Question

How do fossil fuel subsidies and CO_2 emissions from trade relate across different countries?

Data Sources

Two datasets were chosen for this analysis:

1. Fossil Fuel Subsidies

MetadataURL:
https://climatedata.imf.org/datasets/d48cfd2124954fb0900cef95f2db2724_0/explore

Data URL: https://opendata.arcgis.com/datasets/d48cfd2124954fb0900cef95f2db2724_0.csv

Data Type: CSV

Description: This dataset shows the value of fossil fuel subsidies, both explicit and implicit, for various countries from 2015-2021. Subsidies are broken down by fuel type (coal, electricity, natural gas, petroleum) and subsidy type (explicit, implicit).

Data Structure: Semi-structured Data

Data Quality: [Accuracy ☑, Completeness ☒, Consistency ☑, Timeliness ☑, Relevancy ☑]

License: Custom License [Click to see full details]

2. CO_2 Emissions Embodied in Trade

MetadataURL:
https://climatedata.imf.org/datasets/7ba962035bb548bb9893add2b5491896_0/explore

Data URL: https://opendata.arcgis.com/datasets/7ba962035bb548bb9893add2b5491896_0.csv

Data Type: CSV

Description: This dataset contains annual estimates of CO_2 emissions embodied in a country's gross exports from 2015-2021.

Data Structure: Semi-structured Data

Data Quality: [Accuracy ☑, Completeness ☒, Consistency ☑, Timeliness ☑, Relevancy ☑]

License: Custom License [Click to see full details]

I choose these two datasets and it provide the necessary information to analyze the relationship between fossil fuel subsidies provided by countries and their CO_2 emissions from international trade, allowing insights into the environmental impacts of such subsidies.

Data Pipeline

The project follows an **ETL (Extract, Transform, Load)** pipeline structure implemented using Python (pandas, sqlite3) technology. The entry point of the project is ***pipeline.sh*** which runs ***pipeline.py*** which generates an SQLite file named "FossilFuelSubsidiesCO2Emissions.sqlite".

- goodsExtractor (HttpExtractor)
- goodsInterpreter (TextFileInterpreter)
- goodsMetaDataFooterDeleter (RowDeleter)
- timeTransportedGoodsHeaderWriter (CellWriter)
- quantityGoodsHeaderWriter (CellWriter)
- goodsTableInterpreter (TableInterpreter)
- monthCapitalizer (TableTransformer)
- goodsCSVInterpreter (CSVInterpreter)
- goodsLoader (SQLiteLoader)

Transformation and Cleaning Steps: Based on the information provided, the following transformation and cleaning steps were performed:

1. Select Columns: Choosing or deleting desired/unnecessary columns from the datasets.
2. Select Rows: Filtering rows based on specific conditions (for one of the datasets).
3. Melt Table: Both input datasets contained year information as columns. The melt operation was performed to convert these columns into rows.
4. Rename Columns: Columns were renamed for better clarity and understanding.
5. Fix Year Data: Year data was cleaned and formatted consistently across the datasets.
6. Join Data: The two input datasets were joined together to create a combined dataset.
7. Drop Null Values: Rows with null or missing values were dropped from the final dataset.

Problems Encountered and Solutions:

- Handling varying column structures (e.g. years as columns) between datasets
- Joining datasets with different granularity levels
- Cleaning and standardizing year data across datasets

These challenges were addressed through data transformation techniques like melting tables, string manipulations, and proper join operations. One of the examples of melting table is shown below:

Here both input datasets contained year information as columns. The melt operation was performed to convert these columns into rows, making the data easier to work with.

ISO3	***F2015***	***F2016***	***F2017***	***F2018***
ARG	26.057	28.669	26.771	30.708

After melting, it looks like this,

ISO3	***Year***	***Incidents***
ARG	2015	26.057
ARG	2016	28.669
ARG	2017	26.771
ARG	2018	30.708

Error Handling and Changing Input Data: The report mentions that errors were properly handled in the pipeline. Specifically:

1. If the source URL is faulty or unreachable, the execution will stop by showing proper error messages.
2. If the program doesn't find the specified saving directory (if pipeline.py was executed from the project folder), it will resolve the path automatically.
3. If the source data adds more year data in the future, the program automatically converts it to row values due to the dynamic programming approach implemented.

Result and Limitations

Data Structure: Structured Data

Data Quality:

- ✓ Accuracy
- ✓ Completeness
- ✓ Consistency
- ✓ Timeliness
- ✓ Relevancy scores

Format: SQLite file

The final dataset is saved as an ***SQLite file*** named "**FossilFuelSubsidiesCO2Emissions.sqlite**", containing joined information from the two source datasets. This allows for querying and analysis of the relationship between fossil fuel subsidies provided by countries and their CO_2 emissions embodied in international trade. The SQLite format enables efficient querying and analysis using SQL, facilitating flexible data retrieval, and supporting data-driven decision-making processes.

The data stored in SQLite can seamlessly integrate with various data processing and analysis tools. This includes data visualization platforms for creating insightful visualizations, machine learning frameworks for predictive modeling and pattern recognition, as well as business intelligence solutions for comprehensive reporting and analytics.

Potential Use Cases:

- ➢ Analyzing the impact of fossil fuel subsidies on a country's carbon footprint from trade
- ➢ Identifying countries with high subsidies and emissions for policy interventions
- ➢ Studying the correlation between subsidy types/amounts and emissions levels
- ➢ Integrating with other economic/environmental datasets for broader analysis
- ➢ Building data visualizations and dashboards for stakeholders

The SQLite format ensures data portability, easy querying with SQL, and integration with various data analysis and reporting tools. This comprehensive dataset can support evidence-based policymaking, climate change mitigation strategies, and sustainable trade practices.