In [27]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```
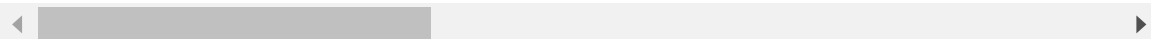
In [4]:
```python
df = pd.read_csv("D:\\Prodigy infotech task\\task1.csv")
df.head()
```

Out[4]:

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962
0	Aruba	ABW	Population, total	SP.POP.TOTL	54608.0	55811.0	56682.0
1	Africa Eastern and Southern	AFE	Population, total	SP.POP.TOTL	130692579.0	134169237.0	137835590.0	14
2	Afghanistan	AFG	Population, total	SP.POP.TOTL	8622466.0	8790140.0	8969047.0
3	Africa Western and Central	AFW	Population, total	SP.POP.TOTL	97256290.0	99314028.0	101445032.0	10
4	Angola	AGO	Population, total	SP.POP.TOTL	5357195.0	5441333.0	5521400.0

5 rows × 68 columns

◄ ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬ ►

In [5]:
```python
df.tail()
```

Out[5]:

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	
261	Kosovo	XKX	Population, total	SP.POP.TOTL	947000.0	966000.0	994000.0	1022
262	Yemen, Rep.	YEM	Population, total	SP.POP.TOTL	5542459.0	5646668.0	5753386.0	5860
263	South Africa	ZAF	Population, total	SP.POP.TOTL	16520441.0	16989464.0	17503133.0	1804
264	Zambia	ZMB	Population, total	SP.POP.TOTL	3119430.0	3219451.0	3323427.0	343
265	Zimbabwe	ZWE	Population, total	SP.POP.TOTL	3806310.0	3925952.0	4049778.0	417

5 rows × 68 columns

◄ ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬ ►
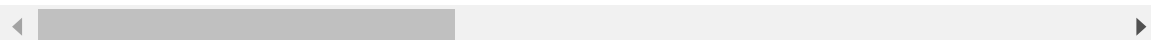
In [6]: `df.info()`

```
<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
```

In [8]: `df.shape`

Out[8]: `(266, 68)`

In [9]: `df.isna().sum()`

Out[9]:
```
Country Name        0
Country Code        0
Indicator Name      0
Indicator Code      0
1960                2
                  ...
2019                1
2020                1
2021                1
2022                1
2023              266
Length: 68, dtype: int64
```

In [10]:
```python
df = df.fillna(method="ffill")
df.head()
```

Out[10]:

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	
0	Aruba	ABW	Population, total	SP.POP.TOTL	54608.0	55811.0	56682.0	
1	Africa Eastern and Southern	AFE	Population, total	SP.POP.TOTL	130692579.0	134169237.0	137835590.0	14
2	Afghanistan	AFG	Population, total	SP.POP.TOTL	8622466.0	8790140.0	8969047.0	
3	Africa Western and Central	AFW	Population, total	SP.POP.TOTL	97256290.0	99314028.0	101445032.0	10
4	Angola	AGO	Population, total	SP.POP.TOTL	5357195.0	5441333.0	5521400.0	

5 rows × 68 columns
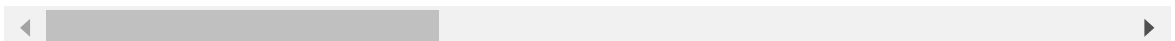
```
In [13]: df.isna().sum()
```

```
Out[13]: Country Name       0
         Country Code       0
         Indicator Name     0
         Indicator Code     0
         1960               0
                           ...
         2019               0
         2020               0
         2021               0
         2022               0
         2023             266
         Length: 68, dtype: int64
```

```
In [14]: df.duplicated().sum()
```

```
Out[14]: 0
```

```
In [15]: df['Country Name'].nunique()
```

```
Out[15]: 266
```

```
In [16]: df['Country Code'].nunique()
```

```
Out[16]: 266
```

```
In [19]: df['Indicator Code'].unique()
```

```
Out[19]: array(['SP.POP.TOTL'], dtype=object)
```

```
In [18]: df['Indicator Code'].value_counts()
```

```
Out[18]: SP.POP.TOTL     266
         Name: Indicator Code, dtype: int64
```

```
In [20]: df['Indicator Name'].value_counts()
```

```
Out[20]: Population, total     266
         Name: Indicator Name, dtype: int64
```

```
In [21]: df.drop(columns=['Country Code','Indicator Code','Indicator Name','2023'],i
```

In [22]: `df.columns`

Out[22]:
```
Index(['Country Name', '1960', '1961', '1962', '1963', '1964', '1965', '19
66',
       '1967', '1968', '1969', '1970', '1971', '1972', '1973', '1974', '19
75',
       '1976', '1977', '1978', '1979', '1980', '1981', '1982', '1983', '19
84',
       '1985', '1986', '1987', '1988', '1989', '1990', '1991', '1992', '19
93',
       '1994', '1995', '1996', '1997', '1998', '1999', '2000', '2001', '20
02',
       '2003', '2004', '2005', '2006', '2007', '2008', '2009', '2010', '20
11',
       '2012', '2013', '2014', '2015', '2016', '2017', '2018', '2019', '20
20',
       '2021', '2022'],
      dtype='object')
```

In [25]:
```python
cols=['1960','1961','1962','1963','1964','1965','1966','1967','1968','1969'
      '1971', '1972', '1973', '1974', '1975','1976', '1977', '1978', '1979'
      '1982', '1983', '1984''1985', '1986', '1987', '1988', '1989','1990',
      '1992', '1993','1994', '1995', '1996', '1997', '1998',  '1999', '2000
      '2002', '2003', '2004', '2005', '2006', '2007','2008', '2009', '2010'
      '2013', '2014', '2015', '2016','2017', '2018', '2019', '2020', '2021'
```
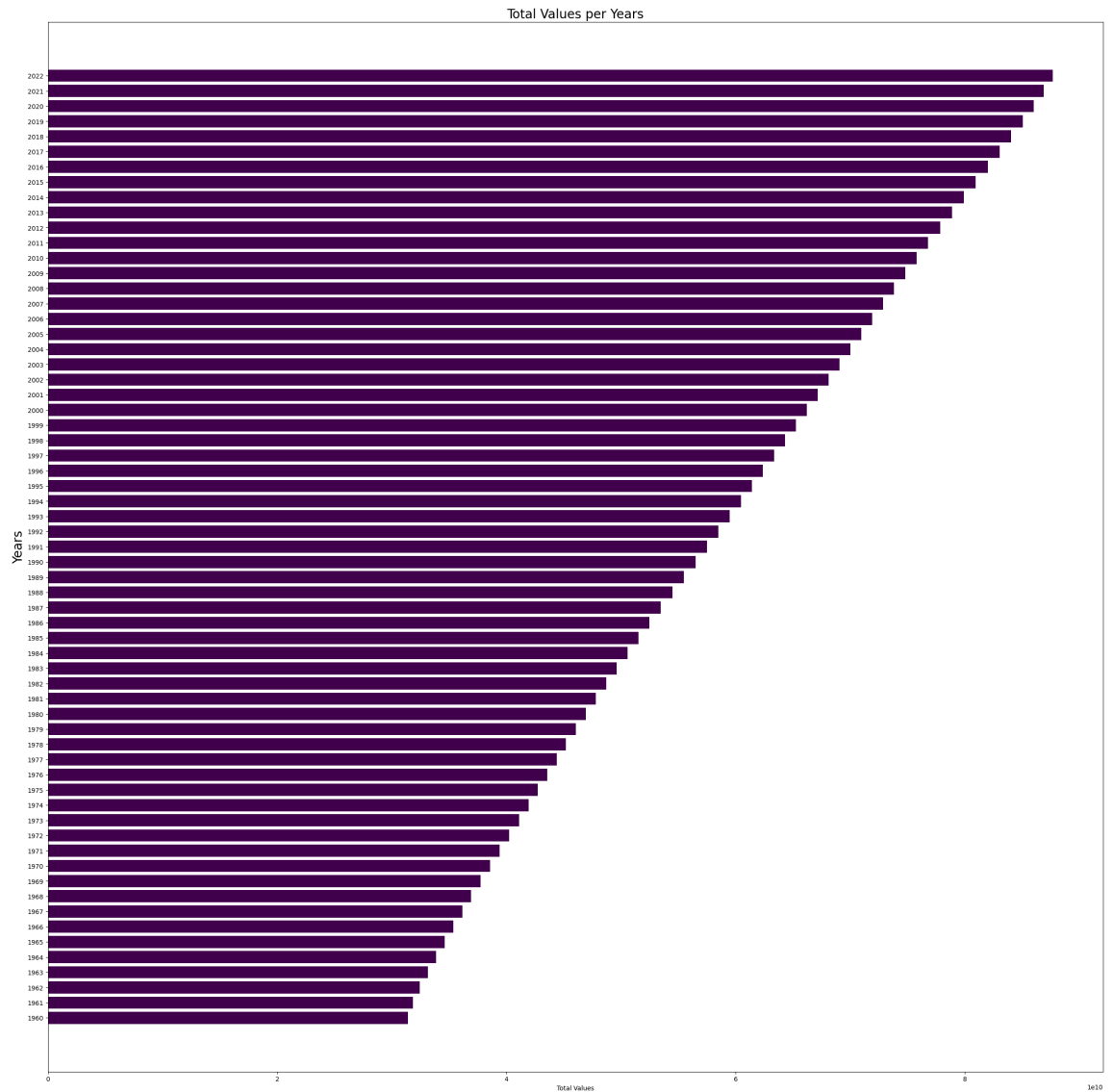
In [28]:
```python
for i in cols:
    fig = plt.figure(figsize=(5,5))
    plt.hist(df[i],color='#079eab',bins=10)
    plt.xlabel(i)
    plt.show()
```

```
In [34]: years = df.columns[1:]

         total_values = df[years].sum()

         plt.figure(figsize=(30, 30))
         plt.barh(years, total_values, color='#42024b')
         plt.xlabel('Total Values')
         plt.ylabel('Years', size=20)
         plt.title('Total Values per Years', size=20)
         plt.show()
```

In [35]:
```python
by_1960 = df.sort_values(by='1960').head(20)
by_1960
```

Out[35]:

	Country Name	1960	1961	1962	1963	1964	1965	1966	1967	
225	Sint Maarten (Dutch part)	2646.0	2888.0	3171.0	3481.0	3811.0	4161.0	4531.0	4930.0	5
147	St. Martin (French part)	4135.0	4258.0	4388.0	4524.0	4666.0	4832.0	5044.0	5294.0	5
179	Nauru	4582.0	4753.0	4950.0	5198.0	5484.0	5804.0	6021.0	6114.0	6
245	Tuvalu	5404.0	5436.0	5471.0	5503.0	5525.0	5548.0	5591.0	5657.0	5
228	Turks and Caicos Islands	5604.0	5625.0	5633.0	5634.0	5642.0	5650.0	5652.0	5662.0	5
255	British Virgin Islands	7850.0	7885.0	7902.0	7919.0	7949.0	8018.0	8139.0	8337.0	8
52	Cayman Islands	8473.0	8626.0	8799.0	8985.0	9172.0	9366.0	9566.0	9771.0	9
164	Northern Mariana Islands	8702.0	8965.0	9252.0	9561.0	9890.0	10229.0	10577.0	10720.0	10
6	Andorra	9443.0	10216.0	11014.0	11839.0	12690.0	13563.0	14546.0	15745.0	17
188	Palau	9446.0	9639.0	9851.0	10076.0	10318.0	10563.0	10813.0	10992.0	11
155	Marshall Islands	15374.0	15867.0	16387.0	16947.0	17537.0	18154.0	18794.0	19665.0	21
212	San Marino	15556.0	15895.0	16242.0	16583.0	16926.0	17273.0	17588.0	17907.0	18
137	Liechtenstein	16472.0	16834.0	17221.0	17625.0	18058.0	18500.0	18957.0	19467.0	20
11	American Samoa	20085.0	20626.0	21272.0	21949.0	22656.0	23391.0	24122.0	24848.0	25
149	Monaco	21797.0	21907.0	22106.0	22442.0	22766.0	23022.0	23198.0	23281.0	23
84	Gibraltar	21822.0	21907.0	22249.0	22796.0	23347.0	23910.0	24477.0	25047.0	25
91	Greenland	32500.0	33700.0	35000.0	36400.0	37600.0	39200.0	40500.0	41900.0	43
256	Virgin Islands (U.S.)	32500.0	34300.0	35000.0	39800.0	40800.0	43500.0	46200.0	49100.0	55
78	Faroe Islands	34154.0	34572.0	34963.0	35385.0	35841.0	36346.0	36825.0	37234.0	37
200	Qatar	36385.0	40111.0	45123.0	50950.0	57531.0	64843.0	73102.0	82517.0	93

20 rows × 64 columns
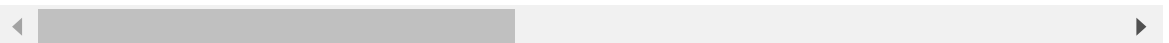
In [40]:
```python
by_1960_t = by_1960.set_index('Country Name').T
for country_name, data_values in by_1960_t.iterrows():
    fig = plt.figure(figsize=(10, 5))
    sns.barplot(x=data_values.index, y=data_values.values)
    plt.xlabel('Countries')
    plt.ylabel('Data Values')
    plt.title(f"{country_name} Data Values from 1960 to 2022")
    plt.xticks(rotation=90)
    plt.show()
```
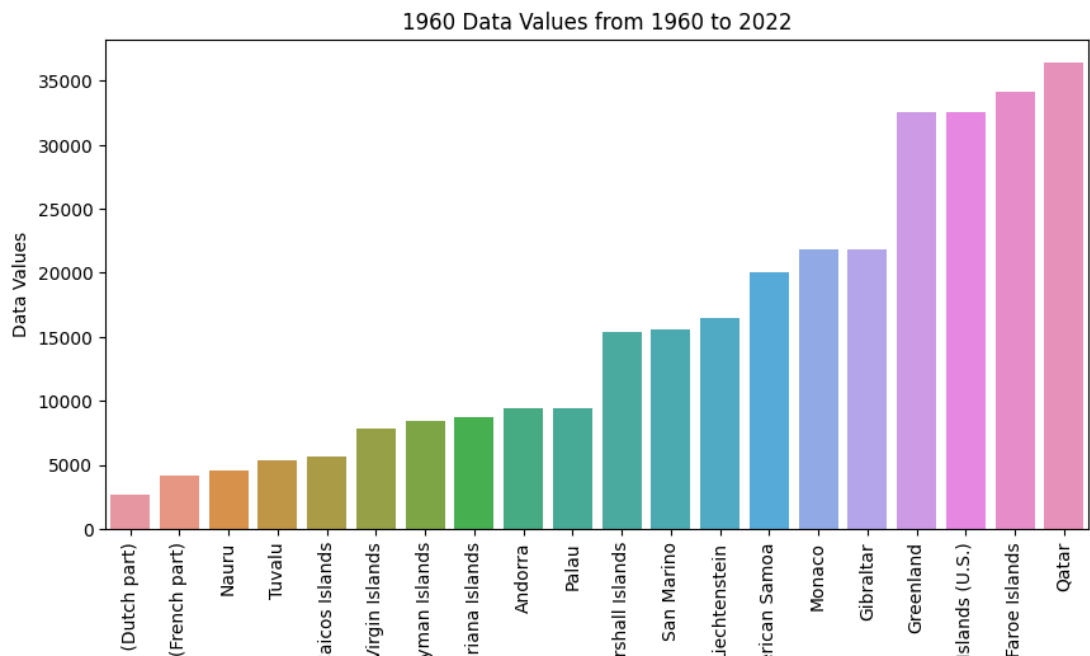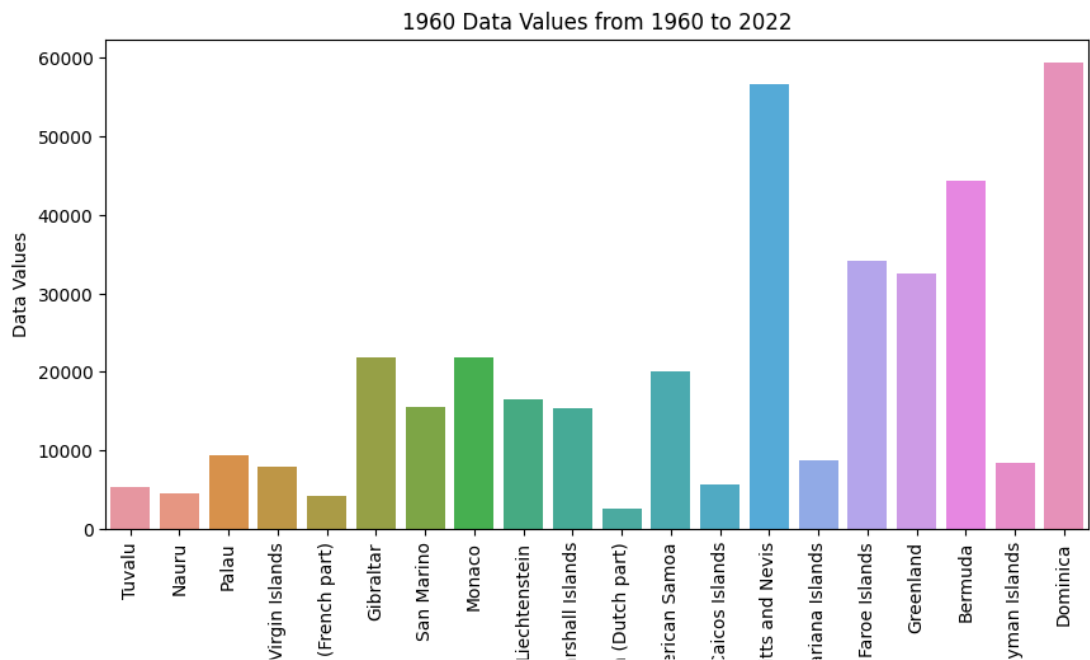

