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

In [2]:
```
df=pd.read_csv("C:\Data Analytics\The_WorldBank\API_SP.POP.TOTL_DS2_en_csv_
```

In [3]:
```
df
```

Out[3]:

	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
1	Africa Eastern and Southern	AFE	Population, total	SP.POP.TOTL	130692579.0	134169237.0	137835590.0
2	Afghanistan	AFG	Population, total	SP.POP.TOTL	8622466.0	8790140.0	8969047.0
3	Africa Western and Central	AFW	Population, total	SP.POP.TOTL	97256290.0	99314028.0	101445032.0
4	Angola	AGO	Population, total	SP.POP.TOTL	5357195.0	5441333.0	5521400.0
...	...	...	...	...	...	...	...
261	Kosovo	XKX	Population, total	SP.POP.TOTL	947000.0	966000.0	994000.0
262	Yemen, Rep.	YEM	Population, total	SP.POP.TOTL	5542459.0	5646668.0	5753386.0
263	South Africa	ZAF	Population, total	SP.POP.TOTL	16520441.0	16989464.0	17503133.0
264	Zambia	ZMB	Population, total	SP.POP.TOTL	3119430.0	3219451.0	3323427.0
265	Zimbabwe	ZWE	Population, total	SP.POP.TOTL	3806310.0	3925952.0	4049778.0

266 rows × 67 columns

In [4]:
```
df.head()
```

Out[4]:

	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	
1	Africa Eastern and Southern	AFE	Population, total	SP.POP.TOTL	130692579.0	134169237.0	137835590.0	14
2	Afghanistan	AFG	Population, total	SP.POP.TOTL	8622466.0	8790140.0	8969047.0	
3	Africa Western and Central	AFW	Population, total	SP.POP.TOTL	97256290.0	99314028.0	101445032.0	10
4	Angola	AGO	Population, total	SP.POP.TOTL	5357195.0	5441333.0	5521400.0	

5 rows × 67 columns

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

Out[5]:

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	
261	Kosovo	XKX	Population, total	SP.POP.TOTL	947000.0	966000.0	994000.0	102
262	Yemen, Rep.	YEM	Population, total	SP.POP.TOTL	5542459.0	5646668.0	5753386.0	586
263	South Africa	ZAF	Population, total	SP.POP.TOTL	16520441.0	16989464.0	17503133.0	1804
264	Zambia	ZMB	Population, total	SP.POP.TOTL	3119430.0	3219451.0	3323427.0	343
265	Zimbabwe	ZWE	Population, total	SP.POP.TOTL	3806310.0	3925952.0	4049778.0	417

5 rows × 67 columns

In [6]:
```
df.shape
```

Out[6]:
```
(266, 67)
```

In [7]:
```
df.columns
```

Out[7]:
```
Index(['Country Name', 'Country Code', 'Indicator Name', 'Indicator Code',
       '1960', '1961', '1962', '1963', '1964', '1965', '1966', '1967', '19
68',
       '1969', '1970', '1971', '1972', '1973', '1974', '1975', '1976', '19
77',
       '1978', '1979', '1980', '1981', '1982', '1983', '1984', '1985', '19
86',
       '1987', '1988', '1989', '1990', '1991', '1992', '1993', '1994', '19
95',
       '1996', '1997', '1998', '1999', '2000', '2001', '2002', '2003', '20
04',
       '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012', '20
13',
       '2014', '2015', '2016', '2017', '2018', '2019', '2020', '2021', '20
22'],
      dtype='object')
```

In [9]:
```
df.dtypes
```

Out[9]:
```
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 [10]:
```
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 [11]:
```
df.describe()
```

Out[11]:

	1960	1961	1962	1963	1964	1965
count	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
std	3.695439e+08	3.740897e+08	3.808061e+08	3.895039e+08	3.982439e+08	4.071153e+08
min	2.646000e+03	2.888000e+03	3.171000e+03	3.481000e+03	3.811000e+03	4.161000e+03
25%	5.132212e+05	5.231345e+05	5.337595e+05	5.449288e+05	5.566630e+05	5.651150e+05
50%	3.757486e+06	3.887144e+06	4.023896e+06	4.139356e+06	4.224612e+06	4.277636e+06
75%	2.670606e+07	2.748694e+07	2.830289e+07	2.914708e+07	3.001684e+07	3.084892e+07
max	3.031474e+09	3.072422e+09	3.126850e+09	3.193429e+09	3.260442e+09	3.328209e+09

8 rows × 63 columns

In [12]:
```
df.duplicated().sum()
```

Out[12]:
```
0
```

In [13]:
```
df.isna().sum().any()
```

Out[13]:
```
True
```

In [14]:
```
df=df.fillna(method="ffill")
df.head()
```

Out[14]:

	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	
1	Africa Eastern and Southern	AFE	Population, total	SP.POP.TOTL	130692579.0	134169237.0	137835590.0	14
2	Afghanistan	AFG	Population, total	SP.POP.TOTL	8622466.0	8790140.0	8969047.0	
3	Africa Western and Central	AFW	Population, total	SP.POP.TOTL	97256290.0	99314028.0	101445032.0	10
4	Angola	AGO	Population, total	SP.POP.TOTL	5357195.0	5441333.0	5521400.0	

5 rows × 67 columns

In [15]:
```
df.isna().sum().any()
```

Out[15]:
```
False
```

In [16]:
```
df['Country Name'].unique()
```

Out[16]:
```
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 Island
s',
       'Chile', 'China', "Cote d'Ivoire", 'Cameroon', 'Congo, Dem. Re
p.',
       'Congo, Rep.', 'Colombia', 'Comoros', 'Cabo Verde', 'Costa Ric
a',
       'Caribbean small states', 'Cuba', 'Curacao', 'Cayman Islands',
       'Cyprus', 'Czechia', 'Germany', 'Djibouti', 'Dominica', 'Denmar
k',
       'Dominican Republic', 'Algeria',
```

In [17]:
```
df['Country Code'].unique()
```

Out[17]:
```
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',
```

In [18]:
```
df['Indicator Name'].unique()
```

Out[18]:
```
array(['Population, total'], dtype=object)
```

In [19]:
```
df['Indicator Code'].unique()
```

Out[19]:
```
array(['SP.POP.TOTL'], dtype=object)
```

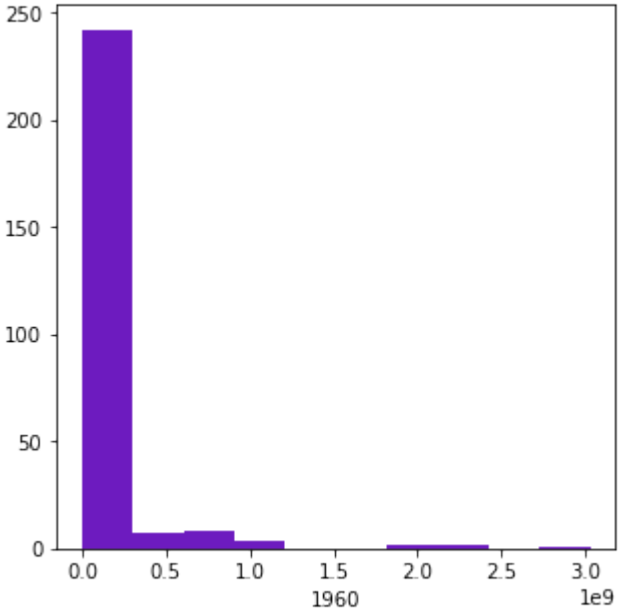
In [20]:
```
df.drop(['Indicator Name','Indicator Code','Country Code'],axis=1, inplace=
```

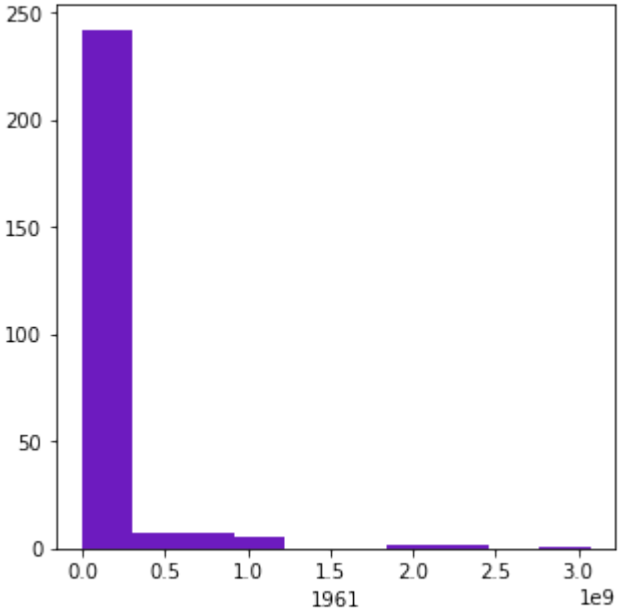
In [21]:
```
df.columns
```

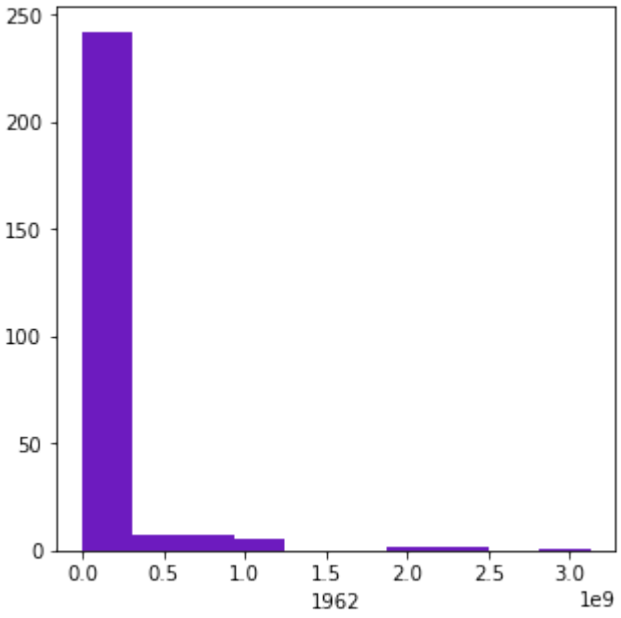
Out[21]:
```
Index(['Country Name', '1960', '1961', '1962', '1963', '1964', '1965', '19
66',
       '1967', '1968', '1969', '1970', '1971', '1972', '1973', '1974', '19
75',
       '1976', '1977', '1978', '1979', '1980', '1981', '1982', '1983', '19
84',
       '1985', '1986', '1987', '1988', '1989', '1990', '1991', '1992', '19
93',
       '1994', '1995', '1996', '1997', '1998', '1999', '2000', '2001', '20
02',
       '2003', '2004', '2005', '2006', '2007', '2008', '2009', '2010', '20
11',
       '2012', '2013', '2014', '2015', '2016', '2017', '2018', '2019', '20
20',
       '2021', '2022'],
      dtype='object')
```

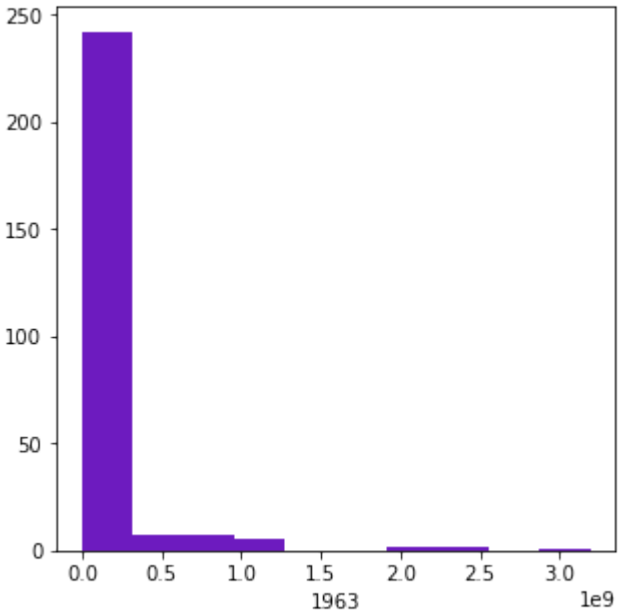
In [22]:
```
cols = ['1960', '1961', '1962', '1963', '1964', '1965', '1966',
        '1967', '1968', '1969', '1970', '1971', '1972', '1973', '1974', '19
        '1976', '1977', '1978', '1979', '1980', '1981', '1982', '1983', '19
        '1985', '1986', '1987', '1988', '1989', '1990', '1991', '1992', '19
        '1994', '1995', '1996', '1997', '1998', '1999', '2000','2001', '200
        '2008', '2009', '2010', '2011', '2012', '2013', '2014', '2015', '20
        '2017', '2018', '2019', '2020', '2021', '2022']
```

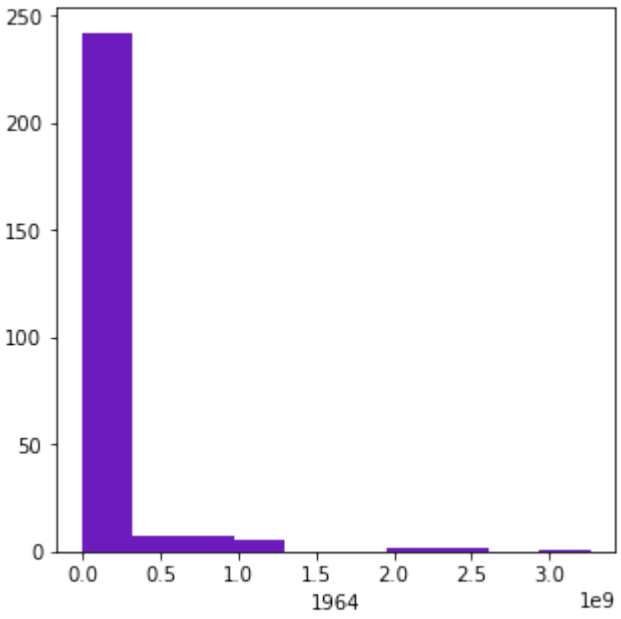
In [23]:
```
for i in cols:
    fig = plt.figure(figsize=(5,5))
    plt.hist(df[i],color='#6d1bbf',bins=10)
    plt.xlabel(i)
    plt.show()
```

