Analysis, Use Cases, and Functions

Dataset: Total Greenhouse Gas Emissions

Dataset URL: https://datasource.kapsarc.org/explore/dataset/total-global-greenhouse-gas-emissions/information/?disjunctive.world_region&disjunctive.country&disjunctive.substance

Can we access Dataset?: Yes, it is publicly available (Public Domain License)

Dataset Topic: This dataset provides the total greenhouse gas emissions of countries from 1970 to 2021. The countries are categorized into regions, and the emissions are further broken down into the types of greenhouse gas substances. A user can select a country and view the country's greenhouse gas emissions on a yearly basis from 1970 to 2021. The greenhouse gas emissions are provided in Kt and that of CO2 is provided in MtCO2.

Type of Data Available (Chart/Values/...):
Values (numerical emissions data).

Is it live data? Frequency of data updates?
No, This is not live data. The dataset extends only to 2021.

How can we access the data (Json/Download/View in site)?
The dataset is available for download in spreadsheet and JSON formats. The dataset can also be served online through API keys.

Dataset Pre-processing Need? List any Steps:

- Handle missing data, if any.
- Convert the timestamp to a human-readable format if needed.

Data Summary:

- **Time Period**: The dataset covers from 1970 to 2021
- **Granularity**: Country-level emissions data.
- **Data Structure**:
    - **Rows**: There are multiple entries per country per year, which means a time-series structure.
    - **Columns**: 7 columns in total:
        - Year: Year of recorded data (e.g., "1986").
        - World Region: The area of the world the country is located (e.g., "Eastern Africa")
        - country: Country name (e.g., "Turkey").
        - Substance: Type of greenhouse gas being emitted (e.g., "CH4").
        - IPCC Annex: Intergovernmental Panel on Climate Change annex (e.g., "Annex I").
        - Value: Amount of greenhouse gas emitted in a given year by a country or region
        - Unit: Standard measuring unit for the greenhouse gas emitted.

Operations Performed (CRUD): The only CRUD operation performed is the reading of the data.

Supported Data

The **GreenhouseGasEmissionsTool** retrieves data for the following:

- **Time Frame**: 1970 to 2021.
- **Countries/Regions**: Includes all countries and regions in the world. Some countries may not have data for all the years.
- **Substances**: Includes greenhouse gases like, CH4, N2O, NF3, CO2, etc.

Key Features

1. **Country-Specific Queries**: Retrieve greenhouse gas emissions data for one or more specified countries (e.g. United States, Jamaica, Fiji).
2. **Year-Specific Queries**: Filter emissions data by a specific year in 'YYYY' format (e.g. 1995).
3. **Substance-Specific Queries**: Filter data by the type of greenhouse gas such as CH4, N2O, NF3, etc.
4. **Combinations of Filters**: Combine country, year, and substance filters to retrieve precise data.
5. **Error Handling**: The tool provides clear error messages when no data matches the criteria or if the input is invalid (e.g., incorrect year format or unavailable country or substance).

Types of Queries Supported

The tool supports a variety of queries, categorized by parameters such as country, year, and substance:

1. **Basic Queries**: Provide overall greenhouse gas emissions data across all countries, substances, and years.
2. **Country-Specific Queries**: Retrieve data for one or more countries.
3. **Year-Specific Queries**: Filter by a specific year.
4. **Substance-Specific Queries**: Aggregate data by substances.
5. **Combination of Country, Year, and Substance**: Combine these filters for precise results, e.g., "Show CH4 emissions for Jamaica in 1972."
6. **Handling Missing Data**: The tool provides clear error messages if the requested data is unavailable or the input is invalid.

Some Use cases for the GreenhouseGasEmissionsTool:

users can ask a variety of questions about greenhouse gas emissions data. Here are some types of questions that can be asked, categorized by the different parameters that the tool can handle:

1.Basic Queries (No Specific Filters)

- Example:
  - "What is the total greenhouse gas emissions for the entire world?

Explanation: These queries don't specify any filters, so the tool will return aggregated data for all countries, all years, and all substances.

2. Country-Specific Queries

- Example:
  - "What is the total greenhouse gas emissions for Jamaica?"

Explanation: These questions specify countries and the tool will aggregate emissions data for the specified countries for all years and substances.

3. Year-Specific Queries

- Example:
  - "What was the total greenhouse gas emissions in 1972?"

Explanation: The tool will filter the emissions data by the specified year, aggregating the results for all countries and substances.

