PRODIGY_DS_01

September 20, 2024

1 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

[4]:

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

Reading the dataset

[6]:

```
df=pd.read_csv('worldpopulationdata.csv')
```

Checking the first 5 rows

[8]:

```
df.head(5)
```

[8]:

```
         Series Name  Series Code    Country Name Country Code        2022  \
0  Population, total  SP.POP.TOTL     Afghanistan          AFG  41128771.0
1  Population, total  SP.POP.TOTL         Albania          ALB   2775634.0
2  Population, total  SP.POP.TOTL         Algeria          DZA  44903225.0
3  Population, total  SP.POP.TOTL  American Samoa          ASM     44273.0
4  Population, total  SP.POP.TOTL         Andorra          AND     79824.0

         2021        2020        2019        2018        2017  …  \
0  40099462.0  38972230.0  37769499.0  36686784.0  35643418.0  …
1   2811666.0   2837849.0   2854191.0   2866376.0   2873457.0  …
2  44177969.0  43451666.0  42705368.0  41927007.0  41136546.0  …
3     45035.0     46189.0     47321.0     48424.0     49463.0  …
4     79034.0     77700.0     76343.0     75013.0     73837.0  …

         2010        2009        2008        2007        2006        2005  \
0  28189672.0  27385307.0  26427199.0  25903301.0  25442944.0  24411191.0
1   2913021.0   2927519.0   2947314.0   2970017.0   2992547.0   3011487.0
2  35856344.0  35196037.0  34569592.0  33983827.0  33435080.0  32956690.0
3     54849.0     55366.0     55891.0     56383.0     56837.0     57254.0
4     71519.0     73852.0     76055.0     78168.0     80221.0     79826.0
```

```
         2004        2003        2002        2001
0  23553551.0  22645130.0  21000256.0  19688632.0
1   3026939.0   3039616.0   3051010.0   3060173.0
2  32510186.0  32055883.0  31624696.0  31200985.0
3     57626.0     57941.0     58177.0     58324.0
4     76933.0     73907.0     70849.0     67820.0

[5 rows x 26 columns]
```

Checking the last 5 rows

```
[10]: df.tail(5)
```

```
[10]:                                  Series Name        Series Code  \
      1080  Population, male (% of total population)  SP.POP.TOTL.MA.ZS
      1081  Population, male (% of total population)  SP.POP.TOTL.MA.ZS
      1082  Population, male (% of total population)  SP.POP.TOTL.MA.ZS
      1083  Population, male (% of total population)  SP.POP.TOTL.MA.ZS
      1084  Population, male (% of total population)  SP.POP.TOTL.MA.ZS

                      Country Name Country Code       2022       2021       2020  \
      1080  Virgin Islands (U.S.)          VIR  46.613382  46.764444  46.914637
      1081      West Bank and Gaza          PSE  49.893678  49.877839  49.858957
      1082             Yemen, Rep.          YEM  50.519031  50.538516  50.554317
      1083                  Zambia          ZMB  49.344602  49.344951  49.338301
      1084                Zimbabwe          ZWE  47.214139  47.167153  47.130679

                 2019       2018       2017  …       2010       2009       2008  \
      1080  47.057307  47.185912  47.314214  …  47.801059  47.834540  47.870063
      1081  49.835542  49.811374  49.785969  …  49.876336  49.898677  49.921445
      1082  50.571320  50.596614  50.616964  …  50.594170  50.582692  50.568876
      1083  49.326233  49.309087  49.288400  …  49.056379  48.981404  48.888443
      1084  47.099796  47.076238  47.051613  …  46.995893  47.049546  47.106068

                 2007       2006       2005       2004       2003       2002  \
      1080  47.877604  47.870702  47.852669  47.825150  47.789128  47.754932
      1081  49.947631  49.983323  50.028649  50.089953  50.167544  50.248196
      1082  50.553633  50.539012  50.522514  50.502720  50.481666  50.459941
      1083  48.784780  48.676944  48.571398  48.476900  48.393634  48.313646
      1084  47.166435  47.190963  47.231433  47.324096  47.387633  47.428426

                 2001
      1080  47.725126
      1081  50.321633
      1082  50.437238
      1083  48.229968
      1084  47.460469
```

```
[5 rows x 26 columns]
```

Checking the shape of the dataset

[12]:
```
df.shape
```

[12]:
```
(1085, 26)
```

Checking the columns of the dataset

[14]:
```
df.columns
```

[14]:
```
Index(['Series Name', 'Series Code', 'Country Name', 'Country Code', '2022',
       '2021', '2020', '2019', '2018', '2017', '2016', '2015', '2014', '2013',
       '2012', '2011', '2010', '2009', '2008', '2007', '2006', '2005', '2004',
       '2003', '2002', '2001'],
      dtype='object')
```

[16]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1085 entries, 0 to 1084
Data columns (total 26 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Series Name   1085 non-null   object
 1   Series Code   1085 non-null   object
 2   Country Name  1085 non-null   object
 3   Country Code  1085 non-null   object
 4   2022          1085 non-null   float64
 5   2021          1085 non-null   float64
 6   2020          1085 non-null   float64
 7   2019          1085 non-null   float64
 8   2018          1085 non-null   float64
 9   2017          1085 non-null   float64
 10  2016          1085 non-null   float64
 11  2015          1085 non-null   float64
 12  2014          1085 non-null   float64
 13  2013          1085 non-null   float64
 14  2012          1085 non-null   float64
 15  2011          1085 non-null   float64
 16  2010          1085 non-null   float64
 17  2009          1085 non-null   float64
 18  2008          1085 non-null   float64
 19  2007          1085 non-null   float64
 20  2006          1085 non-null   float64
 21  2005          1085 non-null   float64
 22  2004          1085 non-null   float64
 23  2003          1085 non-null   float64
 24  2002          1085 non-null   float64
 25  2001          1085 non-null   float64
dtypes: float64(22), object(4)
memory usage: 220.5+ KB
```

[18]:
```
df.describe()
```

[18]:
```
               2022          2021          2020          2019          2018  \
count  1.085000e+03  1.085000e+03  1.085000e+03  1.085000e+03  1.085000e+03
mean   1.461378e+07  1.449711e+07  1.437307e+07  1.422876e+07  1.407966e+07
std    7.832944e+07  7.801505e+07  7.763257e+07  7.712985e+07  7.657562e+07
min    2.749000e+01  2.732503e+01  2.735104e+01  2.676295e+01  2.573928e+01
25%    5.034029e+01  5.035172e+01  5.034171e+01  5.033040e+01  5.033917e+01
50%    1.465500e+05  1.463660e+05  1.461650e+05  1.459570e+05  1.457520e+05
75%    5.903468e+06  5.856733e+06  5.831404e+06  5.814422e+06  5.774185e+06
max    1.417173e+09  1.412360e+09  1.411100e+09  1.407745e+09  1.402760e+09

               2017          2016          2015          2014          2013  \
count  1.085000e+03  1.085000e+03  1.085000e+03  1.085000e+03  1.085000e+03
mean   1.392568e+07  1.376711e+07  1.360705e+07  1.344625e+07  1.328368e+07
std    7.596457e+07  7.528760e+07  7.461740e+07  7.394894e+07  7.325356e+07
min    2.508394e+01  2.464721e+01  2.474106e+01  2.540718e+01  2.594943e+01
25%    5.033041e+01  5.033966e+01  5.033554e+01  5.032504e+01  5.033767e+01
50%    1.441350e+05  1.406060e+05  1.371850e+05  1.349620e+05  1.328960e+05
75%    5.686999e+06  5.629265e+06  5.544490e+06  5.524552e+06  5.480089e+06
max    1.396215e+09  1.387790e+09  1.379860e+09  1.371860e+09  1.363240e+09

