	importing necessary libraries import pandas as pd import numpy as np import matplotlib.pyplot as plt import seaborn as sns Reading CSV file
	df=pd.read_csv("worldpopulationdata.csv") checking top 5 rows data df.head(5) Series Name Series Code Country Name Country Code 2022 2021 2020 2019 2018 2017 2010 2009 2008 2007 2006 2005 2004 2003 2009 2001 2000 2001 2001 2001 2001 2001
	1 Population, total SP.POP.TOTL Albania ALB 2775634.0 2811666.0 2837849.0 2854191.0 2866376.0 2873457.0 2913021.0 2927519.0 2947314.0 2970017.0 2992547.0 3011487.0 3026939.0 3039616.0 3051010.0 3060173.0 291011.0 29
	checking data from bottom df.tail(5) Series Name Series Code Country Name Country Code 2022 2021 2020 2019 2018 2017 2010 2009 2008 2007 2006 2005 2004 2003 2002 1080 Population, male (% of total SP.POP.TOTL.MA.ZS Virgin Islands VIR 46.613382 46.764444 46.914637 47.057307 47.185912 47.314214 47.801059 47.834540 47.870063 47.877604 47.870702 47.852669 47.825150 47.789128 47.754932 47.875493 47.875493 47.875493 47.875493 47.875493 47.875493 47.875493 47.875493
	population, male (% of total population) sp.POP.TOTL.MA.ZS
	Population, male (% of total population) SP.POP.TOTL.MA.ZS Zimbabwe ZWE 47.214139 47.167153 47.130679 47.099796 47.076238 47.076238 47.051613 46.995893 47.049546 47.106068 47.166435 47.190963 47.231433 47.324096 47.387633 47.428426 47.00008 47.000000 47.00000 47.00000 47.00000 47.00000 47.00000 47.00000 47.00000 47.000
2]:	<pre>Index(['Series Name', 'Series Code', 'Country Name', 'Country Code', '2022',</pre>
< F C	df.info() <class 'pandas.core.frame.dataframe'=""> RangeIndex: 1085 entries, 0 to 1084 Data columns (total 26 columns): # Column Non-Null Count Dtype O Series Name 1085 non-null object 1 Series Code 1085 non-null object 2 Country Name 1085 non-null object</class>
	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
n	25 2021 105 non-null floated styles: Indicated 22, object (4) senory usage: 226.5+ KB 4 2015
9]:	8 rows × 22 columns Checking for duplicates values df.duplicated().sum() np.int64(0)
0]: 0]:	Checking for missing values df.isna().sum() Series Name 0
	2020 0 2019 0 2018 0 2017 0 2016 0 2015 0 2014 0 2013 0 2012 0 2011 0 2010 0 2011 0 2010 0 2011 0 2010 0
	2009 0 2008 0 2007 0 2006 0 2005 0 2004 0 2003 0 2001 0 dtype: int64 **
3]:	Checking unique values for columns print(df['Country Name'].unique()) print("\n Total 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' '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' '
	'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'
	'Now 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'
	'Switzerland' 'Syrian Arab Republic' 'Tajikistan' 'Tanzania' 'Thailand' 'Timor-Leste' 'Tonga' 'Trinidad and Tobago' 'Tumisia' '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 Checking country code
[print(df['Country Code'].unique()) print("\n Total 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' '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' 'CGG' 'CRI' 'CIV' 'HRV' 'CUB' 'CVW' 'CYP' 'CZE' 'DNK' 'DJI' 'DMA' 'DDM' 'ECU' 'EGY' 'SLV' 'GNQ' 'ERI' 'EST' 'SWZ' 'ETH' 'FRO' 'FJI' 'FIN' 'FRA' 'PYF' 'GAB' 'GMB' 'GEO' 'DEU' 'GHA' 'GIB' 'GRC' 'GRL' 'GRU' 'GIM' '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' 'MOA' 'MCO' 'MNG' 'MNE' 'MAR' 'MOZ' 'MMR' 'NAW' 'NRU' 'NPL' 'NLD' 'NCL' 'NZL' 'NIC' 'NER' 'NGA' 'MKD' 'MMP' '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'
6]:	Total no of unique country Code: 217 Dropping unnecessary columns df.drop(['Series Name', 'Country Name'], axis=1, inplace=True) Extraction of top 10 countries with repect to total population
