Prodigy InfoTech Internship

Task 1:

Create a bar chart or histogram to visualize the distribution of a categorical or continuous variable, such as the distribution of ages or genders in a population.

Sample Dataset: World Bank Population Dataset

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Understanding the shape of the Dataset:

```
[2]
total=pd.read_csv("/content/drive/MyDrive/Project_Datasets/World_Population_Dataset/Total.csv")
metadata=pd.read_csv("/content/drive/MyDrive/Project_Datasets/World_Population_Dataset/Metadata.csv")
```

```
total.head()
```

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	...	2014	2015	2016	2017	
0	Aruba	ABW	Population, total	SP.POP.TOTL	54608.0	55811.0	56682.0	57475.0	58178.0	58782.0	...	103594.0	104257.0	104874.0	105439.0	
1	Africa Eastern and Southern	AFE	Population, total	SP.POP.TOTL	130692579.0	134169237.0	137835590.0	141630546.0	145605995.0	149742351.0	...	583651101.0	600008424.0	616377605.0	632746570.0	64
2	Afghanistan	AFG	Population, total	SP.POP.TOTL	8622466.0	8790140.0	8969047.0	9157465.0	9355514.0	9565147.0	...	32716210.0	33753499.0	34636207.0	35643418.0	3
3	Africa Western and Central	AFW	Population, total	SP.POP.TOTL	97256290.0	99314028.0	101445032.0	103667517.0	105959979.0	108336203.0	...	397855507.0	408690375.0	419778384.0	431138704.0	44
4	Angola	AGO	Population, total	SP.POP.TOTL	5357195.0	5441333.0	5521400.0	5599827.0	5673199.0	5736582.0	...	27128337.0	28127721.0	29154746.0	30208628.0	3

5 rows × 68 columns

```
total.info()
```

By: Aman Kumar Jha
LinkedIn

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 266 entries, 0 to 265
Data columns (total 68 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Country Name    266 non-null    object 
 1   Country Code    266 non-null    object 
 2   Indicator Name  266 non-null    object 
 3   Indicator Code  266 non-null    object 
 4   1960            264 non-null    float64
 5   1961            264 non-null    float64
 6   1962            264 non-null    float64
 7   1963            264 non-null    float64
 8   1964            264 non-null    float64
 9   1965            264 non-null    float64
 10  1966            264 non-null    float64
 11  1967            264 non-null    float64
 12  1968            264 non-null    float64
 13  1969            264 non-null    float64
 14  1970            264 non-null    float64
 15  1971            264 non-null    float64
 16  1972            264 non-null    float64
 17  1973            264 non-null    float64
 18  1974            264 non-null    float64
 19  1975            264 non-null    float64
 20  1976            264 non-null    float64
 21  1977            264 non-null    float64
 22  1978            264 non-null    float64
 23  1979            264 non-null    float64
 24  1980            264 non-null    float64
 25  1981            264 non-null    float64
 26  1982            264 non-null    float64
 27  1983            264 non-null    float64
 28  1984            264 non-null    float64
```

```
 29  1985            264 non-null    float64
 30  1986            264 non-null    float64
 31  1987            264 non-null    float64
 32  1988            264 non-null    float64
 33  1989            264 non-null    float64
 34  1990            265 non-null    float64
 35  1991            265 non-null    float64
 36  1992            265 non-null    float64
 37  1993            265 non-null    float64
 38  1994            265 non-null    float64
 39  1995            265 non-null    float64
 40  1996            265 non-null    float64
 41  1997            265 non-null    float64
 42  1998            265 non-null    float64
 43  1999            265 non-null    float64
 44  2000            265 non-null    float64
 45  2001            265 non-null    float64
 46  2002            265 non-null    float64
 47  2003            265 non-null    float64
 48  2004            265 non-null    float64
 49  2005            265 non-null    float64
 50  2006            265 non-null    float64
 51  2007            265 non-null    float64
 52  2008            265 non-null    float64
 53  2009            265 non-null    float64
 54  2010            265 non-null    float64
 55  2011            265 non-null    float64
 56  2012            265 non-null    float64
 57  2013            265 non-null    float64
 58  2014            265 non-null    float64
 59  2015            265 non-null    float64
 60  2016            265 non-null    float64
 61  2017            265 non-null    float64
 62  2018            265 non-null    float64
 63  2019            265 non-null    float64
```

```
[4]  63  2019            265 non-null    float64
     64  2020            265 non-null    float64
     65  2021            265 non-null    float64
     66  2022            265 non-null    float64
     67  2023            0 non-null      float64
    dtypes: float64(64), object(4)
    memory usage: 141.4+ KB
```

