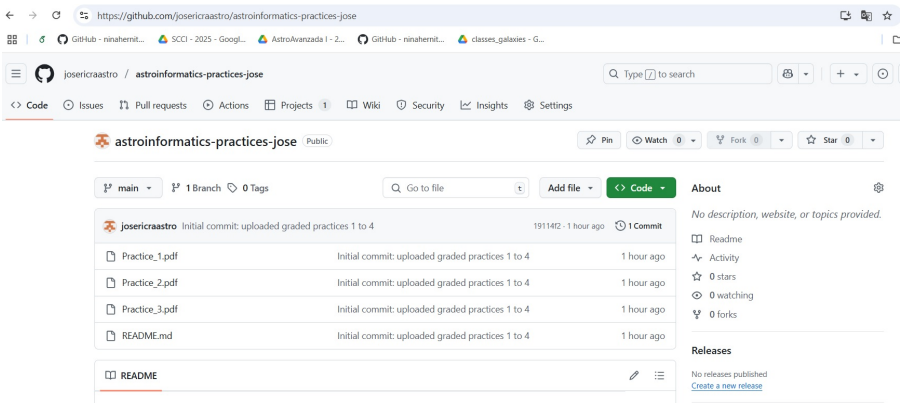

Figure 1: Created repository.

Task 2

A text file was written containing basic information on how the TESS light curves were processed. For this, the procedures followed in each graded practice were taken into account. Figure 2 shows the created `README.md` file.

```
# Astroinformatics I — Graded Practices

## Author
José Luis Ricra Mayorca

## Overview
This repository includes my graded practices for the Astroinformatics I course (Semester 1,
2025) at Universidad de Antofagasta. This repository shows the steps to download TESS data,
assign the most appropriate format for data processing, graph the data, detect outliers, and
obtain basic statistics for each light curve.

## Contents
- Practice_1.pdf: Download and convert TESS light curves
- Files_practice_1: Folder with the files used in that practice 1.
- Practice_2.pdf: Preprocess files and classify stellar temperatures
- Files_practice_2: Folder with the files used in that practice 2.
- Practice_3.pdf: Plot light curves, detect outliers, and compute statistics
- Files_practice_3: Folder with the files used in that practice 3.
- Practice_4.pdf: GitHub submission, documentation, and test cases
- Files_practice_4: Folder with the files used in that practice 4.

## File Descriptions
For each practice there is a folder with the files used in that practice (mainly scripts).

Files_practice_1

    tesscurl_sector_73_lc.sh: Shell script to download light curves in FITS format

    lista.sh: Shell script to create a text file with the list of all files in CSV format

    dividir.sh: Shell script to split the contents of the csv_files.txt file into three text
files

Files_practice_2

    spectral.py: Python script that takes the surface temperature of a star and returns the
spectral type

    jd.py: Python script that takes the day, month, and year to calculate the Julian day

Files_practice_3

    script_data.sh: Shell script to convert CSV data to LC format, extracting the time, flux,
and flux error columns, and replacing the comma delimiter with spaces.

    plot.py: Plots light curves for visual inspection and comparison

    outliers.py: Identifies and flags outliers in flux values

    stat.py: Computes basic statistics (mean, std, min, max, etc.) from the light curve data.

Files_practice_4

    test_cases.md: Describes test cases for each processing step to ensure reproducibility

    README.md: Description of the contents of the repository.

## How to Reproduce
You need:
- Python 3.10+
- NumPy and Matplotlib
- TOPCAT (for initial FITS → CSV conversion)

## License
This repository is for academic purposes only.
```

Figure 2: Contents of the README file.

