

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

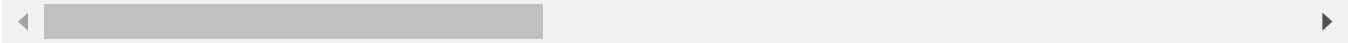
```
In [33]: df = pd.read_csv(r"C:\Users\Sarthak Sarkar\Downloads\API_SP.POP.TOTL_DS2_en_csv_v2_
```

```
In [34]: df
```

Out[34]:

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	
0	Aruba	ABW	Population, total	SP.POP.TOTL	54608.0	55811.0	56682.0	57
1	Africa Eastern and Southern	AFE	Population, total	SP.POP.TOTL	130692579.0	134169237.0	137835590.0	141630
2	Afghanistan	AFG	Population, total	SP.POP.TOTL	8622466.0	8790140.0	8969047.0	9157
3	Africa Western and Central	AFW	Population, total	SP.POP.TOTL	97256290.0	99314028.0	101445032.0	103667
4	Angola	AGO	Population, total	SP.POP.TOTL	5357195.0	5441333.0	5521400.0	5599
...
261	Kosovo	XKX	Population, total	SP.POP.TOTL	947000.0	966000.0	994000.0	1022
262	Yemen, Rep.	YEM	Population, total	SP.POP.TOTL	5542459.0	5646668.0	5753386.0	5860
263	South Africa	ZAF	Population, total	SP.POP.TOTL	16520441.0	16989464.0	17503133.0	18042
264	Zambia	ZMB	Population, total	SP.POP.TOTL	3119430.0	3219451.0	3323427.0	3431
265	Zimbabwe	ZWE	Population, total	SP.POP.TOTL	3806310.0	3925952.0	4049778.0	4177

266 rows × 67 columns



```
In [35]: df.head()
```

Out[35]:

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963
0	Aruba	ABW	Population, total	SP.POP.TOTL	54608.0	55811.0	56682.0	5747
1	Africa Eastern and Southern	AFE	Population, total	SP.POP.TOTL	130692579.0	134169237.0	137835590.0	14163054
2	Afghanistan	AFG	Population, total	SP.POP.TOTL	8622466.0	8790140.0	8969047.0	915746
3	Africa Western and Central	AFW	Population, total	SP.POP.TOTL	97256290.0	99314028.0	101445032.0	10366751
4	Angola	AGO	Population, total	SP.POP.TOTL	5357195.0	5441333.0	5521400.0	559982

5 rows × 67 columns

In [36]: df.tail()

Out[36]:

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963
261	Kosovo	KKX	Population, total	SP.POP.TOTL	947000.0	966000.0	994000.0	1022000.0
262	Yemen, Rep.	YEM	Population, total	SP.POP.TOTL	5542459.0	5646668.0	5753386.0	5860197.0
263	South Africa	ZAF	Population, total	SP.POP.TOTL	16520441.0	16989464.0	17503133.0	18042215.0
264	Zambia	ZMB	Population, total	SP.POP.TOTL	3119430.0	3219451.0	3323427.0	3431381.0
265	Zimbabwe	ZWE	Population, total	SP.POP.TOTL	3806310.0	3925952.0	4049778.0	4177931.0

5 rows × 67 columns

In [38]: df.shape

Out[38]: (266, 67)

In [39]: df.columns

Out[39]:

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'], dtype='object')

In [40]: df.dtypes

```
Out[40]: Country Name      object
Country Code      object
Indicator Name     object
Indicator Code     object
1960               float64
...
2018               float64
2019               float64
2020               float64
2021               float64
2022               float64
Length: 67, dtype: object
```

```
In [41]: df.info()
```

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

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

```
Data columns (total 67 columns):
```

#	Column	Non-Null Count	Dtype
0	Country Name	266 non-null	object
1	Country Code	266 non-null	object
2	Indicator Name	266 non-null	object
3	Indicator Code	266 non-null	object
4	1960	264 non-null	float64
5	1961	264 non-null	float64
6	1962	264 non-null	float64
7	1963	264 non-null	float64
8	1964	264 non-null	float64
9	1965	264 non-null	float64
10	1966	264 non-null	float64
11	1967	264 non-null	float64
12	1968	264 non-null	float64
13	1969	264 non-null	float64
14	1970	264 non-null	float64
15	1971	264 non-null	float64
16	1972	264 non-null	float64
17	1973	264 non-null	float64
18	1974	264 non-null	float64
19	1975	264 non-null	float64
20	1976	264 non-null	float64
21	1977	264 non-null	float64
22	1978	264 non-null	float64
23	1979	264 non-null	float64
24	1980	264 non-null	float64
25	1981	264 non-null	float64
26	1982	264 non-null	float64
27	1983	264 non-null	float64
28	1984	264 non-null	float64
29	1985	264 non-null	float64
30	1986	264 non-null	float64
31	1987	264 non-null	float64
32	1988	264 non-null	float64
33	1989	264 non-null	float64
34	1990	265 non-null	float64
35	1991	265 non-null	float64
36	1992	265 non-null	float64
37	1993	265 non-null	float64
38	1994	265 non-null	float64
39	1995	265 non-null	float64
40	1996	265 non-null	float64
41	1997	265 non-null	float64
42	1998	265 non-null	float64
43	1999	265 non-null	float64
44	2000	265 non-null	float64
45	2001	265 non-null	float64
46	2002	265 non-null	float64
47	2003	265 non-null	float64
48	2004	265 non-null	float64
49	2005	265 non-null	float64
50	2006	265 non-null	float64
51	2007	265 non-null	float64
52	2008	265 non-null	float64
53	2009	265 non-null	float64
54	2010	265 non-null	float64
55	2011	265 non-null	float64
56	2012	265 non-null	float64
57	2013	265 non-null	float64
58	2014	265 non-null	float64

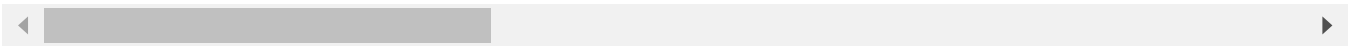
```
59 2015      265 non-null float64
60 2016      265 non-null float64
61 2017      265 non-null float64
62 2018      265 non-null float64
63 2019      265 non-null float64
64 2020      265 non-null float64
65 2021      265 non-null float64
66 2022      265 non-null float64
dtypes: float64(63), object(4)
memory usage: 139.4+ KB
```

```
In [42]: df.describe()
```

Out[42]:

	1960	1961	1962	1963	1964	1965	1966
count	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02
mean	1.172712e+08	1.188807e+08	1.210511e+08	1.237333e+08	1.264378e+08	1.291813e+08	1.320000e+08
std	3.695439e+08	3.740897e+08	3.808061e+08	3.895039e+08	3.982439e+08	4.071153e+08	4.160000e+08
min	2.646000e+03	2.888000e+03	3.171000e+03	3.481000e+03	3.811000e+03	4.161000e+03	4.530000e+03
25%	5.132212e+05	5.231345e+05	5.337595e+05	5.449288e+05	5.566630e+05	5.651150e+05	5.690000e+05
50%	3.757486e+06	3.887144e+06	4.023896e+06	4.139356e+06	4.224612e+06	4.277636e+06	4.330000e+06
75%	2.670606e+07	2.748694e+07	2.830289e+07	2.914708e+07	3.001684e+07	3.084892e+07	3.160000e+07
max	3.031474e+09	3.072422e+09	3.126850e+09	3.193429e+09	3.260442e+09	3.328209e+09	3.390000e+09

8 rows × 63 columns



```
In [44]: df.duplicated().sum()
```

Out[44]: 0

```
In [45]: df.isna().sum().any()
```

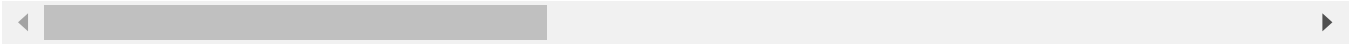
Out[45]: True

```
In [46]: df = df.fillna(method = "ffill")
df.head()
```

Out[46]:

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	19
0	Aruba	ABW	Population, total	SP.POP.TOTL	54608.0	55811.0	56682.0	5747
1	Africa Eastern and Southern	AFE	Population, total	SP.POP.TOTL	130692579.0	134169237.0	137835590.0	14163054
2	Afghanistan	AFG	Population, total	SP.POP.TOTL	8622466.0	8790140.0	8969047.0	915746
3	Africa Western and Central	AFW	Population, total	SP.POP.TOTL	97256290.0	99314028.0	101445032.0	10366751
4	Angola	AGO	Population, total	SP.POP.TOTL	5357195.0	5441333.0	5521400.0	559982

5 rows × 67 columns



In [47]: `df.isna().sum().any()`

Out[47]: `False`

In [48]: `df['Country Name'].unique()`

```

Out[48]: 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)
```

```
In [49]: df['Country Code'].unique()
```

```
Out[49]: 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)
```

```
In [50]: df['Indicator Name'].unique()
```

```
Out[50]: array(['Population, total'], dtype=object)
```

```
In [51]: df['Indicator Code'].unique()
```

```
Out[51]: array(['SP.POP.TOTL'], dtype=object)
```

```
In [52]: df.drop(['Indicator Name', 'Indicator Code', 'Country Code'], axis = 1, inplace = True)
```

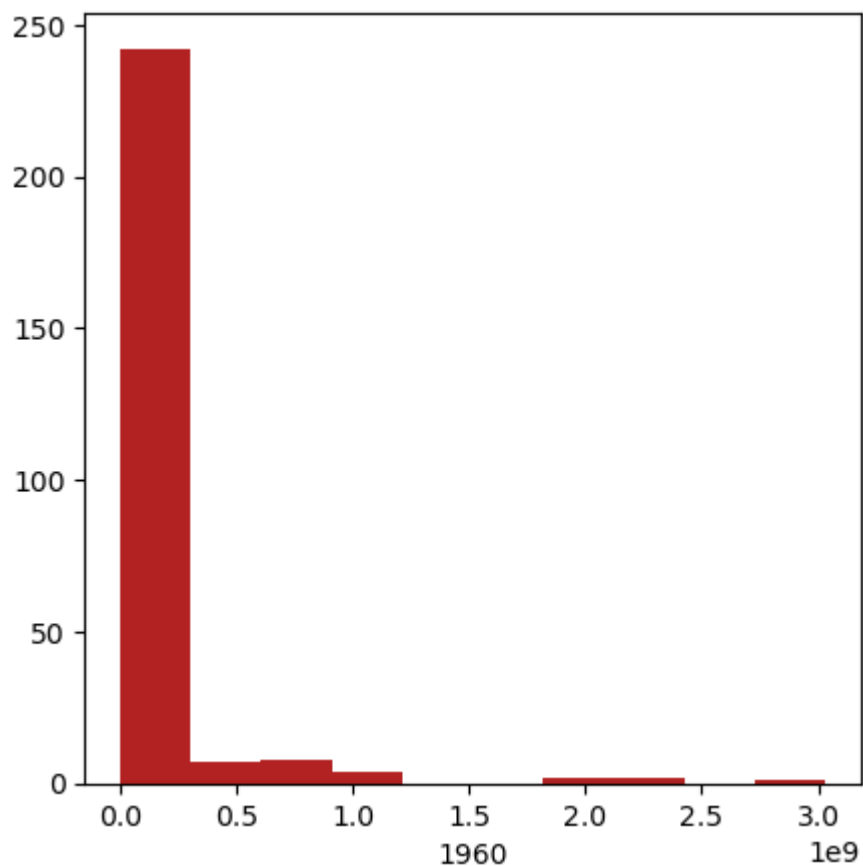
```
In [53]: df.columns
```

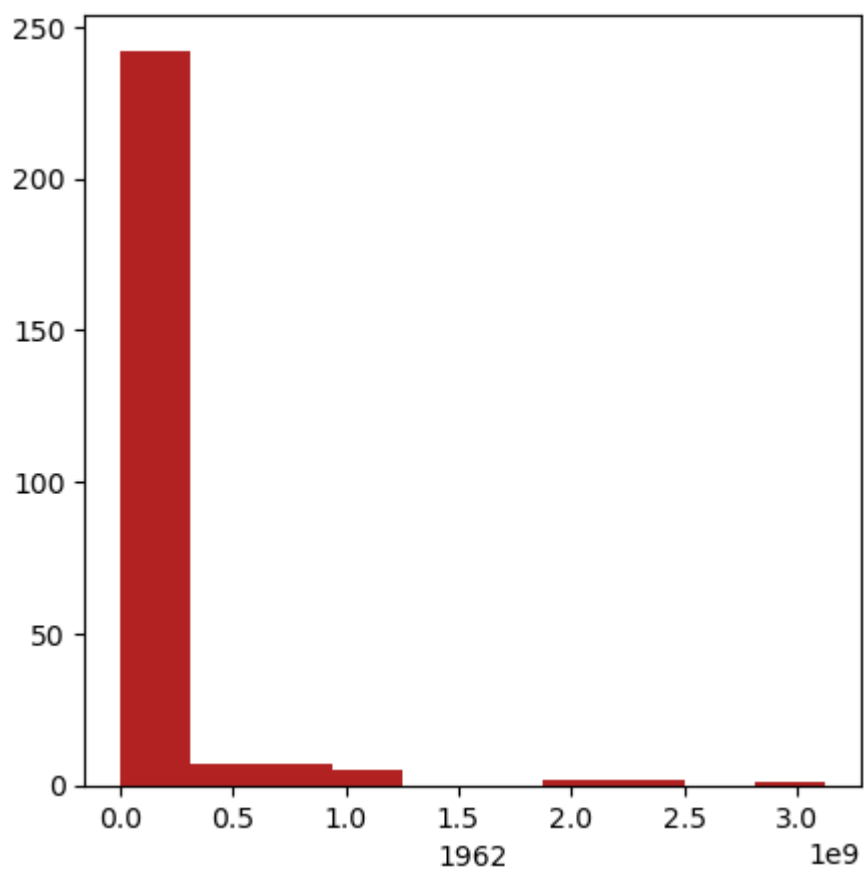
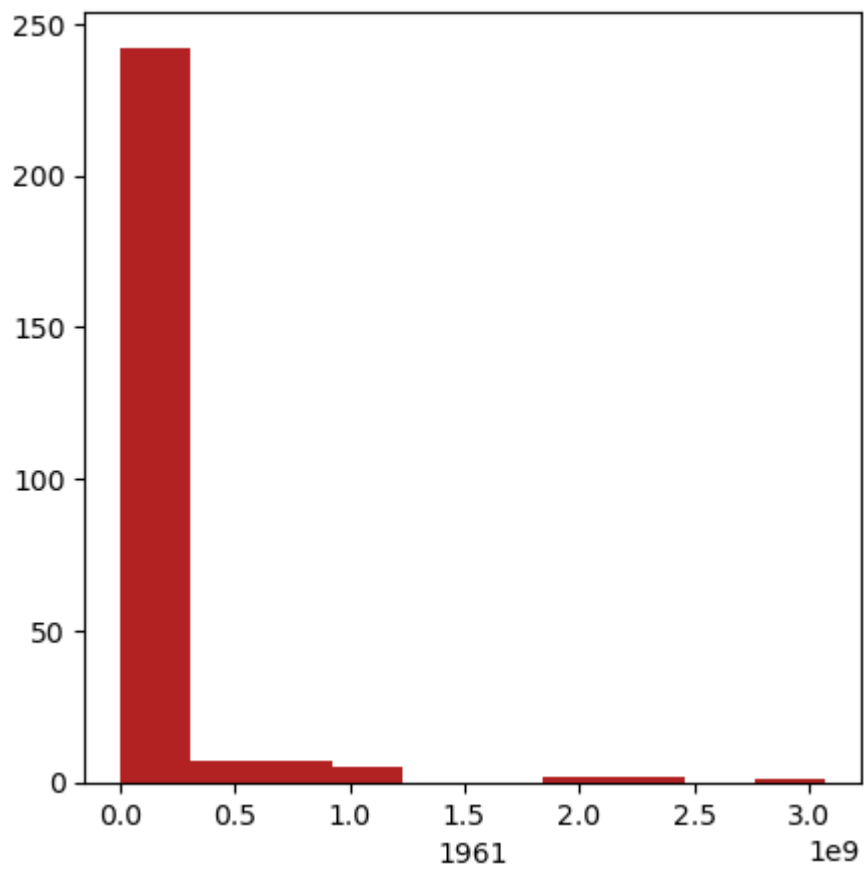


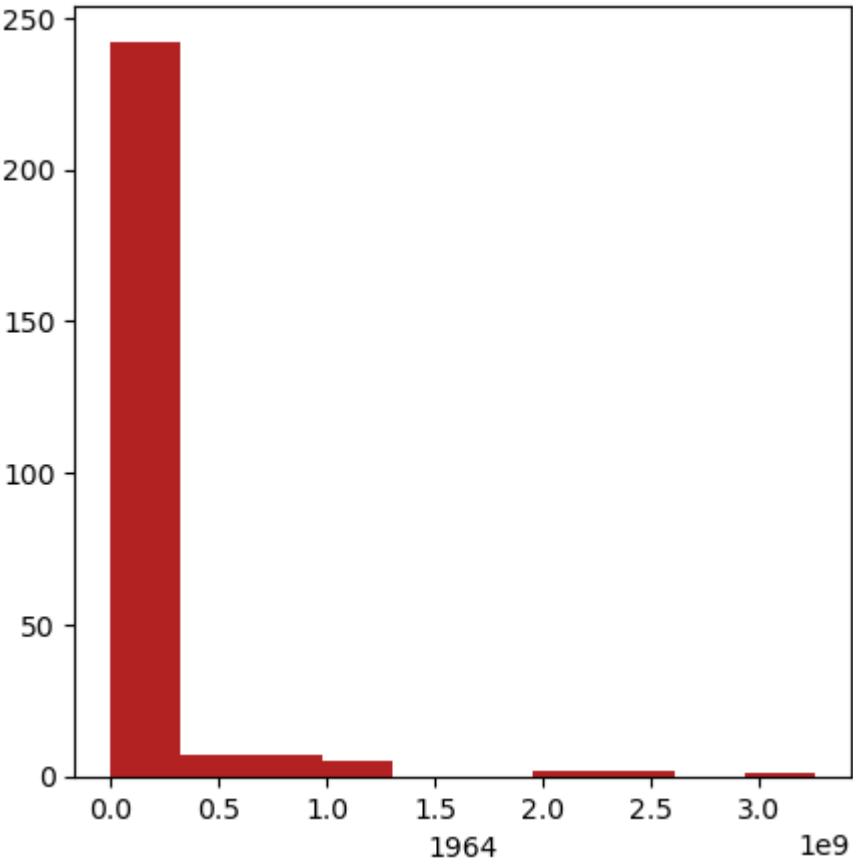
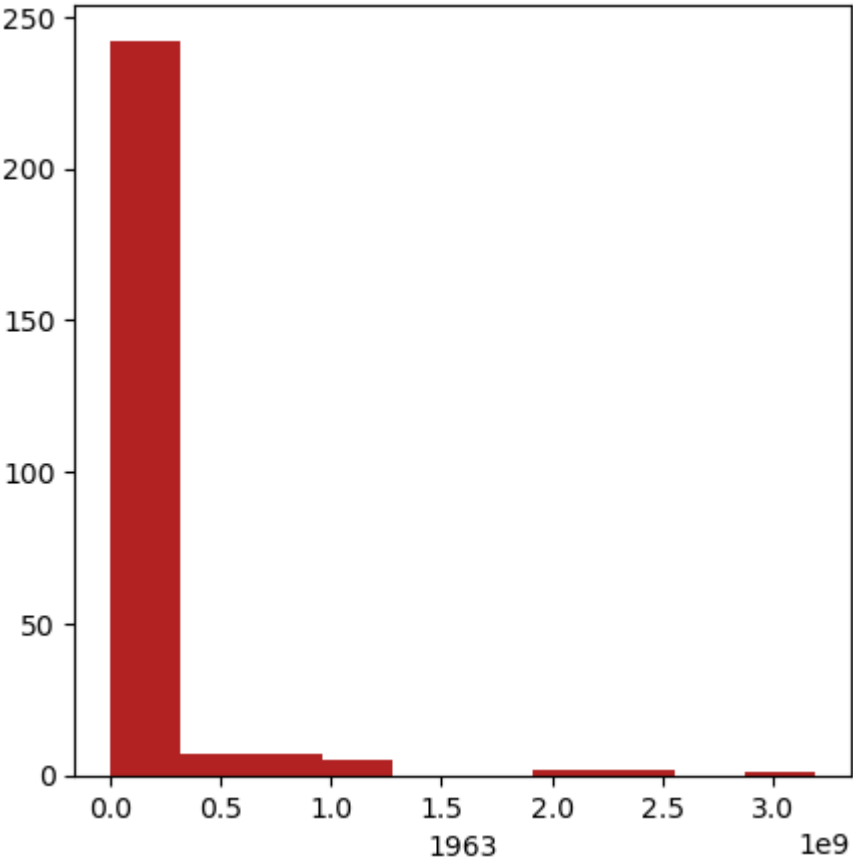
```
Out[53]: Index(['Country Name', '1960', '1961', '1962', '1963', '1964', '1965', '1966',
        '1967', '1968', '1969', '1970', '1971', '1972', '1973', '1974', '1975',
        '1976', '1977', '1978', '1979', '1980', '1981', '1982', '1983', '1984',
        '1985', '1986', '1987', '1988', '1989', '1990', '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'],
        dtype='object')
```

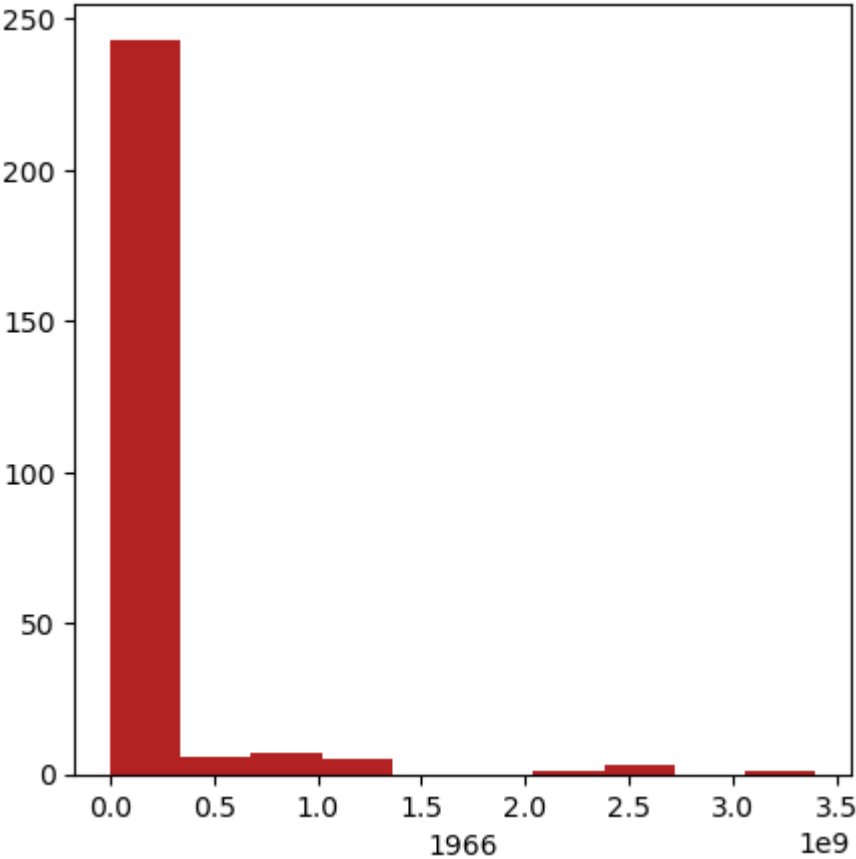
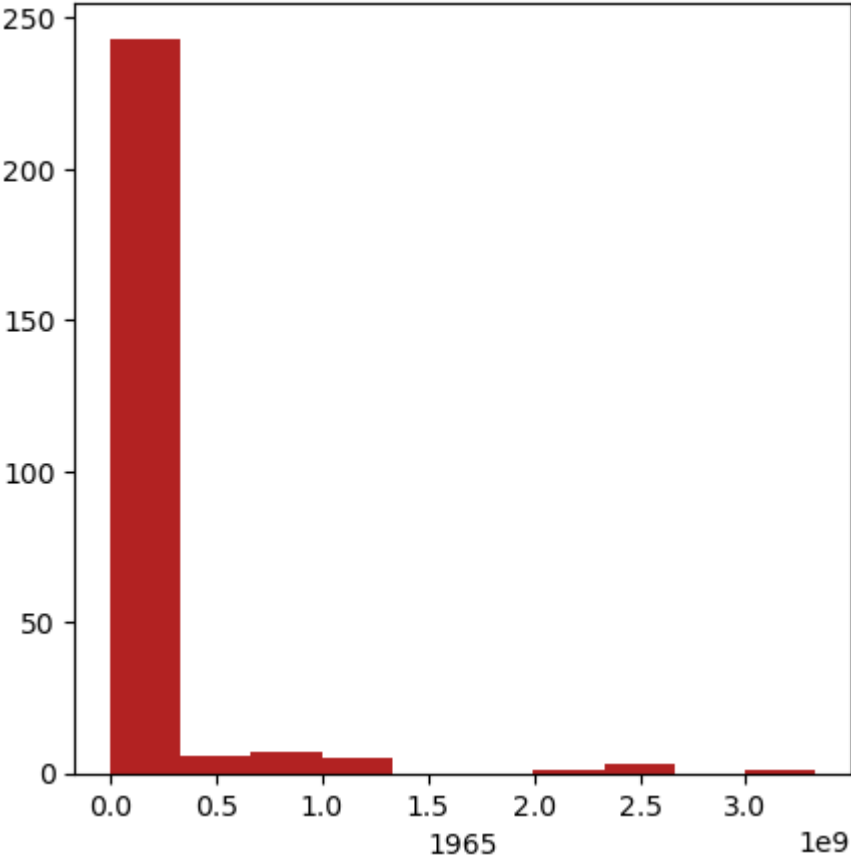
```
In [54]: cols = ['1960', '1961', '1962', '1963', '1964', '1965', '1966', '1967', '1968', '1969', '1970',
        '1976', '1977', '1978', '1979', '1980', '1981', '1982', '1983', '1984', '1985', '1986',
        '1992', '1993', '1994', '1995', '1996', '1997', '1998', '1999', '2000', '2001', '2002',
        '2008', '2009', '2010', '2011', '2012', '2013', '2014', '2015', '2016', '2017', '2018']
```

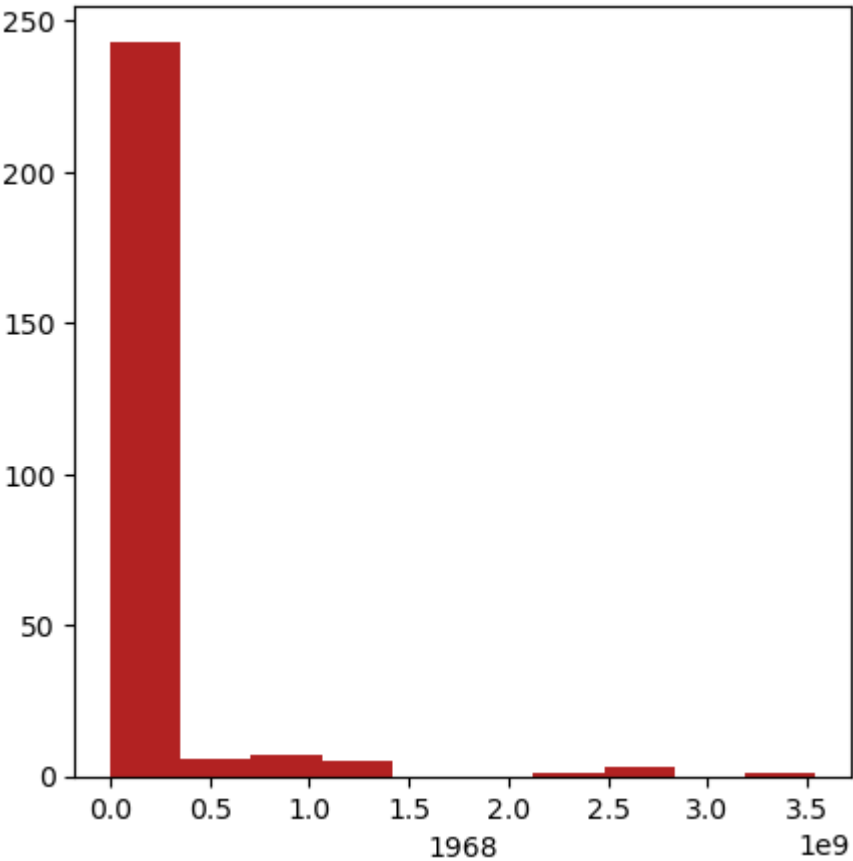
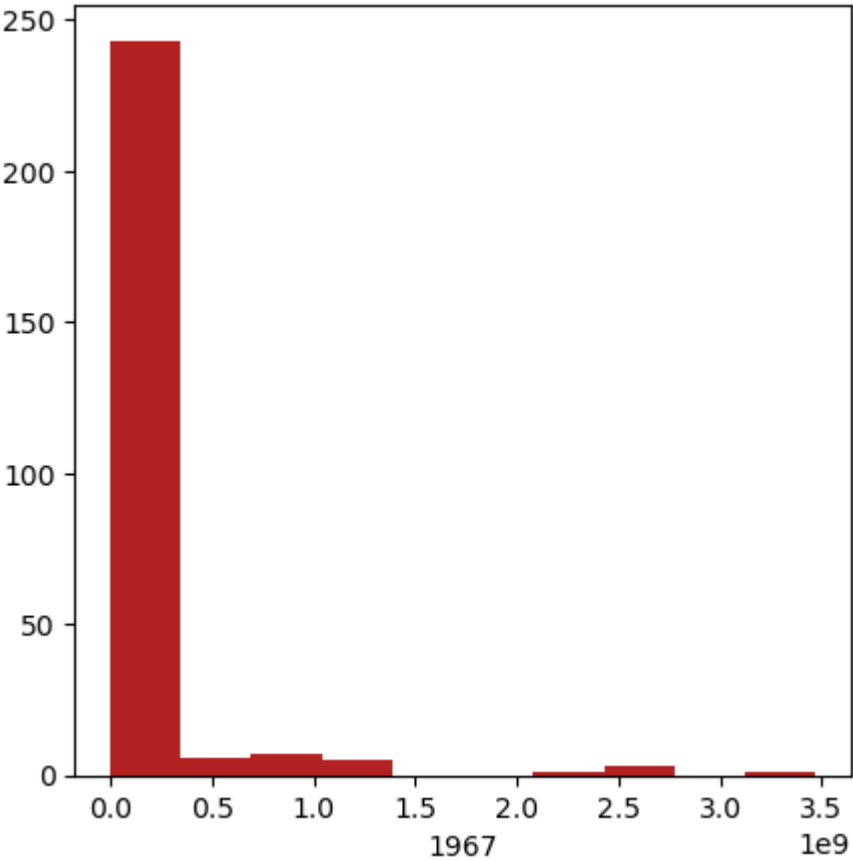
```
In [56]: for i in cols:
        fig = plt.figure(figsize=(5,5))
        plt.hist(df[i],color='#B22222',bins=10)
        plt.xlabel(i)
        plt.show()
```

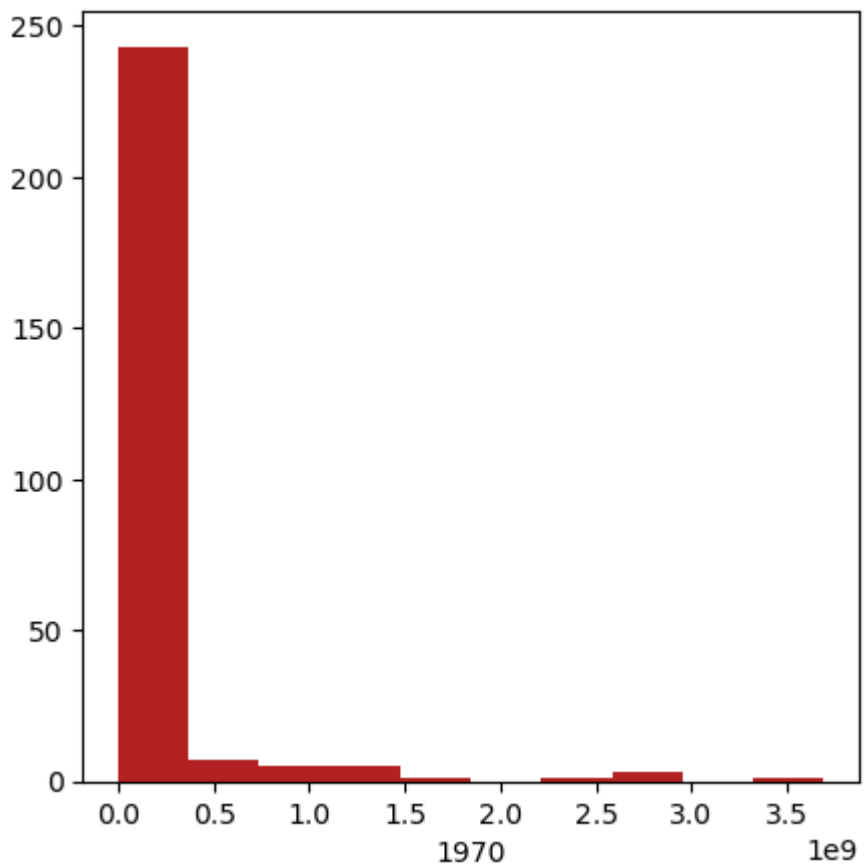
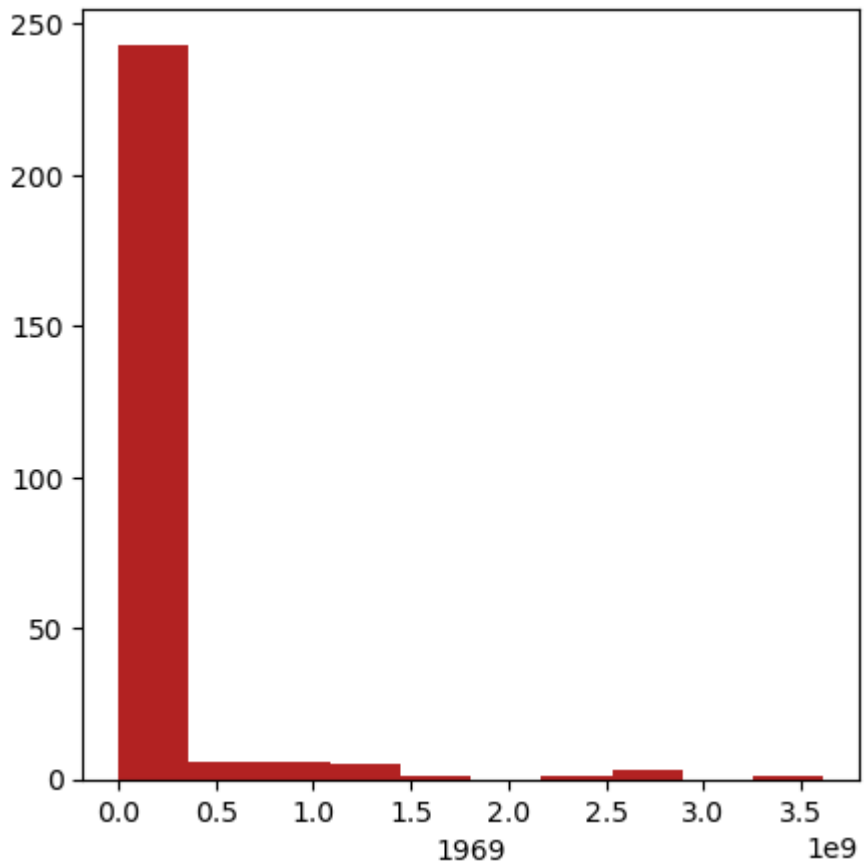


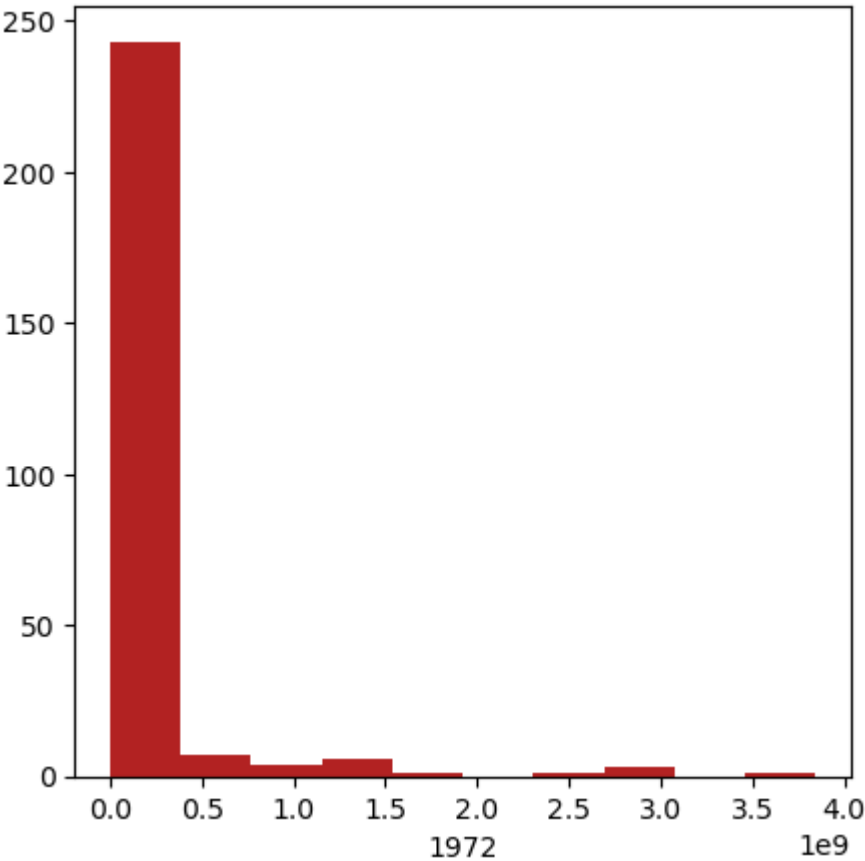
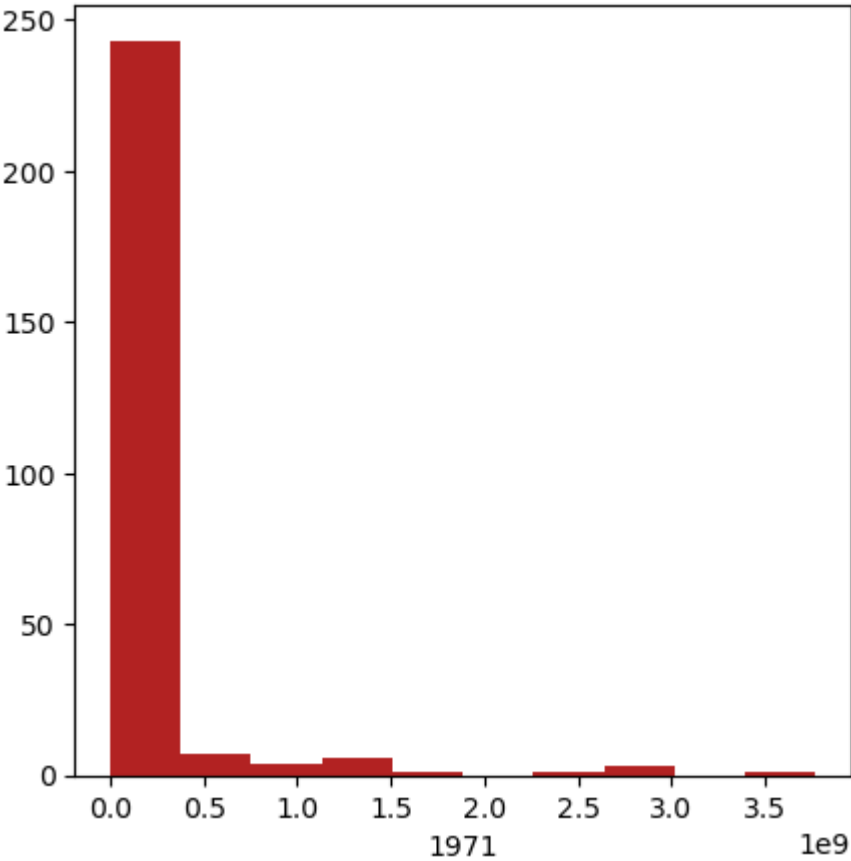


