Astroinformatics I

Project Presentation

José Batista-Mendoza

CITEVA, Universidad de Antofagasta

July 10, 2025

Table of Contents

Introduction

Motivation:

- In modern astronomy, theoretical knowledge must be coupled with strong computational skills.
- Astroinformatics is precisely this intersection: applying informatics tools to astronomical data.
- Bridge the gap between classroom concepts and their tangible application in real-world data handling and analysis.
- Show how the fundamental computational principles provided in the course become powerful instruments for astronomical discovery.

Introduction (Cont.)

Course Overview:

- Astroinformatics I covered a comprehensive range of essential computational topics.
- From foundational Linux command-line operations and shellfor data management,
- Through core Python programming for data manipulation and analysis,
- To specialized astronomical libraries (`astropy`, `pandas`, `numpy`, `matplotlib`) for specific data challenges.
- Finally, integrating crucial software engineering principles for robust and reproducible work.
- The course emphasized hands-on learning through lectures, tutorials, and graded practices.

Introduction (Cont.)

Presentation Goal:

- The primary goal is to illustrate how the theoretical and practical concepts from Lectures and Tutorials were directly applied.
- This presentation will walk through key examples from the **graded practices (Practices 1–4)**.
- This will highlight the practical implementation of tasks like data cleaning, transformation, analysis, visualization, and project organization.
- Ultimately, showcasing a complete workflow for astronomical data processing.

Shell, Linux & Python Fundamentals

Foundation: Based on Lectures and Tutorials 2–5

Key Concepts Applied:

- **File System Navigation & Manipulation**
  - Identifying files (`ls`, wildcards like `*.csv`).
  - Directory management (`dirname`, variable expansion).
  - File creation (`touch`).
- **Text Processing & Data Transformation**
  - Changing delimiters (`sed s/,/ /g`).
  - Removing columns (`awk '{print $1, $2, $3}'`).
  - File renaming (`mv`).

Shell, Linux & Python Fundamentals (Cont.)

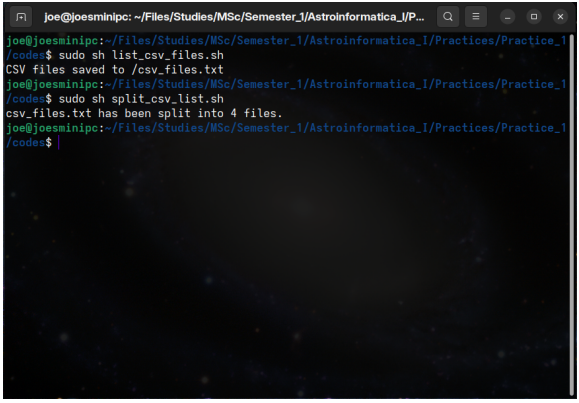
Foundation: Based on Lectures and Tutorials 2–5

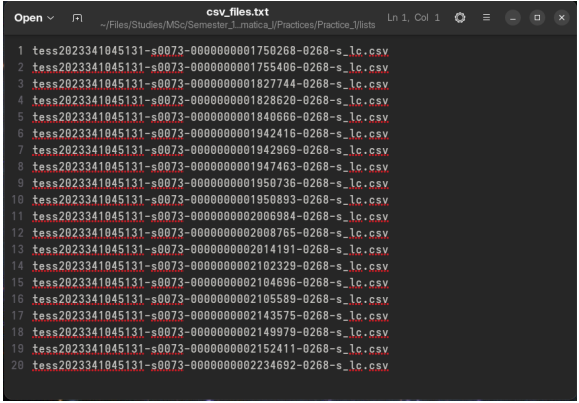
Key Concepts Applied:

- **Shell Scripting Constructs**
  - `for` loops for batch processing.
  - Conditional statements (`[ -e "$csv_file" ]`, `if`).
  - Output Redirection (>>).
- **Python Fundamentals**
  - Defining and calling functions.
  - Performing basic operations, comparisons, data type conversion and string manipulation.
  - Applying conditional logic (`if-elif-else`).
  - Handling command-line arguments (`sys.argv`).
  - Incorporating user interaction (`input()`).
  - Implementing error handling (`try-except` blocks).

