Analyzing the Impact of CO2 Emissions on Temperature Changes

Main Question

How do CO2 emissions correlate with temperature changes over the world?

Data Sources

Datasource 1: FAO Temperature Change

- **Why Chosen:** The dataset provides annual temperature change for various countries and it's from trustworthy source.
- **Source:** Food and Agriculture Organization (FAO)
- **Data Contains:** The FAOSTAT Temperature change on land domain disseminates statistics of mean surface temperature change by country.
- **Metadata URL:** FAO Temperature Change Metadata
- **Data URL:** FAO Temperature Change Data
- **Data Type:** CSV
- **Data Structure and Quality:** The Data is maintained by FAO thus quality can be trusted, The dataset is structured in CSV with columns like years , temperature e.t.c
- **License:** CC BY-NC-SA 3.0 IGO
- **Citation:** Food and Agriculture Organization of the United Nations. (2023). FAOSTAT: Temperature Change. Retrieved from FAOSTAT Temperature Change
- **License Compliance:** The dats is permissible to be utilize for non commercial purposes.

Datasource 2: World Bank CO2 Emissions Dataset

- **Why Chosen:** This dataset offers extensive CO2 emissions data yearly
- **Source:** World Bank
- **Data Contains:** CO2 emissions data (in kilotons) for various countries.
- **Metadata URL:** World Bank CO2 Emissions Metadata
- **Data URL:** World Bank CO2 Emissions Data
- **Data Type:** CSV
- **Data Structure and Quality:** CSV format with columns for country, year, and CO2 emission values. High-quality data from a reputable source.
- **License:** Attribution-NonCommercial 4.0 International (CC BY-NC 4.0)
- **Citation:** World Bank. (2023). CO2 emissions (kt). Retrieved from World Bank CO2 Emissions

- **License Compliance:** The dats is permissible to be utilize for non commercial purposes.

Loading Data and Initial Inspection

Displaying the data sets after applying transformations on it

In [39]:
```
import pandas as pd

# get data from temperature table and display few samples
temp_df = pd.read_sql_table('temperature', 'sqlite:///../data/pipelineDB.
temp_df.head(5)
```

Out[39]:

	Area	Year	Change
0	Afghanistan	1961	0.023667
1	Afghanistan	1962	-0.282250
2	Afghanistan	1963	0.854000
3	Afghanistan	1964	-1.003250
4	Afghanistan	1965	0.011833

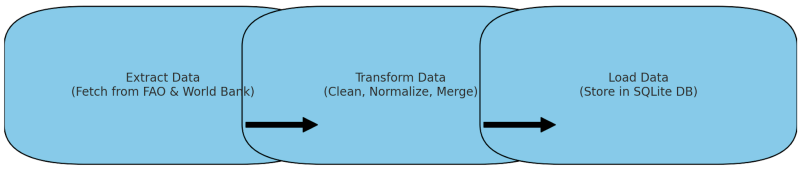
In [40]:
```
# get data from CO2 emissions table  and display few samples

CO2_df = pd.read_sql_table('CO2_emissions', 'sqlite:///../data/pipelineDB
CO2_df.head()
```

Out[40]:

	Area	country_code	Year	co2_emissions
0	Africa Eastern and Southern	AFE	1990	304614.720181
1	Afghanistan	AFG	1990	2046.870000
2	Africa Western and Central	AFW	1990	97190.345000
3	Angola	AGO	1990	6564.200000
4	Albania	ALB	1990	6060.500000

DATA PIPELINE



Data Pipeline Description

In order to study the effect of CO2 emissions on temperature variations, the (ETL) process is automated by the data pipeline for this project.

