ECOWHEATALY Database Generation Report

Arianna Di Poola
Gianfranco Giulioni

April 9, 2025

Objective

The script builds the **ECOWHEATALY** database, a structured JSON file containing detailed farm-level information on wheat production in Italy, based on data from the RICA (Italian FADN) dataset. The goal is to consolidate information about crop production, fertilization, pesticide usage, machinery and labor use, and economic indicators for durum and common wheat producers.

Input Data

The script loads multiple CSV files from the `Stats/RICA_DATA` directory:

- `aziende_grano.csv`: General farm-level characteristics
- `colture_grano.csv`: Crop production and costs
- `fertilizzanti_grano.csv`: Fertilizer usage
- `fitofarmaci_grano.csv`: Pesticide usage
- `bilancio_grano.csv`: Economic balances
- `certificazioni_grano.csv`: Certifications

Processing Steps

1. **Province Name Standardization:** Harmonizes province names for consistency.
2. **Outlier Removal:** (Optional, not implemented here) Uses adjusted boxplot filtering to clean input variables.
3. **Farm Metadata:** For each farm, general info is stored (region, province, orientation, gender, youth).
4. **Yearly Farm Data:** Includes:
    - Standard output, farm acreage (SAU), machine power
5. **Wheat Crop Data:** For durum and common wheat:
    - Quantity, acreage, wheat price, hours of machine use per hectare
    - Labor, machinery, fertilizer and pesticide costs

6. **Fertilizer Classification:** Types grouped (e.g., mineral, organo-mineral) with NPK content.
7. **Pesticide Classification:** Categorized by type and toxicity; only selected classes retained.

Database Structure

The final JSON `ecowheataly_database.json` is a nested dictionary with the structure:

```
{
  "farm_id": {
    "region": "...",
    "province": "...",
    "technical-economic_orientation": "...",
    "years": {
      "2016": {
        "farm_acreage": ...,
        "standard_gross_output": ...,
        "durum_wheat": {
          "produced_quantity": ...,
          "fertilizers": {
            "Mineral": {
              "nitrogen_ha": ...,
              ...
            }
          },
          "phytosanitary": {
            "Herbicide": {
              3: { "distributed_quantity_ha": ..., ... }
            }
          }
        }
      }
    }
  }
}
```

Output

The JSON file is saved to:

`Stats/ecowheataly_database.json`

It can be used for statistical analysis, sustainability modelling, or visualization.

1 Flatten Database for Machine Learning and basic Statistic overview

For basic staticts overview and machine learning analysis, the best way to organzie the data are a regular matrix with earch row uniquely identifiy a single farm and year. To this end, relevant data from the JSON are extracted and reorganized into a flexible array. The extraction

process involves several steps. First, general agricultural and economic variables are collected: produced quantity, PLV, crop acreage, hours of machinery use, and different cost components. Then, fertilizer-related information is extracted and structured both at the aggregate level and broken down by type (Mineral, OrganoMineral, Other, Micro-Mineral). Each type contains specific details such as distribution area, nutrient content, and costs.

Phytosanitary products are extracted by category (Herbicide, Insecticide, Fungicide) and classified by toxicity levels (0–4). These quantities are aggregated per hectare and stored in structured arrays, later converted into a unified 2D dataset.

2 Merging, Cleaning and Output Dataset

All collected data are merged by farm and year into a single DataFrame. Missing values in phytosanitary products are interpreted as non-use and replaced with zeros. Other NaNs are flagged and addressed through filtering. Outliers such as infinite values (e.g., from division by zero) are also removed to ensure numerical consistency. Basic statistics per year are printed to assess coverage and integrity.

To further refine the dataset and mitigate the impact of extreme values, a robust outlier detection procedure was applied to a selected set of continuous numeric variables using an adjusted boxplot method. This approach extends the classic Tukey rule by incorporating an asymmetry correction based on the *simplified medcouple*, a robust estimator of skewness. This allows for more reliable outlier identification in variables that exhibit significant skew.

The variables subjected to this cleaning step are:

- `produced_quantity`
- `PLV` (Production value)
- `fert_costs`
- `phyto_costs`
- `human_costs`
- `thirdy_costs`
- `machinery_costs`
- `Mineral_nitrogen_ha`

For each of these columns, extreme values are flagged as outliers based on their deviation from adjusted Turky fences, which are scaled asymmetrically depending on the estimated skewness of the distribution. If the data are approximately symmetric (i.e., skewness below a threshold of 0.2), the standard Tukey fences are applied. Otherwise, the method adapts the adjusted boxplot method for outliers detection throught the Medouple skewness estimation. The MedCouple is a robust statistical measure used to assess the skewness, or asymmetry, of a univariate distribution. Unlike classical skewness, which is based on the third central moment and highly sensitive to outliers, the MedCouple is designed to be resistant to the influence of extreme values. It operates by analyzing the symmetry of data around the median, rather than the mean, making it particularly suitable for distributions that are not normally distributed or that contain heavy tails. The MedCouple returns a value typically between –1 and 1, where values close to zero indicate symmetry, positive values indicate right-skewed distributions (with longer tails on the right), and negative values indicate left-skewed distributions. One of its most common applications is in the adjusted boxplot method for outlier detection, where it helps to adapt the Tukey fences according to the direction and intensity of the skewness. In such cases, the MedCouple

extends the upper or lower boundary of the boxplot depending on the shape of the distribution, allowing for a more accurate identification of true outliers in asymmetric data. Due to its robustness and simplicity, the MedCouple is widely used in exploratory data analysis, especially when dealing with real-world datasets that may deviate from ideal statistical assumptions.

All flagged outliers are put in the dataset as Not-a-Number (NaN) to ensure statistical robustness and improve the stability of downstream modeling tasks. A summary about the numebr of detected outliers is presented in the following table.

year	n_obs	n_finite_before	n_finite	n_farms
2008	2062	108	102	102
2009	1849	53	52	52
2010	2027	103	98	98
2011	1930	1376	1292	1292
2012	2068	1521	1404	1404
2013	1864	1335	1253	1253
2014	1755	1213	1135	1135
2015	1800	1318	1241	1241
2016	2140	1600	1499	1499
2017	2014	1429	1345	1345
2018	2024	1317	1236	1236
2019	1905	1257	1191	1191
2020	1952	1218	1148	1148
2021	2036	1237	1153	1153
2022	2102	1278	1163	1163

The result is a flattened, cleaned panel dataset saved as `flat_df.csv`, which serves as the input for machine learning and clustering tasks.

Authors

Arianna Di Poola
Gianfranco Giulioni