

## Prodigy Infotech Internship Task 1

Task 1 : Plot Barchart or Histogram to visualize continuous variable.

### Problem Statement

## Task-01

“

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.

### About the Dataset

In this project the dataset we are using is World Development Indicators Dataset 2001 - 2022. The given Data Frame contains information about various indicators for different countries. It has 266 rows and 26 columns.

### Importing Libraries

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

### Loading the dataset

```
df = pd.read_csv("World Development Indicator(2001-2022).csv")
df
```

	Country Name	Country Code	Indicator Name	\
0	Aruba	ABW	Population, total	
1	Africa Eastern and Southern	AFE	Population, total	
2	Afghanistan	AFG	Population, total	
3	Africa Western and Central	AFW	Population, total	

4	Angola	AGO	Population, total
..	...	...	...
261	Kosovo	XKX	Population, total
262	Yemen, Rep.	YEM	Population, total
263	South Africa	ZAF	Population, total
264	Zambia	ZMB	Population, total
265	Zimbabwe	ZWE	Population, total

	Indicator Code	2001	2002	2003	2004 \
0	SP.POP.TOTL	90691.0	91781.0	92701.0	93540.0
1	SP.POP.TOTL	412001885.0	422741118.0	433807484.0	445281555.0
2	SP.POP.TOTL	19688632.0	21000256.0	22645130.0	23553551.0
3	SP.POP.TOTL	277160097.0	284952322.0	292977949.0	301265247.0
4	SP.POP.TOTL	16941587.0	17516139.0	18124342.0	18771125.0
..	...	...	...	...	...
261	SP.POP.TOTL	1701154.0	1702310.0	1703466.0	1704622.0
262	SP.POP.TOTL	19143457.0	19660653.0	20188799.0	20733406.0
263	SP.POP.TOTL	47229714.0	47661514.0	48104048.0	48556071.0
264	SP.POP.TOTL	10191964.0	10508294.0	10837973.0	11188040.0
265	SP.POP.TOTL	11910978.0	11984644.0	12075828.0	12160881.0

	2005	2006	...	2013	2014	2015 \
0	94483.0	95606.0	...	102880.0	103594.0	104257.0
1	457153837.0	469508516.0	...	567892149.0	583651101.0	600008424.0
2	24411191.0	25442944.0	...	31541209.0	32716210.0	33753499.0
3	309824829.0	318601484.0	...	387204553.0	397855507.0	408690375.0
4	19450959.0	20162340.0	...	26147002.0	27128337.0	28127721.0
..	...	...	...	...	...	...
261	1705780.0	1719536.0	...	1818117.0	1812771.0	1788196.0
262	21320671.0	21966298.0	...	26984002.0	27753304.0	28516545.0
263	49017147.0	49491756.0	...	53873616.0	54729551.0	55876504.0
264	11564870.0	11971567.0	...	15234976.0	15737793.0	16248230.0
265	12224753.0	12330490.0	...	13555422.0	13855753.0	14154937.0

	2016	2017	2018	2019	2020 \
0	104874.0	105439.0	105962.0	106442.0	106585.0
1	616377605.0	632746570.0	649757148.0	667242986.0	685112979.0
2	34636207.0	35643418.0	36686784.0	37769499.0	38972230.0
3	419778384.0	431138704.0	442646825.0	454306063.0	466189102.0
4	29154746.0	30208628.0	31273533.0	32353588.0	33428486.0
..	...	...	...	...	...
261	1777557.0	1791003.0	1797085.0	1788878.0	1790133.0
262	29274002.0	30034389.0	30790513.0	31546691.0	32284046.0
263	56422274.0	56641209.0	57339635.0	58087055.0	58801927.0
264	16767761.0	17298054.0	17835893.0	18380477.0	18927715.0
265	14452704.0	14751101.0	15052184.0	15354608.0	15669666.0

	2021	2022
0	106537.0	106445.0

```

1    702977106.0  720839314.0
2     40099462.0  41128771.0
3    478185907.0  490330870.0
4     34503774.0  35588987.0
..      ...      ...
261   1786038.0   1761985.0
262   32981641.0  33696614.0
263   59392255.0  59893885.0
264   19473125.0  20017675.0
265   15993524.0  16320537.0

```

[266 rows x 26 columns]

*# showing first 5 rows*

df.head()

	Country Name	Country Code	Indicator Name	Indicator Code
\				
0	Aruba	ABW	Population, total	SP.POP.TOTL
1	Africa Eastern and Southern	AFE	Population, total	SP.POP.TOTL
2	Afghanistan	AFG	Population, total	SP.POP.TOTL
3	Africa Western and Central	AFW	Population, total	SP.POP.TOTL
4	Angola	AGO	Population, total	SP.POP.TOTL

	2001	2002	2003	2004	2005	\
0	90691.0	91781.0	92701.0	93540.0	94483.0	
1	412001885.0	422741118.0	433807484.0	445281555.0	457153837.0	
2	19688632.0	21000256.0	22645130.0	23553551.0	24411191.0	
3	277160097.0	284952322.0	292977949.0	301265247.0	309824829.0	
4	16941587.0	17516139.0	18124342.0	18771125.0	19450959.0	

	2006	...	2013	2014	2015	2016	\
0	95606.0	...	102880.0	103594.0	104257.0	104874.0	
1	469508516.0	...	567892149.0	583651101.0	600008424.0	616377605.0	
2	25442944.0	...	31541209.0	32716210.0	33753499.0	34636207.0	
3	318601484.0	...	387204553.0	397855507.0	408690375.0	419778384.0	
4	20162340.0	...	26147002.0	27128337.0	28127721.0	29154746.0	

	2017	2018	2019	2020	2021	\
0	105439.0	105962.0	106442.0	106585.0	106537.0	
1	632746570.0	649757148.0	667242986.0	685112979.0	702977106.0	
2	35643418.0	36686784.0	37769499.0	38972230.0	40099462.0	
3	431138704.0	442646825.0	454306063.0	466189102.0	478185907.0	
4	30208628.0	31273533.0	32353588.0	33428486.0	34503774.0	

	2022
0	106445.0
1	720839314.0
2	41128771.0
3	490330870.0

```
4 35588987.0
```

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

```
# showing last 5 rows
```

```
df.tail()
```

	Country Name	Country Code	Indicator Name	Indicator Code	2001
\					
261	Kosovo	XKX	Population, total	SP.POP.TOTL	1701154.0
262	Yemen, Rep.	YEM	Population, total	SP.POP.TOTL	19143457.0
263	South Africa	ZAF	Population, total	SP.POP.TOTL	47229714.0
264	Zambia	ZMB	Population, total	SP.POP.TOTL	10191964.0
265	Zimbabwe	ZWE	Population, total	SP.POP.TOTL	11910978.0

	2002	2003	2004	2005	2006	...	\
261	1702310.0	1703466.0	1704622.0	1705780.0	1719536.0	...	
262	19660653.0	20188799.0	20733406.0	21320671.0	21966298.0	...	
263	47661514.0	48104048.0	48556071.0	49017147.0	49491756.0	...	
264	10508294.0	10837973.0	11188040.0	11564870.0	11971567.0	...	
265	11984644.0	12075828.0	12160881.0	12224753.0	12330490.0	...	

	2013	2014	2015	2016	2017	2018
\						
261	1818117.0	1812771.0	1788196.0	1777557.0	1791003.0	1797085.0
262	26984002.0	27753304.0	28516545.0	29274002.0	30034389.0	30790513.0
263	53873616.0	54729551.0	55876504.0	56422274.0	56641209.0	57339635.0
264	15234976.0	15737793.0	16248230.0	16767761.0	17298054.0	17835893.0
265	13555422.0	13855753.0	14154937.0	14452704.0	14751101.0	15052184.0

	2019	2020	2021	2022
261	1788878.0	1790133.0	1786038.0	1761985.0
262	31546691.0	32284046.0	32981641.0	33696614.0
263	58087055.0	58801927.0	59392255.0	59893885.0
264	18380477.0	18927715.0	19473125.0	20017675.0
265	15354608.0	15669666.0	15993524.0	16320537.0

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

## Basic Understanding of the dataset

```
# showing dimensions of the data
```

```
df.shape
```

```
(266, 26)
```

```
# showing columns
```

```
df.columns
```

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

## About the Columns

1. Country Name: The name of the country.
1. Country Code: The code representing the country.
2. Indicator Name: The name of the indicator or series.
3. Indicator Code: The code representing the indicator or series.
4. Years [2001 - 2022] - Indicators of various countries in different years.

```
# Checking for data types
df.dtypes
```

```
Country Name      object
Country Code      object
Indicator Name     object
Indicator Code     object
2001              float64
2002              float64
2003              float64
2004              float64
2005              float64
2006              float64
2007              float64
2008              float64
2009              float64
2010              float64
2011              float64
2012              float64
2013              float64
2014              float64
2015              float64
2016              float64
2017              float64
2018              float64
2019              float64
2020              float64
2021              float64
2022              float64
dtype: object
```

```
# showing information about the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 266 entries, 0 to 265
```

```
Data columns (total 26 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	2001	265 non-null	float64
5	2002	265 non-null	float64
6	2003	265 non-null	float64
7	2004	265 non-null	float64
8	2005	265 non-null	float64
9	2006	265 non-null	float64
10	2007	265 non-null	float64
11	2008	265 non-null	float64
12	2009	265 non-null	float64
13	2010	265 non-null	float64
14	2011	265 non-null	float64
15	2012	265 non-null	float64
16	2013	265 non-null	float64
17	2014	265 non-null	float64
18	2015	265 non-null	float64
19	2016	265 non-null	float64
20	2017	265 non-null	float64
21	2018	265 non-null	float64
22	2019	265 non-null	float64
23	2020	265 non-null	float64
24	2021	265 non-null	float64
25	2022	265 non-null	float64

```
dtypes: float64(22), object(4)
```

```
memory usage: 54.2+ KB
```

```
# Descriptive Statistics Analysis
```

```
df.describe()
```

	2001	2002	2003	2004	2005
\					
count	2.650000e+02	2.650000e+02	2.650000e+02	2.650000e+02	2.650000e+02
mean	2.492924e+08	2.527984e+08	2.563013e+08	2.598104e+08	2.633363e+08
std	7.901099e+08	8.006981e+08	8.111965e+08	8.216943e+08	8.321848e+08
min	9.621000e+03	9.609000e+03	9.668000e+03	9.791000e+03	9.912000e+03
25%	1.306590e+06	1.341696e+06	1.353548e+06	1.361172e+06	1.369075e+06
50%	8.223941e+06	8.372306e+06	8.567992e+06	8.791853e+06	9.026299e+06
75%	5.010666e+07	5.166207e+07	5.320564e+07	5.481561e+07	5.655025e+07
max	6.226348e+09	6.308141e+09	6.389462e+09	6.470924e+09	6.552700e+09
	2006	2007	2008	2009	2010
\					

count	2.650000e+02	2.650000e+02	2.650000e+02	2.650000e+02	2.650000e+02
mean	2.668780e+08	2.704273e+08	2.740244e+08	2.776644e+08	2.813373e+08
std	8.426177e+08	8.530048e+08	8.635042e+08	8.741322e+08	8.848777e+08
min	1.003000e+04	1.014900e+04	1.024300e+04	1.023300e+04	1.024100e+04
25%	1.376919e+06	1.384861e+06	1.444277e+06	1.527196e+06	1.567220e+06
50%	9.080505e+06	9.148092e+06	9.219637e+06	9.298515e+06	9.483836e+06
75%	5.814398e+07	5.843831e+07	5.882673e+07	5.909536e+07	5.927742e+07
max	6.635110e+09	6.717878e+09	6.801710e+09	6.885850e+09	6.970039e+09

	...	2013	2014	2015	2016 \
count	...	2.650000e+02	2.650000e+02	2.650000e+02	2.650000e+02
mean	...	2.927775e+08	2.966763e+08	3.005450e+08	3.044039e+08
std	...	9.186842e+08	9.301439e+08	9.414551e+08	9.526713e+08
min	...	1.069400e+04	1.089900e+04	1.087700e+04	1.085200e+04
25%	...	1.697753e+06	1.743309e+06	1.788196e+06	1.777557e+06
50%	...	1.014958e+07	1.028212e+07	1.035808e+07	1.032545e+07
75%	...	6.023395e+07	6.078914e+07	6.073058e+07	6.062750e+07
max	...	7.229732e+09	7.317970e+09	7.405278e+09	7.492157e+09

		2017	2018	2019	2020	2021
\						
count	2.650000e+02	2.650000e+02	2.650000e+02	2.650000e+02	2.650000e+02	2.650000e+02
mean	3.082563e+08	3.120264e+08	3.157098e+08	3.192923e+08	3.225167e+08	3.225167e+08
std	9.638565e+08	9.746872e+08	9.851681e+08	9.952286e+08	1.004210e+09	1.004210e+09
min	1.082800e+04	1.086500e+04	1.095600e+04	1.106900e+04	1.120400e+04	1.120400e+04
25%	1.791003e+06	1.797085e+06	1.788878e+06	1.790133e+06	1.786038e+06	1.786038e+06
50%	1.030030e+07	1.039533e+07	1.044767e+07	1.060623e+07	1.050577e+07	1.050577e+07
75%	6.053671e+07	6.042176e+07	5.987258e+07	6.170452e+07	6.358833e+07	6.358833e+07
max	7.578221e+09	7.661777e+09	7.742682e+09	7.820964e+09	7.888161e+09	7.888161e+09

	2022
count	2.650000e+02
mean	3.254826e+08
std	1.012173e+09
min	1.131200e+04
25%	1.761985e+06
50%	1.052607e+07
75%	6.549775e+07
max	7.951150e+09

[8 rows x 22 columns]

## Data preprocessing and Data Cleaning

*# checking for duplicated values*

df.duplicated().sum()

0

```
# checking for null values
```

```
df.isna().sum().any()
```

```
True
```

```
df = df.fillna(method = "ffill")
```

```
df.head()
```

	Country Name	Country Code	Indicator Name	Indicator Code
\				
0	Aruba	ABW	Population, total	SP.POP.TOTL
1	Africa Eastern and Southern	AFE	Population, total	SP.POP.TOTL
2	Afghanistan	AFG	Population, total	SP.POP.TOTL
3	Africa Western and Central	AFW	Population, total	SP.POP.TOTL
4	Angola	AGO	Population, total	SP.POP.TOTL

	2001	2002	2003	2004	2005	\
0	90691.0	91781.0	92701.0	93540.0	94483.0	
1	412001885.0	422741118.0	433807484.0	445281555.0	457153837.0	
2	19688632.0	21000256.0	22645130.0	23553551.0	24411191.0	
3	277160097.0	284952322.0	292977949.0	301265247.0	309824829.0	
4	16941587.0	17516139.0	18124342.0	18771125.0	19450959.0	

	2006	...	2013	2014	2015	2016	\
0	95606.0	...	102880.0	103594.0	104257.0	104874.0	
1	469508516.0	...	567892149.0	583651101.0	600008424.0	616377605.0	
2	25442944.0	...	31541209.0	32716210.0	33753499.0	34636207.0	
3	318601484.0	...	387204553.0	397855507.0	408690375.0	419778384.0	
4	20162340.0	...	26147002.0	27128337.0	28127721.0	29154746.0	

	2017	2018	2019	2020	2021	\
0	105439.0	105962.0	106442.0	106585.0	106537.0	
1	632746570.0	649757148.0	667242986.0	685112979.0	702977106.0	
2	35643418.0	36686784.0	37769499.0	38972230.0	40099462.0	
3	431138704.0	442646825.0	454306063.0	466189102.0	478185907.0	
4	30208628.0	31273533.0	32353588.0	33428486.0	34503774.0	

	2022
0	106445.0
1	720839314.0
2	41128771.0
3	490330870.0
4	35588987.0

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

```
df.isna().sum().any()
```

```
False
```



## Checking for Unique values of the columns

```
df['Country Name'].unique()
```

```
array(['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',  
      'Not classified', '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',
```

```

'New Caledonia', 'Niger', 'Nigeria', 'Nicaragua', 'Netherlands',
'Norway', 'Nepal', 'Nauru', 'New Zealand', 'OECD members', 'Oman',
'Other small states', 'Pakistan', 'Panama', 'Peru', 'Philippines',
'Palau', 'Papua New Guinea', 'Poland', 'Pre-demographic dividend',
'Puerto Rico', 'Korea, Dem. People's Rep.', 'Portugal', 'Paraguay',
'West Bank and Gaza', 'Pacific island small states',
'Post-demographic dividend', 'French Polynesia', 'Qatar',
'Romania', 'Russian Federation', 'Rwanda', 'South Asia',
'Saudi Arabia', 'Sudan', 'Senegal', 'Singapore', 'Solomon Islands',
'Sierra Leone', 'El Salvador', 'San Marino', 'Somalia', 'Serbia',
'Sub-Saharan Africa (excluding high income)', 'South Sudan',
'Sub-Saharan Africa', 'Small states', 'Sao Tome and Principe',
'Suriname', 'Slovak Republic', 'Slovenia', 'Sweden', 'Eswatini',
'Sint Maarten (Dutch part)', 'Seychelles', 'Syrian Arab Republic',
'Turks and Caicos Islands', 'Chad',
'East Asia & Pacific (IDA & IBRD countries)',
'Europe & Central Asia (IDA & IBRD countries)', 'Togo', 'Thailand',
'Tajikistan', 'Turkmenistan',
'Latin America & the Caribbean (IDA & IBRD countries)',
'Timor-Leste', '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.)', 'Vietnam',
'Vanuatu', 'World', 'Samoa', 'Kosovo', 'Yemen, Rep.',
'South Africa', 'Zambia', 'Zimbabwe'], dtype=object)

```

```
df['Country Code'].unique()
```

```

array(['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', 'INX', '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', 'PSE', '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'], dtype=object)
```

```
df['Indicator Name'].unique()
array(['Population, total'], dtype=object)

df['Indicator Code'].unique()
array(['SP.POP.TOTL'], dtype=object)
```

As we can see Indicator Name and Indicator Code Columns has only single value, So we will drop these 2 columns

```
# Dropping Unnecessary Columns
df.drop(['Indicator Name', 'Indicator Code', 'Country Code'], axis = 1, inplace
= True)
```

Hence, Columns dropped

```
# checking for columns
df.columns
```

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

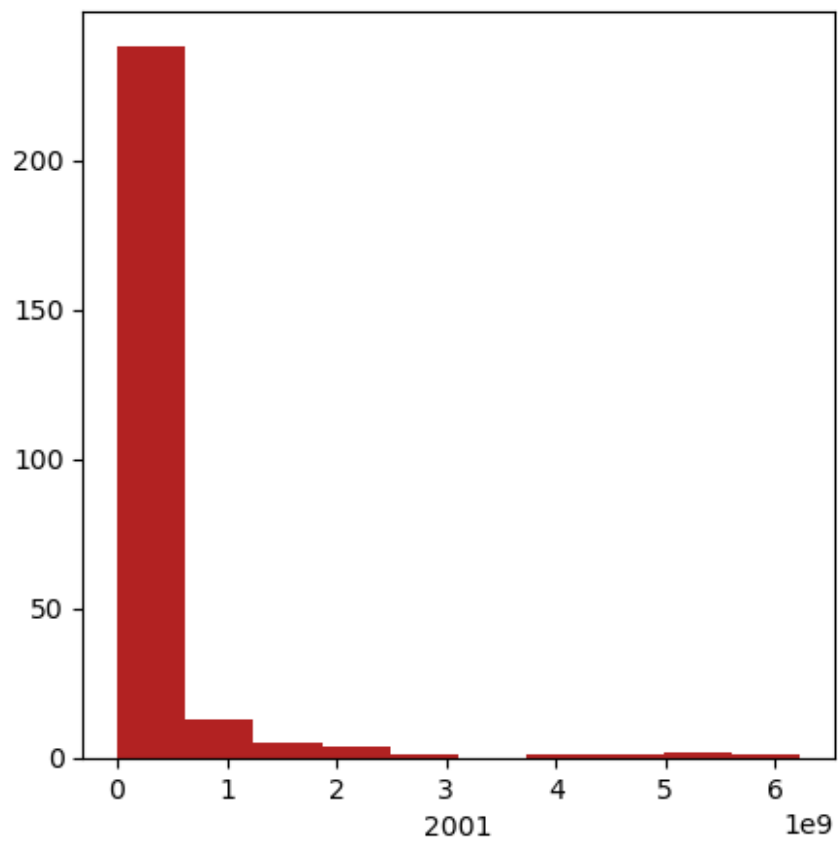
## Data Visualization

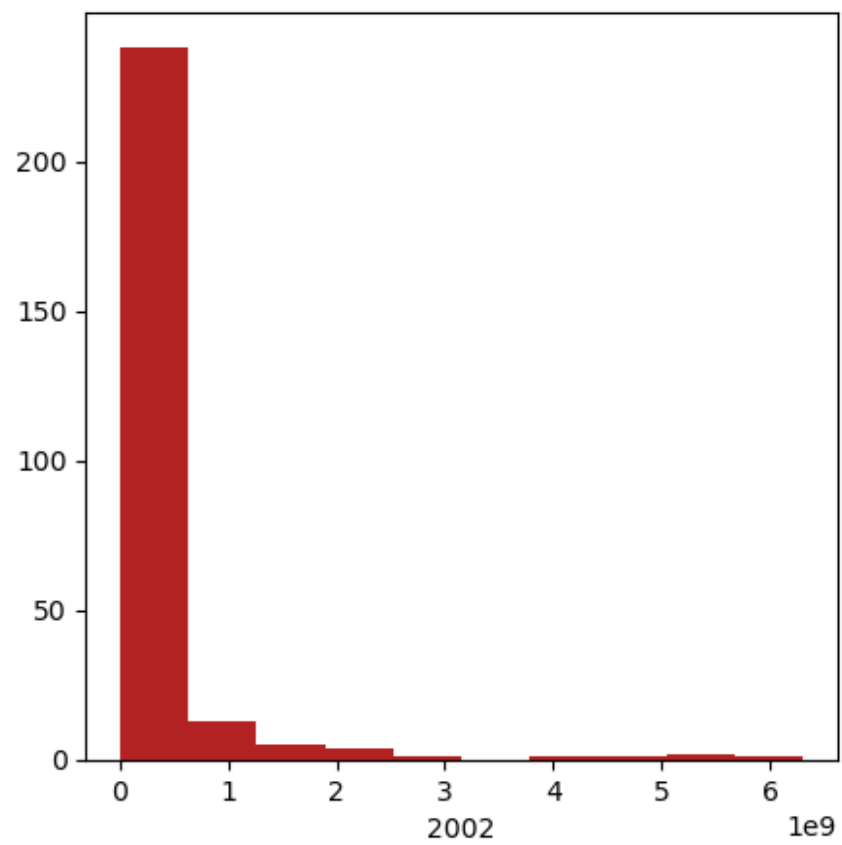
### Plotting Histplot from 2001 to 2022

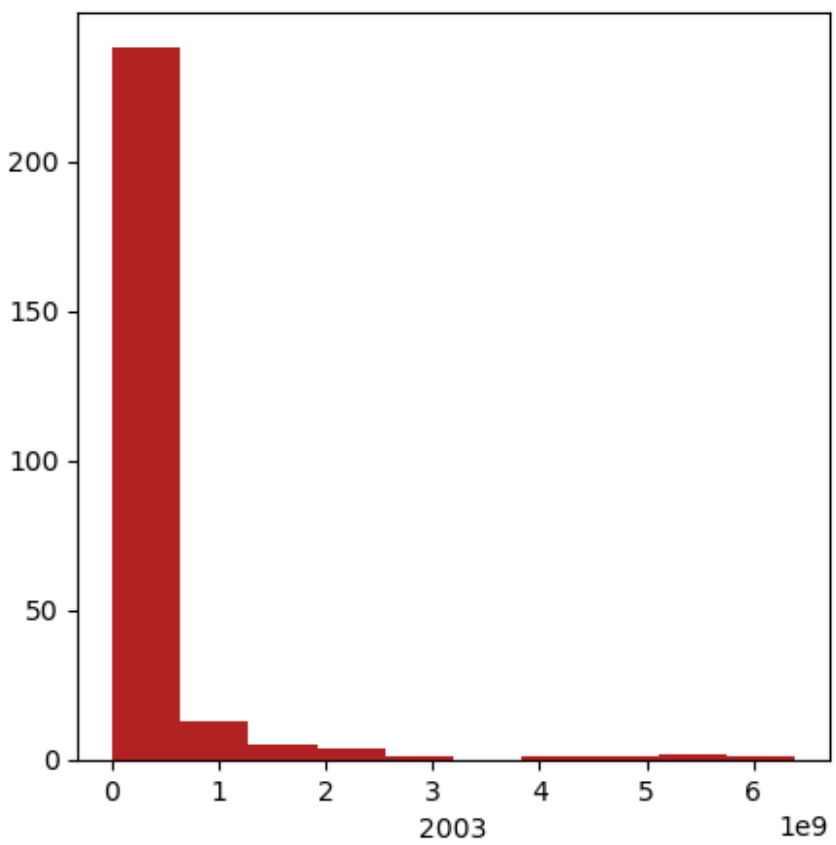
```
cols = ['2001', '2002', '2003', '2004', '2005', '2006', '2007',
        '2008', '2009', '2010', '2011', '2012', '2013', '2014', '2015',
'2016',
        '2017', '2018', '2019', '2020', '2021', '2022']