Technology Utilized

- **Extraction:** Python libraries that download and manage World Bank and FAO datasets via the `requests` , `Pandas` library.
- **Transform:** Pandas library for appplying meanigful transformation, normalization, and data cleaning.
- **Load:** A SQLite database is being loaded from the transformed data in order to store the final data

Steps:

1. **Extract Data**
    - **Description:** The pipeline pulls data from World Bank and FAO database sources. After downloading and extracting the zip file containing both datasets.
    - **Result:** The raw data files are first downloded locally using the `requests` library and kept in data directory with files named `fao_data.zip` and `world_bank_data.zip`
2. **Transform Data**
    - **Description:** The data from the sources are changed normalized with meanigful transformations for consistancy. Both datasets have been combined using common keys in them like `year` and `Area` . All required data cleaning techniques are carried out, including managing missing values and standardizing data formats.
    - **Result:** After the data is combined and tranformed, a merged table called `tempCO2` is produced.
    - **Steps in Transformation:**

    - **Convert FAO Temperature Information:**
        - Only keeping the records where there is a temperature variation

          remove standard daviation category.
        - Eliminate not useful columns like "Area Code," "Area Code (M49)," "Element Code," "Months Code," "Unit," and "Element as these columns have no real purpose in the analysis"
        - Use `pd.melt()` to reshape the data from wide to long format as it will be useful to time series analysis.
        - Eliminating 'Y' from column name as it is ambeded in years e.g Y1998 -> 1998.
        - Compile annual data by calculating the average annual temperature change from given months.
    - **Transform World Bank CO2 Data:**
        - Rename columns (such as "Country Name" to "Area") to maintain consistency between datasets.
        - Eliminate columns like "Indicator Code" and "Indicator Name as they they serve no usecase in our analysis"
        - Use `pd.melt()` to reshape the data from wide to long format useful for time series analysis.
        - Remove records where the CO2 emission numbers are absent.
        - Convert the 'year' and 'co2_emissions' data types to achive consistancy.
        - Rename column names to keep everything uniform between data sources.
3. **Load Data**

    - **Description:** SQLite database is used to load the transformed data .
    - **Output:** An SQLite database (`pipelineDB.sqlite`) with a tables (`tempCO2`) (`temperature`) and (`CO2_emissions`) containing the transformed data.

Problems Encountered and Solutions

- **Problem:** Data formats and columns vary throughout the two diffrent datasets .
    - **Solution:** To guarantee consistency and make the dataset mergable, standardized data formats and column names were used during the transformation stage.
- **Problem:** missing values in important columns.
    - **Solution:** Either eliminated records or used imputation techniques to fill in missing values.

Error Handling and Changing Input Data

- **Error Handling:**
    - **Logging:** logging was implemented to catch any errors or debug later for potential issues.
    - **exceptions:** Try catch exception handlers were used throughout the code to prevent breakage
- **Changing Input Data:**
    - **Flexible Parsing:** used flexible data processing techniques to allow for small format changes in data without causing pipeline flow.

Results and Limitations

Output Data

- **Description:** The sqlite dataset containing tables comprising tranformed data on annual temperature changes and CO2 emissions is the pipeline's output.
- **Format:** The merged data is kept in a table called `tempCO2` in a SQLite database called `pipelineDB.sqlite` .

In [41]:
```
# display merged table results

merged_df = pd.read_sql_table('tempCO2', 'sqlite:///../data/pipelineDB.sq
merged_df.head()
```

Out[41]:

	Area	Year	Change	country_code	co2_emissions
0	Afghanistan	1990	0.714000	AFG	2046.87
1	Afghanistan	1991	0.138333	AFG	1941.37
2	Afghanistan	1992	-0.185917	AFG	1525.47
3	Afghanistan	1993	0.163000	AFG	1527.89
4	Afghanistan	1994	0.469667	AFG	1493.59

Data Quality

- **Consistency:** To make the dataset consistent , the data has been standardized and normalized using various techniques.
- **Completeness:** Imputation or removal of missing values in data sources has been taken care.

Accuracy: A high level of accuracy can be expected as The data comes from reliable sources (the World Bank and the FAO),

Critical Reflection and Potential Issues

- **Availability of Data:** The datasets might not comprehensively cover all countries or wide range or years , which could result in analytical gaps. This might have an impact on how thorough the findings are.
- **Quality of Data:** Different countries may have reported or documented data in inconsistent or using difrrent measures thus resulting in inconsitancy in data.
- **Resolution Time:** It is not possible to do short term analysis as the data set offers annual data.
- **Application and Use:** Non commercial usage on data sources restricts the data sets to be used for commercial application ideas