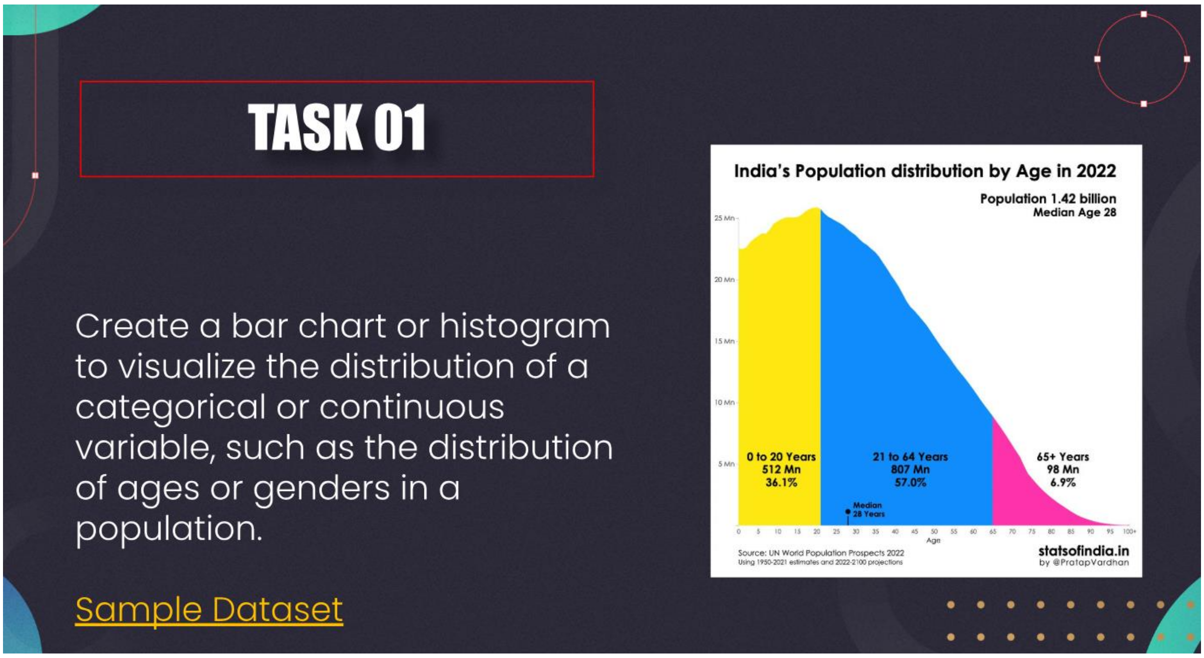
Project Name: Exploring Global Population Trends: A Data Visualization Project

By Shruti Thorat

Project Introduction

- Understanding global population trends is crucial for governments, policymakers, researchers, and businesses.
- Population data offers insights into demographic changes, economic growth, and social dynamics.
- This project aims to visualize world population data from 1960 to 2023, leveraging Python and various data visualization libraries such as Matplotlib and Seaborn.
- By analyzing historical population data, we can uncover patterns, trends, and anomalies that can inform future planning and decision-making.



Project Summary

- This project utilizes a dataset containing population data for various countries over the years 1960 to 2023.

- The dataset includes columns such as Country Name, Country Code, Indicator Name, Indicator Code, and yearly population values.

- The main objectives of the project include:

  1. Data Cleaning and Preparation

  2. Descriptive Statistics

  3. Data Visualization

  4. Analysis and Insights

Business Objective

- The primary business objective of this project is to provide actionable insights into global population trends that can inform strategic planning and decision-making.
- By understanding historical population data and identifying key patterns, stakeholders can make data-driven decisions in areas such as:

1. **Urban Planning and Development**:
    - Plan for infrastructure development and resource allocation based on population growth trends.
2. **Healthcare Services**:
    - Anticipate healthcare needs and services for regions with rapidly growing populations.
3. **Market Research and Expansion**:
    - Identify potential markets for business expansion by analyzing population demographics.
4. **Policy Formulation**:
    - Develop policies that address the needs of diverse populations and promote sustainable growth.
5. **Educational Planning**:
    - Forecast educational needs and allocate resources to regions with high population growth.