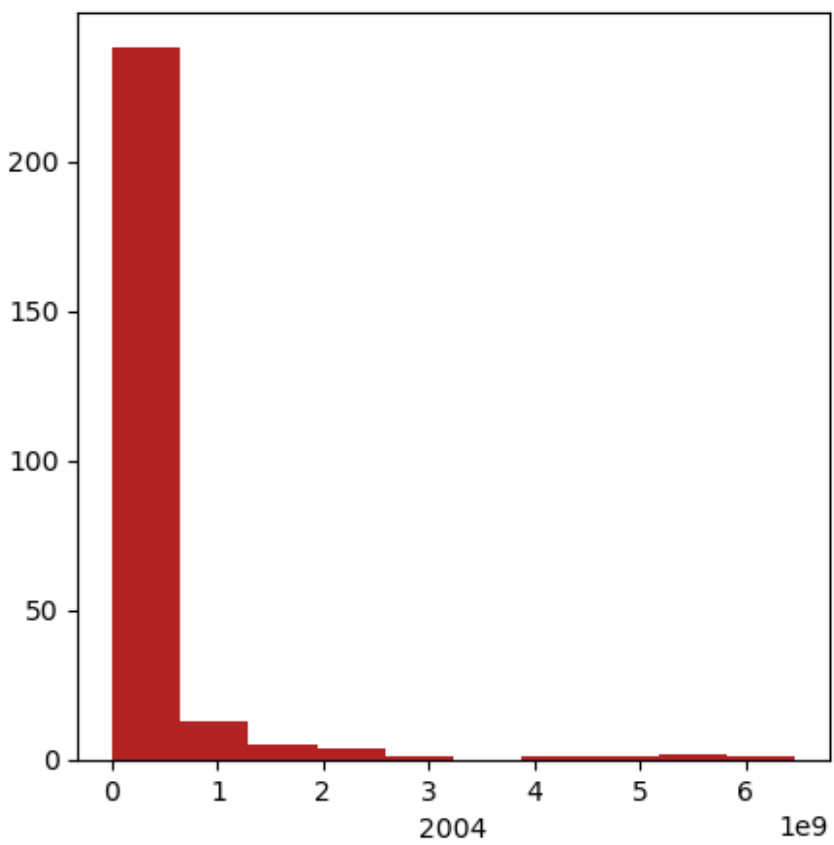
for i in cols:
    fig = plt.figure(figsize=(5,5))
    plt.hist(df[i], color='#B22222', bins=10)
```

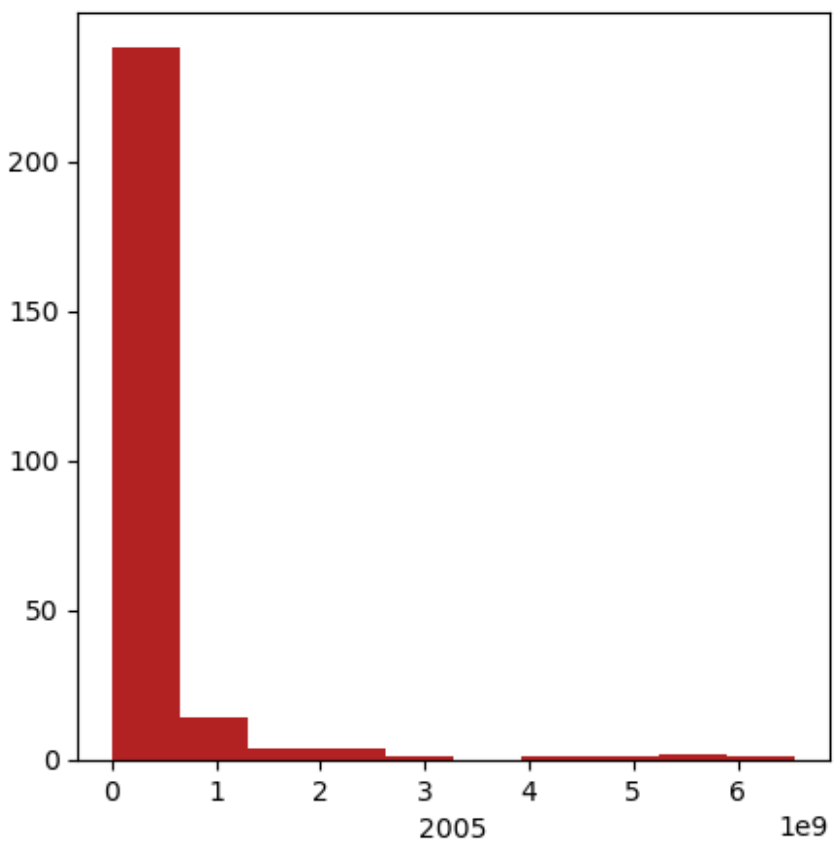
```
plt.xlabel(i)
plt.show()
```



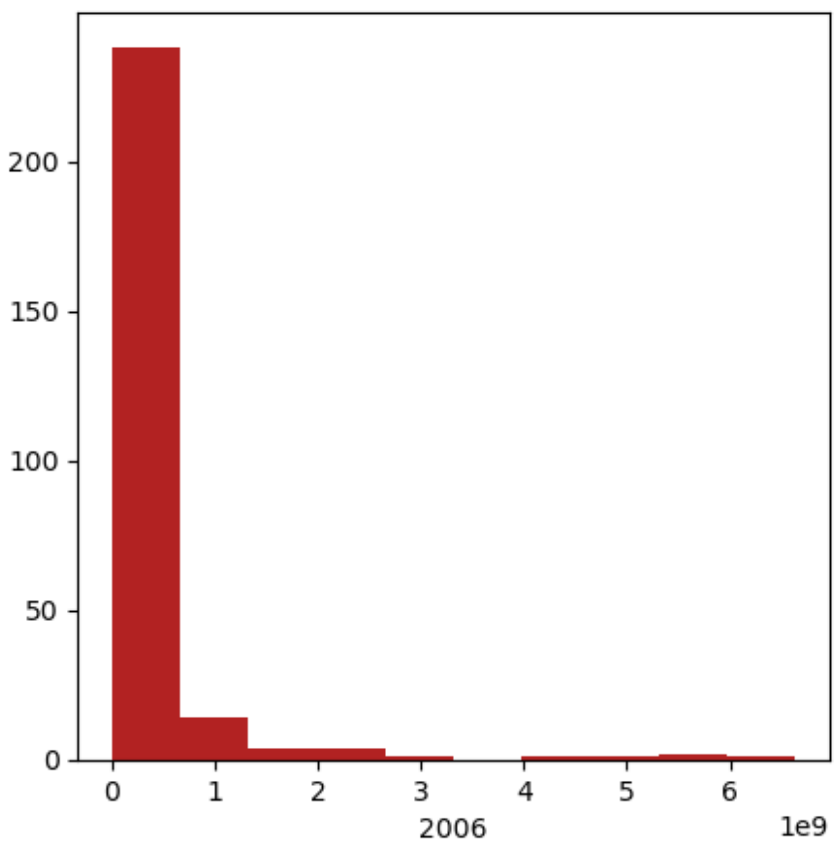


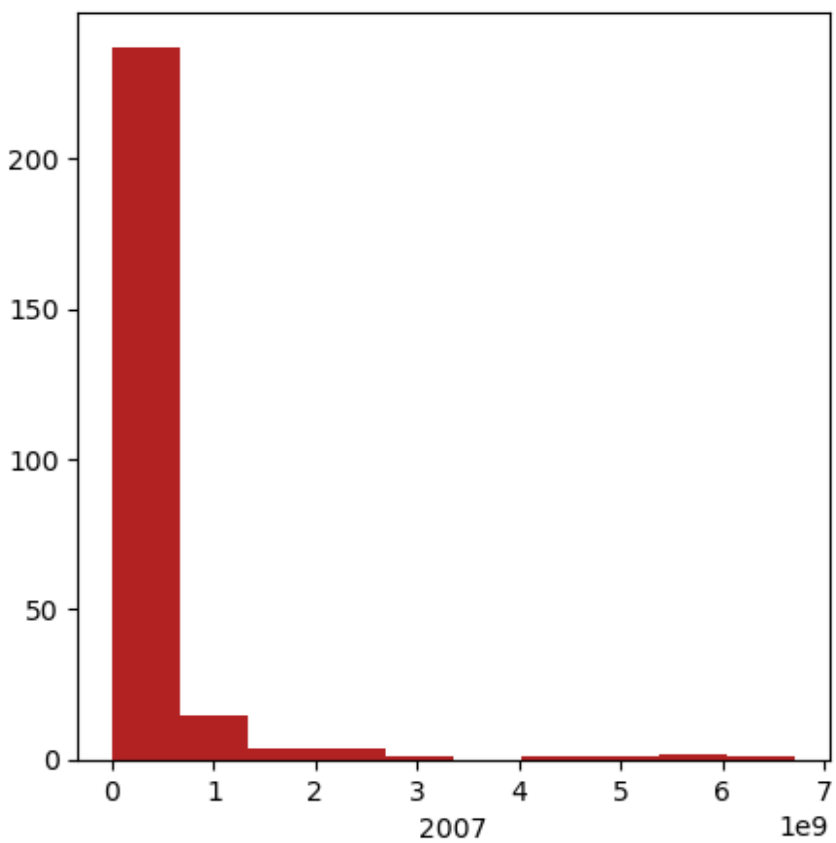


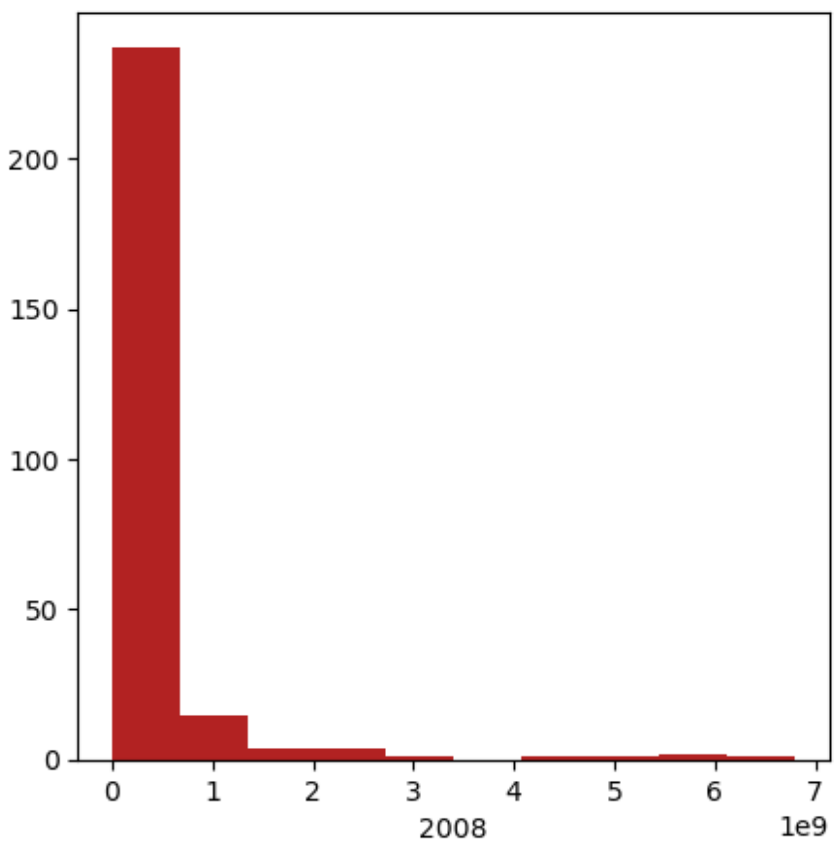


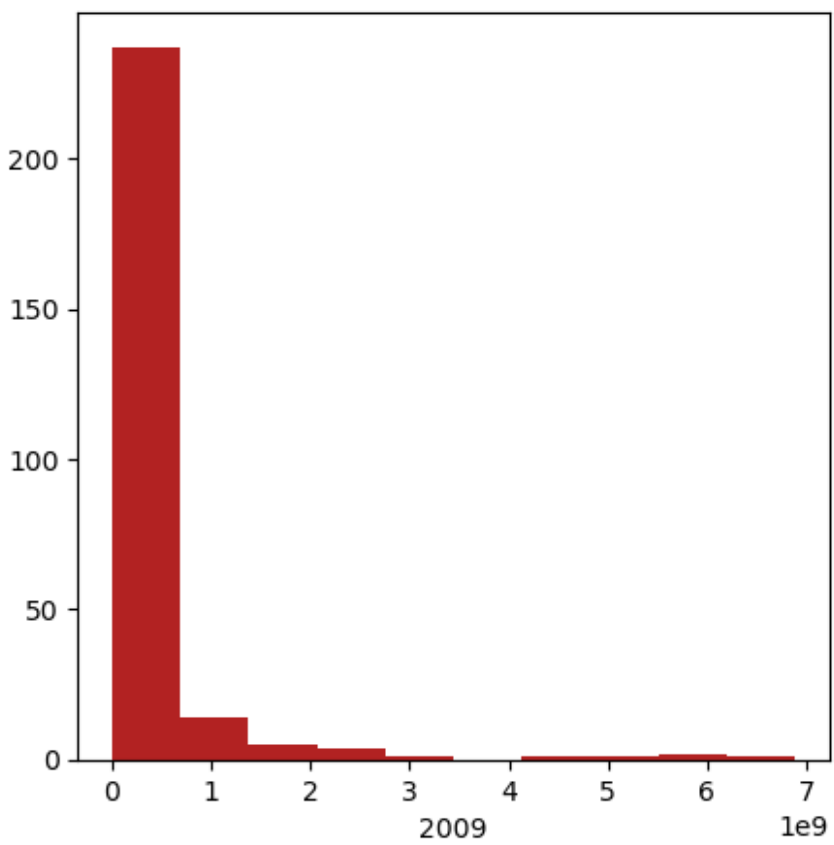


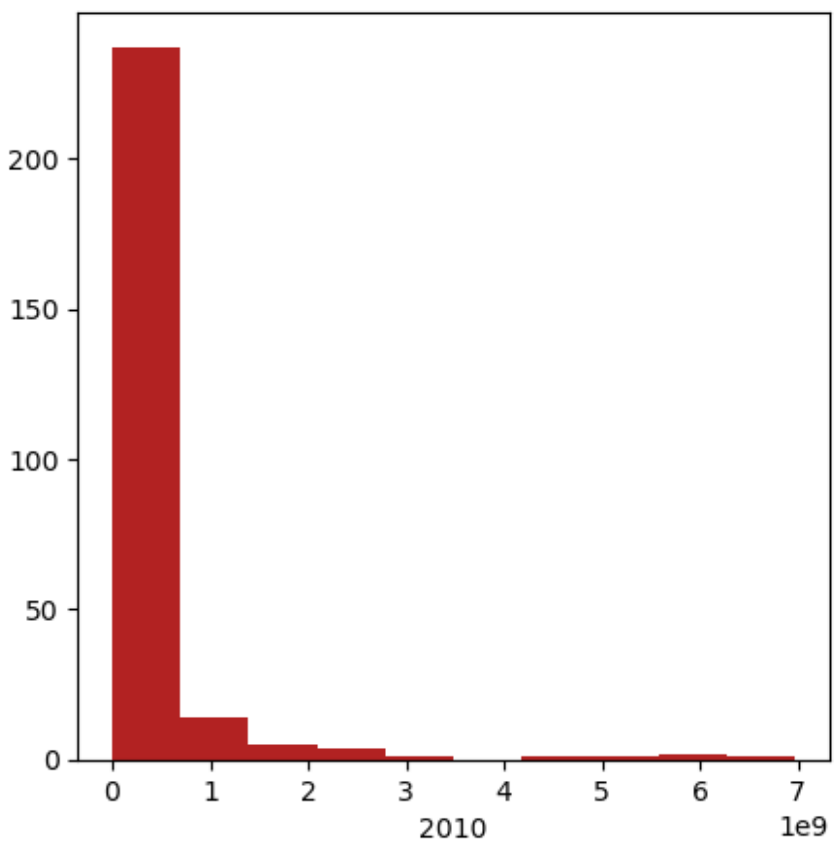


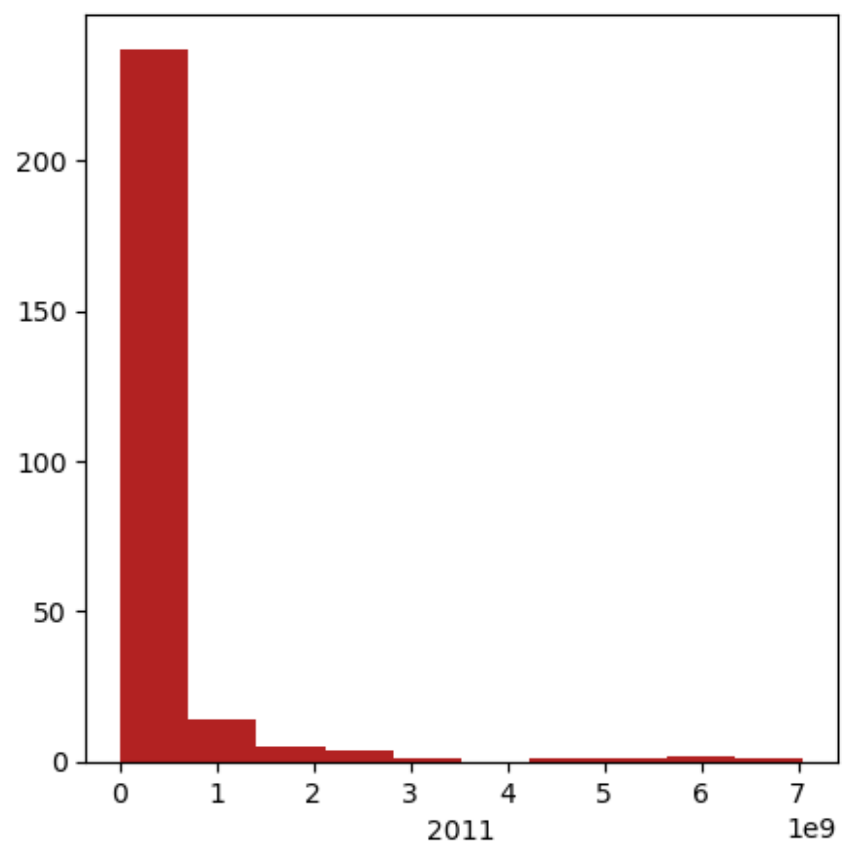


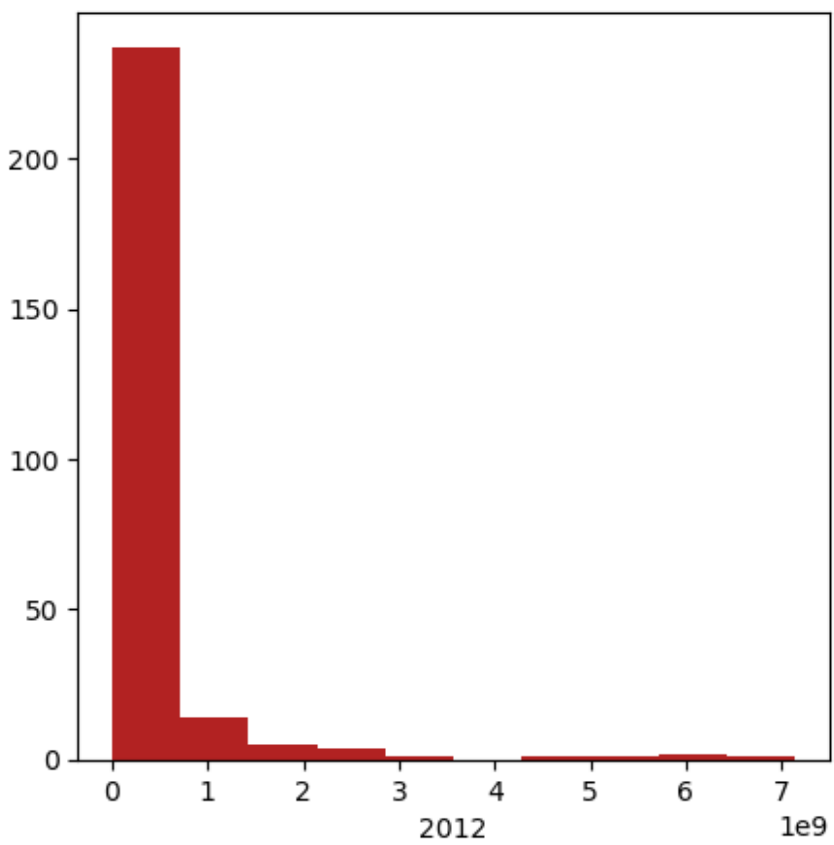