Practice Showcase

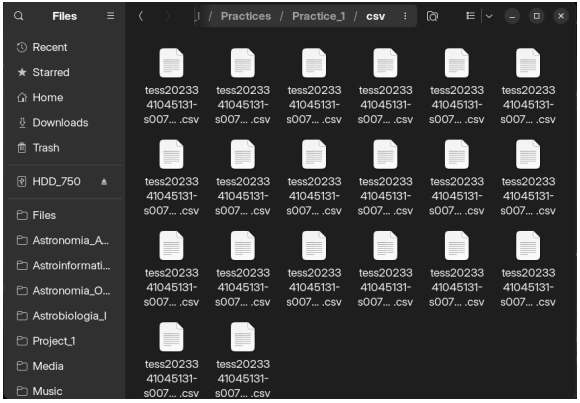
- **Practice 1:** Listing CSV filenames

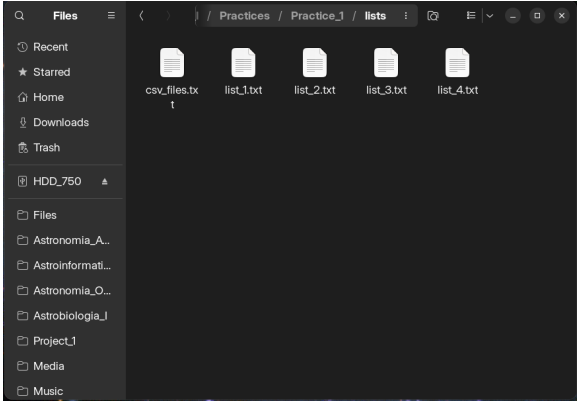




Practice Showcase (Cont.)

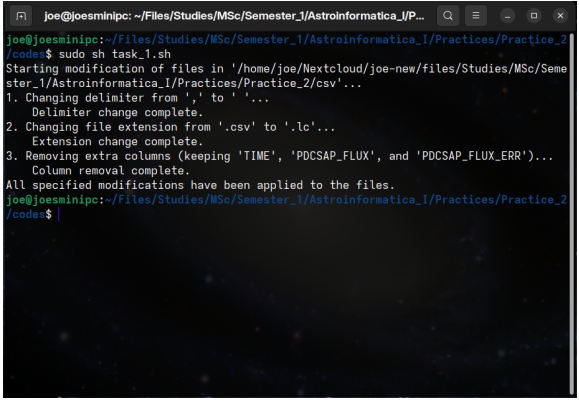
- **Practice 1:** Splitting text files

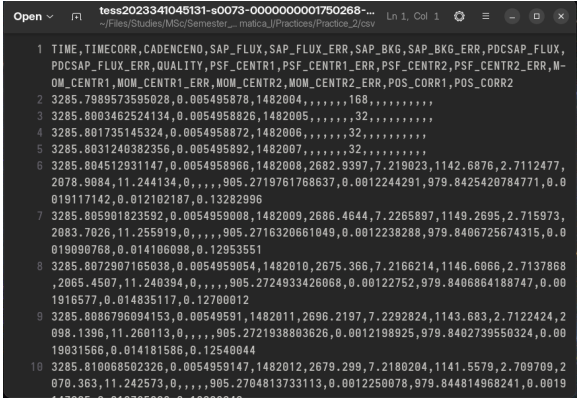




Practice Showcase (Cont.)

- **Practice 2:** Modifying CSV files (delimiter change, column removal, and extension change)





Practice Showcase (Cont.)

- **Practice 2:** Modifying CSV files (Cont.) and basic Python scripting

Open ~/Files/Studies/MSc/Semester_... matica_I/Practices/Practice_2/csv tess2023341045131-s0073-0000000001750268-... Ln 1, Col 1

```
TIME PDCSAP_FLUX PDCSAP_FLUX_ERR
3285.7989573595028
3285.8003462524134
3285.801735145324
3285.8031240382356
3285.804512931147 2078.9084 11.244134
3285.805901823592 2083.7026 11.255919
3285.8072907165038 2065.4507 11.240394
3285.8086796094153 2098.1396 11.260113
3285.810068502326 2070.363 11.242573
3285.8114573952375 2082.0286 11.252562
3285.8128462881491 2079.1445 11.240101
3285.814235180594 2065.8347 11.235728
3285.8156240735057 2084.989 11.237153
3285.8170129664168 2074.002 11.242372
3285.8184018593283 2076.244 11.235358
3285.8197907522399 2055.999 11.213792
3285.8211796451506 2085.9194 11.243473
3285.8225685380622 2100.8157 11.248517
3285.8239574305076 2098.8606 11.244343
3285.8253463234193 2110.7817 11.246265
3285.8267352163302 2102.7703 11.248538
```

joe@joesminipc: ~/Files/Studies/MSc/Semester_1/Astroinformatica_I/P...

```
joe@joesminipc:~/Files/Studies/MSc/Semester_1/Astroinformatica_I/Practices/Practice_2/codes$ source /home/joe/Files/Studies/MSc/Semester_1/Astroinformatica_I/.venv-linux/bin/activate
(.venv-linux) joe@joesminipc:~/Files/Studies/MSc/Semester_1/Astroinformatica_I/Practices/Practice_2/codes$ python3 task_3.py
-------------------------# Julian Date Calculator #--------------------------

Enter the date in the 'd/m/y' format:
Date: 08/06/2008

Your date is 08/06/2008 (8 June, 2008). Is this correct? (Y/n) Y

The Julian date for 08/06/2008 is 2454625
(.venv-linux) joe@joesminipc:~/Files/Studies/MSc/Semester_1/Astroinformatica_I/Practices/Practice_2/codes$
```