Import Libraries

```
[ ] #importing Libraries
    import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
```

Loading the Dataset

```
#Loading the Dataset
df=pd.read_csv('/content/population_Data.csv')
df
```

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	...	2014	2015	2016	2017	2018	2019	2020	2021	2022	
0	Aruba	ABW	Population, total	SP.POP.TOTL	54922	55578	56320	57002	57619	58190	...	106807.0	107906.0	108727.0	108735.0	108908.0	109203	108587.0	107700	107310.0	1
1	Africa Eastern and Southern	AFE	Population, total	SP.POP.TOTL	130072080	133534923	137171659	140945536	144904094	149033472	...	590968990.0	607123269.0	623369401.0	640058741.0	657801085.0	675950189	694446100.0	713090928	731821393.0	7505
2	Afghanistan	AFG	Population, total	SP.POP.TOTL	9035043	9214083	9404406	9604487	9814318	10036008	...	32792523.0	33831764.0	34700612.0	35688935.0	36743039.0	37856121	39068979.0	40000412	40578842.0	414
3	Africa Western and Central	AFW	Population, total	SP.POP.TOTL	97630925	99706674	101854756	104089175	106388440	108772632	...	406992047.0	418127845.0	429454743.0	440882906.0	452195915.0	463365429	474569351.0	485920997	497387180.0	5093
4	Angola	AGO	Population, total	SP.POP.TOTL	5231654	5301583	5354310	5408320	5464187	5521981	...	27160769.0	28157798.0	29183070.0	30234839.0	31297155.0	32375632	33451132.0	34532429	35635029.0	367
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
259	Kosovo	XKX	Population, total	SP.POP.TOTL	984846	1011421	1036950	1062737	1090270	1120168	...	1812788.0	1788274.0	1777568.0	1791019.0	1797086.0	1788891	1790152.0	1786080	1768096.0	17
260	Yemen, Rep.	YEM	Population, total	SP.POP.TOTL	5532301	5655232	5782221	5911135	6048006	6195593	...	30226309.0	31159379.0	32109010.0	33090921.0	34085182.0	35111408	36134863.0	37140230	38222876.0	393
261	South Africa	ZAF	Population, total	SP.POP.TOTL	16440172	16908035	17418522	17954564	18511361	19089380	...	55594838.0	56723537.0	57259551.0	57635162.0	58613001.0	59587885	60562381.0	61502603	62378410.0	632
262	Zambia	ZMB	Population, total	SP.POP.TOTL	3153729	3254086	3358099	3465907	3577017	3692086	...	15895315.0	16399089.0	16914423.0	17441320.0	17973569.0	18513839	19059395.0	19603607	20152938.0	207
263	Zimbabwe	ZWE	Population, total	SP.POP.TOTL	3809389	3930401	4055959	4185877	4320006	4458462	...	14207359.0	14399013.0	14600294.0	14812482.0	15034452.0	15271368	15526888.0	15797210	16069056.0	163

264 rows × 68 columns

```
[ ] df.head(10)
```

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	...	2014	2015	2016	2017	2018	2019	2020	2021	2022	2
0	Aruba	ABW	Population, total	SP.POP.TOTL	54922	55578	56320	57002	57619	58190	...	106807.0	107906.0	108727.0	108735.0	108908.0	109203	108587.0	107700	107310.0	107
1	Africa Eastern and Southern	AFE	Population, total	SP.POP.TOTL	130072080	133534923	137171659	140945536	144904094	149033472	...	590968990.0	607123269.0	623369401.0	640058741.0	657801085.0	675950189	694446100.0	713090928	731821393.0	750503
2	Afghanistan	AFG	Population, total	SP.POP.TOTL	9035043	9214083	9404406	9604487	9814318	10036008	...	32792523.0	33831764.0	34700612.0	35688935.0	36743039.0	37856121	39068979.0	40000412	40578842.0	41454
3	Africa Western and Central	AFW	Population, total	SP.POP.TOTL	97630925	99706674	101854756	104089175	106388440	108772632	...	406992047.0	418127845.0	429454743.0	440882906.0	452195915.0	463365429	474569351.0	485920997	497387180.0	509398
4	Angola	AGO	Population, total	SP.POP.TOTL	5231654	5301583	5354310	5408320	5464187	5521981	...	27160769.0	28157798.0	29183070.0	30234839.0	31297155.0	32375632	33451132.0	34532429	35635029.0	36749
5	Albania	ALB	Population, total	SP.POP.TOTL	1608800	1659800	1711319	1762621	1814135	1864791	...	2889104.0	2880703.0	2876101.0	2873457.0	2866376.0	2854191	2837849.0	2811666	2777689.0	2745
6	Andorra	AND	Population, total	SP.POP.TOTL	9510	10283	11086	11915	12764	13634	...	73737.0	72174.0	72181.0	73763.0	75162.0	76474	77380.0	78364	79705.0	80
7	Arab World	ARB	Population, total	SP.POP.TOTL	91540853	93931683	96428599	99038509	101729760	104494008	...	400231008.0	410190679.0	419808341.0	428315886.0	435998060.0	444281315	453723239.0	460646603	471352066.0	481667
8	United Arab Emirates	ARE	Population, total	SP.POP.TOTL	131334	137989	144946	152211	159692	167103	...	8059440.0	8505237.0	8935095.0	9223225.0	9346701.0	9445785	9401038.0	9575152	10074977.0	10483
9	Argentina	ARG	Population, total	SP.POP.TOTL	20386045	20726276	21072538	21421705	21769453	22112629	...	43024071.0	43477012.0	43900313.0	44288894.0	44654882.0	44973465	45191965.0	45312281	45407904.0	45538