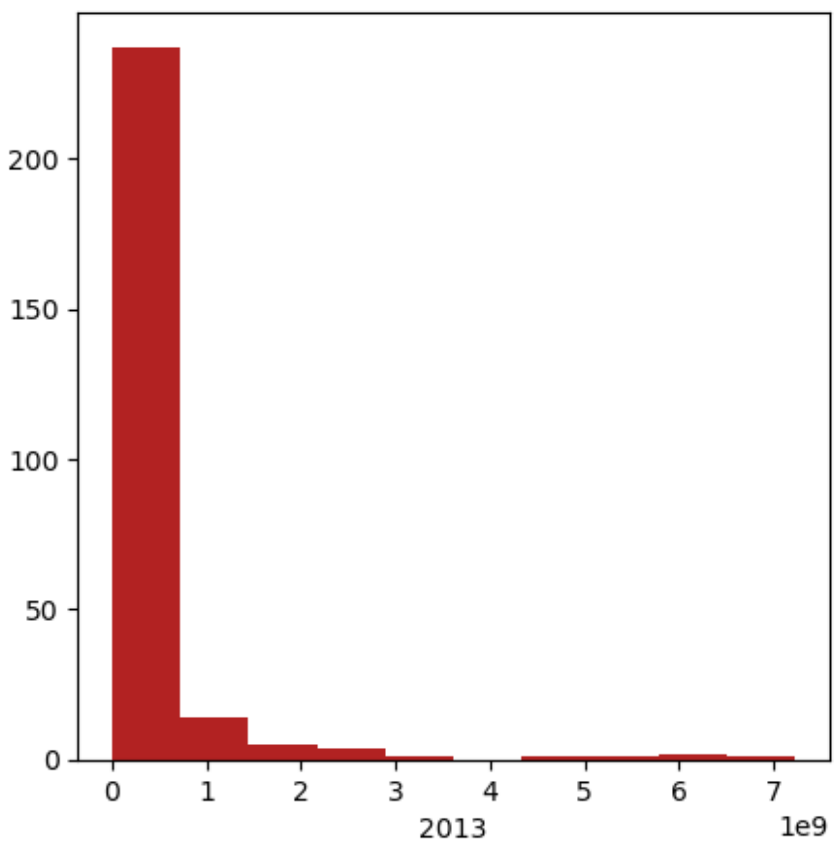




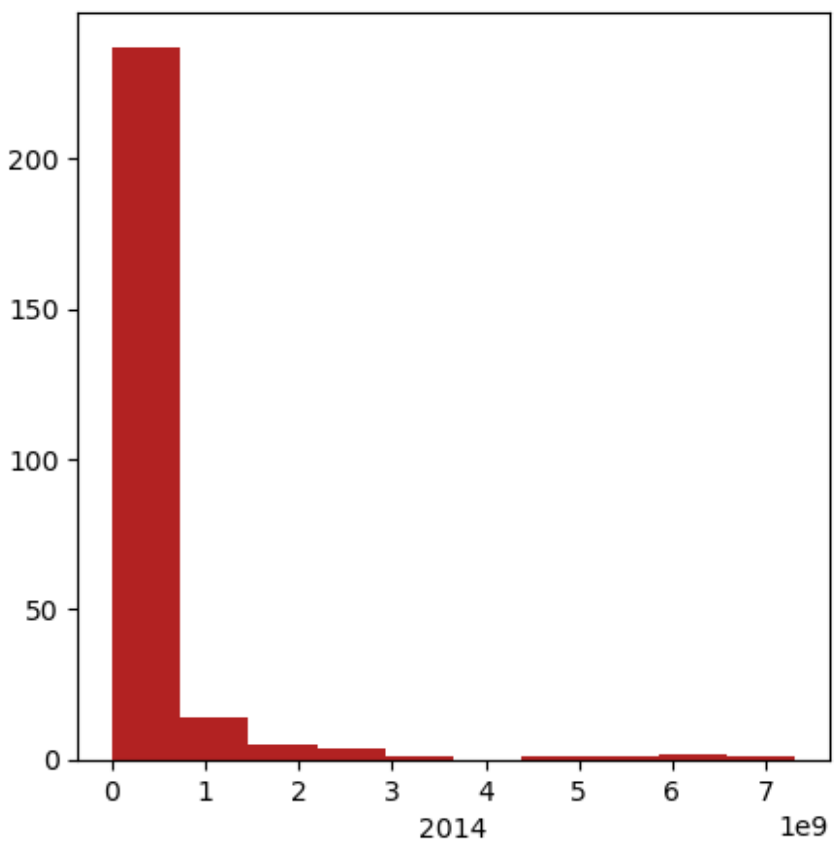


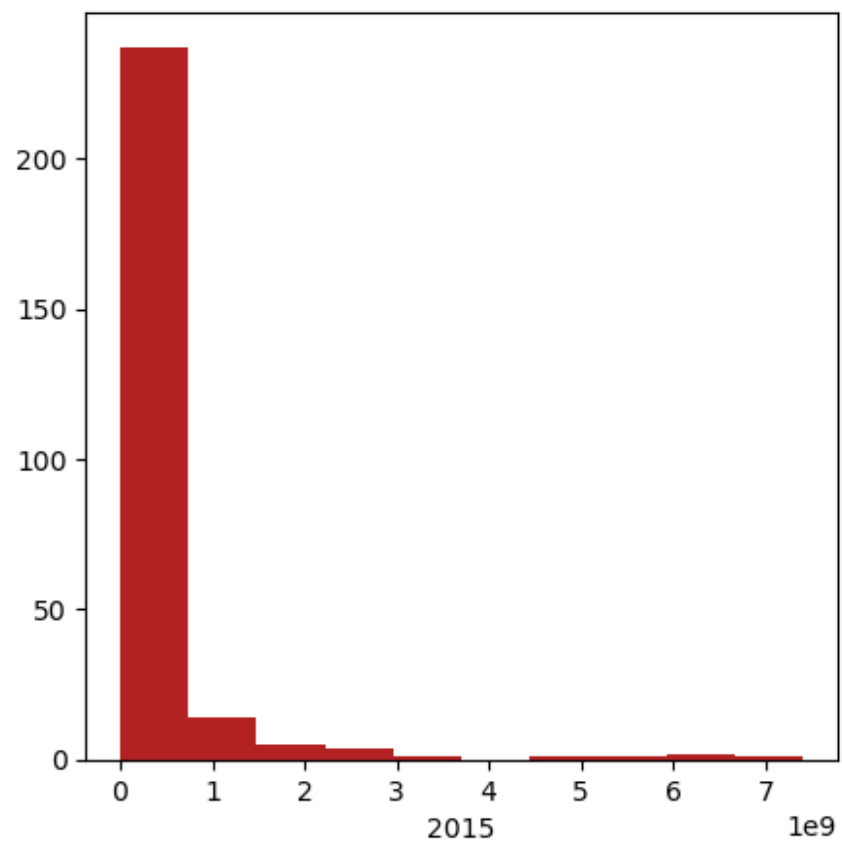


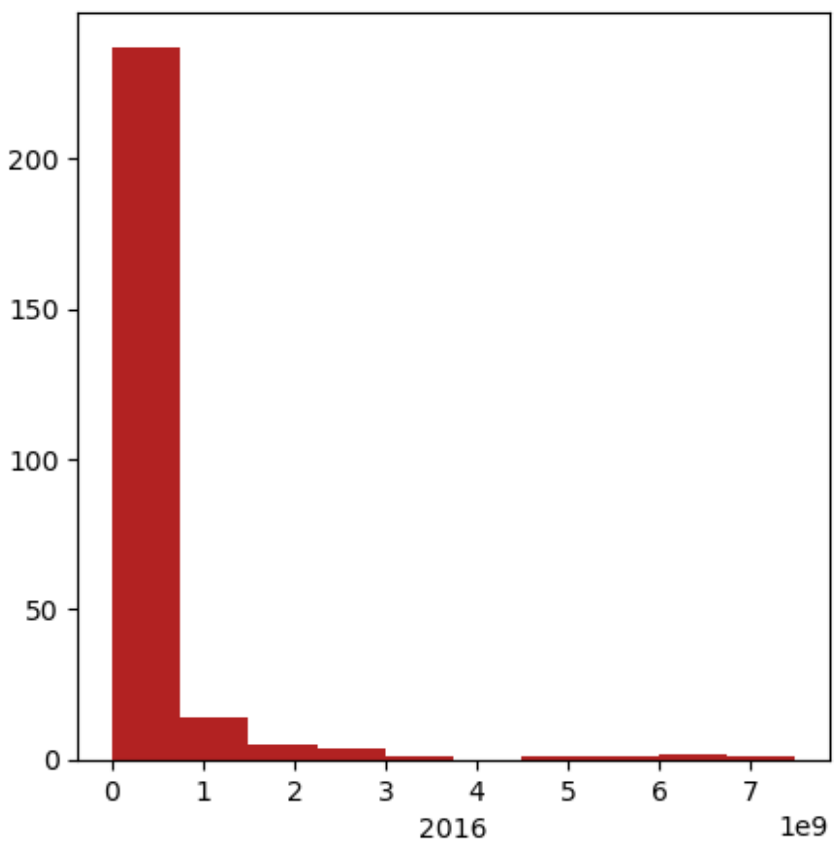


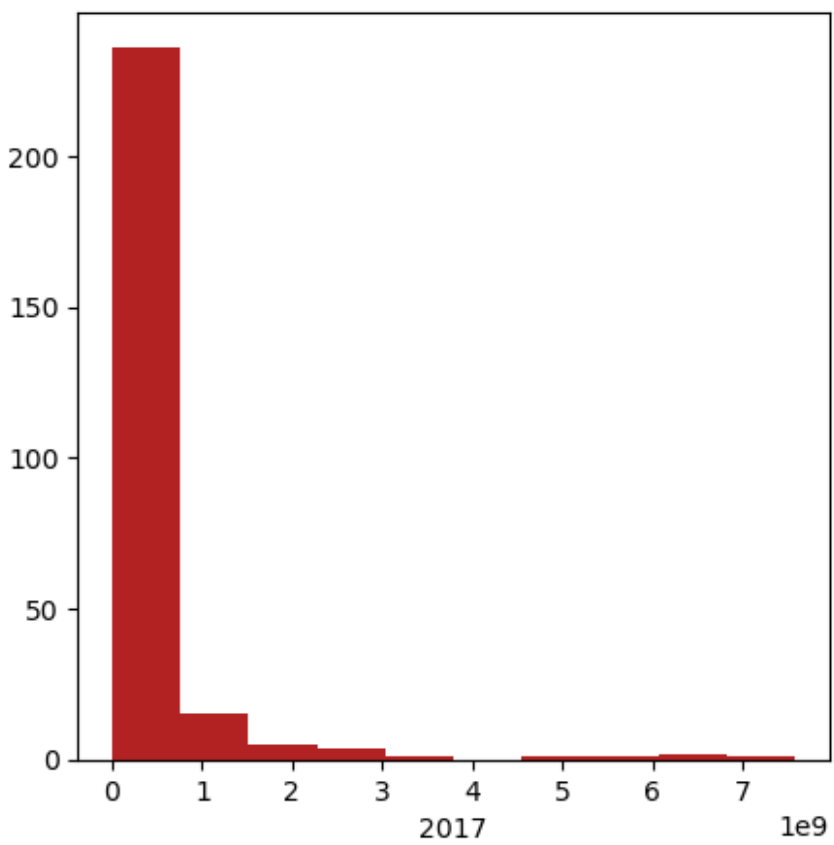


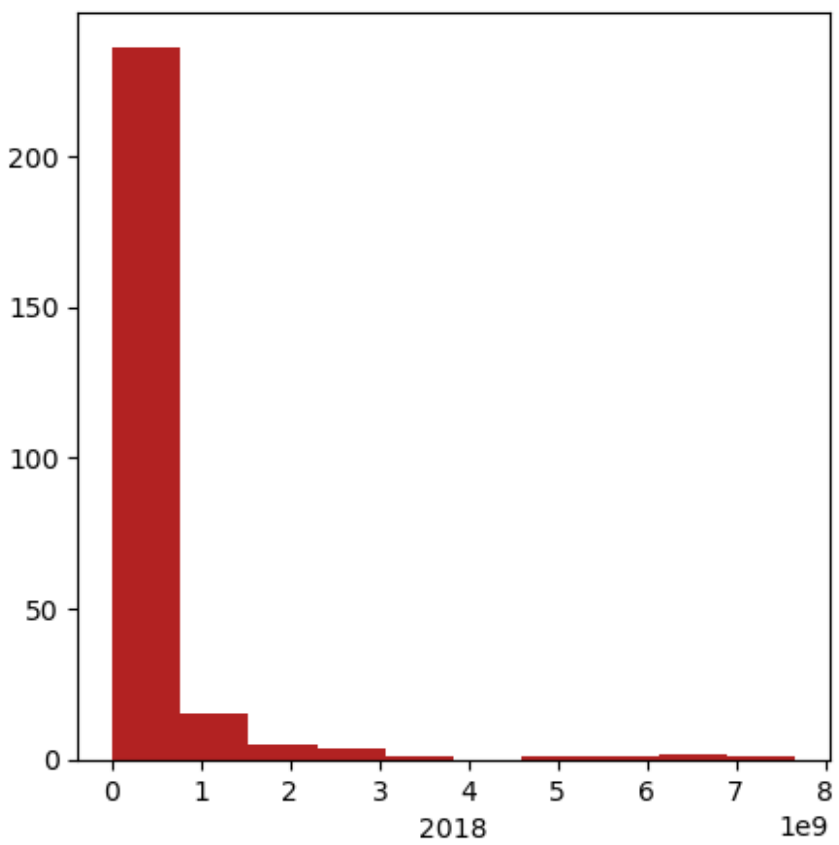


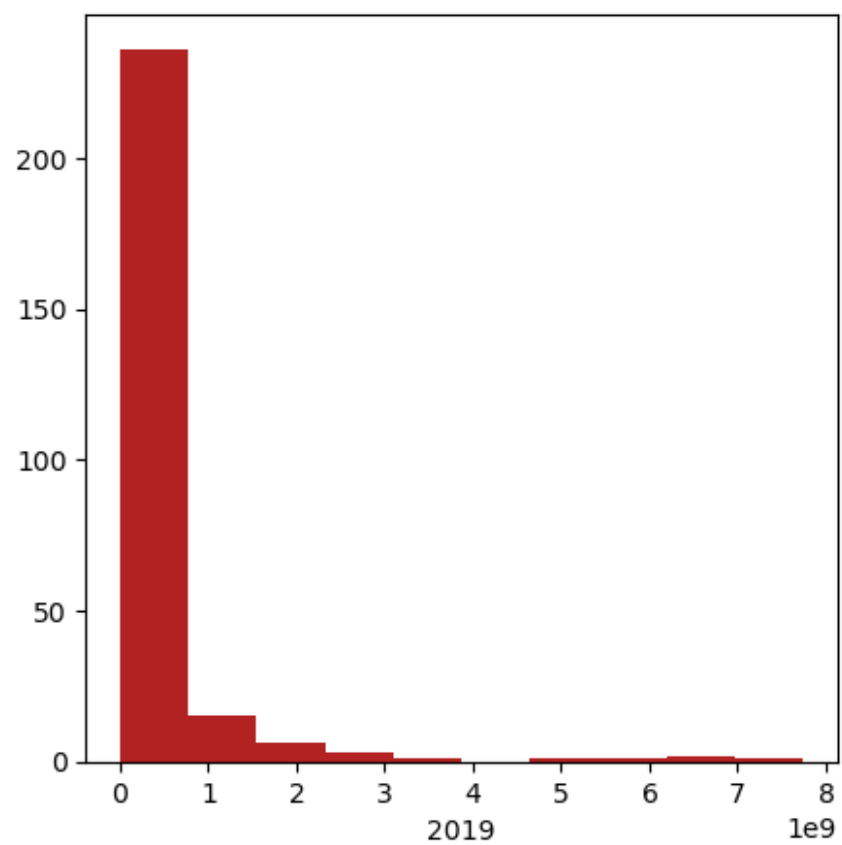


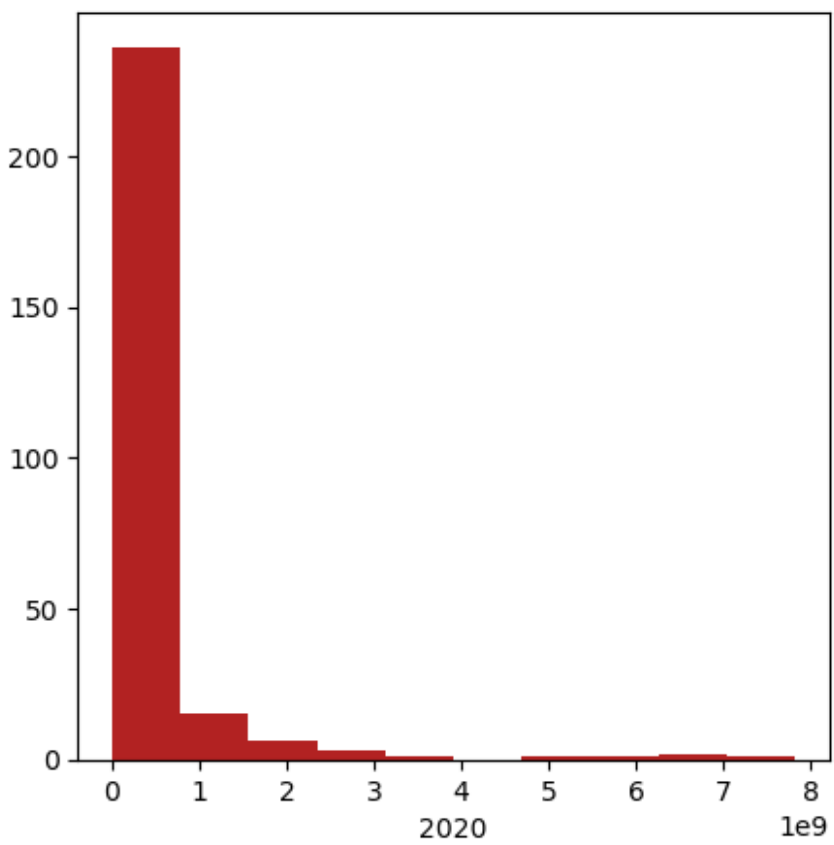


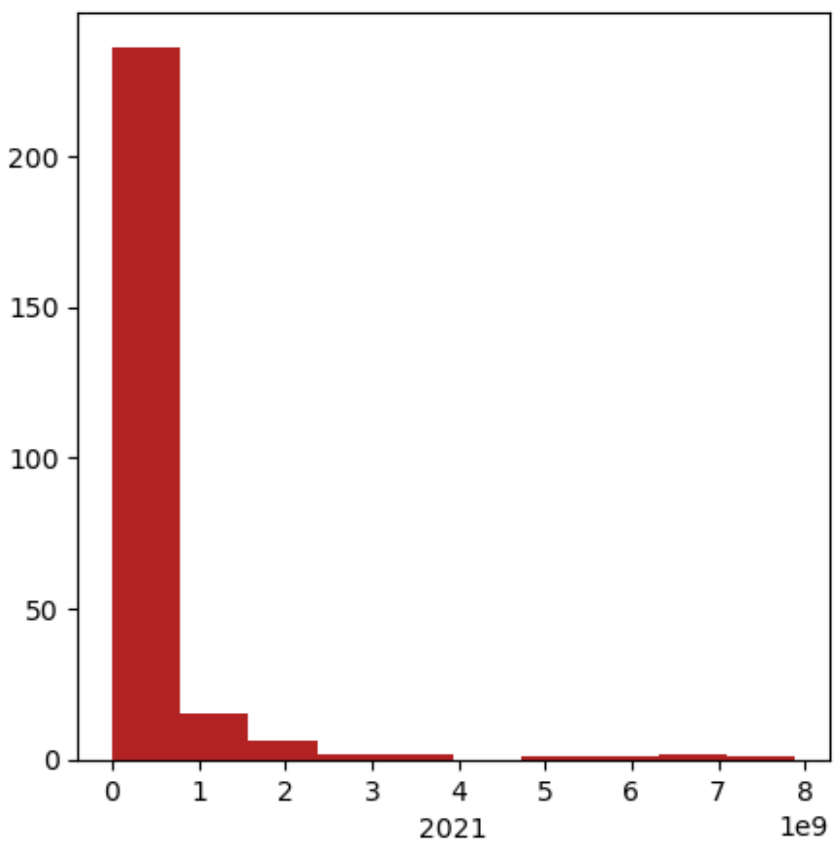




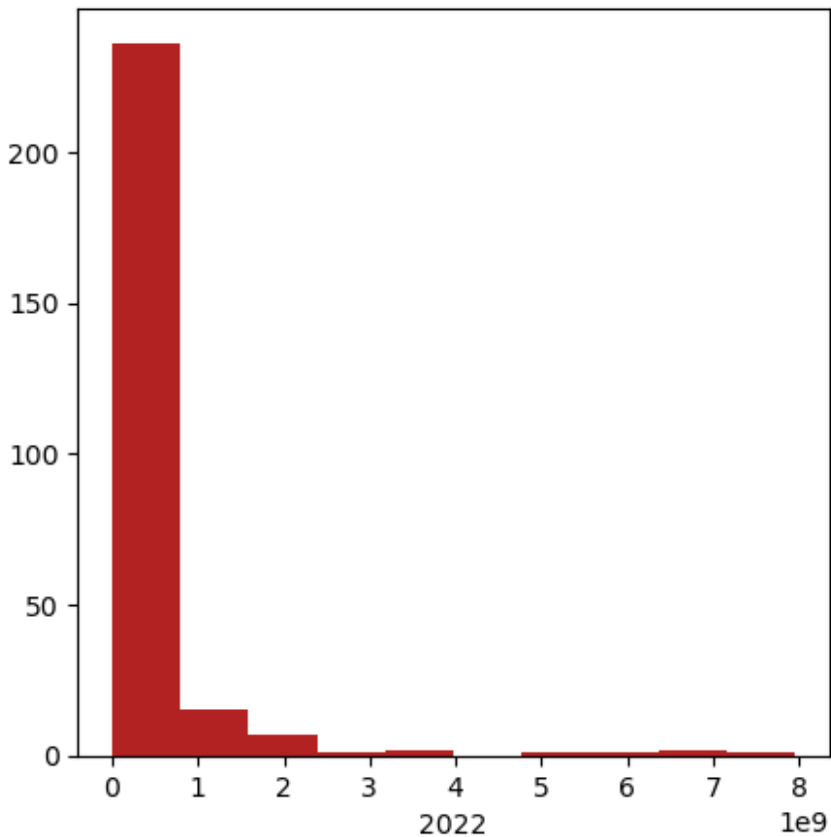










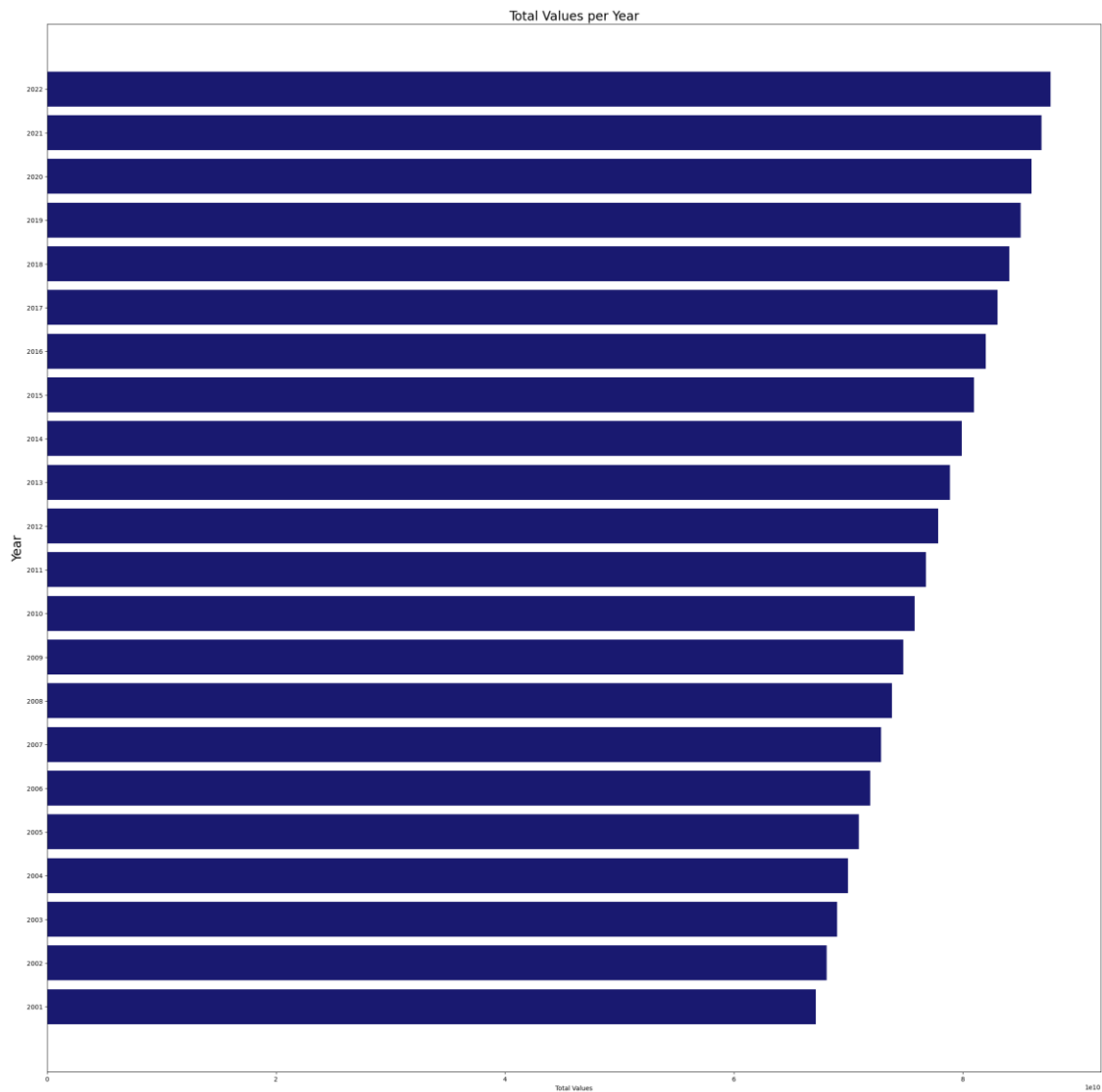


### Plotting Barplot for each year

