```
#importing libaries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

data = pd.read_csv("/content/task2.csv")
data.head()
```

	Country/Other	Population (2020)	Yearly Change	Net Change	Density (P/Km²)	Land Area (Km²)	Migrants (net)	Fert. Rate	Mec Ag
0	Afghanistan	38928346	2.33	886592	60	652860	-62920	4.6	1
1	Albania	2877797	-0.11	-3120	105	27400	-14000	1.6	3
2	Algeria	43851044	1.85	797990	18	2381740	-10000	3.1	2
3	Angola	32866272	3.27	1040977	26	1246700	6413	5.6	1
	Antigua and								

Next steps: Generate code with data View recommended plots

```
# Now we got null values in Urban Pop column so we have to fill it or drop the rows .
# But Since it's a small data set we can't drop the rows , beacuse it will may affect the data set .
# So for this we can fill it with mean , median or mode .
# but before that we have to Analysis the distribution of data in Urban Pop and according to this we can fill the null val
sns.distplot(data["Urban Pop %"],kde=True,bins=20)
plt.xlabel("Urban Pop %")
plt.ylabel("Frequency")
plt.title("Distribution of Urban Pop %")
plt.show()
```

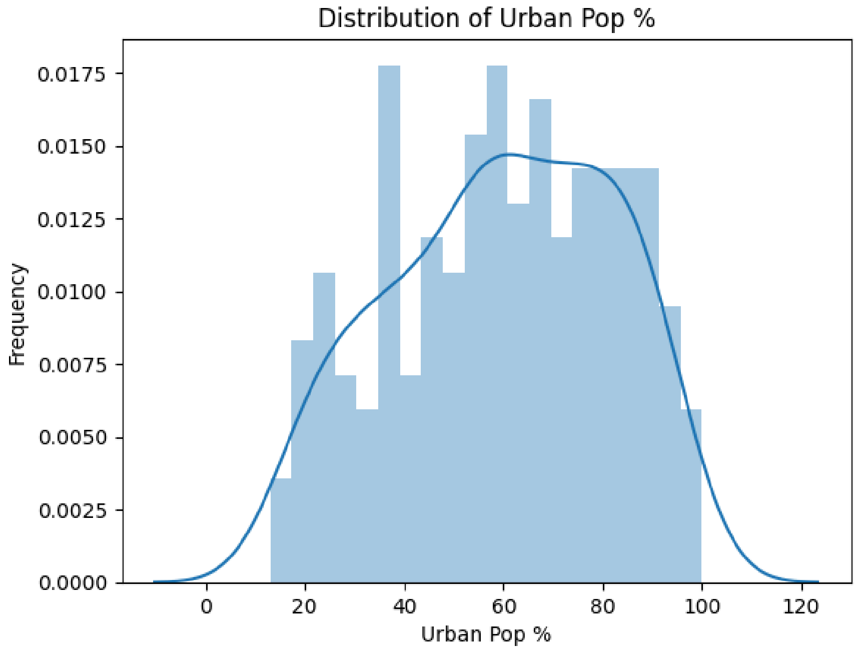
```
<ipython-input-36-9e001037b88d>:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data["Urban Pop %"],kde=True,bins=20)
```

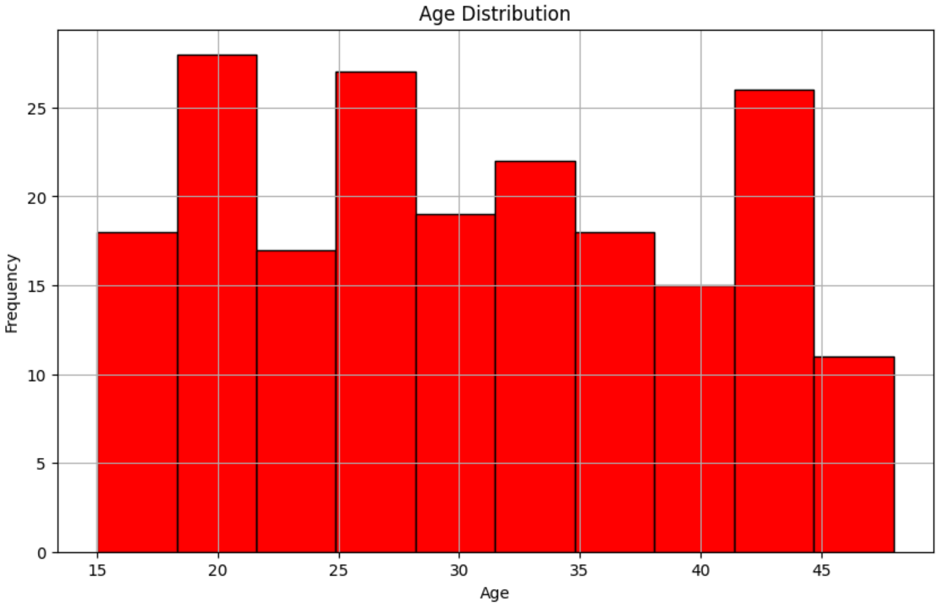


```
#Since we got a centralised graph we can use mean value to fill the missing values
data["Urban Pop %"].fillna(data["Urban Pop %"].mean(),inplace=True)
```

```
# checking the null values
data.isnull().sum()
```

```
Country/Other        0
Population (2020)    0
Yearly Change        0
Net Change           0
Density (P/Km²)      0
Land Area (Km²)      0
Migrants (net)       0
Fert. Rate           0
Med. Age             0
Urban Pop %          0
World Share          0
dtype: int64
```

```
#Bar chart to visualize the distribution of categorical variable such as age
plt.figure(figsize=(10,6))
plt.hist(data["Med. Age"],bins=10,color="red",edgecolor="black")
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.title("Age Distribution")
plt.grid(True)
plt.show()
```



Start coding or generate with AI.