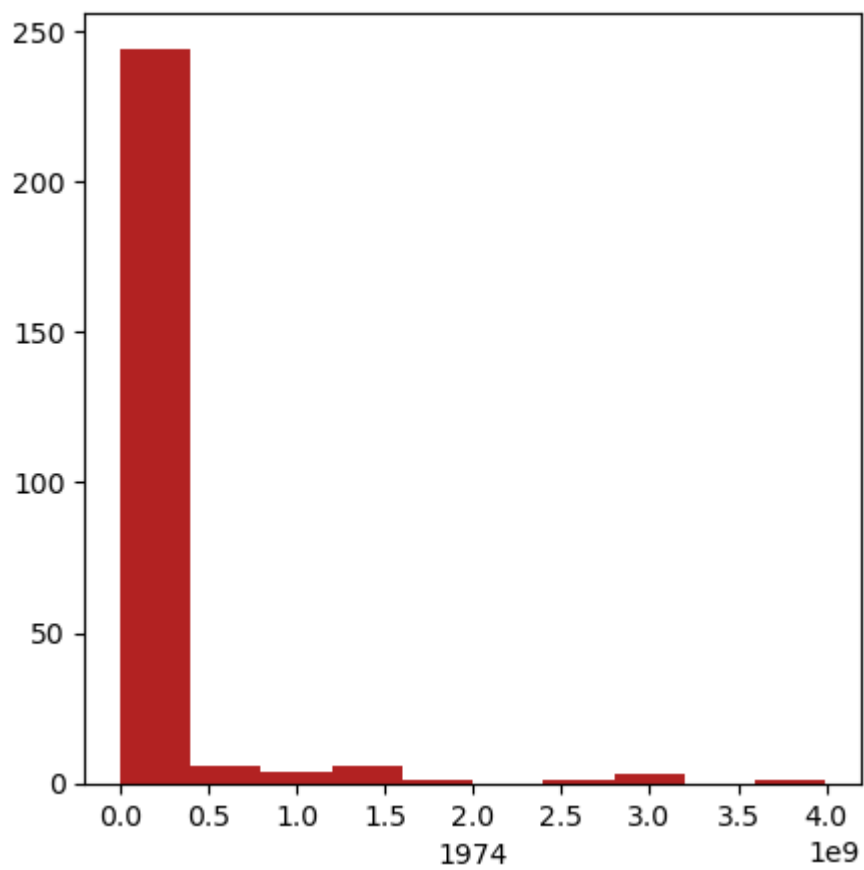
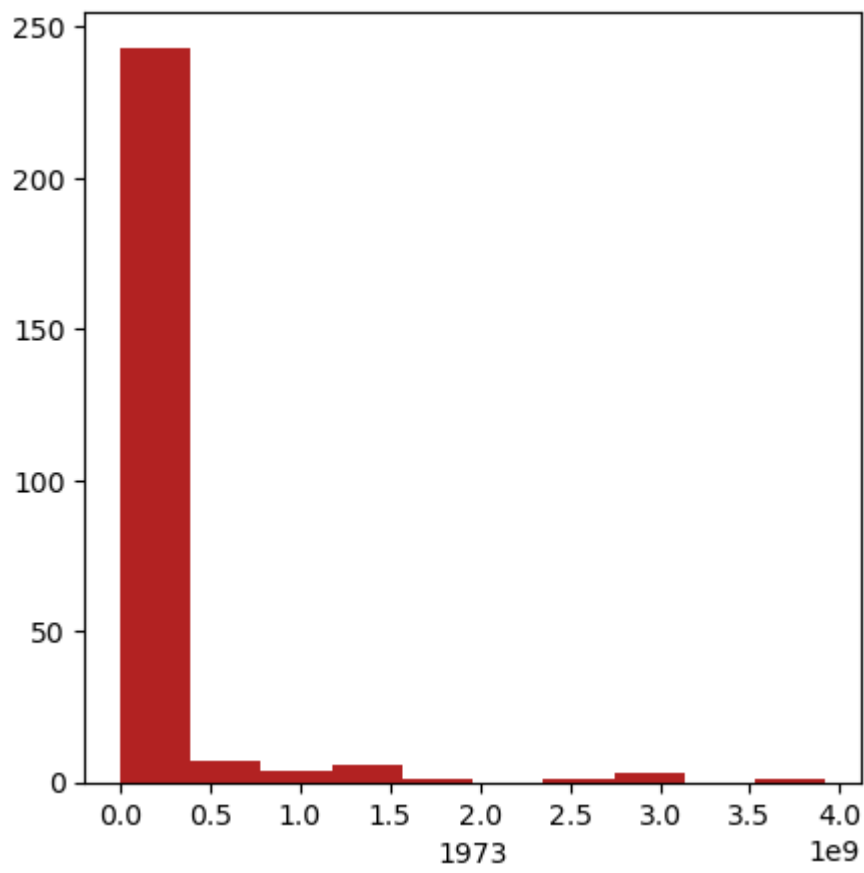


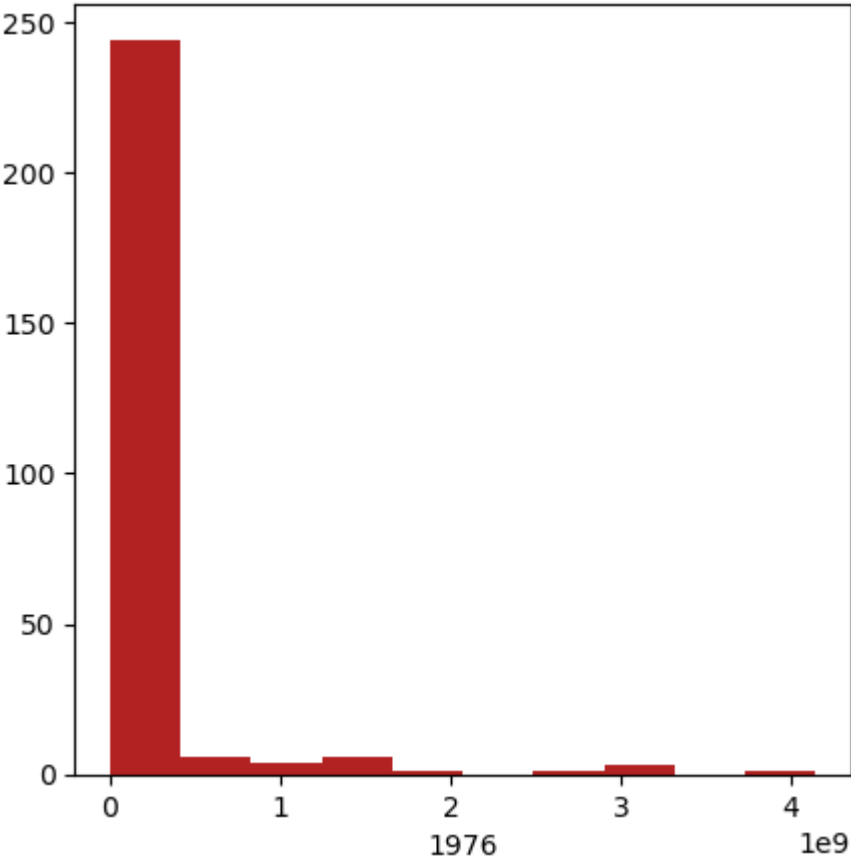
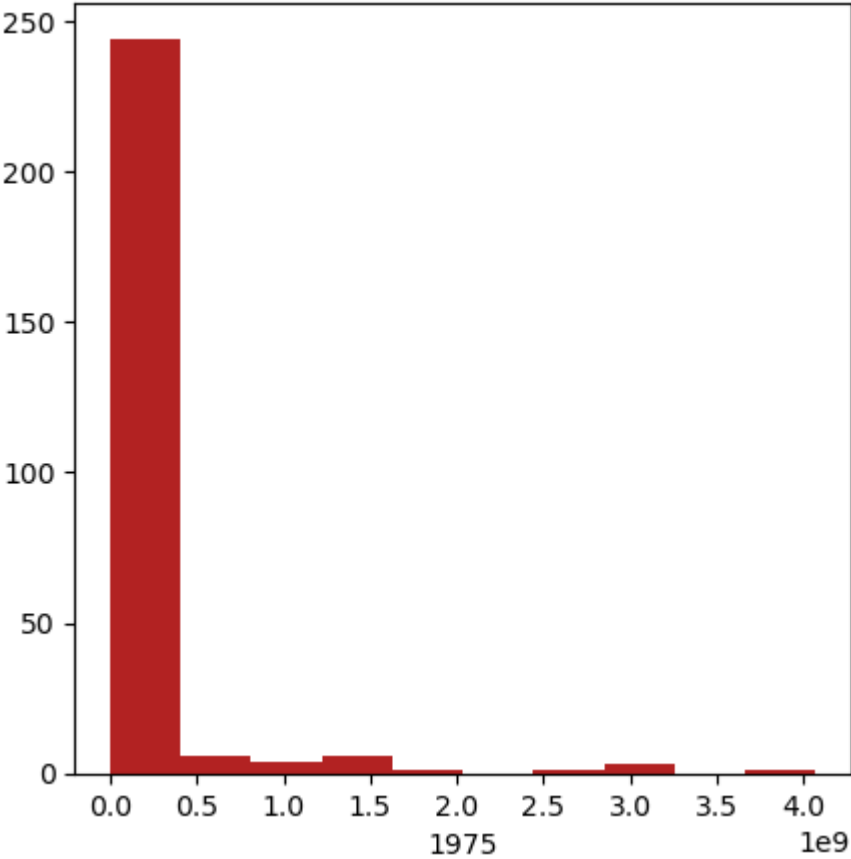


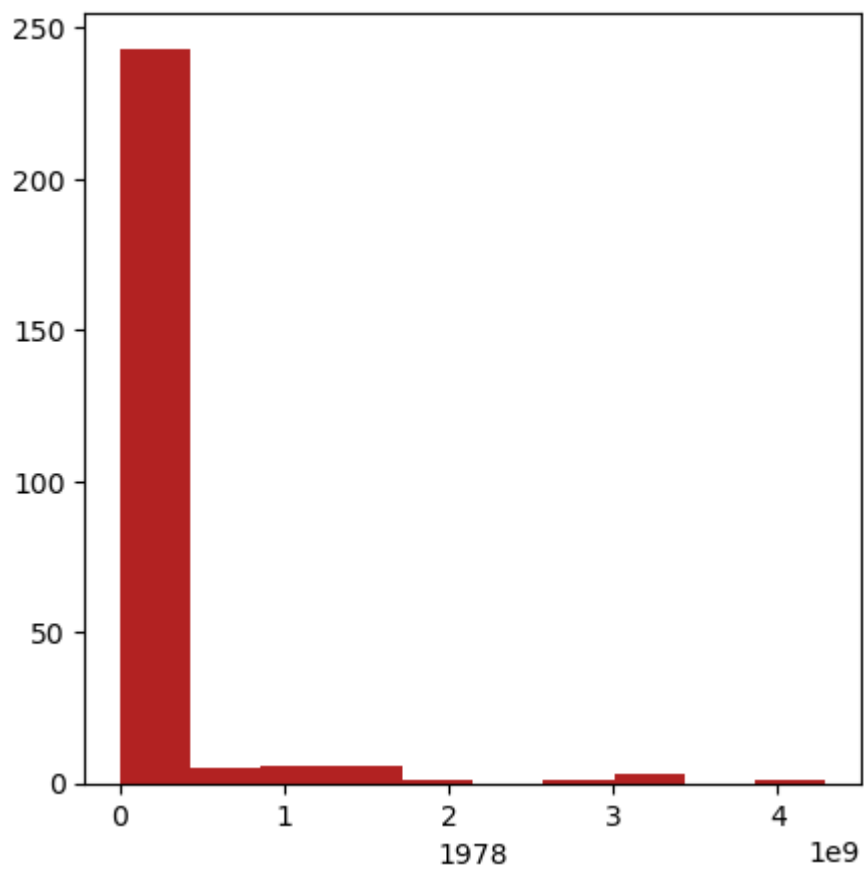
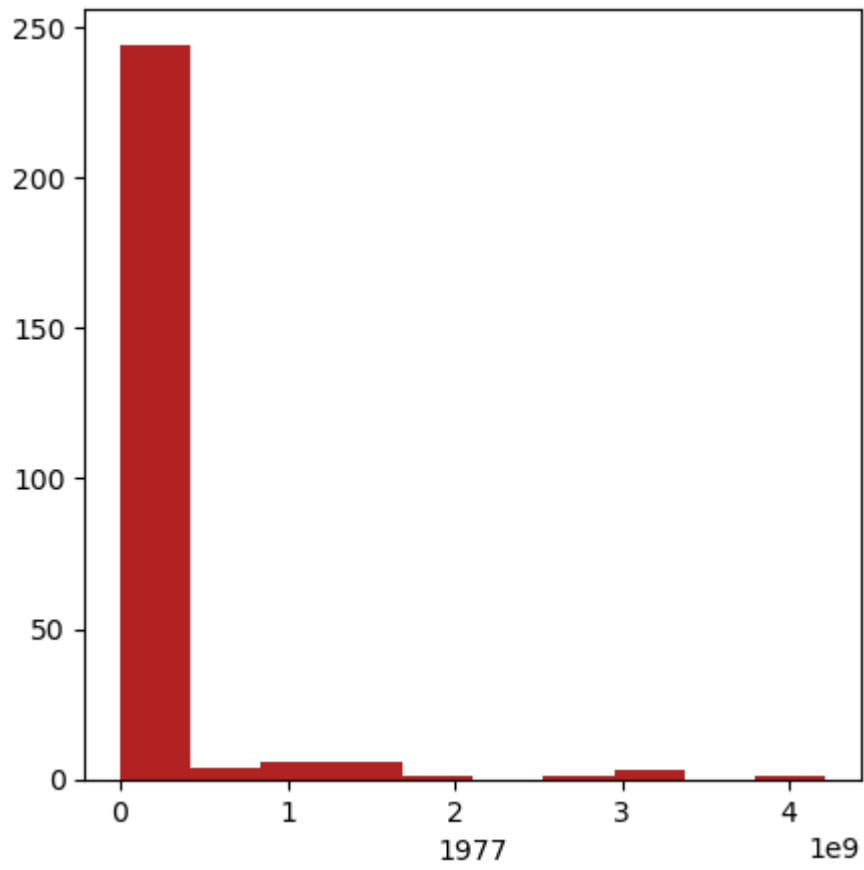


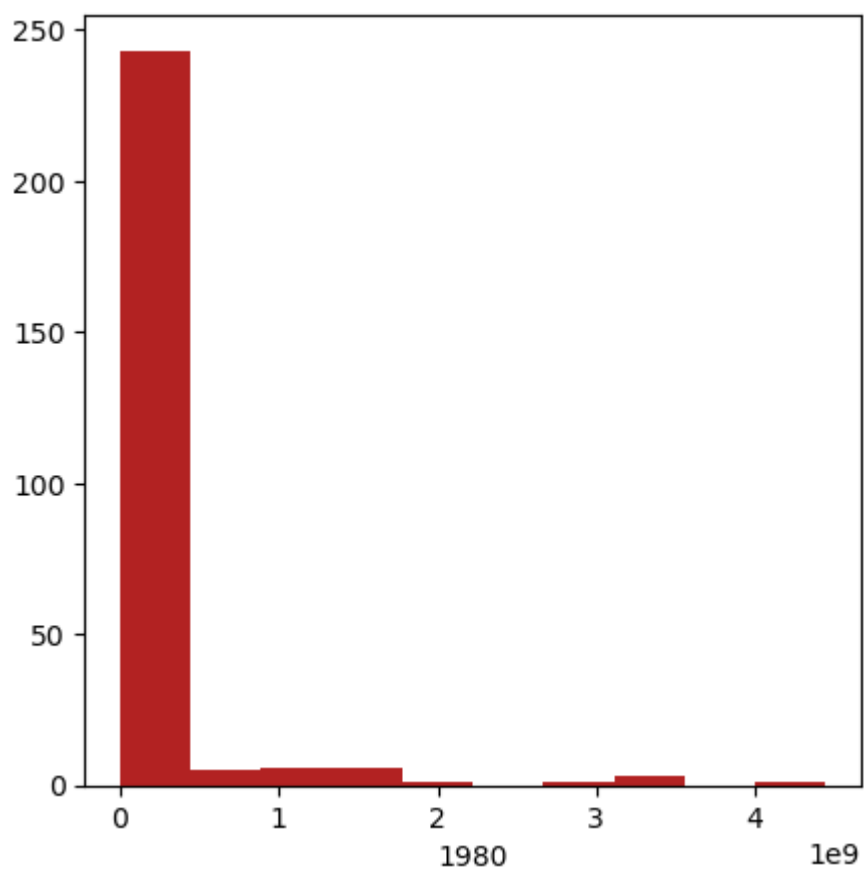
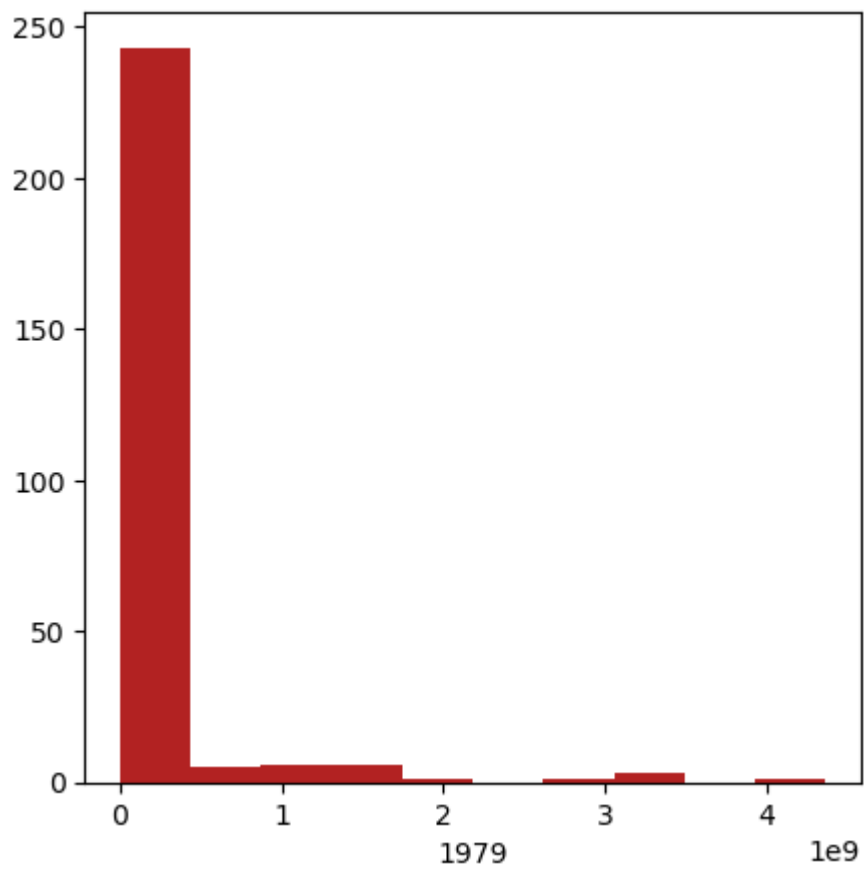


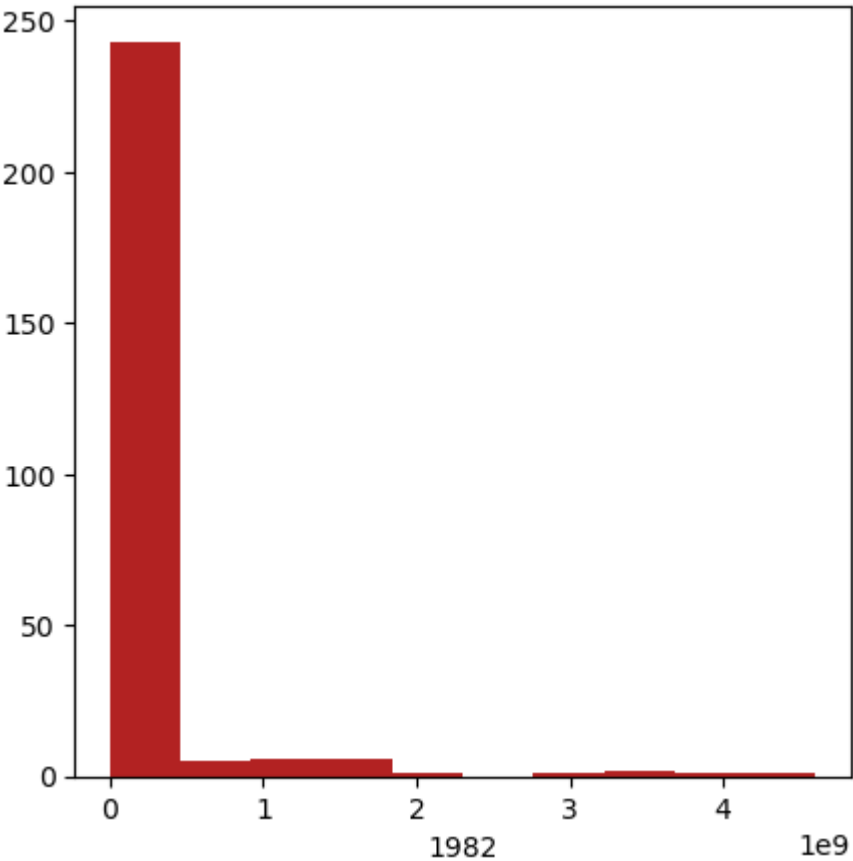
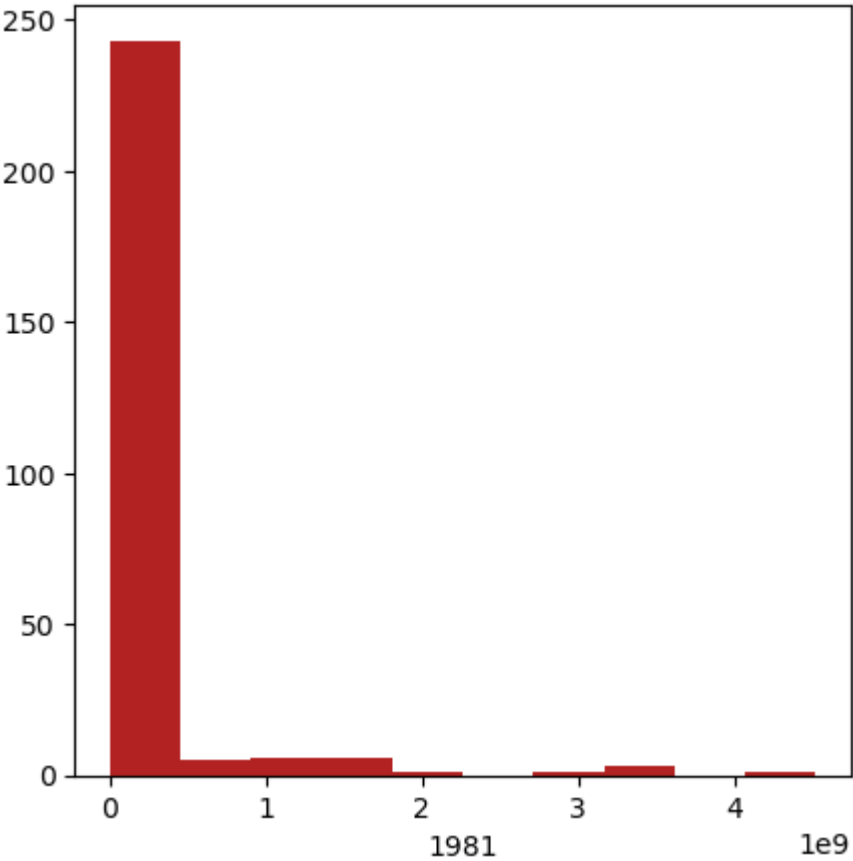


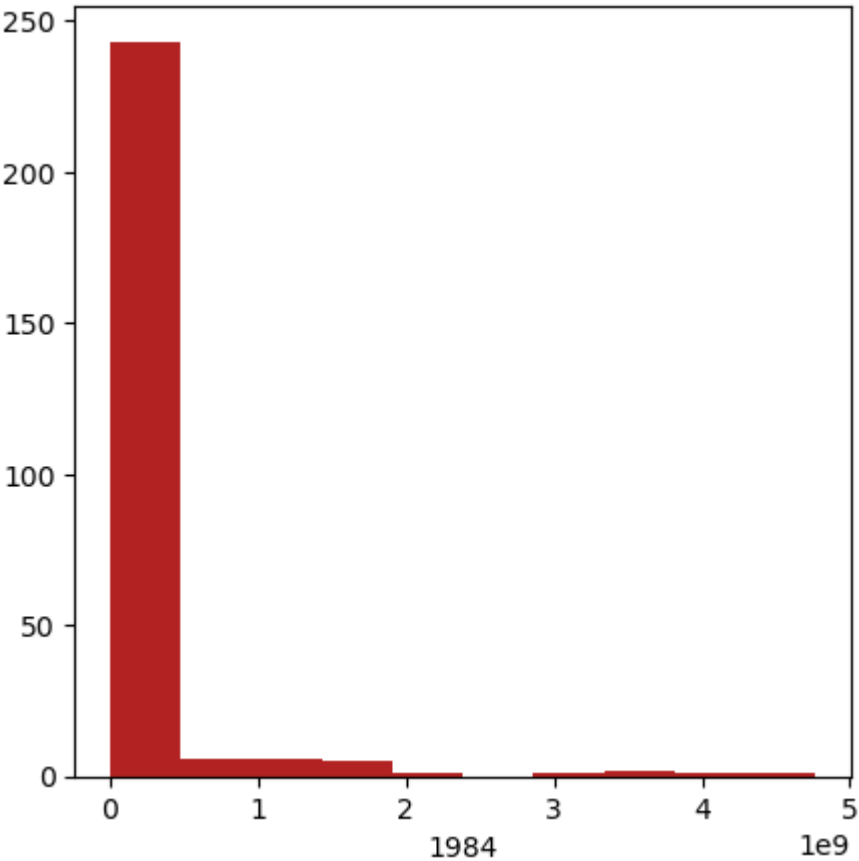
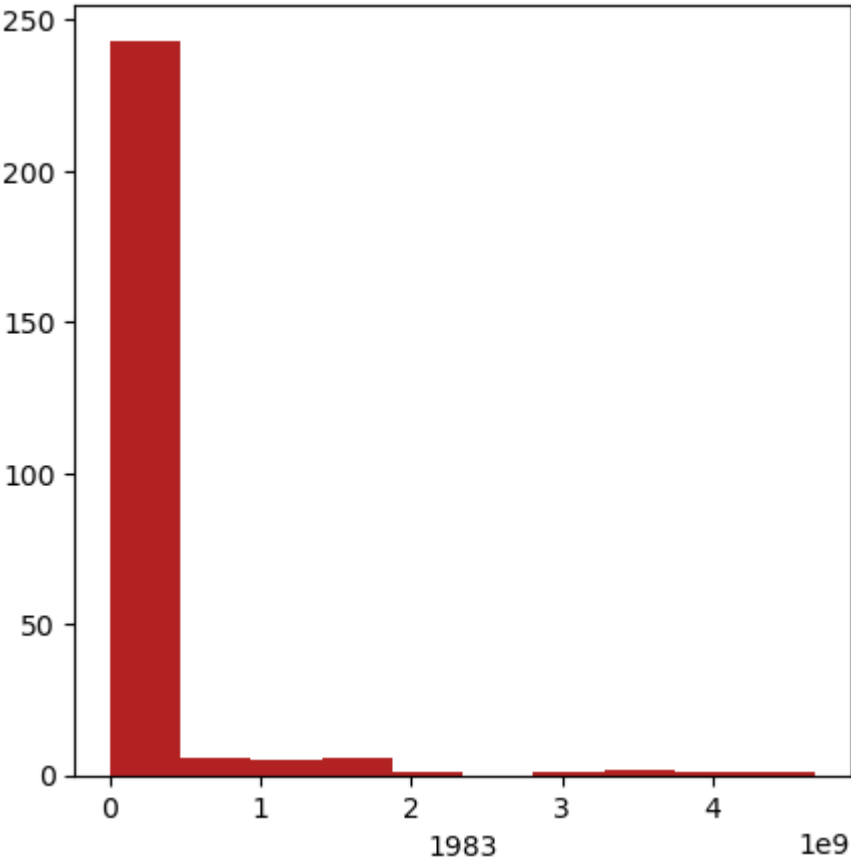


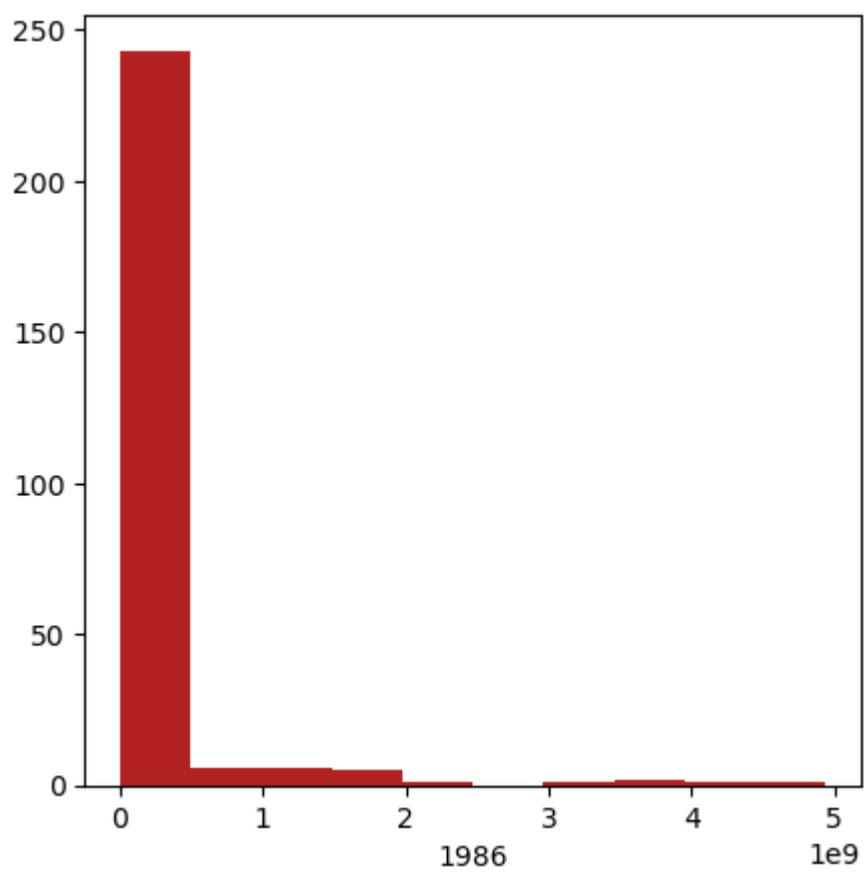
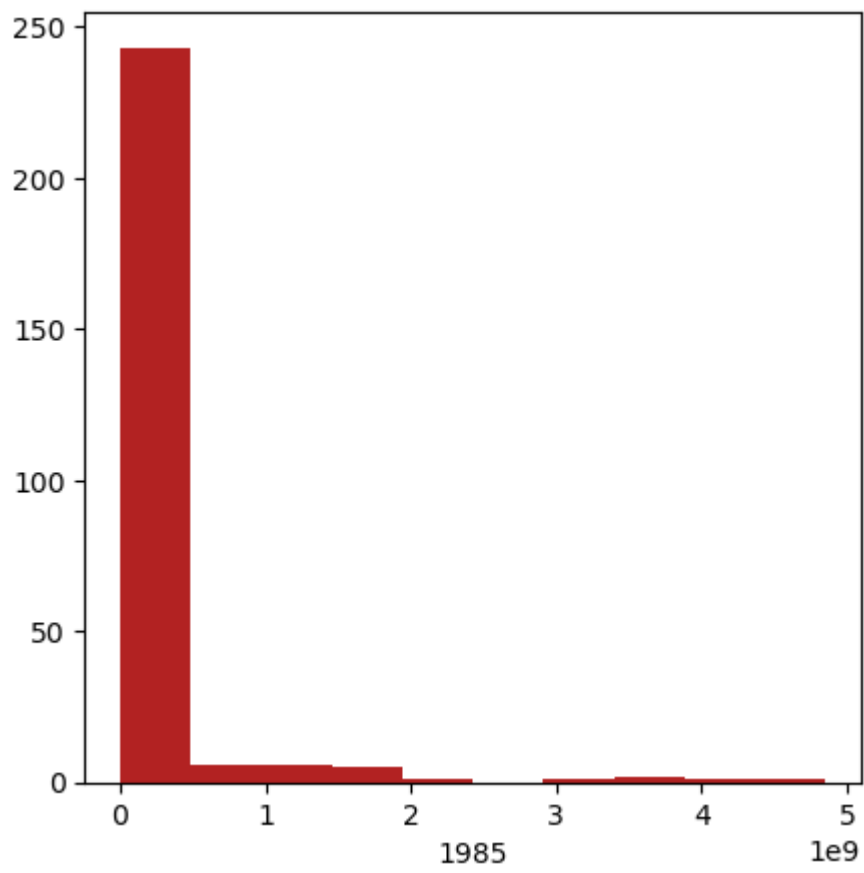


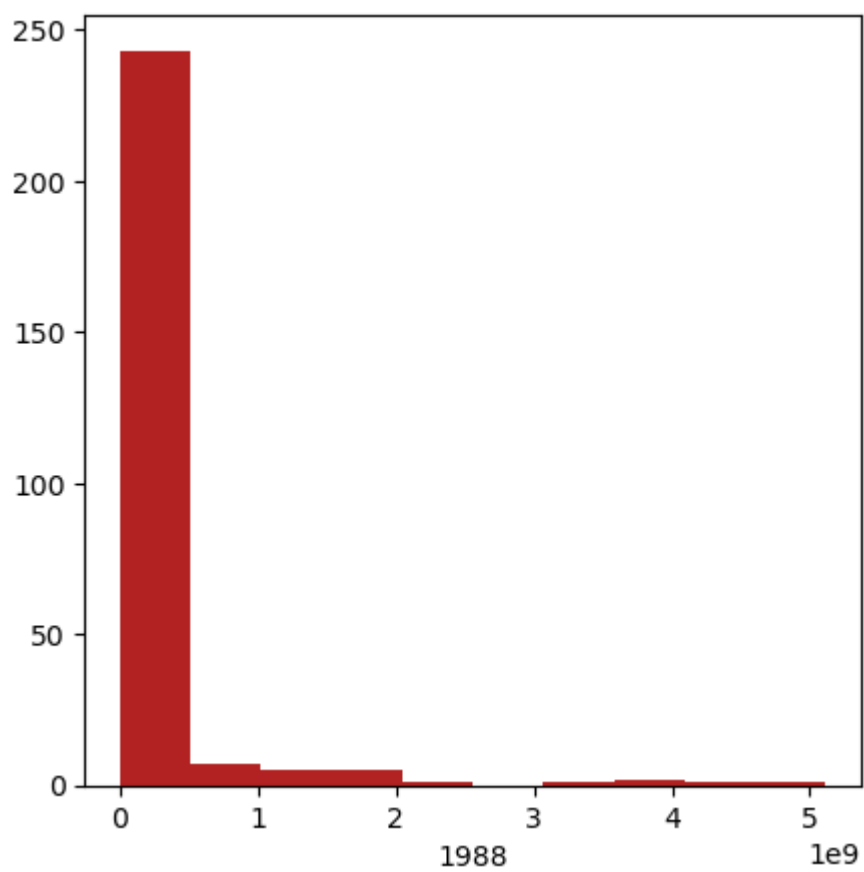
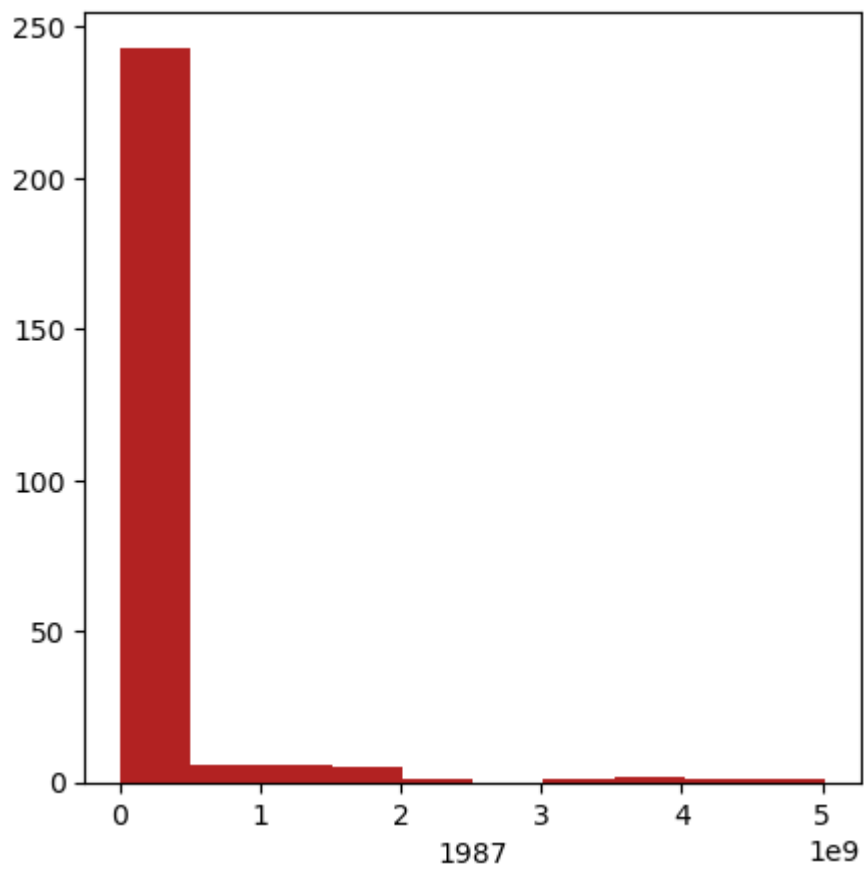


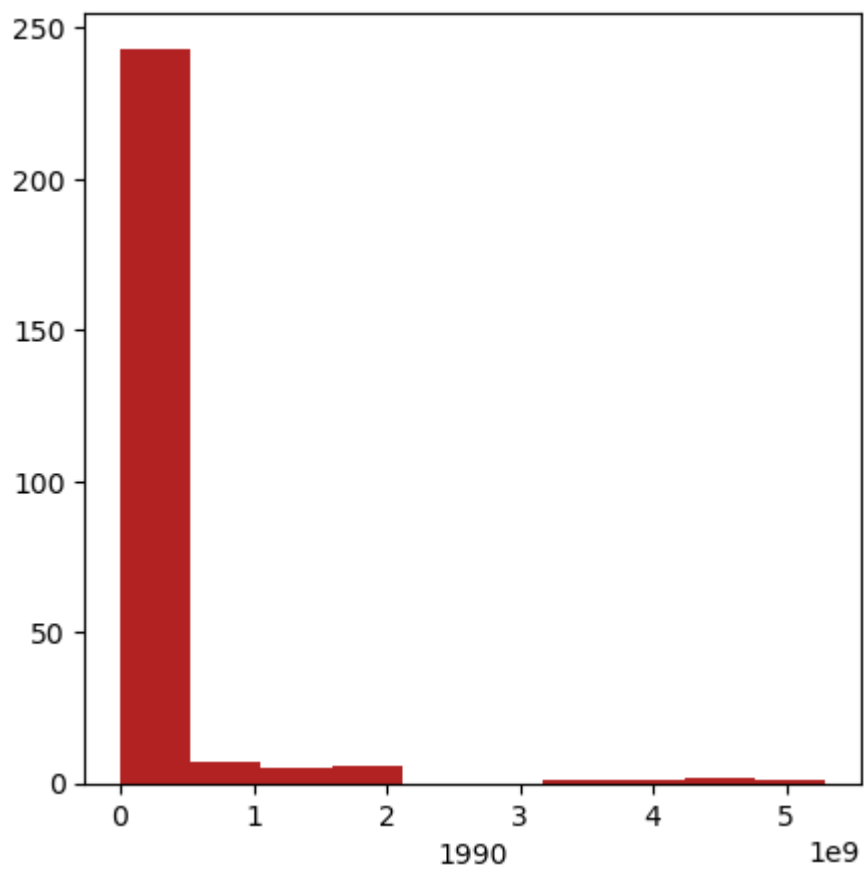
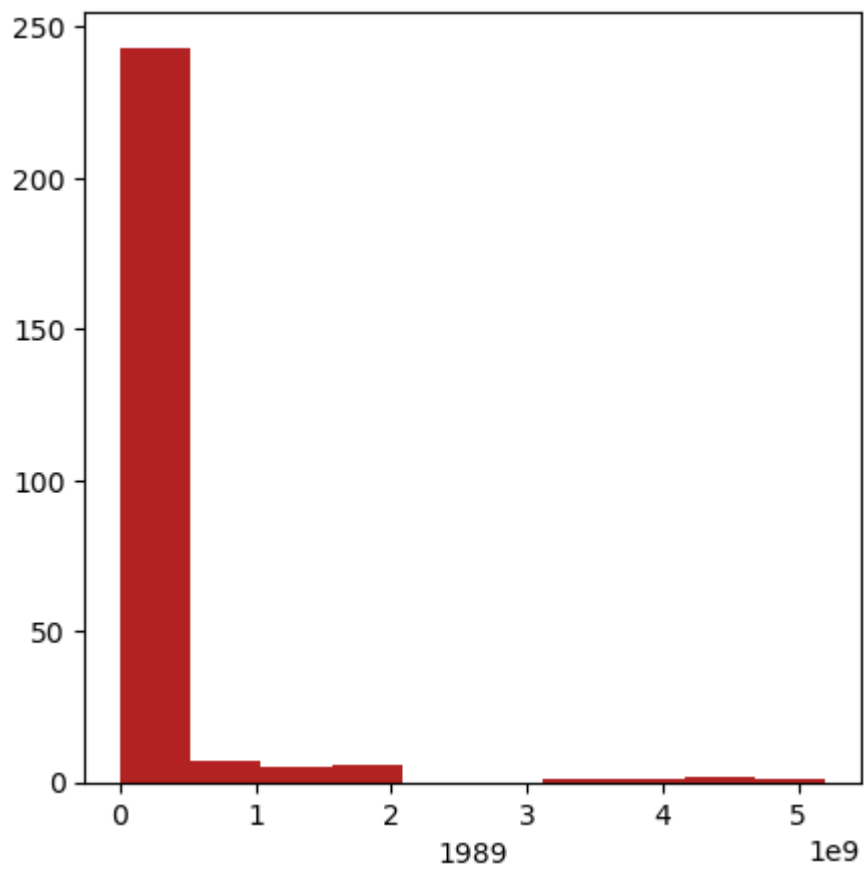