       …          2010          2009          2008          2007  \
count  …  1.085000e+03  1.085000e+03  1.085000e+03  1.085000e+03
mean   …  1.280537e+07  1.265031e+07  1.249535e+07  1.234099e+07
std    …  7.113128e+07  7.047509e+07  6.982016e+07  6.915934e+07
min    …  2.425072e+01  2.339422e+01  2.356750e+01  2.520779e+01
25%    …  5.034833e+01  5.036836e+01  5.037388e+01  5.036880e+01
50%    …  1.263090e+05  1.244660e+05  1.228070e+05  1.209490e+05
75%    …  5.267970e+06  5.187356e+06  5.100083e+06  5.062560e+06
max    …  1.337705e+09  1.331260e+09  1.324655e+09  1.317885e+09

               2006          2005          2004          2003          2002  \
count  1.085000e+03  1.085000e+03  1.085000e+03  1.085000e+03  1.085000e+03
mean   1.218858e+07  1.203685e+07  1.188626e+07  1.173626e+07  1.158653e+07
std    6.849229e+07  6.780708e+07  6.710041e+07  6.638386e+07  6.565651e+07
min    2.831990e+01  3.096426e+01  3.129133e+01  3.137472e+01  3.146521e+01
25%    5.038085e+01  5.037186e+01  5.036210e+01  5.039432e+01  5.039371e+01
50%    1.190890e+05  1.171330e+05  1.152950e+05  1.136960e+05  1.134500e+05
75%    5.007301e+06  4.989584e+06  4.813244e+06  4.758988e+06  4.698968e+06
```

```
max    1.311020e+09  1.303720e+09  1.296075e+09  1.288400e+09  1.280400e+09

               2001
count  1.085000e+03
mean   1.143598e+07
std    6.490862e+07
min    3.156689e+01
25%    5.038254e+01
50%    1.136410e+05
75%    4.535518e+06
max    1.271850e+09

[8 rows x 22 columns]
```

Checking for duplicate rows

[20]:
```
df.duplicated().sum()
```

[20]:
```
0
```

`Observation:` - There are no duplicate rows in the dataset

Checking for missing values

[22]:
```
df.isna().sum()
```

[22]:
```
Series Name     0
Series Code     0
Country Name    0
Country Code    0
2022            0
2021            0
2020            0
2019            0
2018            0
2017            0
2016            0
2015            0
2014            0
2013            0
2012            0
2011            0
2010            0
2009            0
2008            0
2007            0
2006            0
2005            0
```

```
2004            0
2003            0
2002            0
2001            0
dtype: int64
```

`Observation:` - no missing values present

Checking unique values for columns

[24]:
```
print(df['Country Name'].unique())
print("\nTotal no of unique countries:",df['Country Name'].nunique())
```

```
['Afghanistan' 'Albania' 'Algeria' 'American Samoa' 'Andorra' 'Angola'
 'Antigua and Barbuda' 'Argentina' 'Armenia' 'Aruba' 'Australia' 'Austria'
 'Azerbaijan' 'Bahamas, The' 'Bahrain' 'Bangladesh' 'Barbados' 'Belarus'
 'Belgium' 'Belize' 'Benin' 'Bermuda' 'Bhutan' 'Bolivia'
 'Bosnia and Herzegovina' 'Botswana' 'Brazil' 'British Virgin Islands'
 'Brunei Darussalam' 'Bulgaria' 'Burkina Faso' 'Burundi' 'Cabo Verde'
 'Cambodia' 'Cameroon' 'Canada' 'Cayman Islands'
 'Central African Republic' 'Chad' 'Channel Islands' 'Chile' 'China'
 'Colombia' 'Comoros' 'Congo, Dem. Rep.' 'Congo, Rep.' 'Costa Rica'
 "Cote d'Ivoire" 'Croatia' 'Cuba' 'Curacao' 'Cyprus' 'Czechia' 'Denmark'
 'Djibouti' 'Dominica' 'Dominican Republic' 'Ecuador' 'Egypt, Arab Rep.'
 'El Salvador' 'Equatorial Guinea' 'Eritrea' 'Estonia' 'Eswatini'
 'Ethiopia' 'Faroe Islands' 'Fiji' 'Finland' 'France' 'French Polynesia'
 'Gabon' 'Gambia, The' 'Georgia' 'Germany' 'Ghana' 'Gibraltar' 'Greece'
 'Greenland' 'Grenada' 'Guam' 'Guatemala' 'Guinea' 'Guinea-Bissau'
 'Guyana' 'Haiti' 'Honduras' 'Hong Kong SAR, China' 'Hungary' 'Iceland'
 'India' 'Indonesia' 'Iran, Islamic Rep.' 'Iraq' 'Ireland' 'Isle of Man'
 'Israel' 'Italy' 'Jamaica' 'Japan' 'Jordan' 'Kazakhstan' 'Kenya'
 'Kiribati' "Korea, Dem. People's Rep." 'Korea, Rep.' 'Kosovo' 'Kuwait'
 'Kyrgyz Republic' 'Lao PDR' 'Latvia' 'Lebanon' 'Lesotho' 'Liberia'
 'Libya' 'Liechtenstein' 'Lithuania' 'Luxembourg' 'Macao SAR, China'
 'Madagascar' 'Malawi' 'Malaysia' 'Maldives' 'Mali' 'Malta'
 'Marshall Islands' 'Mauritania' 'Mauritius' 'Mexico'
 'Micronesia, Fed. Sts.' 'Moldova' 'Monaco' 'Mongolia' 'Montenegro'
 'Morocco' 'Mozambique' 'Myanmar' 'Namibia' 'Nauru' 'Nepal' 'Netherlands'
 'New Caledonia' 'New Zealand' 'Nicaragua' 'Niger' 'Nigeria'
 'North Macedonia' 'Northern Mariana Islands' 'Norway' 'Oman' 'Pakistan'
 'Palau' 'Panama' 'Papua New Guinea' 'Paraguay' 'Peru' 'Philippines'
 'Poland' 'Portugal' 'Puerto Rico' 'Qatar' 'Romania' 'Russian Federation'
 'Rwanda' 'Samoa' 'San Marino' 'Sao Tome and Principe' 'Saudi Arabia'
 'Senegal' 'Serbia' 'Seychelles' 'Sierra Leone' 'Singapore'
 'Sint Maarten (Dutch part)' 'Slovak Republic' 'Slovenia'
 'Solomon Islands' 'Somalia' 'South Africa' 'South Sudan' 'Spain'
 'Sri Lanka' 'St. Kitts and Nevis' 'St. Lucia' 'St. Martin (French part)'
 'St. Vincent and the Grenadines' 'Sudan' 'Suriname' 'Sweden'
 'Switzerland' 'Syrian Arab Republic' 'Tajikistan' 'Tanzania' 'Thailand'
 'Timor-Leste' 'Togo' 'Tonga' 'Trinidad and Tobago' 'Tunisia' 'Turkiye'
 'Turkmenistan' 'Turks and Caicos Islands' 'Tuvalu' 'Uganda' 'Ukraine'
 'United Arab Emirates' 'United Kingdom' 'United States' 'Uruguay'
 'Uzbekistan' 'Vanuatu' 'Venezuela, RB' 'Vietnam' 'Virgin Islands (U.S.)'
 'West Bank and Gaza' 'Yemen, Rep.' 'Zambia' 'Zimbabwe']

