

CA Energy Storage and Capacity: Exploratory Data Analysis

This project is a Python-based analysis of California's energy storage data.

Installation and Environment

This project makes use of [Poetry](#) to manage the dependencies and the environment. Once Poetry is installed, the environment used to develop this project may be recreated and activated according to the following steps:

We use [Poetry](#) to manage dependencies, which can be seen in the `pyproject.toml`. Install Poetry, then set up the project.

Install dependencies:

```
poetry install
```

Activate the environment:

```
poetry shell
```

To run a script without activating the environment, use:

```
poetry run python notebook/<script_name>.py
```

Project Structure

The project is organized into the following set of directories, which organizes the Python code into two categories - the “toolbox” shared across multiple analyses and the “workbench” where executable scripts are stored.

- *energy_explorer/*: A set of modules combined into a single python package - the “toolbox”.
- *notebook/*: A set of executable scripts which perform the analyses - the “workbench”.
- *data/*: The original and modified data sets from which the analyses are derived.
- *figures/*: Where the output figures are written and organized.
- *docs/*: Additional information.

The Executables (Analysis)

The various components of the EDA and model analyses live in the scripts in `notebook/`. Each one has modifiable components (mostly in the form of `CONST` variable types) which modify how the data is selected and code is run.

- `chart_fuel_types.py`: Generates charts to visualize different fuel types and their capacities.
- `clean_dataframe.py`: Prepares and cleans the energy storage data for analysis.

- `map_energy_storage.py`: Plots geographic locations of energy storage facilities on a map.
- `run_capacity_series.py`: Analyzes and plots capacity time series data for selected energy storage systems.
- `run_correlations.py`: (**INCOMPLETE**) Calculates correlations between acceleration patterns of different energy storage systems.
- `run_similarity.py`: Computes similarity scores between capacity growth rates and predicts future trends based on similar patterns.