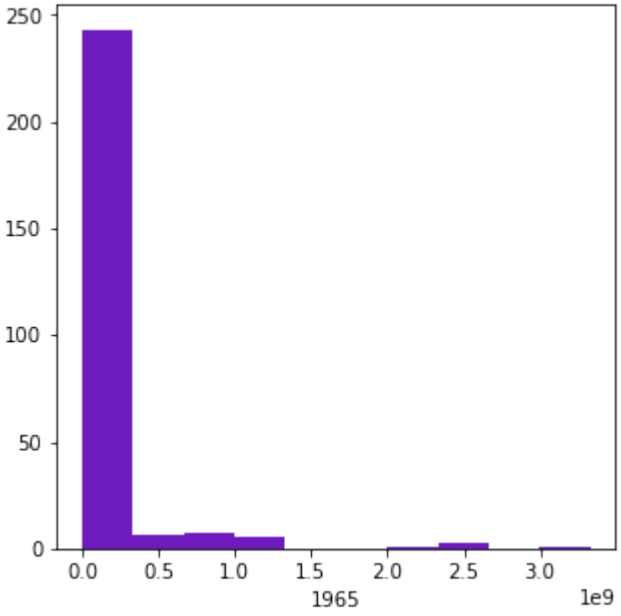


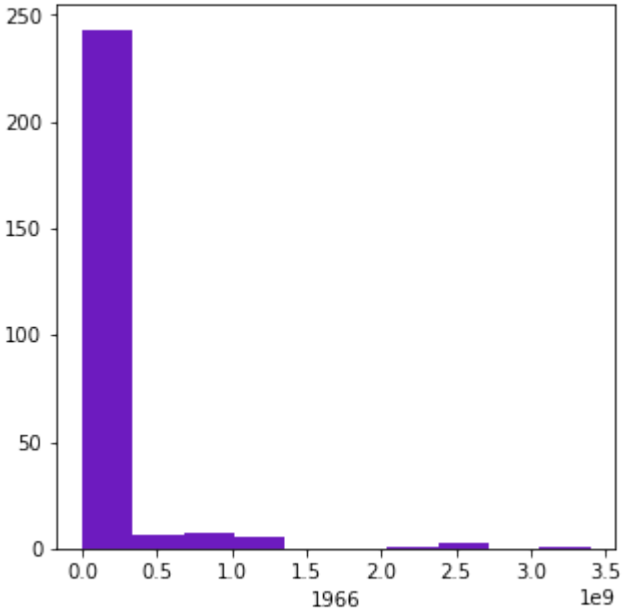


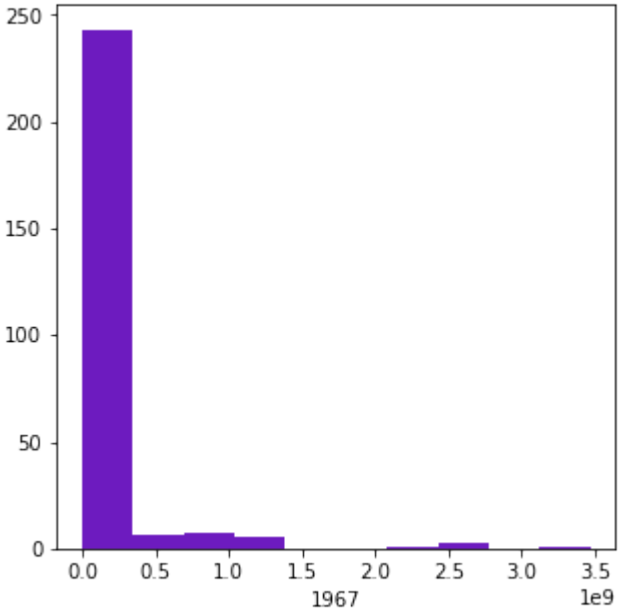

250
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1962
1e9

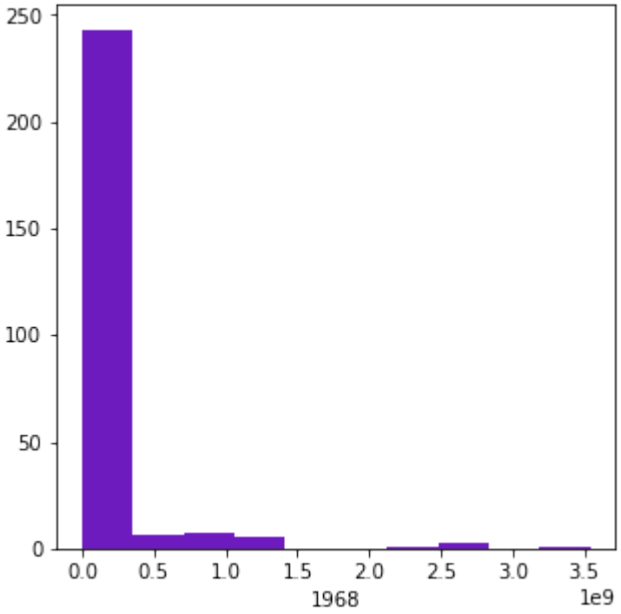

250
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1963
1e9

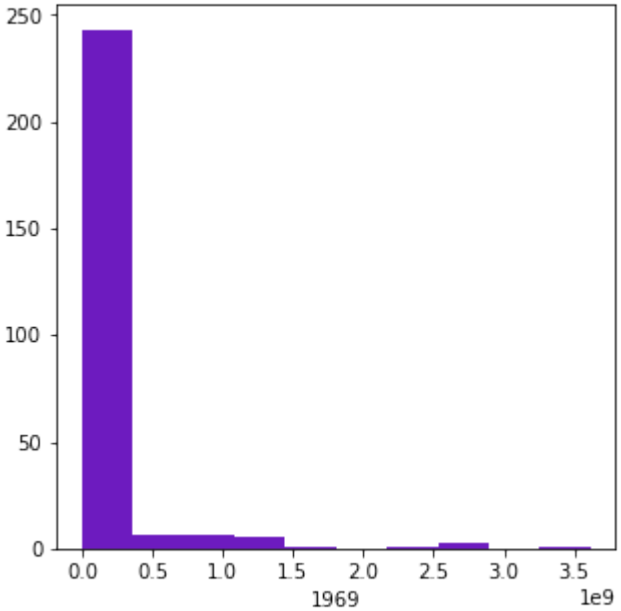

250
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1964
1e9

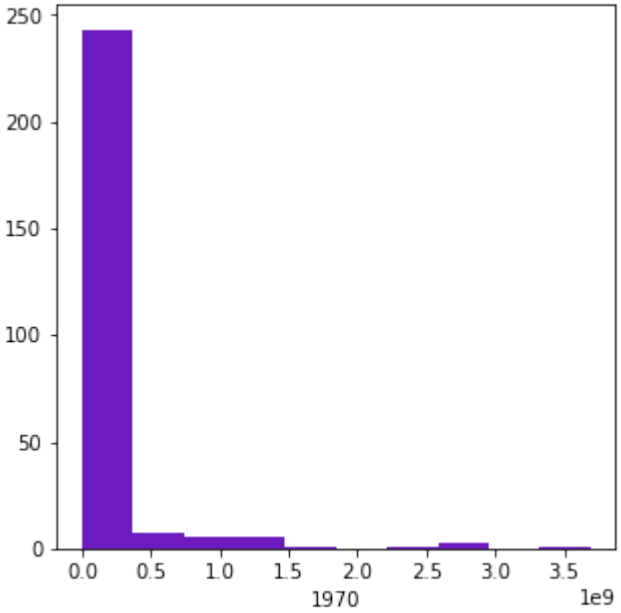

250
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1965
1e9

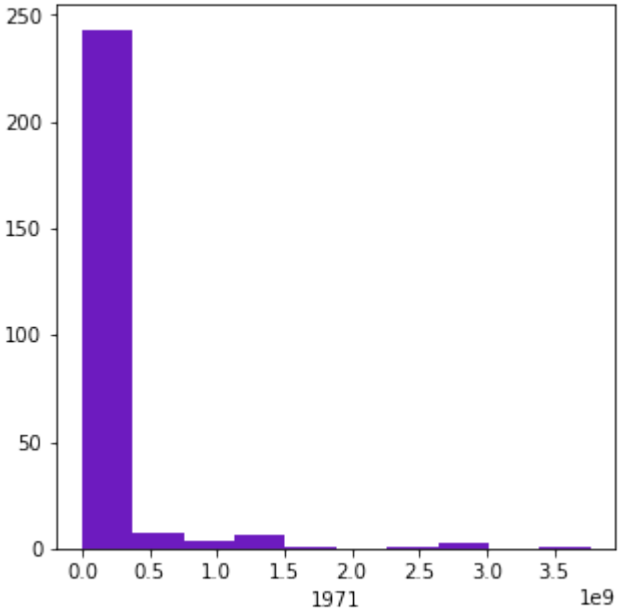

250
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
1966
1e9

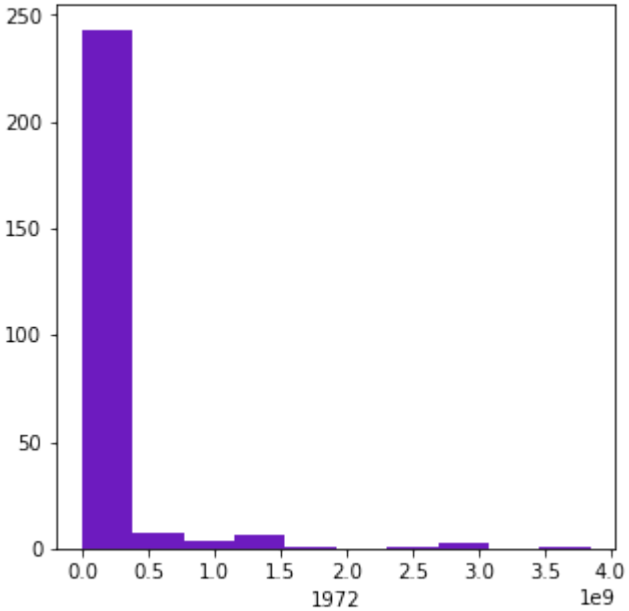

250
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
1967
1e9

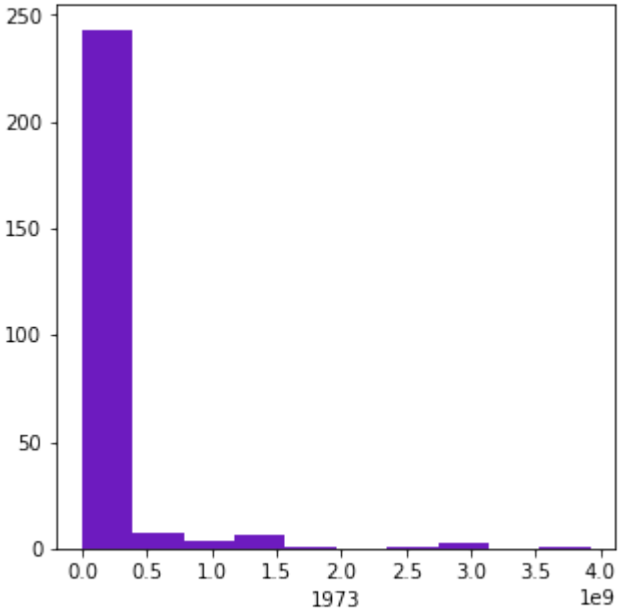

250
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
1968
1e9

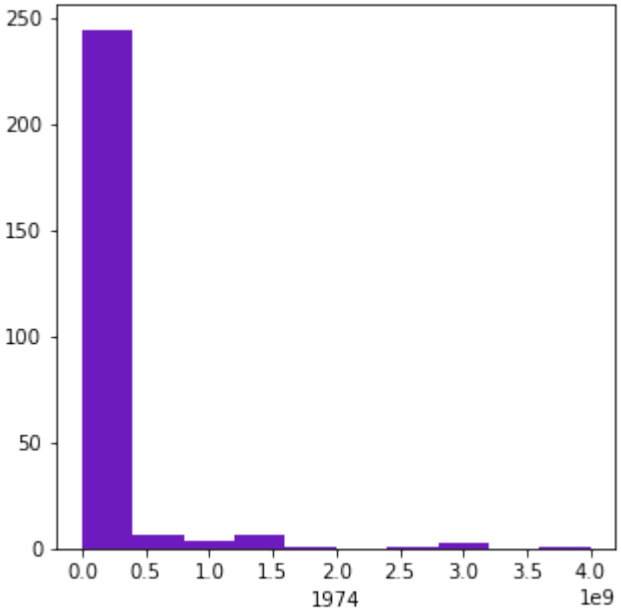

250
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
1969
1e9

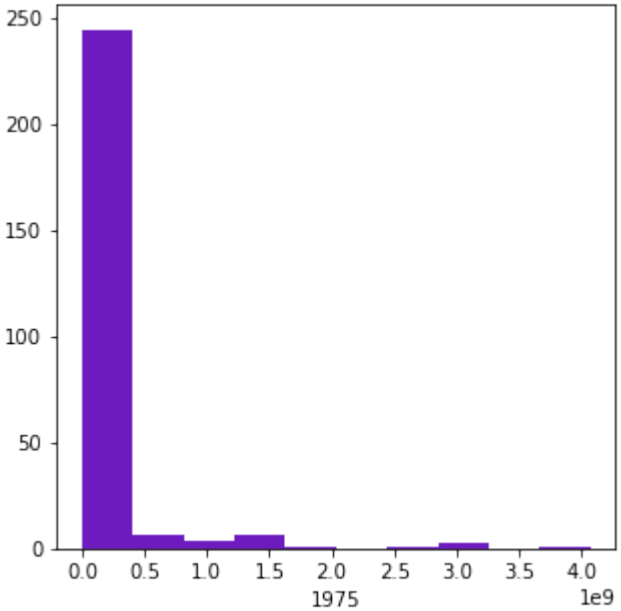

250
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
1970
1e9

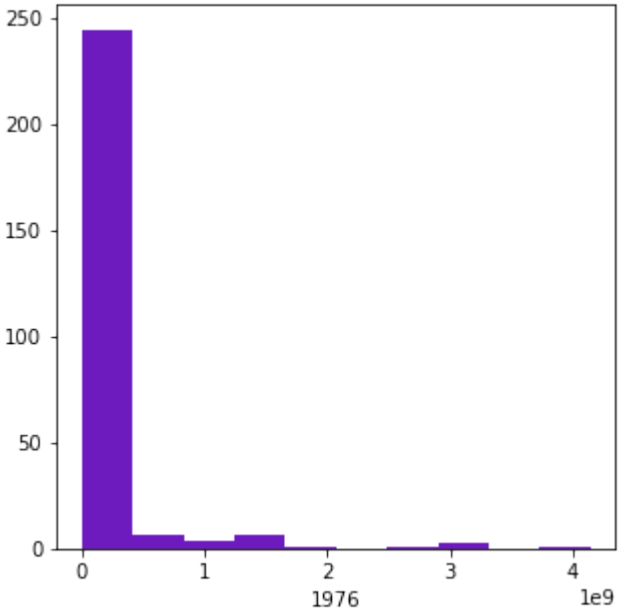

250
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
1971
1e9

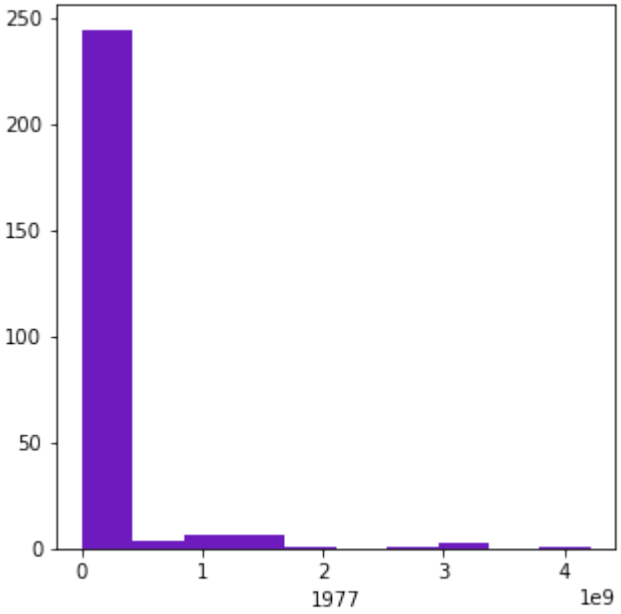

250
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
1972
1e9

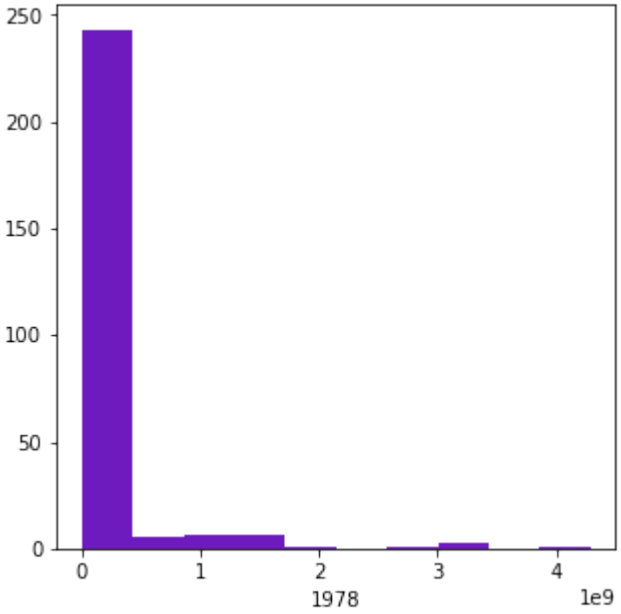

250
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
1973
1e9

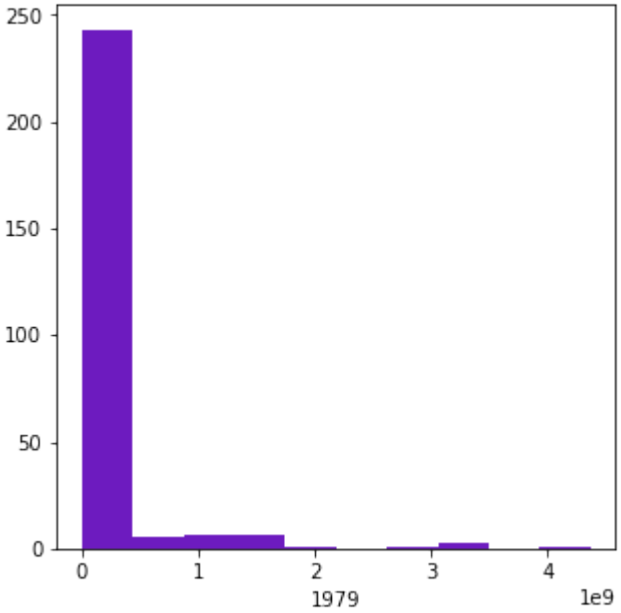

250
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
1974
1e9

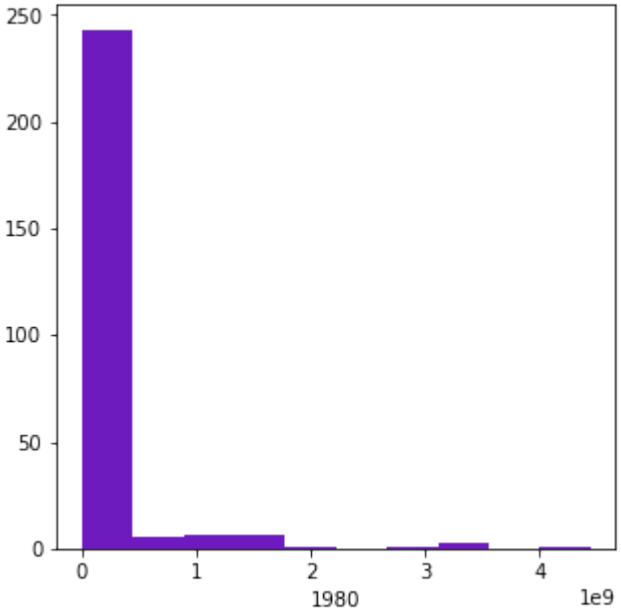

250
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
1975
1e9

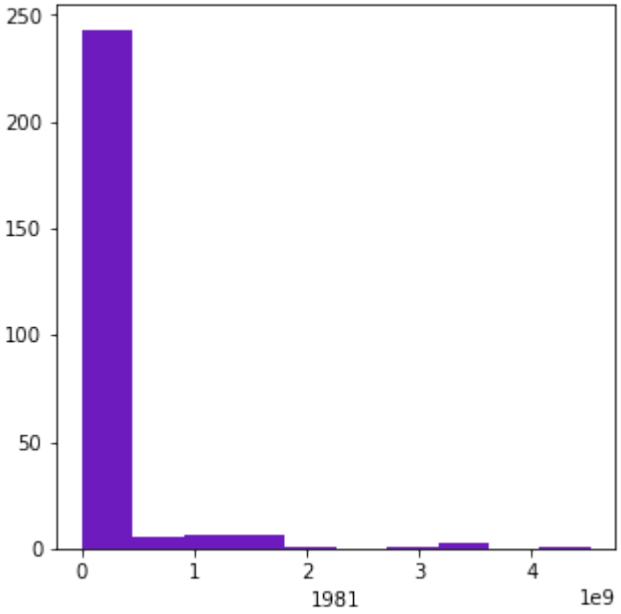

250
200
150
100
50
0
0
1
2
3
4
1976
1e9

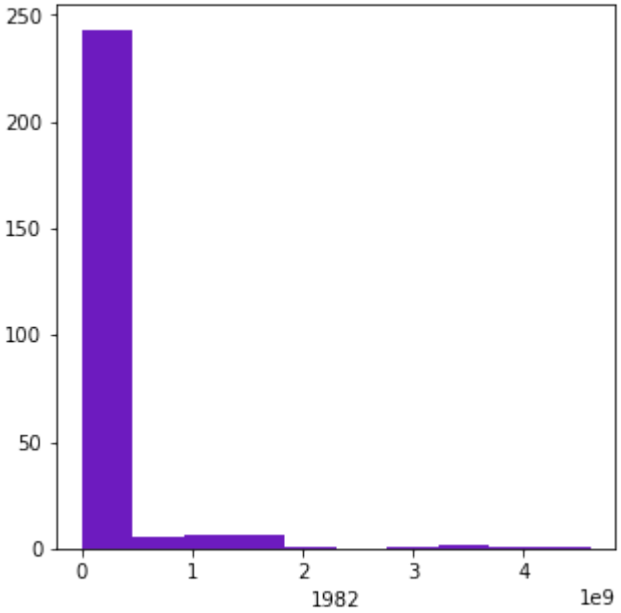

250
200
150
100
50
0
0
1
2
3
4
1977
1e9

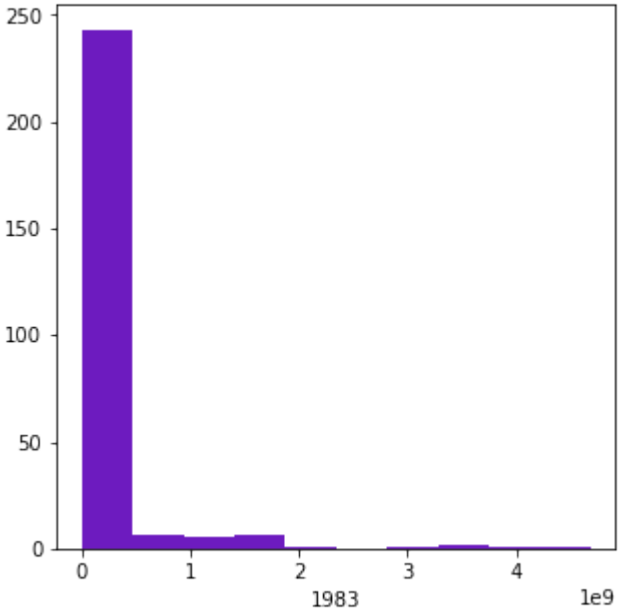

250
200
150
100
50
0
0
1
2
3
4
1978
1e9

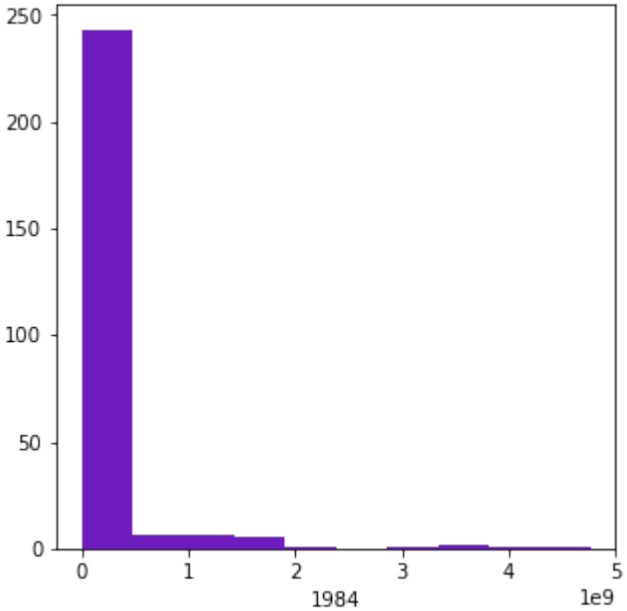

250
200
150
100
50
0
0
1
2
3
4
1979
1e9

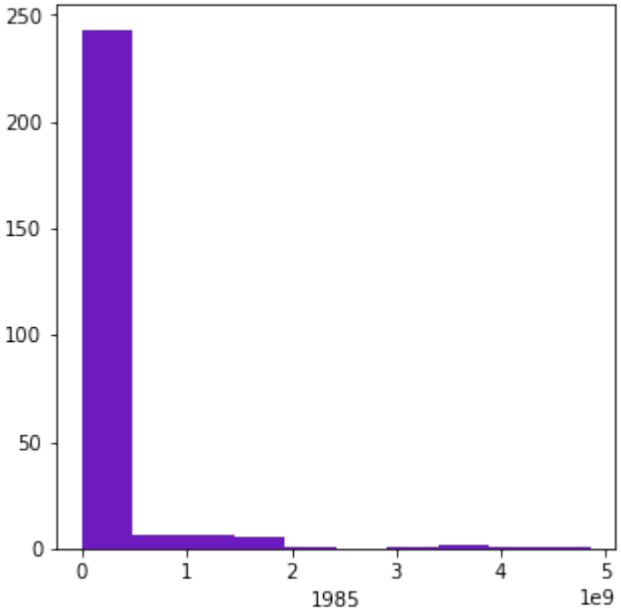

250
200
150
100
50
0
0
1
2
3
4
1980
1e9

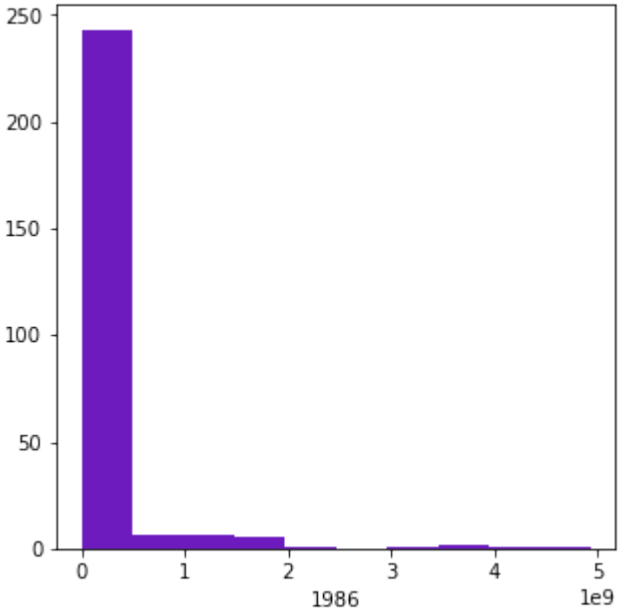

250
200
150
100
50
0
0
1
2
3
4
1981
1e9

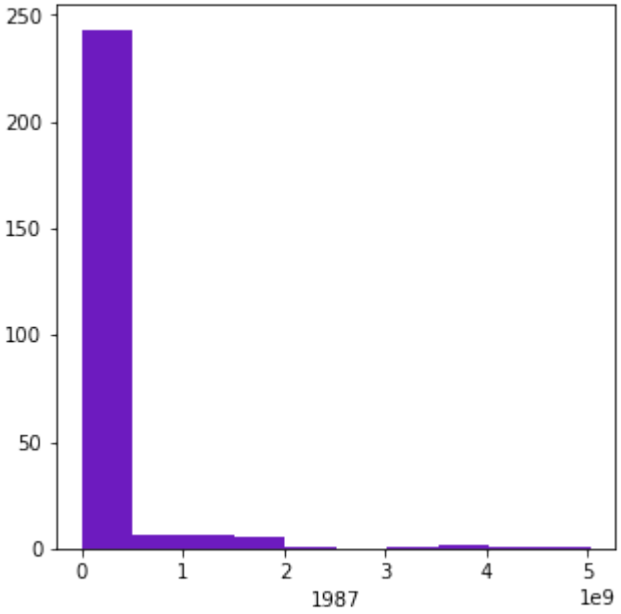

250
200
150
100
50
0
0
1
2
3
4
1982
1e9

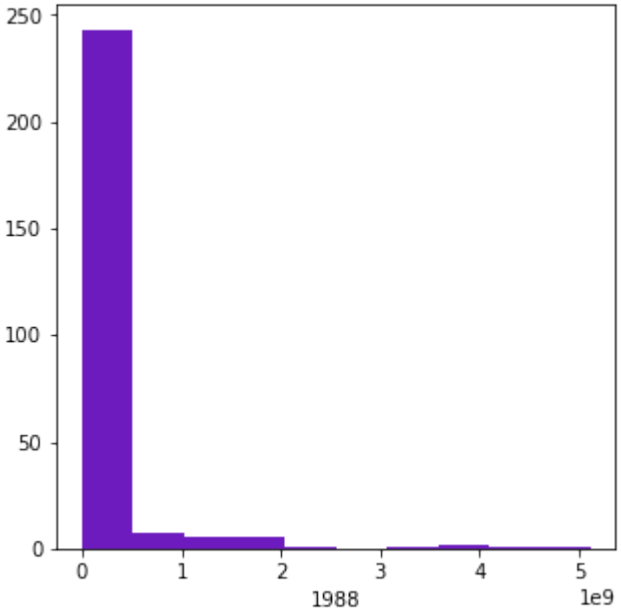

250
200
150
100
50
0
0
1
2
3
4
1983
1e9

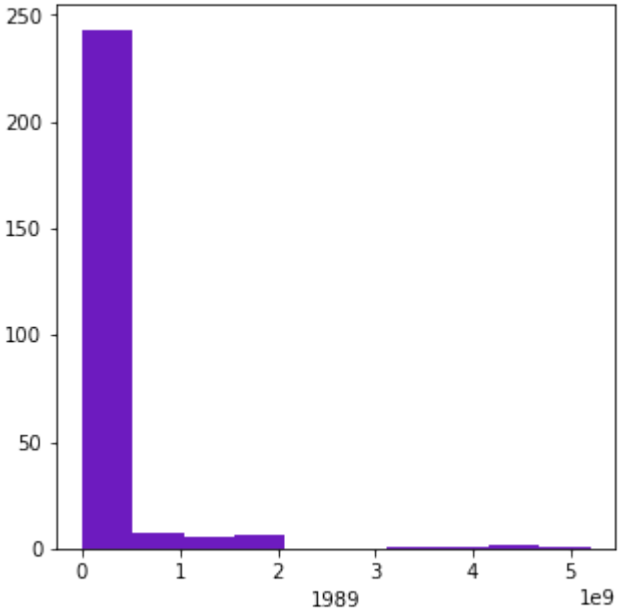

250
200
150
100
50
0
0
1
2
3
4
5
1984
1e9

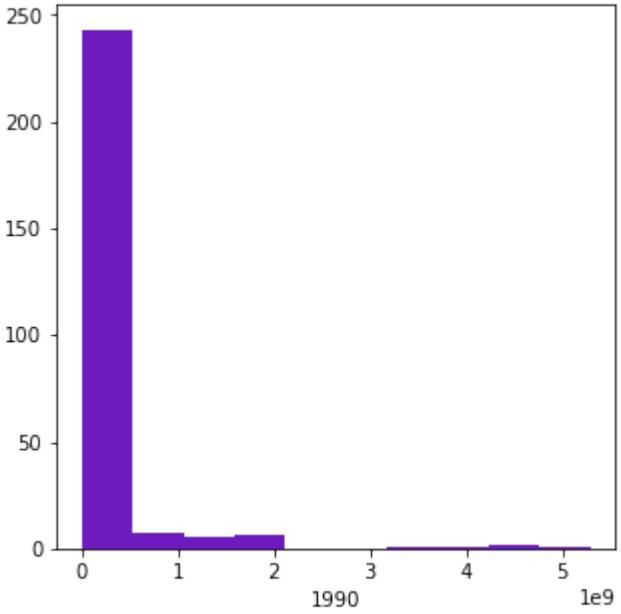

250
200
150
100
50
0
0
1
2
3
4
5
1985
1e9

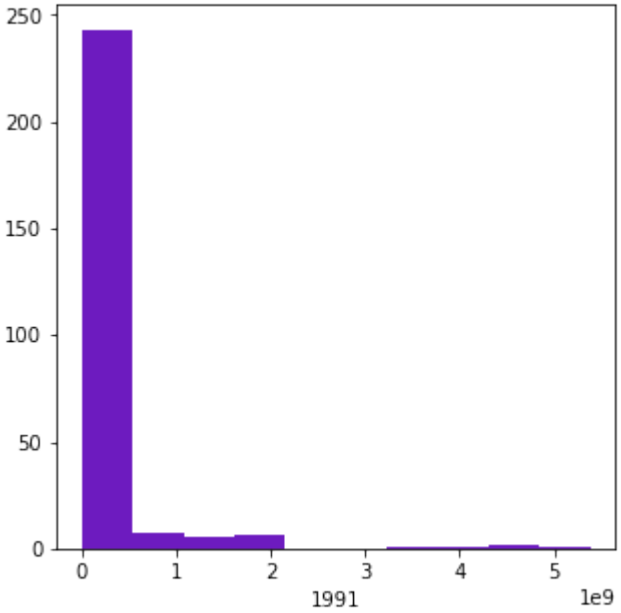

250
200
150
100
50
0
0
1
2
3
4
5
1986
1e9

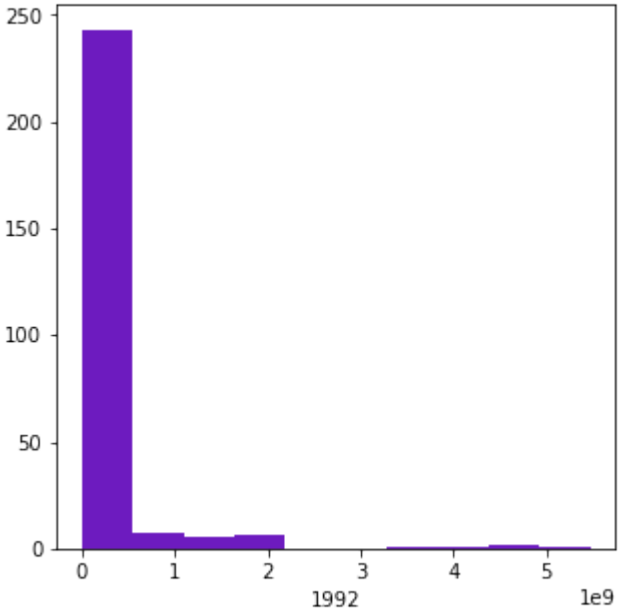

250
200
150
100
50
0
0
1
2
3
4
5
1987
1e9

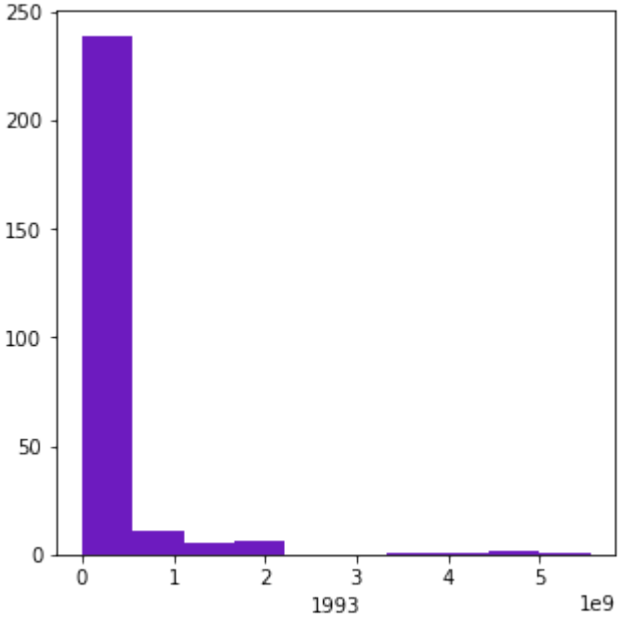

250
200
150
100
50
0
0
1
2
3
4
5
1988
1e9

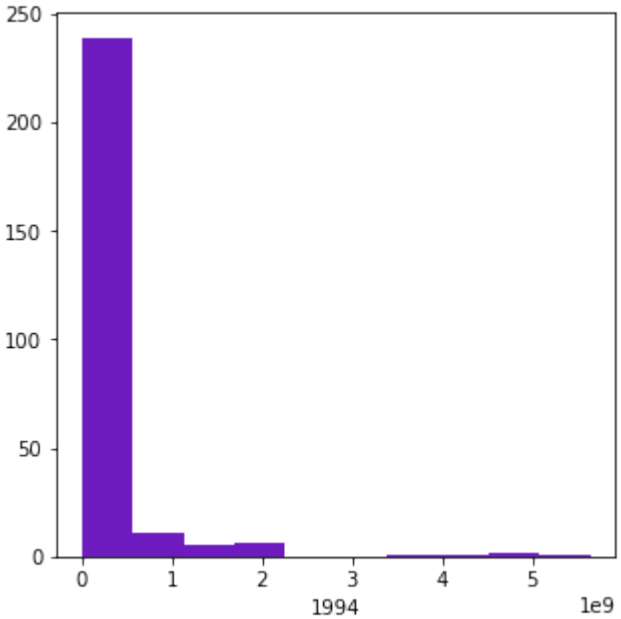

250
200
150
100
50
0
0
1
2
3
4
5
1989
1e9

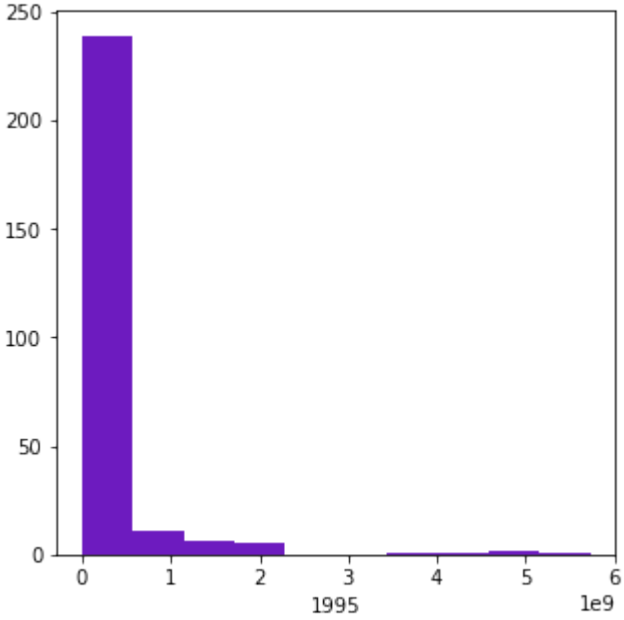

250
200
150
100
50
0
0
1
2
3
4
5
1990
1e9

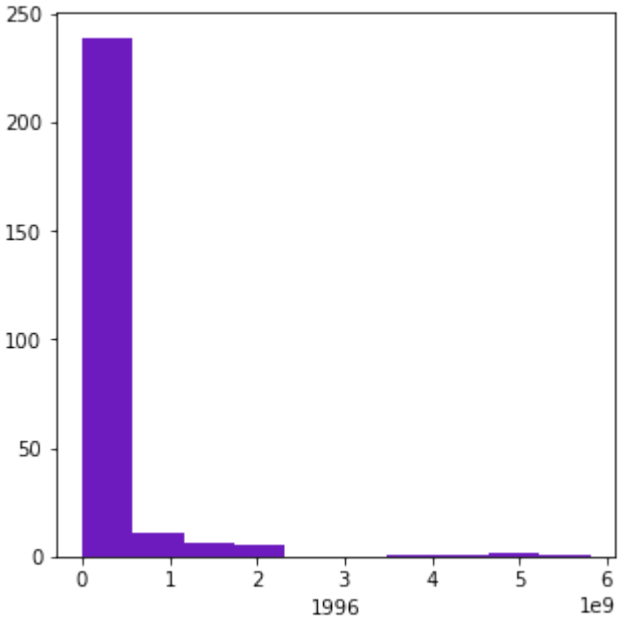

250
200
150
100
50
0
0
1
2
3
4
5
1991
1e9

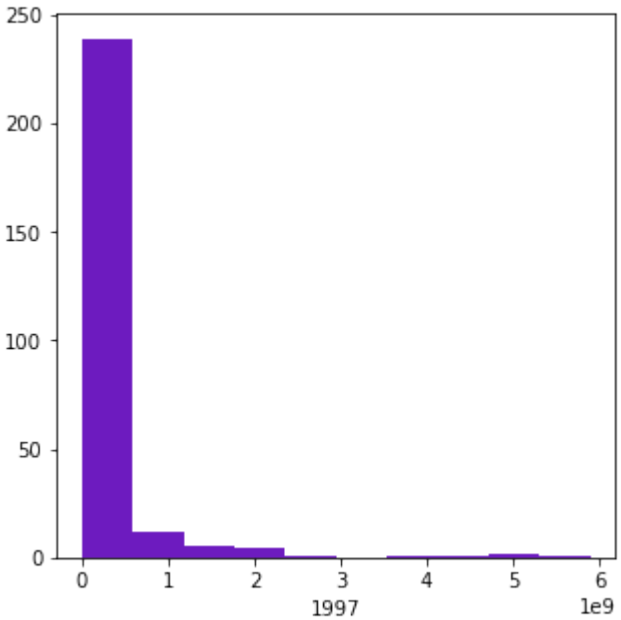

250
200
150
100
50
0
0
1
2
3
4
5
1992
1e9

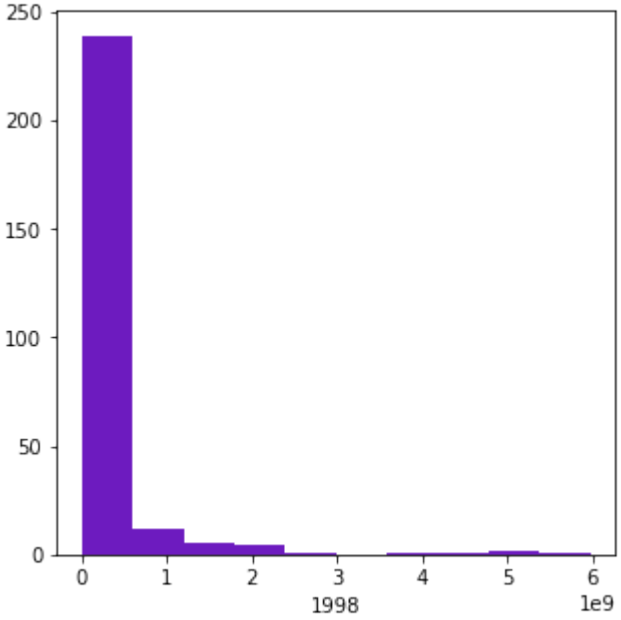

250
200
150
100
50
0
0
1
2
3
4
5
1993
1e9

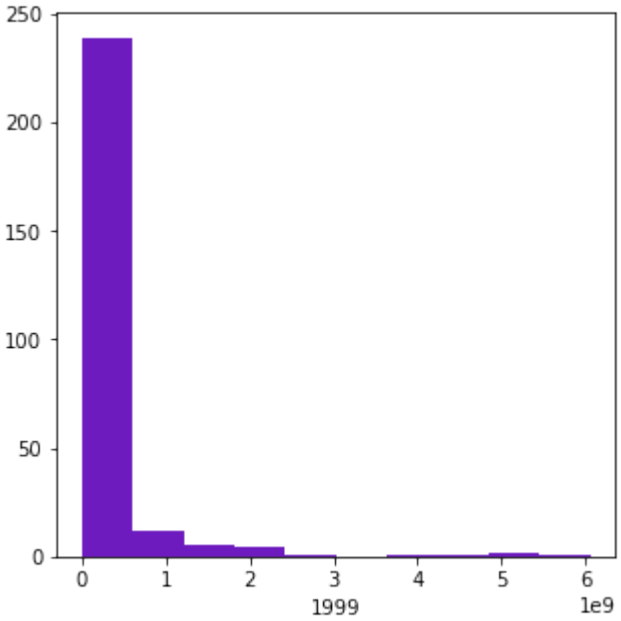

250
200
150
100
50
0
0
1
2
3
4
5
1994
1e9

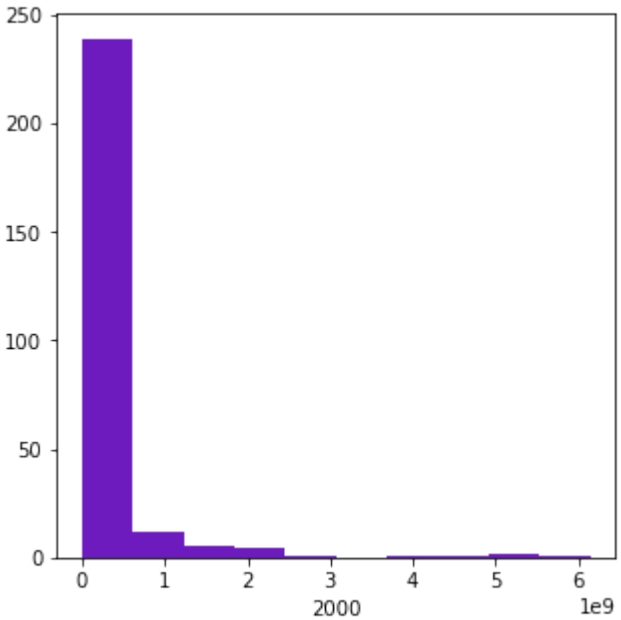

250
200
150
100
50
0
0
1
2
3
4
5
6
1995
1e9

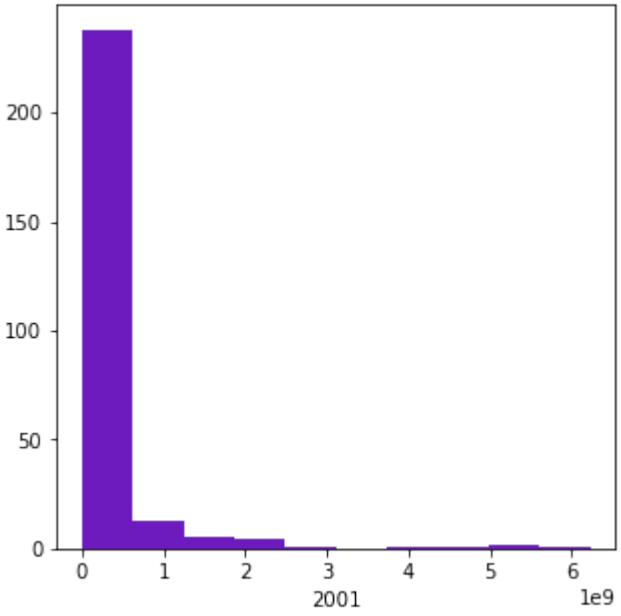

250
200
150
100
50
0
0
1
2
3
4
5
6
1996
1e9

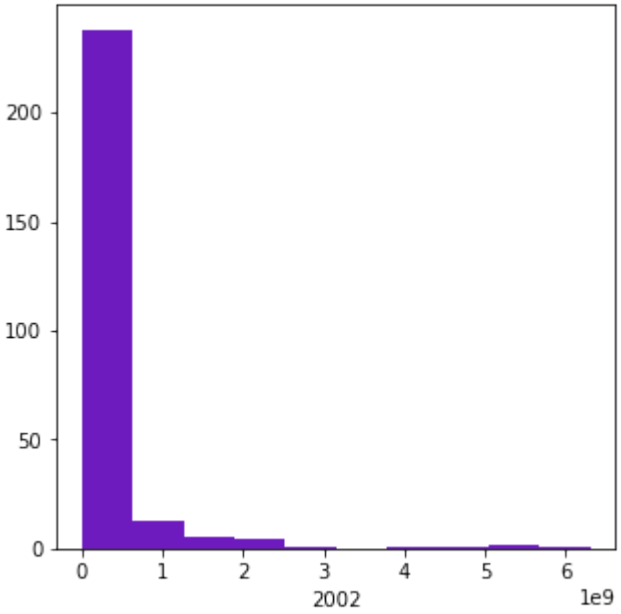

250
200
150
100
50
0
0
1
2
3
4
5
6
1997
1e9

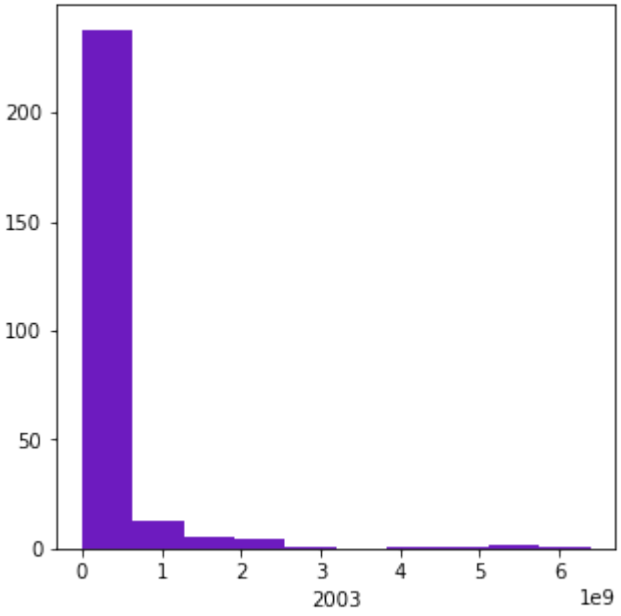

250
200
150
100
50
0
0
1
2
3
4
5
6
1998
1e9

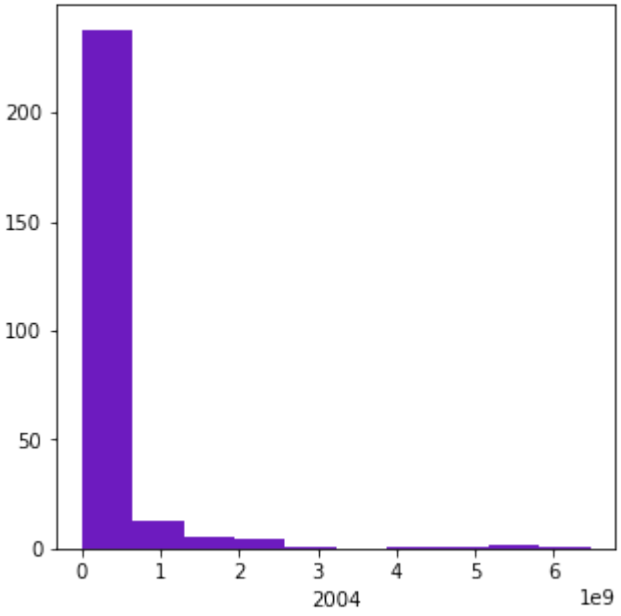

250
200
150
100
50
0
0
1
2
3
4
5
6
1999
1e9

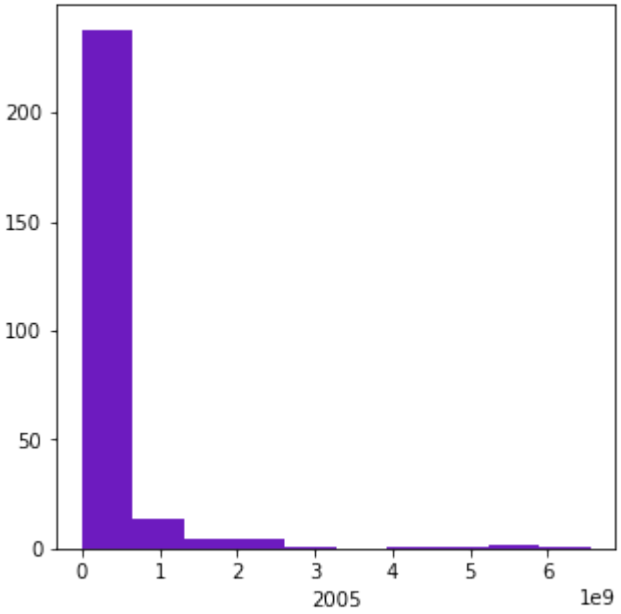

250
200
150
100
50
0
0
1
2
3
4
5
6
2000
1e9

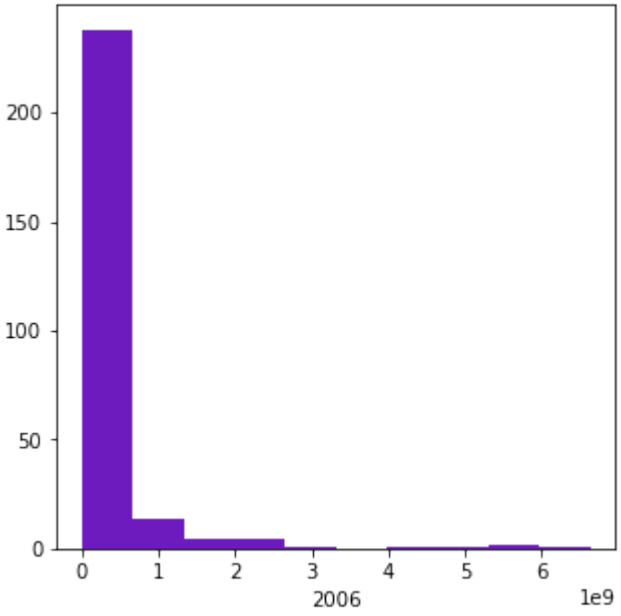

200
150
100
50
0
0
1
2
3
4
5
6
2001
1e9

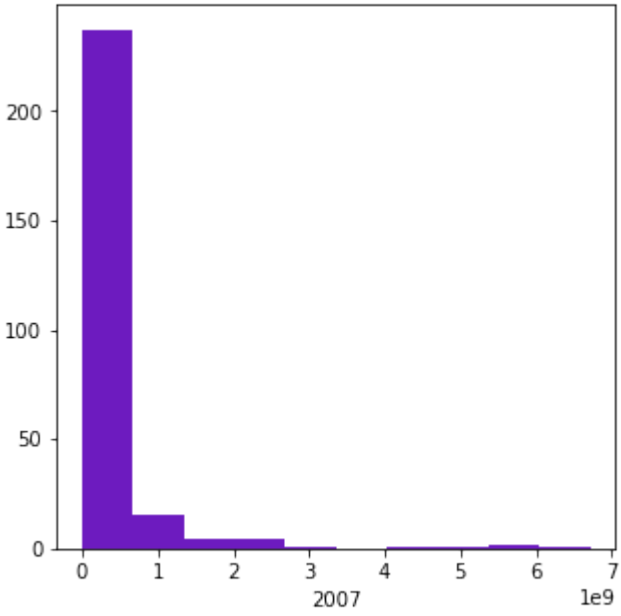

200
150
100
50
0
0
1
2
3
4
5
6
2002
1e9

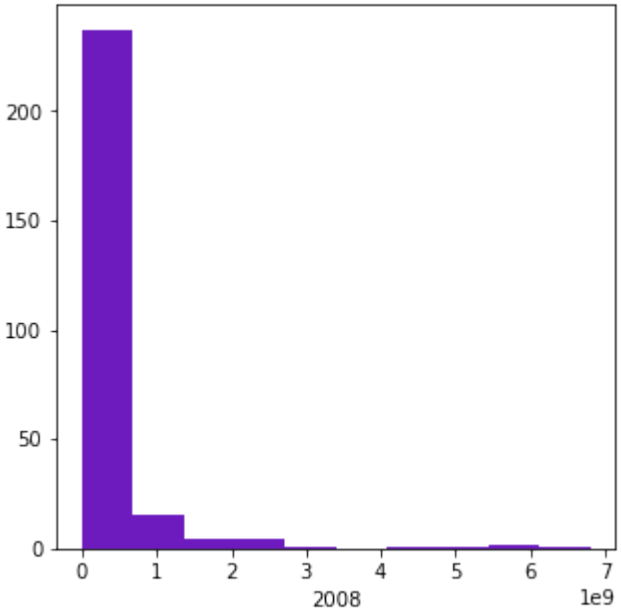

200
150
100
50
0
0
1
2
3
4
5
6
2003
1e9

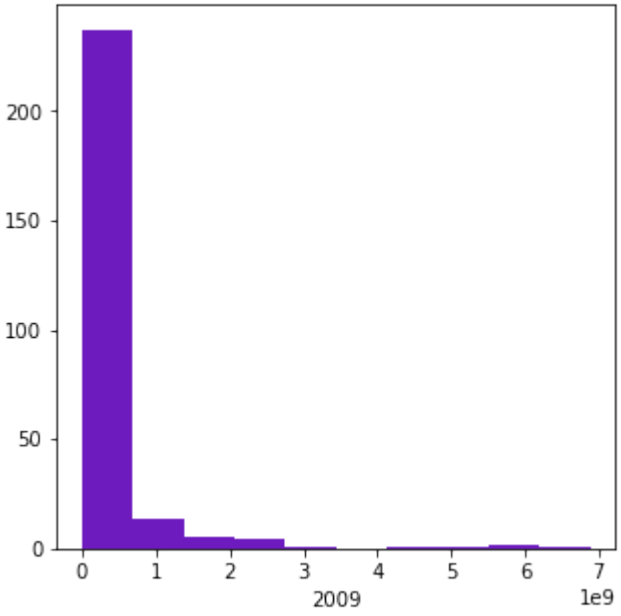

200
150
100
50
0
0
1
2
3
4
5
6
2004
1e9

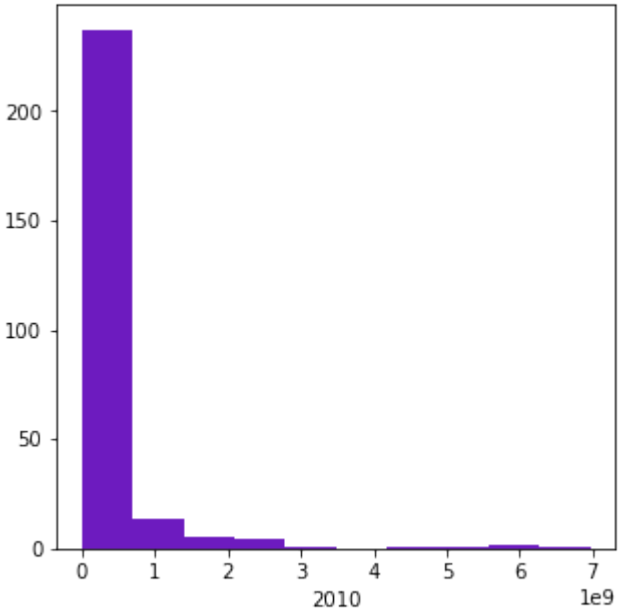

200
150
100
50
0
0
1
2
3
4
5
6
2005
1e9

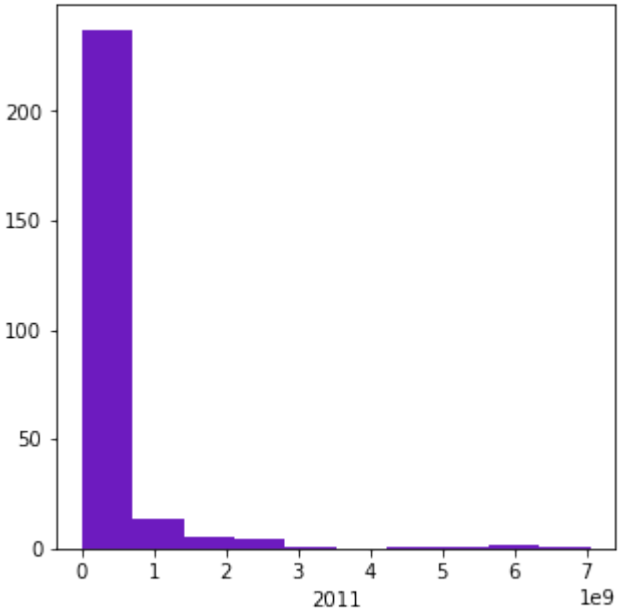

200
150
100
50
0
0
1
2
3
4
5
6
2006
1e9

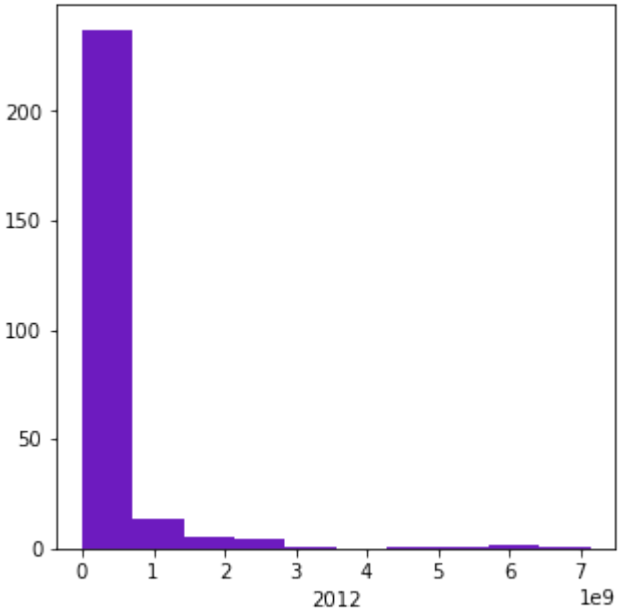

200
150
100
50
0
0
1
2
3
4
5
6
7
2007
1e9

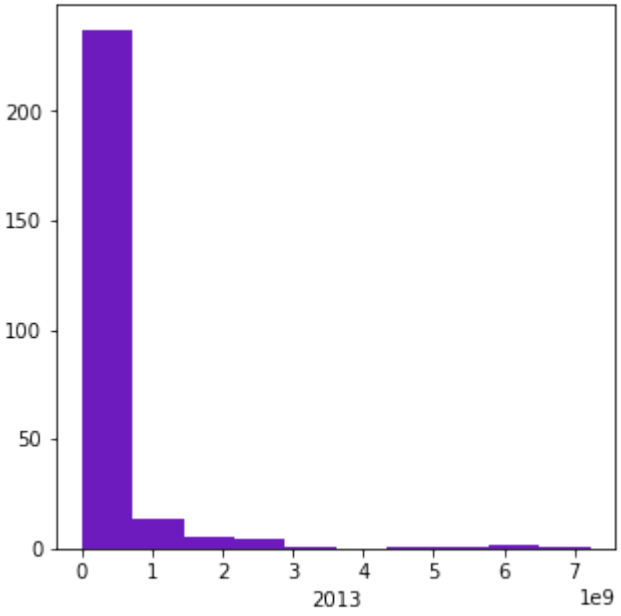

200
150
100
50
0
0
1
2
3
4
5
6
7
2008
1e9

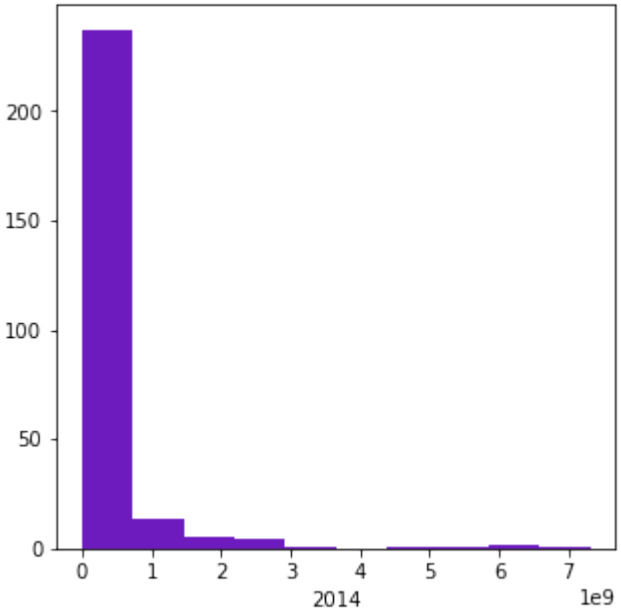

200
150
100
50
0
0
1
2
3
4
5
6
7
2009
1e9

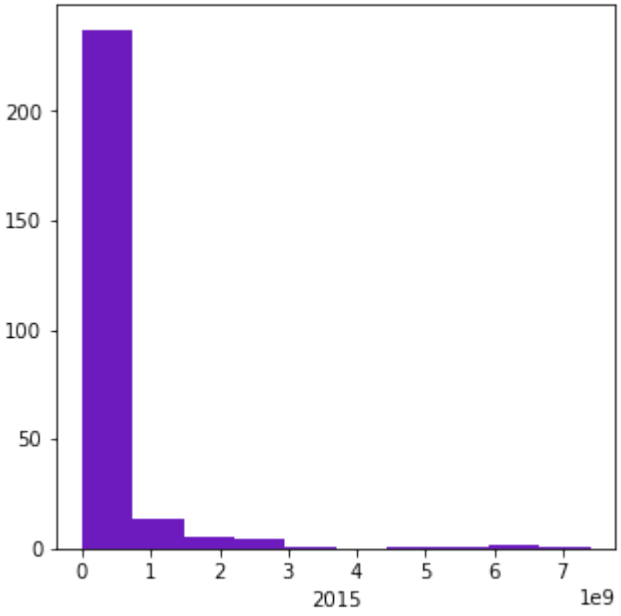

200
150
100
50
0
0
1
2
3
4
5
6
7
2010
1e9

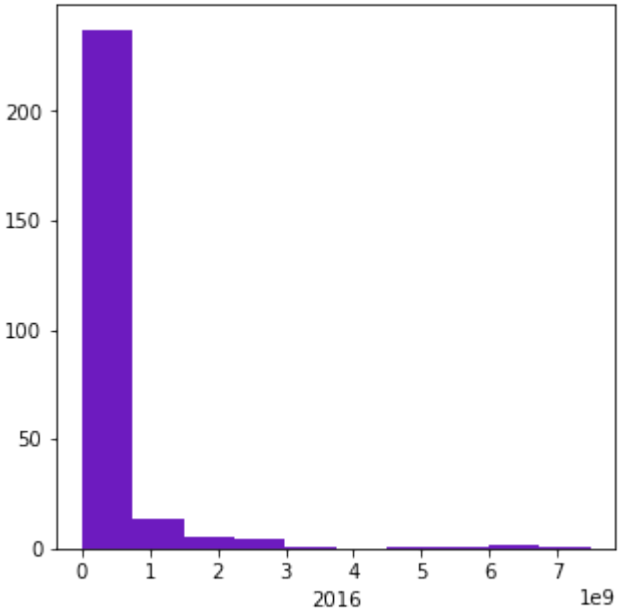

200
150
100
50
0
0
1
2
3
4
5
6
7
2011
1e9

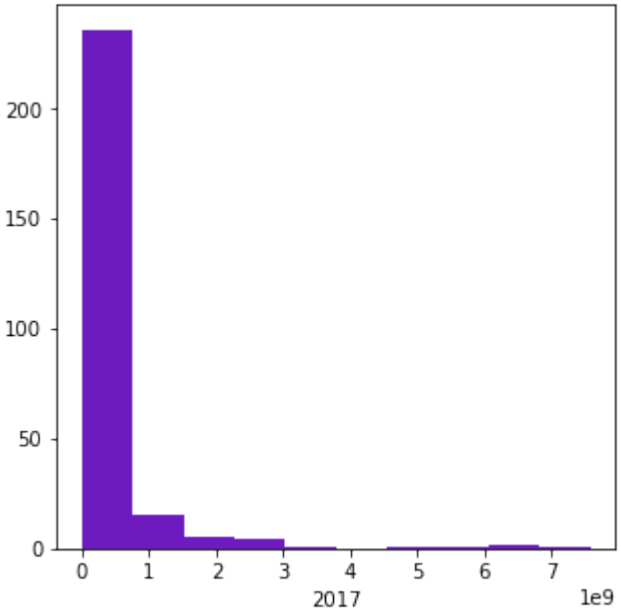

200
150
100
50
0
0
1
2
3
4
5
6
7
2012
1e9

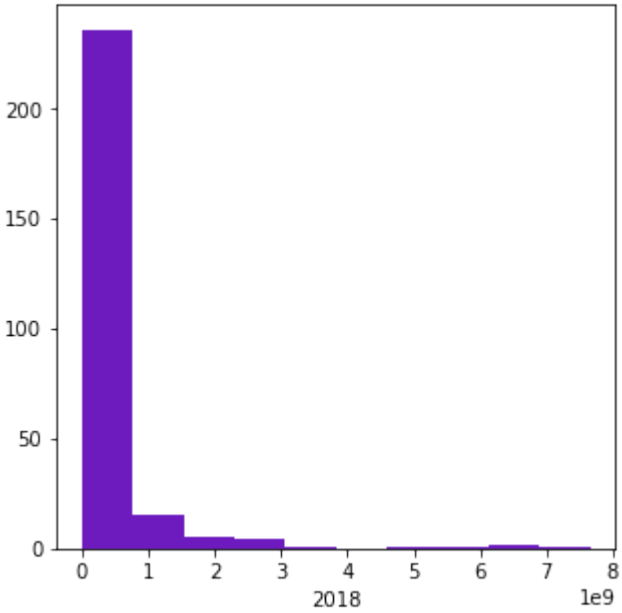

200
150
100
50
0
0 1 2 3 4 5 6 7
2013
1e9

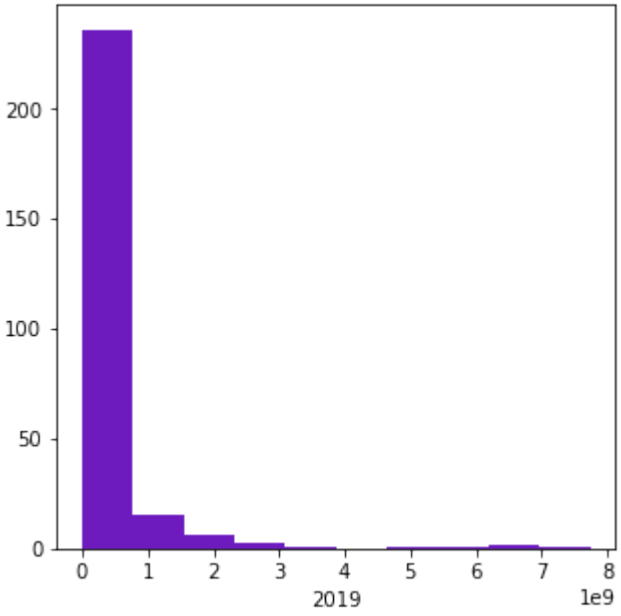

200
150
100
50
0
0 1 2 3 4 5 6 7
2014
1e9

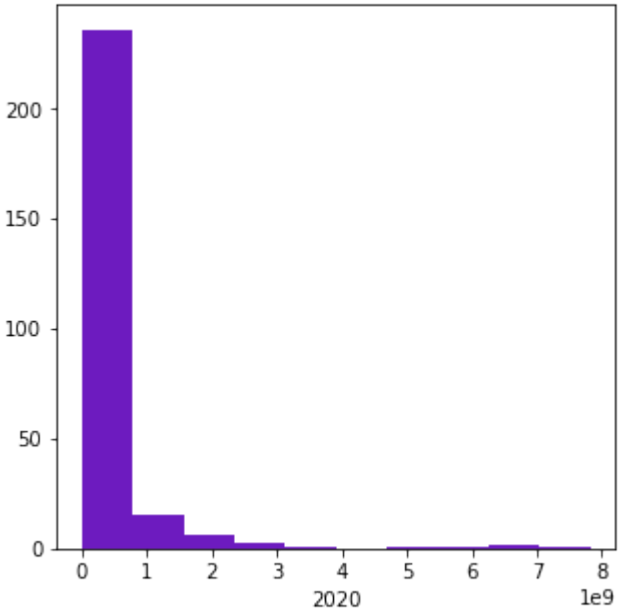

200
150
100
50
0
0 1 2 3 4 5 6 7
2015
1e9

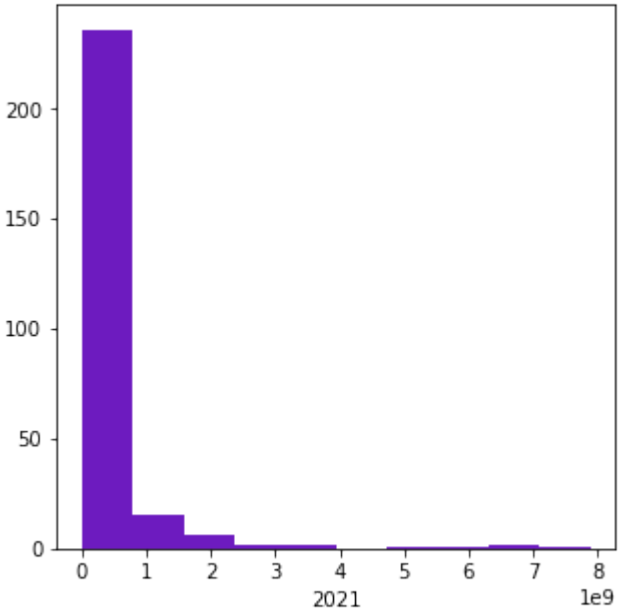

200
150
100
50
0
0 1 2 3 4 5 6 7
2016
1e9

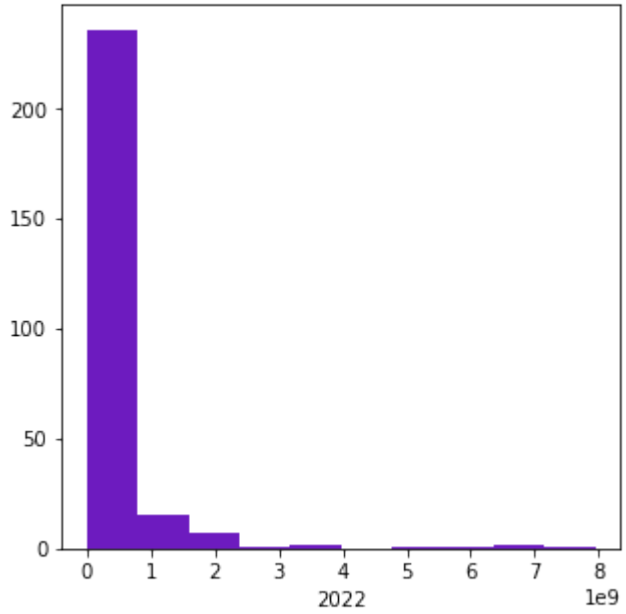

200
150
100
50
0
0 1 2 3 4 5 6 7
2017
1e9


200
150
100
50
0
0 1 2 3 4 5 6 7 8
2018
1e9


200
150
100
50
0
0 1 2 3 4 5 6 7 8
2019
1e9


200
150
100
50
0
0 1 2 3 4 5 6 7 8
2020
1e9

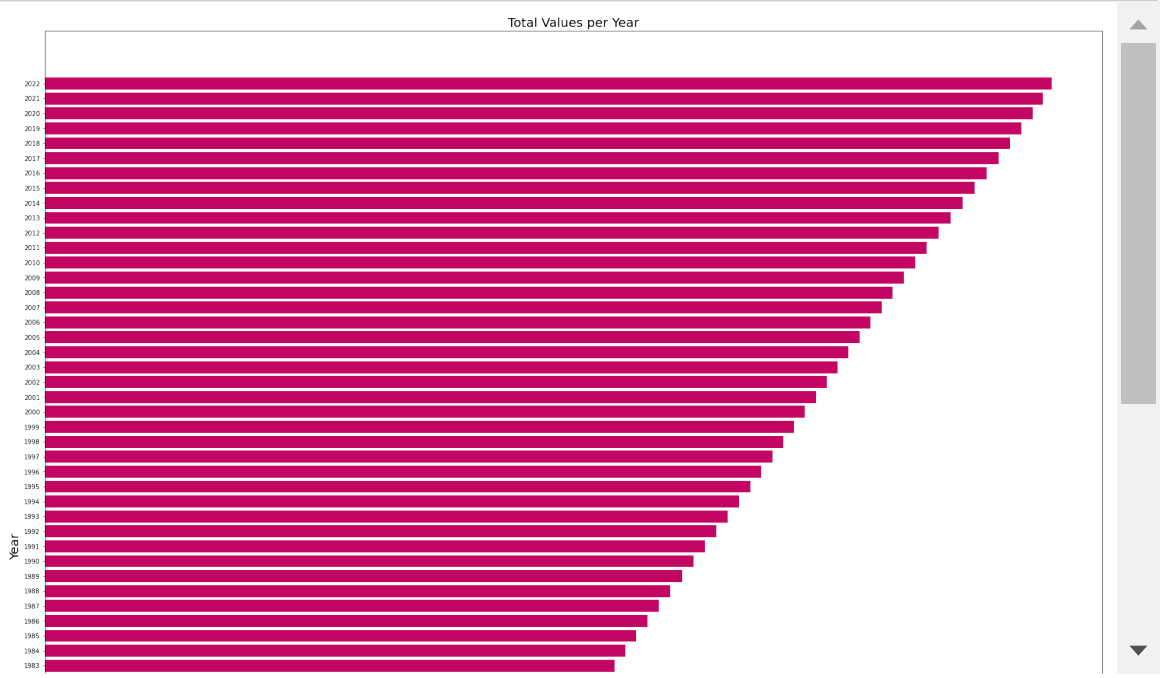

200
150
100
50
0
0 1 2 3 4 5 6 7 8
2021
1e9



In [24]:
```
years = df.columns[1:]

total_values = df[years].sum()

plt.figure(figsize=(30, 30))
plt.barh(years, total_values,color='#c70864')
plt.xlabel('Total Values')
plt.ylabel('Year', size=20)
plt.title('Total Values per Year', size=20)
plt.show()
```

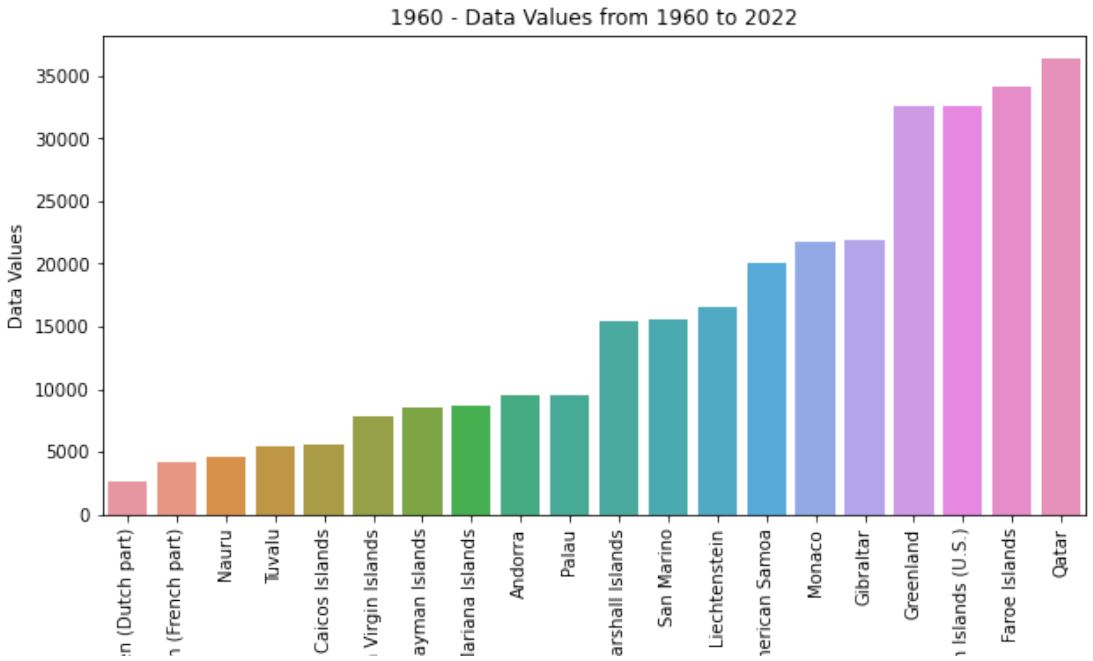


In [25]:
```
country_by_1960 = df.sort_values(by='1960').head(20)
country_by_1960
```

Out[25]:

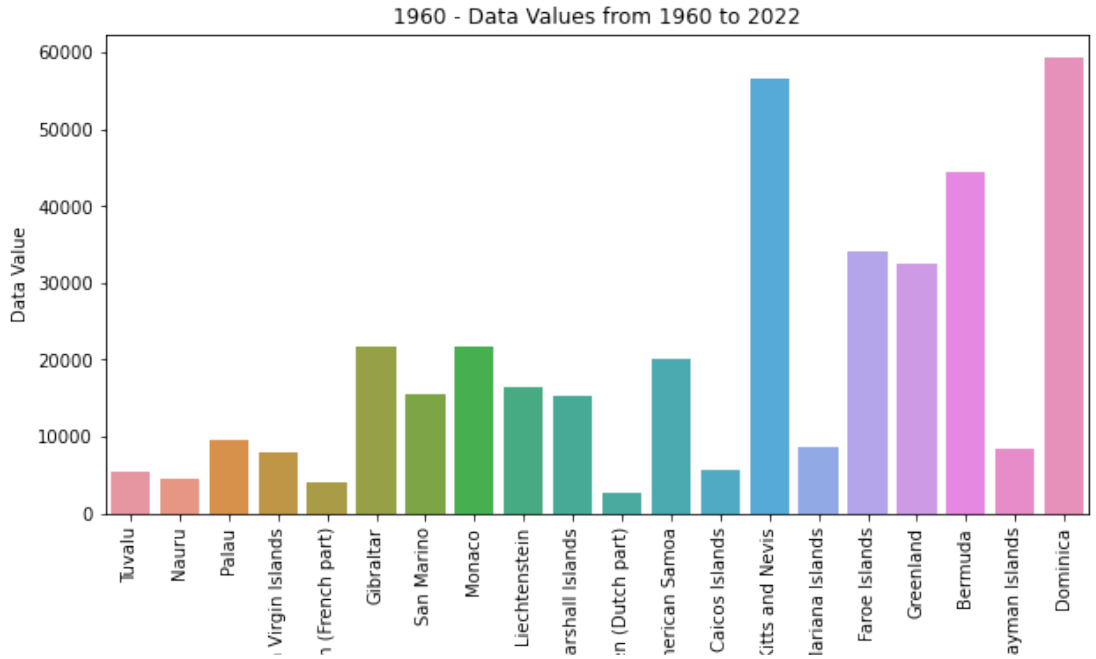
	Country Name	1960	1961	1962	1963	1964	1965	1966	1967
225	Sint Maarten (Dutch part)	2646.0	2888.0	3171.0	3481.0	3811.0	4161.0	4531.0	4930.0
147	St. Martin (French part)	4135.0	4258.0	4388.0	4524.0	4666.0	4832.0	5044.0	5294.0
179	Nauru	4582.0	4753.0	4950.0	5198.0	5484.0	5804.0	6021.0	6114.0
245	Tuvalu	5404.0	5436.0	5471.0	5503.0	5525.0	5548.0	5591.0	5657.0
228	Turks and Caicos Islands	5604.0	5625.0	5633.0	5634.0	5642.0	5650.0	5652.0	5662.0
255	British Virgin Islands	7850.0	7885.0	7902.0	7919.0	7949.0	8018.0	8139.0	8337.0
52	Cayman Islands	8473.0	8626.0	8799.0	8985.0	9172.0	9366.0	9566.0	9771.0

In [27]:
```
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()
```



In [28]:
```
country_by_2022 = df.sort_values(by='2022').head(20)
country_by_2022
```

Out[28]:

	Country Name	1960	1961	1962	1963	1964	1965	1966	1967
245	Tuvalu	5404.0	5436.0	5471.0	5503.0	5525.0	5548.0	5591.0	5657.0
179	Nauru	4582.0	4753.0	4950.0	5198.0	5484.0	5804.0	6021.0	6114.0
188	Palau	9446.0	9639.0	9851.0	10076.0	10318.0	10563.0	10813.0	10992.0
255	British Virgin Islands	7850.0	7885.0	7902.0	7919.0	7949.0	8018.0	8139.0	8337.0
147	St. Martin (French part)	4135.0	4258.0	4388.0	4524.0	4666.0	4832.0	5044.0	5294.0
84	Gibraltar	21822.0	21907.0	22249.0	22796.0	23347.0	23910.0	24477.0	25047.0
212	San Marino	15556.0	15895.0	16242.0	16583.0	16926.0	17273.0	17588.0	17907.0
149	Monaco	21797.0	21907.0	22106.0	22442.0	22766.0	23022.0	23198.0	23281.0
137	Liechtenstein	16472.0	16834.0	17221.0	17625.0	18058.0	18500.0	18957.0	19467.0

In [30]:
```
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 1960 to 2022")
        plt.xticks(rotation=90)
        plt.show()
```