0]:	# Filter data for total population 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) # Get the top ten countries with the highest total population for 2022 total_top_ten_countries = total_population_sorted.head(10) print("Top ten countries of total population\n")
1 8 4 2 9	print(total_top_ten_countries[['Country Code']]) Top ten countries of total population Country Code B9 IND 41 CHN 206 USA 90 IDN 149 PAK
2 1 1 1	Top ten countriers of total population in year 2022 and 2016
7]:	<pre># Create the bor plot plt.figure(figsize=(15, 6)) plt.subplot(2,2,1) sns.barplot(x="2022", y="Country Code", 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()</pre>
	<pre># Create the bar plot plt.figure(figsize=(15, 6)) plt.subplot (2,2,2) sns.barplot(x="2016", y="Country Code", data=total_top_ten_countries, palette="coolwarm") plt. title("Top Ten Countries of Total Population (2016)", fontsize=10) plt.xlabel("Total Population", fontsize=10) plt.ylabel("Country", fontsize=10) plt.show()</pre>
	C:\Users\sneha\AppData\Local\Temp\ipykernel_6308\1066451370.py:5: FutureWarning: Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect. sns.barplot(x="2022", y="Country Code", data=total_top_ten_countries, palette="coolwarm") top Ten Countries of Total Population (2022) IND
	DN - PAK - NGA - BRA - BBA - B
	C:\Users\sneha\AppData\Local\Temp\ipykernel_6308\1066451370.py:15: FutureWarning: Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect. sns.barplot(x="2016", y="Country Code", data=total_top_ten_countries, palette="coolwarm") Top Ten Countries of Total Population (2016) IND - CHN -
	USA - IDN - PAK - NGA - BRA - BGD - RUS - MEX -
33]:	Extraction of bottom 10 countries with resspect to total population #sort data based on the total population for 2022 total_population_sorted = total_population_data.sort_values(by="2022", ascending=True) # Get the bottom ten countries with the highest total population for 2022 total_bottom_ten_countries = total_population_sorted.head(10)
2 1 1 2	<pre>print("Bottom ten countries of total population\n") print(total_bottom_ten_countries[['Country Code']]) Bottom ten countries of total population Country Code 201 TUV 137 NRU 150 PLW 27 VGB 183 MAF</pre>
1 1 1 1	61B 164 SMR 130 MC0 114 LIE 124 MHL bottom ten countriers of total population in year 2022 and 2016 # Create the bor plot
	<pre>plt.figure(figsize=(15, 6)) plt.subplot(2,2,1) sns.barplot(x="2022", y="Country Code", data=total_bottom_ten_countries, palette="coolwarm") plt.title("Bottom Ten Countries of Total Population (2022)", fontsize=10) plt.xlabel("Total Population", fontsize=10) plt.ylabel("Country", fontsize=10) plt.show() # Create the bar plot plt.figure(figsize=(15, 6))</pre>
C	plt.subplot (2,2,2) sns.barplot(x="2016", y="Country Code", data=total_bottom_ten_countries, palette="coolwarm") plt. title("Bottom Ten Countries of Total Population (2016)", fontsize=10) plt.xlabel("Total Population", fontsize=10) plt.ylabel("Country", fontsize=10) plt.show() C:\Users\sneha\AppData\Local\Temp\ipykernel_6308\2617623145.py:4: FutureWarning: Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.
	Sins.barplot(x="2022", y="Country Code", data=total_bottom_ten_countries, palette="coolwarm") Bottom Ten Countries of Total Population (2022) TUV - NRU - PLW - VGB - GIB -
	SMR MCO LIE MHL 0 5000 10000 15000 20000 25000 30000 40000 Total Population C:\Users\sneha\AppData\Local\Temp\ipykernel_6308\2617623145.py:14: FutureWarning: Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.