10 rows × 68 columns

```
df.tail(10)
```

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	...	2014	2015	2016	2017	2018	2019	2020	2021	
254	Virgin Islands (U.S.)	VIR	Population, total	SP.POP.TOTL	32500	34300	35000	39800	40800	43500	...	1.078820e+05	1.077120e+05	1.075160e+05	1.072810e+05	1.070010e+05	106669	1.062900e+05	105870	1.05
255	Viet Nam	VNM	Population, total	SP.POP.TOTL	32531933	33409059	34288560	35249101	36201563	37129656	...	9.167958e+07	9.282325e+07	9.400012e+07	9.517698e+07	9.623732e+07	97173776	9.807919e+07	98935098	9.96
256	Vanuatu	VUT	Population, total	SP.POP.TOTL	64431	66264	68174	70159	72219	74358	...	2.602400e+05	2.661010e+05	2.720870e+05	2.785070e+05	2.852590e+05	291985	2.988580e+05	305868	3.13
257	World	WLD	Population, total	SP.POP.TOTL	3021529236	3062769479	3117373096	3184063049	3251253200	3318997522	...	7.353911e+09	7.441472e+09	7.528523e+09	7.614114e+09	7.696495e+09	7776892015	7.856139e+09	7921184346	7.98
258	Samoa	WSM	Population, total	SP.POP.TOTL	112490	115496	118597	121764	124894	127978	...	2.000370e+05	2.017820e+05	2.034990e+05	2.054150e+05	2.075820e+05	209780	2.119440e+05	213779	2.15
259	Kosovo	XKX	Population, total	SP.POP.TOTL	984846	1011421	1036950	1062737	1090270	1120168	...	1.812788e+06	1.788274e+06	1.777568e+06	1.791019e+06	1.797086e+06	1788891	1.790152e+06	1786080	1.76
260	Yemen, Rep.	YEM	Population, total	SP.POP.TOTL	5532301	5655232	5782221	5911135	6048006	6195593	...	3.022631e+07	3.115938e+07	3.210901e+07	3.309092e+07	3.408518e+07	35111408	3.613486e+07	37140230	3.82
261	South Africa	ZAF	Population, total	SP.POP.TOTL	16440172	16908035	17418522	17954564	18511361	19089380	...	5.559484e+07	5.672354e+07	5.725955e+07	5.763516e+07	5.861300e+07	59587885	6.056238e+07	61502603	6.23
262	Zambia	ZMB	Population, total	SP.POP.TOTL	3153729	3254086	3358099	3465907	3577017	3692086	...	1.589532e+07	1.639909e+07	1.691442e+07	1.744132e+07	1.797357e+07	18513839	1.905940e+07	19603607	2.01
263	Zimbabwe	ZWE	Population, total	SP.POP.TOTL	3809389	3930401	4055959	4185877	4320006	4458462	...	1.420736e+07	1.439901e+07	1.460029e+07	1.481248e+07	1.503445e+07	15271368	1.552689e+07	15797210	1.60

10 rows × 68 columns

+ Code + Text

```
[ ] df.shape
```

```
(264, 68)
```

```
print(df.info())
```

```
 13  1969            264 non-null    int64
 14  1970            264 non-null    int64
 15  1971            264 non-null    int64
 16  1972            264 non-null    int64
 17  1973            264 non-null    int64
 18  1974            264 non-null    int64
 19  1975            264 non-null    int64
 20  1976            264 non-null    int64
 21  1977            264 non-null    int64
 22  1978            264 non-null    int64
 23  1979            264 non-null    int64
 24  1980            264 non-null    int64
 25  1981            264 non-null    int64
 26  1982            264 non-null    int64
 27  1983            264 non-null    int64
 28  1984            264 non-null    int64
 29  1985            264 non-null    int64
 30  1986            264 non-null    int64
 31  1987            264 non-null    int64
 32  1988            264 non-null    int64
 33  1989            264 non-null    int64
 34  1990            264 non-null    int64
 35  1991            264 non-null    int64
 36  1992            264 non-null    float64
 37  1993            264 non-null    int64
 38  1994            264 non-null    float64
 39  1995            264 non-null    int64
 40  1996            264 non-null    float64
 41  1997            264 non-null    float64
 42  1998            264 non-null    float64
 43  1999            264 non-null    int64
 44  2000            264 non-null    float64
 45  2001            264 non-null    float64
 46  2002            264 non-null    int64
 47  2003            264 non-null    int64
 48  2004            264 non-null    float64
 49  2005            264 non-null    float64
```

```
 52  2008            264 non-null    int64
 53  2009            264 non-null    int64
 54  2010            264 non-null    int64
 55  2011            264 non-null    float64
 56  2012            264 non-null    int64
 57  2013            264 non-null    int64
 58  2014            264 non-null    float64
 59  2015            264 non-null    float64
 60  2016            264 non-null    float64
 61  2017            264 non-null    float64
 62  2018            264 non-null    float64
 63  2019            264 non-null    int64
 64  2020            264 non-null    float64
 65  2021            264 non-null    int64
 66  2022            264 non-null    float64
 67  2023            264 non-null    int64
dtypes: float64(18), int64(46), object(4)
memory usage: 140.4+ KB
None
```

```
[ ] df.describe
```

```
pandas.core.generic.NDFrame.describe
def describe(percentiles=None, include=None, exclude=None) -> Self

Generate descriptive statistics.

Descriptive statistics include those that summarize the central
tendency, dispersion and shape of a
dataset's distribution, excluding ``NaN`` values.
```

```
[ ] df.columns
```

```
Index(['Country Name', 'Country Code', 'Indicator Name', 'Indicator Code',
       '1960', '1961', '1962', '1963', '1964', '1965', '1966', '1967', '1968',
       '1969', '1970', '1971', '1972', '1973', '1974', '1975', '1976', '1977',
       '1978', '1979', '1980', '1981', '1982', '1983', '1984', '1985', '1986',
       '1987', '1988', '1989', '1990', '1991', '1992', '1993', '1994', '1995',
       '1996', '1997', '1998', '1999', '2000', '2001', '2002', '2003', '2004',
       '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013',
       '2014', '2015', '2016', '2017', '2018', '2019', '2020', '2021', '2022',
       '2023'],
      dtype='object')
```

```
[ ] #chracking any duplicate value
    print(df.duplicated().sum())
```

```
0
```

```
[ ] duplicated_value=df.duplicated().value_counts()
    print(duplicated_value)
```

```
False    264
Name: count, dtype: int64
```

```
[ ] print(df.isnull().sum())
```

```
Country Name      0
Country Code      0
Indicator Name    0
Indicator Code    0
1960              0
                 ..
2019              0
2020              0
2021              0
2022              0
2023              0
Length: 68, dtype: int64
```

```
[ ] #check Unique values for each variable
    unique_value=df.nunique()
    print(unique_value)
```

```
Country Name      264
Country Code      264
Indicator Name      1
Indicator Code      1
1960              260
                 ...
2019              262
2020              262
2021              262
2022              262
2023              262
Length: 68, dtype: int64
```

```
#let's look unique values for every columns
df.apply=(lambda col: col.unique())
```

Checking Unique value for country name column

```
[ ] print(df['Country Name'].unique())
    print('\n')
    print(f"Total no of unquie country name is {len(df['Country Name'].unique())}")
```

```
['Aruba' 'Africa Eastern and Southern' 'Afghanistan'
 'Africa Western and Central' 'Angola' 'Albania' 'Andorra' 'Arab World'
 'United Arab Emirates' 'Argentina' 'Armenia' 'American Samoa'
 'Antigua and Barbuda' 'Australia' 'Austria' 'Azerbaijan' 'Burundi'
 'Belgium' 'Benin' 'Burkina Faso' 'Bangladesh' 'Bulgaria' 'Bahrain'
 'Bahamas, The' 'Bosnia and Herzegovina' 'Belarus' 'Belize' 'Bermuda'
 'Bolivia' 'Brazil' 'Barbados' 'Brunei Darussalam' 'Bhutan' 'Botswana'
 'Central African Republic' 'Canada' 'Central Europe and the Baltics'
 'Switzerland' 'Channel Islands' 'Chile' 'China' "Cote d'Ivoire"
 'Cameroon' 'Congo, Dem. Rep.' 'Congo, Rep.' 'Colombia' 'Comoros'
 'Cabo Verde' 'Costa Rica' 'Caribbean small states' 'Cuba' 'Curacao'
 'Cayman Islands' 'Cyprus' 'Czechia' 'Germany' 'Djibouti' 'Dominica'
 'Denmark' 'Dominican Republic' 'Algeria'
 'East Asia & Pacific (excluding high income)'
 'Early-demographic dividend' 'East Asia & Pacific'
 'Europe & Central Asia (excluding high income)' 'Europe & Central Asia'
 'Ecuador' 'Egypt, Arab Rep.' 'Euro area' 'Eritrea' 'Spain' 'Estonia'
 'Ethiopia' 'European Union' 'Fragile and conflict affected situations'
 'Finland' 'Fiji' 'France' 'Faroe Islands' 'Micronesia, Fed. Sts.' 'Gabon'
 'United Kingdom' 'Georgia' 'Ghana' 'Gibraltar' 'Guinea' 'Gambia, The'
 'Guinea-Bissau' 'Equatorial Guinea' 'Greece' 'Grenada' 'Greenland'
 'Guatemala' 'Guam' 'Guyana' 'High income' 'Hong Kong SAR, China'
 'Honduras' 'Heavily indebted poor countries (HIPC)' 'Croatia' 'Haiti'
 'Hungary' 'IBRD only' 'IDA & IBRD total' 'IDA total' 'IDA blend'
 'Indonesia' 'IDA only' 'Isle of Man' 'India' 'Ireland'
 'Iran, Islamic Rep.' 'Iraq' 'Iceland' 'Israel' 'Italy' 'Jamaica' 'Jordan'
 'Japan' 'Kazakhstan' 'Kenya' 'Kyrgyz Republic' 'Cambodia' 'Kiribati'
 'St. Kitts and Nevis' 'Korea, Rep.' 'Kuwait'
 'Latin America & Caribbean (excluding high income)' 'Lao PDR' 'Lebanon'
 'Liberia' 'Libya' 'St. Lucia' 'Latin America & Caribbean'
 'Least developed countries: UN classification' 'Low income'
 'Liechtenstein' 'Sri Lanka' 'Lower middle income' 'Low & middle income'
 'Lesotho' 'Late-demographic dividend' 'Lithuania' 'Luxembourg' 'Latvia'
 'Macao SAR, China' 'St. Martin (French part)' 'Morocco' 'Monaco'
 'Moldova' 'Madagascar' 'Maldives' 'Middle East & North Africa' 'Mexico'
 'Marshall Islands' 'Middle income' 'North Macedonia' 'Mali' 'Malta'
 'Myanmar' 'Middle East & North Africa (excluding high income)'
 'Montenegro' 'Mongolia' 'Northern Mariana Islands' 'Mozambique'
 'Mauritania' 'Mauritius' 'Malawi' 'Malaysia' 'North America' 'Namibia'
```

```
 'Middle East & North Africa (IDA & IBRD countries)' 'Tonga'
 'South Asia (IDA & IBRD)' 'Sub-Saharan Africa (IDA & IBRD countries)'
 'Trinidad and Tobago' 'Tunisia' 'Turkiye' 'Tuvalu' 'Tanzania' 'Uganda'
 'Ukraine' 'Upper middle income' 'Uruguay' 'United States' 'Uzbekistan'
 'St. Vincent and the Grenadines' 'Venezuela, RB' 'British Virgin Islands'
 'Virgin Islands (U.S.)' 'Viet Nam' 'Vanuatu' 'World' 'Samoa' 'Kosovo'
 'Yemen, Rep.' 'South Africa' 'Zambia' 'Zimbabwe']


Total no of unquie country name is 264
```

```
print(df['Country Code'].unique())
print('\n')
print(f"Total no of unquie country name is {len(df['Country Code'].unique())}")
```

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


Total no of unquie country name is 264
```

```
[ ] #drop indicator name column
    df=df.drop(['Country Code','Indicator Name','Indicator Code'],axis=1)
    df
```

Country Name	1960	1961	1962	1963	1964	1965	1966	1967	1968	...	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023
Aruba	54922	55578	56320	57002	57619	58190	58694	58990	59069	...	106807.0	107906.0	108727.0	108735.0	108908.0	109203	108587.0	107700	107310.0	107359
Africa Eastern and Southern	130072080	133534923	137171659	140945536	144904094	149033472	153281203	157704381	162329396	...	590968990.0	607123269.0	623369401.0	640058741.0	657801085.0	675950189	694446100.0	713090928	731821393.0	750503764
Afghanistan	9035043	9214083	9404406	9604487	9814318	10036008	10266395	10505959	10756922	...	32792523.0	33831764.0	34700612.0	35688935.0	36743039.0	37856121	39068979.0	40000412	40578842.0	41454761
Africa Western and Central	97630925	99706674	101854756	104089175	106388440	108772632	111246953	113795019	116444636	...	406992047.0	418127845.0	429454743.0	440882906.0	452195915.0	463365429	474569351.0	485920997	497387180.0	509398589
Angola	5231654	5301583	5354310	5408320	5464187	5521981	5581386	5641807	5702699	...	27160769.0	28157798.0	29183070.0	30234839.0	31297155.0	32375632	33451132.0	34532429	35635029.0	36749906
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Kosovo	984846	1011421	1036950	1062737	1090270	1120168	1152586	1187667	1214208	...	1812788.0	1788274.0	1777568.0	1791019.0	1797086.0	1788891	1790152.0	1786080	1768096.0	1756366
Yemen, Rep.	5532301	5655232	5782221	5911135	6048006	6195593	6351494	6516444	6690524	...	30226309.0	31159379.0	32109010.0	33090921.0	34085182.0	35111408	36134863.0	37140230	38222876.0	39390799
South Africa	16440172	16908035	17418522	17954564	18511361	19089380	19690087	20314066	20957287	...	55594838.0	56723537.0	57259551.0	57635162.0	58613001.0	59587885	60562381.0	61502603	62378410.0	63212384
Zambia	3153729	3254086	3358099	3465907	3577017	3692086	3812003	3936343	4065593	...	15895315.0	16399089.0	16914423.0	17441320.0	17973569.0	18513839	19059395.0	19603607	20152938.0	20723965
Zimbabwe	3809389	3930401	4055959	4185877	4320006	4458462	4601217	4748307	4900440	...	14207359.0	14399013.0	14600294.0	14812482.0	15034452.0	15271368	15526888.0	15797210	16069056.0	16340822

ws × 65 columns

```
[ ] df.shape
```

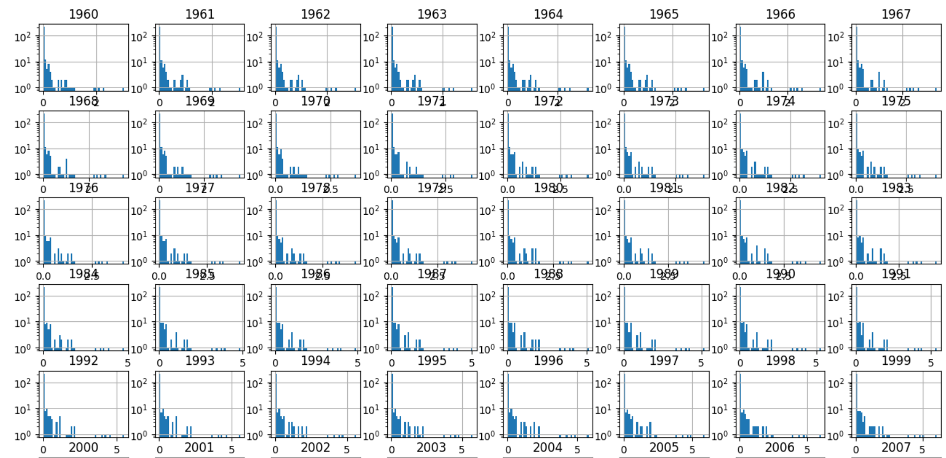
```
(264, 65)
```

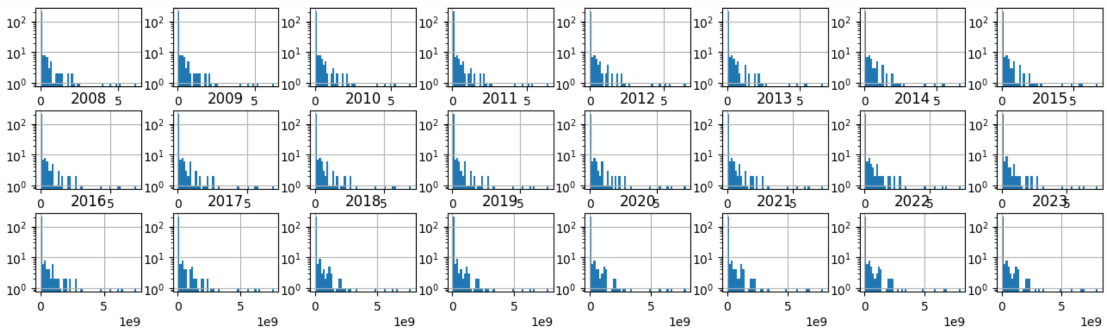
Exploratory Data Analysis (EDA)

Check Data Distribution

- by Histograms: To Visualize Skewness of Each Column

```
#check data distribution
df.hist(figsize=(16,12),bins=50,log=True)
plt.suptitle('Data distribution',fontsize=13)
plt.show()
```





skewness calculation only on the numeric columns, avoiding the inclusion of non-numeric columns.

