In [1]:
```
# Importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# Read the DataSet
df = pd.read_csv("C:\\Users\\HP\\OneDrive\\Desktop\\world_population.csv")
```

In [3]:
```
# Displaying the 1st 5 rows of the data frame
df.head()
```

Out[3]:

	Country/Other	Population (2020)	Yearly Change	Net Change	Density (P/Km²)	Land Area (Km²)	Migrants (net)	Fert. Rate	Med. Age	Urban Pop %	World Share
0	Afghanistan	38928346	2.33	886592	60	652860	-62920	4.6	18	25.0	0.50
1	Albania	2877797	-0.11	-3120	105	27400	-14000	1.6	36	63.0	0.04
2	Algeria	43851044	1.85	797990	18	2381740	-10000	3.1	29	73.0	0.56
3	Angola	32866272	3.27	1040977	26	1246700	6413	5.6	17	67.0	0.42
4	Antigua and Barbuda	97929	0.84	811	223	440	0	2.0	34	26.0	0.00

In [4]:
```
# Shape of the DataSet
df.shape
```

Out[4]:
```
(201, 11)
```

In [5]:
```
# Information about the DataSet
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 201 entries, 0 to 200
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Country/Other       201 non-null    object
 1   Population (2020)   201 non-null    int64
 2   Yearly Change       201 non-null    float64
 3   Net Change          201 non-null    int64
 4   Density (P/Km²)     201 non-null    int64
 5   Land Area (Km²)     201 non-null    int64
 6   Migrants (net)      201 non-null    int64
 7   Fert. Rate          201 non-null    float64
 8   Med. Age            201 non-null    int64
 9   Urban Pop %         194 non-null    float64
 10  World Share         201 non-null    float64
dtypes: float64(4), int64(6), object(1)
memory usage: 17.4+ KB
```

In [6]:
```
# Checking for Null Values
df.isnull().sum()
```

Out[6]:
```
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 %           7
World Share           0
dtype: int64
```

In [7]:
```
df.columns
```

Out[7]:
```
Index(['Country/Other', 'Population (2020)', 'Yearly Change', 'Net Change',
       'Density (P/Km²)', 'Land Area (Km²)', 'Migrants (net)', 'Fert. Rate',
       'Med. Age', 'Urban Pop %', 'World Share'],
      dtype='object')
```

In [8]:
```
# 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 values

sns.distplot(df["Urban Pop %"],kde=True,bins=20)
plt.xlabel("Urban Pop %")
plt.ylabel("Frequency")
plt.title("Distribution of Urban Pop %")
plt.show()
```

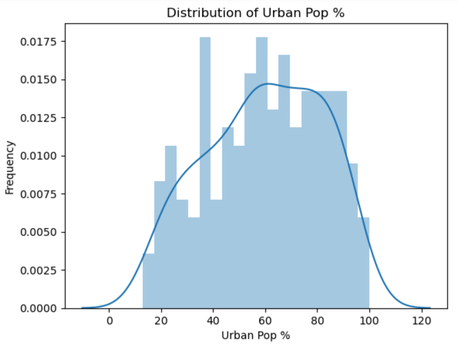
```
C:\Users\HP\AppData\Local\Temp\ipykernel_26964\3770800501.py:6: UserWarning:

`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(df["Urban Pop %"],kde=True,bins=20)
C:\Users\HP\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before o
perating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



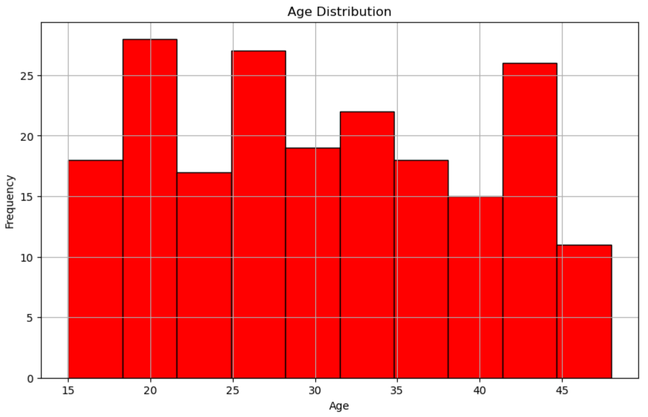
In [9]:
```
# Since we got a centralised graph we can use mean value to fill the missing values
df["Urban Pop %"].fillna(df["Urban Pop %"].mean(),inplace=True)
```

In [10]:
```
# checking the null values
df.isnull().sum()
```

Out[10]:
```
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
```

In [11]:
```
# Now we can see there are no missing values
```

In [12]:
```
# Bar chart to visualize the distribution of categorical variable such as age
plt.figure(figsize=(10,6))
plt.hist(df["Med. Age"],bins=10,color="red",edgecolor="black")
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.title("Age Distribution")
plt.grid(True)
plt.show()
```



In [13]:
```
# Now we can see which age groups are most and least prevalent
```