The file can be viewed at: https://github.com/josericraastro/astroinformatics-practices-jose/blob/main/README.md

Task 3

This section outlines test cases designed to validate the individual steps involved in the processing of TESS light curve data, as implemented across Graded Practices 1–3. To do this, the recommendations of Lecture 8, Software Projects [Hernitschek, 2025b] were taken into account.

```
# Test Cases for Processing TESS Light Curves

This section outlines test cases designed to validate the individual steps involved in the
processing of TESS light curve data, as implemented across Graded Practices 1-3.

---

## Test Case 1 (Practice 1): FITS to CSV Conversion

**Objective:**
Verify that TESS FITS files are correctly converted to CSV format using TOPCAT.

**Input:**
A TESS FITS file (e.g. ‘tess2023341045131-s0073-0000000001750268-0268-s_lc.fits‘)

**Procedure:**
Open the FITS file in TOPCAT → Export as CSV

**Expected Output:**
A CSV file containing columns such as ‘TIME‘, ‘PDCSAP_FLUX‘, ‘PDCSAP_FLUX_ERR‘, and other
relevant metadata.

**Pass Criteria:**
- The number of rows remains unchanged
- Data types and column names are preserved
- File is readable by other processing tools

---

## Test Case 2 (Practice 3): Extract relevant columns from CSV data

**Objective:**
Verify that only the relevant light curve columns (TIME, FLUX and FLUX ERROR) are extracted,
replaces commas with whitespace accurately

**Input:**
Files ‘data*.csv‘ containing lines such as:
‘TIME,...,PDCSAP_FLUX,PDCSAP_FLUX_ERR, ...‘

**Procedure:**
Run: ‘script_data.sh‘

**Expected Output:**
Files ‘col_data*_flux.lc‘ containing exactly three columns: ‘TIME‘, ‘PDCSAP_FLUX‘ and
‘PDCSAP_FLUX_ERR‘ separated by spaces.

**Pass Criteria:**
- All commas are replaced with single spaces
- Output file contains three columns (TIME, FLUX and FLUX ERROR)
- The format of the files is changed from csv to lc

---

## Test Case 3 (Practice 3): Plotting a Light Curve
```

Objective:
Confirm that the plotting script produces clear and accurate flux vs. time graphs.

Input:
A ‘.lc‘ file with three numerical columns: ‘TIME‘, ‘FLUX‘ and ‘FLUX ERROR‘

Procedure:
Run: ‘plot.py‘ using Matplotlib

Expected Output:
A plot saved as ‘.pdf‘ showing a labeled time-series of the light curve

Pass Criteria:
- Axes are labeled correctly
- Data points match original flux values
- No runtime errors occur

Test Case 4 (Practice 3): Outlier Detection

Objective:
Validate the outlier detection mechanism based on the 4σ-from-median rule.

Input:
A ‘.lc‘ file containing at least one anomalous data point

Procedure:
Run: ‘outliers.py‘

Expected Output:
A plot highlighting outliers (e.g., using red dots or markers)

Pass Criteria:
- Outliers beyond 4σ from the median are correctly flagged
- Normal data points are not falsely flagged
- Plot clearly distinguishes between normal and anomalous data

---

Test Case 5 (Practice 3): Statistical Summary

Objective:
Ensure that basic statistics are correctly calculated and reported.

Input:
A ‘.lc‘ file with valid flux values

Procedure:
Run: ‘stat.py‘

Expected Output:
Printed or saved values for:
- Minimum
- Maximum
- Mean
- Median
- Standard deviation
- Amplitude

```
**Pass Criteria:**
- Computed values match NumPy or manual verification
- Output is clear and correctly formatted

---
```

The file `test_cases.md` can be viewed at: https://github.com/josericraastro/astroinformatics-practices-jose/blob/main/Files_practice_4/test_cases.md

References

N. Hernitschek. Tutorial 8: Github. https://github.com/ninahernitschek/astroinformatica_I_2025_1/blob/main/tutorial8.pdff, 2025a. PDF available on GitHub.

N. Hernitschek. Lecture 8: Software projects. https://github.com/ninahernitschek/astroinformatica_I_2025_1/blob/main/astroinformaticaI_lecture8_softwareprojects.pdf, 2025b. PDF available on GitHub.