```
# Extract the years (columns) from the DataFrame
years = df.columns[1:] # Skip the first two columns ('Country Name' and
                        'Country Code')

# Sum the values for each year across all countries
total_values = df[years].sum()

# Plotting the bar chart
plt.figure(figsize=(30, 30)) # Adjust the figure size as per your preference
plt.barh(years, total_values, color='#191970')
plt.xlabel('Total Values')
plt.ylabel('Year', size=20)
plt.title('Total Values per Year', size=20)
plt.show()
```



## Insights

*As we can see with the passage of time (from 2001 to 2022) the world development indicator keep rising for the different countries.*

## Plotting Indicators of top 10 countries for different years by the base\_year 2001

```
country_by_2001 = df.sort_values(by='2001').head(10)
country_by_2001
```

Country Name		2001	2002	2003	2004	2005
245	Tuvalu	9621.0	9609.0	9668.0	9791.0	9912.0
179	Nauru	10363.0	10351.0	10344.0	10335.0	10318.0
228	Turks and Caicos Islands	19578.0	20598.0	21739.0	22869.0	23995.0
188	Palau	19828.0	19851.0	19880.0	19907.0	19831.0
255	British Virgin Islands	20657.0	21288.0	21982.0	22715.0	23497.0
212	San Marino	27335.0	27969.0	28601.0	29093.0	29508.0

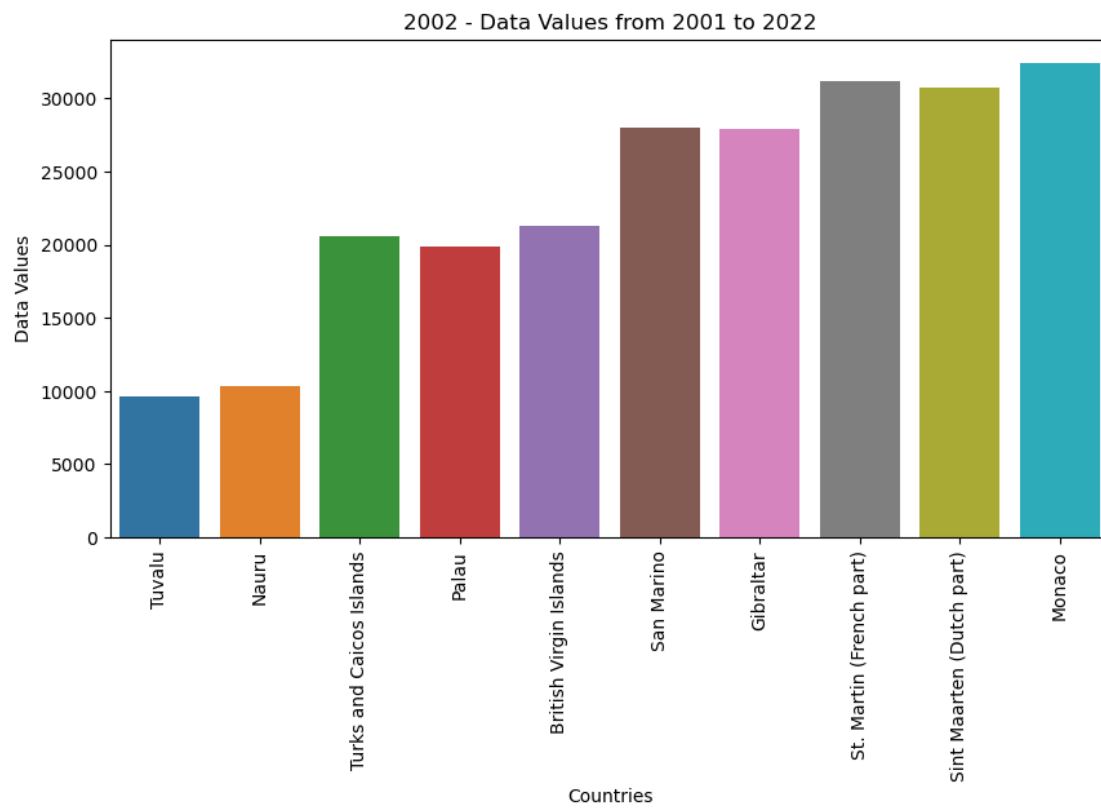
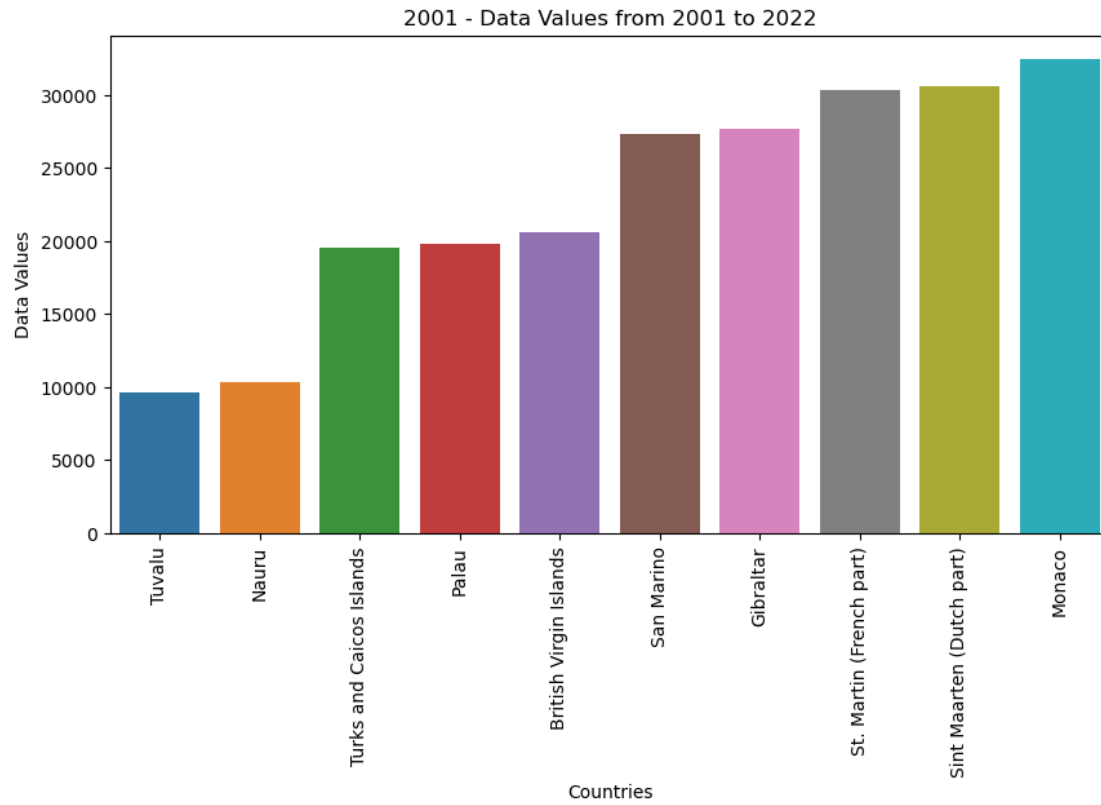
84		Gibraltar	27721.0	27892.0	28301.0	28716.0	29155.0
147	St. Martin (French part)		30387.0	31160.0	31929.0	32697.0	33452.0
225	Sint Maarten (Dutch part)		30600.0	30777.0	31472.0	32488.0	33011.0
149		Monaco	32444.0	32386.0	32316.0	32236.0	32141.0

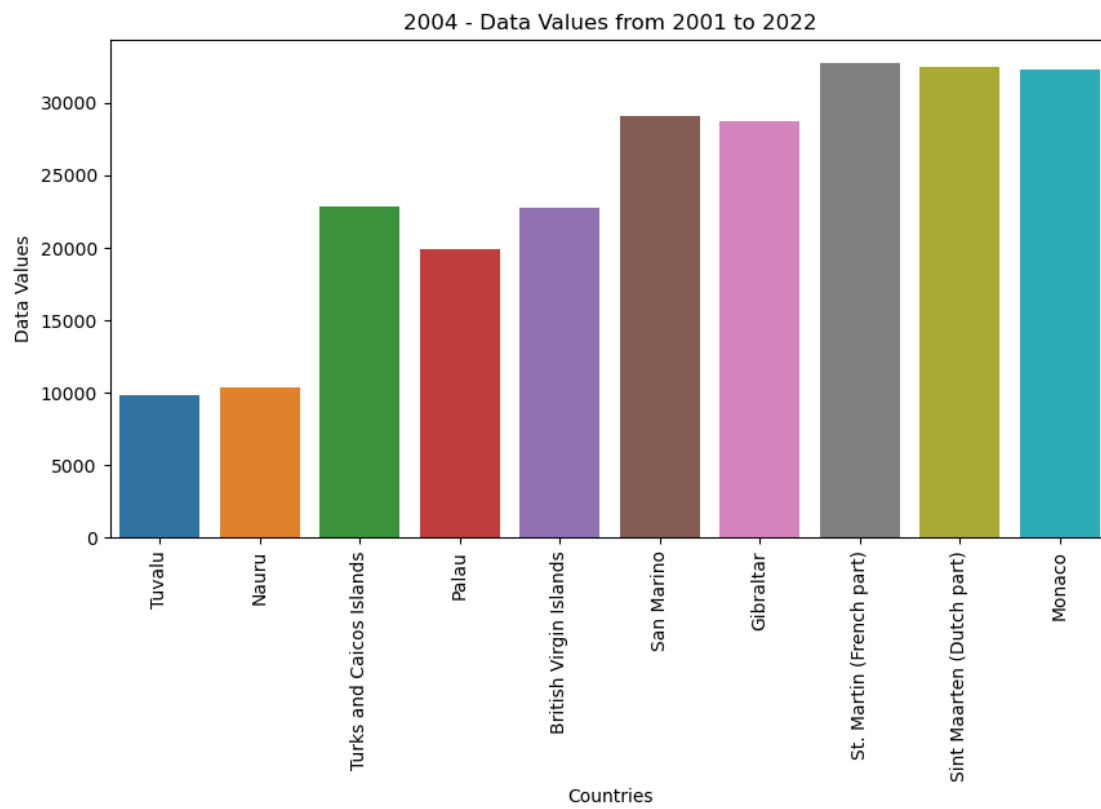
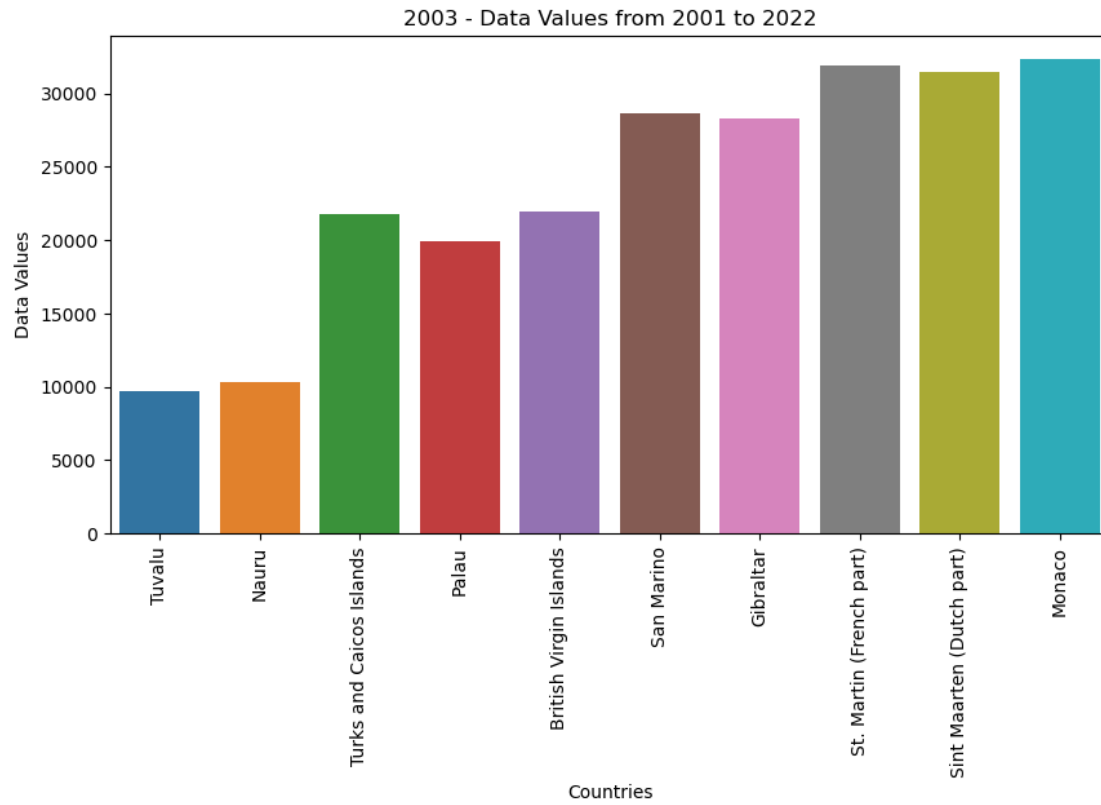
	2006	2007	2008	2009	...	2013	2014	2015	\
245	10030.0	10149.0	10272.0	10408.0	...	10918.0	10899.0	10877.0	
179	10294.0	10267.0	10243.0	10233.0	...	10694.0	10940.0	11185.0	
228	25128.0	26268.0	27422.0	28581.0	...	33594.0	34985.0	36538.0	
188	19619.0	19366.0	19102.0	18826.0	...	17805.0	17796.0	17794.0	
255	24323.0	25191.0	26115.0	27044.0	...	28657.0	28971.0	29366.0	
212	29959.0	30372.0	30700.0	31059.0	...	33285.0	33389.0	33570.0	
84	29587.0	29996.0	30398.0	30819.0	...	32411.0	32452.0	32520.0	
147	34183.0	34887.0	35541.0	36132.0	...	35639.0	35261.0	35020.0	
225	33441.0	33811.0	33964.0	34238.0	...	36607.0	37685.0	38825.0	
149	32011.0	31823.0	31862.0	32401.0	...	35425.0	36110.0	36760.0	

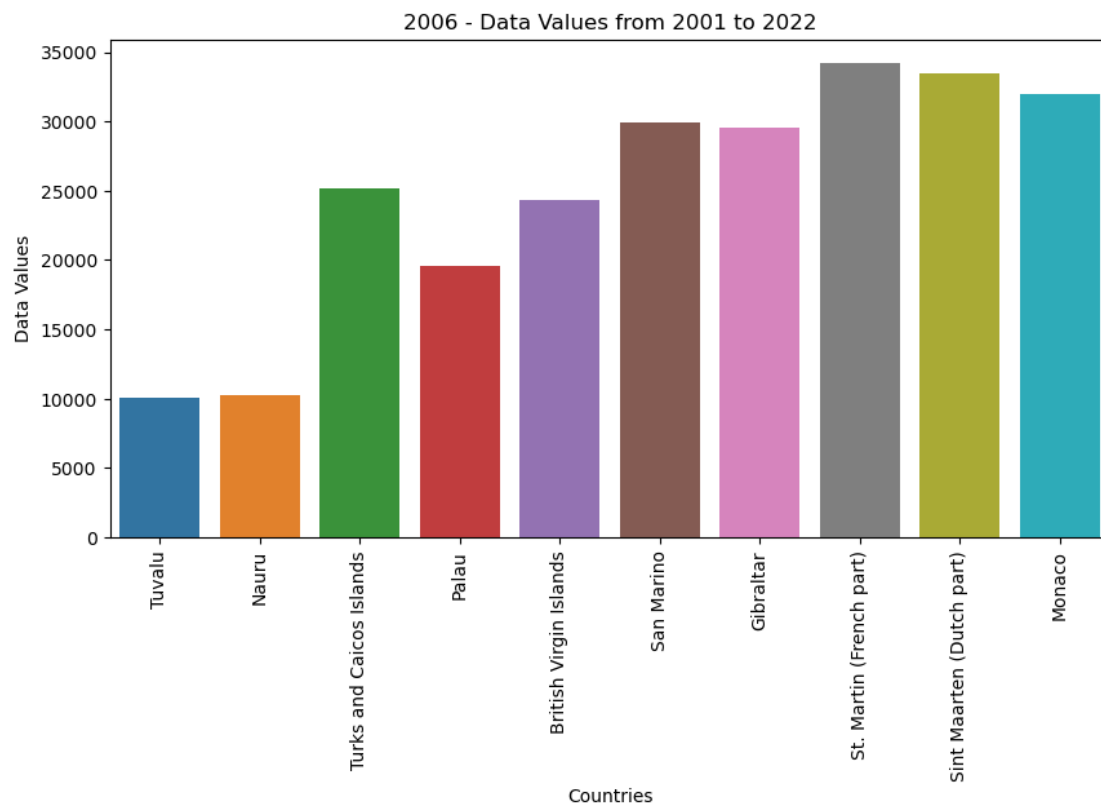
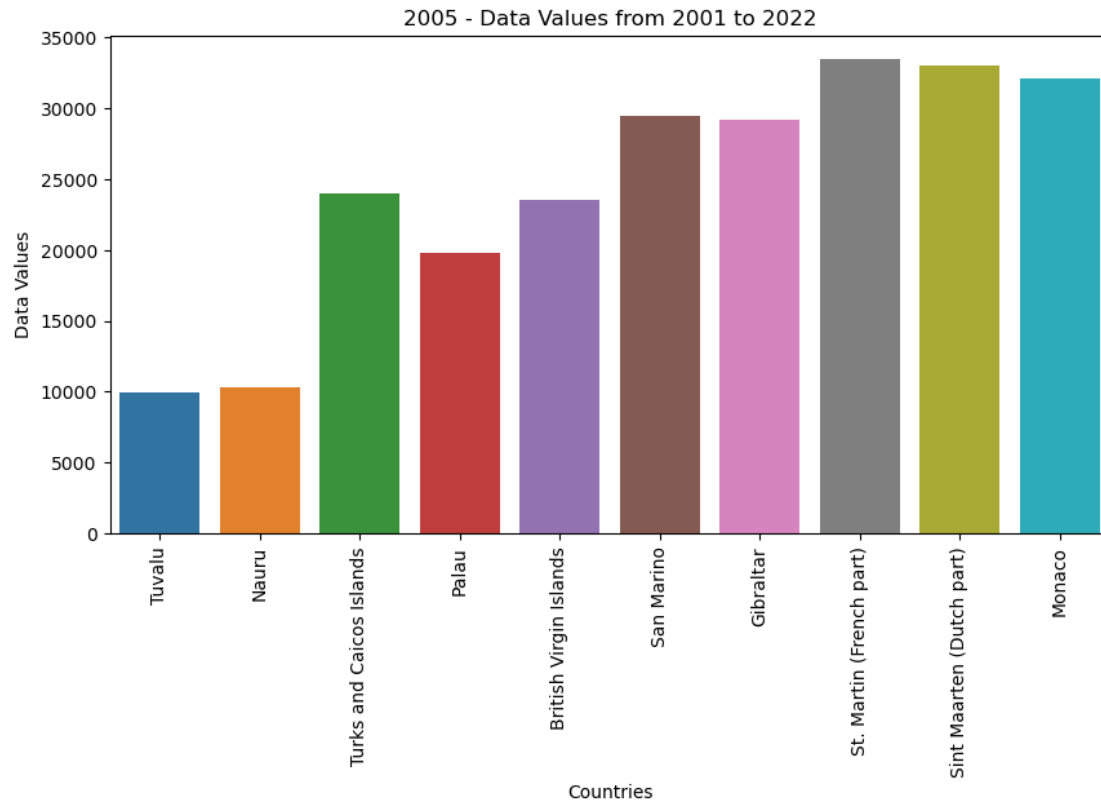
	2016	2017	2018	2019	2020	2021	2022
245	10852.0	10828.0	10865.0	10956.0	11069.0	11204.0	11312.0
179	11437.0	11682.0	11924.0	12132.0	12315.0	12511.0	12668.0
228	38246.0	39844.0	41487.0	43080.0	44276.0	45114.0	45703.0
188	17816.0	17837.0	17864.0	17916.0	17972.0	18024.0	18055.0
255	29739.0	30060.0	30335.0	30610.0	30910.0	31122.0	31305.0
212	33834.0	34056.0	34156.0	34178.0	34007.0	33745.0	33660.0
84	32565.0	32602.0	32648.0	32685.0	32709.0	32669.0	32649.0
147	34811.0	34496.0	33852.0	33121.0	32553.0	31948.0	31791.0
225	39969.0	40574.0	40895.0	41608.0	42310.0	42846.0	43389.0
149	37071.0	37044.0	37029.0	37034.0	36922.0	36686.0	36469.0

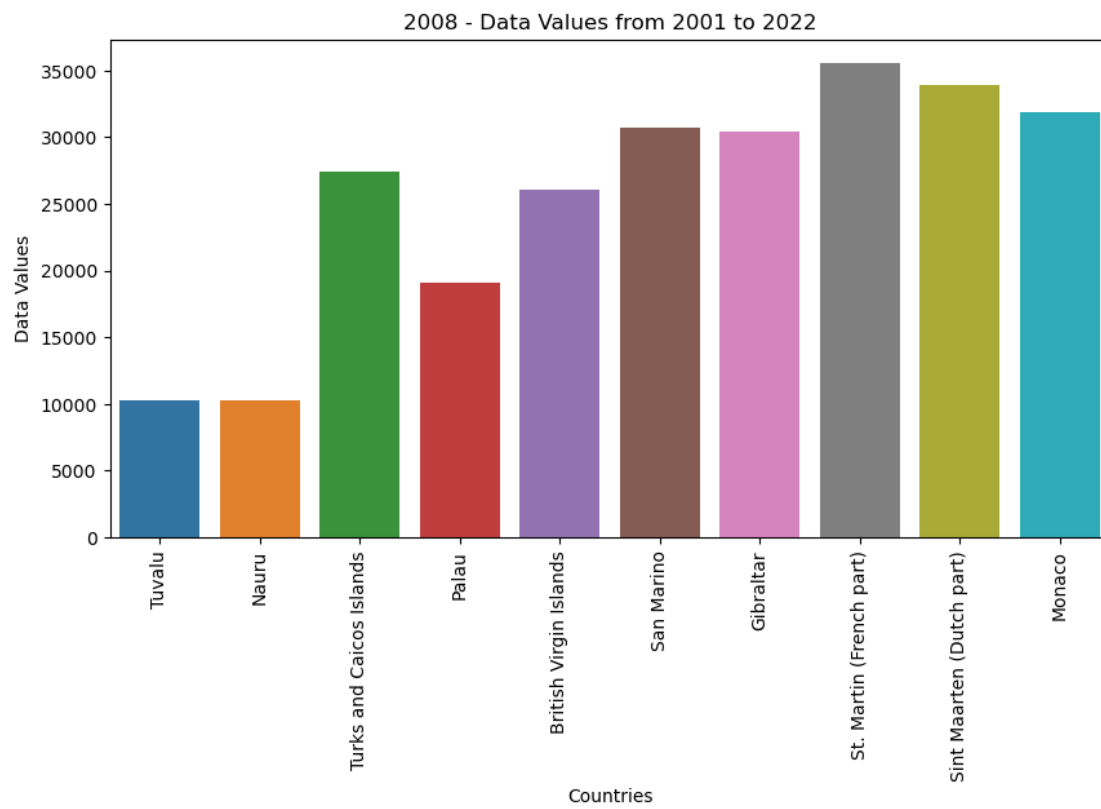
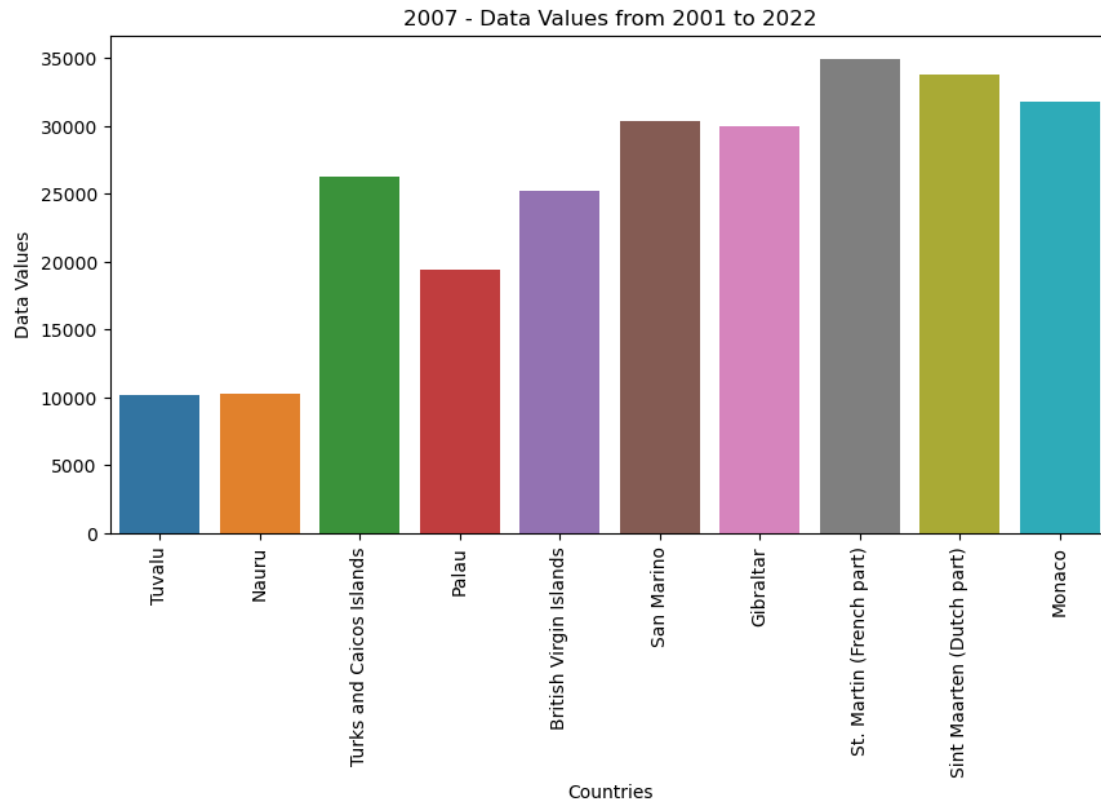
[10 rows x 23 columns]

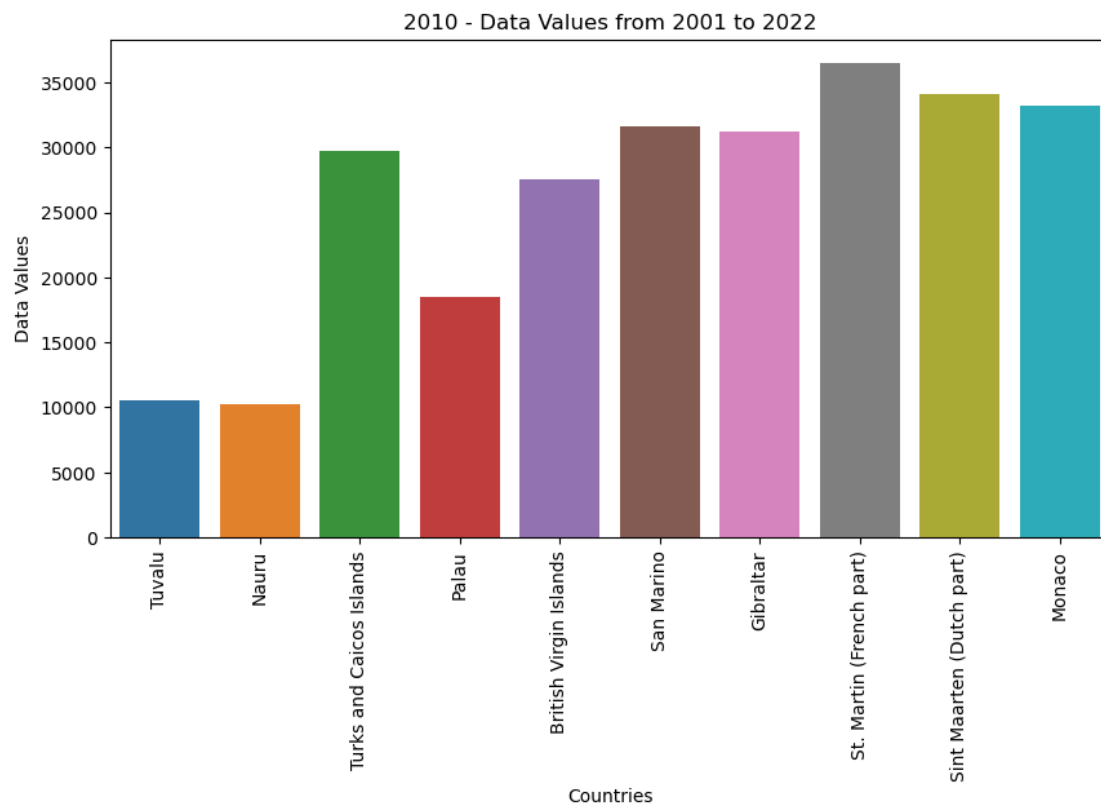
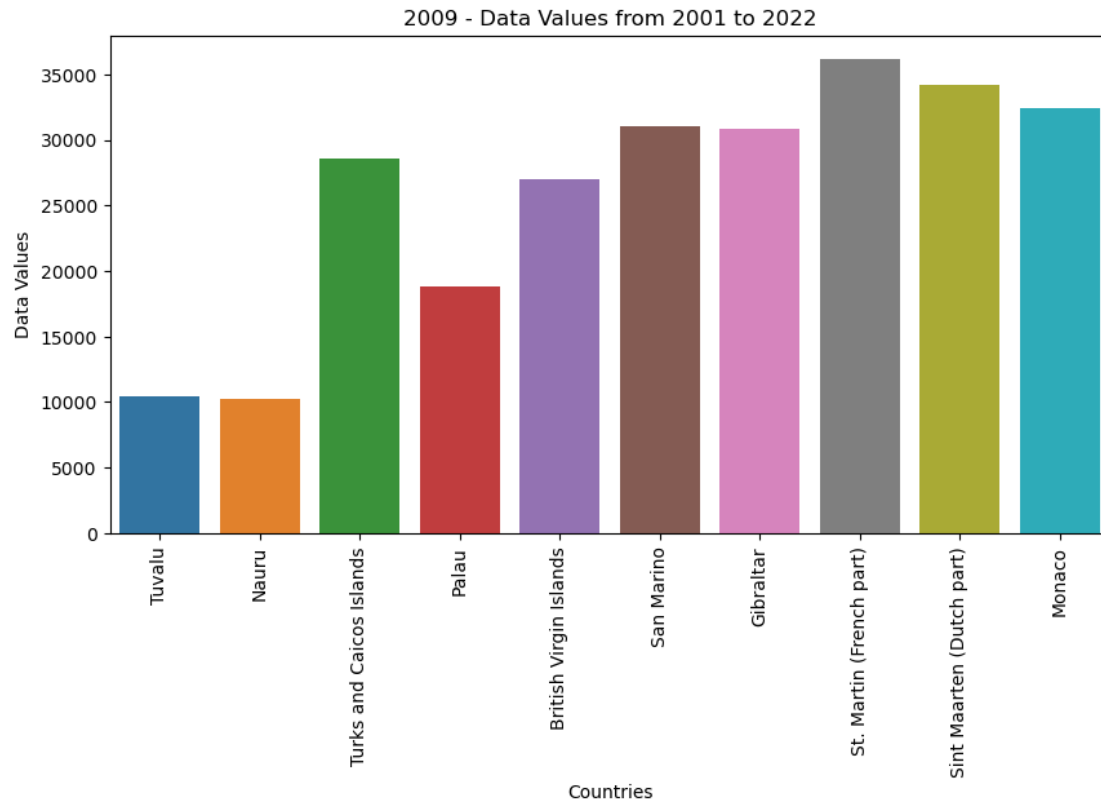
```
country_by_2001_t = country_by_2001.set_index('Country Name').T
for country_name, data_values in country_by_2001_t.iterrows():
    fig = plt.figure(figsize=(10, 5))
    sns.barplot(x=data_values.index, y=data_values.values)
    plt.xlabel('Countries')
    plt.ylabel('Data Values')
    plt.title(f"{country_name} - Data Values from 2001 to 2022")
    plt.xticks(rotation=90)
    plt.show()
```



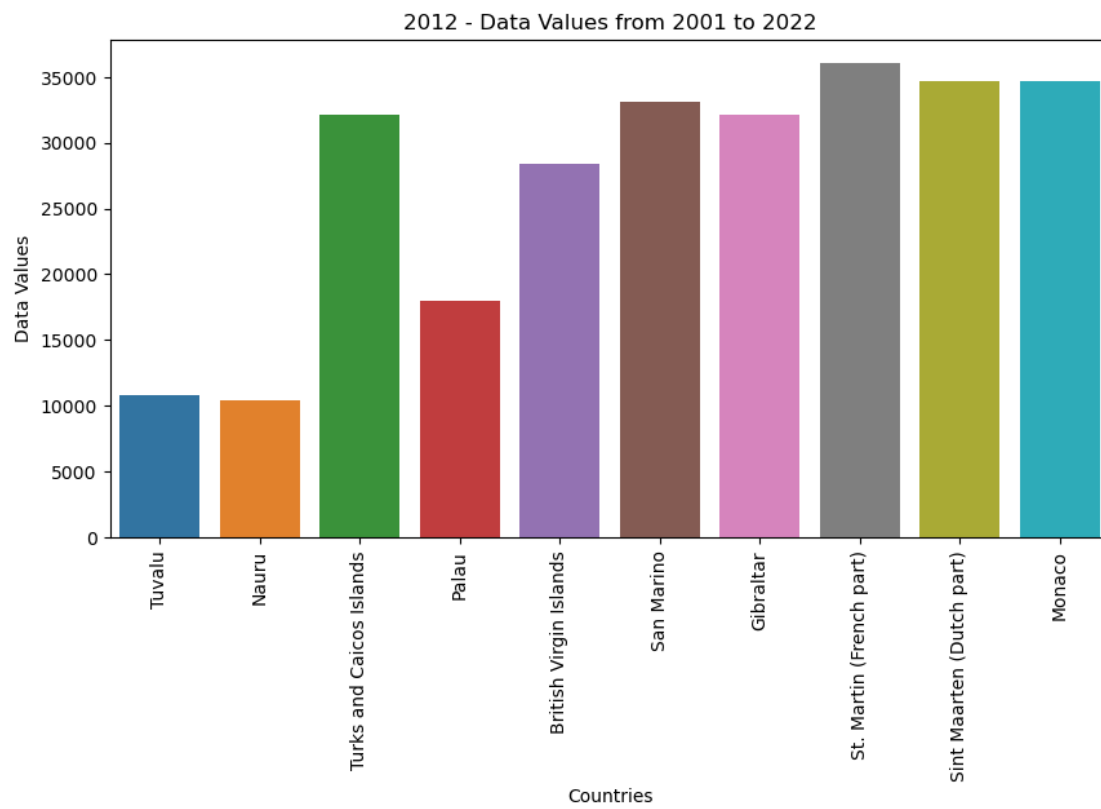