```
#select only numeric column
numeric_column=df.select_dtypes(include=np.number)
numeric_column

#calculate skewness for numeric column
skewness=numeric_column.skew()
skewness
```

	0
1960	4.890496
1961	4.897392
1962	4.901252
1963	4.902471
1964	4.903891
...	...
2019	4.966489
2020	4.965437
2021	4.963988
2022	4.962397
2023	4.960944

64 rows × 1 columns

dtype: float64

```
total_population_data = df[df['Country Name'] == 'Aruba']
print(total_population_data)
```

```
  Country Name   1960   1961   1962   1963   1964   1965   1966   1967   1968  \
0        Aruba  54922  55578  56320  57002  57619  58190  58694  58990  59069

   ...      2014      2015      2016      2017      2018    2019      2020  \
0  ...  106807.0  107906.0  108727.0  108735.0  108908.0  109203  108587.0

     2021      2022    2023
0  107700  107310.0  107359

[1 rows x 65 columns]
```

```
total_population_data_sorted=total_population_data.sort_values(by='2023',ascending=False)
total_population_data_sorted
```

	Country Name	1960	1961	1962	1963	1964	1965	1966	1967	1968	...	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023
0	Aruba	54922	55578	56320	57002	57619	58190	58694	58990	59069	...	106807.0	107906.0	108727.0	108735.0	108908.0	109203	108587.0	107700	107310.0	107359

1 rows × 65 columns

```
top_10_countries=total_population_data_sorted.head(10)
top_10_countries
```

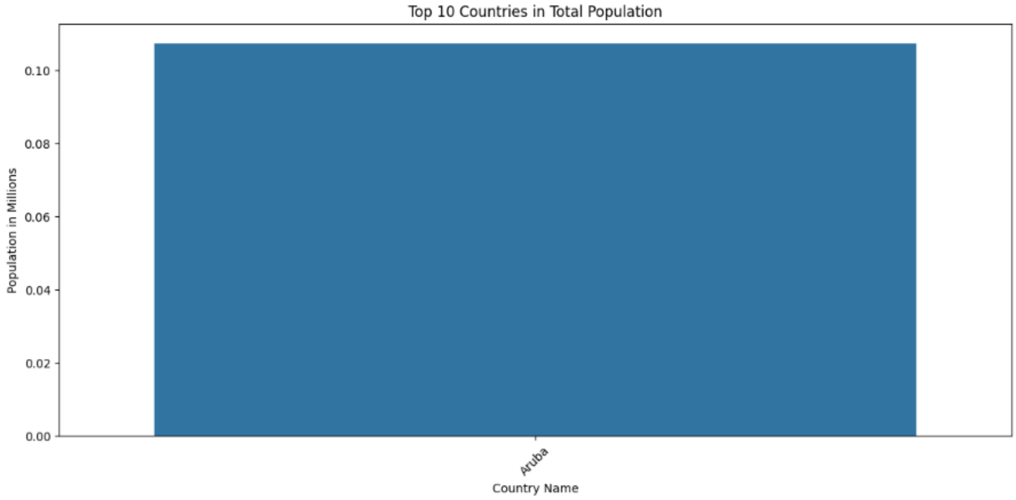
	Country Name	1960	1961	1962	1963	1964	1965	1966	1967	1968	...	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023
0	Aruba	54922	55578	56320	57002	57619	58190	58694	58990	59069	...	106807.0	107906.0	108727.0	108735.0	108908.0	109203	108587.0	107700	107310.0	107359

1 rows × 65 columns