	sns.barplot(x="2016", y="Country Code", data=total_bottom_ten_countries, palette="coolwarm") Bottom Ten Countries of Total Population (2016) TUV - NRU - PLW - VGB - MAF - MAF - NATURAL POPULATION (ACCOUNTS TO A COUNTRIES TO A COU
	GIB - SMR - MCO - LIE - MHL - O 10000 20000 30000 40000 50000 Total Population
3]:	<pre>#Filter data for male population male_population_data = df[df['Series Code'] =='SP.POP.TOTL.MA.IN'] #Sort data based on the male population for 2022 male_population_sorted =male_population_data.sort_values(by="2022", ascending=False) #Get the top 10 countries with the highest male population for 2022 male_top_ten_countries = male_population_sorted.head(10)</pre>
T 5 4 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	print("Top ten countries of male population") print(male_top_ten_countries[['Country Code']]) Top ten countries of male population
5	Heilter data for female population female_population_data = df[df['Series Code'] =='SP.POP.TOTL.MA.IN']
T 5	#Sort data based on the male population for 2022 female_population_sorted =female_population_data.sort_values(by="2022",ascending=False) #Get the top 10 countries with the highest female population for 2022 female_top_ten_countries = female_population_sorted.head(10) print("Top ten countries of female population") print(female_top_ten_countries[['Country Code']]) Top ten countries of female population
6 5 5 4 4 4 5 5	CHN 640 USA 6524 IDN 6583 PAK 6578 NGA 640 BRA 640 BGD 6595 RUS 661 MEX
7]:	Top ten countries with highest male and female population in 2022 # Create the bor plot plt.figure(figsize=(15, 6)) plt.subplot(2,2,1) sns.barplot(x="2022", y="Country Code", data=male_top_ten_countries, palette="viridis") plt.title("top Ten Countries of male Population (2022)", fontsize=10) plt.ylabel("Male Population", fontsize=10) plt.ylabel("Country", fontsize=10)
	<pre>plt.show() # Create the bar plot plt.figure(figsize=(15, 6)) plt.subplot (2,2,2) sns.barplot(x="2016", y="Country Code", data=female_top_ten_countries, palette="viridis") plt. title("Top Ten Countries of female Population (2016)", fontsize=10) plt.xlabel("Female Population", fontsize=10) plt.ylabel("Country", fontsize=10) plt.show()</pre>
C	C:\Users\sneha\AppData\Local\Temp\ipykernel_6308\2034849361.py:5: FutureWarning: Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect. sns.barplot(x="2022", y="Country Code", data=male_top_ten_countries, palette="viridis") top Ten Countries of male Population (2022) IND - CHN -
	USA - IDN - PAK - NGA - BRA - BGD - RUS - MEX -
	Male Population C:\Users\sneha\AppData\Local\Temp\ipykernel_6308\2034849361.py:15: FutureWarning: Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect. sns.barplot(x="2016", y="Country Code", data=female_top_ten_countries, palette="viridis") Top Ten Countries of female Population (2016) IND
	CHN - USA - IDN - PAK - BRA - BGD - RUS - MEX -
	Stacked Bar plot Top 10 Countries with Male & Female Population
9]:	<pre>#merge male and female population data on 'Country Code' merge_data=pd.merge(male_population_data, female_population_data, on="Country Code", suffixes=("_male","_female")) #merged data #calculate the total population for each country (male + female) merge_data["Total population"] = merge_data["2022_male"] + merge_data["2022_female"] merge_data.head()</pre>
2]:	merge_data.head() Series Code_male
6]:	4 SP.POP.TOTL.MA.IN AND 40786.0 40361.0 39615.0 38842.0 38071.0 37380.0 36628.0 36188.0 36307.0 37787.0 39191.0 40535.0 41829.0 41679.0 40082.0 38429.0 36773.0 3700 3700 3700 3700 3700 3700 3700
77]: 83]:	<pre>top_10_countries = sorted_data.head(10) #create the staked bar plot plt.figure(figsize=(12,6)) sns.barplot(x="Country Code",y="2022_female", data=top_10_countries, color="red", label="female Population") sns.barplot(x="Country Code",y="2022_male", data=top_10_countries, bottom=top_10_countries["2022_female"], color="green", label="Male Population") plt.xlabel("Country") plt.ylabel("Population") plt.legend()</pre>
	1.0 - 1.0 -
-	
-	0.2 - 0.0