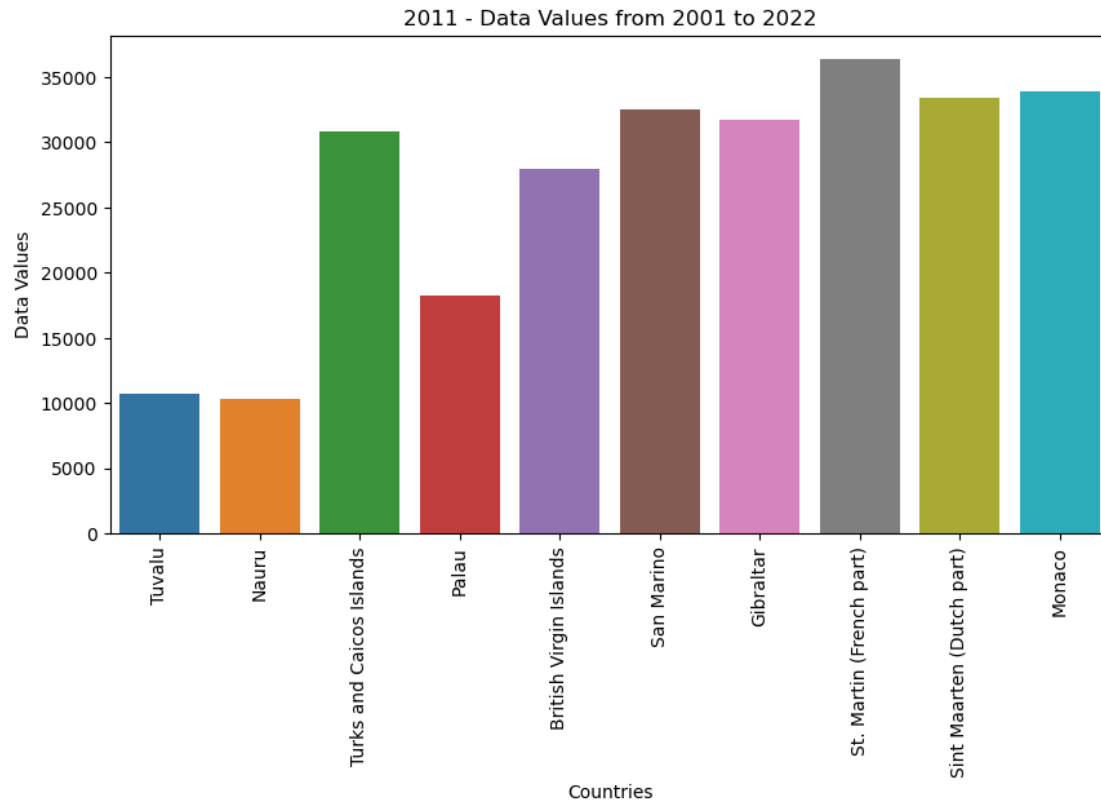


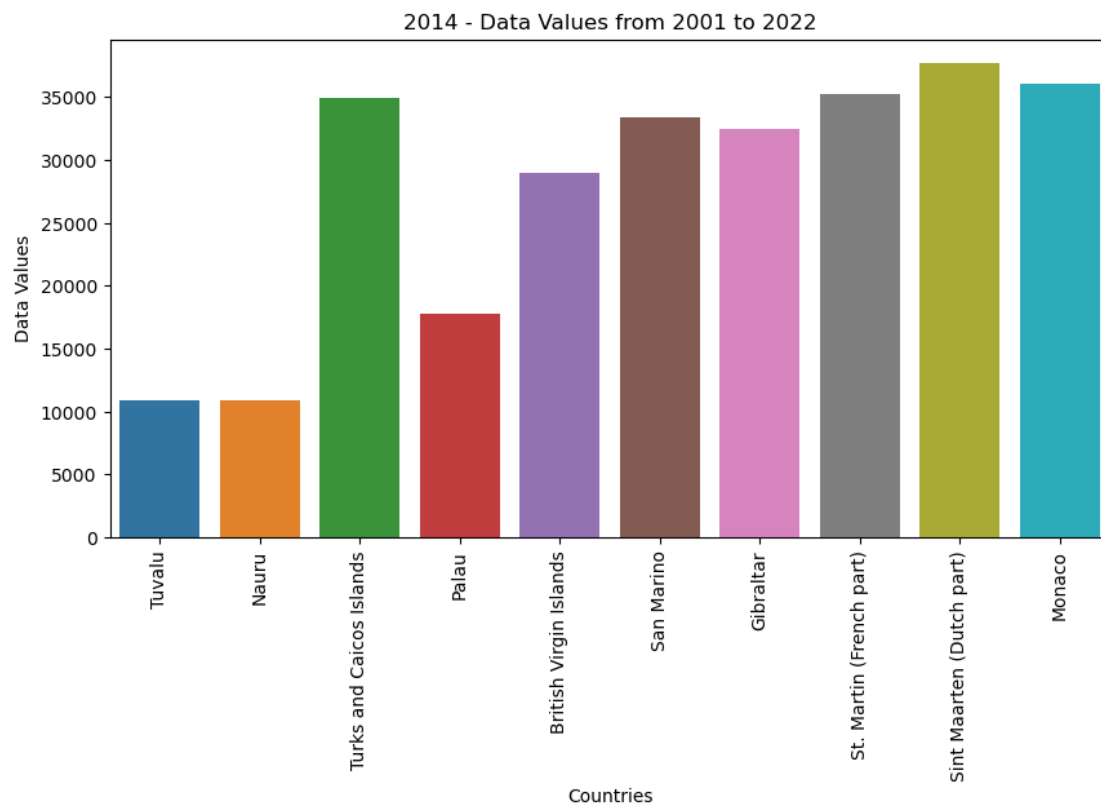
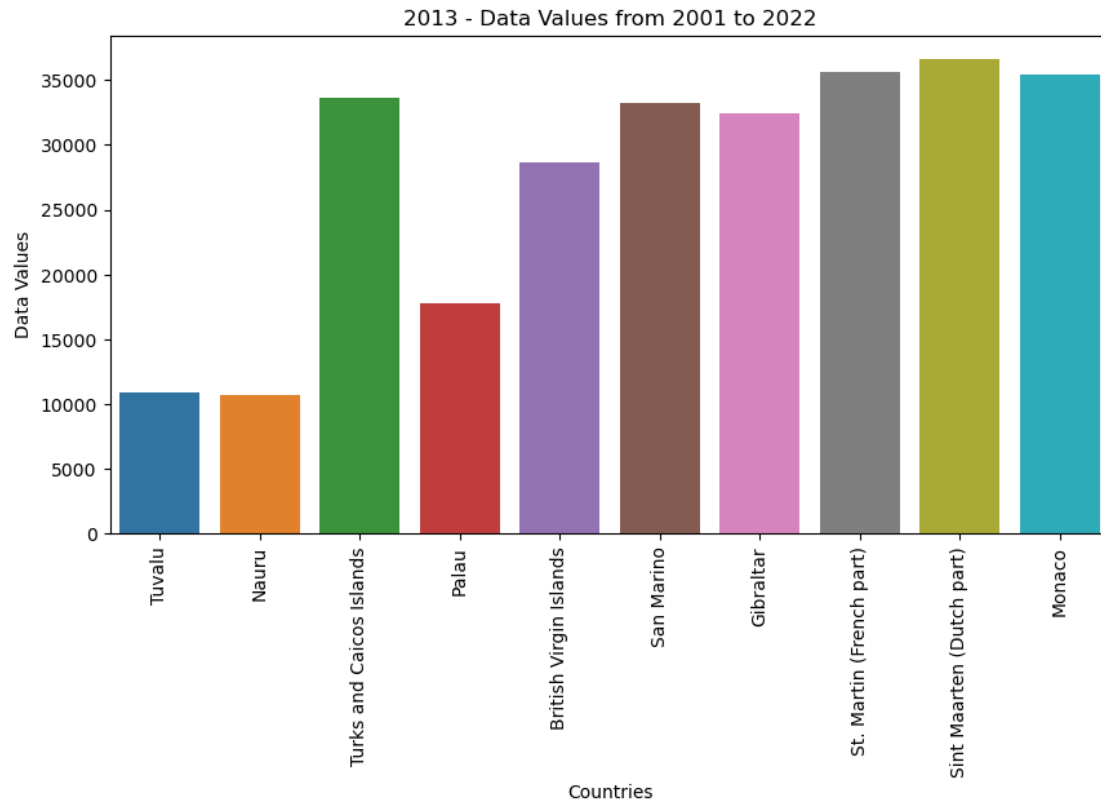


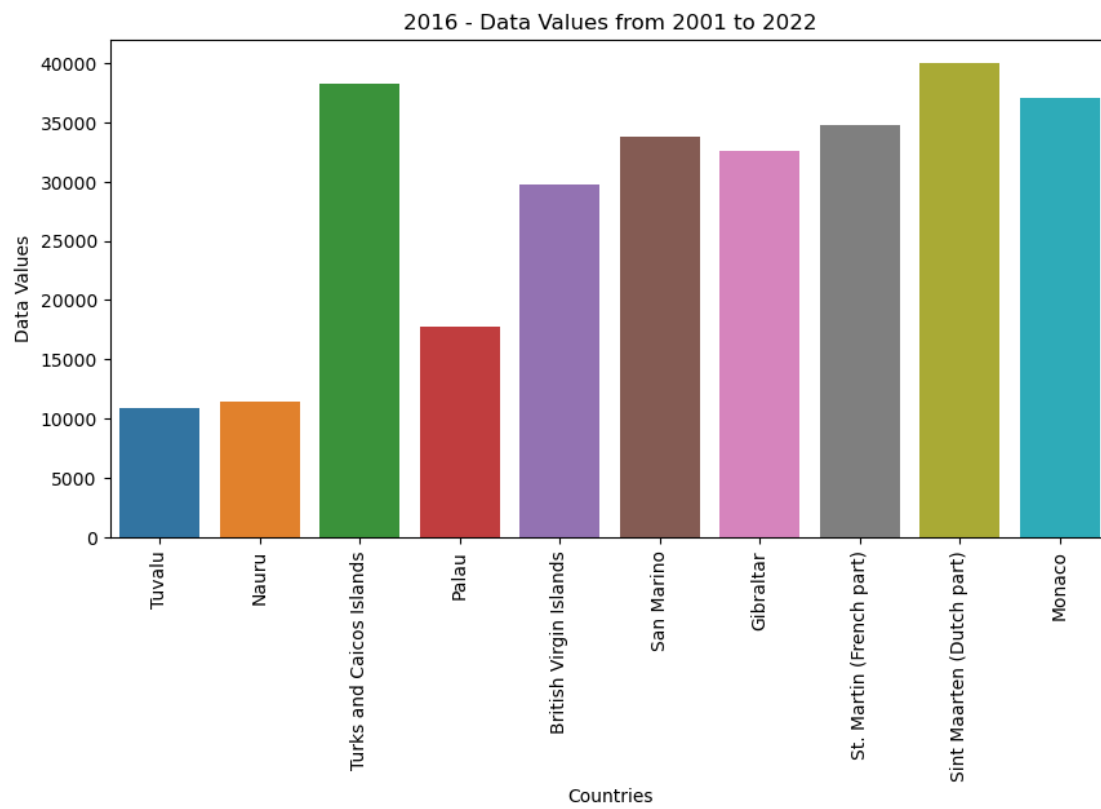
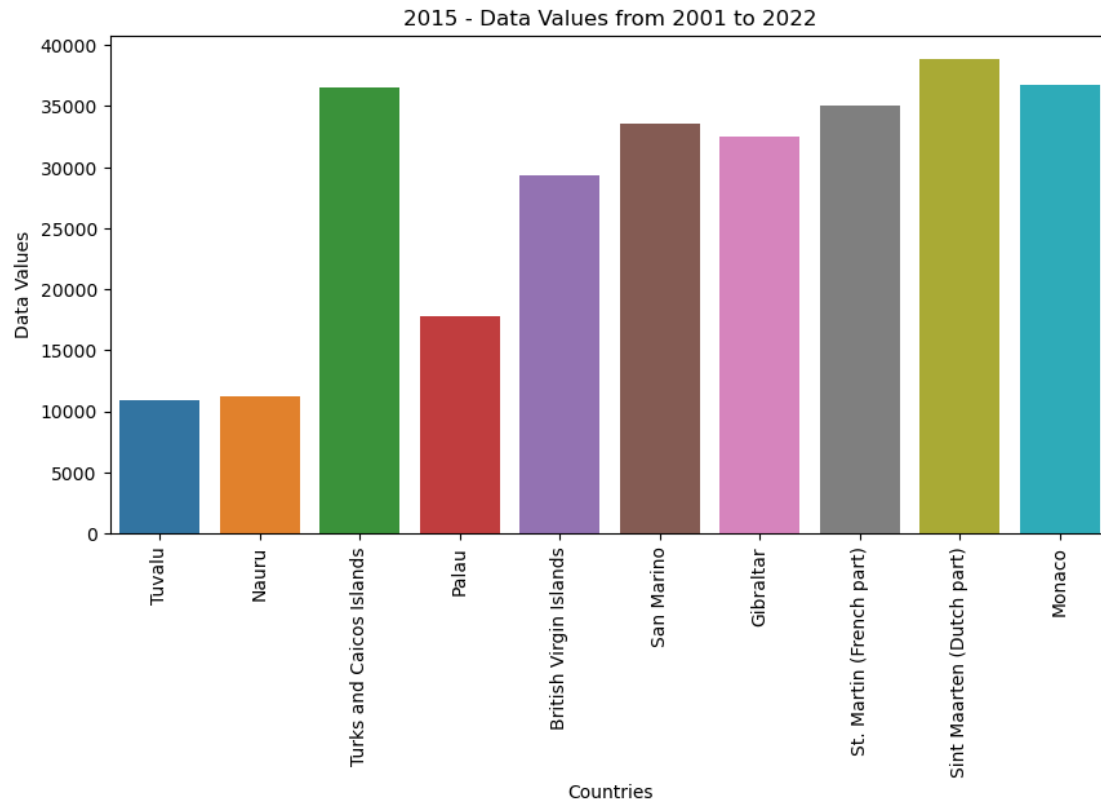


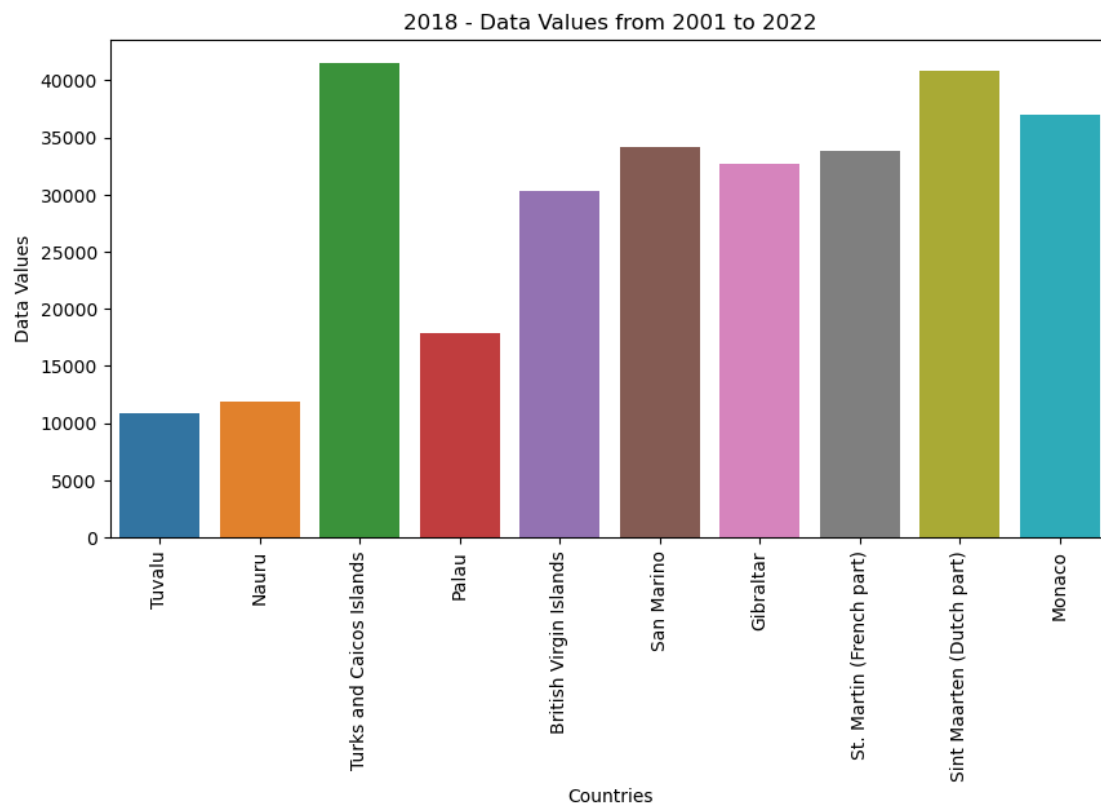
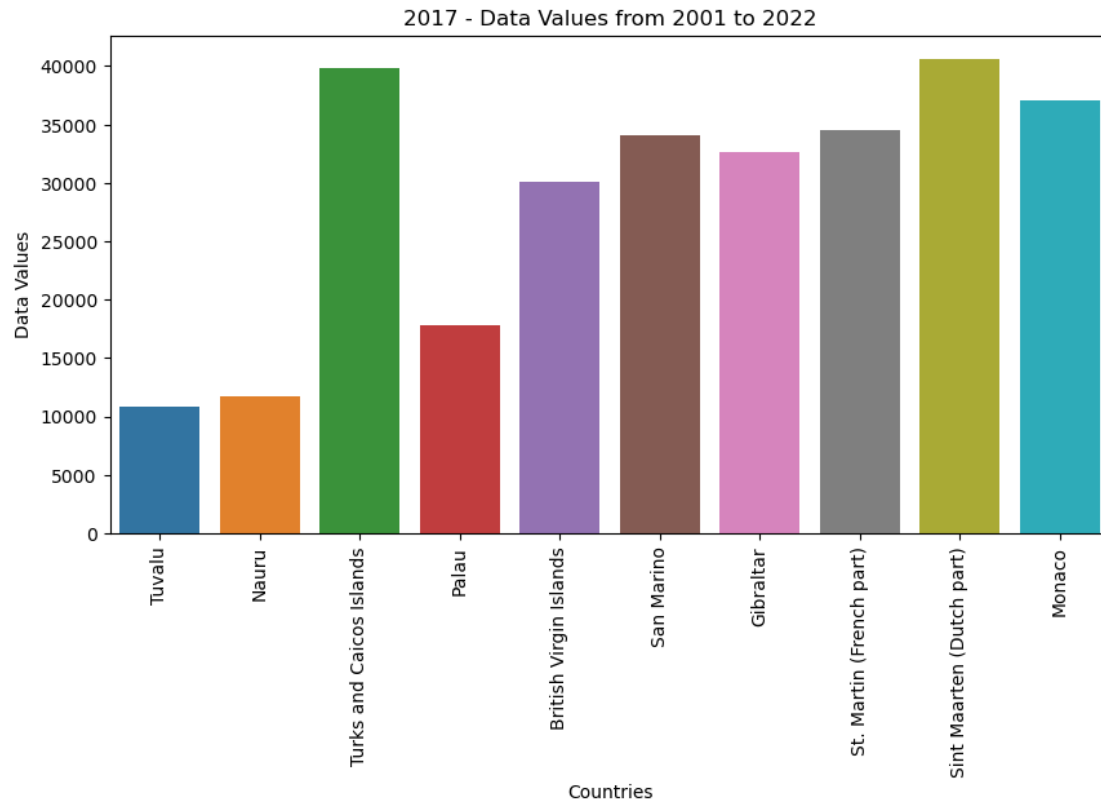


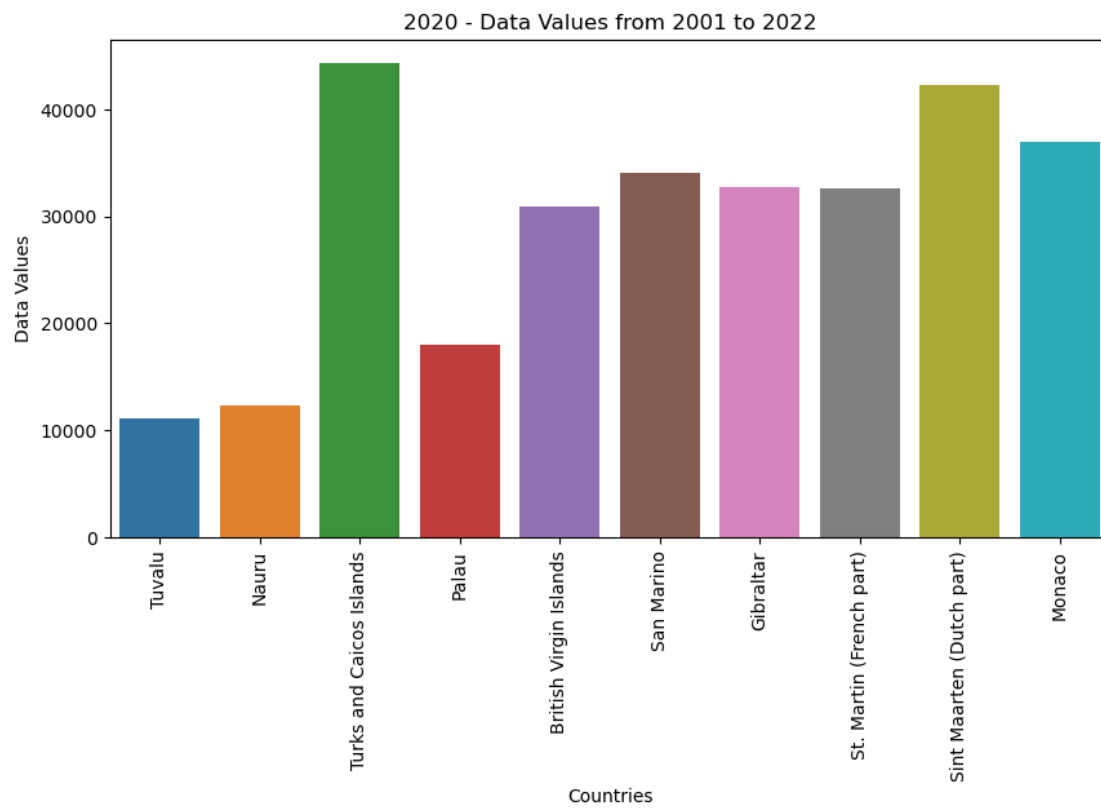
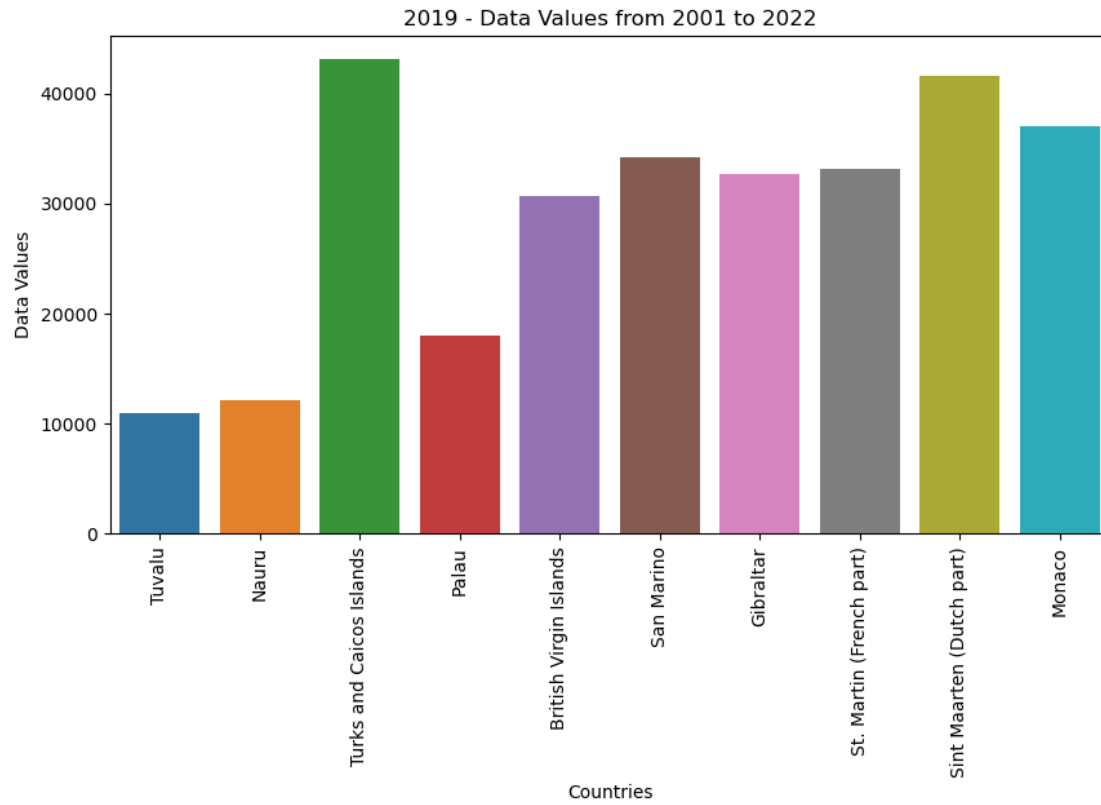


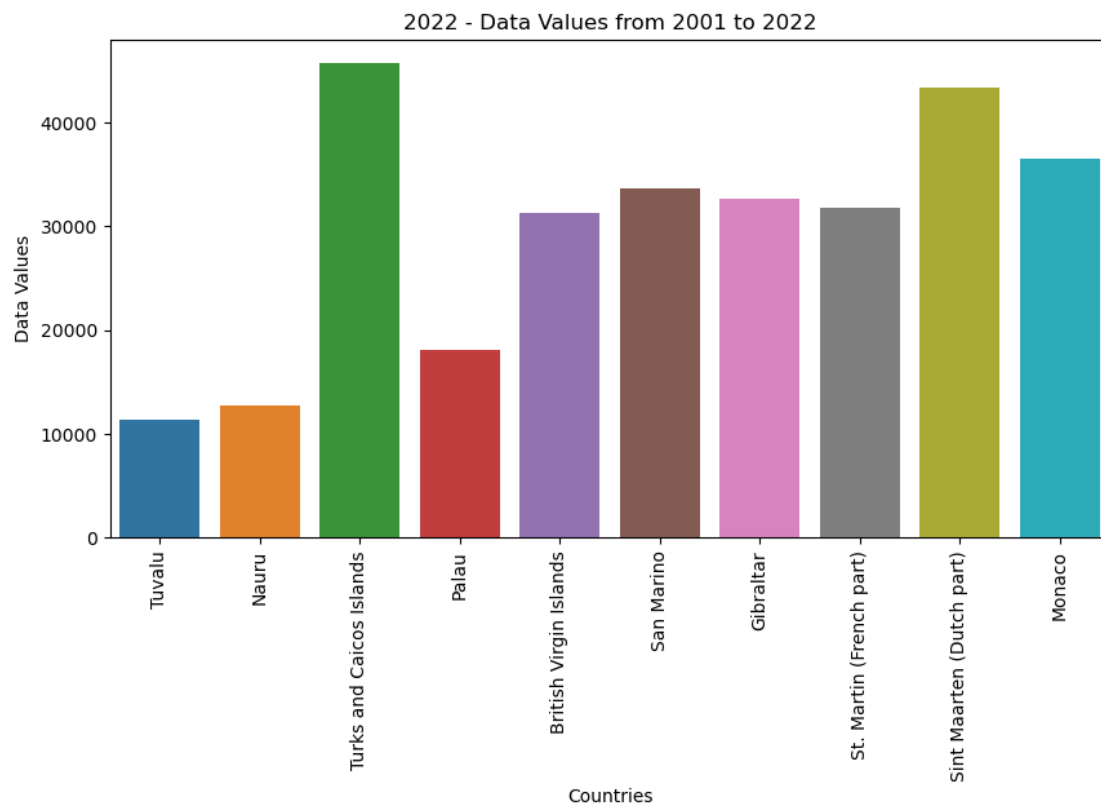
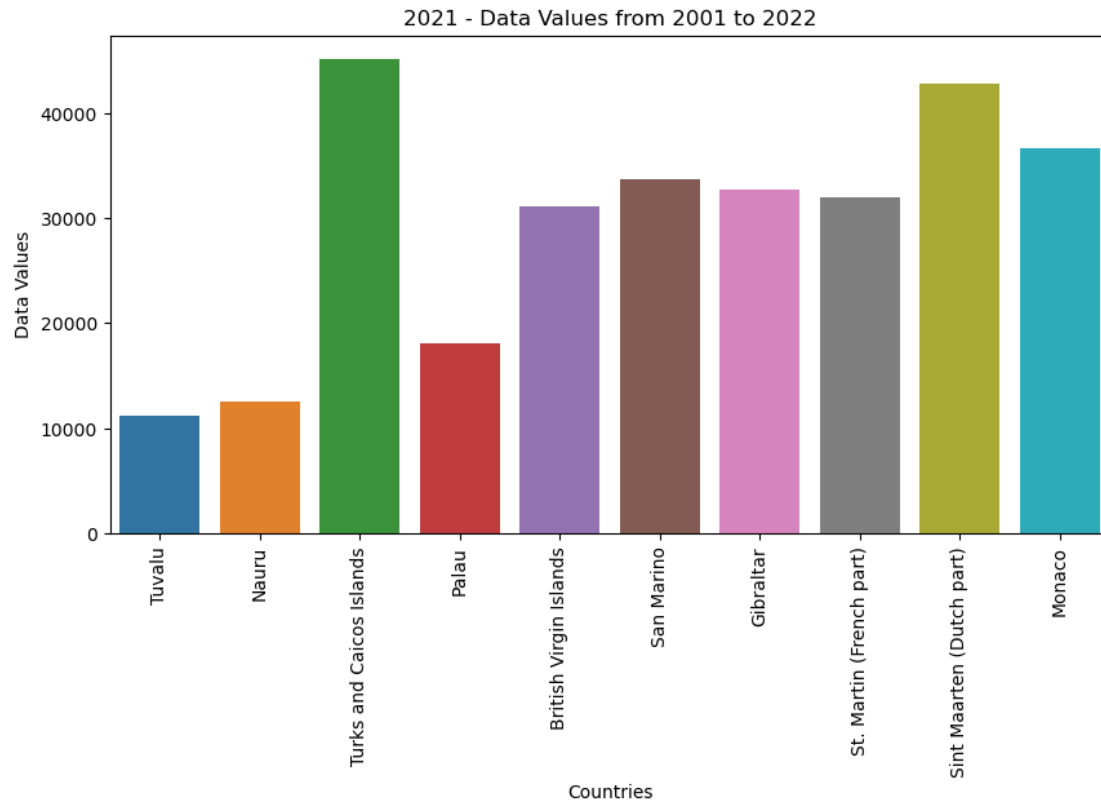












## Insights :

The Name of the country whose indicator is highest year wise with reference to base\_year 2001

1. 2001 & 2002 = Monaco
1. 2003 = St.Martin (French part) & Monaco both are having highest indicator
2. 2004 - 2012 = St. Martin (French part)
3. 2013 - 2017 = St. Martin (Dutch part)
4. 2018 - 2022 = Turks and Caicos Islands

Plotting Indicators of top 10 countries for different years by the base\_year 2001

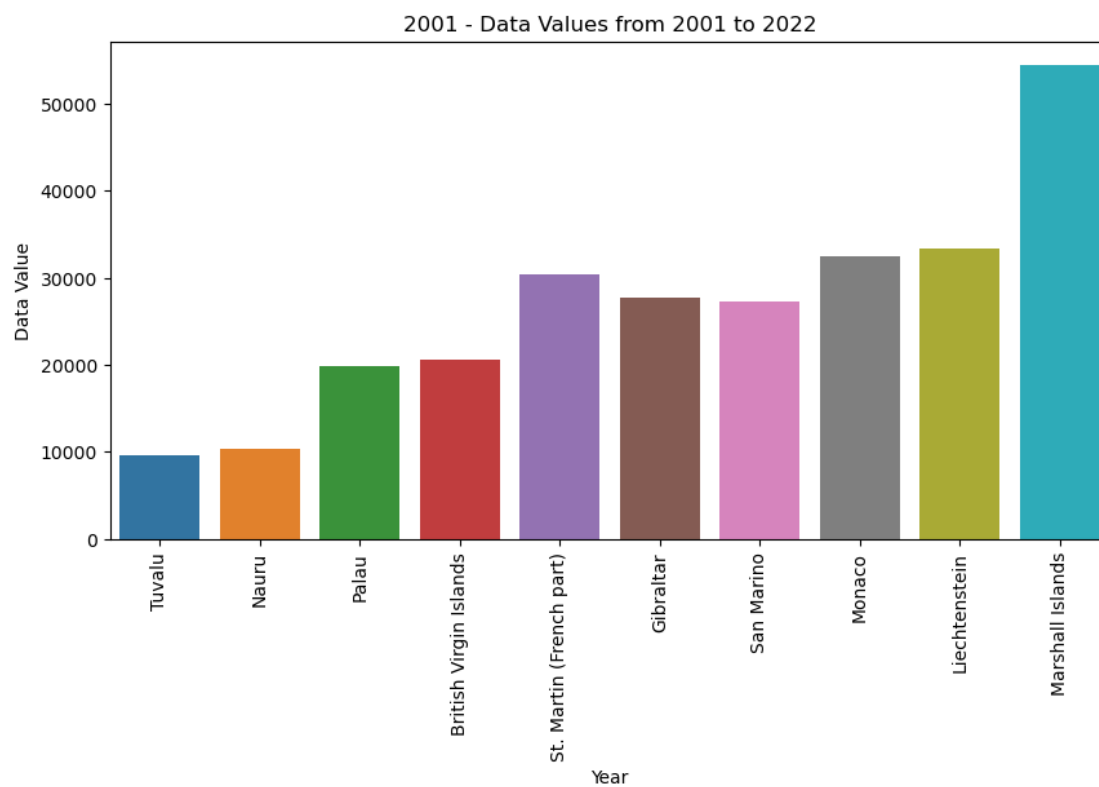
```
country_by_2022 = df.sort_values(by='2022').head(10)
country_by_2022
```

	Country Name	2001	2002	2003	2004	2005	\			
245	Tuvalu	9621.0	9609.0	9668.0	9791.0	9912.0				
179	Nauru	10363.0	10351.0	10344.0	10335.0	10318.0				
188	Palau	19828.0	19851.0	19880.0	19907.0	19831.0				
255	British Virgin Islands	20657.0	21288.0	21982.0	22715.0	23497.0				
147	St. Martin (French part)	30387.0	31160.0	31929.0	32697.0	33452.0				
84	Gibraltar	27721.0	27892.0	28301.0	28716.0	29155.0				
212	San Marino	27335.0	27969.0	28601.0	29093.0	29508.0				
149	Monaco	32444.0	32386.0	32316.0	32236.0	32141.0				
137	Liechtenstein	33376.0	33693.0	34000.0	34300.0	34603.0				
155	Marshall Islands	54413.0	54496.0	54493.0	54435.0	54337.0				
		2006	2007	2008	2009	...	2013	2014	2015	\
245		10030.0	10149.0	10272.0	10408.0	...	10918.0	10899.0	10877.0	
179		10294.0	10267.0	10243.0	10233.0	...	10694.0	10940.0	11185.0	
