**Introduction to Data Science**

**Name: Muhammad Anas Aamir**

**Reg#: FA21-BSE-012**

**Question #1**

**Answer:**

import pandas as pd

import matplotlib.pyplot as plt

# Load the dataset 'world\_pop.csv' with columns 'Country Name', 'Country Code', and population data from 1960 to 2020

data = pd.read\_csv('world\_pop.csv')

# Filter data for the year 2020 and sort by population

data\_2020 = data[['Country Name', '2020']].sort\_values(by='2020', ascending=False)

# Select the top 10 most populated countries in 2020

top\_10\_countries\_2020 = data\_2020.head(10)

# Extract data for the top 10 countries from the original dataset

top\_10\_data = data[data['Country Name'].isin(top\_10\_countries\_2020['Country Name'])]

# Set the figure size

plt.figure(figsize=(12, 8))

# Plot population data for the top 10 countries

for country in top\_10\_data['Country Name'].unique():

country\_data = top\_10\_data[top\_10\_data['Country Name'] == country]

plt.plot(country\_data.columns[1:], country\_data.values[0][1:], label=country)

# Add titles and labels

plt.title('Population Trends of Top 10 Most Populated Countries (1960-2020)', fontsize=16)

plt.xlabel('Year', fontsize=14)

plt.ylabel('Population', fontsize=14)

# Add legend

plt.legend(loc='upper left', fontsize=10, bbox\_to\_anchor=(1, 1))

# Set grid

plt.grid(True)

# Show the plot

plt.tight\_layout()

plt.show()

**Question #2**

**Answer:**

import pandas as pd

import matplotlib.pyplot as plt

# Load the dataset from 'world\_pop.csv'

data = pd.read\_csv('world\_pop.csv')

# Task 1: Display the population of the 10 least populous countries in 2015 using a horizontal bar chart

least\_populous\_2015 = data[['Country Name', '2015']].sort\_values(by='2015').head(10)

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

plt.barh(least\_populous\_2015['Country Name'], least\_populous\_2015['2015'], color='skyblue')

plt.xlabel('Population')

plt.ylabel('Country')

plt.title('Population of 10 Least Populous Countries in 2015')

plt.gca().invert\_yaxis() # Invert y-axis to show the country with the lowest population at the top

plt.show()

# Task 2: Calculate the population change for Pakistan, India, United States, and United Kingdom from 1970 to 2010

countries\_of\_interest = ['Pakistan', 'India', 'United States', 'United Kingdom']

pop\_change\_1970\_2010 = data.loc[data['Country Name'].isin(countries\_of\_interest), ['Country Name', '1970', '2010']]

pop\_change\_1970\_2010['Population Change'] = pop\_change\_1970\_2010['2010'] - pop\_change\_1970\_2010['1970']

pop\_change\_1970\_2010\_millions = pop\_change\_1970\_2010.copy()

pop\_change\_1970\_2010\_millions['Population Change'] /= 1e6 # Convert population change to millions

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

plt.bar(pop\_change\_1970\_2010\_millions['Country Name'], pop\_change\_1970\_2010\_millions['Population Change'], color='lightgreen')

plt.xlabel('Country')

plt.ylabel('Population Change (Millions)')

plt.title('Population Change from 1970 to 2010')

plt.show()

# Task 3: Calculate Pakistan's population growth between 2010 and 2020

pakistan\_population\_growth = data.loc[data['Country Name'] == 'Pakistan', ['Country Name', '2010', '2020']]

pakistan\_population\_growth['Population Growth'] = pakistan\_population\_growth['2020'] - pakistan\_population\_growth['2010']

plt.figure(figsize=(8, 6))

plt.plot(pakistan\_population\_growth.columns[1:], pakistan\_population\_growth.values[0][1:], marker='o', color='orange')

plt.xlabel('Year')

plt.ylabel('Population')

plt.title('Pakistan Population Growth (2010-2020)')

plt.grid(True)

plt.show()

**Question #3**

**Answer:**

[11:07 pm, 28/03/2024] Anas Aamir: import pandas as pd

import matplotlib.pyplot as plt

# Load the diamonds dataset (assuming you have it as 'diamonds.csv')

diamonds = pd.read\_csv('diamonds.csv')

# Filter diamonds with clarity 'SI2' and color 'E'

filtered\_diamonds = diamonds[(diamonds['clarity'] == 'SI2') & (diamonds['color'] == 'E')]

# Set figure size

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

# Create a scatter plot

plt.scatter(filtered\_diamonds['carat'], filtered\_diamonds['price'], c=filtered\_diamonds['cut'], cmap='viridis', alpha=0.7)

# Add labels and title

plt.xlabel('Carat')

plt.ylabel('Price')

plt.title('Relationship between Carat and Price for SI2 Clarity, Color E Diamonds')

plt.grid(True)

# Add legend for cut values

cut\_legend = plt.legend(handles=[plt.Line2D([0], [0], marker='o', color='w', m…

[11:10 pm, 28/03/2024] Anas Aamir: import folium

# Create a map centered at an appropriate location (e.g., Washington, D.C.)

us\_map = folium.Map(location=[38.8951100, -77.0363700], zoom\_start=4)

# Example nuclear waste storage sites (you can replace these with actual data)

nuclear\_sites = [

{'name': 'Site A', 'location': [40.712776, -74.005974]},

{'name': 'Site B', 'location': [34.052235, -118.243683]},

{'name': 'Site C', 'location': [41.878113, -87.629799]}

]

# Add markers for each nuclear waste storage site

for site in nuclear\_sites:

folium.Marker(location=site['location'], popup=site['name']).add\_to(us\_map)

# Save the map to an HTML file

us\_map.save('nuclear\_waste\_sites\_map.html')

**Question #4**

**Answer:**

import folium

# Load the dataset from 'nuclear\_waste\_sites.csv'

data = pd.read\_csv(' nuclear\_waste\_sites.csv')

# Create a map centered at an appropriate location (e.g., Washington, D.C.)

us\_map = folium.Map(location=[38.8951100, -77.0363700], zoom\_start=4)

# Example nuclear waste storage sites (you can replace these with actual data)

nuclear\_waste\_sites = [

{'name': 'Site A', 'location': [40.712776, -74.005974]},

{'name': 'Site B', 'location': [34.052235, -118.243683]},

{'name': 'Site C', 'location': [41.878113, -87.629799]}

]

# Add markers for each nuclear waste storage site

for site in nuclear\_waste\_sites:

folium.Marker(location=site['location'], popup=site['name']).add\_to(us\_map)

# Save the map to an HTML file

us\_map.save('nuclear\_waste\_sites.html')

**Question #5**

**Answer:**

import folium

# Load the dataset from 'pak-heritage-sites.csv'

data = pd.read\_csv(' pak-heritage- sites.csv')

# Create a map centered at an appropriate location (e.g., Islamabad)

pakistan\_map = folium.Map(location=[33.6844, 73.0479], zoom\_start=6)

# Example heritage sites (you can replace these with actual data)

pak-heritage-sites = [

{'name': 'Moenjodaro', 'location': [27.3292, 68.1347]},

{'name': 'Taxila', 'location': [33.7294, 72.8158]},

{'name': 'Lahore Fort', 'location': [31.5497, 74.3436]}

]

# Add markers for each heritage site

for site in pak-heritage-sites:

folium.Marker(location=site['location'], popup=site['name']).add\_to(pakistan\_map)

# Save the map to an HTML file

pakistan\_map.save(' pak-heritage-sites-map.html')