Practice 4

Astroinformatics I
Semester 1, 2025

Ana Pereira Aguirre

July 14, 2025

1 Introduction

This report contains the solutions to Practice 4, whose tasks consist of organizing the code developed in the previous practices into a GitHub repository, documenting the data processing steps, and identifying test cases to ensure the robustness of the Python scripts. All code and explanations are developed below.

2 GitHub Repository Creation

A GitHub repository was created where all the scripts, graphics, and documentation developed in the previous practices are stored. The structure is organized by folders, with one directory per practice. github.com/anapereira-ast/Astroinformatics_Practices
Each folder includes:

- Scripts and code.
- Light curve files.
- README documentation.
- Plots of the light curves.

3 Documentation of TESS Light Curve Processing

The processing of the light curves is performed by following the steps below:

1. First you must download and execute the `requirements.txt` file, where all the packages to be used in the PYTHON scripts are located.
2. Data acquisition: The .fits files of light curves of TESS Sector 73 are downloaded through the script `tesscurl_sector_73.sh`
3. Format conversion (with TOPCAT): The `.fits` files must be converted to `.csv` format using TOPCAT. A step-by-step guide is available in the PDF: `astroinformatics_practiceI_apereira.pdf`. This process results in 20 light curve files in `.fits` format.
4. Format conversion (with BASH): This same process can be done through bash scripts and these scripts can be found in practice 2 and the pdf where the step by step of these tasks is explained.

5. Plot the light curves: The next step is to plot these light curves. To better manipulate these files, it is possible to extract only the columns of interest, which are `TIME`, `PDCSAP_FLUX`, and `PDCSAP_FLUX_ERR`, and the script that performs this task is in the folder of practice 2.

   To plot, if you want to do it in TOPCAT, in the pdf of practice 1 you will find the steps to follow. In case you want to do it through a PYTHON script, inside the folder practice 3 you will find the jupyter notebook that develops this task in addition to the identification of outliers and the light curves plotted in a `.pdf` file in practice 1 or `.jpg` file in the folder `LC_PLOT`.

4 Identification of Test Cases

When processing TESS light curves, it is important to identify and anticipate common issues that can affect data quality or cause the code to fail. Below are a few critical edge cases that were considered during development:

1. Missing or unexpected column names:
   The script expects the input `.csv` files to include specific TESS columns such as `TIME`, `PDCSAP_FLUX`, and `PDCSAP_FLUX_ERR`. If any of these are missing—due to formatting issues or because the file comes from another telescope with a different structure the code will raise a `KeyError`.

   Possible consequence: The script crashes without generating any plot.

   Tip: To make the code compatible with other instruments, adjust the column names accordingly or implement a column mapping step.

2. Filename pattern does not match expected format:
   The script extracts an identifier from the filename using a regular expression based on the format `tess_lc_123456.csv`. If a file does not match this naming pattern (e.g., `lightcurve_final.csv`), the expression fails and an `AttributeError` is raised when trying to access group(1) from a None result.

   Possible consequence: The plot title fails to render correctly, or the script stops with an error.

3. Empty DataFrame after NaN filtering:
   If a file contains too many `NaN` values in key columns, the DataFrame becomes empty after applying `dropna()`. Attempting to plot or process an empty DataFrame can result

in silent failures or generate blank output files.

Possible consequence: A blank figure is saved, or runtime warnings/errors are triggered during further analysis.

Tip: Include a condition to skip files with empty DataFrames after cleaning.