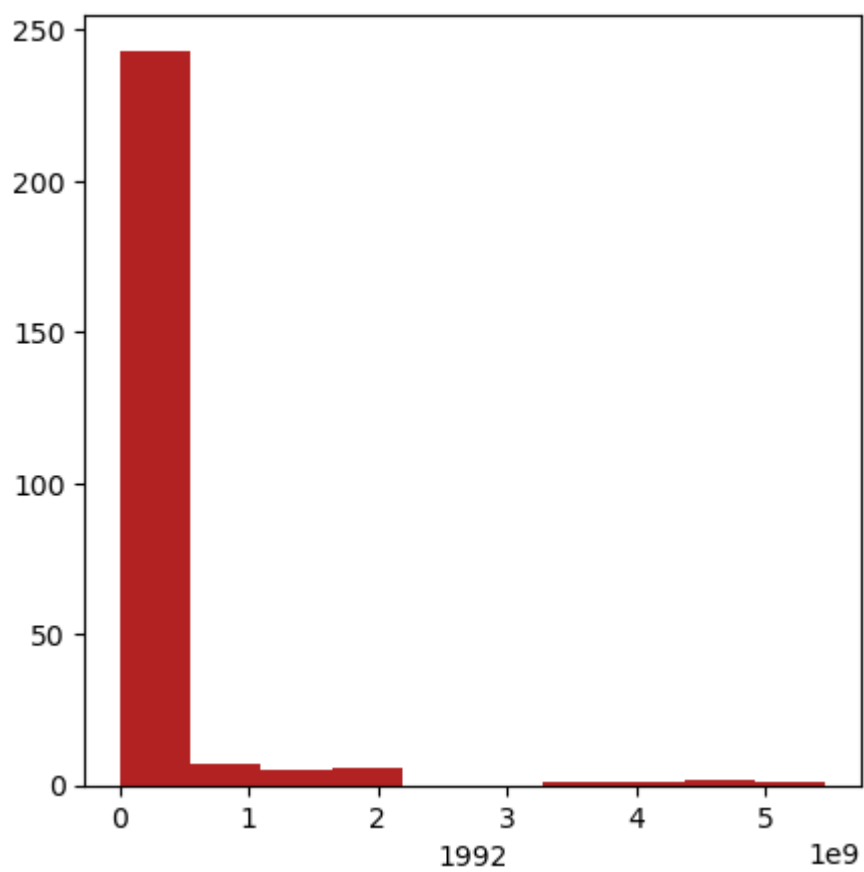
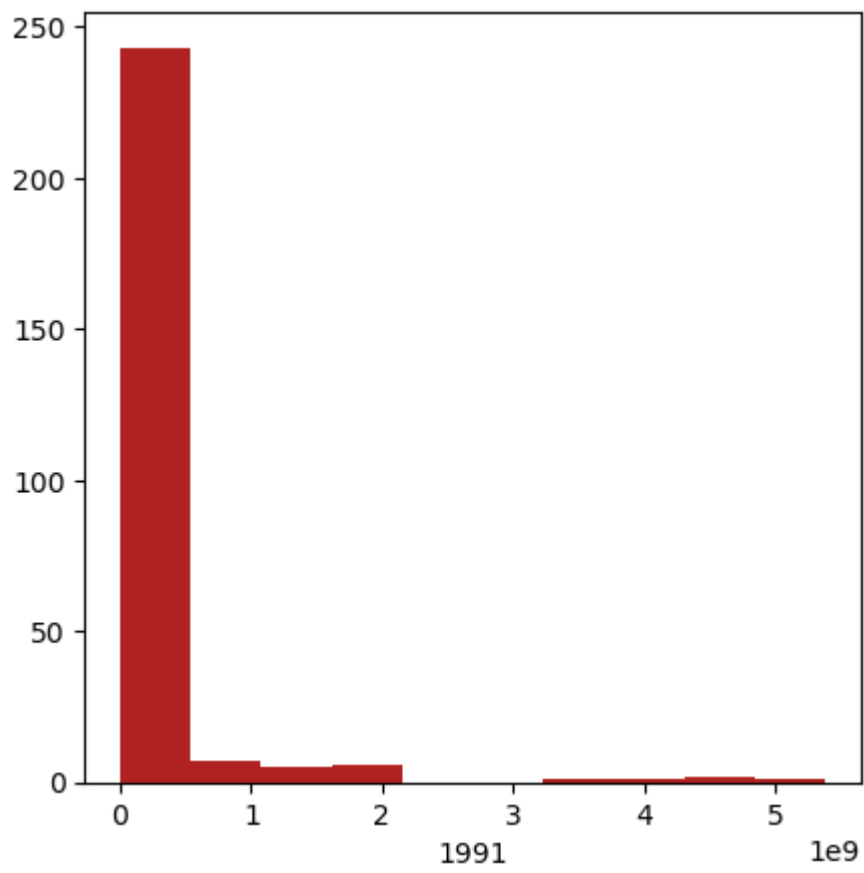


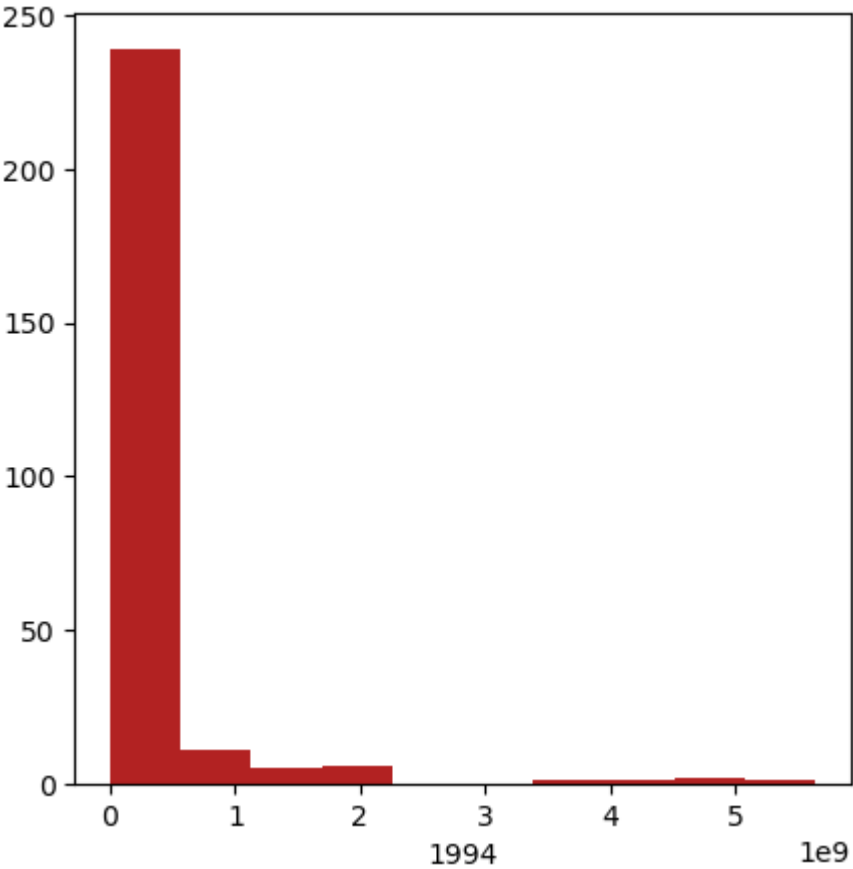
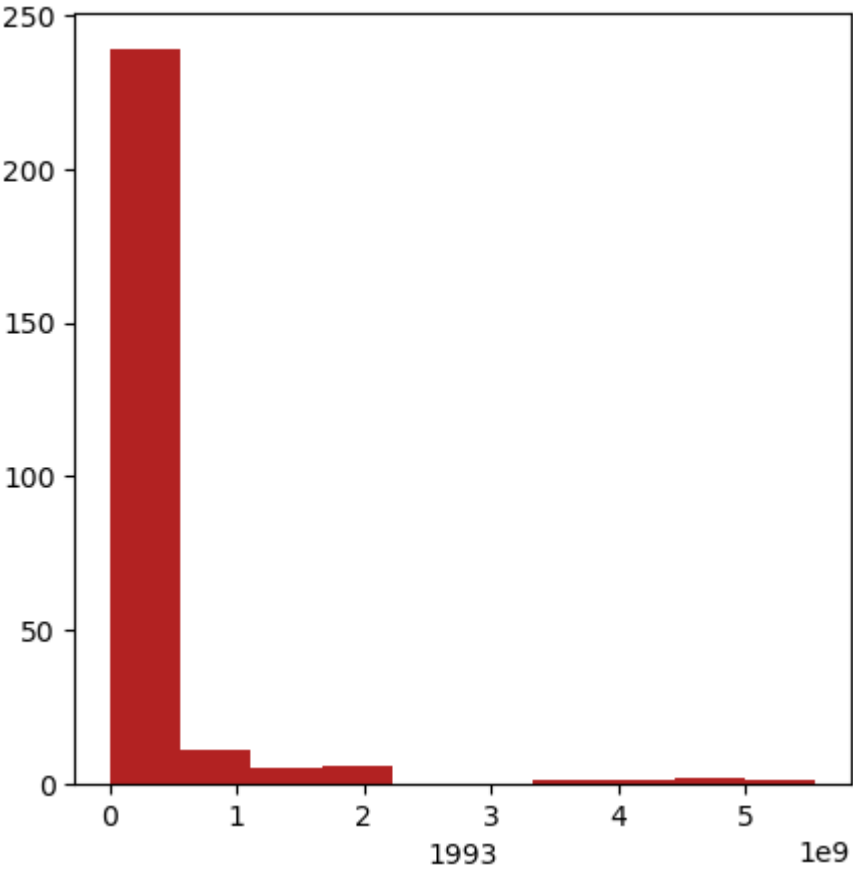


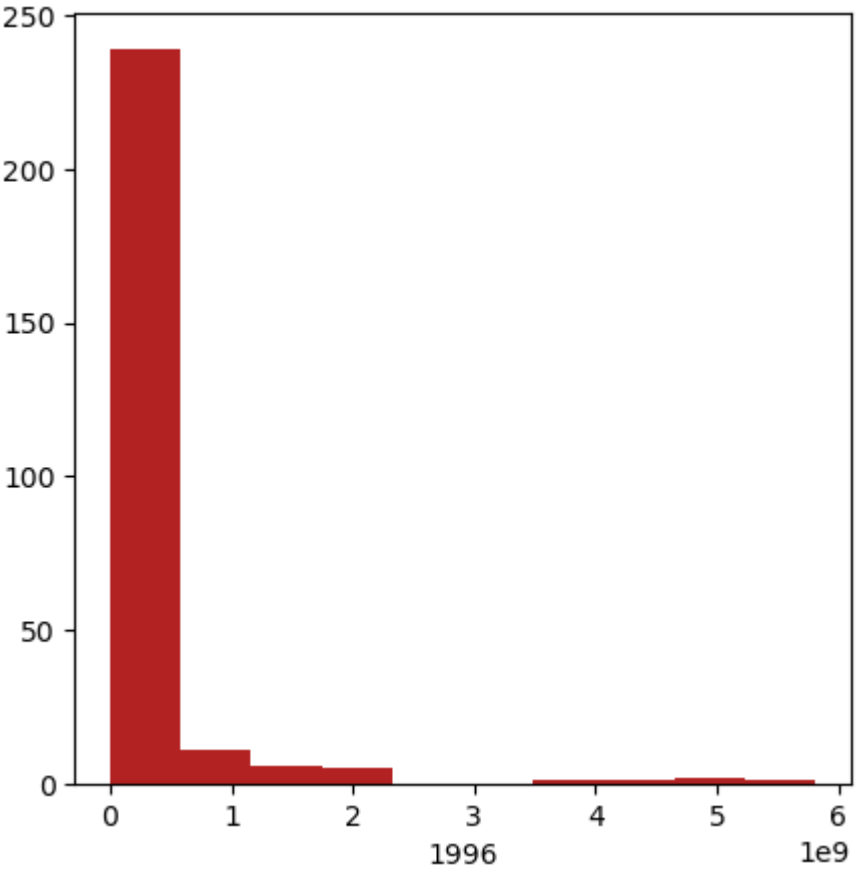
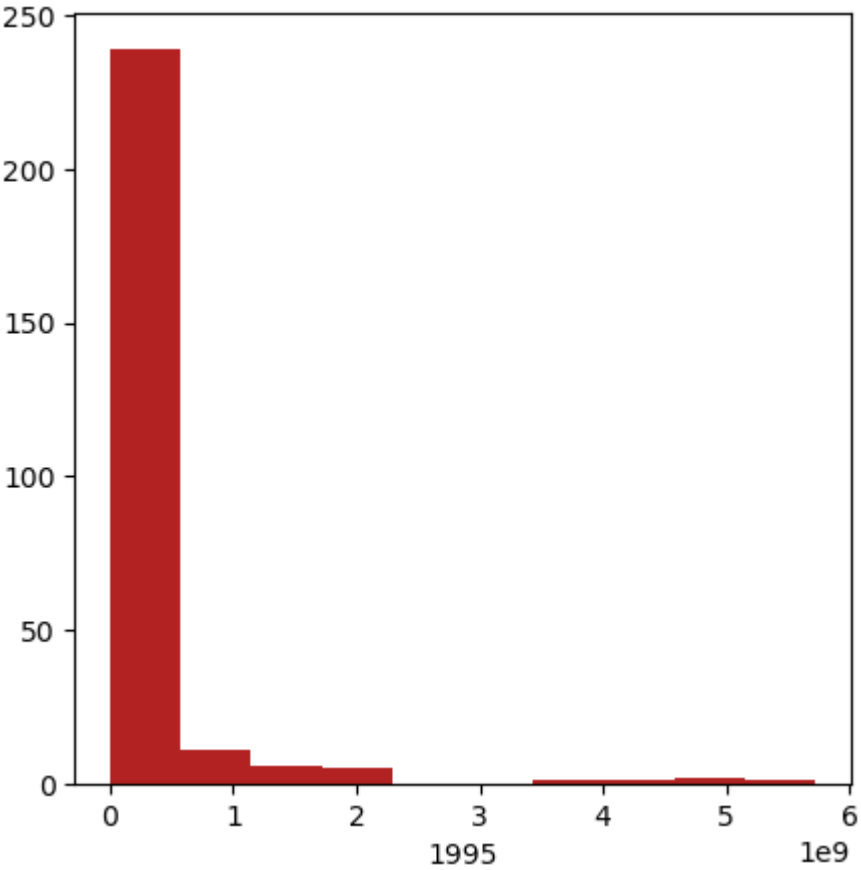


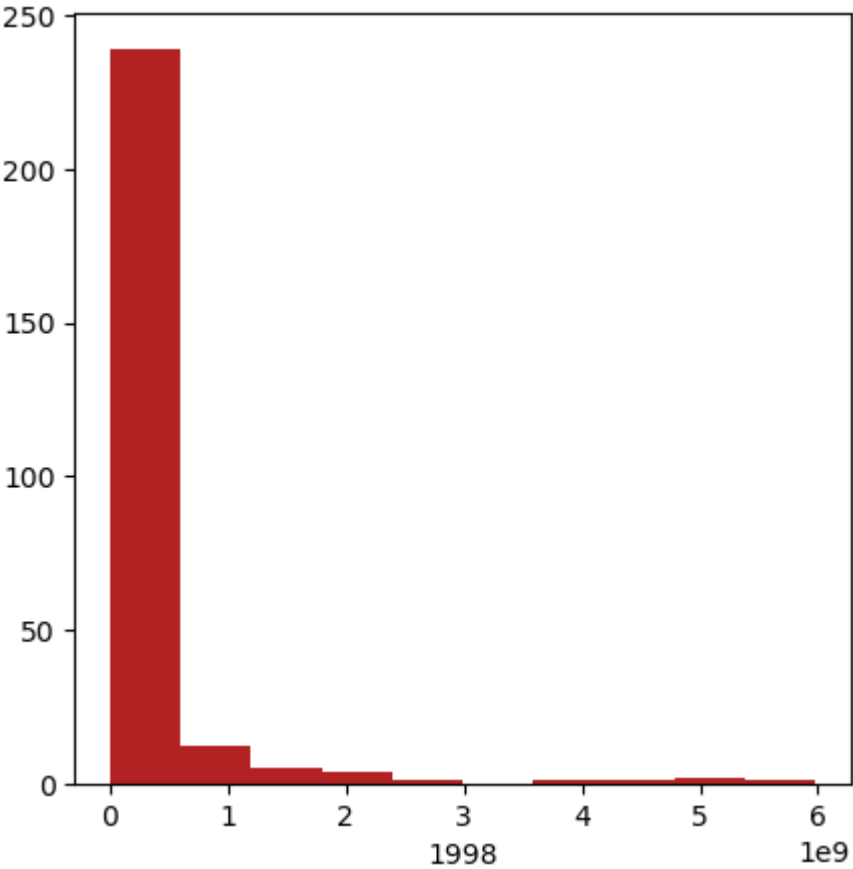
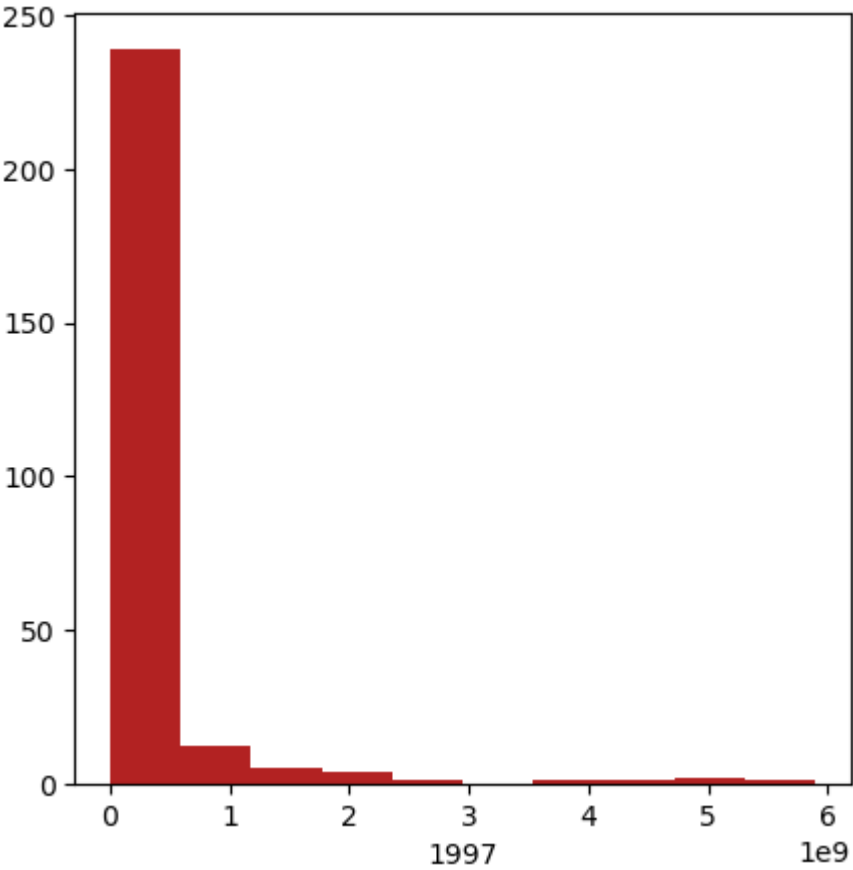


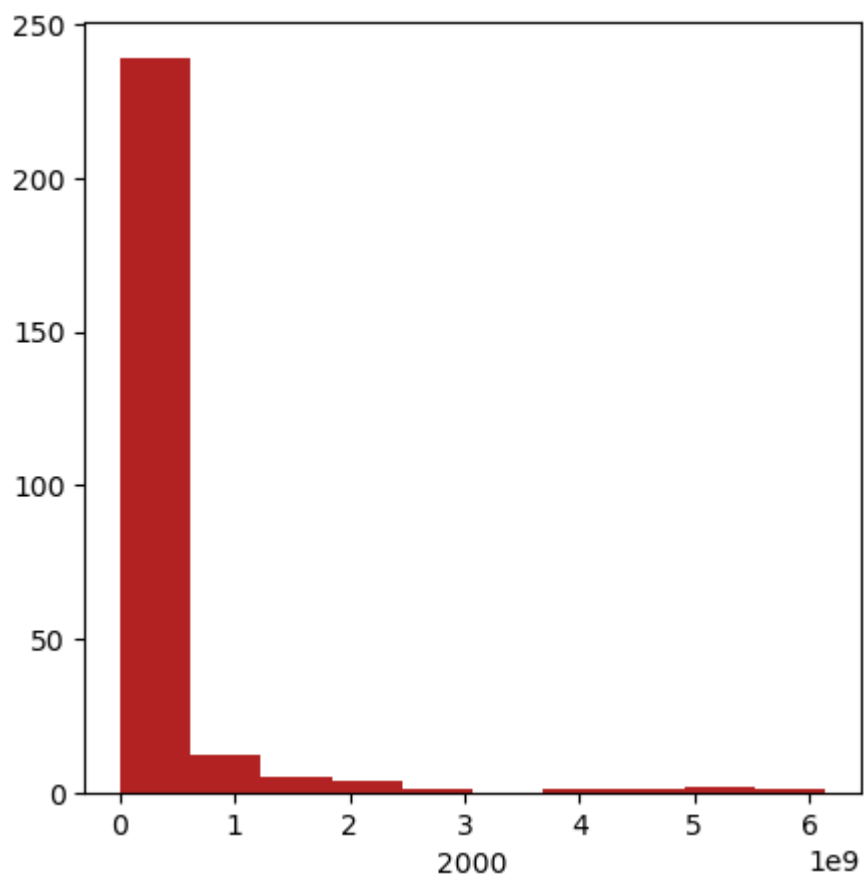
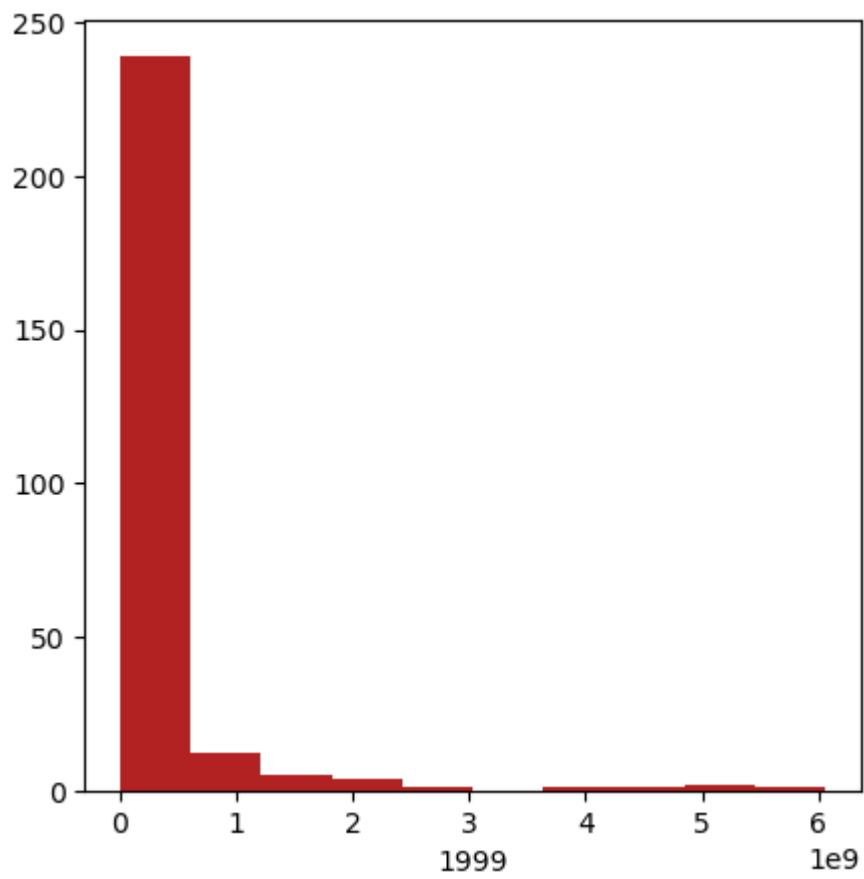


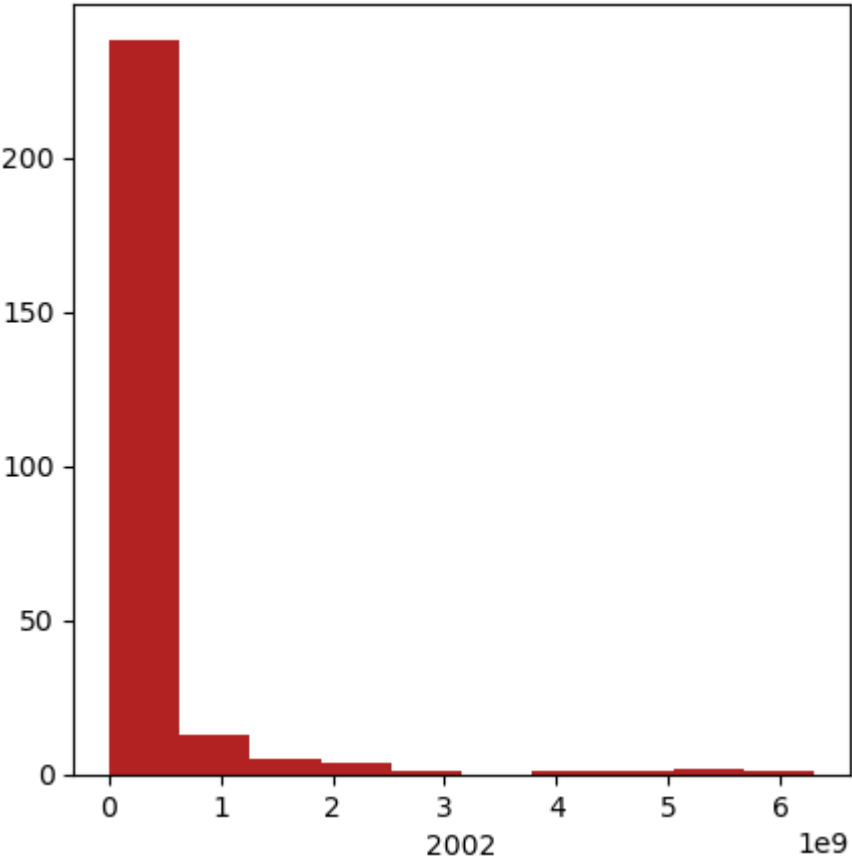
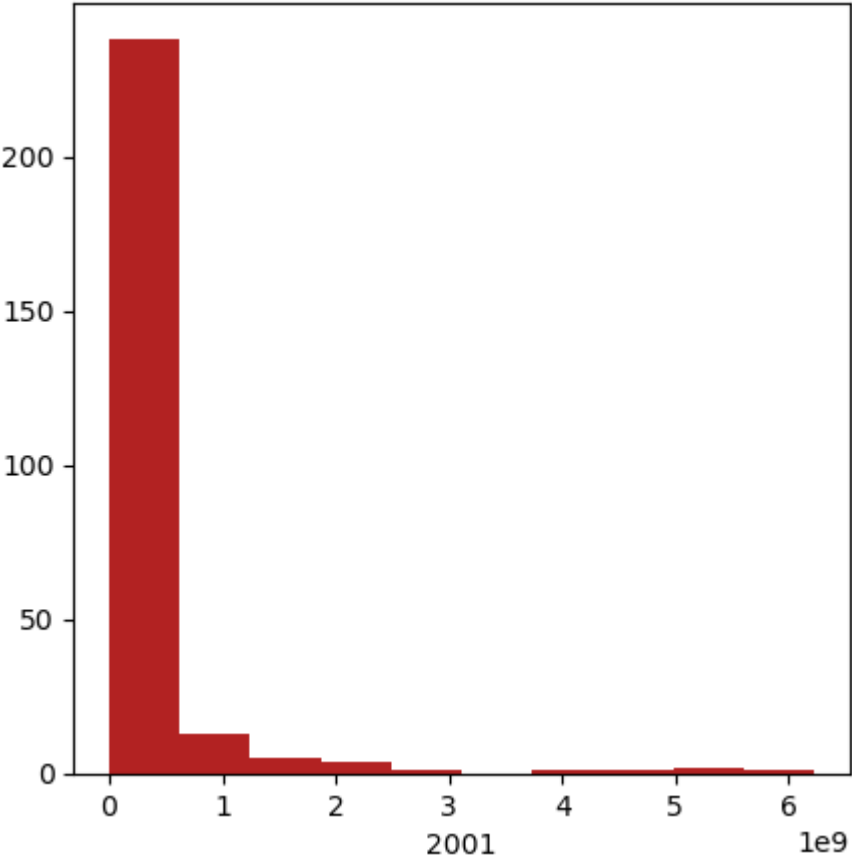


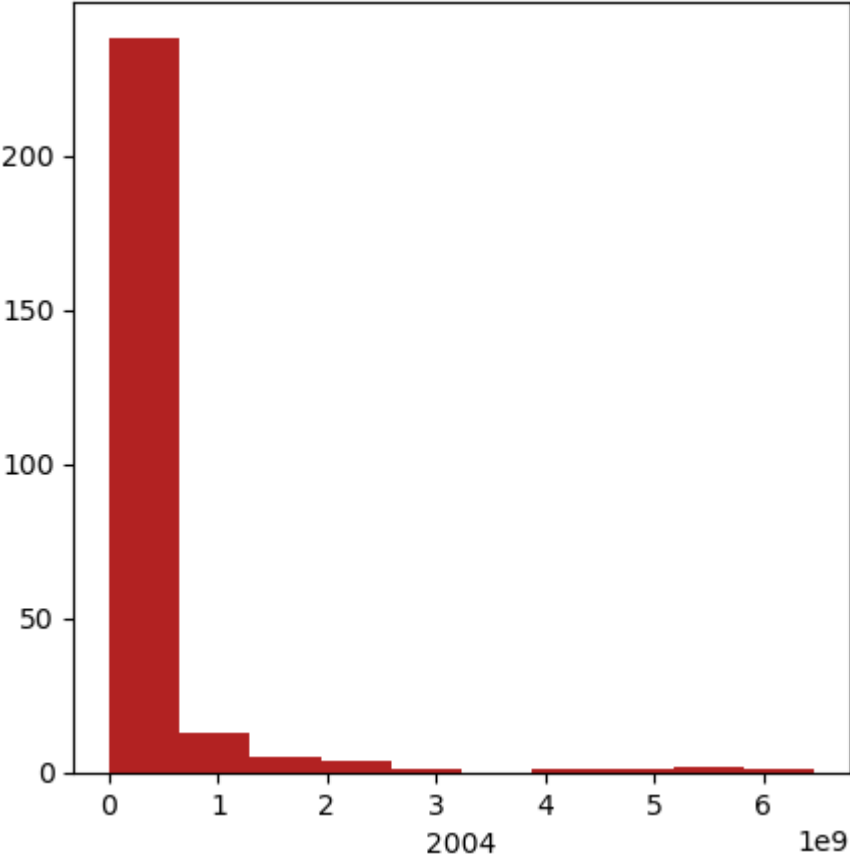
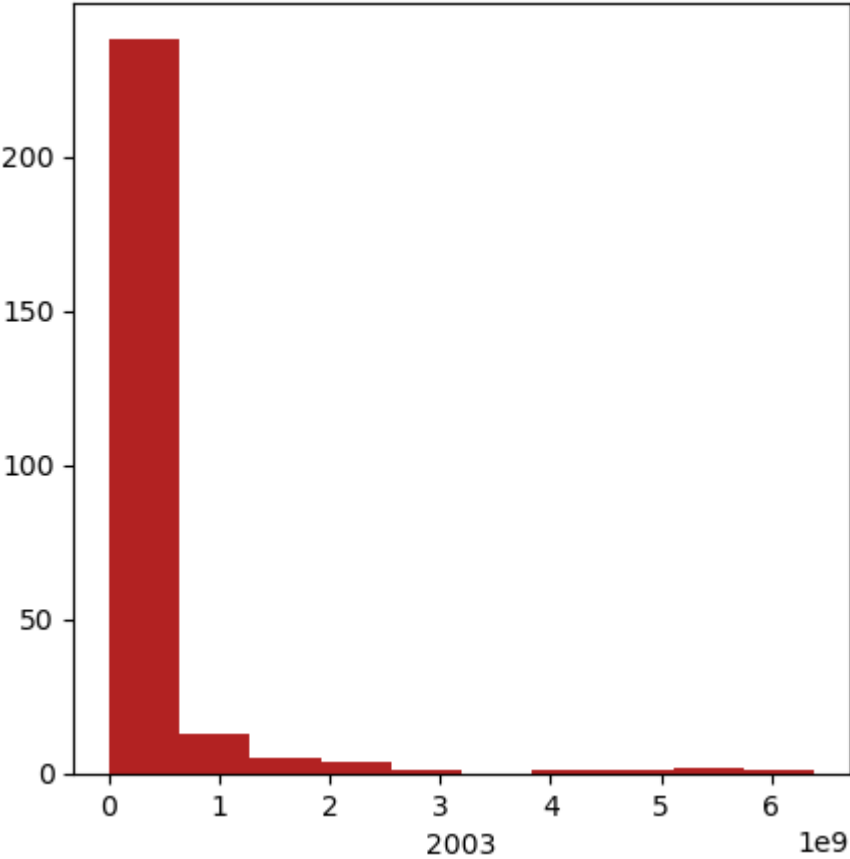


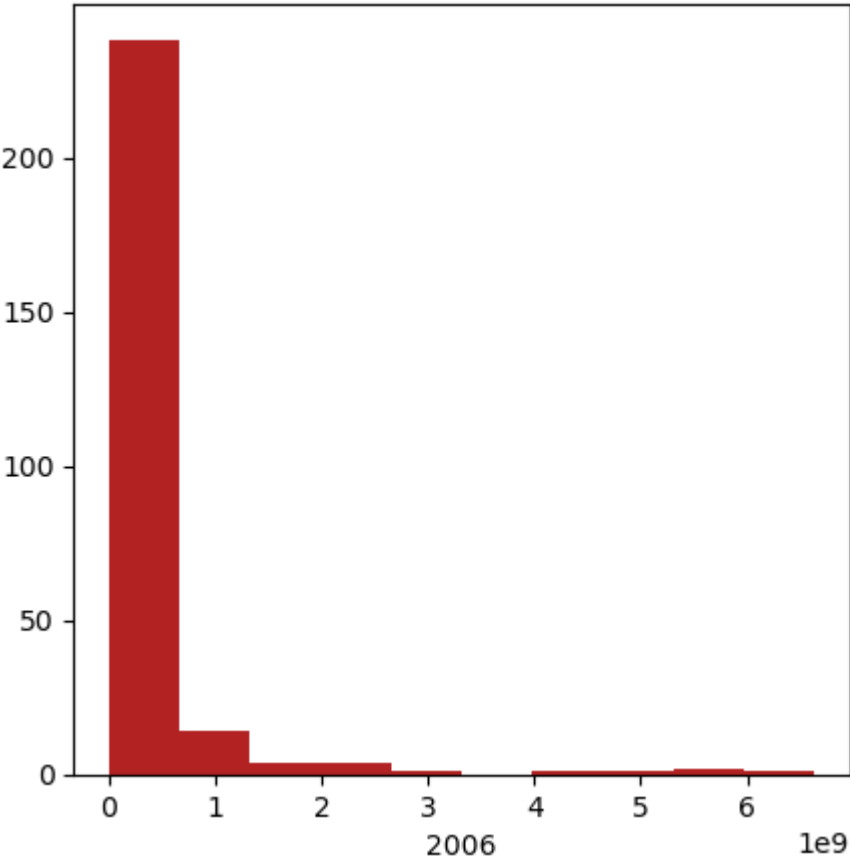
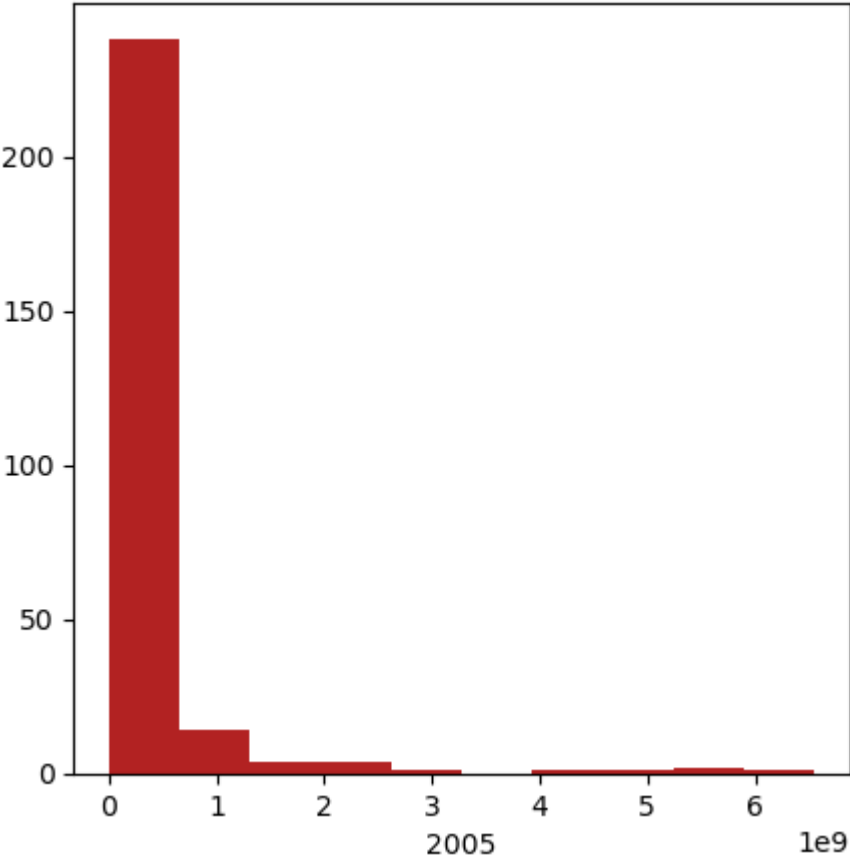


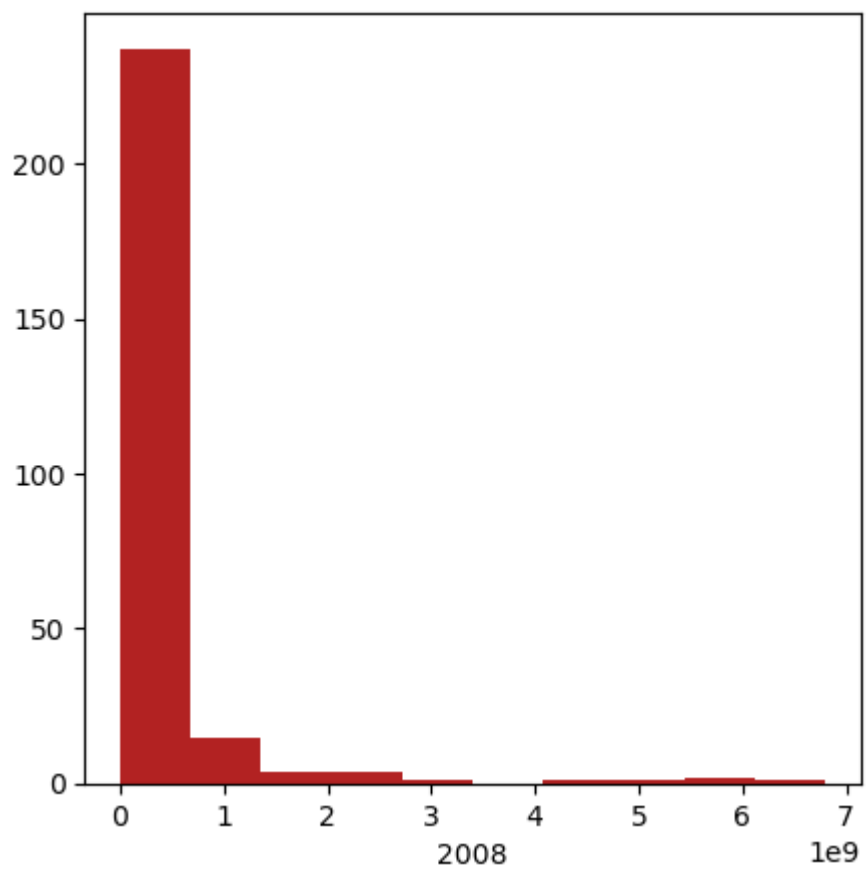
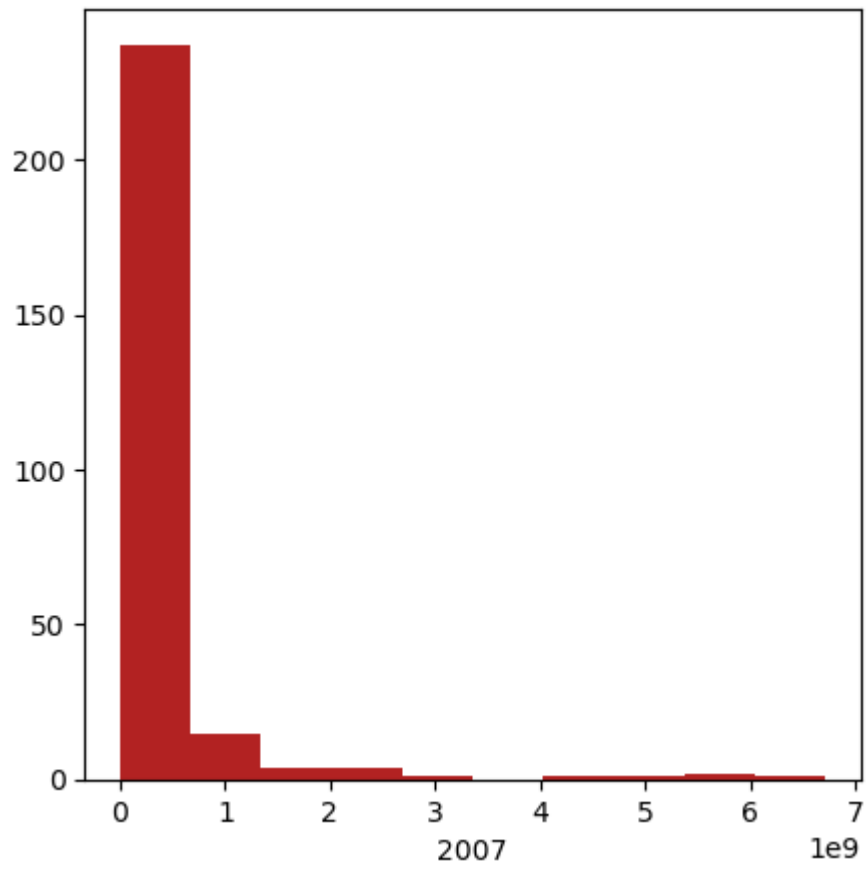


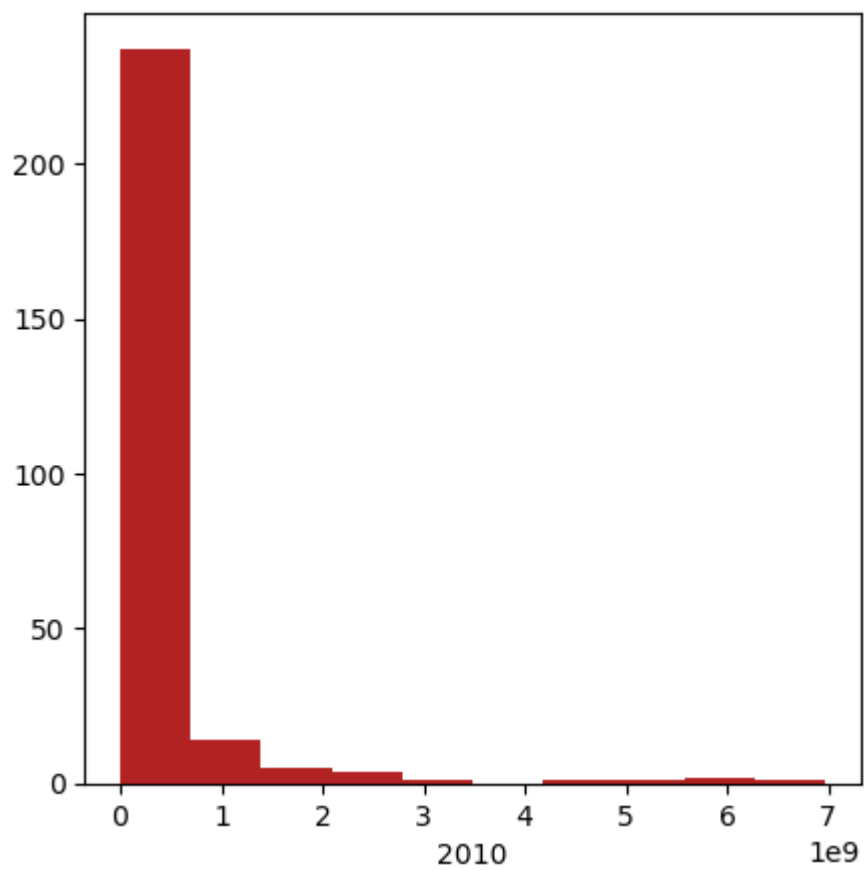
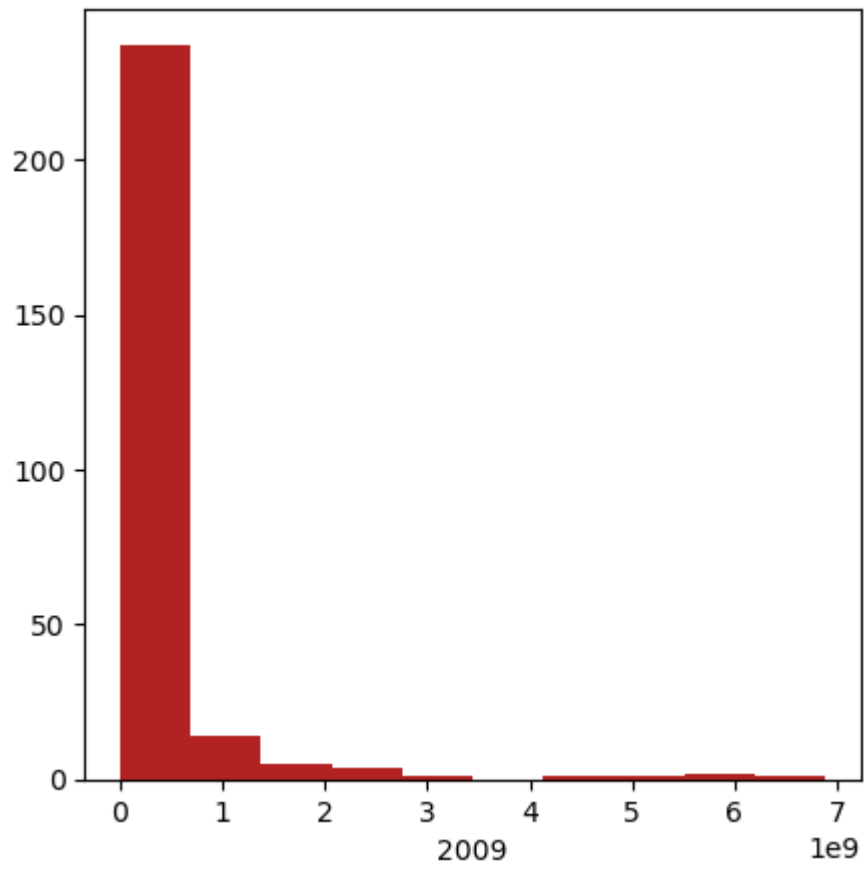


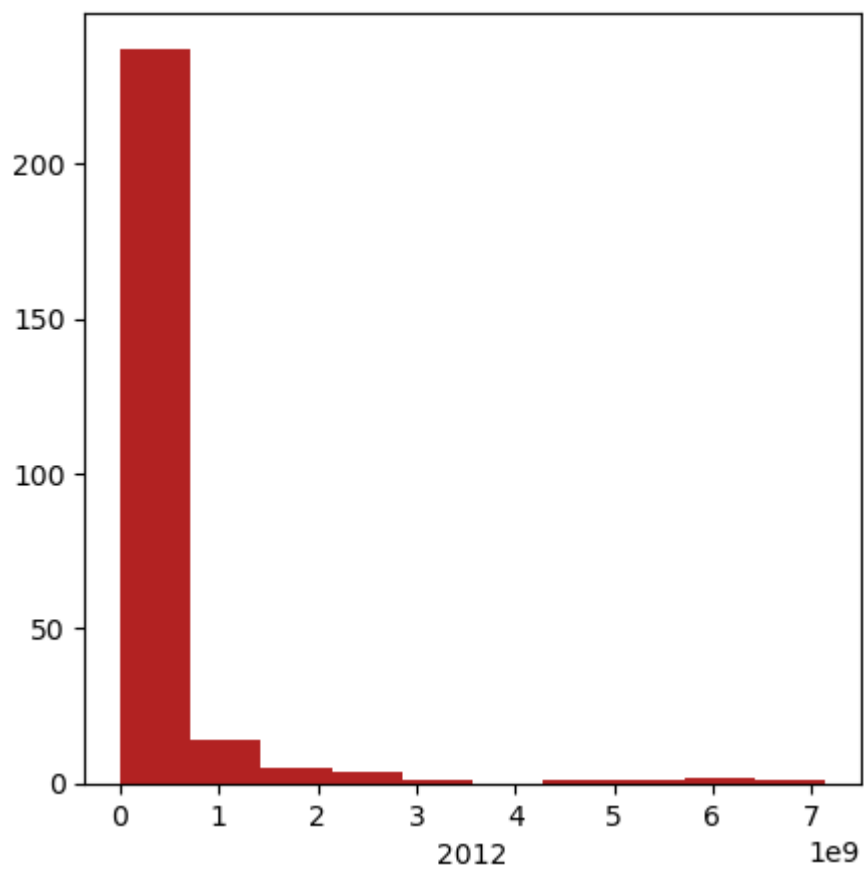
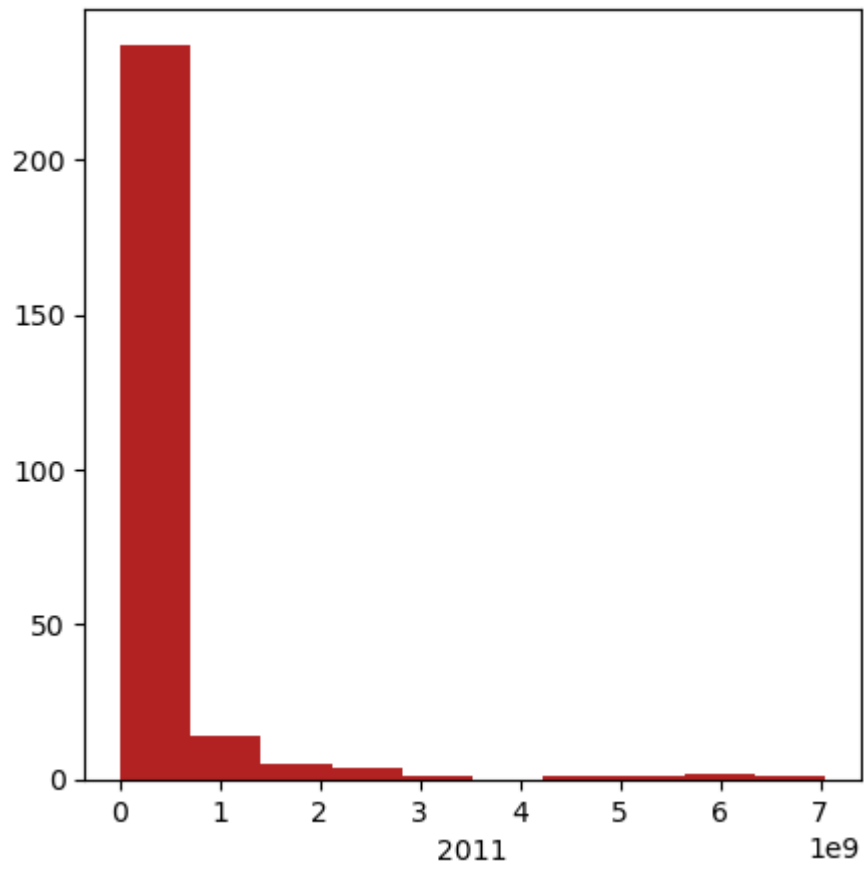


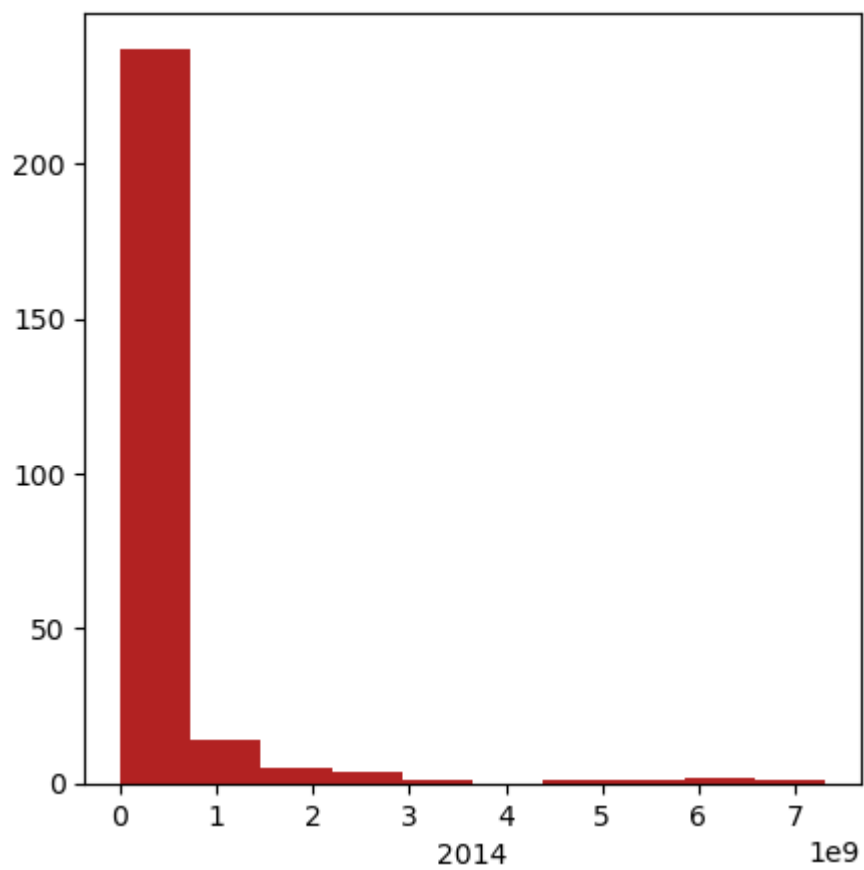
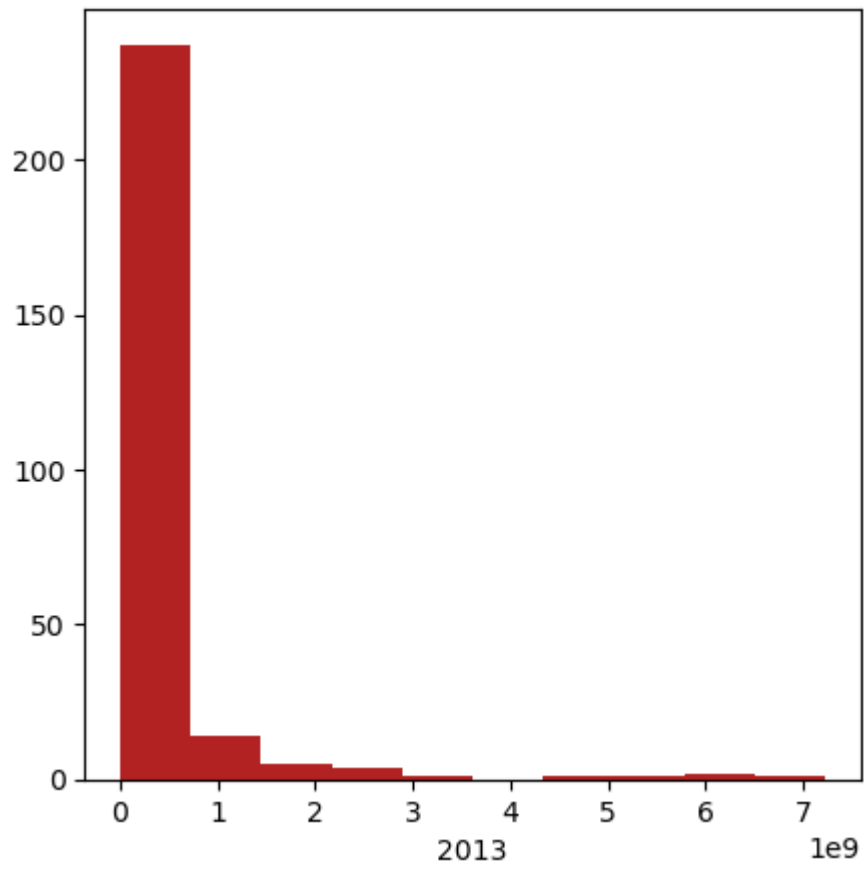


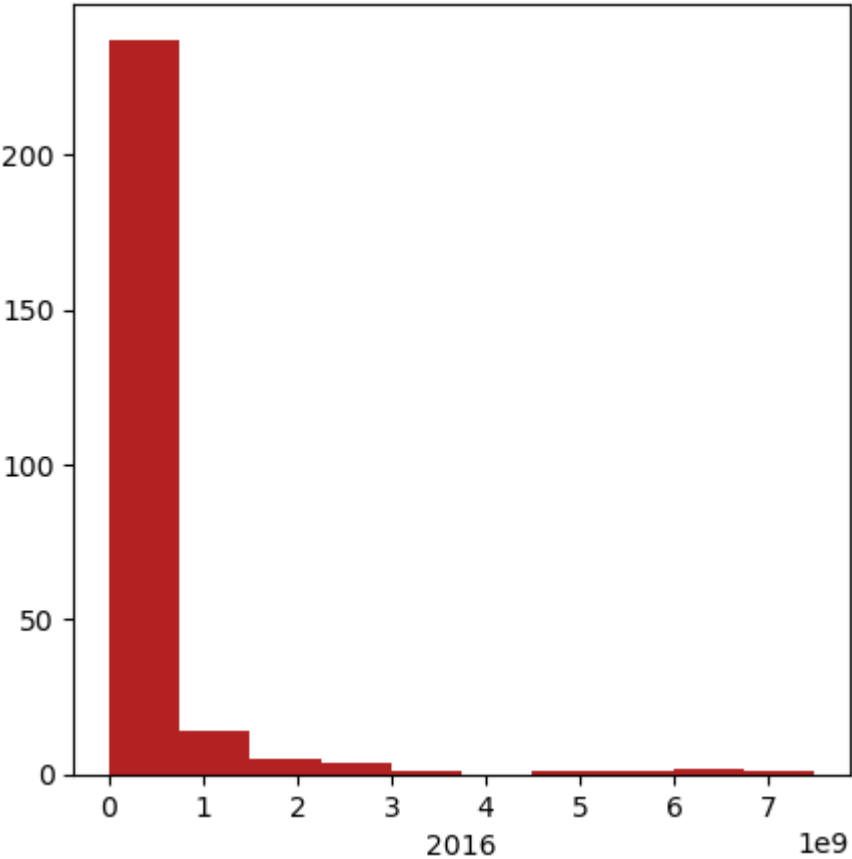
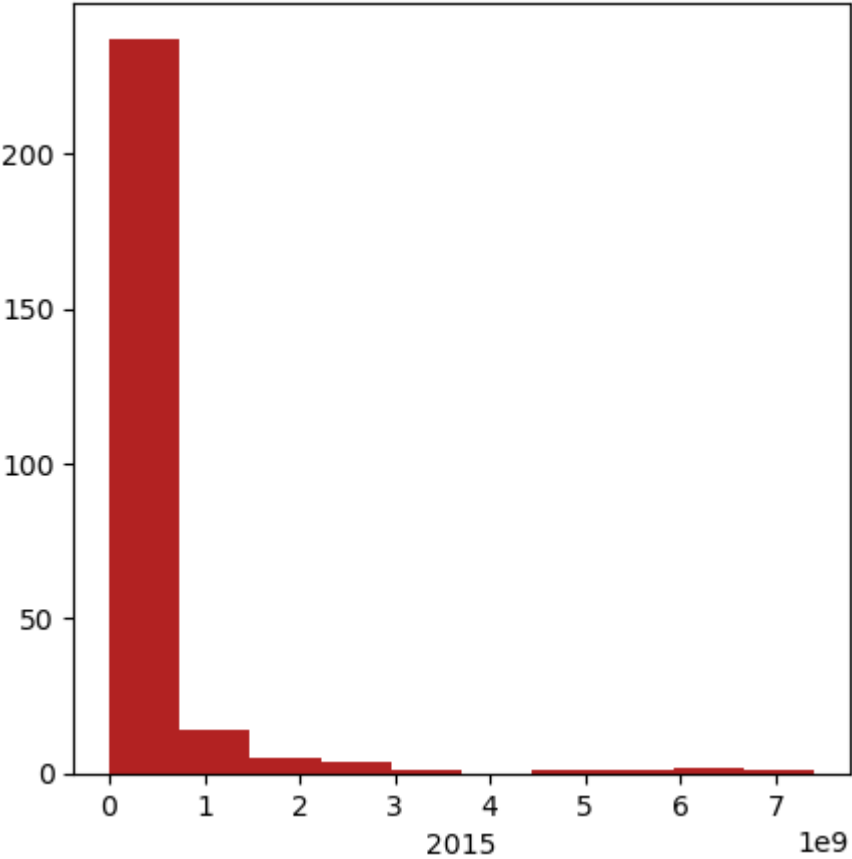


