In [69]:
```python
import pandas as pd
import numpy as np
```

Excel format

In [3]:
```python
df= pd.read_excel(r'C:\Users\ASUS\OneDrive\Desktop\prescription-drugs-introduced-to-market-2.xlsx')
```

In [4]:
```python
df
```

Out[4]:

	Manufacturer Name	NDC Number	Drug Product Description	Date Introduced to Market	WAC at Introduction	Marketing/Pricing Plan Description	Marketing/Pricing Plan Non-Public Indicator	Estimated Number of Patients	Breakthrough Therapy Indicator	Priority Review Indicator	Ac
0	Accord Healthcare, Inc.	16729043445	Daptomycin (SDV); 350 mg; 10 pk [lyo] Vial	2020-02-27	680.00	Marketing: Accord markets generic Daptomycin ...	NaN	NaN	NaN	NaN	
1	Aimmune Therapeutics	71881011313	PALFORZIA Initial Dose Escalation Card -- 0.5 ...	2020-03-10	30.00	Marketing activities that support the launch o...	NaN	1600000.0	1.0	NaN	
2	Aimmune Therapeutics	71881011130	PALFORZIA Maintenance Dosing Pack -- 300 mg, 3...	2020-03-10	890.00	Marketing activities that support the launch o...	NaN	1600000.0	1.0	NaN	
3	Aimmune Therapeutics	71881010345	PALFORZIA Up-Dosing Pack -- 12 mg	2020-03-10	445.00	Marketing activities that support the launch o...	NaN	1600000.0	1.0	NaN	
4	Aimmune Therapeutics	71881010730	PALFORZIA Up-Dosing Pack -- 120 mg	2020-03-10	445.00	Marketing activities that support the launch o...	NaN	1600000.0	1.0	NaN	
...	...	...	...	...	...	...	...	...	...	...	
149	Zydus Pharmaceuticals (USA) Inc.	70710151406	Fondaparinux Inj 2.5mg/0.5ml (10x0.5ml)	2020-01-13	705.00	NaN	1.0	NaN	NaN	NaN	
150	Zydus Pharmaceuticals (USA) Inc.	70710151409	Fondaparinux Inj 2.5mg/0.5ml (2x0.5ml)	2020-01-13	1211.70	NaN	1.0	NaN	NaN	NaN	
151	Zydus Pharmaceuticals (USA) Inc.	70710151509	Fondaparinux Inj 5mg/0.4ml (2x0.4ml)	2020-01-13	950.60	NaN	1.0	NaN	NaN	NaN	
152	Zydus Pharmaceuticals (USA) Inc.	70710151609	Fondaparinux Inj 7.5mg/0.6ml (2x0.6ml)	2020-01-13	950.60	NaN	1.0	NaN	NaN	NaN	
153	Zydus Pharmaceuticals (USA) Inc.	70710113301	Chlorpromazine HCl 200mg Tab	2020-02-11	989.35	NaN	1.0	NaN	NaN	NaN	

154 rows × 16 columns

In [5]:
```python
df.shape
```

Out[5]:
```
(154, 16)
```

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 154 entries, 0 to 153
Data columns (total 16 columns):
 #   Column                                   Non-Null Count  Dtype  
---  ------                                   --------------  -----  
 0   Manufacturer Name                        154 non-null    object 
 1   NDC Number                               154 non-null    int64  
 2   Drug Product Description                 153 non-null    object 
 3   Date Introduced to Market                154 non-null    object 
 4   WAC at Introduction                      154 non-null    float64
 5   Marketing/Pricing Plan Description       79 non-null     object 
 6   Marketing/Pricing Plan Non-Public Indicator  75 non-null     float64
 7   Estimated Number of Patients             115 non-null    float64
 8   Breakthrough Therapy Indicator           34 non-null     float64
 9   Priority Review Indicator                31 non-null     float64
 10  Acquisition Date                         24 non-null     object 
 11  Acquisition Price                        15 non-null     float64
 12  Acquisition Price Non-Public Indicator   11 non-null     float64
 13  Acquisition Price Comment                14 non-null     object 
 14  General Comments                         144 non-null    object 
 15  Supporting Documents                     7 non-null      object 
dtypes: float64(7), int64(1), object(8)
memory usage: 19.4+ KB
```

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

Out[7]:
```
Index(['Manufacturer Name', 'NDC Number', 'Drug Product Description',
       'Date Introduced to Market', 'WAC at Introduction',
       'Marketing/Pricing Plan Description',
       'Marketing/Pricing Plan Non-Public Indicator',
       'Estimated Number of Patients', 'Breakthrough Therapy Indicator',
       'Priority Review Indicator', 'Acquisition Date', 'Acquisition Price',
       'Acquisition Price Non-Public Indicator', 'Acquisition Price Comment',
       'General Comments', 'Supporting Documents'],
      dtype='object')
```

In [8]:
```
df.describe()
```

Out[8]:

	NDC Number	WAC at Introduction	Marketing/Pricing Plan Non-Public Indicator	Estimated Number of Patients	Breakthrough Therapy Indicator	Priority Review Indicator	Acquisition Price	Acquisition Price Non-Public Indicator
count	1.540000e+02	154.000000	75.0	1.150000e+02	34.0	31.0	1.500000e+01	11.0
mean	4.662565e+10	5929.715519	1.0	1.787460e+06	1.0	1.0	1.550633e+10	1.0
std	3.002006e+10	9192.344602	0.0	4.837304e+06	0.0	0.0	3.032129e+10	0.0
min	2.298026e+06	2.230000	1.0	0.000000e+00	1.0	1.0	0.000000e+00	1.0
25%	3.185300e+09	787.500000	1.0	3.750000e+03	1.0	1.0	1.250000e+08	1.0
50%	6.330407e+10	1322.330000	1.0	3.300000e+04	1.0	1.0	6.350000e+08	1.0
75%	7.169901e+10	6714.620000	1.0	1.600000e+06	1.0	1.0	3.777500e+09	1.0
max	7.628207e+10	48633.600000	1.0	3.100000e+07	1.0	1.0	7.400000e+10	1.0

In [10]:
```
df.isnull().any()  # check null values
```

Out[10]:
```
Manufacturer Name                              False
NDC Number                                     False
Drug Product Description                        True
Date Introduced to Market                      False
WAC at Introduction                            False
Marketing/Pricing Plan Description              True
Marketing/Pricing Plan Non-Public Indicator     True
Estimated Number of Patients                    True
Breakthrough Therapy Indicator                  True
Priority Review Indicator                       True
Acquisition Date                                True
Acquisition Price                               True
Acquisition Price Non-Public Indicator          True
Acquisition Price Comment                       True
General Comments                                True
Supporting Documents                            True
dtype: bool
```

In [11]:
```
df['Manufacturer Name'].unique()
```

Out[11]:
```
array(['Accord Healthcare, Inc.', 'Aimmune Therapeutics',
       'Ajanta Pharma USA, Inc.', 'Allergan', 'American Regent ',
       'Amneal Pharmaceuticals', 'AstraZeneca ', 'AvKare',
       'Bionpharma, Inc', 'Blueprint Medicines Corporation',
       'Bristol Myers Squibb', 'Celltrion USA, Inc.', 'Cipla USA, Inc.',
       'Deciphera Pharmaceuticals, LLC', "Dr. Reddy's Laboratories, Inc.",
       'Eli Lilly and Company', 'Epizyme, Inc.',
       'Exelan Pharmaceuticals, Inc.', 'Fresenius Kabi USA LLC',
       'Gilead Sciences, Inc.', 'Glenmark Pharmaceuticals Inc., USA',
       'Granules Pharmaceuticals Inc.',
       'Heritage Pharmacueticals Inc. D/B/A Avet Pharmacueticals Inc',
       'Hikma Pharmaceuticals USA Inc', 'Horizon Therapeutics USA, Inc.',
       'Immunomedics, Inc.', 'Incyte Corporation',
       'Intra-Cellular Therapies, Inc.', 'Janssen Biotech, Inc.',
       'Karyopharm Therapeutics Inc.', 'Mayne Pharma Inc',
       'Merck & Co., Inc.', 'Mylan Institutional Inc',
       'Novadoz Pharmaceuticals, LLC', 'Novartis',
       'Noven Therapeutics, LLC', 'Novo', 'Par Pharmaceutical  ',
       'Pfizer', 'Sanofi', 'Seattle Genetics, Inc.', 'Shionogi Inc.',
       'SK Life Science, Inc.', 'SpecGx', 'SUN PHARMACEUTICALS',
       'Sunovion Pharmaceuticals Inc.', 'TARO PHARMACEUTICALS',
       'Teva Parenteral Medicines, Inc.  ', 'Teva Pharmaceuticals USA',
       'Tolmar Pharmaceuticals, Inc.', 'UroGen Pharma, Inc.',
       'Zydus Pharmaceuticals (USA) Inc.'], dtype=object)
```

json format

In [16]:
```
df_json = pd.read_json(r'C:\Users\ASUS\OneDrive\Desktop\iris.json')
```

In [17]:
```
df_json
```

Out[17]:

	sepalLength	sepalWidth	petalLength	petalWidth	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

In [18]:
```
df_json.shape
```

Out[18]:
```
(150, 5)
```

In [19]:
```
df_json.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   sepalLength  150 non-null    float64
 1   sepalWidth   150 non-null    float64
 2   petalLength  150 non-null    float64
 3   petalWidth   150 non-null    float64
 4   species      150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

In [20]:
```
len(df_json)
```

Out[20]:
```
150
```

In [21]:
```
df_json.head()
```

Out[21]:

	sepalLength	sepalWidth	petalLength	petalWidth	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [22]:
```
df_json.tail()
```

Out[22]:

	sepalLength	sepalWidth	petalLength	petalWidth	species
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

In [25]:
```
df_json.isnull().any()
```

Out[25]:
```
sepalLength    False
sepalWidth     False
petalLength    False
petalWidth     False
species        False
dtype: bool
```

In [26]:
```
df_json.describe()
```

Out[26]:

	sepalLength	sepalWidth	petalLength	petalWidth
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [27]:
```
df_json['species'].unique()
```

Out[27]:
```
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

In [28]:
```
df_json[0:10]
```

Out[28]:

	sepalLength	sepalWidth	petalLength	petalWidth	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
5	5.4	3.9	1.7	0.4	setosa
6	4.6	3.4	1.4	0.3	setosa
7	5.0	3.4	1.5	0.2	setosa
8	4.4	2.9	1.4	0.2	setosa
9	4.9	3.1	1.5	0.1	setosa

```
In [29]: df_json[::5]
```

Out[29]:

	sepalLength	sepalWidth	petalLength	petalWidth	species
0	5.1	3.5	1.4	0.2	setosa
5	5.4	3.9	1.7	0.4	setosa
10	5.4	3.7	1.5	0.2	setosa
15	5.7	4.4	1.5	0.4	setosa
20	5.4	3.4	1.7	0.2	setosa
25	5.0	3.0	1.6	0.2	setosa
30	4.8	3.1	1.6	0.2	setosa
35	5.0	3.2	1.2	0.2	setosa
40	5.0	3.5	1.3	0.3	setosa
45	4.8	3.0	1.4	0.3	setosa
50	7.0	3.2	4.7	1.4	versicolor
55	5.7	2.8	4.5	1.3	versicolor
60	5.0	2.0	3.5	1.0	versicolor
65	6.7	3.1	4.4	1.4	versicolor
70	5.9	3.2	4.8	1.8	versicolor
75	6.6	3.0	4.4	1.4	versicolor
80	5.5	2.4	3.8	1.1	versicolor
85	6.0	3.4	4.5	1.6	versicolor
90	5.5	2.6	4.4	1.2	versicolor
95	5.7	3.0	4.2	1.2	versicolor
100	6.3	3.3	6.0	2.5	virginica
105	7.6	3.0	6.6	2.1	virginica
110	6.5	3.2	5.1	2.0	virginica
115	6.4	3.2	5.3	2.3	virginica
120	6.9	3.2	5.7	2.3	virginica
125	7.2	3.2	6.0	1.8	virginica
130	7.4	2.8	6.1	1.9	virginica
135	7.7	3.0	6.1	2.3	virginica
140	6.7	3.1	5.6	2.4	virginica
145	6.7	3.0	5.2	2.3	virginica

xml format

```
In [32]: df_xml= pd.read_xml(r'C:\Users\ASUS\OneDrive\Desktop\AEO 2011 Final.xml')
```

In [33]:
```
df_xml
```

Out[33]:

	href	study	region	cases	table	yearFilter	label	number	name	data
0	http://eia.gov/oiaf/aeo/tablebrowser/aeo_query...	AEO2011	0-0	ref2011-d020911a	1-AEO2011	0	Table 1. Total Energy Supply, Disposition, and...	1	Total Energy Supply, Disposition, and Price Su...	NaN
1	http://eia.gov/oiaf/aeo/tablebrowser/aeo_query...	AEO2011	1-0	ref2011-d020911a	2-AEO2011	0	Table 2. Energy Consumption by Sector and Sour...	2	Energy Consumption by Sector and Source - Unit...	NaN
2	http://eia.gov/oiaf/aeo/tablebrowser/aeo_query...	AEO2011	1-0	ref2011-d020911a	3-AEO2011	0	Table 3. Energy Prices by Sector and Source - ...	3	Energy Prices by Sector and Source - United St...	NaN
3	http://eia.gov/oiaf/aeo/tablebrowser/aeo_query...	AEO2011	0-0	ref2011-d020911a	4-AEO2011	0	Table 4. Residential Sector Key Indicators and...	4	Residential Sector Key Indicators and Consumption	NaN
4	http://eia.gov/oiaf/aeo/tablebrowser/aeo_query...	AEO2011	0-0	ref2011-d020911a	5-AEO2011	0	Table 5. Commercial Sector Key Indicators and ...	5	Commercial Sector Key Indicators and Consumption	NaN
...	...	...	...	...	...	...	...	...	...	...
835	http://eia.gov/oiaf/aeo/tablebrowser/aeo_query...	AEO2011	0-0	lp2011lno-d022511a	97-AEO2011	0	Table 97. World Metallurgical Coal Flows By Im...	97	World Metallurgical Coal Flows By Importing Re...	NaN
836	http://eia.gov/oiaf/aeo/tablebrowser/aeo_query...	AEO2011	0-0	lp2011lno-d022511a	98-AEO2011	0	Table 98. World Total Coal Flows By Importing ...	98	World Total Coal Flows By Importing Regions an...	NaN
837	http://eia.gov/oiaf/aeo/tablebrowser/aeo_query...	AEO2011	0-0	lp2011lno-d022511a	100-AEO2011	0	Table 100. Employment and Shipments by Industr...	100	Employment and Shipments by Industry, and Inco...	NaN
838	http://eia.gov/oiaf/aeo/tablebrowser/aeo_query...	AEO2011	0-0	lp2011lno-d022511a	101-AEO2011	0	Table 101. Imported Liquids by Source	101	Imported Liquids by Source	NaN
839	http://eia.gov/oiaf/aeo/tablebrowser/aeo_query...	AEO2011	0-0	lp2011lno-d022511a	20-AEO2011	0	Table 20. Conversion Factors	20	Conversion Factors	NaN

840 rows × 10 columns

In [34]:
```
df_xml.shape
```

Out[34]:
```
(840, 10)
```

In [35]:
```
df_xml.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 840 entries, 0 to 839
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   href        840 non-null    object 
 1   study       840 non-null    object 
 2   region      840 non-null    object 
 3   cases       840 non-null    object 
 4   table       840 non-null    object 
 5   yearFilter  840 non-null    int64  
 6   label       840 non-null    object 
 7   number      840 non-null    int64  
 8   name        840 non-null    object 
 9   data        0 non-null      float64
dtypes: float64(1), int64(2), object(7)
memory usage: 65.8+ KB
```

In [36]:
```
df_xml.columns
```

Out[36]:
```
Index(['href', 'study', 'region', 'cases', 'table', 'yearFilter', 'label',
       'number', 'name', 'data'],
      dtype='object')
```

```
In [37]: df_xml.isnull().any()
```

```
Out[37]: href          False
         study         False
         region        False
         cases         False
         table         False
         yearFilter    False
         label         False
         number        False
         name          False
         data           True
         dtype: bool
```

```
In [39]: df_xml.describe()
```

Out[39]:

	yearFilter	number	data
count	840.0	840.000000	0.0
mean	0.0	55.321429	NaN
std	0.0	38.501292	NaN
min	0.0	1.000000	NaN
25%	0.0	18.000000	NaN
50%	0.0	59.000000	NaN
75%	0.0	72.000000	NaN
max	0.0	148.000000	NaN

```
In [40]: df_xml[:10]
```

Out[40]:

	href	study	region	cases	table	yearFilter	label	number	name	data
0	http://eia.gov/oiaf/aeo/tablebrowser/aeo_query...	AEO2011	0-0	ref2011-d020911a	1-AEO2011	0	Table 1. Total Energy Supply, Disposition, and...	1	Total Energy Supply, Disposition, and Price Su...	NaN
1	http://eia.gov/oiaf/aeo/tablebrowser/aeo_query...	AEO2011	1-0	ref2011-d020911a	2-AEO2011	0	Table 2. Energy Consumption by Sector and Sour...	2	Energy Consumption by Sector and Source - Unit...	NaN
2	http://eia.gov/oiaf/aeo/tablebrowser/aeo_query...	AEO2011	1-0	ref2011-d020911a	3-AEO2011	0	Table 3. Energy Prices by Sector and Source - ...	3	Energy Prices by Sector and Source - United St...	NaN
3	http://eia.gov/oiaf/aeo/tablebrowser/aeo_query...	AEO2011	0-0	ref2011-d020911a	4-AEO2011	0	Table 4. Residential Sector Key Indicators and...	4	Residential Sector Key Indicators and Consumption	NaN
4	http://eia.gov/oiaf/aeo/tablebrowser/aeo_query...	AEO2011	0-0	ref2011-d020911a	5-AEO2011	0	Table 5. Commercial Sector Key Indicators and ...	5	Commercial Sector Key Indicators and Consumption	NaN
5	http://eia.gov/oiaf/aeo/tablebrowser/aeo_query...	AEO2011	0-0	ref2011-d020911a	6-AEO2011	0	Table 6. Industrial Sector Key Indicators and ...	6	Industrial Sector Key Indicators and Consumption	NaN
6	http://eia.gov/oiaf/aeo/tablebrowser/aeo_query...	AEO2011	0-0	ref2011-d020911a	7-AEO2011	0	Table 7. Transportation Sector Key Indicators ...	7	Transportation Sector Key Indicators and Deliv...	NaN
7	http://eia.gov/oiaf/aeo/tablebrowser/aeo_query...	AEO2011	0-0	ref2011-d020911a	8-AEO2011	0	Table 8. Electricity Supply, Disposition, Pric...	8	Electricity Supply, Disposition, Prices, and E...	NaN
8	http://eia.gov/oiaf/aeo/tablebrowser/aeo_query...	AEO2011	0-0	ref2011-d020911a	9-AEO2011	0	Table 9. Electricity Generating Capacity	9	Electricity Generating Capacity	NaN
9	http://eia.gov/oiaf/aeo/tablebrowser/aeo_query...	AEO2011	0-0	ref2011-d020911a	10-AEO2011	0	Table 10. Electricity Trade	10	Electricity Trade	NaN

html format

