README file for "Clean Growth Counties Statistics"

Rousalis Stylianos

April 16, 2025

1 Overview

This README file describes the replication package associated with the "Clean Growth Counties Statistics" project. The directory of the replication folder contains the following objects:

1. `code`: folder containing the codes to produce the output files
2. `input`: folder containing the files used as inputs for our analysis
3. `output`: folder containing the files that are produced by the codes
4. `shapefile`: folder containing the shapefiles used for our analysis
5. `README.pdf`: this file; details the replication package

2 Statement about Rights

I certify that the author of the manuscript has legitimate access to and permission to use the data employed in this manuscript. I also certify that the author has documented permission to redistribute/publish the data contained in this replication package.

3 Software and Package Requirements

3.1 Software requirements

- Python (3.11.5)

3.2 Package requirements

- Python Packages used (and automatically installed in the code) are:
  - pandas (2.2.3)
  - numpy (1.26.4)
  - geopandas (1.0.1)
  - matplotlib (3.9.2)
  - shapely (2.0.6)
  - geopy (2.4.1)

3.3 Memory, Runtime, and Storage Requirements

- Approximate time needed to reproduce the Python analysis on a standard 2025 desktop machine: < 5 minutes

- Approximate storage space needed: 90 MB - 100 MB
- The code was last run on a Windows device with the following specifications:
  - Processor: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
  - Installed RAM: 32,0 GB
  - System type: 64-bit operating system, x64-based processor

4 Replication Folder Structure and File Descriptions

- **code**
  - `_master.py`: main script responsible for generating the output
  - `counties_lines_distance_by_capacity_bin.py`: code processing the transmission lines
  - `counties_plants_capacity_by_type.py`: code processing the power plants
- **input**

  This folder contains all the input data needed to run the Python programs, sourced from the replication package of the paper Clean Growth (Arkolakis and Walsh, 2023).
  - Power Plants (Global Power Plant (GPP) database from the World Resources Institute):

    - `gpp_bleed.csv`: holds information about the power plants, including their capacity, fuel types, and geographic coordinates.
  - Transmission Lines (from the Abstraction Algorithm descriped in the replication package of the electricity grid):
    - `links_cap.csv`: describes the lines, including columns for line identification number, the coordinates of each endpoint and the capacity of each line in MW.
    - `vertices.csv`: describes the power stations, including columns for station identification number, and the coordinates of each station. Note that these stations are used only for the abstraction of the power lines. We use the Global Power Plant database to measure power assets by county, as previously stated.
    - `intersections.csv`: describes which lines intersect which stations. It contains only two columns, one for the station identification number, and one for the line. It does not contain geographic information.

- **output**

  This folder contains all the output data generated by the programs:

    - `counties_lines_distance_by_capacity_bin.xlsx`: output data for the transmission lines generated by `counties_lines_distance_by_capacity_bin.py`
    - `counties_plants_capacity_by_type.xlsx`: output data for the power plants generated by `counties_plants_capacity_by_type.py`
    - `counties_lines_plants_statistics.xlsx`: integrates both sets of data.

- **shapefile**

  This folder contains county boundary shapefiles sourced from the US Census Bureau.

5 Instructions to Replicators

Run `code/_master.py` to go from input data to final output. Note that order is important: the programs need to be run in the order indicated in the master do file.

6 Code Description

6.1 `counties_lines_distance_by_capacity_bin.py`

Some lines are entirely within a single county, while others span across multiple counties.To assign each line segment to the appropriate county, we first determine which county the start and end points of the line belong to, according to their coordinates.

If both the start and end points fall within the same county, no further processing is needed. However, if the line crosses into different counties between the start and end points, we use the "overlay" command in Geopandas. This command splits the original line into smaller segments, each corresponding to a specific county polygon.

So, a line crossing, for example, three counties is divided into three segments, each associated with a different county. This approach ensures that each line segment is fully contained within a single county. Calculating the total distance of all lines within each county is then straightforward.

Also, lines have varying maximum capacities. To maintain consistency with the figures in the paper Clean Growth (Arkolakis and

Walsh, 2023), we cap any capacity exceeding 5000 MW at 5000 MW. Then, we categorize each line into bins based on its capacity: those with a capacity less than 1000 MW, 2000 MW, 3000 MW, 4000 MW, or 5000 MW.

For each county, we report the total length of lines in kilometers, along with the number of lines that are fully contained within the county and those that span across county borders, categorized by maximum capacity.

6.2 `counties_plants_capacity_by_type.py`

Each power plant is assigned to a county polygon according to its coordinates.

We classify power plants by type—Fossil (gas, oil, coal, hydro), Renewable (solar, wind, biomass, waste, geothermal), as well as specifically Solar and Wind and Other (cogeneration, storage, nuclear, petcoke, wave and tidal).

For each county, we provide the number of plants within its boundaries and the total capacity in MW for each type and overall.

7 References

Arkolakis, C., Walsh, C., 2023. Clean growth. NBER Working Paper 31615.