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

```
data = pd.read_csv('/content/population data.csv')
data.head()
```

	Country Name	Country Code	1960	1961	1962	1963	1964	1965	1966	1967	...	2013	
0	Aruba	ABW	54608	55811	56682	57475	58178	58782	59291	59522	...	102880	1
1	Africa Eastern and Southern	AFE	130692579	134169237	137835590	141630546	145605995	149742351	153955516	158313235	...	567892149	583
2	Afghanistan	AFG	8622466	8790140	8969047	9157465	9355514	9565147	9783147	10010030	...	31541209	327
3	Africa Western and Central	AFW	97256290	99314028	101445032	103667517	105959979	108336203	110798486	113319950	...	387204553	397
4	Angola	AGO	5357195	5441333	5521400	5599827	5673199	5736582	5787044	5827503	...	26147002	271

5 rows × 65 columns

```
data.shape
```

```
(264, 65)
```

```
data.dtypes
```

```
Country Name    object
Country Code    object
1960             int64
1961             int64
1962             int64
                 ...
2018             int64
2019             int64
2020             int64
2021             int64
2022             int64
Length: 65, dtype: object
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 264 entries, 0 to 263
Data columns (total 65 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Country Name  264 non-null    object
 1   Country Code  264 non-null    object
 2   1960          264 non-null    int64
 3   1961          264 non-null    int64
 4   1962          264 non-null    int64
 5   1963          264 non-null    int64
 6   1964          264 non-null    int64
 7   1965          264 non-null    int64
 8   1966          264 non-null    int64
 9   1967          264 non-null    int64
 10  1968          264 non-null    int64
 11  1969          264 non-null    int64
 12  1970          264 non-null    int64
 13  1971          264 non-null    int64
 14  1972          264 non-null    int64
 15  1973          264 non-null    int64
 16  1974          264 non-null    int64
 17  1975          264 non-null    int64
 18  1976          264 non-null    int64
 19  1977          264 non-null    int64
 20  1978          264 non-null    int64
 21  1979          264 non-null    int64
 22  1980          264 non-null    int64
 23  1981          264 non-null    int64
 24  1982          264 non-null    int64
 25  1983          264 non-null    int64
 26  1984          264 non-null    int64
 27  1985          264 non-null    int64
```

```
 28  1986          264 non-null    int64
 29  1987          264 non-null    int64
 30  1988          264 non-null    int64
 31  1989          264 non-null    int64
 32  1990          264 non-null    int64
 33  1991          264 non-null    int64
 34  1992          264 non-null    int64
 35  1993          264 non-null    int64
 36  1994          264 non-null    int64
 37  1995          264 non-null    int64
 38  1996          264 non-null    int64
 39  1997          264 non-null    int64
 40  1998          264 non-null    int64
 41  1999          264 non-null    int64
 42  2000          264 non-null    int64
 43  2001          264 non-null    int64
 44  2002          264 non-null    int64
 45  2003          264 non-null    int64
 46  2004          264 non-null    int64
 47  2005          264 non-null    int64
 48  2006          264 non-null    int64
 49  2007          264 non-null    int64
 50  2008          264 non-null    int64
 51  2009          264 non-null    int64
```

```
data.describe()
```

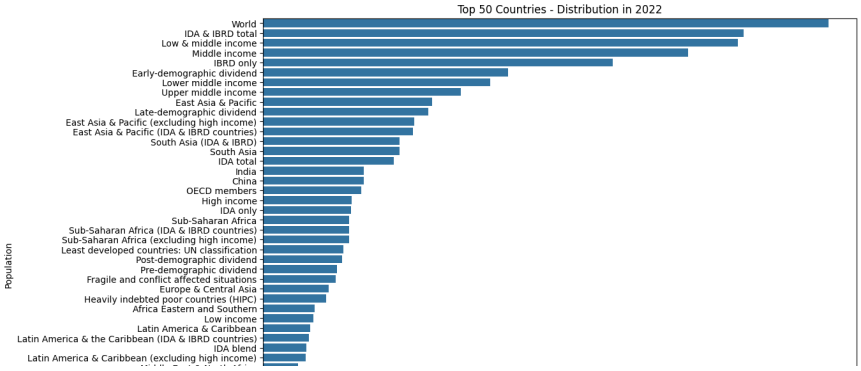
	1960	1961	1962	1963	1964	19
count	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e-
mean	1.172712e+08	1.188807e+08	1.210511e+08	1.237333e+08	1.264378e+08	1.291813e-
std	3.695439e+08	3.740897e+08	3.808061e+08	3.895039e+08	3.982439e+08	4.071153e-
min	2.646000e+03	2.888000e+03	3.171000e+03	3.481000e+03	3.811000e+03	4.161000e-
25%	5.132212e+05	5.231345e+05	5.337595e+05	5.449288e+05	5.566630e+05	5.651150e-
50%	3.757486e+06	3.887144e+06	4.023896e+06	4.139356e+06	4.224612e+06	4.277636e-
75%	2.670606e+07	2.748694e+07	2.830289e+07	2.914708e+07	3.001684e+07	3.084892e-
max	3.031474e+09	3.072422e+09	3.126850e+09	3.193429e+09	3.260442e+09	3.328209e-

8 rows × 63 columns

```
year_to_visualize = '2022'
data_for_year = data[['Country Name', year_to_visualize]]
data_for_year = data_for_year.sort_values(by=year_to_visualize, ascending=False)

# Select the first 40 countries
data_for_year_top50 = data_for_year.head(50)

# Create a vertical bar chart using Seaborn
plt.figure(figsize=(12, 10))
sns.barplot(y='Country Name', x=year_to_visualize, data=data_for_year_top50, orient='h')
plt.title(f'Top 50 Countries - Distribution in {year_to_visualize}')
plt.xlabel('Country Name')
plt.ylabel('Population')
plt.show()
```




```
year_2010 = '2010'
year_2020 = '2020'
data_selected_years = data[['Country Name', year_2010, year_2020]]

# Create a scatter plot using Seaborn
plt.figure(figsize=(10, 6))
sns.regplot(x=year_2010, y=year_2020, data=data_selected_years, line_kws={'color': 'cyan'})
plt.title(f'Population in {year_2010} vs. {year_2020}')
plt.xlabel(f'Population in {year_2010}')
plt.ylabel(f'Population in {year_2020}')
plt.show()
```