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",category=matplotlib.cbook.mplDeprecation) This is sir's statement, error coming so modified  
warnings.filterwarnings("ignore",category=matplotlib.MatplotlibDeprecationWarning)  
  
#Let us read the dataset  
data = pd.read\_csv('/Users/Ramesh/Desktop/Py/Indicators.bz2', compression='bz2')  
  
print("data.shape: ", data.shape)  
#This is a really large dataset, at least in terms of the number of rows.  
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))  
  
  
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))  
  
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]  
# stage dataset contain indicators matching the USA for country code & CO2  
#emissions over time.  
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  
# Plot the Histogram  
plt.bar(years,co2)  
plt.show()  
  
  
#Let us create a line plot.  
plt.plot(stage['Year'].values, stage['Value'].values)  
# Label the axes  
plt.xlabel('Year')  
plt.ylabel(stage['IndicatorName'].iloc[0])  
# Label the figure  
plt.title('CO2 Emissions in USA')  
# Start the y axis at 0 and x axis from 1959  
plt.axis([1959, 2011,0,25])  
plt.show()  
  
  
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.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])  
# apply our mask  
co2\_2011 = data[mask1 & mask2]  
co2\_2011.head()  
print(len(co2\_2011))  
  
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()  
  
  
hist\_indicator = 'GDP per capita \(constant 2005'  
hist\_country = 'USA'  
mask1 = data['IndicatorName'].str.contains(hist\_indicator)  
mask2 = data['CountryCode'].str.contains(hist\_country)  
# Stage is just those indicators matching the USA for country code and CO2  
#emissions over time.  
gdp\_stage = data[mask1 & mask2]  
# Plot gdp\_stage vs stage  
print("GDP: ",gdp\_stage.head())  
stage.head()  
# Switch to a line plot  
plt.plot(gdp\_stage['Year'].values, gdp\_stage['Value'].values)  
# Label the axes  
plt.xlabel('Year')  
plt.ylabel(gdp\_stage['IndicatorName'].iloc[0])  
#Label the figure  
plt.title('GDP Per Capita USA')  
plt.show()  
  
print("GDP Min Year = ", gdp\_stage['Year'].min(), "max: ",  
gdp\_stage['Year'].max())  
print("CO2 Min Year = ", stage['Year'].min(), "max: ", stage['Year'].max())  
  
  
  
#We have 3 extra years of GDP data, so let's trim those off so the scatterplot  
# has equal length arrays to compare (this is actually required by scatterplot)  
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()  
# Grid lines, Xticks, Xlabel, Ylabel  
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()  
  
#This does not look like a strong relationship. We can test this by looking at correlation.  
print(np.corrcoef(gdp\_stage\_trunc['Value'],stage['Value']))