INFO 511 Foundations of Data Science – Fall 2025
Milestone 2 – project proposal
Nathan Herling
submitted: Tuesday-November-04-2025

Introduction, motivation, feasibility:
NASA's **OSIRIS-Rex** mission to the asteroid **Bennu**—including the detection of **organic** (prebiotic) molecules—has sparked widespread excitement about human space exploration and our place in the universe. Building on that motivation, I propose to examine whether there is a **linear association** between **calendar year** and the **annual number of "Found" meteorites** recorded on Earth. In other words: *Do yearly counts of found meteorites increase, decrease, or remain stable over time?* The dataset is **publicly available** via NASA/Meteoritical Society sources and has already been downloaded, making the project feasible within the course time-frame.

Framework, theory, guiding principles:
Scientific backdrop. The OSIRIS-REx mission to **Bennu**—and the detection of organic compounds in the returned samples—motivates a broader, empirical question about small-body material reaching Earth. Rather than speculating about origins, this project focuses on an **observable footprint**: the temporal pattern of **found** meteorites.

Team/Individual:
I will be working on my own.

IRB statement:
This data is publicly available and does not contain any personal information on human subjects. Therefore no IRB statement is necessary.

Research Question:
Is there a linear association between calendar year and the annual number of "Found" meteorites recorded in the Meteorite Landings dataset?

Mathematical model:

Fit an OLS with year as the sole predictor of the annual count:

$$Y_t = \beta_0 + \beta_1 \cdot \text{Year}_t + \varepsilon_t \quad (1)$$

Interpretation:

- Year_t: predictor, year (e.g., 1900)
- Y_t: outcome (Count of meteorites, 'fall==Found' in the dataset).
- β_0: expected change in the number of found meteorites per 1-year increase.
- β_1: expected count when Year = 0.
- ε_t: error term.

Hypothesis [Pearson correlation]:

- **H_0:** $\rho = 0$ (no linear association between year found and meteorite counts).
- **H_1:** $\rho \neq 0$ (a non-zero linear association exists).

Compute r and its p-value.

If $p < \alpha$ (e.g., 0.05): reject H_0 → evidence of a linear trend (positive or negative).
Report r, p, and a confidence interval for r.

Citation Style:
IEEE (Institute of Electrical and Electronics Engineers)

Dataset used – provenance:
https://catalog.data.gov/dataset/meteorite-landings

Dataset – type, variables, features:

Meteorite Landing Features

Columns analyzed: **10**

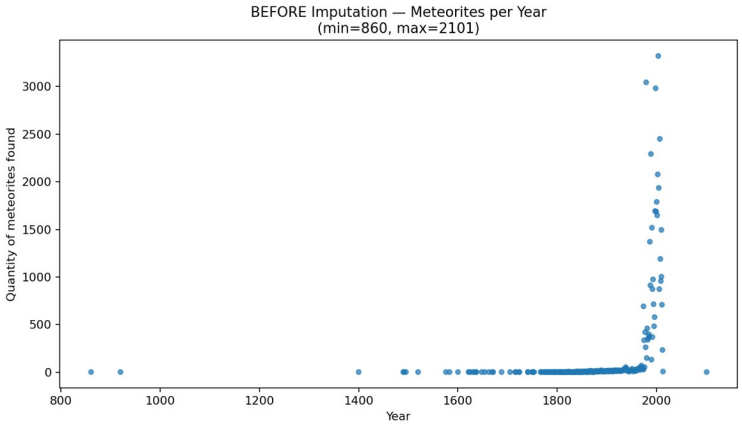
Column	Pandas dtype	Type (mapped)	# Unique	% Missing	Description
GeoLocation	object	categorical	17100	16.00%	String version of the site location, typically '(lat, lon)'; may be missing or imprecise for older found records.
fall	object	categorical	2	0.00%	Event status: 'Fell' (observed fall) or 'Found' (recovered without a witnessed fall).
name	object	categorical	45716	0.00%	Catalog name of the meteorite specimen (may include locality or sequence numbers).
nametype	object	categorical	2	0.00%	Name status used by the Meteoritical Society, usually 'Valid' or 'Relict'.
recclass	object	categorical	466	0.00%	Official meteorite classification (e.g., H5, L6, Iron), indicating composition and petrologic type.
id	int64	numerical	45716	0.00%	Unique catalog identifier for the meteorite record.
mass (g)	float64	numerical	12576	0.29%	Reported mass of the specimen in grams; may represent an individual or main mass.
reclat	float64	numerical	12738	16.00%	Recorded latitude (decimal degrees) of the find location; may be missing or rounded.
reclong	float64	numerical	14640	16.00%	Recorded longitude (decimal degrees) of the find location; may be missing or rounded.
year	float64	numerical	265	0.64%	Year (or date) associated with the find/record; for found meteorites this often reflects year of discovery.

Table 1. Initial feature EDA.

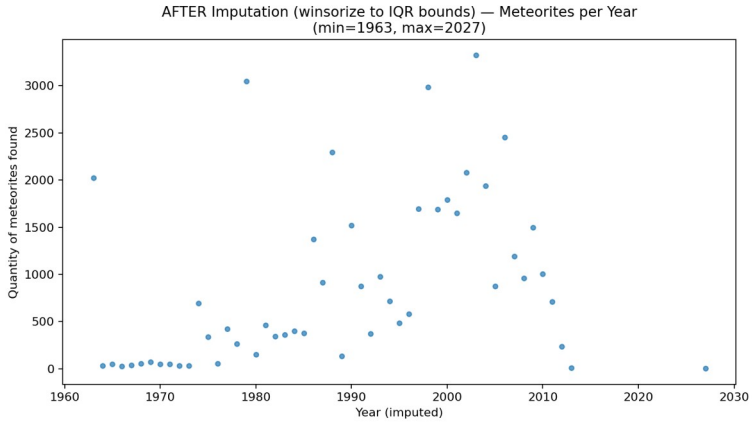
EDA plan – initial plan
I will begin by characterizing the dataset's structure—inspecting dimensions, types, unique keys, and potential duplicates—then visualizing basic relationships with a quick scatter-plot matrix to spot obvious trends, clusters, and outliers. **Table 1** indicates some missingness, so I will profile its extent and pattern (by feature and by record) and then select an imputation strategy and determine if **MCAR**, **MAR**, **MNAR** need to be considered. Distributional checks will include histograms and boxplots for each feature, with IQR-based rules to flag outliers and assess skew; transformations or robust statistics will be considered as needed. This EDA will culminate in a documented data-cleaning plan (duplicate resolution, feature pruning/engineering, and chosen imputation approach) implemented once the project formally begins. Python scripts will be used for all the EDA, when a script is used it will be indicated in the report, and a folder will be included in the project that contains all *.py files.

EDA process - praxis
The dataset was reduced to include only the **id** and **year** features, representing each meteorite and its associated discovery or fall year. Missing or invalid (initially assumed to be incorrect data types or years beyond 2100 – see '**Note**' below) year values were **imputed** to maintain continuity in the temporal record. Outliers in the **year** feature were identified and imputed using an **IQR-based imputation strategy** to minimize distortion from anomalous entries. Finally, a **temporal aggregation transform** was applied, grouping meteorite records by year and **counting the number of observations per year** to produce an annual summary of meteorite discoveries.

Note: A year filter was used, but initially set to be less than 2100. The next level of EDA will take this into account by setting the filter to only include years less than or equal to 2025.



Graph 1. Raw data, before outlier reduction. (generated with Python)



Graph 2. Data after outlier reduction. Note – it was discovered here that there is an entry from '2027' that needs to be addressed via imputation. (generated with Python)

Data and EDA Overview

The dataset was reduced to include only the **'id'** and **'year'** features, representing each meteorite and its associated discovery or fall year. Initial inspection revealed instances of missing values of the year feature, which were imputed to maintain a continuous temporal record. An **IQR-based imputation strategy** to mitigate distortion from anomalous entries was performed as can be seen in the change from Graph 1 to Graph 2. An additional outlier was detected in the **year** feature—record dated in the future (2027). This outlier was only detected when the data was graphed, in Graph 2. The next iteration of EDA will take this now known outlier into account.

A **temporal aggregation transform** was performed, grouping meteorite records by **year** and computing the **number of meteorites recorded per year**. This aggregation simplified the dataset, yielding a clearer and more interpretable view of the data's temporal distribution and improving the interpretability of visualizations.

