ECOLOGICAL FAMILY

Driven by three component scores (Eco-1, Eco-2, Eco-3).

Eco-1: Land impacts (from `ecological_score1_coal_all.csv`)

Field	U	Direction	Meaning (terse)
`Deforestation_ha_per_year`	h	Higher=worse	Annual forest loss linked to coal.
`Land_Mined_ha_per_year`	h	Higher=worse	New mining footprint per year.
`Land_Restoration_Ratio`	0	Higher=**better**	Share of disturbed land reclaimed.

Per-metric score: normalize each (invert restoration).
Eco-1 composite: mean of the three sub-scores.

Eco-2: Ocean & water (from `ecological_score2_coal_all.csv`)

Field	Units	Direction	Meaning
`Ocean_Acid_Load`	Mt CO_2-eq	Higher=worse	Ocean CO_2 uptake linked to coal emissions (acidification pressure).
`Heavy_Metals_mgL` / `Runoff_mgL` / similar	mg/L	Higher=worse	Proxy water pollution indicators (if present / numeric).

Per-metric score: normalize each pollutant and acid load.
Eco-2 composite: mean of available sub-scores.

Eco-3: Air pollution (from `ecological_score3_coal_all.csv`)

Field	Units	Direction	Meaning
`SO2_t`	tonnes	Higher=worse	Total SO_2 emissions from coal.
`NOx_t`	tonnes	Higher=worse	Total NO_x emissions from coal.

`Coal_Ash_t`	tonnes	Higher=worse	Solid waste generated (coal ash).

Per-metric score: normalize each.
Eco-3 composite: mean of the three.

Ecological composite (family)

`Ecological_Composite` = mean(**Eco-1**, **Eco-2**, **Eco-3**).

How the script applies this
`ecological_scores.py`

- Normalizes every numeric column; auto-inverts “good” metrics by token.
- Calculates `Eco1_composite`, `Eco2_composite`, `Eco3_composite`, then `Ecological_Composite`.
- Saves CSV + two charts.

Run

bash

CopyEdit

- `python ecological_scores.py`
- `# or, z-score normalization:`
- `python -c "import ecological_scores as s; s.compute_ecological(norm_method='zscore')"`
-