Total no of unique countries: 217
```

[26]:
```
print(df['Country Code'].unique())
print("\nTotal no of unique country code:",df['Country Code'].nunique())
```

```
['AFG' 'ALB' 'DZA' 'ASM' 'AND' 'AGO' 'ATG' 'ARG' 'ARM' 'ABW' 'AUS' 'AUT'
 'AZE' 'BHS' 'BHR' 'BGD' 'BRB' 'BLR' 'BEL' 'BLZ' 'BEN' 'BMU' 'BTN' 'BOL'
 'BIH' 'BWA' 'BRA' 'VGB' 'BRN' 'BGR' 'BFA' 'BDI' 'CPV' 'KHM' 'CMR' 'CAN'
 'CYM' 'CAF' 'TCD' 'CHI' 'CHL' 'CHN' 'COL' 'COM' 'COD' 'COG' 'CRI' 'CIV'
 'HRV' 'CUB' 'CUW' 'CYP' 'CZE' 'DNK' 'DJI' 'DMA' 'DOM' 'ECU' 'EGY' 'SLV'
 'GNQ' 'ERI' 'EST' 'SWZ' 'ETH' 'FRO' 'FJI' 'FIN' 'FRA' 'PYF' 'GAB' 'GMB'
 'GEO' 'DEU' 'GHA' 'GIB' 'GRC' 'GRL' 'GRD' 'GUM' 'GTM' 'GIN' 'GNB' 'GUY'
 'HTI' 'HND' 'HKG' 'HUN' 'ISL' 'IND' 'IDN' 'IRN' 'IRQ' 'IRL' 'IMN' 'ISR'
 'ITA' 'JAM' 'JPN' 'JOR' 'KAZ' 'KEN' 'KIR' 'PRK' 'KOR' 'XKX' 'KWT' 'KGZ'
 'LAO' 'LVA' 'LBN' 'LSO' 'LBR' 'LBY' 'LIE' 'LTU' 'LUX' 'MAC' 'MDG' 'MWI'
 'MYS' 'MDV' 'MLI' 'MLT' 'MHL' 'MRT' 'MUS' 'MEX' 'FSM' 'MDA' 'MCO' 'MNG'
 'MNE' 'MAR' 'MOZ' 'MMR' 'NAM' 'NRU' 'NPL' 'NLD' 'NCL' 'NZL' 'NIC' 'NER'
 'NGA' 'MKD' 'MNP' 'NOR' 'OMN' 'PAK' 'PLW' 'PAN' 'PNG' 'PRY' 'PER' 'PHL'
 'POL' 'PRT' 'PRI' 'QAT' 'ROU' 'RUS' 'RWA' 'WSM' 'SMR' 'STP' 'SAU' 'SEN'
 'SRB' 'SYC' 'SLE' 'SGP' 'SXM' 'SVK' 'SVN' 'SLB' 'SOM' 'ZAF' 'SSD' 'ESP'
 'LKA' 'KNA' 'LCA' 'MAF' 'VCT' 'SDN' 'SUR' 'SWE' 'CHE' 'SYR' 'TJK' 'TZA'
 'THA' 'TLS' 'TGO' 'TON' 'TTO' 'TUN' 'TUR' 'TKM' 'TCA' 'TUV' 'UGA' 'UKR'
 'ARE' 'GBR' 'USA' 'URY' 'UZB' 'VUT' 'VEN' 'VNM' 'VIR' 'PSE' 'YEM' 'ZMB'
 'ZWE']

Total no of unique country code: 217
```

[122]:
```
df['Series Name'].unique()
```

[122]:
```
array(['Population, total', 'Population, female', 'Population, male',
       'Population, female (% of total population)',
       'Population, male (% of total population)'], dtype=object)
```

[123]:
```
df['Series Code'].unique()
```

[123]:
```
array(['SP.POP.TOTL', 'SP.POP.TOTL.FE.IN', 'SP.POP.TOTL.MA.IN',
       'SP.POP.TOTL.FE.ZS', 'SP.POP.TOTL.MA.ZS'], dtype=object)
```

Dropping unnecessary columns

[28]:
```
df.drop(['Series Name','Country Code'],axis=1,inplace=True)
```

[30]:
```
df.columns
```

[30]:
```
Index(['Series Code', 'Country Name', '2022', '2021', '2020', '2019', '2018',
       '2017', '2016', '2015', '2014', '2013', '2012', '2011', '2010', '2009',
       '2008', '2007', '2006', '2005', '2004', '2003', '2002', '2001'],
      dtype='object')
```

Extraction of top-10 countries with respect to total population

[32]:
```
total_population_data = df[df['Series Code'] == 'SP.POP.TOTL']

# Sort data based on the total population for 2022
total_population_sorted = total_population_data.sort_values(by="2022",␣
 ↪ascending=False)
total_top_ten_countries = total_population_sorted.head(10)
print("Top ten countries of total population\n")
print(total_top_ten_countries[['Country Name']] )
```

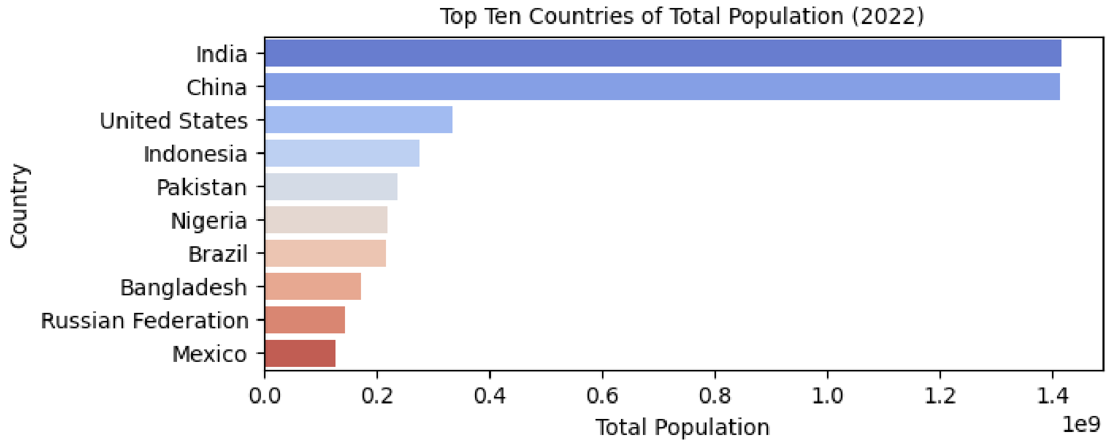
```
Top ten countries of total population

           Country Name
89                India
41                China
206       United States
90            Indonesia
149            Pakistan
144             Nigeria
26               Brazil
15           Bangladesh
161  Russian Federation
127              Mexico
```

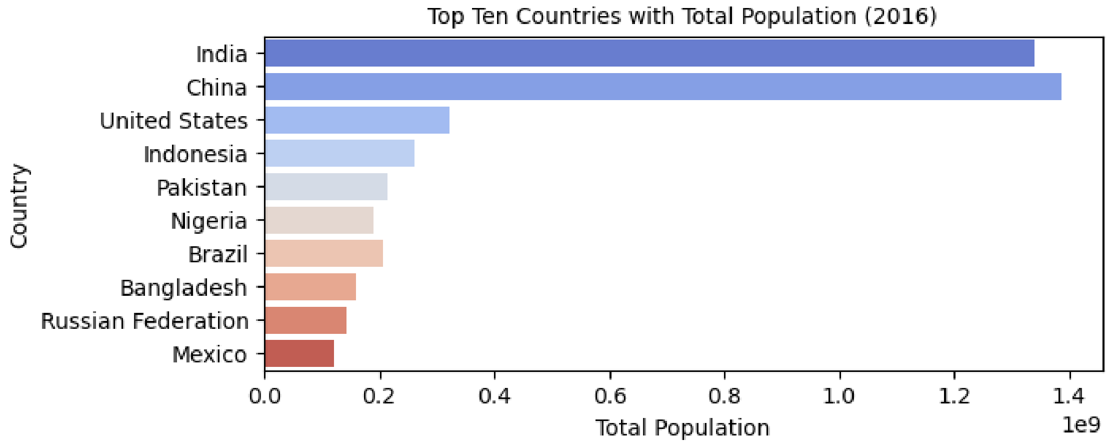
1.1 Bar Plot

Top ten countries of total population in year 2022 and 2016

[34]:
```
# Create the bar plot
plt.figure(figsize=(15, 6))
plt.subplot(2,2,1)
sns.barplot(x="2022", y="Country Name", data=total_top_ten_countries,␣
 ↪palette="coolwarm")
plt.title("Top Ten Countries of Total Population (2022)",fontsize=10)
plt.xlabel("Total Population",fontsize=10)
plt.ylabel("Country",fontsize=10)
plt.show()