In [44]:
```python
by_2022 = df.sort_values(by='2022').head(20)
by_2023
```

Out[44]:

	Country Name	1960	1961	1962	1963	1964	1965	1966	1967
245	Tuvalu	5404.0	5436.0	5471.0	5503.0	5525.0	5548.0	5591.0	5657.0
179	Nauru	4582.0	4753.0	4950.0	5198.0	5484.0	5804.0	6021.0	6114.0
188	Palau	9446.0	9639.0	9851.0	10076.0	10318.0	10563.0	10813.0	10992.0
255	British Virgin Islands	7850.0	7885.0	7902.0	7919.0	7949.0	8018.0	8139.0	8337.0
147	St. Martin (French part)	4135.0	4258.0	4388.0	4524.0	4666.0	4832.0	5044.0	5294.0
84	Gibraltar	21822.0	21907.0	22249.0	22796.0	23347.0	23910.0	24477.0	25047.0
212	San Marino	15556.0	15895.0	16242.0	16583.0	16926.0	17273.0	17588.0	17907.0
149	Monaco	21797.0	21907.0	22106.0	22442.0	22766.0	23022.0	23198.0	23281.0
137	Liechtenstein	16472.0	16834.0	17221.0	17625.0	18058.0	18500.0	18957.0	19467.0

In [46]:
```python
by_2022_t = by_2022.set_index('Country Name').T
for country_name, data_values in by_2022_t.iterrows():
    fig = plt.figure(figsize=(10, 5))
    sns.barplot(x=data_values.index, y=data_values.values)
    plt.xlabel('year')
    plt.ylabel('Data Values')
    plt.title(f"{country_name} Data Values from 1960 to 2022")
    plt.xticks(rotation=90)
    plt.show()
```


In [ ]: