

database1

April 27, 2024

```
[ ]: #dataset1
    #27/04/2024
```

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

```
[2]: df = pd.read_csv('C:\\Users\\lenovo\\Desktop\\courses\\Task 1\\API_SP.POP.2529.
    ↪FE.5Y_DS2_en_csv_v2_52627.csv', sep=';')
```

```
[3]: df
```

```
[3]:
```

	Country Name	Country Code	Indicator Name	\
0	Aruba	ABW	Population, total	
1	Africa Eastern and Southern	AFE	Population, total	
2	Afghanistan	AFG	Population, total	
3	Africa Western and Central	AFW	Population, total	
4	Angola	AGO	Population, total	
..	
261	Kosovo	XKX	Population, total	
262	Yemen, Rep.	YEM	Population, total	
263	South Africa	ZAF	Population, total	
264	Zambia	ZMB	Population, total	
265	Zimbabwe	ZWE	Population, total	

	Indicator Code	1960	1961	1962	1963	\
0	SP.POP.TOTL	54608.0	55811.0	56682.0	57475.0	
1	SP.POP.TOTL	130692579.0	134169237.0	137835590.0	141630546.0	
2	SP.POP.TOTL	8622466.0	8790140.0	8969047.0	9157465.0	
3	SP.POP.TOTL	97256290.0	99314028.0	101445032.0	103667517.0	
4	SP.POP.TOTL	5357195.0	5441333.0	5521400.0	5599827.0	
..	
261	SP.POP.TOTL	947000.0	966000.0	994000.0	1022000.0	
262	SP.POP.TOTL	5542459.0	5646668.0	5753386.0	5860197.0	
263	SP.POP.TOTL	16520441.0	16989464.0	17503133.0	18042215.0	
264	SP.POP.TOTL	3119430.0	3219451.0	3323427.0	3431381.0	

265	SP.POP.TOTL	3806310.0	3925952.0	4049778.0	4177931.0
-----	-------------	-----------	-----------	-----------	-----------

	1964	1965	...	2013	2014	2015	\
0	58178.0	58782.0	...	102880.0	103594.0	104257.0	
1	145605995.0	149742351.0	...	567892149.0	583651101.0	600008424.0	
2	9355514.0	9565147.0	...	31541209.0	32716210.0	33753499.0	
3	105959979.0	108336203.0	...	387204553.0	397855507.0	408690375.0	
4	5673199.0	5736582.0	...	26147002.0	27128337.0	28127721.0	
..	
261	1050000.0	1078000.0	...	1818117.0	1812771.0	1788196.0	
262	5973803.0	6097298.0	...	26984002.0	27753304.0	28516545.0	
263	18603097.0	19187194.0	...	53873616.0	54729551.0	55876504.0	
264	3542764.0	3658024.0	...	15234976.0	15737793.0	16248230.0	
265	4310332.0	4447149.0	...	13555422.0	13855753.0	14154937.0	

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

	2021	2022
0	106537.0	106445.0
1	702977106.0	720859132.0
2	40099462.0	41128771.0
3	478185907.0	490330870.0
4	34503774.0	35588987.0
..
261	1786038.0	1761985.0
262	32981641.0	33696614.0
263	59392255.0	59893885.0
264	19473125.0	20017675.0
265	15993524.0	16320537.0

[266 rows x 67 columns]

[4]: df.head()

[4]:	Country Name	Country Code	Indicator Name	Indicator Code	\
0	Aruba	ABW	Population, total	SP.POP.TOTL	

1	Africa Eastern and Southern	AFE	Population, total	SP.POP.TOTL
2	Afghanistan	AFG	Population, total	SP.POP.TOTL
3	Africa Western and Central	AFW	Population, total	SP.POP.TOTL
4	Angola	AGO	Population, total	SP.POP.TOTL

	1960	1961	1962	1963	1964 \
0	54608.0	55811.0	56682.0	57475.0	58178.0
1	130692579.0	134169237.0	137835590.0	141630546.0	145605995.0
2	8622466.0	8790140.0	8969047.0	9157465.0	9355514.0
3	97256290.0	99314028.0	101445032.0	103667517.0	105959979.0
4	5357195.0	5441333.0	5521400.0	5599827.0	5673199.0

	1965 ...	2013	2014	2015	2016 \
0	58782.0 ...	102880.0	103594.0	104257.0	104874.0
1	149742351.0 ...	567892149.0	583651101.0	600008424.0	616377605.0
2	9565147.0 ...	31541209.0	32716210.0	33753499.0	34636207.0
3	108336203.0 ...	387204553.0	397855507.0	408690375.0	419778384.0
4	5736582.0 ...	26147002.0	27128337.0	28127721.0	29154746.0

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

	2022
0	106445.0
1	720859132.0
2	41128771.0
3	490330870.0
4	35588987.0

[5 rows x 67 columns]

```
[5]: df.columns
```

```
[5]: 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')
```

```
[6]: df.dtypes
```

```
[6]: 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
```

```
[7]: df.describe()
```

```
[7]:
```

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

	1965	1966	1967	1968	1969 \
count	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02	2.640000e+02
mean	1.291813e+08	1.320404e+08	1.348980e+08	1.378358e+08	1.408789e+08
std	4.071153e+08	4.164504e+08	4.257424e+08	4.353218e+08	4.452927e+08
min	4.161000e+03	4.531000e+03	4.930000e+03	5.354000e+03	5.646000e+03
25%	5.651150e+05	5.691470e+05	5.773872e+05	5.832700e+05	5.875942e+05
50%	4.277636e+06	4.331825e+06	4.385700e+06	4.450934e+06	4.530800e+06
75%	3.084892e+07	3.163010e+07	3.209247e+07	3.249927e+07	3.277149e+07
max	3.328209e+09	3.398480e+09	3.468371e+09	3.540164e+09	3.614573e+09

	...	2013	2014	2015	2016 \
count	...	2.650000e+02	2.650000e+02	2.650000e+02	2.650000e+02
mean	...	2.927593e+08	2.966353e+08	3.004815e+08	3.043263e+08
std	...	9.186237e+08	9.300118e+08	9.412522e+08	9.524238e+08
min	...	1.069400e+04	1.089900e+04	1.087700e+04	1.085200e+04
25%	...	1.697753e+06	1.743309e+06	1.788196e+06	1.777557e+06
50%	...	1.014958e+07	1.028212e+07	1.035808e+07	1.032545e+07
75%	...	6.023395e+07	6.078914e+07	6.073058e+07	6.062750e+07
max	...	7.229303e+09	7.317040e+09	7.403850e+09	7.490415e+09

	2017	2018	2019	2020	2021 \
count	2.650000e+02	2.650000e+02	2.650000e+02	2.650000e+02	2.650000e+02
mean	3.081770e+08	3.119641e+08	3.156701e+08	3.192595e+08	3.225221e+08
std	9.636041e+08	9.744884e+08	9.850405e+08	9.951225e+08	1.004225e+09
min	1.082800e+04	1.086500e+04	1.095600e+04	1.106900e+04	1.120400e+04
25%	1.791003e+06	1.797085e+06	1.788878e+06	1.790133e+06	1.786038e+06
50%	1.030030e+07	1.039533e+07	1.044767e+07	1.060623e+07	1.050577e+07
75%	6.053671e+07	6.042176e+07	5.987258e+07	6.170452e+07	6.358833e+07
max	7.576442e+09	7.660371e+09	7.741775e+09	7.820206e+09	7.888306e+09

	2022
count	2.650000e+02
mean	3.254733e+08
std	1.012163e+09
min	1.131200e+04
25%	1.761985e+06
50%	1.048694e+07
75%	6.549775e+07
max	7.950947e+09

[8 rows x 63 columns]

```
[8]: df.duplicated().sum()
```

```
[8]: 0
```

```
[9]: df.isna().sum().any()
```

```
[9]: True
```

```
[10]: df = df.fillna(method="ffill")
```

```
[11]: df.isna().sum().any()
```

```
[11]: False
```

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

```
[13]: df.columns
```

```
[13]: 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')

```

```

[14]: import matplotlib.pyplot as plt

# Créer une figure et un axe
fig, ax = plt.subplots(figsize=(12, 6))

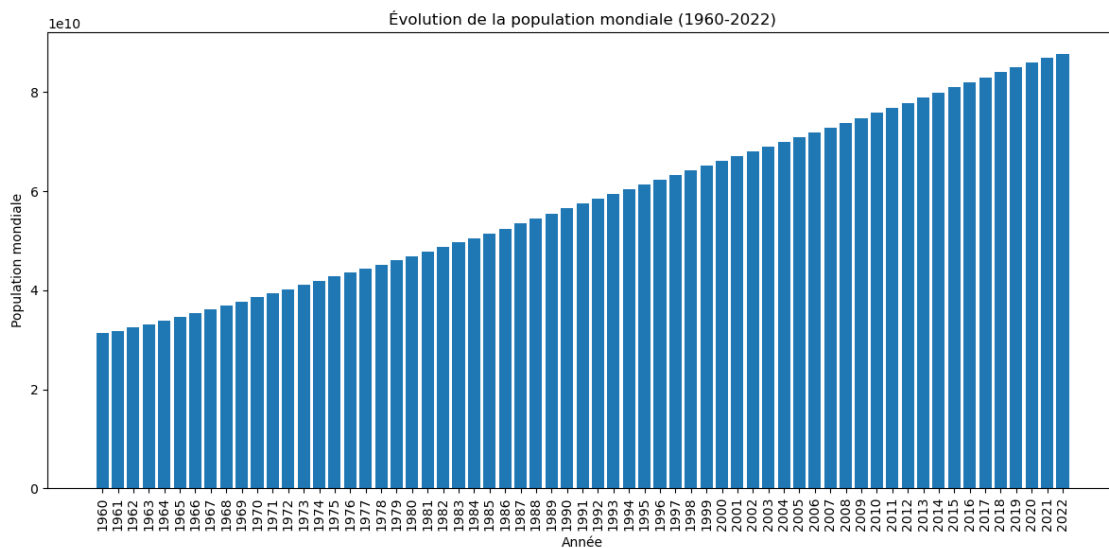
# Calculer la population mondiale par année en faisant la somme sur toutes les
↳ lignes
world_population = df.iloc[:, 1:].sum()

# Créer un graphique à barres horizontales (histogramme)
ax.bar(world_population.index, world_population.values)

# Ajouter des titres et des étiquettes
ax.set_title('Évolution de la population mondiale (1960-2022)')
ax.set_xlabel('Année')
ax.set_ylabel('Population mondiale')

# Faire pivoter les étiquettes d'année pour une meilleure lisibilité
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()

```



```

[ ]: import matplotlib.pyplot as plt

```

```

data = df.set_index('Country Name').loc[:, '1960':'2022']

data = data.T

# Créer un graphique pour chaque pays
for country in data.columns:
    plt.figure(figsize=(12, 6))
    plt.bar(data.index, data[country])
    plt.title(f'Population de {country} (1960-2022)')
    plt.xlabel('Année')
    plt.ylabel('Population')
    plt.xticks(rotation=90)
    plt.tight_layout()
    plt.show()

```