plt.figure(figsize=(15, 6))
plt.subplot(2,2,2)
```

```
sns.barplot(x="2016", y="Country Name", data=total_top_ten_countries,␣
 ↪palette="coolwarm")
plt.title("Top Ten Countries with Total Population (2016)",fontsize=10)
plt.xlabel("Total Population",fontsize=10)
plt.ylabel("Country",fontsize=10)
plt.show()
```

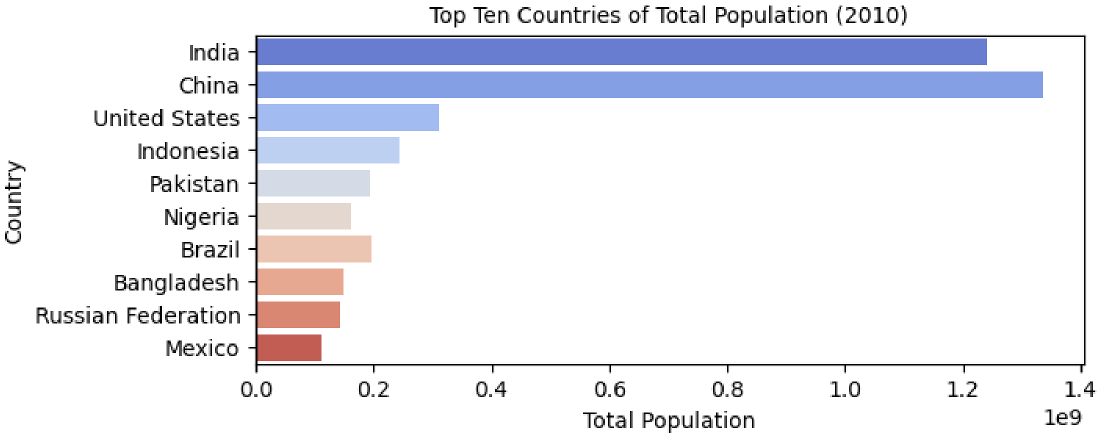


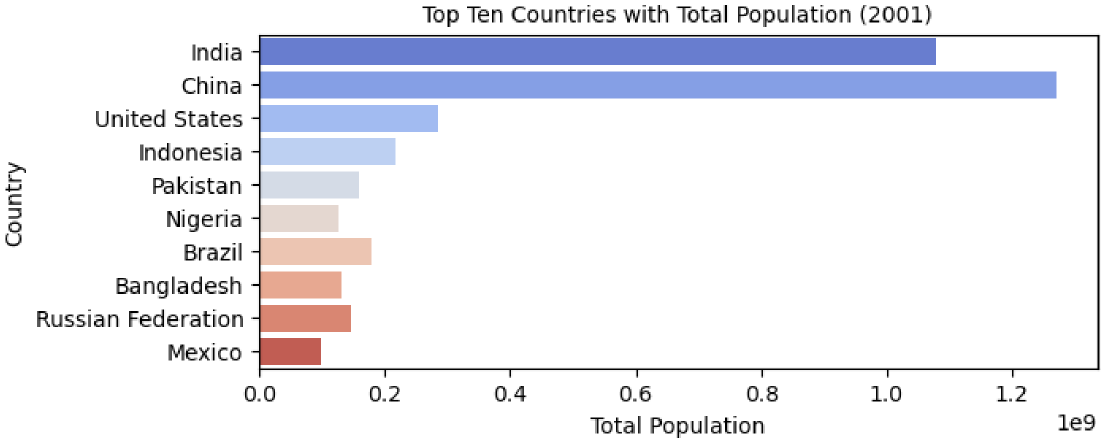


Top ten countries of total population in year 2010 and 2001

[36]:

```
# Create the bar plot
plt.figure(figsize=(15, 6))
plt.subplot(2,2,1)
sns.barplot(x="2010", y="Country Name", data=total_top_ten_countries,␣
 ↪palette="coolwarm")
plt.title("Top Ten Countries of Total Population (2010)",fontsize=10)
plt.xlabel("Total Population",fontsize=10)
plt.ylabel("Country",fontsize=10)
plt.show()

plt.figure(figsize=(15, 6))
plt.subplot(2,2,2)
sns.barplot(x="2001", y="Country Name", data=total_top_ten_countries,␣
 ↪palette="coolwarm")
plt.title("Top Ten Countries with Total Population (2001)",fontsize=10)
plt.xlabel("Total Population",fontsize=10)
plt.ylabel("Country",fontsize=10)
plt.show()
```





Extraction of bottom-10 countries with respect to total population

[38]:

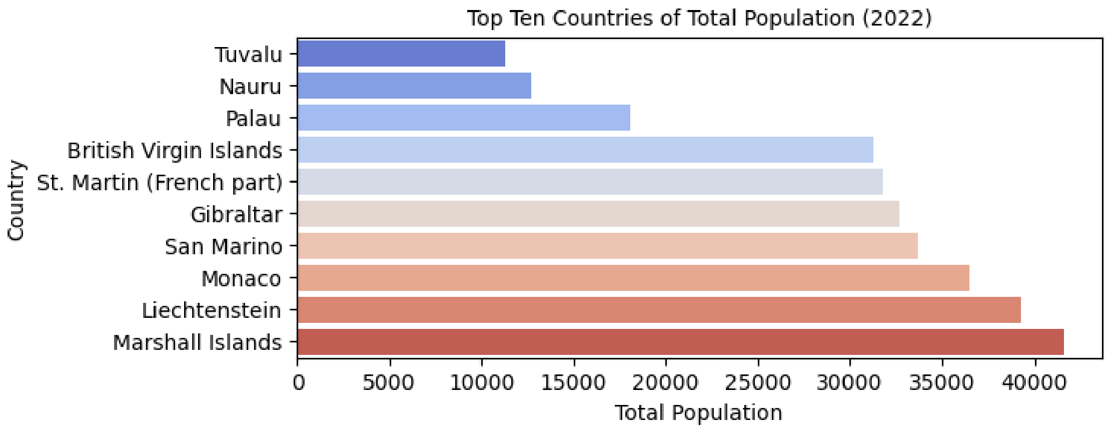
```
# Sort data based on the total population for 2022
total_population_sorted1 = total_population_data.sort_values(by="2022",␣
 ↪ascending=True)
total_bottom_ten_countries = total_population_sorted1.head(10)
print("Bottom ten countries of total population\n")
print(total_bottom_ten_countries[['Country Name']] )
```

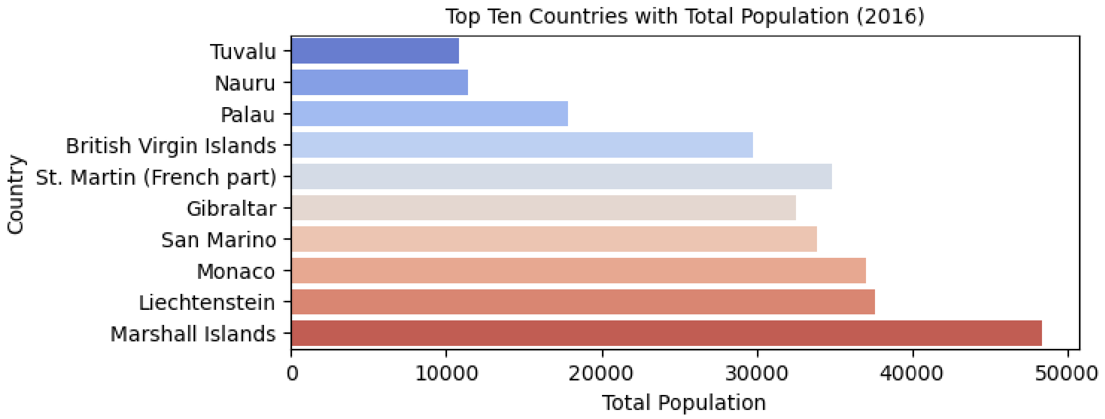
```
Bottom ten countries of total population