188		19619.0	19366.0	19102.0	18826.0	...	17805.0	17796.0	17794.0	
255		24323.0	25191.0	26115.0	27044.0	...	28657.0	28971.0	29366.0	
147		34183.0	34887.0	35541.0	36132.0	...	35639.0	35261.0	35020.0	
84		29587.0	29996.0	30398.0	30819.0	...	32411.0	32452.0	32520.0	
212		29959.0	30372.0	30700.0	31059.0	...	33285.0	33389.0	33570.0	
149		32011.0	31823.0	31862.0	32401.0	...	35425.0	36110.0	36760.0	
137		34889.0	35150.0	35401.0	35675.0	...	36806.0	37096.0	37355.0	
155		54208.0	54038.0	53816.0	53593.0	...	51352.0	50419.0	49410.0	
		2016	2017	2018	2019	2020	2021	2022		
245		10852.0	10828.0	10865.0	10956.0	11069.0	11204.0	11312.0		
179		11437.0	11682.0	11924.0	12132.0	12315.0	12511.0	12668.0		
188		17816.0	17837.0	17864.0	17916.0	17972.0	18024.0	18055.0		
255		29739.0	30060.0	30335.0	30610.0	30910.0	31122.0	31305.0		
147		34811.0	34496.0	33852.0	33121.0	32553.0	31948.0	31791.0		
84		32565.0	32602.0	32648.0	32685.0	32709.0	32669.0	32649.0		
212		33834.0	34056.0	34156.0	34178.0	34007.0	33745.0	33660.0		
149		37071.0	37044.0	37029.0	37034.0	36922.0	36686.0	36469.0		
137		37609.0	37889.0	38181.0	38482.0	38756.0	39039.0	39327.0		

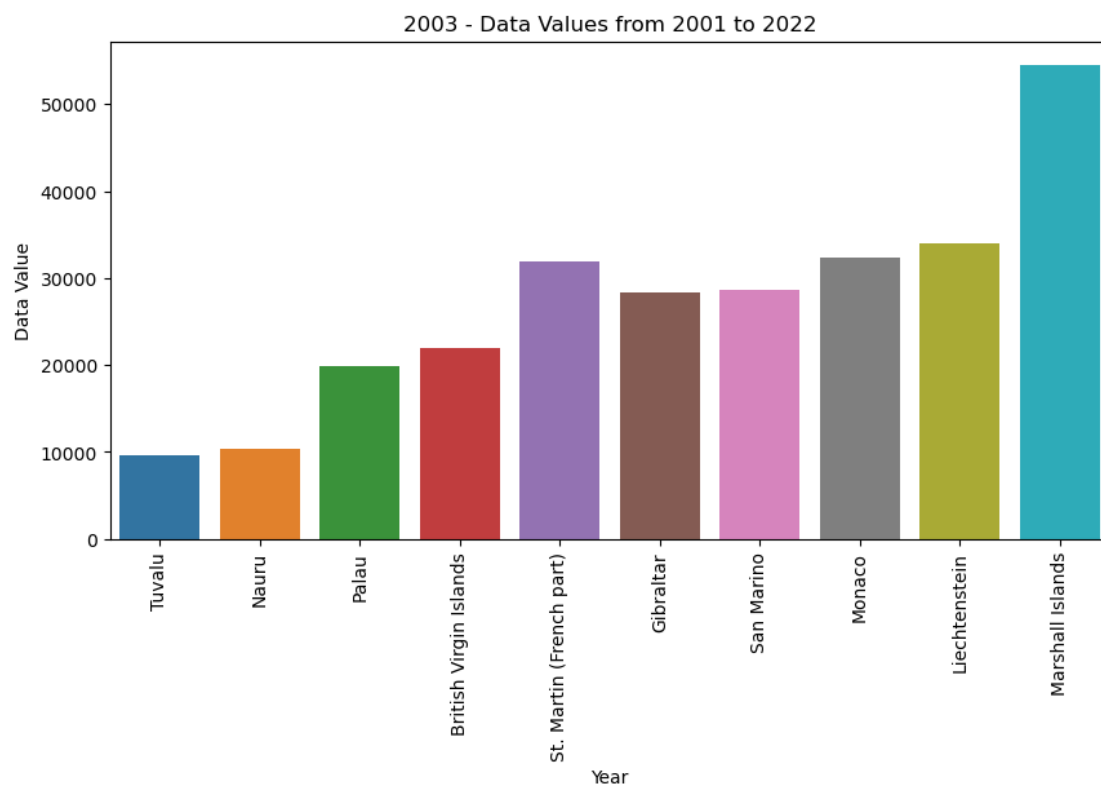
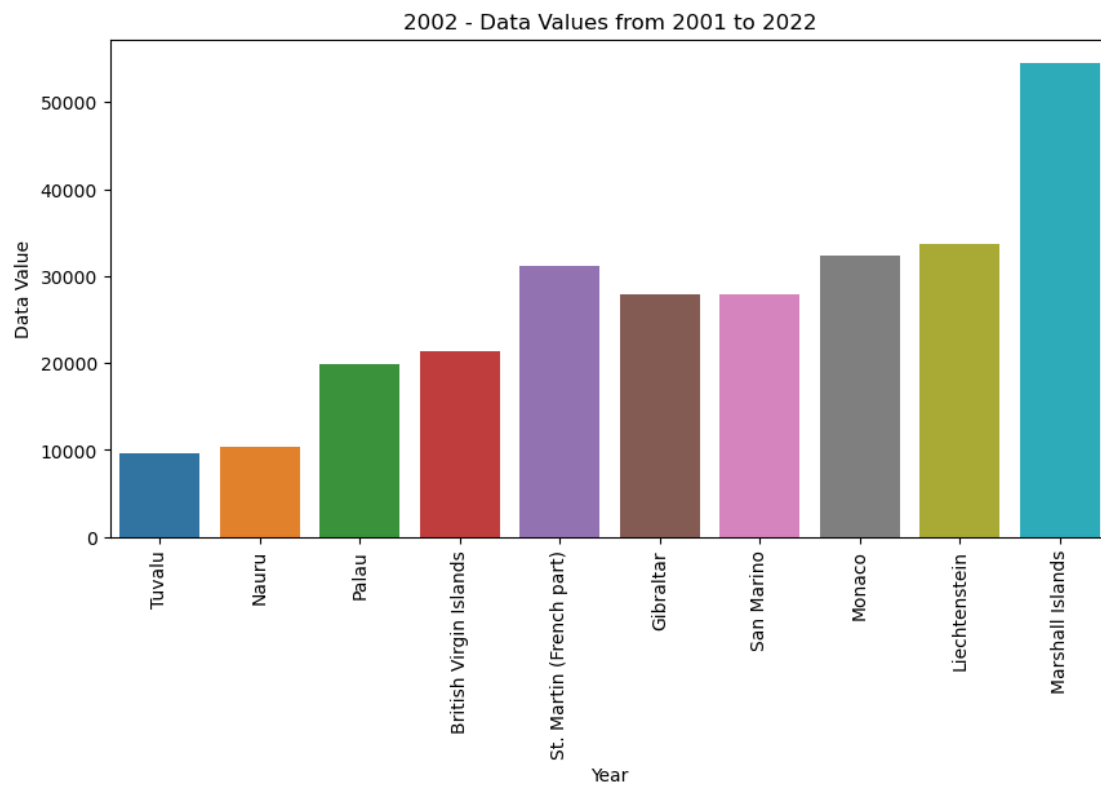
```
155 48329.0 47187.0 45989.0 44728.0 43413.0 42050.0 41569.0
```

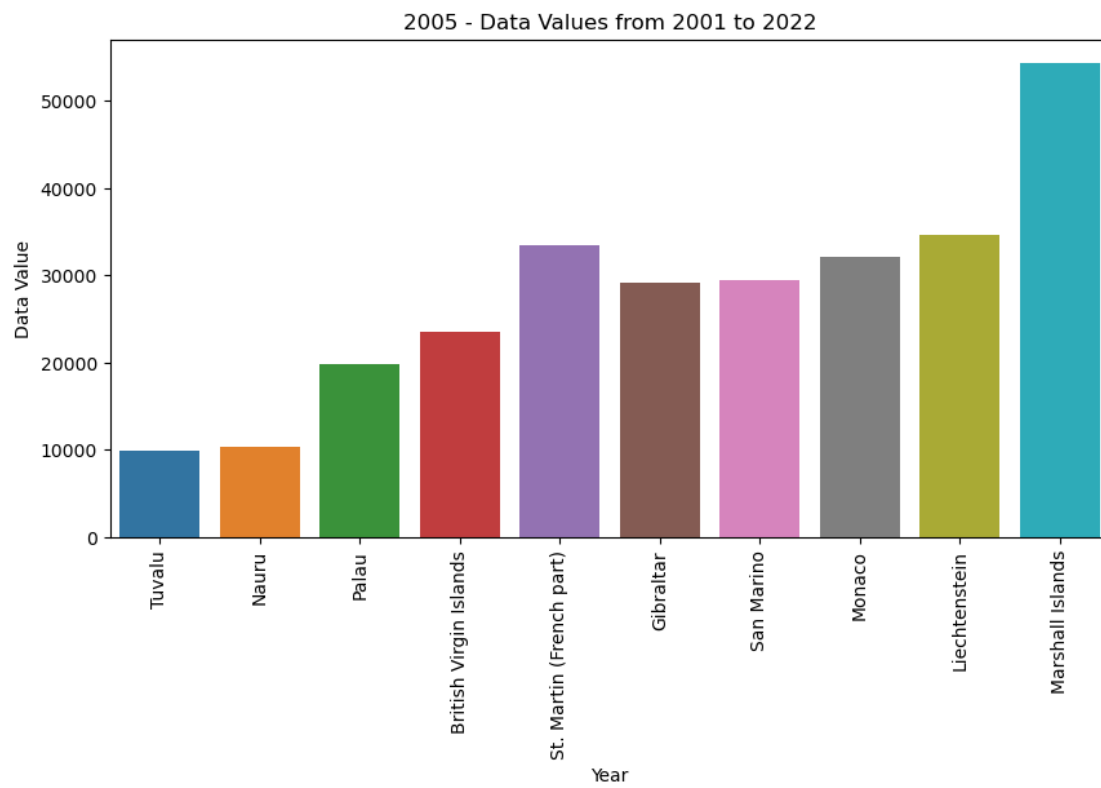
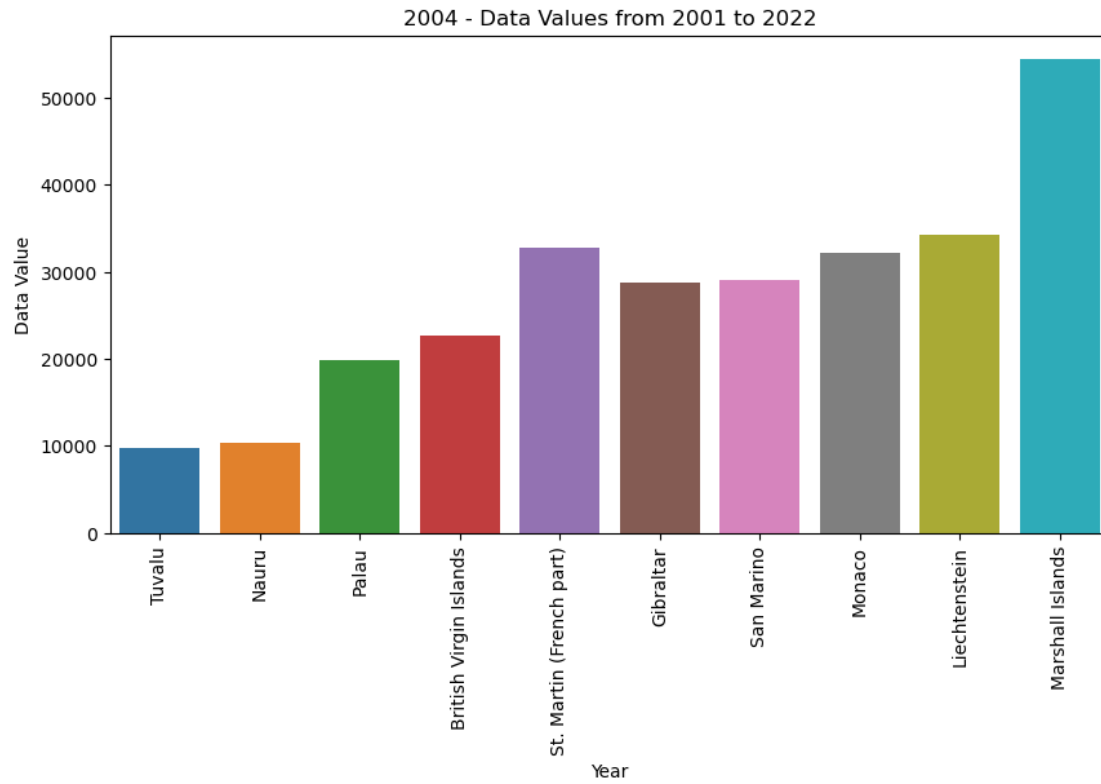
```
[10 rows x 23 columns]
```

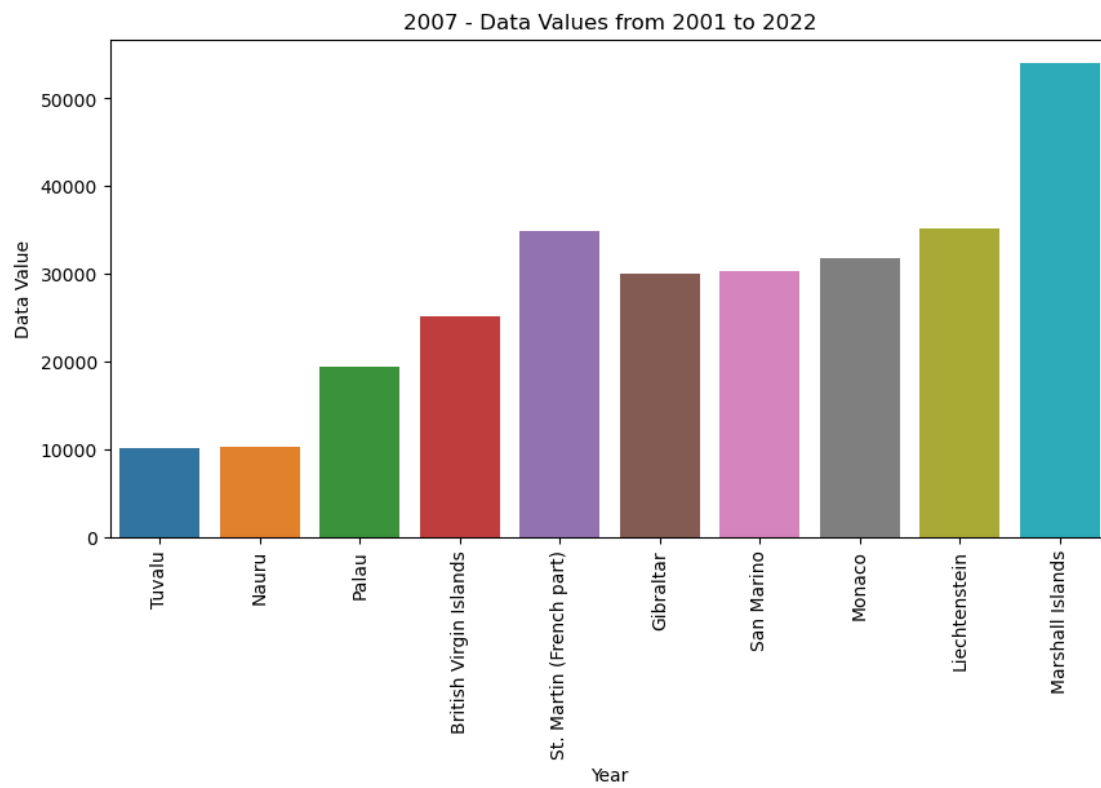
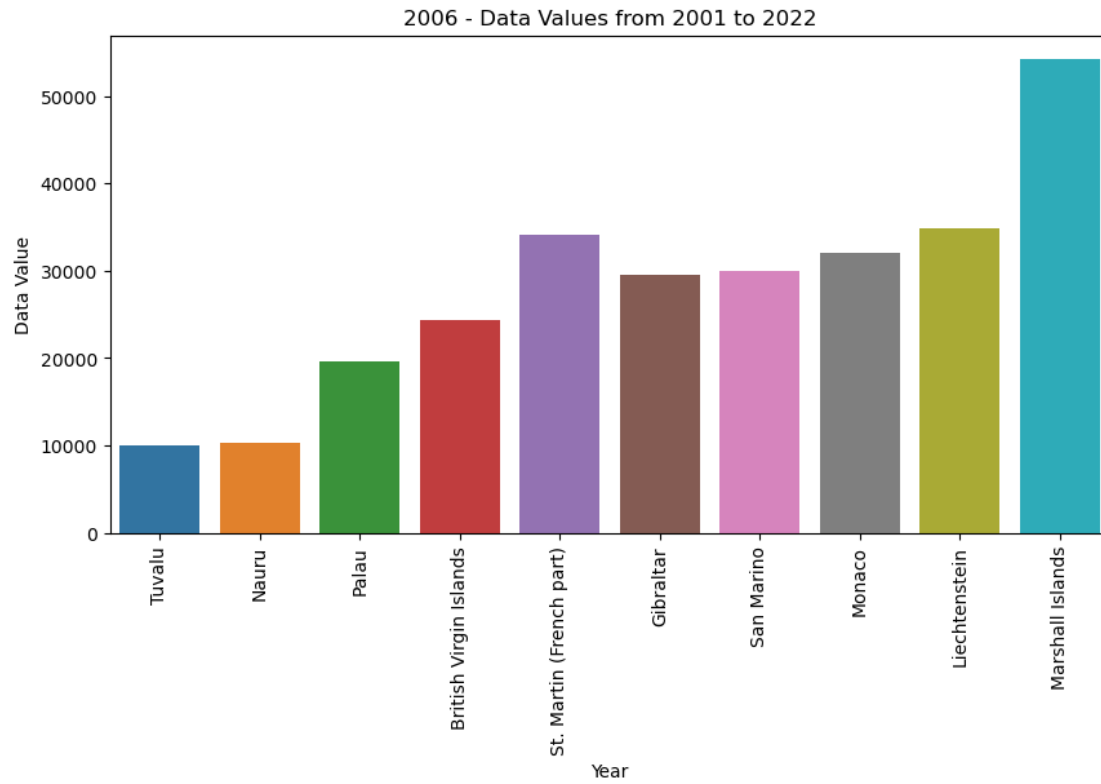
```
country_by_2022_t = country_by_2022.set_index('Country Name').T
for country_name, data_values in country_by_2022_t.iterrows():
    fig = plt.figure(figsize=(10, 5))
    sns.barplot(x=data_values.index, y=data_values.values)
    plt.xlabel('Year')
    plt.ylabel('Data Value')
    plt.title(f"{country_name} - Data Values from 2001 to 2022")
    plt.xticks(rotation=90)
    plt.show()