```
In [64]: data1=pd.read_html(r'C:\Users\ASUS\OneDrive\Desktop\World Population by Year.html')
```

In [67]:
```
data1
```

Out[67]:
```
[    Year  World Population Yearly Change  Net Change  Density (P/Km²)
 0   2023        8045311447        0.88 %  70206291.0             54.0
 1   2022        7975105156        0.83 %  65810005.0             54.0
 2   2021        7909295151        0.87 %  68342271.0             53.0
 3   2020        7840952880        0.98 %  76001848.0             53.0
 4   2019        7764951032        1.06 %  81161204.0             52.0
 ..   ...               ...           ...         ...              ...
 92 -1000          50000000           NaN         NaN              NaN
 93 -2000          27000000           NaN         NaN              NaN
 94 -3000          14000000           NaN         NaN              NaN
 95 -4000           7000000           NaN         NaN              NaN
 96 -5000           5000000           NaN         NaN              NaN

 [97 rows x 5 columns]]
```

In [70]:
```
flat_data=np.reshape(data1,(97,5)) # convert to 2 dimmensional data
```

In [71]:
```
df_html=pd.DataFrame(flat_data) #convert to dataframe
```

In [72]:
```
df_html
```

Out[72]:

	0	1	2	3	4
0	2023	8045311447	0.88 %	70206291.0	54.0
1	2022	7975105156	0.83 %	65810005.0	54.0
2	2021	7909295151	0.87 %	68342271.0	53.0
3	2020	7840952880	0.98 %	76001848.0	53.0
4	2019	7764951032	1.06 %	81161204.0	52.0
...	...	...	...	...	...
92	-1000	50000000	NaN	NaN	NaN
93	-2000	27000000	NaN	NaN	NaN
94	-3000	14000000	NaN	NaN	NaN
95	-4000	7000000	NaN	NaN	NaN
96	-5000	5000000	NaN	NaN	NaN

97 rows × 5 columns

In [73]:
```
df_html.columns=[ 'Year','World Population', 'Yearly Change' , 'Net Change' , 'Density (P/Km²)']  # assigning column's nam
```

In [74]:
```
df_html
```

Out[74]:

	Year	World Population	Yearly Change	Net Change	Density (P/Km²)
0	2023	8045311447	0.88 %	70206291.0	54.0
1	2022	7975105156	0.83 %	65810005.0	54.0
2	2021	7909295151	0.87 %	68342271.0	53.0
3	2020	7840952880	0.98 %	76001848.0	53.0
4	2019	7764951032	1.06 %	81161204.0	52.0
...	...	...	...	...	...
92	-1000	50000000	NaN	NaN	NaN
93	-2000	27000000	NaN	NaN	NaN
94	-3000	14000000	NaN	NaN	NaN
95	-4000	7000000	NaN	NaN	NaN
96	-5000	5000000	NaN	NaN	NaN

97 rows × 5 columns

In [75]:
```
df_html.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 97 entries, 0 to 96
Data columns (total 5 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Year              97 non-null     object
 1   World Population  97 non-null     object
 2   Yearly Change     73 non-null     object
 3   Net Change        73 non-null     object
 4   Density (P/Km²)   73 non-null     object
dtypes: object(5)
memory usage: 3.9+ KB
```

In [76]:

```
df_html.head()
```

Out[76]:

	Year	World Population	Yearly Change	Net Change	Density (P/Km²)
0	2023	8045311447	0.88 %	70206291.0	54.0
1	2022	7975105156	0.83 %	65810005.0	54.0
2	2021	7909295151	0.87 %	68342271.0	53.0
3	2020	7840952880	0.98 %	76001848.0	53.0
4	2019	7764951032	1.06 %	81161204.0	52.0

In [79]:

```
df_html.head(20)
```

Out[79]:

	Year	World Population	Yearly Change	Net Change	Density (P/Km²)
0	2023	8045311447	0.88 %	70206291.0	54.0
1	2022	7975105156	0.83 %	65810005.0	54.0
2	2021	7909295151	0.87 %	68342271.0	53.0
3	2020	7840952880	0.98 %	76001848.0	53.0
4	2019	7764951032	1.06 %	81161204.0	52.0
5	2018	7683789828	1.10 %	83967424.0	52.0
6	2017	7599822404	1.15 %	86348166.0	51.0
7	2016	7513474238	1.17 %	86876701.0	50.0
8	2015	7426597537	1.19 %	87584118.0	50.0
9	2014	7339013419	1.22 %	88420049.0	49.0
10	2013	7250593370	1.24 %	88895449.0	49.0
11	2012	7161697921	1.25 %	88572496.0	48.0
12	2011	7073125425	1.25 %	87522320.0	47.0
13	2010	6985603105	1.27 %	87297197.0	47.0
14	2009	6898305908	1.27 %	86708636.0	46.0
15	2008	6811597272	1.27 %	85648728.0	46.0
16	2007	6725948544	1.27 %	84532326.0	45.0
17	2006	6641416218	1.27 %	83240099.0	45.0
18	2005	6558176119	1.27 %	82424641.0	44.0
19	2004	6475751478	1.28 %	81853113.0	43.0

clipboard

In [94]:

```
df_clip=pd.read_clipboard() # Reads data from the clipboard and returns a Pandas DataFrame object
```

In [95]:

```
df_clip
```

Out[95]:

	GCAG	2016	0.9363
0	GISTEMP	2016	0.9900
1	GCAG	2015	0.8998
2	GISTEMP	2015	0.8700
3	GCAG	2014	0.7408
4	GISTEMP	2014	0.7400
...	...	...	...
268	GISTEMP	1882	-0.1000
269	GCAG	1881	-0.0628
270	GISTEMP	1881	-0.1200
271	GCAG	1880	-0.1148
272	GISTEMP	1880	-0.2000

273 rows × 3 columns

In [96]:

```
df_clip.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 273 entries, 0 to 272
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   GCAG    273 non-null    object 
 1   2016    273 non-null    int64  
 2   0.9363  273 non-null    float64
dtypes: float64(1), int64(1), object(1)
memory usage: 6.5+ KB
```

In [97]:

```
df_clip.shape
```

Out[97]:

```
(273, 3)
```

In [99]:

```
df_clip.head(10)
```

Out[99]:

	GCAG	2016	0.9363
0	GISTEMP	2016	0.9900
1	GCAG	2015	0.8998
2	GISTEMP	2015	0.8700
3	GCAG	2014	0.7408
4	GISTEMP	2014	0.7400
5	GCAG	2013	0.6679
6	GISTEMP	2013	0.6500
7	GCAG	2012	0.6240
8	GISTEMP	2012	0.6300
9	GCAG	2011	0.5788

In [100]:

```
df_clip.tail()
```

Out[100]:

	GCAG	2016	0.9363
268	GISTEMP	1882	-0.1000
269	GCAG	1881	-0.0628
270	GISTEMP	1881	-0.1200
271	GCAG	1880	-0.1148
272	GISTEMP	1880	-0.2000

In [101]:

```
df_clip.isnull().any()
```

Out[101]:

```
GCAG      False
2016      False
0.9363    False
dtype: bool
```

In [102]:

```
df_clip.describe()
```

Out[102]:

	2016	0.9363
count	273.000000	273.000000
mean	1947.750916	0.033292
std	39.477053	0.315965
min	1880.000000	-0.470000
25%	1914.000000	-0.205500
50%	1948.000000	-0.056800
75%	1982.000000	0.227300
max	2016.000000	0.990000

In [ ]: