Sidharth Santhosh

AJ23SYD011

**World Energy Emission Dashboard**

The World Energy Emission Dashboard is a data visualisation tool that offers a clear picture of worldwide energy production, consumption, and CO2 emissions. The dashboard was created with the help of the Python modules Holoviews, Panel, and Datashader.

The dashboard is split into four parts, each of which displays a distinct component of the data:

* **CO2 Emission by Country:** A bar chart depicts the total CO2 emissions for each nation in the dataset in this section. The viewer may simply examine which nations pollute the most and how their emissions have changed over time.
* **Energy Consumption by Type:** This section includes a line chart that depicts the overall energy consumption for each energy source (coal, oil, gas, and so on) through time. The user may readily view which forms of energy are most consumed and how their usage has evolved over time.
* **Energy Production by Type:** This section includes a line chart that depicts the overall energy production for each energy source through time. The user may readily examine which forms of energy are most often generated and how their production has evolved over time.
* **GDP by Country:** This section includes a line chart displaying the total GDP for each nation in the dataset. The user may quickly determine which nations have the most wealth and how their wealth has evolved over time.

Everyone interested in global energy production, consumption, and CO2 emissions will find the dashboard useful. It enables users to visualise complicated data quickly and simply, gaining insights into trends and patterns. The dashboard makes it simpler for policymakers, academics, and the public to make educated decisions on energy policy and climate change by giving a clear and succinct summary of the facts.

**Dataset:**

<https://www.kaggle.com/datasets/lobosi/c02-emission-by-countrys-grouth-and-population>

The data was obtained from the US Energy Administration and combined for easy examination. It is a compilation of some important components that contribute to CO2 emissions, including everything from the production and consumption of each major energy source for each country, as well as its pollution rating each year. Each country's GDP, population, energy intensity per capita (person), and energy intensity per GDP (per person GDP) are also included. All the data ranges from the 1980s to 2019.

**Feature Descriptions:**

* Country - Country in question
* Energy\_type - Type of energy source
* Year - Year the data was recorded.
* Energy\_consumption - Amount of Consumption for the specific energy source, measured (quad Btu)
* Energy\_production - Amount of Production for the specific energy source, measured (quad Btu)
* GDP - Countries GDP at purchasing power parities, measured (Billion 2015$ PPP)
* Population - Population of specific Country, measured (Mperson)
* Energy\_intensity\_per\_capita - Energy intensity is a measure of the energy inefficiency of an economy. It is calculated as units of energy per unit of capita (capita = individual person), measured (MMBtu/person)
* Energy\_intensity\_by\_GDP- Energy intensity is a measure of the energy inefficiency of an economy. It is calculated as units of energy per unit of GDP, measred (1000 Btu/2015$ GDP PPP)
* CO2\_emission - The amount of C02 emitted, measured (MMtonnes CO2)

**Data Cleaning:**

The emissions data set had a large number of null values and 0 duplicate entries. The null values have been converted to 0, resulting in a flow on the data set. According to the column 'Country,' the data set 'world' contains about 180 records that were removed from the dataset. Records having a specific entry in a column should be deleted if the entry does not fit the rest of the data or is unrelated to the research. It is crucial, however, to ensure that removing such entries has no significant influence on the dataset's representativeness or integrity.  
  
  
**Dashboard: -**

**Import statements: -**

This code imports several Python libraries and configures the environment for data visualisation with HoloViews, Panel, and datashader.

* Pandas is a popular Python data analysis toolkit used for data manipulation and analysis.
* NumPy is a Python library for numerical computation.
* HoloViews is a Python toolkit for producing interactive visualisations. It enables users to construct high-level visualisations using declarative syntax and a little bit of code.
* Panel is a Python framework for building configurable web apps and dashboards. It is built on HoloViews and allows you to develop interactive controls that can be used to change visualisations in real time.
* HvPlot is a HoloViews-based Pandas plotting addon. It offers a high-level API for easily constructing interactive plots and charts using Pandas dataframes.
* Datashader is a package that rasterizes enormous datasets into graphics that can be shown on a computer screen, allowing for efficient visualisation. It is very effective for visualising datasets containing millions or billions of data points.
* The hv. extension('bokeh') line allows HoloViews visualisations to leverage the Bokeh backend, which allows interactive plots and charts to be presented in a web browser.

Finally, there is the holoviews. operation. The datashader line imports a collection of procedures for HoloViews that use datashader to draw huge datasets effectively. To facilitate efficient visualisation of huge datasets, the datashade and dynspread functions from datashader are also imported.

**Creating widgets: -**

A widget in Python is a graphical user interface (GUI) component that allows users to interact with a programme. Buttons, checkboxes, sliders, dropdown menus, and other interactive components that let users to input data, make selections, or initiate programme activities are examples of widgets.

Graphical user interface, text, application

Description automatically generatedThis code creates three interactive widgets using the Panel library for Python, allowing users to select different options for visualizing data from a dataframe called df.

Image 1: - code snippet

The first widget, yaxis\_co2\_source, generates a dropdown list of unique nation names from the dataframe df's 'nation' column. The unique () function returns a numpy array containing the column's unique values, which is subsequently transformed to a list using the tolist () method. This widget allows users to choose a nation and view CO2 emissions data for that country.

