```
# -*- coding: utf-8 -*-
"""
Created on Wed Dec 13 14:12:32 2023

@author: Hari Ghanapuram
"""

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

def read_and_transform_data(filename):
    encoding = 'ISO-8859-1'

    # Read the data
    df = pd.read_csv(filename, encoding=encoding)

    # Transpose the DataFrame to have years as columns
    df_years = df.set_index(['Series Name', 'Country Name', 'Country
Code', 'Series Code']).T

    # Clean the transposed DataFrame
    df_years.columns = df_years.columns.droplevel(0)

    # Transpose the DataFrame again to have countries as columns
    df_countries = df_years.T

    return df_years, df_countries

filename = "data.csv"
df_years, df_countries = read_and_transform_data(filename)

df = pd.read_csv(filename,encoding = 'ISO-8859-1')

# Selecting a few indicators and countries for exploration
selected_columns = ['Country Name', '2013', '2014', '2015', '2016',
'2017', '2018', '2019']
selected_countries = ['Australia', 'India', 'United States', 'China']

# Subsetting the DataFrame
subset_df = df[df['Country
Name'].isin(selected_countries)][selected_columns]

# Display summary statistics
summary_statistics = subset_df.describe()
print(summary_statistics)

# Selecting columns for correlation analysis
correlation_columns = ['2013', '2014', '2015', '2016', '2017', '2018',
'2019']

# Subsetting the DataFrame for correlation analysis
correlation_df = df[df['Country
Name'].isin(selected_countries)][correlation_columns]

# Calculating correlations
correlation_matrix = correlation_df.corr()

# Displaying the correlation matrix
print("Correlation Matrix:")
print(correlation_matrix)

import seaborn as sns
```

```
selected_columns = ['Series Name', 'Country Name', '2013', '2014',
'2015', '2016', '2017', '2018', '2019', '2020', '2021', '2022']
df_selected = df[selected_columns]

grouped_df = df_selected.groupby(['Series Name', 'Country Name']).sum()

# Let's plot the trends for each indicator and country
for series_name, group in grouped_df.groupby('Series Name'):
    plt.figure(figsize=(12, 6))
    plt.title(f'Trend of {series_name} (2013-2022) for Each Country')

    for country_name, country_data in group.groupby('Country Name'):
        plt.plot(country_data.columns[1:],
country_data.values.flatten()[1:], label=country_name)

    plt.xlabel('Year')
    plt.ylabel('Value')
    plt.legend()
    plt.show()

    plt.show()

import numpy as np

selected_columns = ['Series Name', 'Country Name', '2013', '2014',
'2015', '2016', '2017', '2018', '2019', '2020', '2021', '2022']
df_selected = df[selected_columns]


grouped_df = df_selected.groupby(['Series Name', 'Country Name']).sum()


grouped_df_transposed = grouped_df.T


for series_name, series_data in grouped_df.groupby('Series Name'):
    plt.figure(figsize=(34, 8))
    bar_width = 0.1
    index = np.arange(len(grouped_df_transposed.index))

    for i, country_name in
enumerate(series_data.index.get_level_values('Country Name').unique()):
        plt.bar(index + i * bar_width, grouped_df_transposed[series_name,
country_name], width=bar_width, label=country_name)

    plt.xlabel('Year')
    plt.ylabel('Value')
    plt.title(f'Comparison of {series_name} Across Countries (2013-
2022)')
    plt.xticks(index + bar_width *
((len(series_data.index.get_level_values('Country Name').unique()) - 1) /
2),
               grouped_df_transposed.index)
    plt.legend()
    plt.show()

selected_columns = ['Series Name', 'Country Name', '2013', '2014',
'2015', '2016', '2017', '2018', '2019', '2020', '2021', '2022']
df_selected = df[selected_columns]

# Filter data for CO2 emissions and GDP
co2_data = df_selected[df_selected['Series Name'] == 'CO2 emissions
(kt)']
gdp_data = df_selected[df_selected['Series Name'] == 'GDP (current US$)']
```

```
# Merge CO2 and GDP data on 'Country Name'
merged_data = pd.merge(co2_data, gdp_data, on='Country Name',
suffixes=('_CO2', '_GDP'))

# Extract numerical columns for correlation matrix
numerical_columns = merged_data.columns[2:]

# Calculate the correlation matrix
correlation_matrix = merged_data[numerical_columns].corr()

# Create a heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f",
linewidths=.5)
plt.title('Correlation Matrix between CO2 Emissions and GDP')
plt.show()

selected_columns = ['Series Name', 'Country Name', '2013', '2014',
'2015', '2016', '2017', '2018', '2019', '2020', '2021', '2022']
df_selected = df[selected_columns]

# Filter data for the United States
us_data = df_selected[df_selected['Country Name'] == 'United States']


us_data = us_data.drop(columns=['Country Name'])

# Set the 'Series Name' column as the index
us_data = us_data.set_index('Series Name')

# Calculate the correlation matrix
correlation_matrix = us_data.T.corr()

# Create a heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f",
linewidths=.5)
plt.title('Correlation Matrix for Indicators in the United States (2013-
2022)')
plt.show()

selected_columns = ['Series Name', 'Country Name', '2013', '2014',
'2015', '2016', '2017', '2018', '2019', '2020', '2021', '2022']
df_selected = df[selected_columns]

# Filter data for the India
us_data = df_selected[df_selected['Country Name'] == 'India']


us_data = us_data.drop(columns=['Country Name'])

# Set the 'Series Name' column as the index
us_data = us_data.set_index('Series Name')

# Calculate the correlation matrix
correlation_matrix = us_data.T.corr()

# Create a heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='inferno', fmt=".2f",
linewidths=.5)
plt.title('Correlation Matrix for Indicators in the India (2013-2022)')
plt.show()
```

```
selected_columns = ['Series Name', 'Country Name', '2013', '2014',
'2015', '2016', '2017', '2018', '2019', '2020', '2021', '2022']
df_selected = df[selected_columns]

# Filter data for the India
us_data = df_selected[df_selected['Country Name'] == 'Australia']


us_data = us_data.drop(columns=['Country Name'])

# Set the 'Series Name' column as the index
us_data = us_data.set_index('Series Name')

# Calculate the correlation matrix
correlation_matrix = us_data.T.corr()

# Create a heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='viridis', fmt=".2f",
linewidths=.5)
plt.title('Correlation Matrix for Indicators in the Australia (2013-
2022)')
plt.show()
```