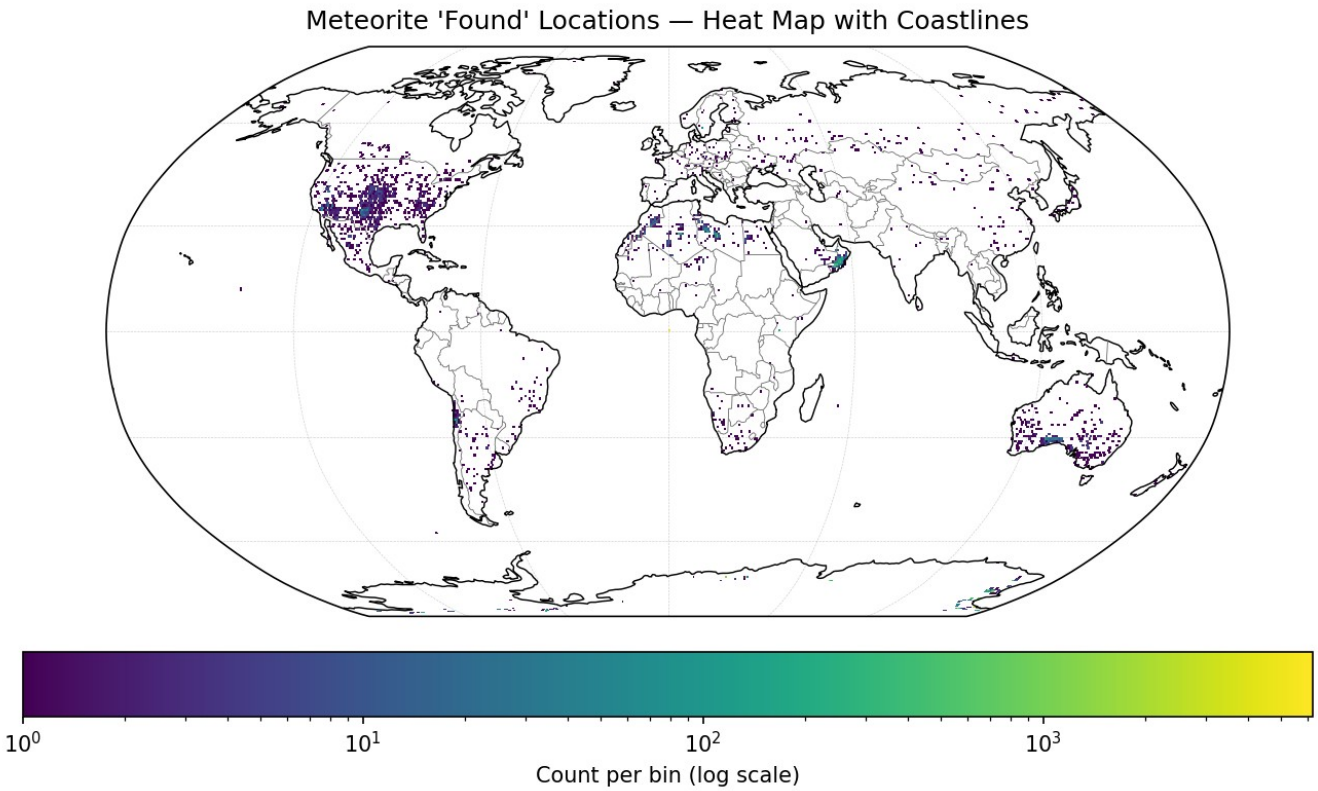
Issues and Handling

- **Missing data:** Rows with missing or zero **year** values were imputed.
- **Outliers:** Outliers were removed/imputted using median or IQR-bounded replacement. Future-dated values and a general 'valid year' check will be implemented in the next EDA-iteration.
- **Visualization challenge:** Raw scatterplots of individual meteorites produced visually dense, hard-to-read time-series graphs; aggregation by year resolved this issue.

Next Steps

Further EDA will handle known invalid dates and ensure that all temporal data points fall within a valid and realistic range. Additional validation steps will confirm that each record aligns with plausible meteorite discovery years. Once the dataset is fully cleaned and verified, the focus will shift toward exploring the **shape and distribution of the aggregated data**, using simple transformations such as **log(y+1)** to assess whether it improves visualization and interpretability. No new models will be introduced at this stage; instead, the emphasis will remain on confirming data quality, structure, and suitability for the existing model framework.

Appendix:
Here's a geo-location heat map of 'fall==Found' from the datasets (generated with Python).



Generative AI Tool use acknowledgment:
ChatGPT (OpenAI, 2025) was employed as a generative AI assistant to support debugging of Python code, refinement of data visualization methods, and exploration of potential data transformation strategies during the analysis process.