The second widget, yaxis\_energy, creates a dropdown list of unique energy types from the 'Energy\_type' column of the dataframe df. This widget allows users to select a specific type of energy to visualize energy consumption data for that energy type.

The third widget, slider\_year, generates a slider with discrete values for the dataframe df's 'Year' column. Users may use the slider to pick a certain year to see data for that year.

A picture containing kitchenware

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedThese widgets work together to allow users to dynamically explore and visualise many elements of the data in df. Users may access statistics for a given country, energy type, and year by selecting that combination of choices. These widgets may be used in conjunction with other visualisation technologies, such as HoloViews, to build interactive dashboards a nd data visualisations.

Image 4: - Third Widget

Image 3: - Second Widget

Image 2: - First Widget

**Creating Pipelines: -**

A pipeline in Python is a series of data processing processes that are conducted in a precise order to turn raw data into a more useable form or to extract insights from it. Each pipeline step receives input data from the previous phase, performs some processing or transformation on it, and then delivers the result to the next pipeline step.

Image 5: - Code snippet

This code defines four pipelines, each of which performs a series of data transformations and generates a HoloViews plot or chart. Here's what each pipeline does:

* Co2\_source\_pipeline: This pipeline organises the data by year and country, adds the CO2 emission figures, and provides an area map of each nation's total CO2 emissions over time.
* Energy\_consumption\_pipeline: This pipeline organises the data by year and energy type, accumulates the energy consumption figures, and produces a line plot depicting total energy consumption over time for each energy type.
* Energy\_production\_pipeline: This pipeline organises the data by year and energy type, accumulates the energy production numbers, and provides a line plot displaying total energy production over time for each energy type.
* Gdp\_pipeline: This pipeline organises the data by year and country, adds the GDP numbers (after removing any rows with null values), and provides a line plot of each nation's total GDP over time.

To produce the accompanying plot or chart, each pipeline uses the hvplot function from the hvplot. pandas’ package. The hvplot method creates a HoloViews object, which may then be customised and combined with other HoloViews objects to build more complicated visualisations.

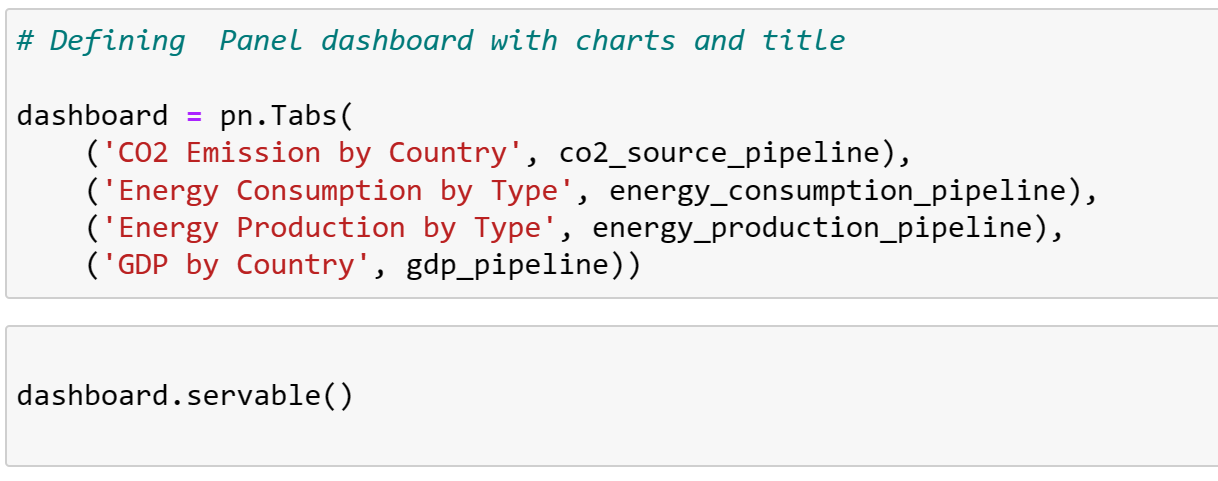


Image 6: - Code snippet

This code generates a dashboard in Python by combining the charts created by the pipelines specified before into a single, interactive interface.

Using the names and pipelines given as parameters, the pn. Tabs method generates a tabbed interface with one tab for each chart. Each pipeline is supplied as the tab's content, along with a label. Finally, there's the dashboard. The servable () method converts the dashboard into a web app that can be browsed and interacted with in a web browser.

Overall, this code provides a simple but effective approach to visualise and analyse the data contained in the DataFrame, allowing users to obtain insights into the links between CO2 emissions, energy use and production, and GDP across different countries and years.

**Chart, line chart

Description automatically generatedChart, line chart

Description automatically generatedChart, line chart

Description automatically generatedChart, histogram

Description automatically generatedDashboard Screen Shots:**

Image 10: - Output

Image 9: - Output

Image 8: - Output

Image 7: - Output

**References:**

* <https://panel.holoviz.org/reference/widgets/Select.html>
* <https://www.tutorialspoint.com/python_pandas/python_pandas_panel.htm>
* <https://www.kaggle.com/datasets>
* <https://youtu.be/uhxiXOTKzfs>