```
print('top_10_countries in total population')
print(top_10_countries[['Country Name','2023']])
```

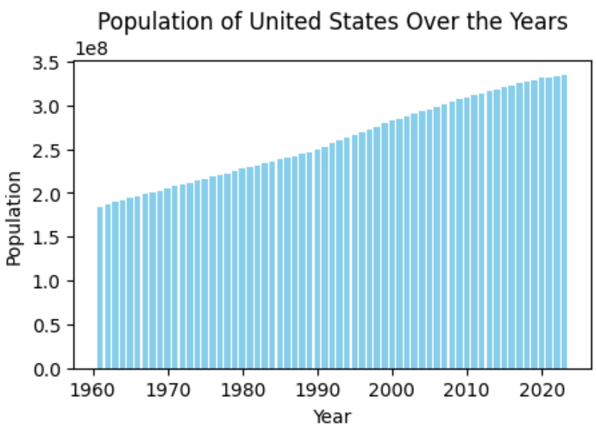
```
top_10_countries in total population
  Country Name    2023
0        Aruba  107359
```

```python
import matplotlib.pyplot as plt
import seaborn as sns  # Import seaborn
import pandas as pd

plt.figure(figsize=(12, 6))

sns.barplot(x=top_10_countries['Country Name'], y=top_10_countries['2023'] / 1e6)

plt.title('Top 10 Countries in Total Population')
plt.xlabel('Country Name')
plt.ylabel('Population in Millions')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.tight_layout()
plt.show()
```

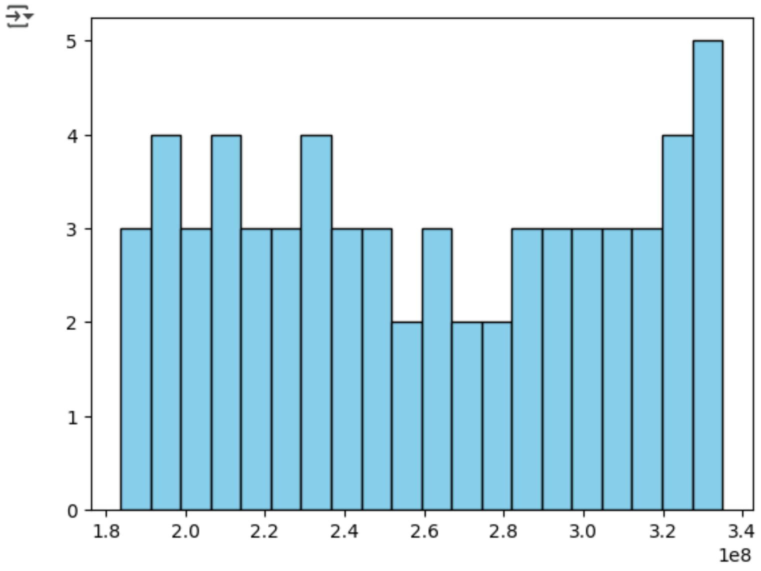



```python
df_final = df.drop(['Country Code', 'Indicator Name', 'Indicator Code'], axis=1)
```

```python
chosen_country = 'United States'


country_data = df_final[df_final['Country Name'] == chosen_country]


years = country_data.columns[2:].astype(int)
population = country_data.iloc[:, 2:].values.flatten()

plt.figure(figsize=(5, 3))
plt.bar(years, population, color='skyblue')
plt.xlabel('Year')
plt.ylabel('Population')
plt.title(f'Population of {chosen_country} Over the Years')


plt.show()
```



```
[ ] plt.hist(population, bins=20, color='skyblue', edgecolor='black')
    plt.show()
```



Conclusion

The Global Population Trends Data Visualization project has provided a comprehensive analysis of population data from 1960 to 2023, offering valuable insights into demographic changes across different countries. Through data cleaning, descriptive statistics, and a variety of visualizations, we have uncovered key patterns and trends in the global population landscape.

Key takeaways from this project include:

1. **Identification of Demographic Patterns**:
    - We identified the most and least populated countries and observed significant growth trends in specific regions.
    - The data revealed how population sizes have evolved over the decades, highlighting periods of rapid growth and stability.
2. **Insightful Visualizations**:
    - Bar charts and line plots effectively showcased the population sizes and trends for top countries.
    - Histograms provided a clear view of population distributions, emphasizing shifts and changes over time.
    - Grouped and stacked bar charts offered a comparative perspective, making it easier to analyze and understand complex data.
3. **Business and Policy Implications**:
    - The insights gained can assist urban planners, healthcare providers, market researchers, and policymakers in making informed decisions.
    - By understanding population dynamics, stakeholders can develop strategies that address the needs of growing populations and promote sustainable development.
4. **Data-Driven Decision Making**:
    - The project emphasized the importance of using data-driven approaches to analyze and interpret population data.
    - Visualizations not only made the data more accessible but also highlighted trends that might have been overlooked in raw data.