                 Country Name
201                    Tuvalu
137                     Nauru
150                     Palau
27     British Virgin Islands
183  St. Martin (French part)
75                  Gibraltar
164                San Marino
130                    Monaco
114             Liechtenstein
124          Marshall Islands
```

Bottom ten countries of total population in year 2022 and 2016

[40]:

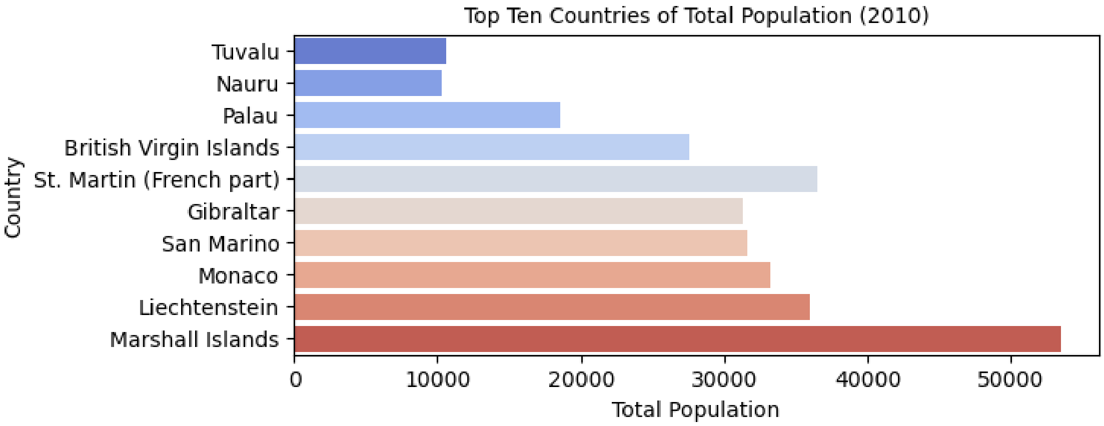
```
# Create the bar plot
plt.figure(figsize=(15, 6))
plt.subplot(2,2,1)
sns.barplot(x="2022", y="Country Name", data=total_bottom_ten_countries,␣
 ↪palette="coolwarm")
plt.title("Top Ten Countries of Total Population (2022)",fontsize=10)
plt.xlabel("Total Population",fontsize=10)
plt.ylabel("Country",fontsize=10)
plt.show()

plt.figure(figsize=(15, 6))
plt.subplot(2,2,2)
sns.barplot(x="2016", y="Country Name", data=total_bottom_ten_countries,␣
 ↪palette="coolwarm")
plt.title("Top Ten Countries with Total Population (2016)",fontsize=10)
plt.xlabel("Total Population",fontsize=10)
plt.ylabel("Country",fontsize=10)
plt.show()
```

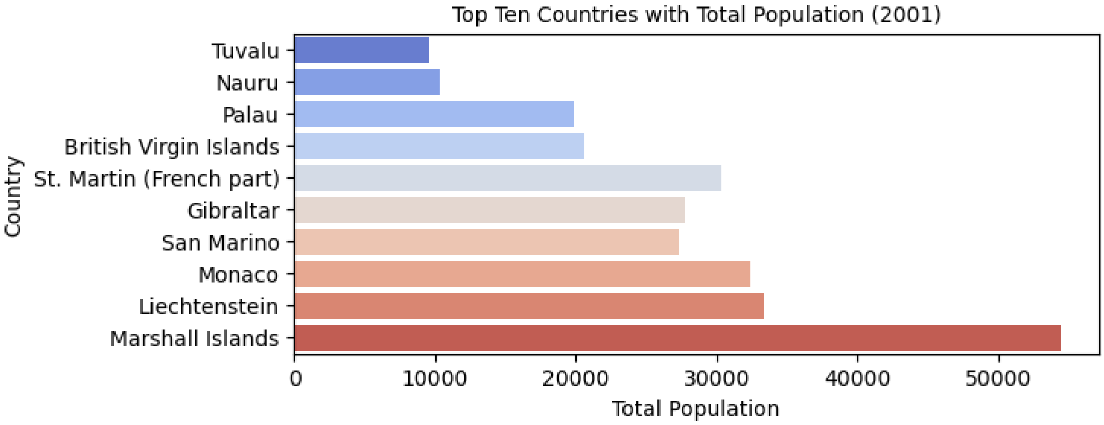




Bottom ten countries of total population in year 2010 and 2001

[41]:
```
# Create the bar plot
plt.figure(figsize=(15, 6))
plt.subplot(2,2,1)
sns.barplot(x="2010", y="Country Name", data=total_bottom_ten_countries,␣
  ↪palette="coolwarm")
plt.title("Top Ten Countries of Total Population (2010)",fontsize=10)
plt.xlabel("Total Population",fontsize=10)
plt.ylabel("Country",fontsize=10)
plt.show()

plt.figure(figsize=(15, 6))
plt.subplot(2,2,2)
sns.barplot(x="2001", y="Country Name", data=total_bottom_ten_countries,␣
  ↪palette="coolwarm")
plt.title("Top Ten Countries with Total Population (2001)",fontsize=10)
plt.xlabel("Total Population",fontsize=10)
plt.ylabel("Country",fontsize=10)
plt.show()
```





#### Extraction of top ten countries with highest male population

[44]:

```
# Filter data for male population
male_population_data = df[df["Series Code"] == "SP.POP.TOTL.MA.IN"]
male_population_sorted = male_population_data.sort_values(by="2022",␣
  ↪ascending=False)
male_top_ten_countries = male_population_sorted.head(10)
print("Top ten countries of male population")
print(male_top_ten_countries[['Country Name']] )
```

```
Top ten countries of male population
           Country Name
523               India
```

```
475                China
640        United States
524            Indonesia
583             Pakistan
578              Nigeria
460               Brazil
449           Bangladesh
595   Russian Federation
561               Mexico
```

Extraction of top ten countries with highest female population

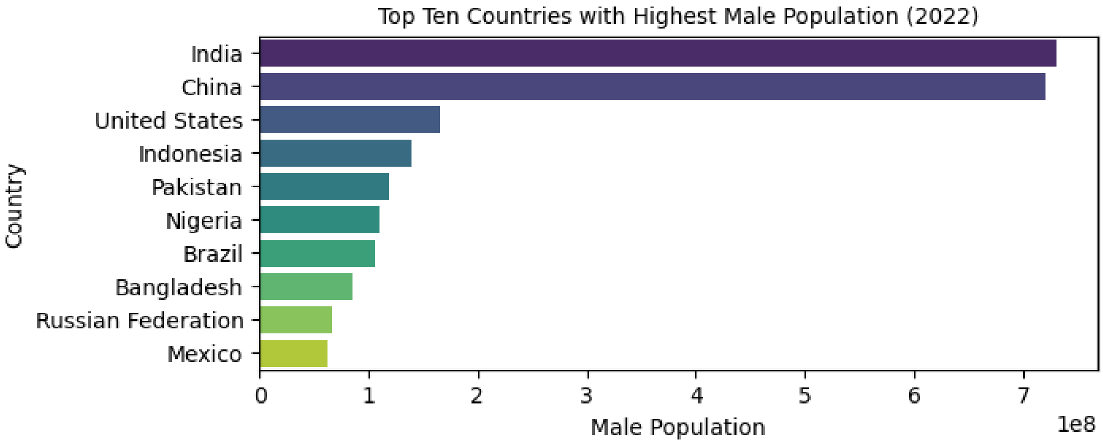
[46]:
```
# Filter data for female population
female_population_data = df[df["Series Code"] == "SP.POP.TOTL.FE.IN"]
female_population_sorted = female_population_data.sort_values(by="2022",␣
 ↪ascending=False)
female_top_ten_countries = female_population_sorted.head(10)
print("Top ten countries of female population")
print(female_top_ten_countries[['Country Name']] )
```

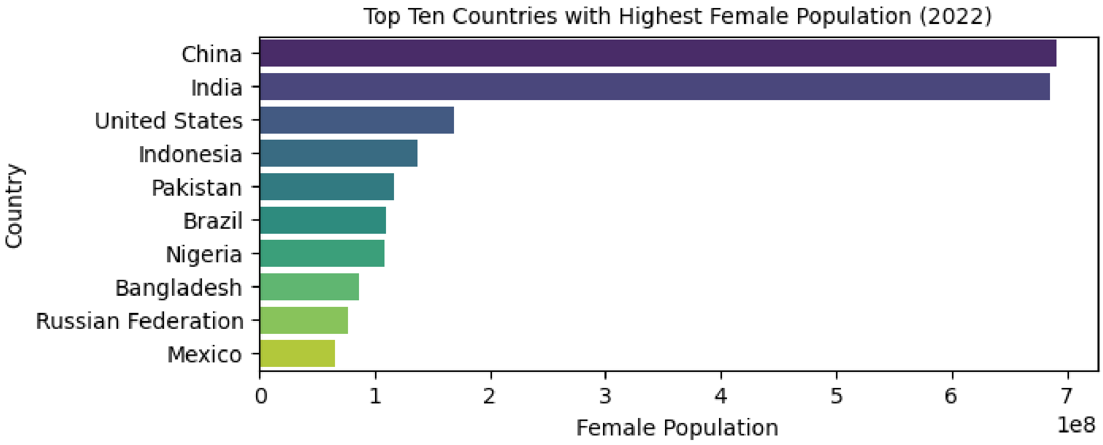
```
Top ten countries of female population
           Country Name
258               China
306               India
423       United States
307           Indonesia
366            Pakistan
243              Brazil
361             Nigeria
232          Bangladesh
378  Russian Federation
344              Mexico
```

Top ten countries with highest male and female population in 2022

[48]:
```
# Create the bar plot
plt.figure(figsize=(15, 6))
plt.subplot(2,2,1)
sns.barplot(x="2022", y="Country Name", data=male_top_ten_countries,␣
 ↪palette="viridis")
plt.title("Top Ten Countries with Highest Male Population (2022)",size=10)
plt.xlabel("Male Population",size=10)
plt.ylabel("Country",size=10)
plt.show()

plt.figure(figsize=(15, 6))
plt.subplot(2,2,2)
```

```
sns.barplot(x="2022", y="Country Name", data=female_top_ten_countries,␣
 ↪palette="viridis")
plt.title("Top Ten Countries with Highest Female Population (2022)",size=10)
plt.xlabel("Female Population",size=10)
plt.ylabel("Country",size=10)
plt.show()
```





Extraction of top ten countries with lowest male population

[50]:
```
male_lowest_ten_countries = male_population_sorted.tail(10)
print("Top ten countries of lowest male population")
print(male_lowest_ten_countries[['Country Name']] )
```

```
Top ten countries of lowest male population
                  Country Name
558           Marshall Islands
```

```
548             Liechtenstein
564                    Monaco
598                San Marino
509                 Gibraltar
617  St. Martin (French part)
461    British Virgin Islands
584                     Palau
571                     Nauru
635                    Tuvalu
```

Extraction of top ten countries with lowest female population

[52]:

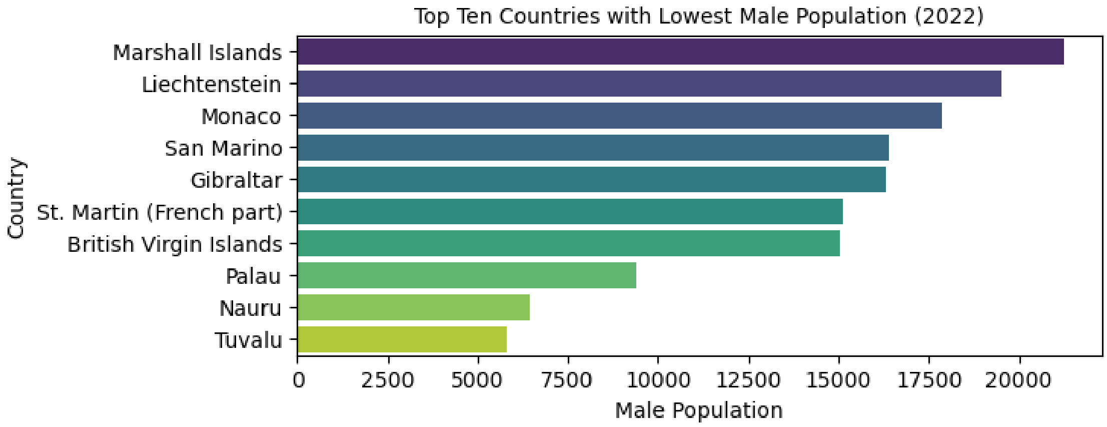
```
female_lowest_ten_countries = female_population_sorted.tail(10)
print("Top ten countries of lowest female population")
print(female_lowest_ten_countries[['Country Name']] )
```

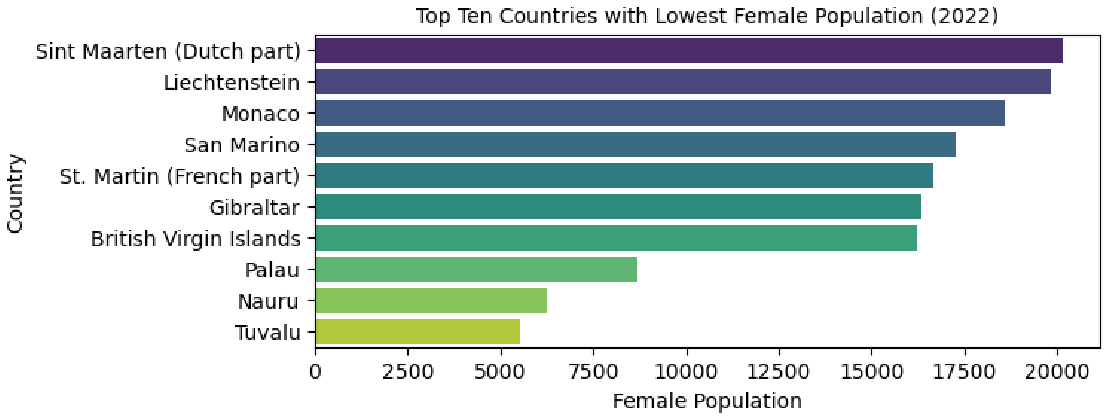
```
Top ten countries of lowest female population
                  Country Name
389  Sint Maarten (Dutch part)
331              Liechtenstein
347                     Monaco
381                 San Marino
400   St. Martin (French part)
292                  Gibraltar
244     British Virgin Islands
367                      Palau
354                      Nauru
418                     Tuvalu
```

Top ten countries with lowest male and female population in 2022

[54]:

```
# Create the bar plot
plt.figure(figsize=(15, 6))
plt.subplot(2,2,1)
sns.barplot(x="2022", y="Country Name", data=male_lowest_ten_countries,␣
  ↪palette="viridis")
plt.title("Top Ten Countries with Lowest Male Population (2022)",size=10)
plt.xlabel("Male Population",size=10)
plt.ylabel("Country",size=10)
plt.show()

plt.figure(figsize=(15, 6))
plt.subplot(2,2,2)
sns.barplot(x="2022", y="Country Name", data=female_lowest_ten_countries,␣
  ↪palette="viridis")
plt.title("Top Ten Countries with Lowest Female Population (2022)",size=10)
plt.xlabel("Female Population",size=10)
plt.ylabel("Country",size=10)
plt.show()
```





1.2 Stacked Bar Plot

Top 10 Countries with Male and Female Populations (2022)

[55]:
```
# Merge male and female population data on 'Country Name'
merged_data = pd.merge(male_population_data, female_population_data,␣
  ↪on="Country Name", suffixes=("_male", "_female"))
```

[172]:
```
merged_data
```

[172]:
```
        Series Code_male            Country Name    2022_male    2021_male  \
0     SP.POP.TOTL.MA.IN              Afghanistan   20766442.0   20254878.0
1     SP.POP.TOTL.MA.IN                  Albania    1384548.0    1404454.0
2     SP.POP.TOTL.MA.IN                  Algeria   22862237.0   22497244.0
3     SP.POP.TOTL.MA.IN           American Samoa      21873.0      22289.0
```

```
4    SP.POP.TOTL.MA.IN                Andorra     40786.0     40361.0
..                 …                        …           …           …
212  SP.POP.TOTL.MA.IN  Virgin Islands (U.S.)     49137.0     49510.0
213  SP.POP.TOTL.MA.IN     West Bank and Gaza   2516444.0   2455361.0
214  SP.POP.TOTL.MA.IN            Yemen, Rep.  17023203.0  16668432.0
215  SP.POP.TOTL.MA.IN                 Zambia   9877642.0   9609004.0
216  SP.POP.TOTL.MA.IN               Zimbabwe   7705601.0   7543690.0

      2020_male   2019_male   2018_male   2017_male   2016_male   2015_male  \
0    19692301.0  19090409.0  18549862.0  18028696.0  17520861.0  17071446.0
1     1419264.0   1428828.0   1435881.0   1440219.0   1442176.0   1444890.0
2    22132899.0  21756903.0  21362603.0  20961313.0  20556314.0  20152232.0
3       22921.0     23535.0     24134.0     24701.0     25240.0     25739.0
4       39615.0     38842.0     38071.0     37380.0     36628.0     36188.0
..          …           …           …           …          …           …
212     49866.0     50196.0     50489.0     50759.0     50999.0     51208.0
213   2394860.0   2334948.0   2275925.0   2217868.0   2173706.0   2125660.0
214  16320979.0  15953578.0  15578957.0  15202496.0  14820156.0  14439156.0
215   9338613.0   9066397.0   8794716.0   8525934.0   8260471.0   8000338.0
216   7385220.0   7231989.0   7086002.0   6940631.0   6796658.0   6652836.0

     …  2010_female  2009_female  2008_female  2007_female  2006_female  \
0    …   13949295.0   13557331.0   13088192.0   12835340.0   12614497.0
1    …    1454108.0    1462978.0    1474838.0    1488396.0    1501918.0
2    …   17573708.0   17249096.0   16941031.0   16653361.0   16384158.0
3    …      27189.0      27406.0      27626.0      27842.0      28044.0
4    …      35212.0      36065.0      36864.0      37633.0      38392.0
..   …          …            …            …            …            …
212  …      56560.0      56549.0      56507.0      56467.0      56492.0
213  …    1897763.0    1848287.0    1798811.0    1749079.0    1703735.0
214  …   12224951.0   11874775.0   11531790.0   11195418.0   10864747.0
215  …    7026189.0    6794701.0    6569350.0    6351748.0    6144175.0
216  …    6805605.0    6714016.0    6638373.0    6578079.0    6511613.0

     2005_female  2004_female  2003_female  2002_female  2001_female
0     12109086.0   11690825.0   11247647.0   10438055.0    9793166.0
1      1513578.0    1523393.0    1531532.0    1538490.0    1543533.0
2     16150274.0   15932047.0   15709725.0   15497822.0   15288132.0
3        28230.0      28392.0      28521.0      28608.0      28649.0
4        38147.0      36852.0      35478.0      34076.0      32669.0
..          …            …            …            …            …
212      56555.0      56593.0      56652.0      56692.0      56744.0
213    1659247.0    1615402.0    1572199.0    1530053.0    1489250.0
214   10548931.0   10262472.0    9997157.0    9739899.0    9488026.0
215    5947650.0    5764425.0    5593084.0    5431354.0    5276383.0
216    6450827.0    6405855.0    6353380.0    6300516.0    6257972.0
```

```
[217 rows x 47 columns]
```

[58]:
```
# Calculate the total population for each country (male + female)
merged_data["Total Population"] = merged_data["2022_male"] +␣
 ↪merged_data["2022_female"]
```

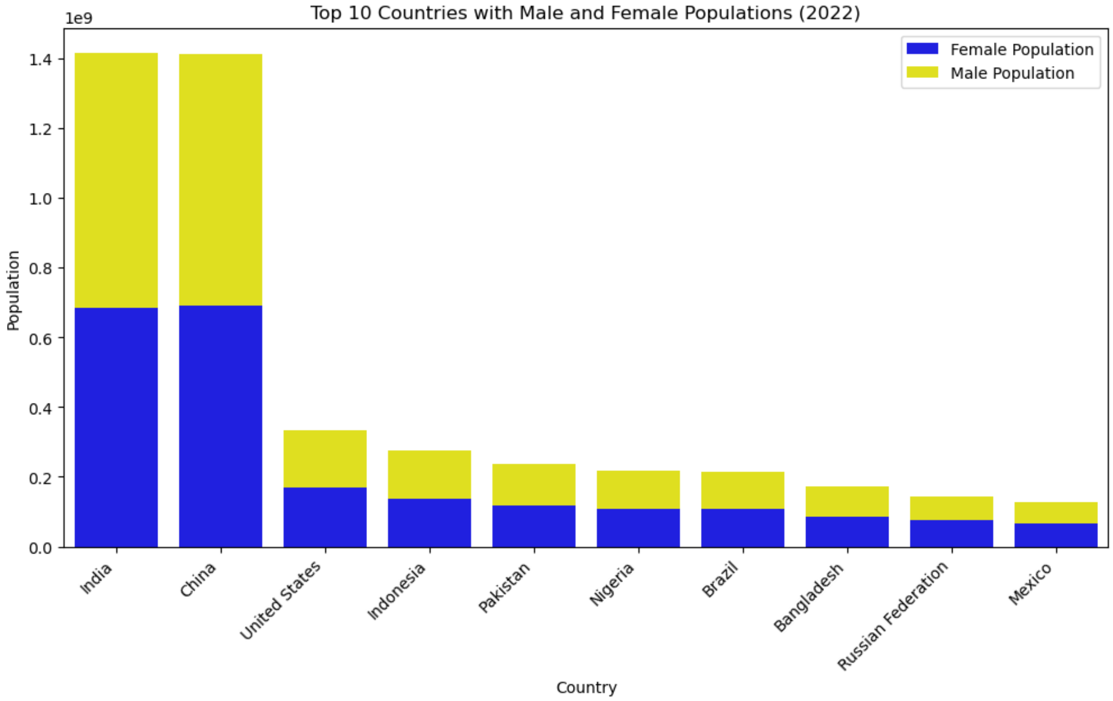
[60]:
```
merged_data.head()
```

[60]:
```
      Series Code_male    Country Name   2022_male   2021_male   2020_male  \
0  SP.POP.TOTL.MA.IN     Afghanistan  20766442.0  20254878.0  19692301.0
1  SP.POP.TOTL.MA.IN         Albania   1384548.0   1404454.0   1419264.0
2  SP.POP.TOTL.MA.IN         Algeria  22862237.0  22497244.0  22132899.0
3  SP.POP.TOTL.MA.IN  American Samoa     21873.0     22289.0     22921.0
4  SP.POP.TOTL.MA.IN         Andorra     40786.0     40361.0     39615.0

    2019_male   2018_male   2017_male   2016_male   2015_male  …  \
0  19090409.0  18549862.0  18028696.0  17520861.0  17071446.0  …
1   1428828.0   1435881.0   1440219.0   1442176.0   1444890.0  …
2  21756903.0  21362603.0  20961313.0  20556314.0  20152232.0  …
3     23535.0     24134.0     24701.0     25240.0     25739.0  …
4     38842.0     38071.0     37380.0     36628.0     36188.0  …

   2009_female  2008_female  2007_female  2006_female  2005_female  \
0   13557331.0   13088192.0   12835340.0   12614497.0   12109086.0
1    1462978.0    1474838.0    1488396.0    1501918.0    1513578.0
2   17249096.0   16941031.0   16653361.0   16384158.0   16150274.0
3      27406.0      27626.0      27842.0      28044.0      28230.0
4      36065.0      36864.0      37633.0      38392.0      38147.0

   2004_female  2003_female  2002_female  2001_female  Total Population
0   11690825.0   11247647.0   10438055.0    9793166.0        41128771.0
1    1523393.0    1531532.0    1538490.0    1543533.0         2775633.0
2   15932047.0   15709725.0   15497822.0   15288132.0        44903224.0
3      28392.0      28521.0      28608.0      28649.0           44272.0
4      36852.0      35478.0      34076.0      32669.0           79824.0

[5 rows x 48 columns]
```

[62]:
```
sorted_data = merged_data.sort_values(by="Total Population", ascending=False)
top_10_countries = sorted_data.head(10)
```

[72]:
```
# Create the stacked bar plot
plt.figure(figsize=(12, 6))
sns.barplot(x="Country Name", y="2022_female", data=top_10_countries,␣
 ↪color="blue", label="Female Population")
```

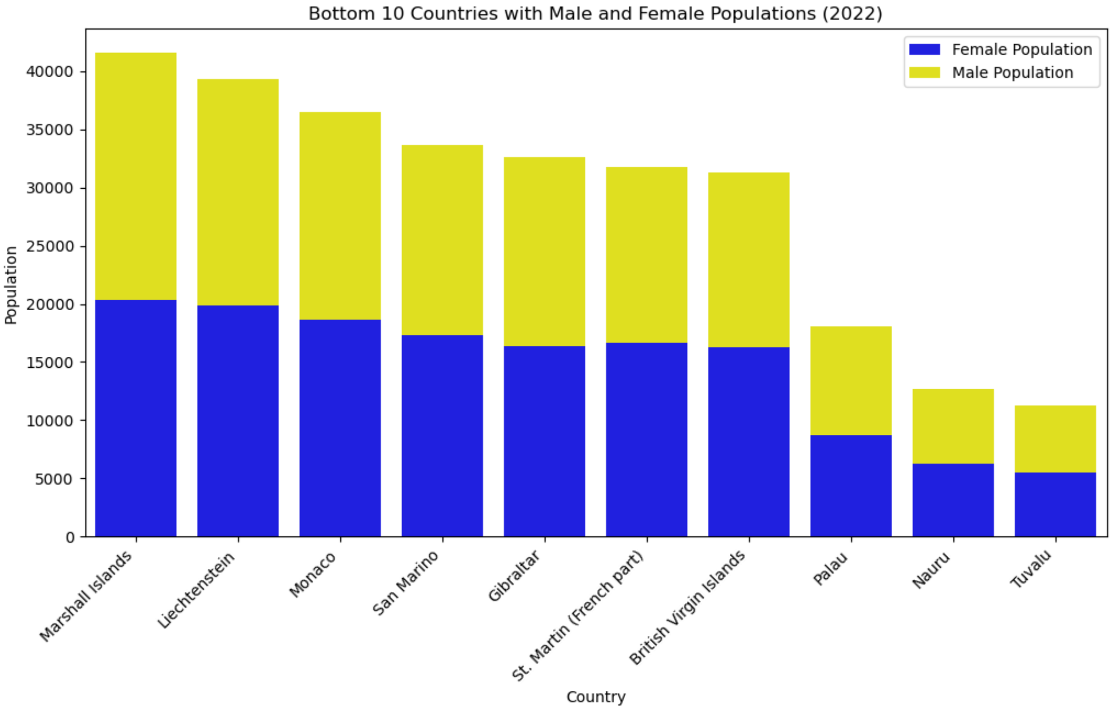
```
sns.barplot(x="Country Name", y="2022_male", data=top_10_countries,␣
 ↪bottom=top_10_countries["2022_female"], color="yellow", label="Male␣
 ↪Population")
plt.title("Top 10 Countries with Male and Female Populations (2022)")
plt.xlabel("Country")
plt.ylabel("Population")
plt.legend()
plt.xticks(rotation=45, ha="right")
plt.show()
```



Bottom 10 Countries with Male and Female Populations (2022)

[65]:
```
bottom_10_countries = sorted_data.tail(10)
```

[70]:
```
plt.figure(figsize=(12, 6))
sns.barplot(x="Country Name", y="2022_female", data=bottom_10_countries,␣
 ↪color="blue", label="Female Population")
sns.barplot(x="Country Name", y="2022_male", data=bottom_10_countries,␣
 ↪bottom=bottom_10_countries["2022_female"], color="yellow", label="Male␣
 ↪Population")
plt.title("Bottom 10 Countries with Male and Female Populations (2022)")
plt.xlabel("Country")
plt.ylabel("Population")
plt.legend()
plt.xticks(rotation=45, ha="right")
plt.show()
```


Bottom 10 Countries with Male and Female Populations (2022)
Female Population
Male Population
Population
Marshall Islands
Liechtenstein
Monaco
San Marino
Gibraltar
St. Martin (French part)
British Virgin Islands
Palau
Nauru
Tuvalu
Country