electricityMap Data Analyst Challenge

This repository contains my work and answers for electricityMap's Data Analyst challenge, which is outlined in `electricityMap Data Analyst Challenge.pdf`.

Input Dataset

The dataset contains data regarding how many megawatts (MW) of electricity were generated by production type in the DK2 bidding zone (representing the eastern portion of Denmark). This data is broken down by hour for the entire year of 2020.

The data regarding electricity production by production type for East Denmark (bidding zone DK2) was sourced from ENTSO-E. It can be found here.

In this dataset, MTU stands for "market time unit". Times in data/actual_generation_per_production_type_2020.csv are shown in CET.

All columns in this dataset other than "Area" and "MTU" use Megawatts as their unit. Because each row of data in these columns represents an hour of electricity production, Megawatts is equivalent to Megawatt Hours.

Many of the columns regarding types of electricity production are filled with 'n/e' ('not expected'). This means that that type of electricity production does not occur within the bidding zone. These columns could also contain 'n/a' ('not available'), meaning that that type of electricity production is expected but the data is missing.

Emission Factors

Emission factors for each production type of electricity are contained in emissionFactors.json.

The "value" argument of each type of production type represents the emission factors for that production type. The unit of these values is grams of carbon dioxide equivalent per kilowatt hour of generation (gCO_2eq/kWh).

More information about these emission factors is available in the Parliamentary Office of Science and Technology's "Carbon Footprint of Electricity Generation" article published in October 2006, which can be found here.

Code Files

There are nine .py files in this repository.

- `SQLiteStDev.py` contains an additional SQLite aggregation function to calculate standard deviation.
- `run.py` runs all code to complete the steps outlined in the Steps section.
- `create_db.py` contains the code which imports the `.csv` dataset, cleans it, and creates two SQLite databases (one for electricity production and one for emissions) from the dataset.
- `create_db_methods.py` contains the support functions for `create_db.py`.

- `query_db.py` contains the code which queries from the SQLite databases to get the data which will be visualized.
- `query_db_methods.py` contains the support functions for `query_db.py`.
- `summary_stats.py` contains the code which gets summary statistics regarding emissions data.
- `visualize_analyses.py` contains the code which visualizes the data queried by `query_db.py`.
- `visualize_analyses_methods.py` contains the support functions for `visualize_analyses.py`.

There are three folders in this repository.

- `data` contains the original dataset, the databases created by `create_db.py`, datasets created by `query_db.py`, and `emissionFactors.json`, which contains emission factors for each production type.
- `figures` contains all data visualizations created by `visualize_analyses.py`.
- `tables` contains the table created by `summary_stats.py`.

Steps

The first four steps occur in create_db.py: importing the data, cleaning the data, creating the SQLite database, and inserting the cleaned dataset into the database.

Importing the Data

This step takes place in `create_db.py`. Data is imported using pandas' `read_csv`.

Cleaning

This step takes place in `create_db.py`. Four steps were taken during cleaning:

1. Columns that were entirely 'n/e' were dropped from the dataset because that type of electricity production did not occur in the DK2 bidding zone during 2020. This resulted in dropping thirteen columns.
2. Columns that were over 10% 'n/a' (missing data) were flagged. No columns were flagged.
3. The format of the MTU (datetime) column was changed so that SQLite could recognized the column as containing datetimes.
4. Column names were adjusted.

Creating the SQLite Databases

This step takes place in `create_db.py`. This step was completed using the sqlite3 package. Two databases are created and stored in the `data` folder.

- `entsoe_2020_data.db` contains the original cleaned dataset plus a generated `TotalProduction` column.
- `entsoe_2020_emissions_data.db` contains the original cleaned dataset weighted by the emission factors from `data/emissionFactors.json`, plus a generated `TotalEmission` column.

Inserting Data into the Database.

This step takes place in `create_db.py`. This step was completed using the sqlite3 and pandas package.

Querying Data from the Database

This step takes place in `query_db.py`. During this step, we create seven new datasets which will later be visualized. All datasets are created by querying the databases and/or performing calculations on the queried datasets. All datasets queried from the databases are broken down by season. Seasons are defined as follows:

- Spring: February, March, April
- Summer: May, June, July
- Fall: August, September, October
- Winter: November, December, January

Dataset 1: Total Average Production over 24 Hours

This dataset contains how much energy is produced on average over 24 hours. It is broken down by season. Its unit is megawatt hours (MWH).

Dataset 2: Total Average Emissions over 24 Hours

For this dataset, we will query the a day's worth of average hourly production for each production type. We will then multiply this production data by the respective emission factors to determine the average hourly emissions for each production types. The unit of the emission factor is grams carbon dioxide equivalent per kilowatt hour, which is equivalent to kilograms carbon dioxide equivalent per megawatt hour. We multiply production (megawatt hours) by the emission factor (kilograms carbon dioxide equivalent per megawatt hour), giving us a unit of kilograms carbon dioxide equivalent ($kgCO_2eq$) for Dataset 2.

Dataset 3: Efficiency over 24 Hours

This dataset divides the Total Production by Total Emissions over 24 hours to demonstrate how much electricity is produced per kilogram of CO_2 equivalent emissions. Its unit is $MWH/kgCO_2e$.

Dataset 4: Average Solar Production over 24 Hours

This dataset contains how much electricity on average is produced by solar over 24 hours. It is broken down by season. Its unit is megawatt hours (MWH).

Dataset 5: Average Offshore Wind Production over 24 Hours

This dataset contains how much electricity on average is produced by offshore wind over 24 hours. It is broken down by season. Its unit is megawatt hours (MWH).

Dataset 6: Average Onshore Wind Production over 24 Hours

This dataset contains how much electricity on average is produced by onshore wind over 24 hours. It is broken down by season. Its unit is megawatt hours (MWH).

Dataset 7: Average Coal Production over 24 Hours

This dataset contains how much electricity on average is produced by coal over 24 hours. It is broken down by season. Its unit is megawatt hours (MWH).

Visualizing the Data

This step takes place in `visualize_analyses.py`. During this step, bar charts are created for each of the seven datasets. These visualizations are saved in `figures`.

Summary Statistics

This step takes place in `summary_stats.py`. During this step, standard deviation, normalized standard deviation, range, and normalized range are calculated for emissions data.