**COMP5339 Project Assignment 1 Report**

**Group TUT18 - 02**

*jzhe 0409*

*()*

*()*

**1. Dataset Description**

**1.1 Data Retrieval Methods**

We collected three following 3 datasets as required:

**NGER:** The National Greenhouse and Energy Reporting (NGER) dataset, providing annual electricity sector emissions and generation data across Australia. 10 CSV files are used here, for one per year from 2014 to 2023 (NGER.ID0075.csv to NGER.ID0243.csv).

**CER:** The Clean Energy Regulator (CER) dataset lists large-scale renewable power stations, including proposed and planned projects (historical-accredited-power-stations-and-projects-0.csv).

**ABS:** The Australian Bureau of Statistics (ABS) “Data by Regions” dataset, we used one .xlsx file (14100DO0003\_2011-24.xlsx), which covers the “economy and industry” during 2011–2024.

**Retrieval Methods:** All 3 files retraveled in the same pipeline with pull strategy:

Step 1: Download the files directly from the official website.

Step 2: Import each file into Python with pandas (read\_csv() or read\_excel()).

**Justification:**  
Together, these datasets enable a multi-dimensional analysis of Australia’s electricity sector: NGER provides historical emissions and generation trends, CER adds facility-level renewable projects, and ABS introduces socio-economic context. This combination not only supports temporal trend analysis, but also makes it possible to study about the regional variations.

**1.2 Preprocessing Methods (Integration + Cleaning)**

Pipeline: Dataset-level preprocessing -> Integration -> Overall preprocessing.

This can minimize datasets inconsistencies before merging, allows each one’s unique issues to be addressed locally, ensuring that integration yields a reliable, analysis-ready dataset.

**NGER:** Added year column -> coerced numerics -> computed generation-weighted emission intensity -> aggregated by state-year and nationally -> finally exported as NGER\_cleaned.csv.  
**CER:** Imported with ISO-8859-1 -> standardized states -> extracted accreditation year -> filtered years during 2015–2024 -> aggregated by state-year (capacity + counts) -> finally exported as CER\_cleaned.csv.

**ABS:** Detected header row -> feature selection (keep useful ones only) -> standardized states & added “AUS” totals -> removed inconsistent ACT rows -> finally exported as ABS\_cleaned.csv.

**Integration:** Normalized column names -> standardized state/year -> outer-joined NGER, CER, and ABS to retain coverage.

**Common preprocessing:** Dropped rows missing state/year -> filtered year ≥2015 -> finally saved merged dataset as ABS\_NGER\_CER\_merged.csv.

**1.3 Key insights, Challenges and Solutions**

**Insights**

Integration succeeded because all datasets shared the common keys of **state and year**. Weighted emission intensity (from NGER) provides a normalized measure for cross-state comparisons.

**Challenges encountered and solutions**

* Inconsistent state names -> Solved with explicit mapping dictionary.
* Multi-row ABS headers -> Addressed by programmatically detecting header row with “Code / Label / Year”.
* Differing year coverage across datasets -> Standardized to 2015–2023 overlap.
* Malformed numeric fields -> Resolved with coercive conversion to numeric.
* Might considering keep all dataset in runtime memory, instead of such many file IOs.

**2. Data Exploration**

**2.1 Visualizations**

To better understand the data, we performed initial exploratory visualization, focusing on the geographic distribution of power stations across Australia.

**2.1.1 Visualization 1 – Scatter Plot** (See in Appendix)

**Chart type:** Basic scatter plot (Longitude vs Latitude).

**Purpose:** Provides a quick geographic view of facility locations without external map context.

**Aspect described:** Raw spatial distribution of power stations in terms of their geographic coordinates.

**2.1.2 Visualization 2 – Geospatial Map with Basemap** (See in Appendix)

**Chart type:** Point map, using GeoPandas + Contextily overlaid on a basemap.

**Purpose:** Places power plant locations onto a real geographic background for clearer interpretation.

**Aspect described:** Highlights actual positioning of facilities within Australian states and territories, showing population and infrastructure alignment.

**2.2 Key Exploration Activities and Findings**

Converted CSV latitude/longitude values into geospatial points and re-projected into Web Mercator (EPSG:3857) to enable map overlay.

**Findings:**

* Power stations are **densely clustered along the eastern and southeastern coast**, particularly in New South Wales, Victoria, and Queensland.
* Noticeable **concentration in Western Australia’s southwest** and in **Tasmania**.
* Northern and central Australia show **sparser distribution**, consistent with lower population density and energy demand.

These geographic patterns align with Australia’s demographic and industrial concentration, giving early insights into where generation capacity is most needed.

**3. Data Augmentation**

**3.1 The web APIs applied**

* **Google Maps Geocoding API** was employed to retrieve latitude, longitude, and formatted addresses for accredited power stations.
* A **local cache file** (geocode\_cache.csv) was used to store previously retrieved results, functioning as a supplementary data source to minimize redundant API requests.

**3.2 API Usage**

We used the **googlemaps** Python package with an API key to query the Google Geocoding API. Queries combined power station name, state, postcode, and “Australia” for locality.

* **Geocoding:** geocode\_once() returned latitude, longitude, status, and address; geocode\_with\_backoff() retried failed requests with exponential backoff.
* **Caching & rate limiting:** Results were stored in geocode\_cache.csv and throttled to 5 QPS with time.sleep(0.2).
* **Augmentation:** Returned values were added to the dataset as latitude, longitude, geocode status, and formatted address.
* **Postprocessing:** Accreditation start dates were converted to datetime, and Year columns were derived.

**3.3 Relevance Justification**

Augmentation required geographic coordinates, which is right what geocoding can access. Google Maps was chosen over alternatives (e.g., OpenStreetMap, Mapbox) for its high Australian coverage, reliable formatted addresses.

What’s more, the mature Python client ensures accurate enrichment of each record for spatial analysis.

**3.4 Challenges & Solutions**

* Quota limits -> Mitigated with local caching and skipping resolved queries.
* Incomplete details -> ZERO\_RESULTS cases flagged for transparency.
* Network/API errors -> Handled with exponential backoff retries.
* Encoding issues -> CER files imported with ISO-8859-1 to avoid decoding errors.

**4. Data Transformation and Storage**

**4.1 Diagram of Database Schema:** Relational Model Diagram (see in appendix)

**4.2 Design Choices**

We used a **normalized relational model** with two tables:

* **fact\_state\_year:** state-level yearly electricity and emissions stats.
* **dim\_facility:** facility metadata, geolocation, and foreign key linking to state-year.

This design ensures:

* **Granularity separation:** aggregated vs. station-level data.
* **Integrity:** foreign key enforces consistency between tables.
* **Scalability:** easily accommodates new years, states, or facilities.
* **Spatial & temporal queries:** PostGIS geom(Point, 4326) enables spatial indexing.

PostgreSQL with PostGIS and SQLAlchemy ORM offers a robust, portable, and maintainable foundation.

**4.3 Transformation and Storage**

We transformed and stored the cleaned datasets into PostgreSQL with PostGIS enabled.

**Step 1 – Connection:** Connected to local PostgreSQL (port 5433, db=5339) via SQLAlchemy and enabled PostGIS (CREATE EXTENSION postgis) to support geometry(Point, 4326).

**Step 2 – Facility Data Cleaning and Normalization:**

* Auto-detected latitude/longitude columns from flexible name patterns.
* Renamed to lat, lon, converted to numeric for consistency.

**Step 3 – Table Writing:**

* Wrote dim\_facility (from stations\_with\_geo.csv) and fact\_state\_year (from ABS\_NGER\_CER\_merged.csv) using

pandas.to\_sql(..., if\_exists="replace").

**Step 4 – Geometry Column:**

* Added geom geometry(Point, 4326) to dim\_facility.
* Populated with ST\_SetSRID(ST\_MakePoint(lon, lat), 4326) for stations with valid coordinates.

This converts each power station’s location into a spatially indexed geometry, allowing for distance queries, mapping, and more in Assignment 2.

**Outcomes:**

* Both datasets are stored in a clean, query-ready form in PostgreSQL.
* Facility data includes precise geolocation using geometry(Point, 4326).
* The structure allows efficient joins between fact\_state\_year and dim\_facility, with potential for optional FOREIGN KEY constraints in future refinement.

**Future Works -- Some Optional Enhancements:**

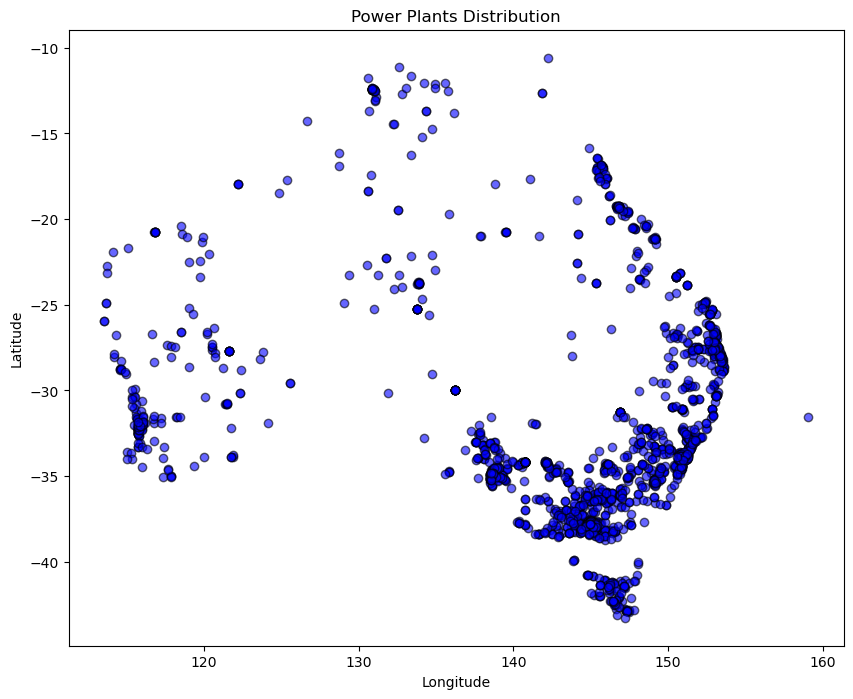
* Add a foreign key from dim\_facility(state, year) → fact\_state\_year(state, year)
* Add SQLAlchemy ORM models for both tables
* Define column types and constraints (e.g., nullable=False, unique=True)
* Create indexes on geom, state, year for fast querying

**5. Team Contributions**

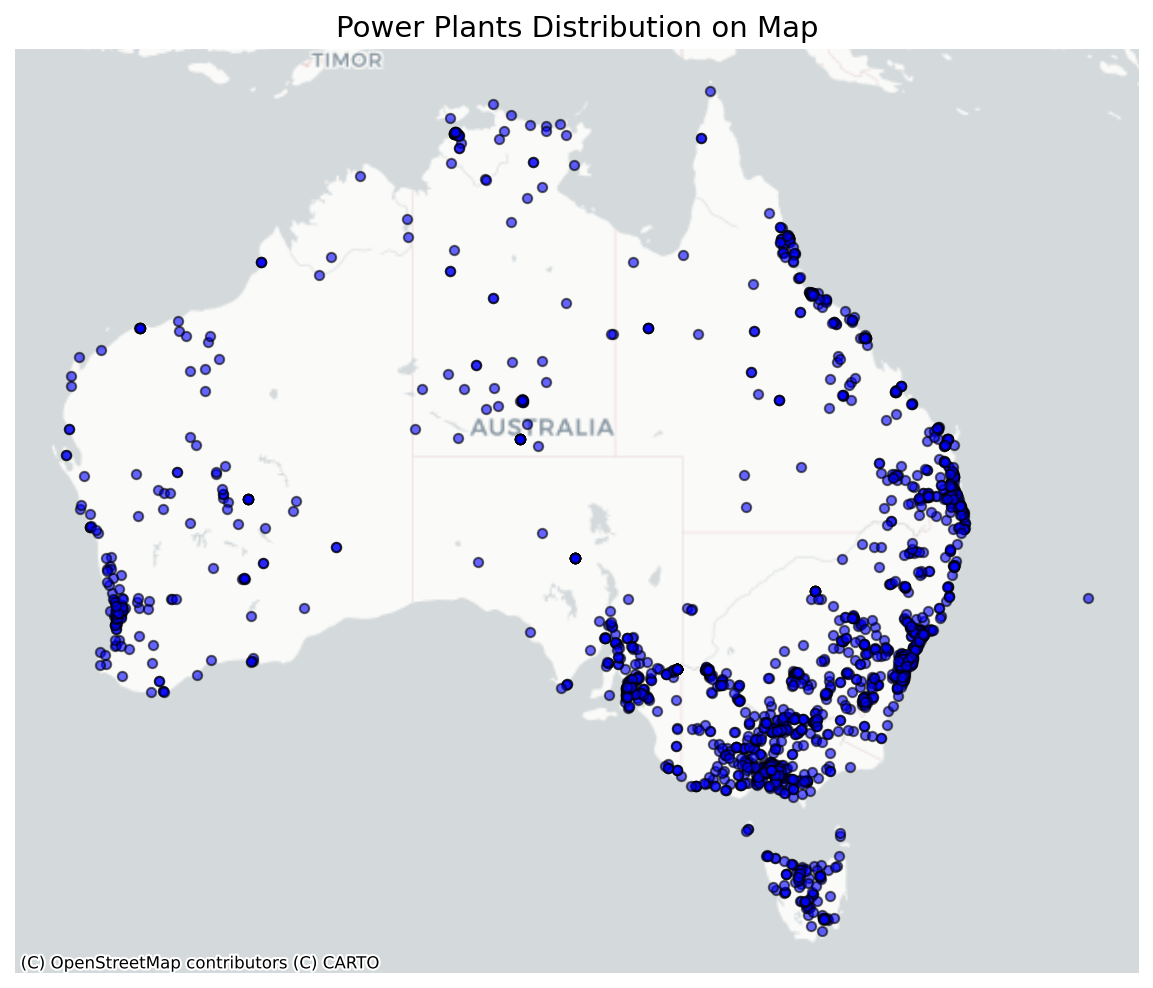
|  |  |  |
| --- | --- | --- |
| Member name | Level of contribution | Task |
| Jiachen Zheng (*jzhe0409*) | Major contributor | Documentation and Reporting  Overall final checks & requirements.txt |
| Shixing Xv () | Major contributor | Data Augmentation  EDA |
| Zhanghao Lyu () | Major contributor | Data Acquisition  Data Integration and Cleaning  Data Transformation and Storage |

**6. Appendix**

1. Visualization 1 in 2.1: Basic scatter plot (Longitude vs Latitude)

****

2. Visualization 2 in 2.1: Geospatial Map with Basemap

****

3. Database Schema Diagram: Relational Model Diagram