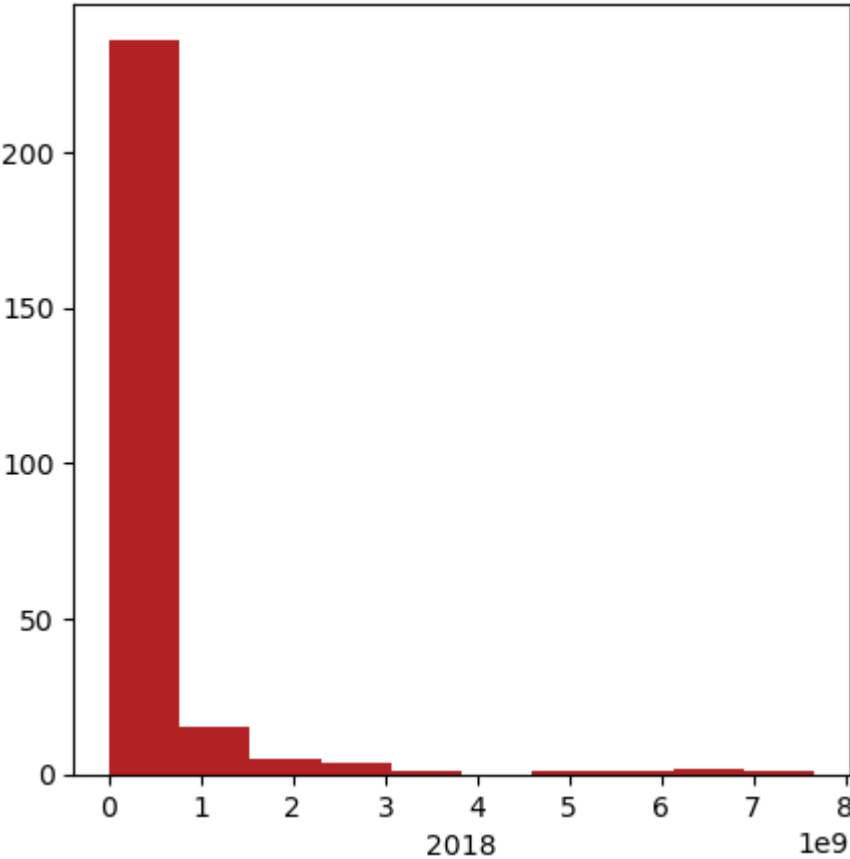
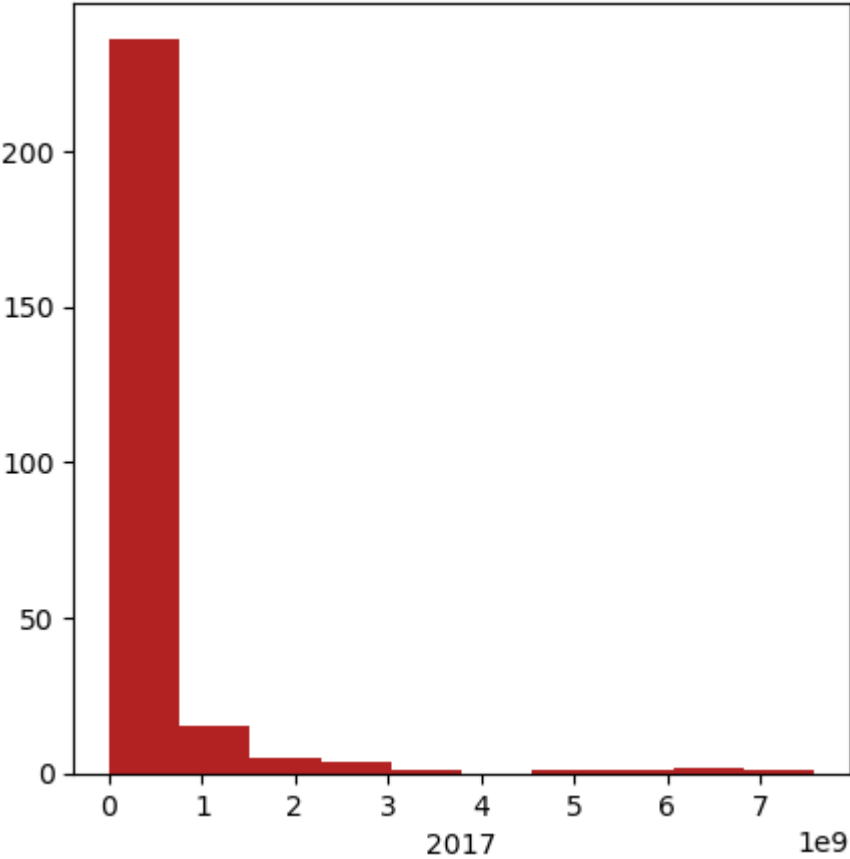


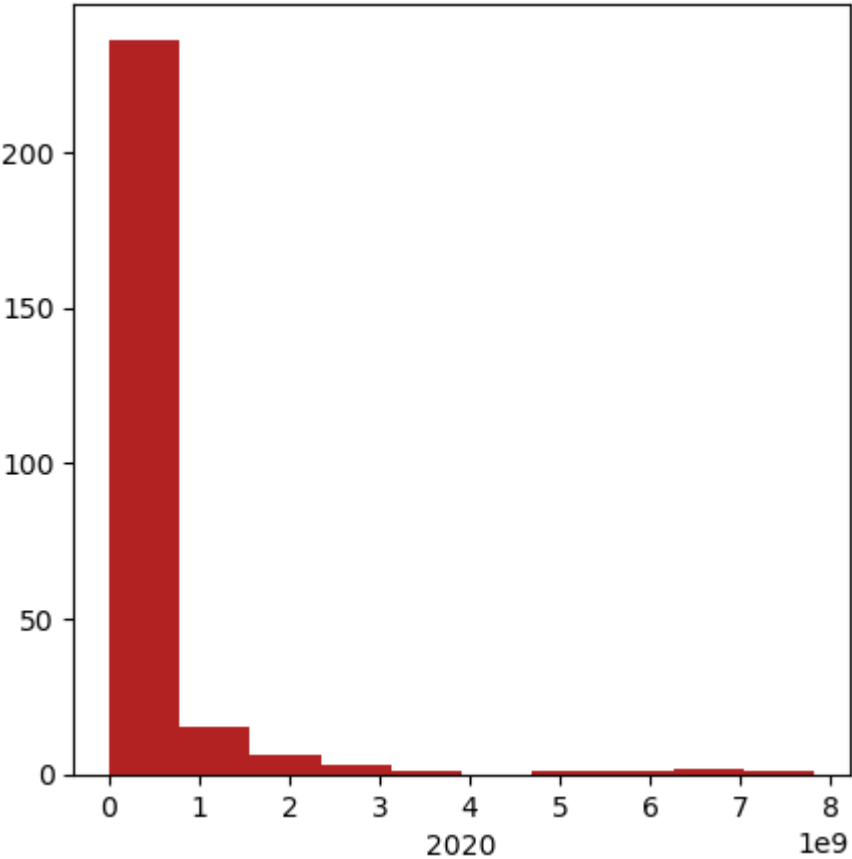
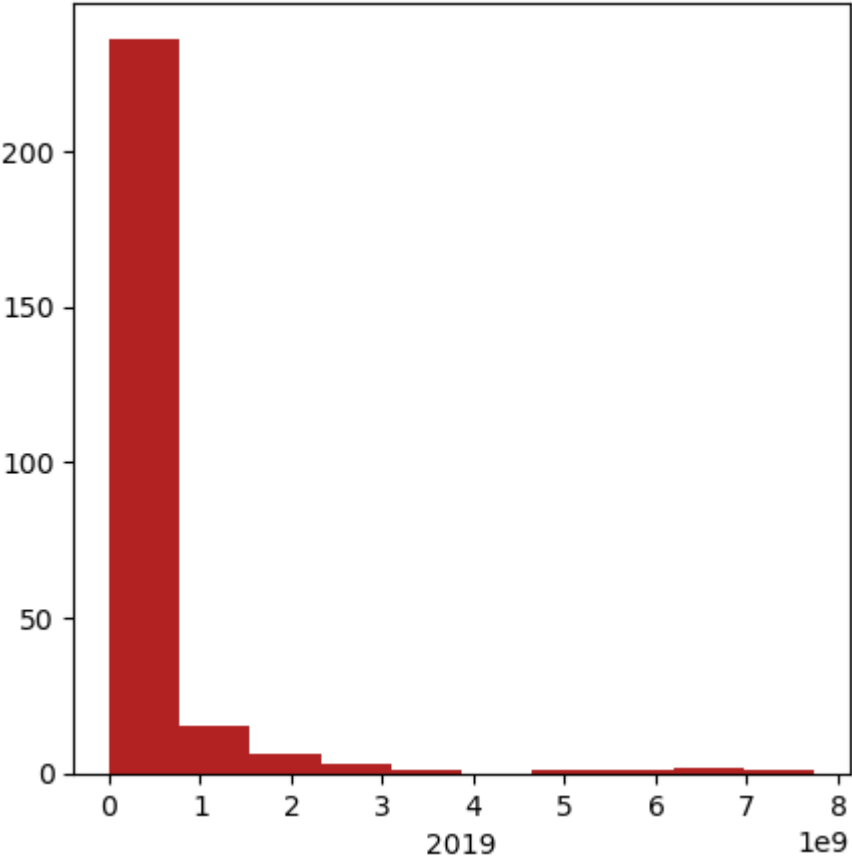


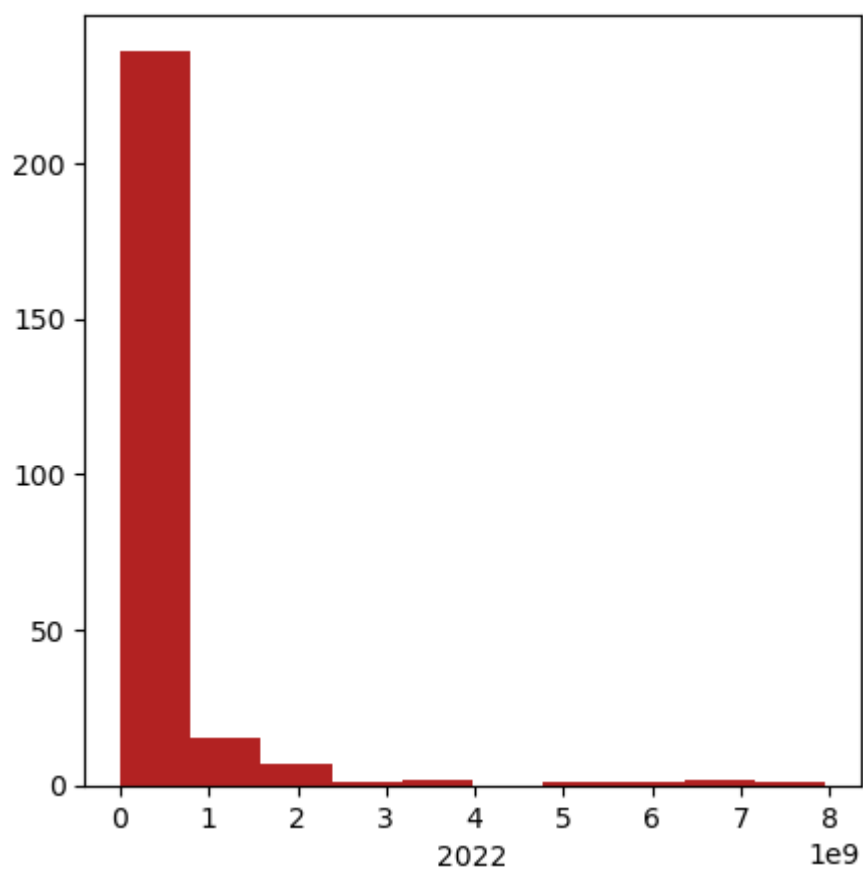
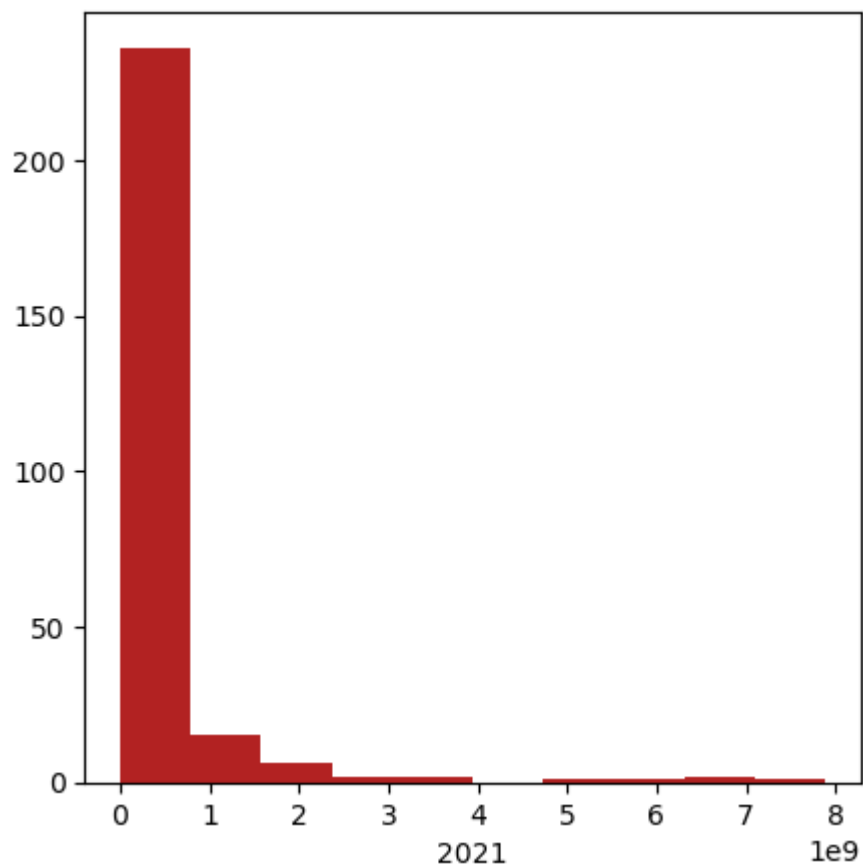








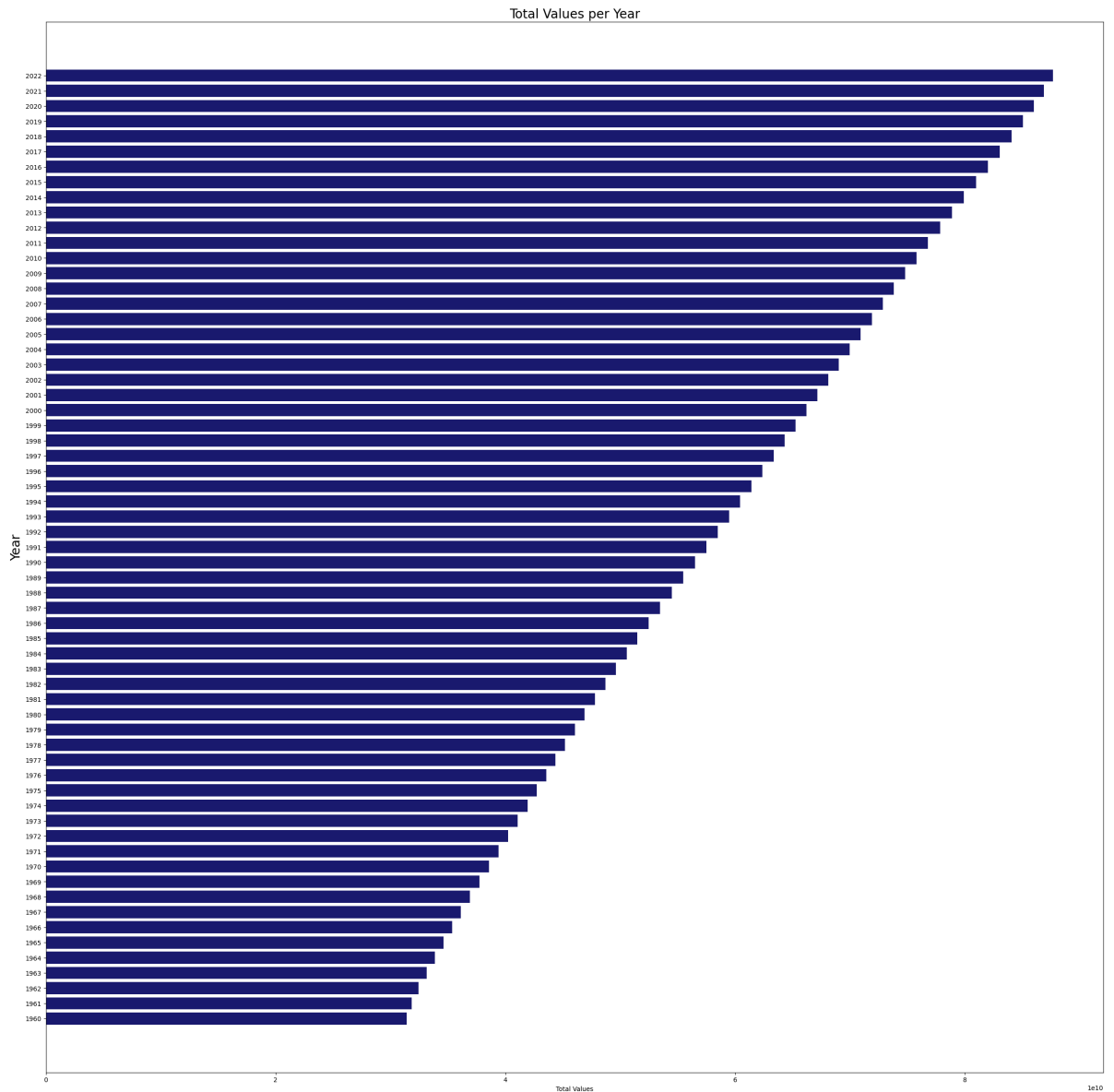




```
In [57]: years = df.columns[1:]  
  
total_values = df[years].sum()  
  
plt.figure(figsize=(30,30))  
plt.barh(years, total_values,color='#191970')  
plt.xlabel('Total Values')
```



```
plt.ylabel('Year', size=20)
plt.title('Total Values per Year', size=20)
plt.show()
```



```
In [58]: country_by_1960 = df.sort_values(by='1960').head(20)
country_by_1960
```

Out[58]:

	Country Name	1960	1961	1962	1963	1964	1965	1966	1967	1968
225	Sint Maarten (Dutch part)	2646.0	2888.0	3171.0	3481.0	3811.0	4161.0	4531.0	4930.0	5354.0
147	St. Martin (French part)	4135.0	4258.0	4388.0	4524.0	4666.0	4832.0	5044.0	5294.0	5497.0
179	Nauru	4582.0	4753.0	4950.0	5198.0	5484.0	5804.0	6021.0	6114.0	6288.0
245	Tuvalu	5404.0	5436.0	5471.0	5503.0	5525.0	5548.0	5591.0	5657.0	5729.0
228	Turks and Caicos Islands	5604.0	5625.0	5633.0	5634.0	5642.0	5650.0	5652.0	5662.0	5668.0
255	British Virgin Islands	7850.0	7885.0	7902.0	7919.0	7949.0	8018.0	8139.0	8337.0	8649.0
52	Cayman Islands	8473.0	8626.0	8799.0	8985.0	9172.0	9366.0	9566.0	9771.0	9981.0
164	Northern Mariana Islands	8702.0	8965.0	9252.0	9561.0	9890.0	10229.0	10577.0	10720.0	10440.0
6	Andorra	9443.0	10216.0	11014.0	11839.0	12690.0	13563.0	14546.0	15745.0	17079.0
188	Palau	9446.0	9639.0	9851.0	10076.0	10318.0	10563.0	10813.0	10992.0	11079.0
155	Marshall Islands	15374.0	15867.0	16387.0	16947.0	17537.0	18154.0	18794.0	19665.0	21001.0
212	San Marino	15556.0	15895.0	16242.0	16583.0	16926.0	17273.0	17588.0	17907.0	18291.0
137	Liechtenstein	16472.0	16834.0	17221.0	17625.0	18058.0	18500.0	18957.0	19467.0	20011.0
11	American Samoa	20085.0	20626.0	21272.0	21949.0	22656.0	23391.0	24122.0	24848.0	25608.0
149	Monaco	21797.0	21907.0	22106.0	22442.0	22766.0	23022.0	23198.0	23281.0	23481.0
84	Gibraltar	21822.0	21907.0	22249.0	22796.0	23347.0	23910.0	24477.0	25047.0	25610.0
91	Greenland	32500.0	33700.0	35000.0	36400.0	37600.0	39200.0	40500.0	41900.0	43400.0
256	Virgin Islands (U.S.)	32500.0	34300.0	35000.0	39800.0	40800.0	43500.0	46200.0	49100.0	55700.0
78	Faroe Islands	34154.0	34572.0	34963.0	35385.0	35841.0	36346.0	36825.0	37234.0	37630.0
200	Qatar	36385.0	40111.0	45123.0	50950.0	57531.0	64843.0	73102.0	82517.0	93022.0

20 rows × 64 columns



In [59]:

```
country_by_1960_t = country_by_1960.set_index('Country Name').T
for country_name, data_values in country_by_1960_t.iterrows():
    fig = plt.figure(figsize=(10,5))
    sns.barplot(x=data_values.index, y=data_values.values)
    plt.xlabel('Countries')
    plt.ylabel('Data Values')
    plt.title(f"{country_name} - Data Values from 1960 to 2022")
    plt.xticks(rotation=90)
    plt.show()
```

