**Analysis of Global Data on Sustainable Energy**

**Introduction**

This project involves the analysis of a comprehensive dataset on global sustainable energy. The dataset includes various metrics related to renewable energy, access to electricity, and environmental impacts across different countries and regions. The goal of this project is to explore the data, uncover trends, and generate insights that could inform policy decisions and sustainable energy practices globally.

**Dataset Overview**

* **Source**:

The dataset used in this project is [Global Data on Sustainable Energy]

(<https://www.kaggle.com/datasets/anshtanwar/global-data-on-sustainable-energy>)

from Kaggle.

This comprehensive dataset showcases sustainable energy indicators and other useful factors across all countries from 2000 to 2020, vital aspects such as electricity access, renewable energy, carbon emissions, energy intensity, financial flows, and economic growth.

* **Data Dictionary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **Field** | **Variable** | **FIELD DESCRIPTION** | **VARIABLE TYPE** |
| 1 | Country | country\_name | The name of the country or region for which the data is reported. | object |
| 2 | Year | year | The year for which the data is reported, ranging from 2000 to 2020. | int64 |
| 3 | Access to electricity (% of population) | access\_to\_electricity\_by\_population | The percentage of population with access to electricity. | float64 |
| 4 | Access to clean fuels for cooking | access\_to\_clean\_fuels | The percentage of the population with primary reliance on clean fuels. | float64 |
| 5 | Renewable-electricity-generating-capacity-per-capita | renewable\_electricity\_generating\_capacity\_per\_capita | Installed Renewable energy capacity per person. | float64 |
| 6 | Financial flows to developing countries | financial\_flows\_to\_developing\_countries\_usd | Aid and assistance from developed countries for clean energy projects. | float64 |
| 7 | Renewable energy share in total final energy consumption | renewable\_energy\_share\_% | Percentage of renewable energy in final energy consumption. | float64 |
| 8 | Electricity from fossil fuels | electricity\_from\_fossil\_fuels | Electricity generated from fossil fuels (coal, oil, gas) in terawatt-hours. | float64 |
| 9 | Electricity from nuclear | electricity\_from\_nuclear | Electricity generated from nuclear power in terawatt-hours. | float64 |
| 10 | Electricity from renewables | electricity\_from\_renewables | Electricity generated from renewable sources (hydro, solar, wind, etc.) in terawatt-hours. | float64 |
| 11 | Low-carbon electricity | electricity\_from\_low\_carbon | Percentage of electricity from low-carbon sources (nuclear and renewables). | float64 |
| 12 | Primary energy consumption per capita | primary\_energy\_consumption\_per\_capita | Energy consumption per person in kilowatt-hours. | float64 |
| 13 | Energy intensity level of primary energy | energy\_intensity\_level | Energy use per unit of GDP at purchasing power parity. | float64 |
| 14 | Value\_co2\_emissions (metric tons per capita) | value\_co2\_emissions | Carbon dioxide emissions per person in metric tons. | float64 |
| 15 | Renewables | renewables\_% | Equivalent primary energy that is derived from renewable sources. | float64 |
| 16 | GDP growth | gdp\_growth | Annual GDP growth rate based on constant local currency. | float64 |
| 17 | GDP per capita | gdp\_per\_capita | Gross domestic product per person. | float64 |
| 18 | Density | population\_density | Population density in persons per square kilometer. | object |
| 19 | Land Area | land\_area\_km^2 | Total land area in square kilometers. | float64 |
| 20 | Latitude | latitude | Latitude of the country's centroid in decimal degrees. | float64 |
| 21 | Longitude | longitude | Longitude of the country's centroid in decimal degrees. | float64 |

* **Problem Statement**

We will use the available data to perform the following

* + 1. Exploratory Data Analysis: Identify the trends or patterns behind our current climate conditions.
    2. Carbon Emission analysis: study CO2 emissions, support climate strategies.
* **Data Preprocessing**

Data Normalization / Cleaning :

* + 1. *#drop all duplicate countries and append them to a list*
    2. df\_countries\_list = df1['Entity'].drop\_duplicates().to\_list()
    3. ['Afghanistan', 'Albania', 'Algeria', 'Angola', 'Antigua and Barbuda', 'Argentina', 'Armenia', …..]
    4. Merging columns from another source
* *#reading the 2nd file for country\_code and region*
* country\_code\_df = pd.read\_csv('country\_codes.csv'
  + 1. Dropped unnecessary columns and retained required columns.
    2. Checked for different country names in new dataset and renamed them accordingly to match the main dataset.
    3. *#renaming countries to match the origin dataset*
* country\_code\_df = country\_code\_df.replace('United States of America', 'United States')
* country\_code\_df = country\_code\_df.replace('United Kingdom of Great Britain and Northern Ireland', 'United Kingdom')
* country\_code\_df = country\_code\_df.replace('Türkiye', 'Turkey')
* country\_code\_df = country\_code\_df.replace('Netherlands, Kingdom of the', 'Netherlands')
  + 1. Ran a check to match countries in both datasets and dropping the missing ones.
    2. Created new columns for country\_code and region in df1
    3. Looped through df1 and country\_code\_df. If the country was a match, updated rows with country\_code and region into df1.
    4. *Loop through `df` and update `country\_code` and `region`*

for index, row in df1.iterrows():

match = valid\_countries[valid\_countries['name'] == row['Entity']]

if not match.empty:

df1.at[index, 'country\_code'] = match['alpha-3'].values[0]

df1.at[index, 'region'] = match['region'].values[0]

* **Identifying the missing values**

pip install plotly

import plotly.express as px

nan\_count = df.isna().sum().reset\_index()

nan\_count.columns = ['Column', 'Missing Values']

fig = px.bar(nan\_count,

x='Column', y='Missing Values', labels={'x': 'Columns', 'y': 'Missing Values Count'}, title='Missing Values in Each Column', height=600)

fig. show()

A graph with blue and white text

Description automatically generated

* **Imputing the missing values**

Considerations

* While imputing missing data for certain fields, we will need to use “MEAN” ( Average ) values. However, the MEAN value we calculate should be of that specific entity ( country ) over the years, rather than taking the MEAN value for the entire column.
* The values of each column was grouped by ‘Entity’ (country).
* Using a loop, mean for the column BY country was imputed and missing values were filled.
* However, for two columns namely 'Financial flows to developing countries (US $)' and 'Renewables (% equivalent primary energy)'], the missing values were not replaced by mean. Upon eye-balling, it was noticed that these records belonged to developed countries that were not receiving financial aid, hence we replaced the null values by ‘zero’ rather then mean to prevent our data from being skewed.

*# Specify the columns you want to leave with NaN values*

columns\_to\_leave = ['Financial flows to developing countries (US $)', 'Renewables (% equivalent primary energy)']

*# Group the DataFrame by 'country'*

grouped = df1.groupby('Entity')

*# Iterate over each column*

for column in df1.columns:

*# Exclude the specified columns*

if column not in columns\_to\_leave:

*# Check if there are NaNs and if the column is numeric*

if df1[column].isna().any() and df1[column].dtype in [np.float64, np.int64]:

*# Fill NaNs with the mean calculated within each group*

df1[column] = grouped[column].transform(lambda x: x.fillna(x.mean()))

print(df1[column])

grouped = df1.groupby('Entity')

*# Iterate over each column*

for column in df1.columns:

if df1[column].isna().any():

df1[column] = grouped[column].transform(lambda x: x.fillna(0))

Run a final check for missing values

*#final check for missing values*

df1.isna().sum()

* **Entity Relation Diagram**

A screenshot of a computer

Description automatically generated

# **Renaming columns according to ER Diagram**

*#renaming column names*

df1 = df1.rename(columns={'Entity':'country\_name', 'Year':'year',

'Access to electricity (% of population)':'access\_to\_electricity\_by\_population',

'Access to clean fuels for cooking':'access\_to\_clean\_fuels',

'Renewable-electricity-generating-capacity-per-capita':'renewable\_electricity\_generating\_capacity\_per\_capita',

'Financial flows to developing countries (US $)':'financial\_flows\_to\_developing\_countries\_usd',

'Renewable energy share in the total final energy consumption (%)':'renewable\_energy\_share\_%',

'Electricity from fossil fuels (TWh)':'electricity\_from\_fossil\_fuels',

'Electricity from nuclear (TWh)':'electricity\_from\_nuclear',

'Electricity from renewables (TWh)':'electricity\_from\_renewables',

'Low-carbon electricity (% electricity)':'electricity\_from\_low\_carbon',

'Primary energy consumption per capita (kWh/person)':'primary\_energy\_consumption\_per\_capita',

'Energy intensity level of primary energy (MJ/$2017 PPP GDP)':'energy\_intensity\_level',

'Value\_co2\_emissions\_kt\_by\_country':'value\_co2\_emissions',

'Renewables (% equivalent primary energy)':'renewables\_%',

'Density\_P/Km2':'population\_density',

'Land Area\_Km2':'land\_area\_km^2',

'Latitude':'latitude', 'Longitude':'longitude'})

# **Creating tables using DuckDB**

* Total of 5 tables to be created:
  + Year
  + Countries
  + Environmental
  + Economic
  + Energy
* Subset the dataset for required columns for each table.
* Create an id for each table

Creating indexes for each respective table

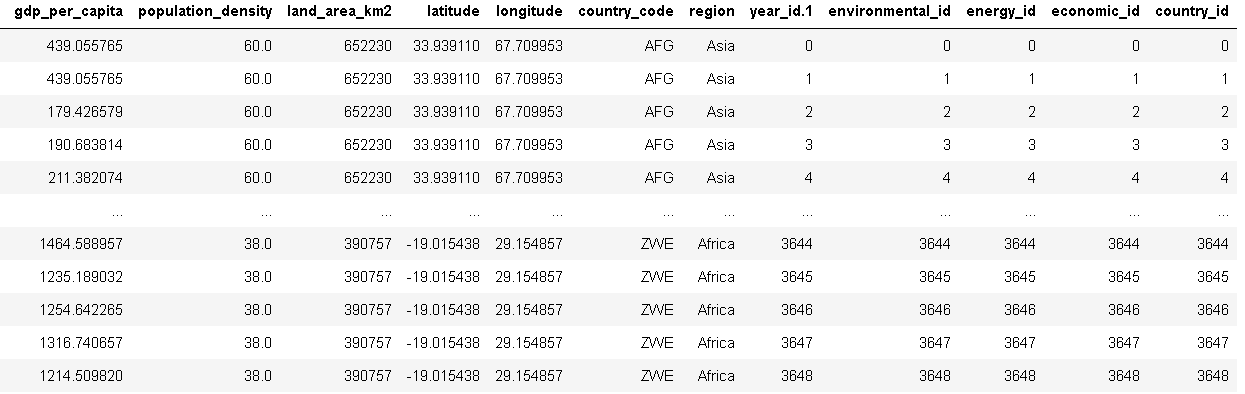
#adding in indexes for each table which is to be created.

df1['environmental\_id'] = df1.index

df1['energy\_id'] = df1.index

df1['economic\_id'] = df1.index

df1['country\_id'] = df1.index



# **Utilizing DuckDB to create a database**

* Install duckdb to be used as a DBMS
* Create tables based off ER Diagram
* Run SQL queries

Creating tables for years, environmental\_metrics

#creating tables for years and environmental\_metrics with primary key and foreign key allocation

con.sql("CREATE TABLE years (year\_id INTEGER PRIMARY KEY, year INTEGER)");

con.sql("CREATE TABLE environmental\_metrics (environmental\_id INTEGER PRIMARY KEY, year\_id INTEGER REFERENCES years(year\_id))");

#inserting of data from dataframe

con.sql("INSERT INTO years (SELECT year\_id, year FROM df1)")

#adding in the columns for environmental\_tables

con.sql("ALTER TABLE environmental\_metrics ADD COLUMN value\_co2\_emissions FLOAT")

con.sql("ALTER TABLE environmental\_metrics ADD COLUMN renewables\_percent FLOAT")

con.sql("ALTER TABLE environmental\_metrics ADD COLUMN primary\_energy\_consumption\_per\_capita FLOAT")

con.sql("ALTER TABLE environmental\_metrics ADD COLUMN renewable\_energy\_share\_percent FLOAT")

con.sql("ALTER TABLE environmental\_metrics ADD COLUMN country\_id INTEGER")

con.sql("INSERT INTO environmental\_metrics (SELECT environmental\_id, year\_id, value\_co2\_emissions, renewables\_percent, primary\_energy\_consumption\_per\_capita, renewable\_energy\_share\_percent, country\_id FROM df1)")

Creating countries table

con.sql("CREATE TABLE countries (country\_id INTEGER PRIMARY KEY, country\_name TEXT , country\_code TEXT, region TEXT, latitude FLOAT, longitude FLOAT, land\_area\_km2 FLOAT, population\_density FLOAT)")

con.sql("INSERT INTO countries (SELECT country\_id, country\_name, country\_code, region, latitude, longitude, land\_area\_km2, population\_density FROM df1)")

con.sql("CREATE TABLE energy\_metrics (energy\_id INTEGER PRIMARY KEY, year\_id INTEGER REFERENCES years(year\_id), country\_id INTEGER REFERENCES countries(country\_id))");

con.sql("ALTER TABLE energy\_metrics ADD COLUMN access\_to\_electricity\_by\_population FLOAT");

con.sql("ALTER TABLE energy\_metrics ADD COLUMN access\_to\_clean\_fuels FLOAT");

con.sql("ALTER TABLE energy\_metrics ADD COLUMN electricity\_from\_fossil\_fuels FLOAT");

con.sql("ALTER TABLE energy\_metrics ADD COLUMN electricity\_from\_nuclear FLOAT");

con.sql("ALTER TABLE energy\_metrics ADD COLUMN electricity\_from\_renewables FLOAT");

con.sql("ALTER TABLE energy\_metrics ADD COLUMN electricity\_from\_low\_carbon FLOAT");

con.sql("ALTER TABLE energy\_metrics ADD COLUMN renewable\_electricity\_generating\_capacity\_per\_capita FLOAT");

con.sql("INSERT INTO energy\_metrics (SELECT energy\_id, year\_id, country\_id, access\_to\_electricity\_by\_population, access\_to\_clean\_fuels, electricity\_from\_fossil\_fuels, electricity\_from\_nuclear, electricity\_from\_renewables, electricity\_from\_low\_carbon, renewable\_electricity\_generating\_capacity\_per\_capita FROM df1)");

con.sql("CREATE TABLE economic\_metrics (economic\_id INTEGER PRIMARY KEY, year\_id INTEGER REFERENCES years(year\_id), country\_id INTEGER REFERENCES countries(country\_id))");

con.sql("ALTER TABLE economic\_metrics ADD COLUMN financial\_flows\_to\_developing\_countries\_usd FLOAT");

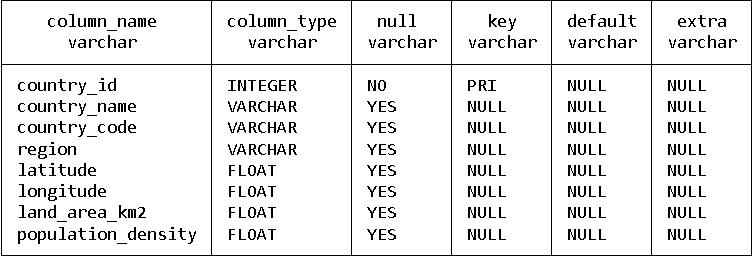
con.sql("ALTER TABLE economic\_metrics ADD COLUMN gdp\_growth FLOAT");

con.sql("ALTER TABLE economic\_metrics ADD COLUMN gdp\_per\_capita FLOAT");

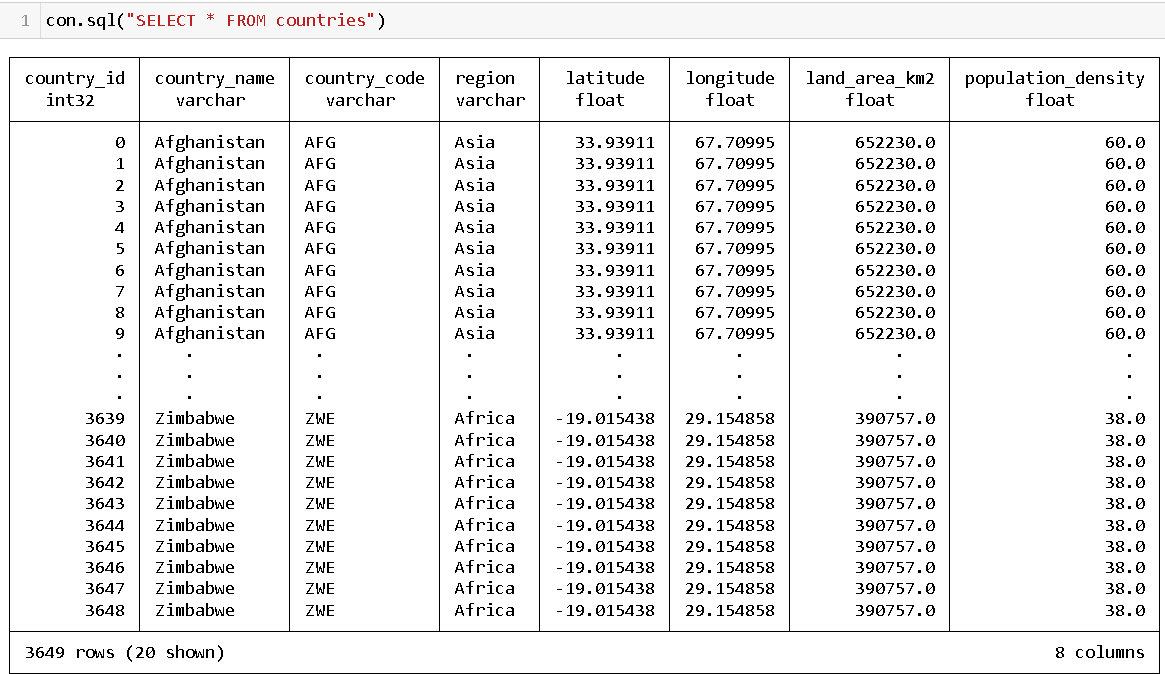
con.sql("ALTER TABLE economic\_metrics ADD COLUMN energy\_intensity\_level FLOAT");

con.sql("INSERT INTO economic\_metrics (SELECT economic\_id, year\_id, country\_id, financial\_flows\_to\_developing\_countries\_usd, gdp\_growth, gdp\_per\_capita, energy\_intensity\_level FROM df1)");

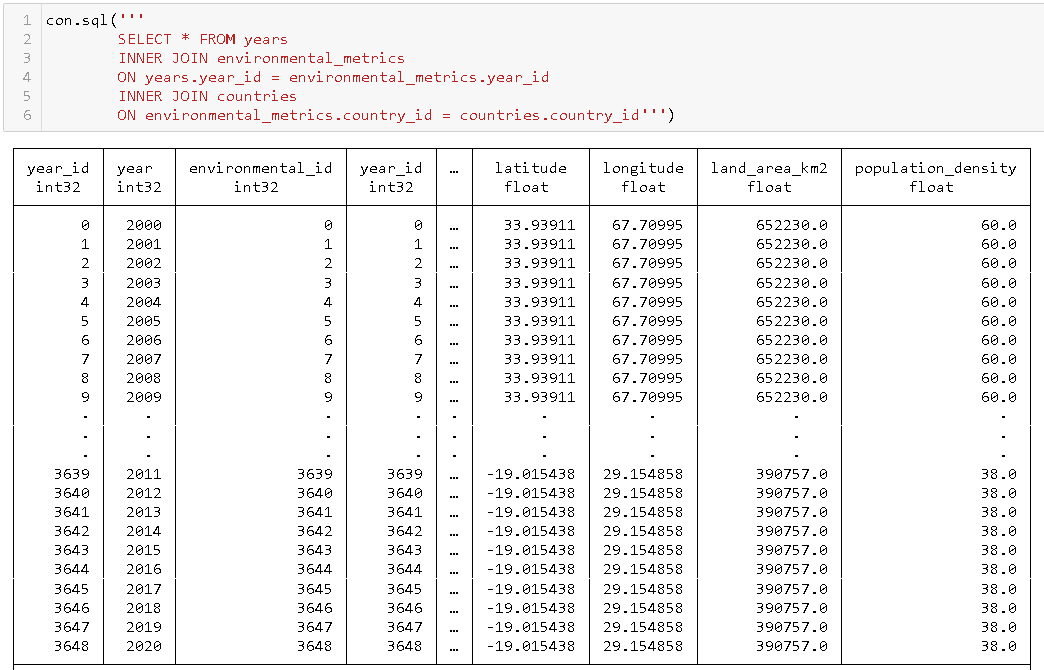
SHOW country\_table



SELECT ALL ROWS FROM COUNTRIES



INNER JOINS between countries, years and environmental\_metrics



Creating energy\_metrics table

con.sql("CREATE TABLE energy\_metrics (energy\_id INTEGER PRIMARY KEY, year\_id INTEGER REFERENCES years(year\_id), country\_id INTEGER REFERENCES countries(country\_id))");

con.sql("ALTER TABLE energy\_metrics ADD COLUMN access\_to\_electricity\_by\_population FLOAT");

con.sql("ALTER TABLE energy\_metrics ADD COLUMN access\_to\_clean\_fuels FLOAT");

con.sql("ALTER TABLE energy\_metrics ADD COLUMN electricity\_from\_fossil\_fuels FLOAT");

con.sql("ALTER TABLE energy\_metrics ADD COLUMN electricity\_from\_nuclear FLOAT");

con.sql("ALTER TABLE energy\_metrics ADD COLUMN electricity\_from\_renewables FLOAT");

con.sql("ALTER TABLE energy\_metrics ADD COLUMN electricity\_from\_low\_carbon FLOAT");

con.sql("ALTER TABLE energy\_metrics ADD COLUMN renewable\_electricity\_generating\_capacity\_per\_capita FLOAT");

con.sql("INSERT INTO energy\_metrics (SELECT energy\_id, year\_id, country\_id, access\_to\_electricity\_by\_population, access\_to\_clean\_fuels, electricity\_from\_fossil\_fuels, electricity\_from\_nuclear, electricity\_from\_renewables, electricity\_from\_low\_carbon, renewable\_electricity\_generating\_capacity\_per\_capita FROM df1)");

Creating economics\_table

con.sql("CREATE TABLE economic\_metrics (economic\_id INTEGER PRIMARY KEY, year\_id INTEGER REFERENCES years(year\_id), country\_id INTEGER REFERENCES countries(country\_id))");

con.sql("ALTER TABLE economic\_metrics ADD COLUMN financial\_flows\_to\_developing\_countries\_usd FLOAT");

con.sql("ALTER TABLE economic\_metrics ADD COLUMN gdp\_growth FLOAT");

con.sql("ALTER TABLE economic\_metrics ADD COLUMN gdp\_per\_capita FLOAT");

con.sql("ALTER TABLE economic\_metrics ADD COLUMN energy\_intensity\_level FLOAT");

con.sql("INSERT INTO economic\_metrics (SELECT economic\_id, year\_id, country\_id, financial\_flows\_to\_developing\_countries\_usd, gdp\_growth, gdp\_per\_capita, energy\_intensity\_level FROM df1)");

# **Conclusion:**

In line with our problem statement, we discovered the trend that less developed regions (economic) and countries tend to have lower greenhouse gas emissions. Over the years, there is an increasing adoption in the use of renewables. Additionally, it is noteworthy that while country such as Iceland have one of the highest energy consumptions in the world, most of its energy production stems from geothermal and hydropower renewable resources. Thus, environmental factors will include policies, working together with other countries, R&D and capitalising on geographical resources (if available).