Python for Astronomical Data Analysis

Foundation: Based on Lectures and Tutorials 6–7

Key Concepts Applied:

- **Data Manipulation (`pandas`, `numpy`)**
  - Loading and structuring data with `pandas.DataFrame`.
  - Numerical operations and statistical analysis (`numpy` for mean, std, median, etc.).
- **Astronomical Data Handling (`astropy`)**
  - `astropy.timeseries.TimeSeries` for light curves.
  - Time conversion (`astropy.time.Time`, BTJD to JD TDB, BJD to UTC).
  - Lomb-Scargle periodograms (`astropy.timeseries.LombScargle`).

Python for Astronomical Data Analysis (Cont.)

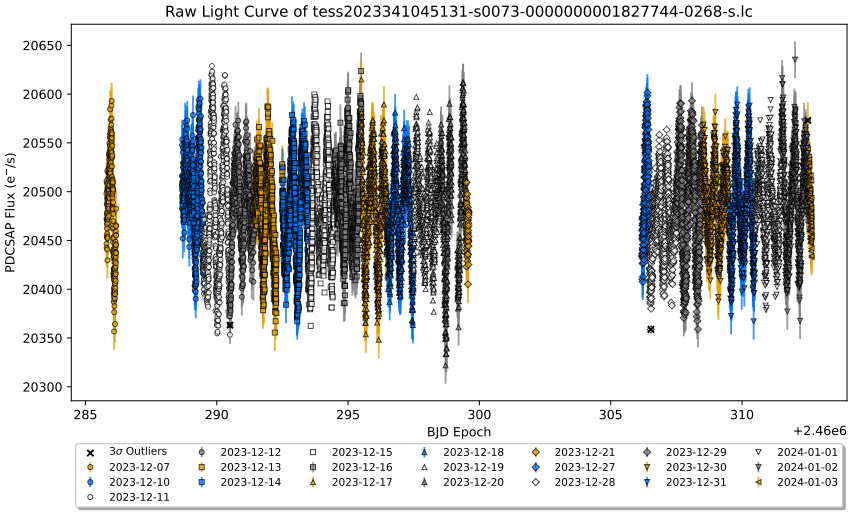
Foundation: Based on Lectures and Tutorials 6–7

Key Concepts Applied:

- **Data Visualization (`matplotlib`)**
  - Generating scatter plots with error bars.
  - Plot customization (titles, labels, legends, colorblind-friendly palettes, markers).
  - Subplots for combined visualizations (light curve + periodogram).
  - Annotations on plots.
- **Astronomical Concepts:**
  - Outlier detection, phase-folding and period finding for TESS light curves.

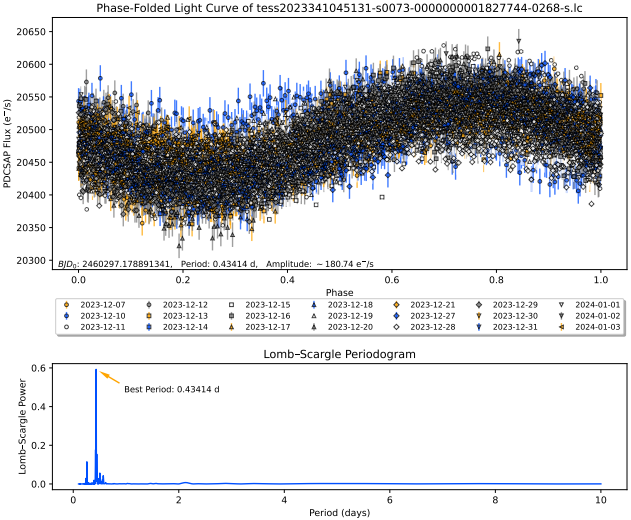
Practice Showcase

- **Practice 3:** Processing TESS light curves (raw plotting and outlier detection)



Practice Showcase (Cont.)

- **Practice 3:** Processing TESS light curves (phase-folding and period finding)



Software Project Management & Quality

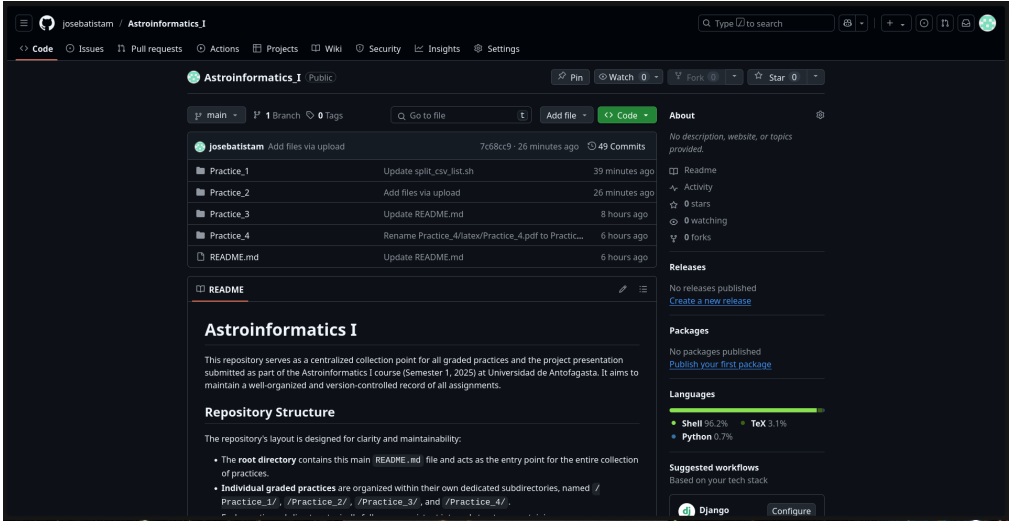
Foundation: Based on Lecture and Tutorial 8

Key Concepts Applied:

- **Version Control (Git & GitHub)**
  - Creating and structuring GitHub repositories.
  - Organizing project files (`/codes`, `/latex`, `/csv`, etc.).
  - Implied Git workflow (`git add`, `git commit`, `git push`).
- **Documentation**
  - Importance of basic documentation for code and workflows.
  - Role of `README.md` for project overview.
- **Software Testing**
  - Identifying test cases for functions (e.g., `get_lc_data`, `fold_lc`).
  - Considering test scenarios (valid data, missing data, known periodicity, no periodicity).

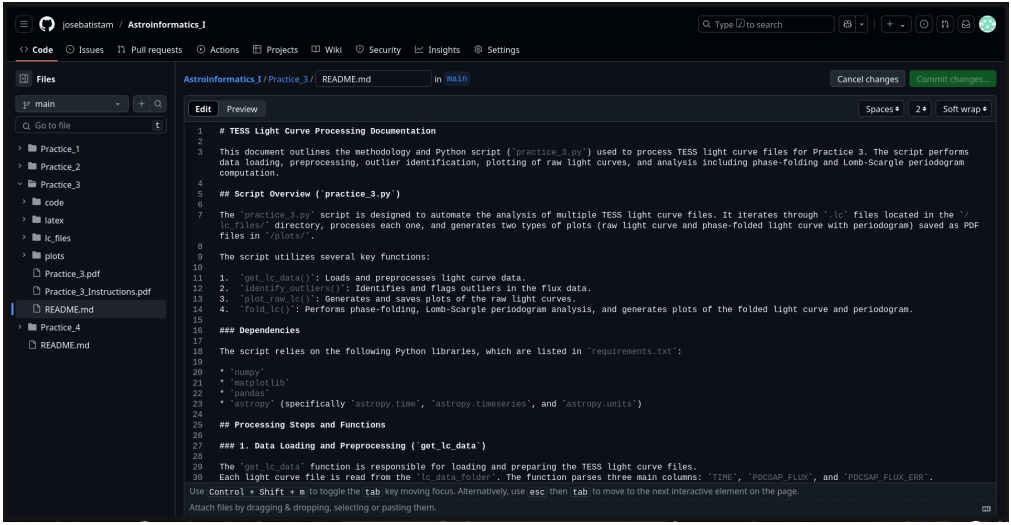
Practice Showcase

- **Practice 4:** Managing a project (GitHub repository set up)



Practice Showcase (Cont.)

- **Practice 4:** Managing a project (GitHub documentation creation)



Conclusions

- The course succeeds in reinforcing the **interconnectedness** of seemingly disparate tools and principles.
- Linux/Shell provides the essential command-line foundation for data wrangling, file management, and automating initial steps. It's the 'glue' that orchestrates tasks.
- Python serves as the primary computational engine, allowing for sophisticated data analysis, algorithm implementation, and statistical modeling.

Conclusions (Cont.)

- Astronomical Libraries (Astropy, Lightkurve, etc.) bridge general-purpose Python with domain-specific astronomical knowledge, providing robust tools for units, time systems, and specialized analysis (e.g., periodograms, phase-folding).
- Software Engineering Principles (Version Control with Git/GitHub, Documentation, Testing) are paramount for ensuring that the code is **reproducible, maintainable, scalable, and collaborative**. They transform individual scripts into robust scientific tools.
- Together, these elements form a powerful and synergistic toolkit, essential for navigating the complexities of modern astronomical data.