4. Substance-Specific Queries

- Example:
  - "What was the total CH4 emission?"

Explanation: These questions specify a substance or substances, and the tool will aggregate emissions data for those substances for all countries and years.

5. Country and Year Queries

- Example:
  - "What was the greenhouse gas emission for Jamaica in 1972?"

Explanation: The tool will filter the emissions data by the specified country and year, aggregating the results for all substances.

6. Country and Substance Queries

- Example:
  - "What was the total CH4 emission for Jamaica?"

Explanation: These queries specify both the country and substance and the tool will aggregate emissions data for those filters for all years.

7. Year and Substance Queries

- Example:
  - "What was the CH4 emissions for 1972?"

Explanation: The tool will filter the emissions data by the specified year and substance, aggregating the results for all countries.

8. Country, Year, and Substance Queries

- Example:
  - "What was the CH4 emissions for Jamaica in 1972?"

Explanation: These queries specify all three parameters—country, year, and substance—allowing the tool to filter the emissions data accordingly.

9. Aggregate Emissions by Substance For All Countries

- Example:
  - "What was the total CH4 emissions in 1972?"

Explanation: These queries aggregate emissions for a specific substance in a specific year for all countries.

10. Handling Multiple Countries

Explanation: When multiple countries are specified, the tool will aggregate and provide results for each country, optionally considering the specified year or substance.

11. Queries Involving Missing Data Handling

Handling the incorrect parameters and if the user's query is outside the supported year range or if the specified country or substance is not in the dataset, inform them of the valid parameters.

- "Show the error message if no data is found for the specified criteria."

Explanation: The tool is equipped to handle cases where no data matches the criteria and will provide an appropriate error message.

This variety of questions allows for flexible querying of the total greenhouse gas emissions data, covering a range of scenarios that users might encounter. The GreenhouseGasEmissionsTool is designed to interpret these queries and provide relevant aggregated results based on the specified criteria.

Setting Up the Tool and Performing Integration and Unit Testing

1. Load Files into the Environment

- Load the files in the Greenhouse gas emissions folder into the development environment be it a locally installed IDE or an online environment.

2. Install Required Dependencies

- Install the required dependencies in your environment. These dependencies can be found in the requirements.txt file.

3. Install the Llama Stack Client

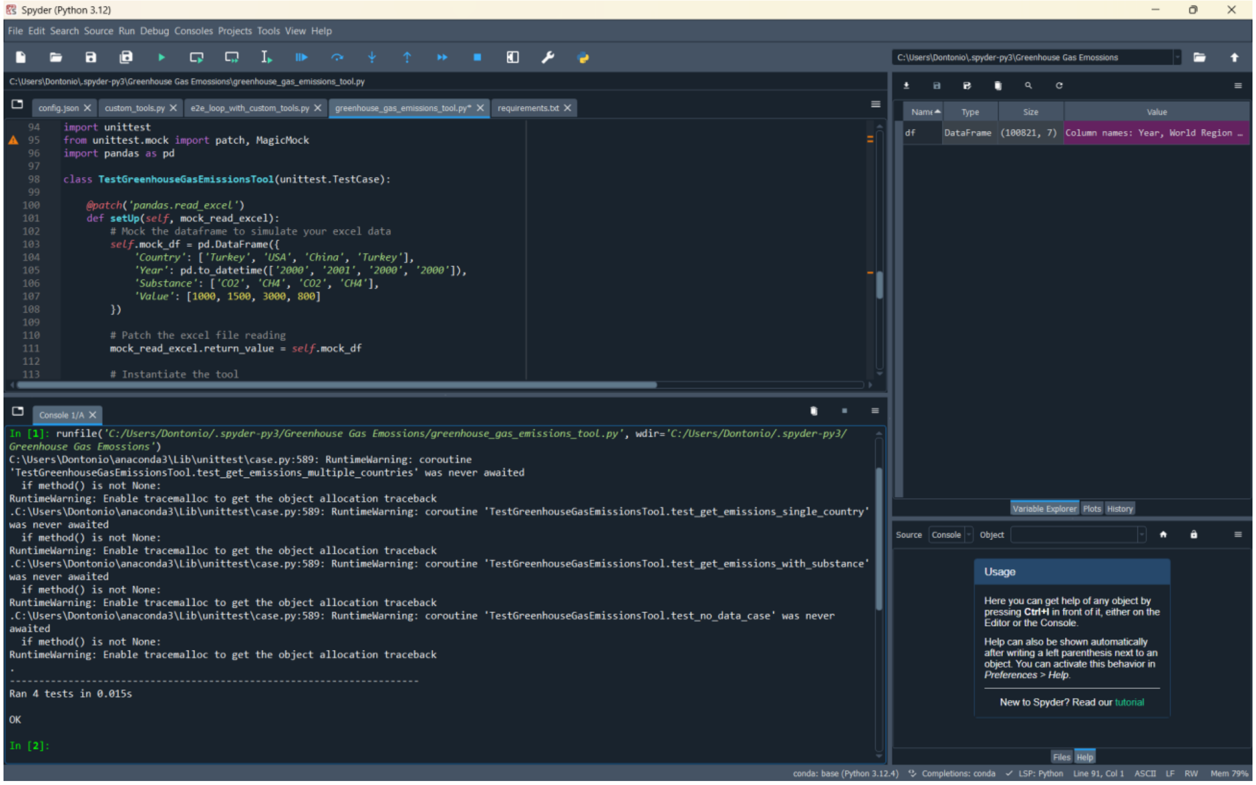
- This is required for some functions in the CustomTool class to work as the GreenhouseGasEmissionTool uses the SingleMessageCustomTool from the CustomTool class.

4. Run the Custom and Greenhouse Gas Emission Tool Scripts

- Execute the custom tool and greenhouse gas emissions tool scripts to initialize the necessary functions:

5. Run the Unit Test Script

- Run the Unit Test Script after running the main GreenhouseGasEmissionsTool test and validate the individual functions in this tool.


Spyder (Python 3.12)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Dontonio\.spyder-py3\Greenhouse Gas Emossions\greenhouse_gas_emissions_tool.py
config.json
custom_tools.py
e2e_loop_with_custom_tools.py
greenhouse_gas_emissions_tool.py*
requirements.txt
C:\Users\Dontonio\.spyder-py3\Greenhouse Gas Emossions

```python
import unittest
from unittest.mock import patch, MagicMock
import pandas as pd

class TestGreenhouseGasEmissionsTool(unittest.TestCase):

    @patch('pandas.read_excel')
    def setUp(self, mock_read_excel):
        # Mock the dataframe to simulate your excel data
        self.mock_df = pd.DataFrame({
            'Country': ['Turkey', 'USA', 'China', 'Turkey'],
            'Year': pd.to_datetime(['2000', '2001', '2000', '2000']),
            'Substance': ['CO2', 'CH4', 'CO2', 'CH4'],
            'Value': [1000, 1500, 3000, 800]
        })

        # Patch the excel file reading
        mock_read_excel.return_value = self.mock_df

        # Instantiate the tool
```

```
In [1]: runfile('C:/Users/Dontonio/.spyder-py3/Greenhouse Gas Emossions/greenhouse_gas_emissions_tool.py', wdir='C:/Users/Dontonio/.spyder-py3/
Greenhouse Gas Emossions')
C:\Users\Dontonio\anaconda3\Lib\unittest\case.py:589: RuntimeWarning: coroutine
'TestGreenhouseGasEmissionsTool.test_get_emissions_multiple_countries' was never awaited
  if method() is not None:
RuntimeWarning: Enable tracemalloc to get the object allocation traceback
.C:\Users\Dontonio\anaconda3\Lib\unittest\case.py:589: RuntimeWarning: coroutine 'TestGreenhouseGasEmissionsTool.test_get_emissions_single_country'
was never awaited
  if method() is not None:
RuntimeWarning: Enable tracemalloc to get the object allocation traceback
.C:\Users\Dontonio\anaconda3\Lib\unittest\case.py:589: RuntimeWarning: coroutine 'TestGreenhouseGasEmissionsTool.test_get_emissions_with_substance'
was never awaited
  if method() is not None:
RuntimeWarning: Enable tracemalloc to get the object allocation traceback
.C:\Users\Dontonio\anaconda3\Lib\unittest\case.py:589: RuntimeWarning: coroutine 'TestGreenhouseGasEmissionsTool.test_no_data_case' was never
awaited
  if method() is not None:
RuntimeWarning: Enable tracemalloc to get the object allocation traceback
.
----------------------------------------------------------------------
Ran 4 tests in 0.015s

OK

In [2]:
```

Name	Type	Size	Value
df	DataFrame	(100821, 7)	Column names: Year, World Region …

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial