COM-480 - Data visualization

Milestone 2 : AstroTrio Group

Authors:

Iman Attia
Léo Carron
Maud Dupont-Roc

Date: April 18, 2025

Website available here

1 Introduction

In this Milestone we continue the work from Milestone 1 with a space themed visualization website for educational purposes. After a closer look to the data, we had to drop the synthetic dataset that contained too much unreliable information, and refine our data story in a structured way. We introduce three new datasets: a planet dataset[1], solar system body data[2], and a NASA exoplanet database[3]. We explore them in *docs/static/data/EDA_Milestone2.ipynb*.

Our visualizations are on 3 different scales: the Earth, the Solar System and the Milky Way. On the Earth, we investigate rocket launches since 1957, on a map. On the Solar System, we do general visualizations on the different bodies. On the Milky Way, we explore the planets that were discovered by Nasa since 1992.

2 Visualizations

2.1 Implemented visualization - Launch Map (Click here)

We present an interactive world map that visualizes all rocket launches from Earth since 1957. The map dynamically evolves over time, allowing users to explore how launch activity has changed year by year. It also includes interactive controls to filter launches by success or failure and to observe trends across different countries.

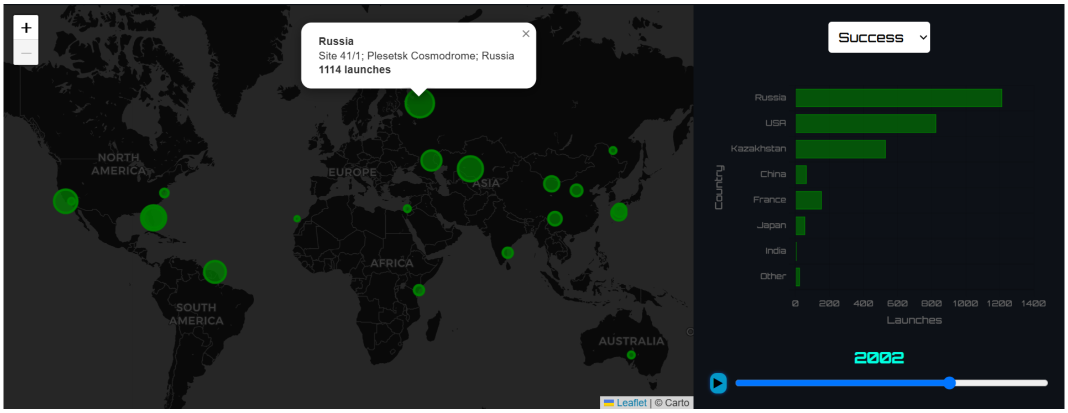


Figure 1: Interactive map showing global rocket launches since 1957. The user can explore launches year by year using the time slider, which can be played automatically with the play button to its left. A menu allows filtering by success or failure. On the right side, there is also a horizontal bar chart displaying the top countries by number of launches for the selected year and status.

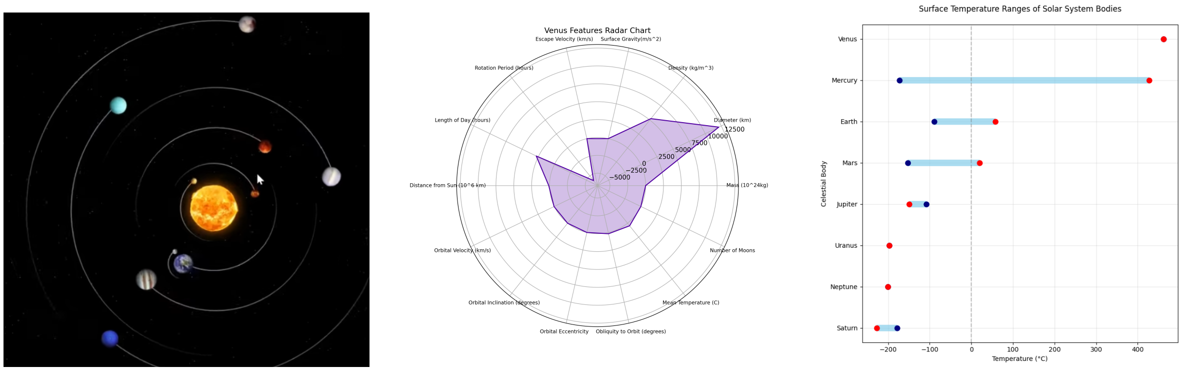
Additional ideas Further improvements could include displaying mission objectives (e.g., satellite, crewed, scientific) and integrating estimated CO_2 emissions per launch based on available rocket specifications. A timeline heatmap could show launch intensity over the years per country, while a comparative view would allow users to explore differences in frequency, success rates, or mission types. Key historical events like the Apollo 11 launch could also be highlighted as interactive annotations.

2.2 Statistics about the solar system

We take inspiration from Nasa's interactive Solar System visualizations [4], and aim to display comparative data of the solar system bodies in the scope of the educational purpose of our project. We visualize the Solar System with:

[1] https://www.kaggle.com/datasets/iamsouravbanerjee/planet-dataset
[2] https://www.kaggle.com/datasets/jaredsavage/solar-system-major-bodies-data
[3] https://www.kaggle.com/datasets/adityamishraml/nasaexoplanets/data
[4] https://eyes.nasa.gov/apps/solar-system/#/home

1. A dynamical solar system representation with the main bodies going around. Planets are selectable to display numerical information in a radar chart and categorical features such as Surface features, Color, and Composition.
2. General visualizations comparing values across the planets and other bodies, such as Temperature ranges and Other numerical information displayed individually, in dynamical time series plots.



(a) Simple solar system html from https://github.com/CogniSolver/solar-system

(b) Example of a radar chart. The features will need further selection and adapted scaling.

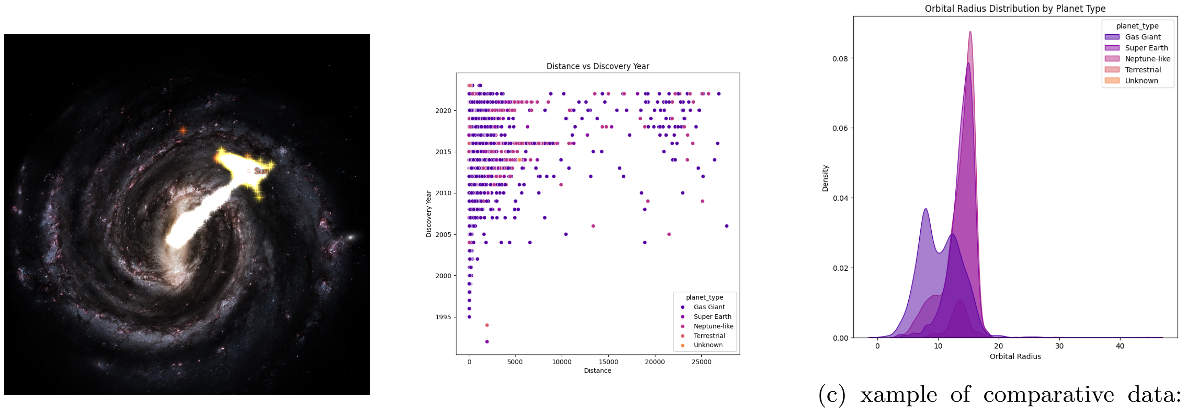
(c) Example of comparative data: Surface temperature ranges of solar system bodies.

Figure 2: Solar system visualizations: orbit map, radar chart and temperature ranges.

Additional ideas We could enhance the solar system view with elliptical orbits instead of circular ones, and include a 3D interactive version for better spatial understanding. Additional layers could show major moons and provide comparable stats as for planets. Artificial objects like probes (Voyager, Pioneer) could be visualized to illustrate the scale of human exploration. Planetary rotation and day/night cycles could also be animated for added realism.

2.3 Exoplanets discovered in the Milky Way

We take inspiration from Nasa's interactive Exoplanet visualizations [5], and aim to display the dynamics behind exoplanets discoveries. We visualize the Milky Way with a map, and make it dynamical by displaying and highlighting over time the exoplanets being discovered. Additionally, we will have interactive panels exploring the different properties of the exoplanets.



(a) Milky Way from https://eyes.nasa.gov/apps/exo/#/

(b) Exoplanets: Distance vs. Discovery Year

(c) xample of comparative data: Orbital radius distribution by planet type.

Figure 3: Milky Way and exoplanet discoveries: visualization, discovery timeline, and orbital distribution.

[5] https://eyes.nasa.gov/apps/exo/#/

Additional ideas The exoplanet section could benefit from a filterable interface based on temperature, radius, or habitability metrics. It could be extended with discovery method breakdowns (e.g., transit, radial velocity) and a dynamic diagram linking exoplanets to their host stars. A 3D star map with hover-based info panels, and visualization of estimated Earth Similarity Index, would make the discoveries more tangible and informative.

3 Tools

For maps, we plan to use Leaflet.js combined with D3 overlays, following the techniques introduced in Lecture 8. More specifically for the launch map, we will create a proportional symbol map on a Mercator projection

For temporal and distribution visualizations such as histograms and time series (e.g., rocket launches over time, exoplanet discovery timelines), we will use D3.js following techniques from Lectures 2, 3, and 4. These visualizations will include animated transitions and interactive filters based on time or categories.

Finally, the project will be deployed on GitHub Pages using modular HTML/CSS/JavaScript. Data is preprocessed in Python using Jupyter Notebooks before being passed to the front-end.

4 Website deployment

We used Python Flask web framework work to structure our website and easily test it locally on our machines. But, for website deployment, we used Github Pages. Thus, we wrote a Python script gen_static_pages.py that generates static HTML pages that can be supported by Github Pages.