```

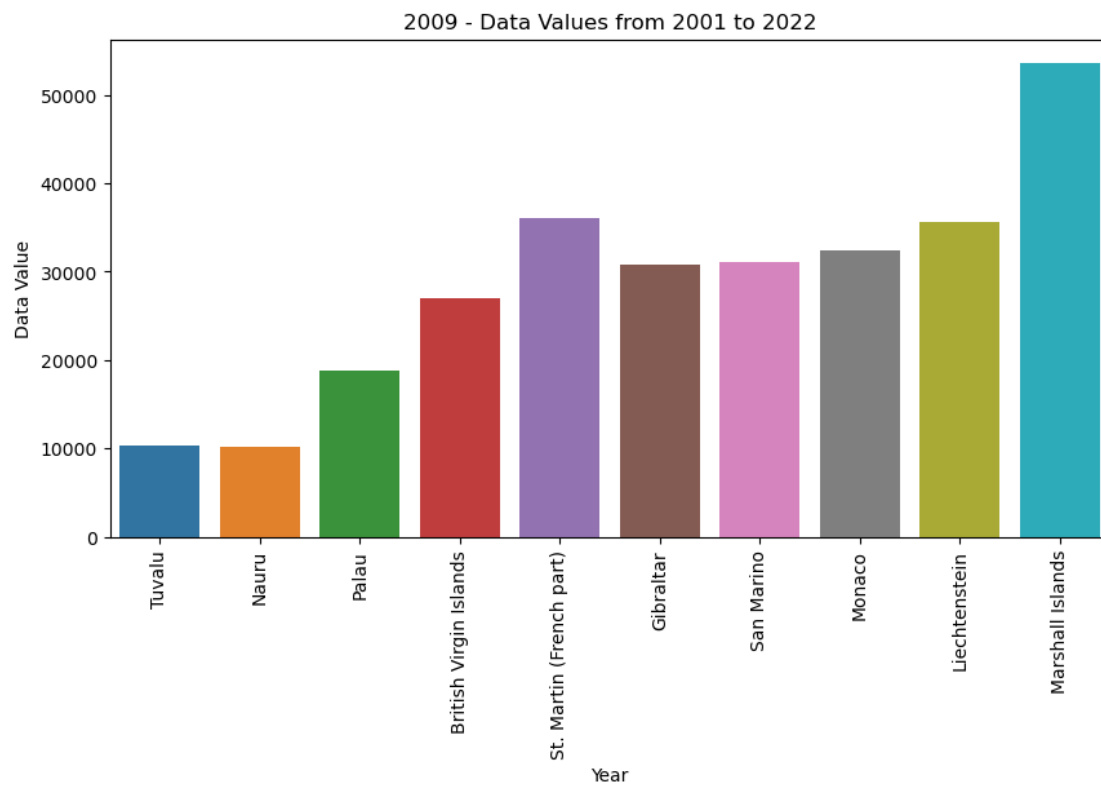
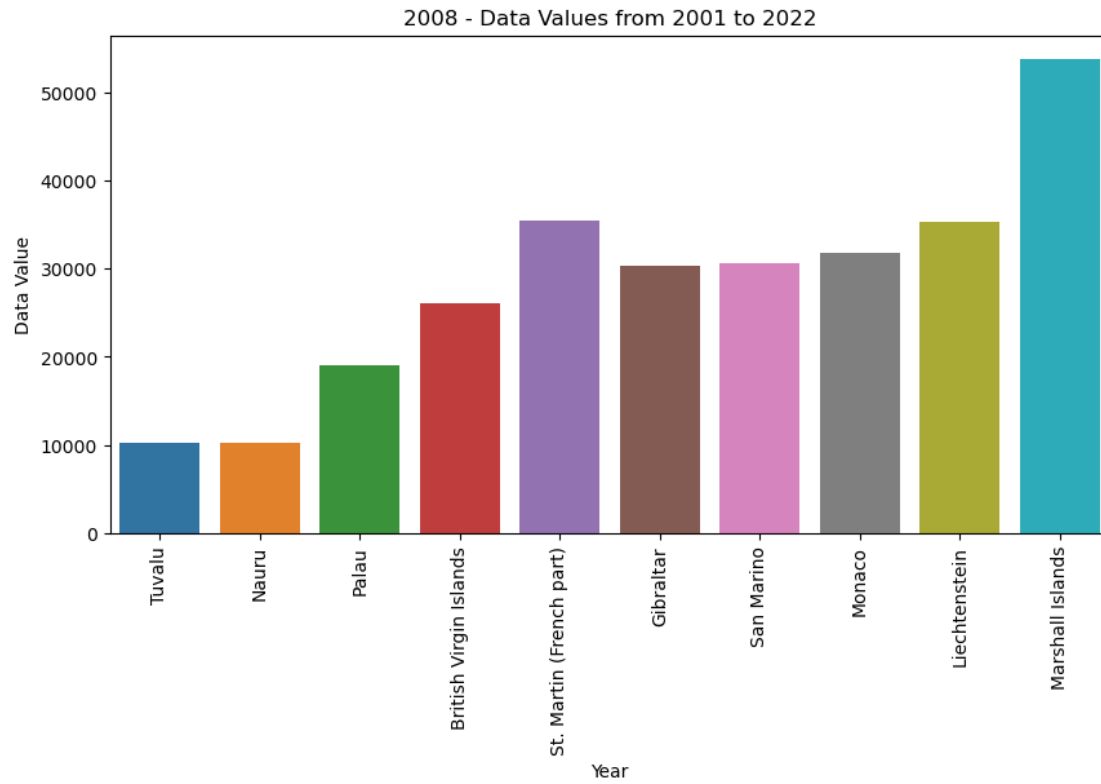


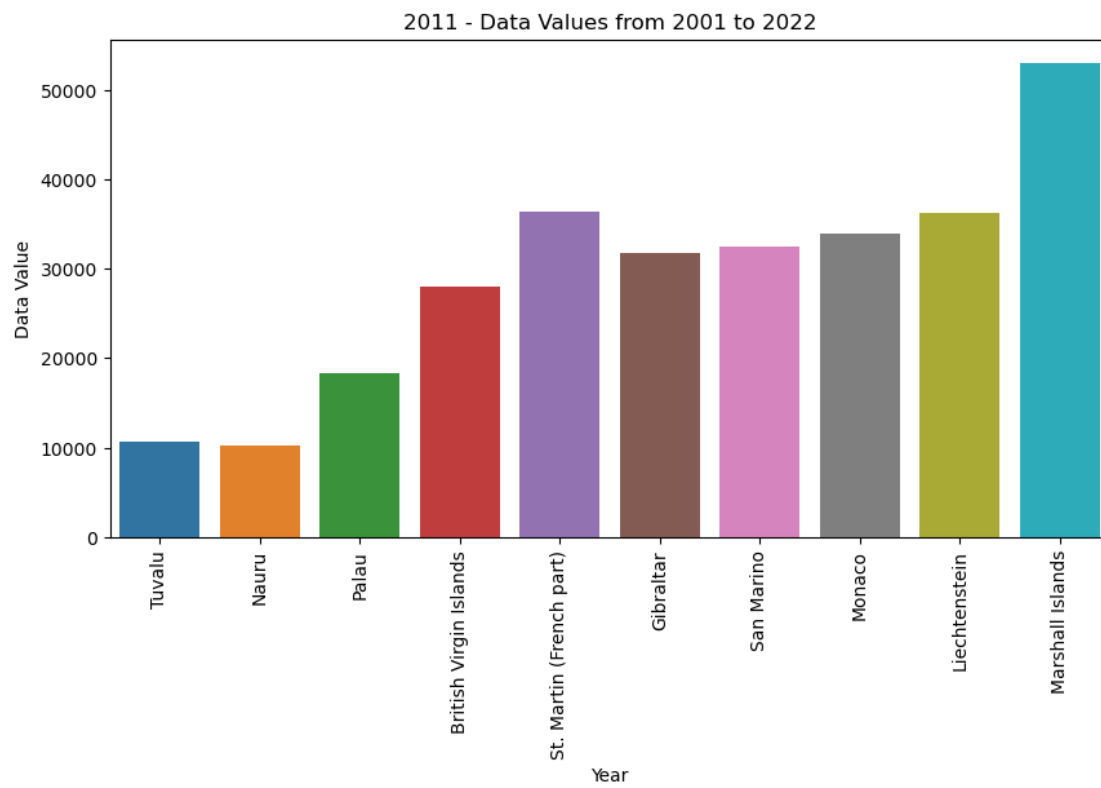
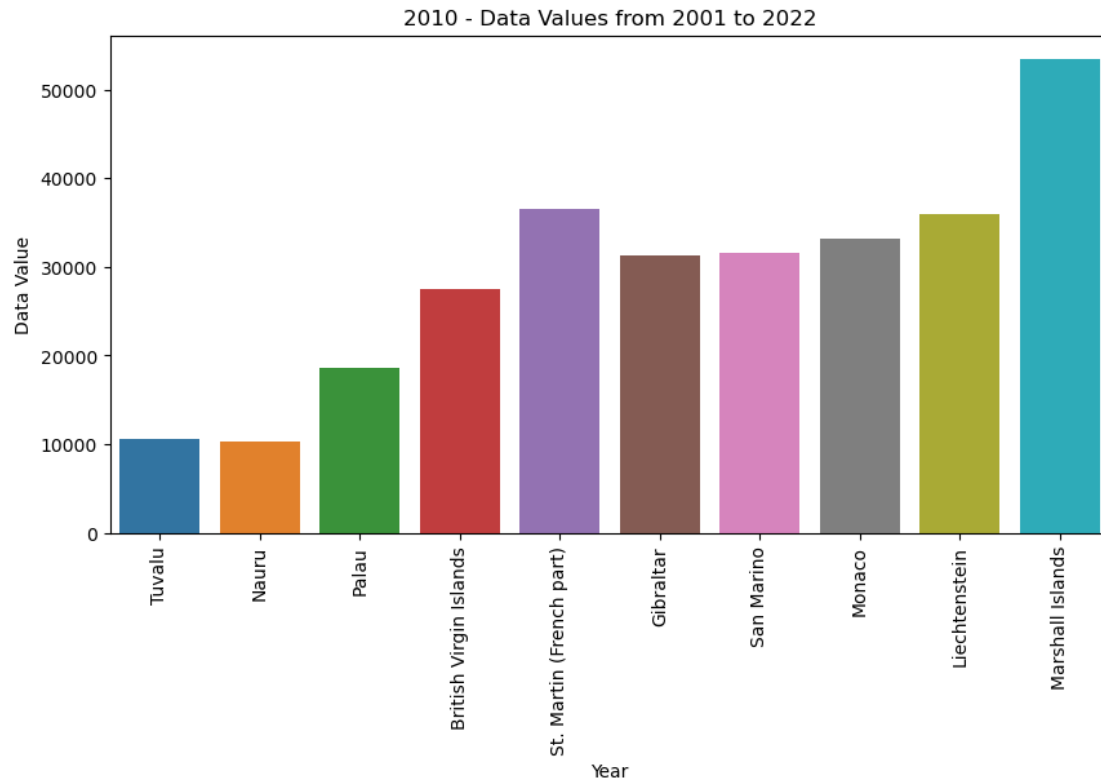


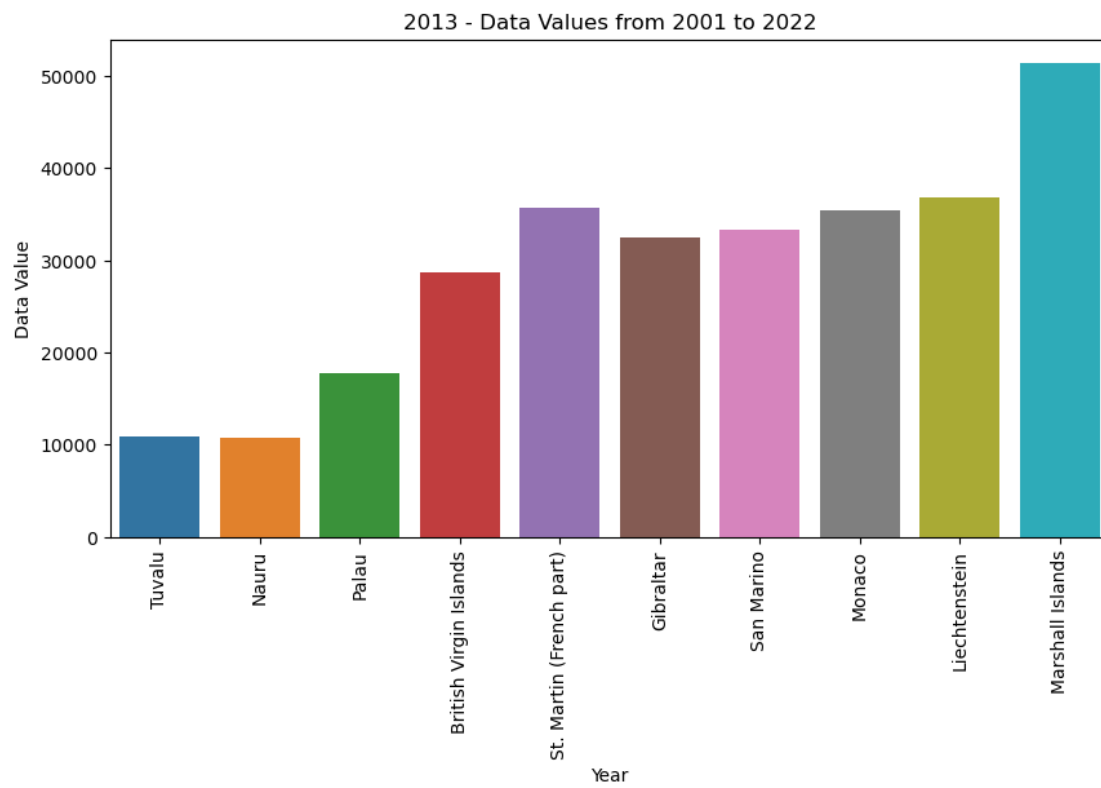
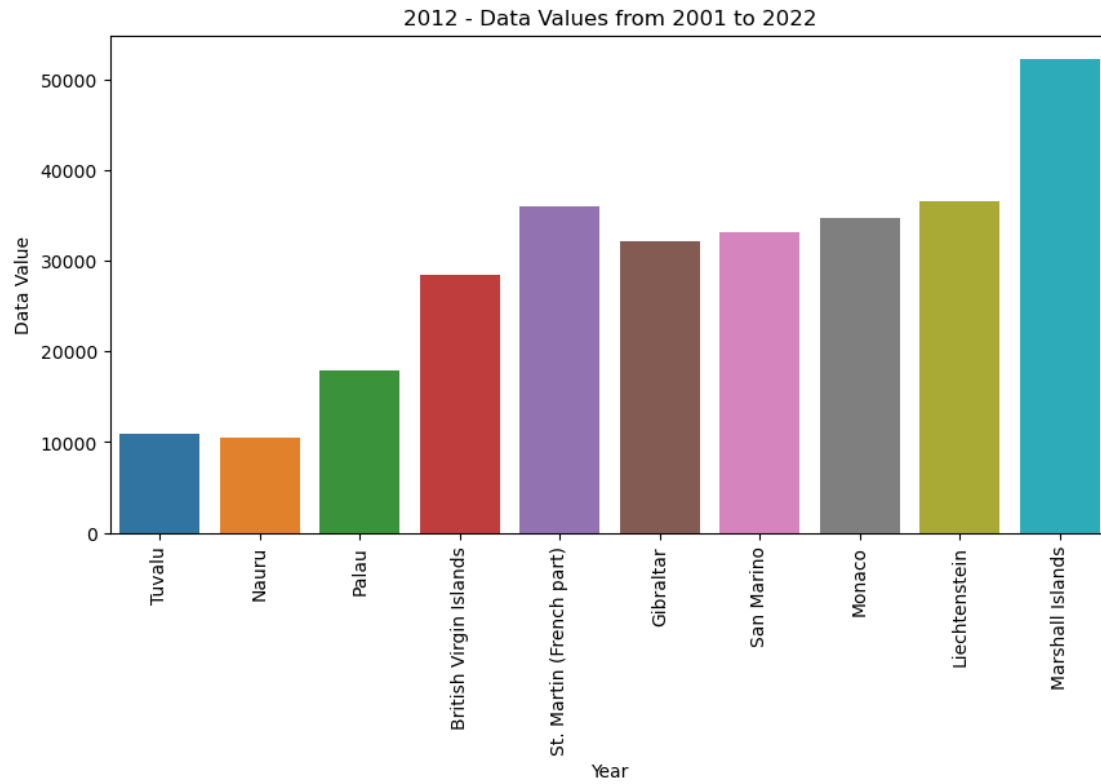


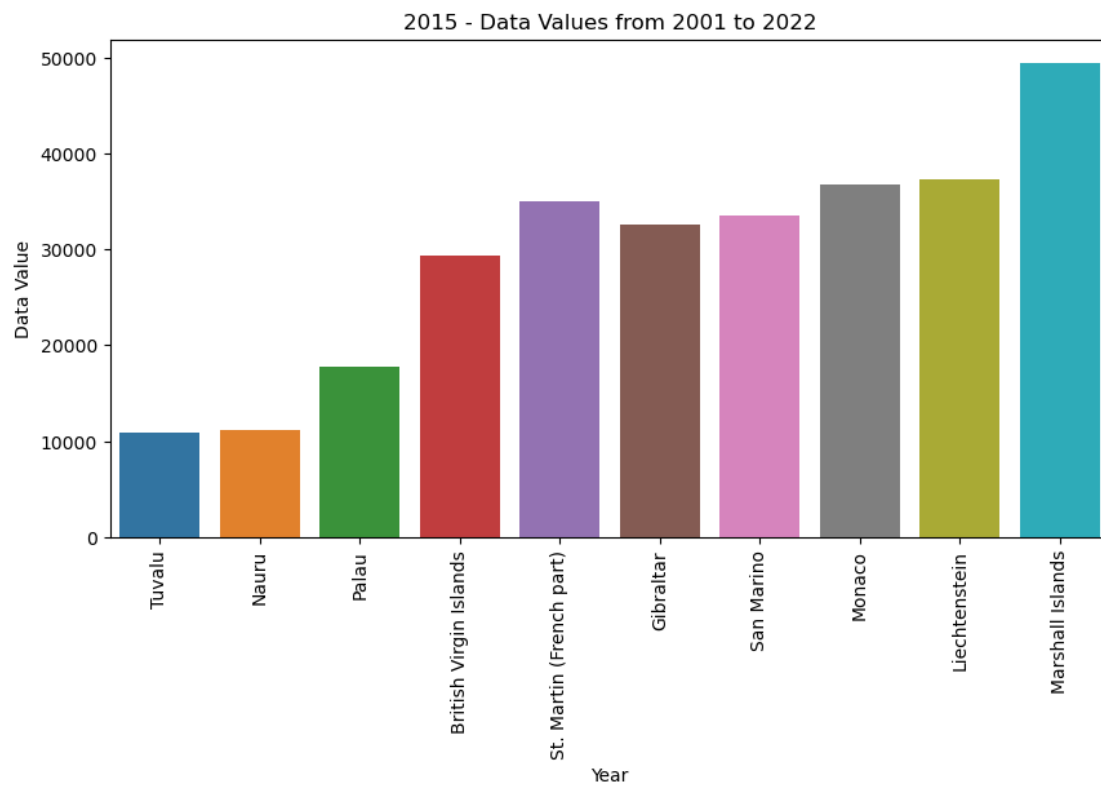
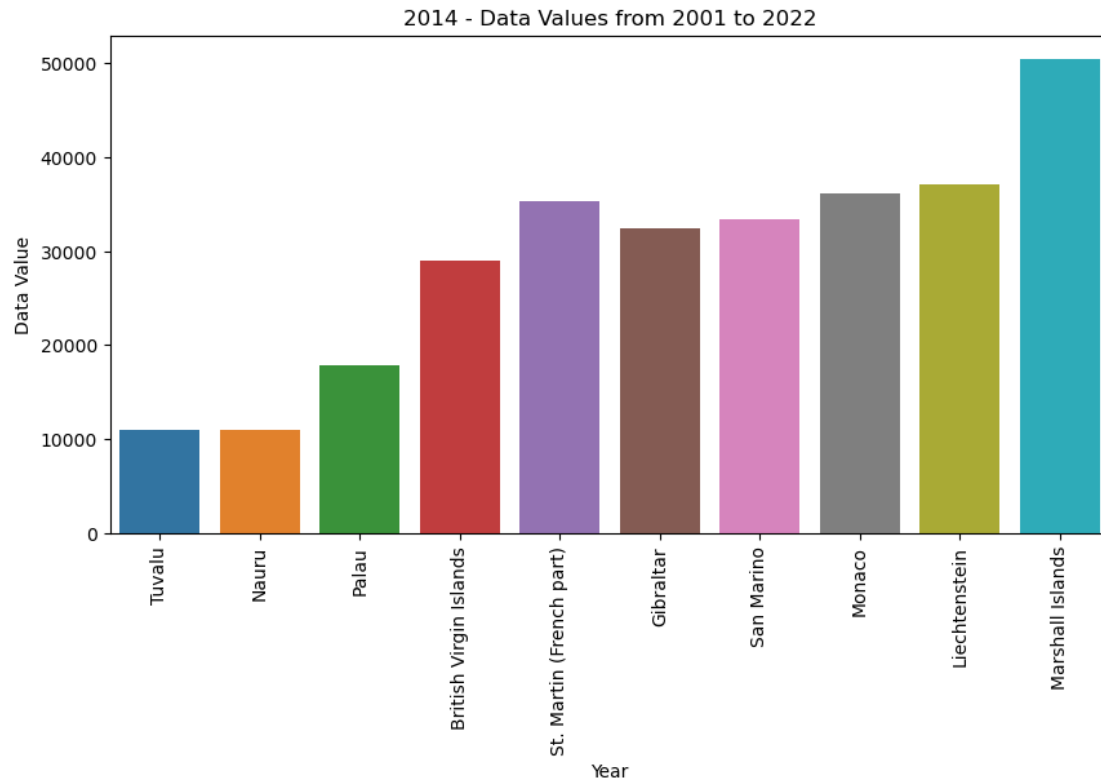


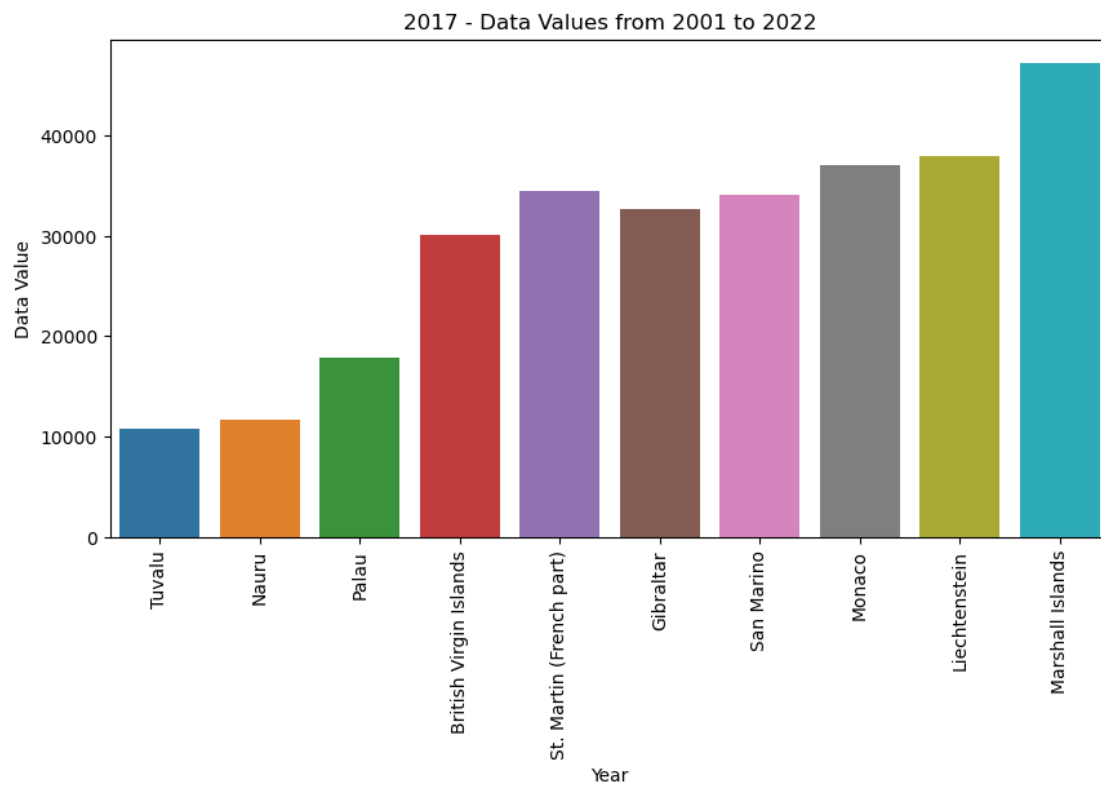
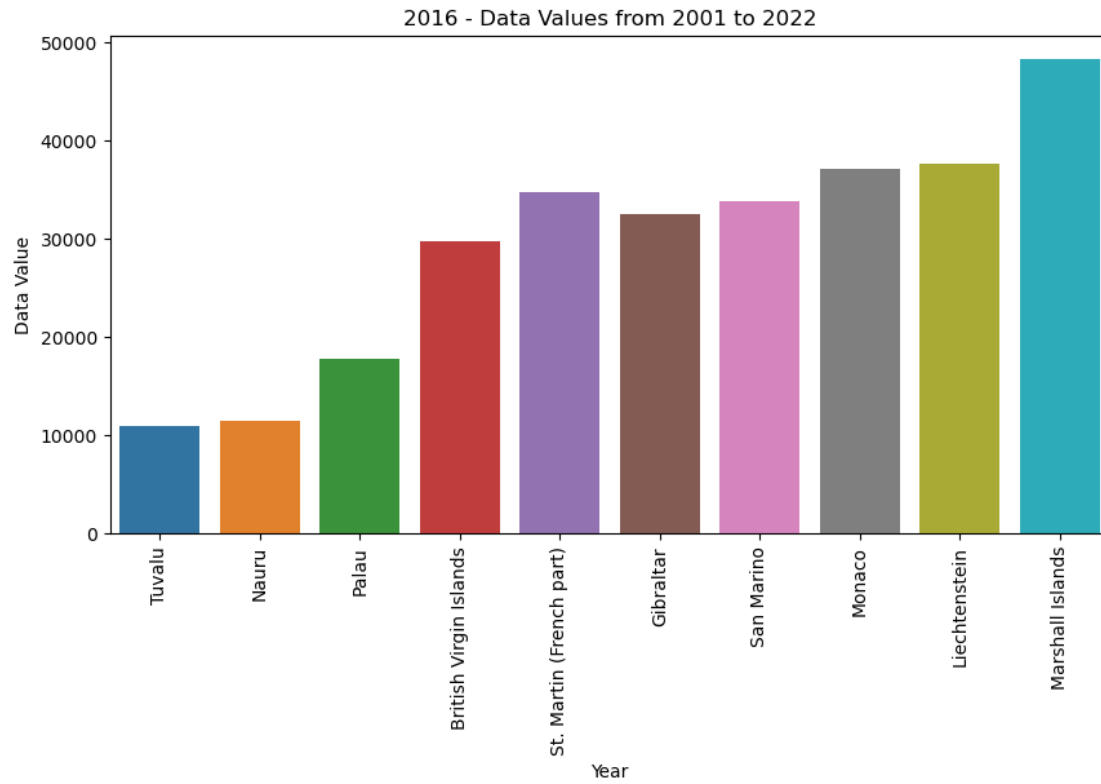




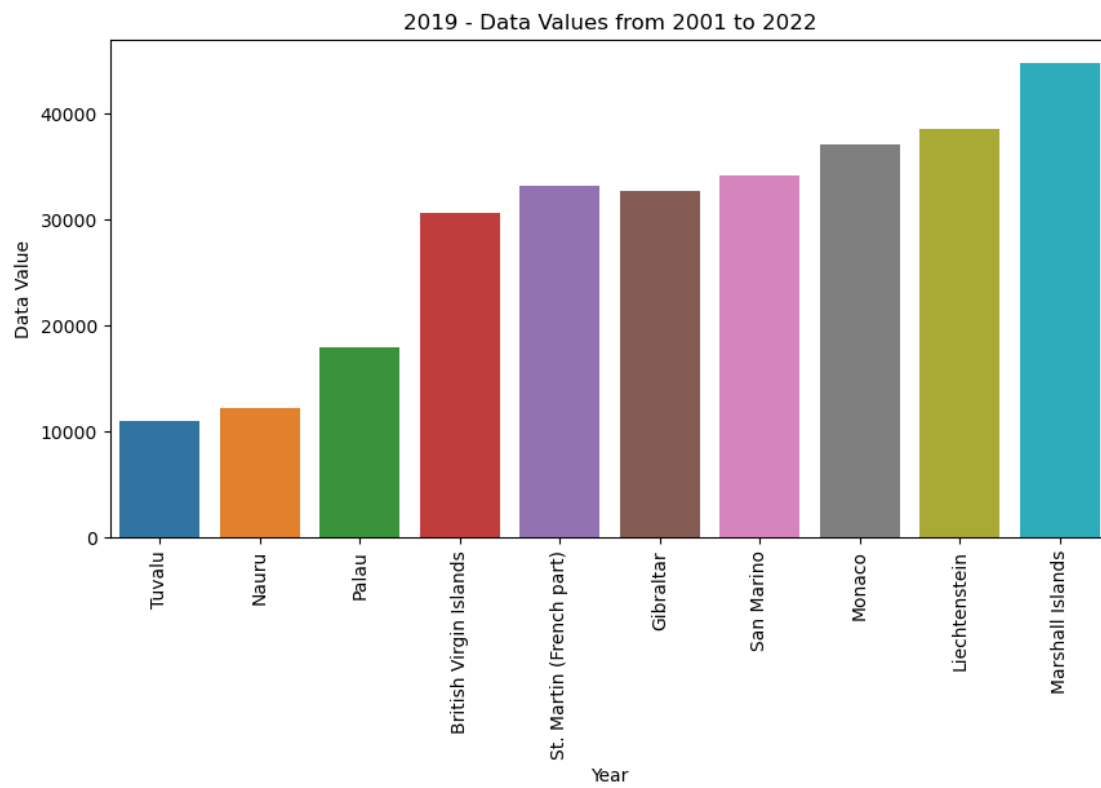
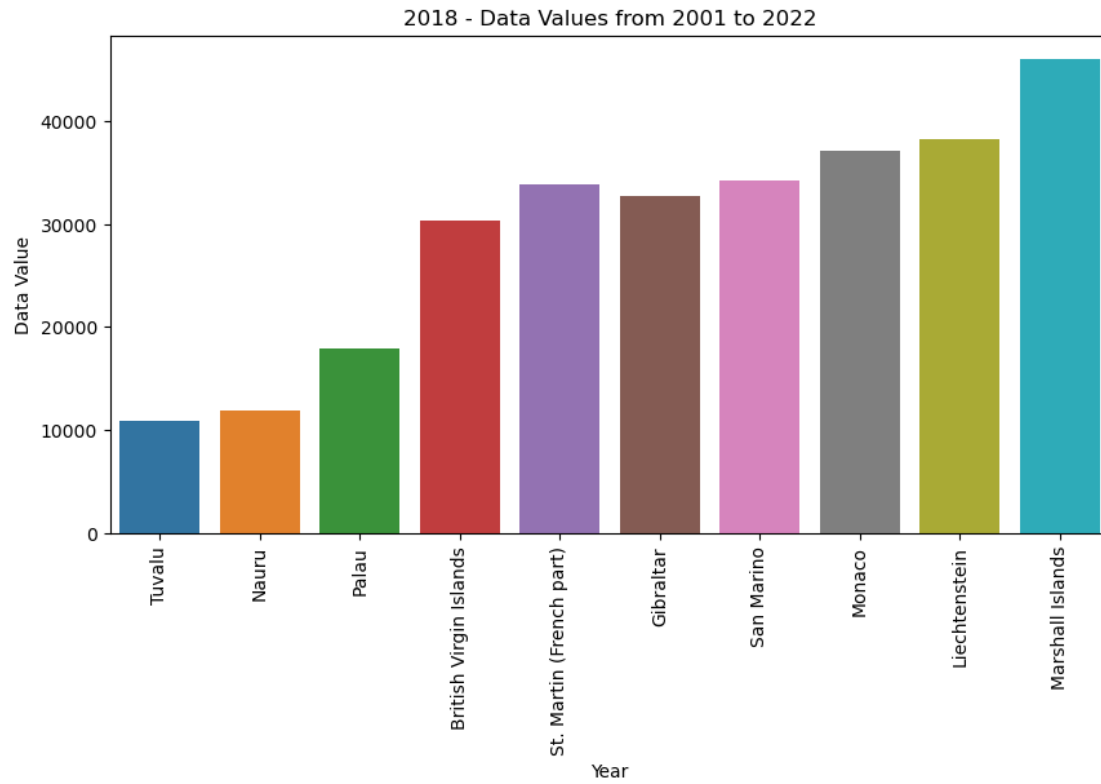


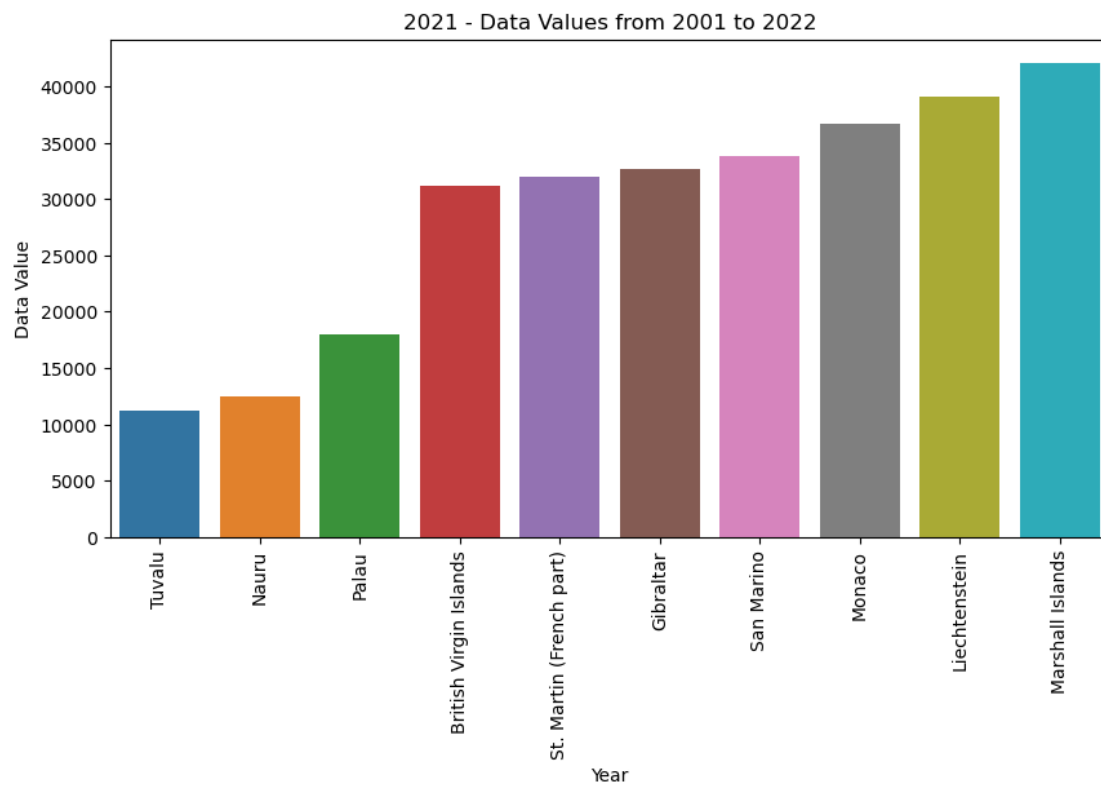
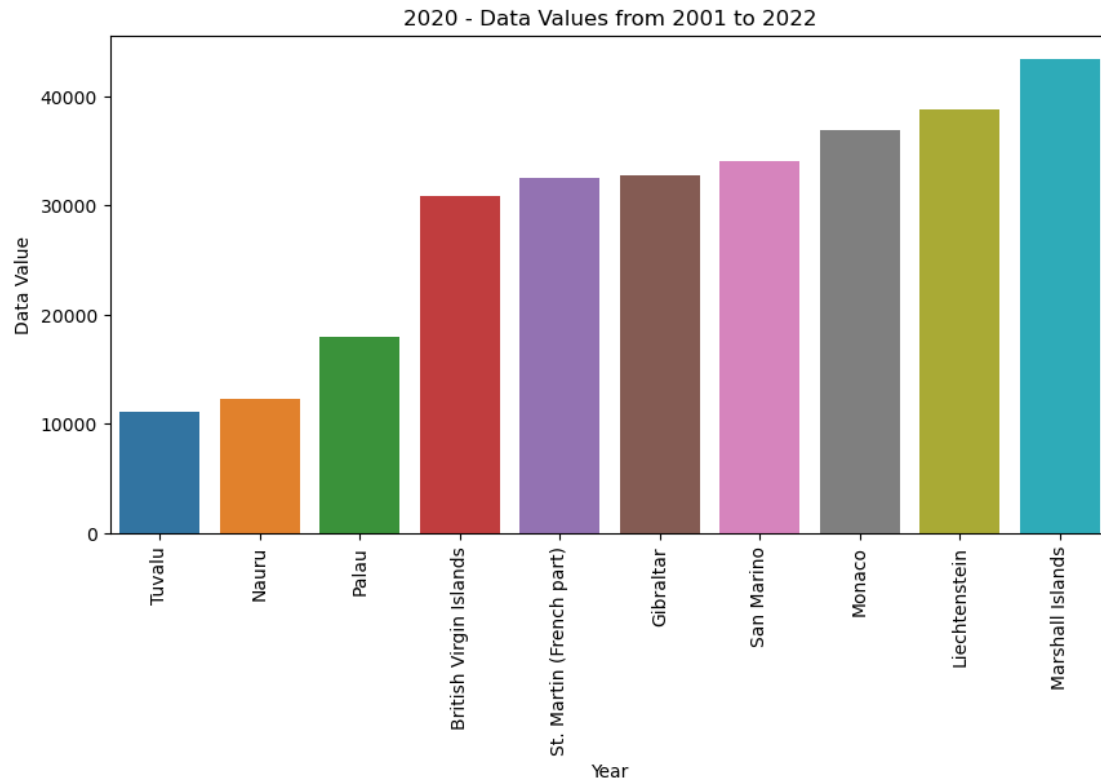


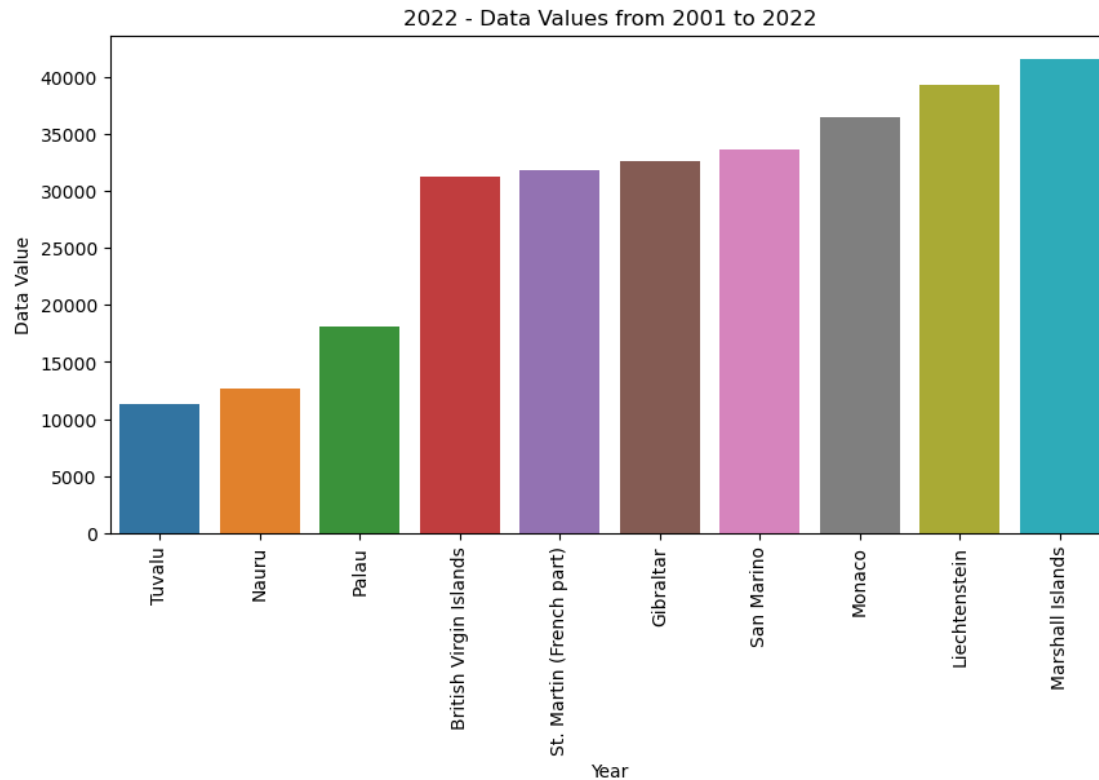












## Insights :

### The Name of the country whose indicator is highest year wise with reference to base\_year 2022

1. In all the years(2001-2022) Marshall Islands Country is having the highest indicator value.
1. There are many other countries who are having the good indicator value -
  - 2.1. In the Initial years Liechtenstein, Monaco and St. Martin(French Part)
  - 2.2. In the Late years Gibraltar and San Marino (other than mentioned above)

## Conclusion

In this Project we have used World Development Indicator Dataset which represents indicator values of different countries from 2001 to 2022. we have seen that with the coming of next year the values of indicator keep rising. Also I have once used 2001 as a base year where I have seen, Monaco named Country having the highest indicator in the initial years and in the last years Turks and Caicos Islands Country having the highest Indicator. Whereas , when 2022 is considered as a base year t Marshall Islands is having the highest indicator value. These insights shows with the change in the years the indicator values of different countries changes rapidly and countries also fluctuates in terms of highest indicator.