```
[5] 1 total.describe()
```

	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	...	2014	2015	2016	
count	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	...	2.650000e+02	2.650000e+02	2.650000e+02	2
mean	1.172860e+08	1.188956e+08	1.210661e+08	1.237484e+08	1.264530e+08	1.291965e+08	1.320558e+08	1.349134e+08	1.378513e+08	1.408944e+08	...	2.966485e+08	3.004946e+08	3.043392e+08	3
std	3.695500e+08	3.740958e+08	3.808121e+08	3.895098e+08	3.982497e+08	4.071209e+08	4.164559e+08	4.257477e+08	4.353270e+08	4.452978e+08	...	9.300107e+08	9.412509e+08	9.524225e+08	9
min	2.646000e+03	2.888000e+03	3.171000e+03	3.481000e+03	3.811000e+03	4.161000e+03	4.531000e+03	4.930000e+03	5.354000e+03	5.646000e+03	...	1.089900e+04	1.087700e+04	1.085200e+04	1
25%	5.132212e+05	5.231345e+05	5.337595e+05	5.449288e+05	5.566630e+05	5.651150e+05	5.691470e+05	5.773872e+05	5.832700e+05	5.875942e+05	...	1.743309e+06	1.788196e+06	1.777557e+06	1
50%	3.757486e+06	3.887144e+06	4.023896e+06	4.139356e+06	4.224612e+06	4.277636e+06	4.331825e+06	4.385700e+06	4.450934e+06	4.530800e+06	...	1.028212e+07	1.035808e+07	1.032545e+07	1
75%	2.670606e+07	2.748694e+07	2.830289e+07	2.914708e+07	3.001684e+07	3.084892e+07	3.163010e+07	3.209247e+07	3.249927e+07	3.277149e+07	...	6.078914e+07	6.073058e+07	6.062750e+07	6
max	3.031474e+09	3.072422e+09	3.126850e+09	3.193429e+09	3.260442e+09	3.328209e+09	3.398480e+09	3.468371e+09	3.540164e+09	3.614573e+09	...	7.317040e+09	7.403850e+09	7.490415e+09	7

8 rows × 64 columns

```
[6] 1 metadata.head()
```

	Country Code	Region	IncomeGroup	SpecialNotes	TableName
0	ABW	Latin America & Caribbean	High income	NaN	Aruba
1	AFE	NaN	NaN	26 countries, stretching from the Red Sea in t...	Africa Eastern and Southern
2	AFG	South Asia	Low income	The reporting period for national accounts dat...	Afghanistan
3	AFW	NaN	NaN	22 countries, stretching from the westernmost ...	Africa Western and Central
4	AGO	Sub-Saharan Africa	Lower middle income	The World Bank systematically assesses the app...	Angola

Next steps: Generate code with metadata | View recommended plots

```
[7] 1 metadata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 265 entries, 0 to 264
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Country Code  265 non-null    object
 1   Region        217 non-null    object
 2   IncomeGroup   216 non-null    object
 3   SpecialNotes  126 non-null    object
 4   TableName     265 non-null    object
dtypes: object(5)
memory usage: 10.5+ KB
```

```
1 metadata.describe()
```

	Country Code	Region	IncomeGroup	SpecialNotes	TableName
count	265	217	216	126	265
unique	265	7	4	111	265
top	ABW	Europe & Central Asia	High income	The reporting period for national accounts dat...	Aruba
freq	1	58	82	7	1

Data Cleaning:

```
[10] 1 total = total.drop(columns=['Indicator Name', 'Indicator Code', '2023']).dropna()
     2 metadata = metadata.drop(columns=['SpecialNotes']).dropna()
     3 data = (total.merge(metadata, on='Country Code')
     4 .rename(columns={'Country Name':'Country','IncomeGroup':'Income'}))
```

By: Aman Kumar Jha
LinkedIn

```
1 data.head()
```

	Country	Country Code	1960	1961	1962	1963	1964	1965	1966	1967	...	2016	2017	2018	2019	2020	2021	20
0	Aruba	ABW	54608.0	55811.0	56682.0	57475.0	58178.0	58782.0	59291.0	59522.0	...	104874.0	105439.0	105962.0	106442.0	106585.0	106537.0	10644
1	Afghanistan	AFG	8622466.0	8790140.0	8969047.0	9157465.0	9355514.0	9565147.0	9783147.0	10010030.0	...	34636207.0	35643418.0	36686784.0	37769499.0	38972230.0	40099462.0	4112877
2	Angola	AGO	5357195.0	5441333.0	5521400.0	5599827.0	5673199.0	5736582.0	5787044.0	5827503.0	...	29154746.0	30208628.0	31273533.0	32353588.0	33428486.0	34503774.0	3558898
3	Albania	ALB	1608800.0	1659800.0	1711319.0	1762621.0	1814135.0	1864791.0	1914573.0	1965598.0	...	2876101.0	2873457.0	2866376.0	2854191.0	2837849.0	2811666.0	277768
4	Andorra	AND	9443.0	10216.0	11014.0	11839.0	12690.0	13563.0	14546.0	15745.0	...	72540.0	73837.0	75013.0	76343.0	77700.0	79034.0	7982

5 rows × 68 columns

