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 numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from google.colab import drive
drive.mount('/content/drive')
from google.colab import files
uploaded = files.upload()
```

Understanding the shape of the Dataset:

```
population_df=pd.read_csv('/content/drive/My Drive/Total_Population.csv')
metadata_df = pd.read_csv('/content/drive/My Drive/Metadata_Country.csv')
```

```
[76] population_df.head()
```

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	...
0	Aruba	ABW	Population, total	SP.POP.TOTL	54608.0	55811.0	56682.0	57475.0	58178.0	58782.0	...
1	Africa Eastern and Southern	AFE	Population, total	SP.POP.TOTL	130692579.0	134169237.0	137835590.0	141630546.0	145605995.0	149742351.0	...
2	Afghanistan	AFG	Population, total	SP.POP.TOTL	8622466.0	8790140.0	8969047.0	9157465.0	9355514.0	9565147.0	...
3	Africa Western and Central	AFW	Population, total	SP.POP.TOTL	97256290.0	99314028.0	101445032.0	103667517.0	105959979.0	108336203.0	...
4	Angola	AGO	Population, total	SP.POP.TOTL	5357195.0	5441333.0	5521400.0	5599827.0	5673199.0	5736582.0	...

5 rows × 68 columns

...	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023
...	103594.0	104257.0	104874.0	105439.0	105962.0	106442.0	106585.0	106537.0	106445.0	NaN
...	583651101.0	600008424.0	616377605.0	632746570.0	649757148.0	667242986.0	685112979.0	702977106.0	720859132.0	NaN
...	32716210.0	33753499.0	34636207.0	35643418.0	36686784.0	37769499.0	38972230.0	40099462.0	41128771.0	NaN
...	397855507.0	408690375.0	419778384.0	431138704.0	442646825.0	454306063.0	466189102.0	478185907.0	490330870.0	NaN
...	27128337.0	28127721.0	29154746.0	30208628.0	31273533.0	32353588.0	33428486.0	34503774.0	35588987.0	NaN

By :- Amolak Singh
singhamolak974@gmail.com

```
population_df.info()
```

```
<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
 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
```

By :- Amolak Singh
singhamolak974@gmail.com

```
metadata_df.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

```
[58] metadata_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 216 entries, 0 to 264
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Country Code  216 non-null    object
 1   Region        216 non-null    object
 2   IncomeGroup   216 non-null    object
 3   TableName     216 non-null    object
dtypes: object(4)
memory usage: 8.4+ KB
```

Data Cleaning:

```
population_df = population_df.drop(columns=['Indicator Name', 'Indicator Code','2023']).dropna()
metadata_df = metadata_df.drop(columns=['SpecialNotes', ]).dropna()
df = (population_df.merge(metadata_df, on='Country Code')
.rename(columns={'Country Name': 'Country', 'IncomeGroup':'Income'}))
```

```
df
```

	Country	Country Code	1960	1961	1962	1963	1964	1965	1966	1967	...	2016	2017	2018	2019	2020	2021	2022	Region	Income	TableName
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	106445.0	Latin America & Caribbean	High income	Aruba
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	41128771.0	South Asia	Low income	Afghanistan
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	35588987.0	Sub-Saharan Africa	Lower middle income	Angola
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	2777689.0	Europe & Central Asia	Upper middle income	Albania
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	79824.0	Europe & Central Asia	High income	Andorra
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
210	Kosovo	XKX	947000.0	966000.0	994000.0	1022000.0	1050000.0	1078000.0	1106000.0	1135000.0	...	1777557.0	1791003.0	1797085.0	1788878.0	1790133.0	1786038.0	1761985.0	Europe & Central Asia	Upper middle income	Kosovo
211	Yemen, Rep.	YEM	5542459.0	5646668.0	5753386.0	5860197.0	5973803.0	6097298.0	6228430.0	6368014.0	...	29274002.0	30034389.0	30790513.0	31546691.0	32284046.0	32981641.0	33696614.0	Middle East & North Africa	Low income	Yemen, Rep.
212	South Africa	ZAF	16520441.0	16989464.0	17503133.0	18042215.0	18603097.0	19187194.0	19789771.0	20410677.0	...	56422274.0	56641209.0	57339635.0	58087055.0	58801927.0	59392255.0	59893885.0	Sub-Saharan Africa	Upper middle income	South Africa
213	Zambia	ZMB	3119430.0	3219451.0	3323427.0	3431381.0	3542764.0	3658024.0	3777680.0	3901288.0	...	16767761.0	17298054.0	17835893.0	18380477.0	18927715.0	19473125.0	20017675.0	Sub-Saharan Africa	Lower middle income	Zambia
214	Zimbabwe	ZWE	3806310.0	3925952.0	4049778.0	4177931.0	4310332.0	4447149.0	4588529.0	4734694.0	...	14452704.0	14751101.0	15052184.0	15354608.0	15669666.0	15993524.0	16320537.0	Sub-Saharan Africa	Lower middle income	Zimbabwe

215 rows × 68 columns

```
[9] df.to_csv ('/content/drive/My Drive/new_df.csv')
```

```
df.head()
```

	Country	Country Code	1960	1961	1962	1963	1964	1965	1966	1967	...	2016	2017	2018	2019	2020	2021	2022	Region	Income	TableName
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	106445.0	Latin America & Caribbean	High income	Aruba
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	41128771.0	South Asia	Low income	Afghanistan
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	35588987.0	Sub-Saharan Africa	Lower middle income	Angola
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	2777689.0	Europe & Central Asia	Upper middle income	Albania
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	79824.0	Europe & Central Asia	High income	Andorra

5 rows × 68 columns

By :- Amolak Singh
singhamolak974@gmail.com

```
[11] df = df.melt(id_vars=['Country', 'Region', 'Income'],
     value_vars=[str(year) for year in range(1960, 2023)],
     var_name= 'Year',
     value_name='Population')
```

```
df
```

	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

Next steps: Generate code with df View recommended plots

```
[13] df.to_csv ('/content/drive/My Drive/new01_df.csv')
```

```
df.head()
```

	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

Next steps: Generate code with df View recommended plots

By :- Amolak Singh
singhamolak974@gmail.com

```
[15] df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13545 entries, 0 to 13544
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Country     13545 non-null  object 
 1   Region      13545 non-null  object 
 2   Income      13545 non-null  object 
 3   Year        13545 non-null  object 
 4   Population  13545 non-null  float64
dtypes: float64(1), object(4)
memory usage: 529.2+ KB
```

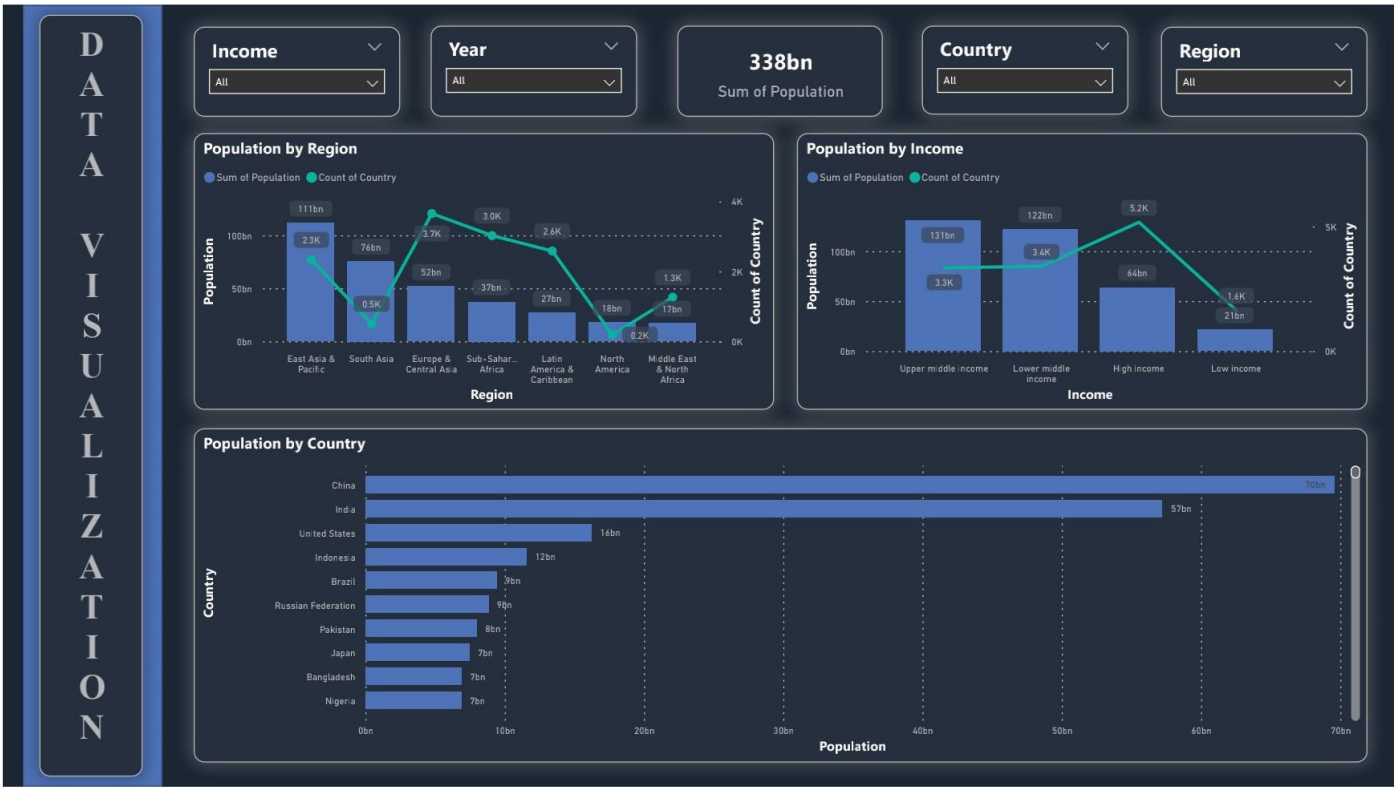
```
[16] df.duplicated().sum()

0
```

```
[17] df.isna().sum()

Country       0
Region        0
Income        0
Year          0
Population    0
dtype: int64
```

Data Visualization (Using Power BI) :



By :- Amolak Singh
singhamolak974@gmail.com