# import all needed libraries

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

# import the cleaned dataset from a csv file

data = pd.read\_csv(r'data\_cleaned.csv')

print("Shape of the dataset:")

data.shape

print("Descriptive statistics:")

data.describe().T

# Group by Year and calculate mean CO₂ per capita

df\_yearly\_avg = data.groupby('year')['co2\_per\_cap'].mean().reset\_index()

# Plot

plt.figure(figsize=(10, 6))

sns.lineplot(data=data, x='year', y='co2\_per\_cap', marker='o')

plt.title('Global Average CO₂ Emissions per Capita Over Time')

plt.ylabel('CO₂ per Capita (metric tons)')

plt.xlabel('Year')

plt.grid(True)

plt.tight\_layout()

plt.show()

# Plot

plt.figure(figsize=(10, 6))

sns.lineplot(data=data, x='pop', y='co2\_ttl', marker='^')

plt.title('Total CO₂ Emissions against Population')

plt.ylabel('Total CO₂ emissions')

plt.xlabel('Population')

plt.grid(True)

plt.tight\_layout()

plt.show()

# create a column for the total energy use

data['en\_ttl'] = data['en\_per\_gdp'] \* data['gdp'] /1000

# select all features

features\_all = data[['country','year','cereal\_yield','fdi\_perc\_gdp','gni\_per\_cap',

                     'en\_per\_gdp', 'en\_per\_cap', 'en\_ttl', 'co2\_ttl', 'co2\_per\_cap',

                     'co2\_per\_gdp', 'pop\_urb\_aggl\_perc', 'prot\_area\_perc', 'gdp',

                     'pop\_growth\_perc', 'pop', 'urb\_pop\_growth\_perc']]

# plot a correlation of all features

# correlation matrix

sns.set\_theme(font\_scale=2)

f,ax=plt.subplots(figsize=(30,20))

sns.heatmap(features\_all.drop(['country'], axis=1).corr(), annot=True, cmap='coolwarm', fmt = ".2f",

            center=0, vmin=-1, vmax=1)

plt.title('Correlation between features', fontsize=25, weight='bold' )

plt.show()

sns.set\_theme(font\_scale=1)

features\_for\_vif = data[['cereal\_yield','fdi\_perc\_gdp','gni\_per\_cap', 'en\_per\_cap', 'co2\_per\_cap',

                     'pop\_urb\_aggl\_perc', 'prot\_area\_perc', 'gdp',  'pop\_growth\_perc', 'urb\_pop\_growth\_perc']]

vif\_data = pd.DataFrame()

vif\_data["feature"] = features\_for\_vif.columns # Use the columns from the features\_for\_vif DataFrame

vif\_data["VIF"] = [variance\_inflation\_factor(features\_for\_vif.values, i)

                   for i in range(features\_for\_vif.shape[1])] # Use the values and number of columns from features\_for\_vif

print(vif\_data)

features = features\_all[['cereal\_yield','fdi\_perc\_gdp','gni\_per\_cap', 'en\_per\_cap', 'co2\_per\_cap',

                     'pop\_urb\_aggl\_perc', 'prot\_area\_perc', 'gdp',  'pop\_growth\_perc', 'urb\_pop\_growth\_perc']]

# a dictionary with feature labels

labels\_dict = {'gni\_per\_cap':'GNI per capita [Atlas $]',

               'gdp':'Gross Domestic Product [$]',

               'cereal\_yield':'Cereal yield [kg/ha]',

               'prot\_area\_perc': 'Nationally terrestrial protected areas [% of total land area]',

               'fdi\_perc\_gdp': 'Foreign Direct Investment [% of GDP]',

               'pop\_urb\_aggl\_perc': 'Population in urban agglomerations > 1mln [%]',

               'urb\_pop\_growth\_perc':'Urban population growth [annual %]',

               'pop\_growth\_perc': 'Population growth [annual %]',

               'co2\_per\_cap':'CO2 emissions per capita [t]',

               'en\_per\_cap':'Energy use per capita [kg oil eq]' }

# get unique values in country column

unique\_countries = data['country'].unique()

unique\_countries

# Select countries to compare

selected\_countries = ['IND', 'USA', 'PAK', 'RUS', 'NZL']

# Use the correct DataFrame variable 'data' and column names 'country', 'year', and 'co2\_per\_cap'

df\_selected = data[data['country'].isin(selected\_countries)]

# Plot

plt.figure(figsize=(12, 6))

# Use the correct DataFrame variable 'df\_selected' and column names 'year' and 'co2\_per\_cap' for plotting

sns.lineplot(data=df\_selected, x='year', y='co2\_per\_cap', hue='country', marker='o')

plt.title('CO₂ Emissions per Capita Over Time (Selected Countries)')

plt.ylabel('CO₂ per Capita (metric tons)')

plt.xlabel('Year')

plt.legend(title='Country')

plt.grid(True)

plt.tight\_layout()

plt.show()

# select only rows for half of the countries chosen randomly in order to ensure better visibility

chosen\_countries=['IND', 'LMC', 'LMY', 'MAR', 'MEX', 'MIC', 'MNA', 'MOZ', 'MYS',

'NGA', 'NLD', 'NZL', 'PAK', 'PAN', 'PER', 'PHL', 'PRT', 'PRY',

'ROM', 'SAS', 'SAU', 'SDN', 'SEN', 'SLV', 'SSA', 'SWE', 'SYR',

'TGO', 'THA', 'TUR', 'TZA', 'UMC', 'URY', 'USA', 'VEN', 'VNM',

'WLD', 'ZAF', 'ZAR', 'ZMB', 'ECA', 'POL', 'RUS', 'UKR', 'YEM',

'ETH', 'BEL']

features\_chosen = features\_all[features\_all['country'].isin(chosen\_countries)]

sns.set\_theme(font\_scale=1.3)

sns.pairplot(data=features\_chosen, hue='country')

# choose features and label columns

feature\_cols = ['country', 'cereal\_yield','fdi\_perc\_gdp','gni\_per\_cap', 'en\_per\_cap', 'pop\_urb\_aggl\_perc',

                    'prot\_area\_perc', 'gdp',  'pop\_growth\_perc', 'urb\_pop\_growth\_perc', 'co2\_per\_cap']

# Keep as DataFrame for filtering

features\_for\_plot = data[feature\_cols].copy() # Create a copy to avoid SettingWithCopyWarning

# remove the ARE outliers from the DataFrame used for plotting

features\_for\_plot = features\_for\_plot[features\_for\_plot['country']!='ARE']

# 4d plot

sns.set\_theme(style="whitegrid", font\_scale=2)

cmap = sns.cubehelix\_palette(rot=-.2, as\_cmap=True)

g = sns.relplot(

    data=features\_for\_plot, # Use the DataFrame specifically prepared for plotting

    x="urb\_pop\_growth\_perc", y="co2\_per\_cap",

    hue="en\_per\_cap", size="pop\_urb\_aggl\_perc",

    palette=cmap, sizes=(10, 200),

    height=10, aspect= 4/3

)

g.ax.set\_xlabel(labels\_dict['urb\_pop\_growth\_perc'])

g.ax.set\_ylabel(labels\_dict['co2\_per\_cap'])