```
[12] 1 data=data.melt(id_vars=['Country','Region','Income'],
     2               value_vars=[str(year) for year in range(1960,2023)],
     3               var_name='Year',
     4               value_name='Population')
```

```
[13] 1 data.to_csv("/content/drive/MyDrive/Project_Datasets/World_Population_Dataset/total_cleaned.csv")
```

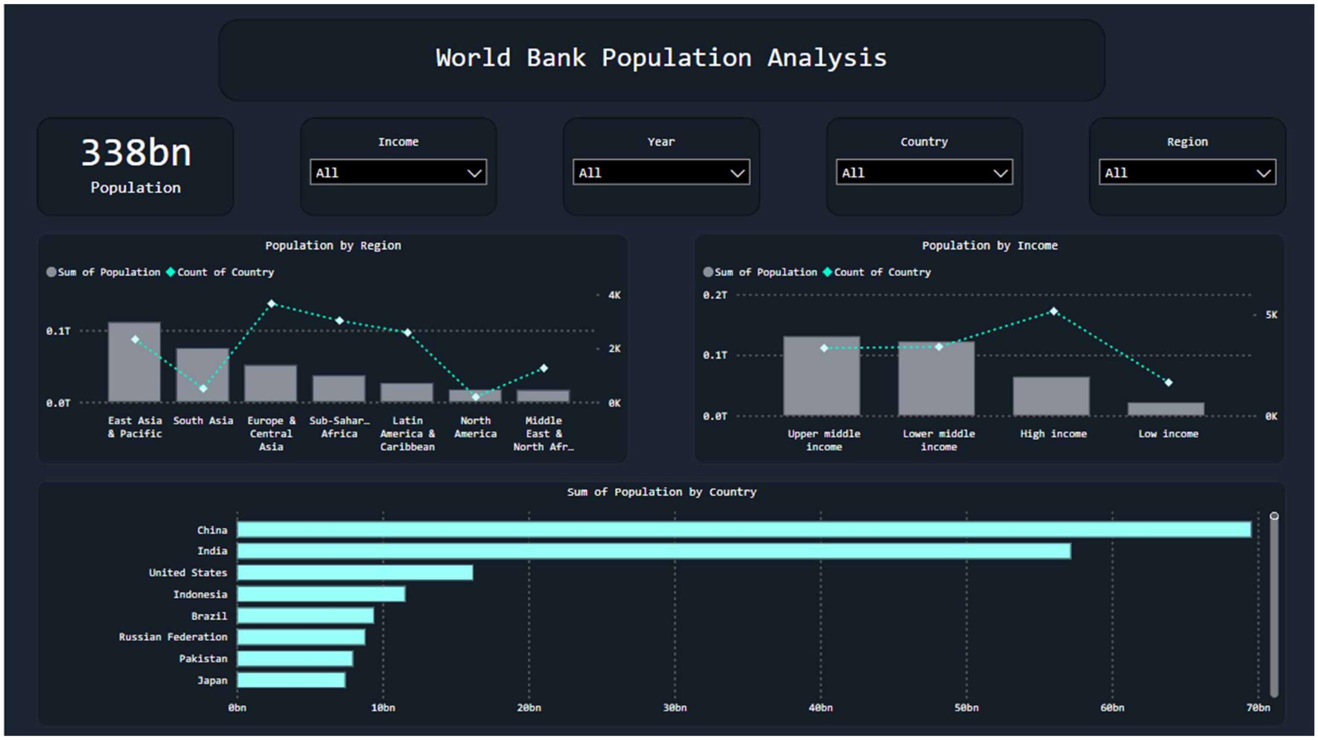
```
1 data
```

	Country	Region	Income	Year	Population
0	Aruba	Latin America & Caribbean	High income	1960	54608.0
1	Afghanistan	South Asia	Low income	1960	8622466.0
2	Angola	Sub-Saharan Africa	Lower middle income	1960	5357195.0
3	Albania	Europe & Central Asia	Upper middle income	1960	1608800.0
4	Andorra	Europe & Central Asia	High income	1960	9443.0
...	...	...	...	...	...
13540	Kosovo	Europe & Central Asia	Upper middle income	2022	1761985.0
13541	Yemen, Rep.	Middle East & North Africa	Low income	2022	33696614.0
13542	South Africa	Sub-Saharan Africa	Upper middle income	2022	59893885.0
13543	Zambia	Sub-Saharan Africa	Lower middle income	2022	20017675.0
13544	Zimbabwe	Sub-Saharan Africa	Lower middle income	2022	16320537.0

13545 rows × 5 columns

By: Aman Kumar Jha
LinkedIn

Data Visualization (Using Power BI):



By: Aman Kumar Jha
[LinkedIn]