import pandas as pd  
import numpy as np  
import random  
import matplotlib.pyplot as plt  
import matplotlib.cbook  
import zipfile  
import bz2  
import warnings  
warnings.filterwarnings("ignore")  
data = pd.read\_csv('https://raw.githubusercontent.com/PoojaSarkar02/PoojaSarkar02/main/co2-emissions-vs-gdp.csv')  
print("data.shape: ", data.shape)  
print("Sample Data: \n",data.head())  
print("Columns: \n",data.columns)  
*#How many UNIQUE country names are there ?*Entity = data['Entity'].unique().tolist()  
print("Number of countries: ",len(Entity))  
Code = data['Code'].unique().tolist()  
print("Number of country codes: ",len(Code))  
years = data['Year'].unique().tolist()  
print("Number of years: ",len(years))  
print(min(years)," to ",max(years))

population\_data = data['Population (historical estimates)']  
print(population\_data)

for other data

import pandas as pd  
import numpy as np  
import random  
import matplotlib.pyplot as plt  
import matplotlib.cbook  
import zipfile  
import bz2  
import warnings  
warnings.filterwarnings("ignore")  
data = pd.read\_csv(r'H:\WDIData\_T.csv')  
print("data.shape: ", data.shape)  
print("Sample Data: \n",data.head())  
print("Columns: \n",data.columns)  
*#How many UNIQUE country names are there ?*countries = data['CountryName'].unique().tolist()  
print("Number of countries: ",len(countries))  
*#How many unique country codes are there  
# #It should be the same as number of unique countries.*countryCodes = data['CountryCode'].unique().tolist()  
print("Number of country codes: ",len(countryCodes))  
*#Are there many indicators or few ?  
# #How many unique indicators are there?*indicators = data['IndicatorName'].unique().tolist()  
print("Number of indicators: ",len(indicators))  
*#How many years of data do we have ?*years = data['Year'].unique().tolist()  
print("Number of years: ",len(years))  
*#What's the range of years?*print(min(years)," to ",max(years))  
*################# ##  
# Data Visualization #################  
# #Let us pick a country and an indicator to explore CO2 Emissions per capita and the USA.  
# #To select CO2 emissions for the United States, We will take the intersection  
# of two masks, one with all the rows that contains the string,  
# "C02 emissions" and the other which contains all the rows containing the string, "USA".*hist\_indicator = 'CO2 emissions \\(metric'  
hist\_country = 'USA'  
mask1 = data['IndicatorName'].str.contains(hist\_indicator)  
mask2 = data['CountryCode'].str.contains(hist\_country)  
stage = data[mask1 & mask2]  
print (stage.shape)  
stage.head()  
print("Indicator Name: ", stage['IndicatorName'].iloc[0])  
*#Let us see how emissions have changed over time using MatplotLib*years = stage['Year'].values *# get the years*co2 = stage['Value'].values *# get the values*plt.bar(years,co2)  
plt.close('all')  
plt.show()  
  
*#Let us create a line plot.*plt.plot(stage['Year'].values, stage['Value'].values)  
plt.xlabel('Year')  
plt.ylabel(stage['IndicatorName'].iloc[0])  
plt.axis([1959, 2011,0,25])  
plt.close('all')  
plt.show()  
*#Using Histograms to explore the distribution of value*hist\_data = stage['Value'].values  
print(hist\_data)  
print(len(hist\_data))  
*# Histogram of the data*plt.hist(hist\_data, 10, density=False, facecolor='green') *# 10 is the number of bins*plt.xlabel(stage['IndicatorName'].iloc[0])  
plt.ylabel('# of Years')  
plt.title('Histogram Example')  
plt.grid(True)  
*#plt.close('all')*plt.show()  
*# select CO2 emissions for all countries in 2011*hist\_indicator = 'CO2 emissions \(metric'  
hist\_year = 2011  
mask1 = data['IndicatorName'].str.contains(hist\_indicator)  
mask2 = data['Year'].isin([hist\_year])  
co2\_2011 = data[mask1 & mask2]  
co2\_2011.head()  
print("CO2",(len(co2\_2011)))

*#Let us plot a histogram of the emmissions per capita by country*fig, ax = plt.subplots()  
ax.annotate("USA",xy=(18, 5), xycoords='data',xytext=(18, 30), textcoords='data', arrowprops=dict(arrowstyle="->",connectionstyle="arc3"))  
plt.hist(co2\_2011['Value'], 10, density=False, facecolor='green')  
plt.xlabel(stage['IndicatorName'].iloc[0])  
plt.ylabel('# of Countries')  
plt.title('Histogram of CO2 Emissions Per Capita')  
plt.grid(True)  
plt.show()

*#Relationship between GPD and CO2 Emissions in USA*import matplotlib.pyplot as plt  
hist\_indicator = 'GDP per capita \(constant 2005'  
hist\_country = 'USA'  
mask1 = data['IndicatorName'].str.contains(hist\_indicator)  
mask2 = data['CountryCode'].str.contains(hist\_country)  
gdp\_stage = data[mask1 & mask2]  
stage.head()  
print("Filtered Data:", gdp\_stage.head()) *# Check what data is being filtered*if not gdp\_stage.empty:  
 plt.plot(gdp\_stage['Year'].values, gdp\_stage['Value'].values)  
 plt.xlabel('Year')  
 plt.ylabel(gdp\_stage['IndicatorName'].iloc[0])  
 plt.title('GDP Per Capita USA')  
 plt.show()  
else:  
 print("No data found for the given filters.")  
  
  
  
  
*#ScatterPlot for comparing GDP against CO2 emissions (per capita)*print("GDP Min Year = ", gdp\_stage['Year'].min(), "max: ", gdp\_stage['Year'].max())  
print("CO2 Min Year = ", stage['Year'].min(), "max: ", stage['Year'].max())  
gdp\_stage\_trunc = gdp\_stage[gdp\_stage['Year'] < 2012]  
print(len(gdp\_stage\_trunc))  
print(len(stage))  
import matplotlib.pyplot as plt  
fig, axis = plt.subplots()  
axis.yaxis.grid(True)  
axis.set\_title('CO2 Emissions vs. GDP (per capita)',fontsize=10)  
axis.set\_xlabel(gdp\_stage\_trunc['IndicatorName'].iloc[0],fontsize=10)  
axis.set\_ylabel(stage['IndicatorName'].iloc[0],fontsize=10)  
X = gdp\_stage\_trunc['Value']  
Y = stage['Value']  
axis.scatter(X, Y)  
plt.show()

*#ScatterPlot for comparing GDP against CO2 emissions (per capita)*print("GDP Min Year = ", gdp\_stage['Year'].min(), "max: ", gdp\_stage['Year'].max())  
print("CO2 Min Year = ", stage['Year'].min(), "max: ", stage['Year'].max())  
gdp\_stage\_trunc = gdp\_stage[gdp\_stage['Year'] < 2012]  
print(len(gdp\_stage\_trunc))  
print(len(stage))  
import matplotlib.pyplot as plt  
fig, axis = plt.subplots()  
axis.yaxis.grid(True)  
axis.set\_title('CO2 Emissions vs. GDP (per capita)',fontsize=10)  
axis.set\_xlabel(gdp\_stage\_trunc['IndicatorName'].iloc[0],fontsize=10)  
axis.set\_ylabel(stage['IndicatorName'].iloc[0],fontsize=10)  
X = gdp\_stage\_trunc['Value']  
Y = stage['Value']  
axis.scatter(X, Y)  
plt.show()