#importing neccessary python packages

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

#function for line plot

def plot\_population\_chart(data\_path):

# Reading and processing the data

data = pd.read\_csv(data\_path, skiprows=3)

data\_new = data.loc[:, ~data.columns.isin(["Country Code", "Indicator Name", "Indicator Code"])]

data\_tran = data\_new.transpose()

data\_tran.columns = data\_tran.iloc[0]

data\_tran = data\_tran[1:64]

# Defining the data for x and y axis

x = data\_tran.index.astype(int)

y1 = data\_tran["Afghanistan"].astype(float)

y2 = data\_tran["Japan"].astype(float)

y3 = data\_tran["Brazil"].astype(float)

y4 = data\_tran["United Kingdom"].astype(float)

# Selecting the x value to plot the x axis

df\_tran = data\_tran[data\_tran.index.str.isnumeric()]

x = df\_tran.index.astype(int)

plt.xticks(x[::10], x[::10])

# Selecting the y value to plot the y axis

y\_tick\_values = np.linspace(

min(min(y1), min(y2), min(y3), min(y4)),

max(max(y1), max(y2), max(y3), max(y4)),

5

)

y\_tick\_labels = [f'{int(value):,}' for value in y\_tick\_values]

plt.yticks(y\_tick\_values, y\_tick\_labels)

# Plotting the data

plt.plot(x, y1, label="Afghanistan")

plt.plot(x, y2, label="Japan")

plt.plot(x, y3, label="Brazil")

plt.plot(x, y4, label="United Kingdom")

# Customizing the labels

plt.xlabel("Years")

plt.ylabel("Population")

plt.title("Population Chart")

plt.legend()

plt.show()

#function for bar chart

def plot\_population\_growth\_chart(data\_path, countries, years):

# Reading data

data\_bar = pd.read\_csv(data\_path, skiprows=3)

data\_bar = data\_bar.loc[:, ~data\_bar.columns.isin(["Country Code", "Indicator Name", "Indicator Code"])]

# Selection of country data

country\_data = [data\_bar[data\_bar["Country Name"] == country] for country in countries]

# Selection of data for specific years

populations = [country[years].values.flatten() for country in country\_data]

# Creating the bar positions for x-axis

bar\_positions = np.arange(len(years))

# Plotting the bar chart

bar\_width = 0.25

for i, population in enumerate(populations):

plt.bar(bar\_positions + i \* bar\_width, population, width=bar\_width, label=countries[i])

# Customizing the labels and the title

plt.xlabel('Year')

plt.ylabel('Population')

plt.title('Population Growth through the Years')

plt.xticks(bar\_positions + (len(countries) - 1) \* bar\_width / 2, years)

plt.legend()

# Show the plot

plt.show()

#function for pie chart

def plot\_population\_distribution(data\_path, countries, year):

# Read data

data\_pi = pd.read\_csv(data\_path, skiprows=3)

data\_pi = data\_pi.loc[:, ~data\_pi.columns.isin(["Country Code", "Indicator Name", "Indicator Code"])]

# Select data for specified countries and year

country\_data = [data\_pi[data\_pi["Country Name"] == country] for country in countries]

# Get population data for the specified year

populations = [country[year].values[0] for country in country\_data]

# Create a pie chart

plt.pie(populations, labels=countries, autopct='%1.1f%%', startangle=90, colors=plt.cm.Paired.colors)

# Customize the plot title

plt.title(f'Population Distribution in {year}')

# Show the plot

plt.show()

#plotting the line, bar and pie charts

plot\_population\_chart("C:/Users/abigi/OneDrive/Desktop/Assignments/Applied data science 1/API\_SP.POP.TOTL\_DS2\_en\_csv\_v2\_6011311.csv")

countries\_to\_plot = ["Afghanistan", "United Arab Emirates", "Cuba"]

years\_to\_plot = ["1960", "2000", "2004", "2008", "2012"]

data\_path = "C:/Users/abigi/OneDrive/Desktop/Assignments/Applied data science 1/API\_SP.POP.TOTL\_DS2\_en\_csv\_v2\_6011311.csv"

plot\_population\_growth\_chart(data\_path, countries\_to\_plot, years\_to\_plot)

countries\_to\_plot = ["Afghanistan", "United Arab Emirates", "Cuba", "Switzerland", "Greece", "Norway"]

data\_path = "C:/Users/abigi/OneDrive/Desktop/Assignments/Applied data science 1/API\_SP.POP.TOTL\_DS2\_en\_csv\_v2\_6011311.csv"

year\_to\_plot = "2012"

plot\_population\_distribution(data\_path, countries\_to\_plot, year\_to\_plot)