

```
import pandas as pd
import numpy as np
import plotly.express as px
```

```
country = pd.read_csv("/content/Metadata_Country.csv")
fertility = pd.read_csv("/content/fertility_rate.csv")
population = pd.read_csv("/content/country_population.csv")
life_expectancy = pd.read_csv("/content/life_expectancy.csv")
```

```
country.head()
```

	Country Code	Region	IncomeGroup	SpecialNotes	TableName	Unnamed: 5
0	ABW	Latin America & Caribbean	High income	SNA data for 2000-2011 are updated from offici...	Aruba	NaN
1	AFG	South Asia	Low income	Fiscal year end: March 20; reporting period fo...	Afghanistan	NaN
		Sub-Saharan	Lower middle			

```
country.isna().sum()
```

```
Country Code    0
Region          46
IncomeGroup     46
SpecialNotes    116
TableName       0
Unnamed: 5      263
dtype: int64
```


```
fertility.head()
```

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	19
0	Aruba	ABW	Fertility rate, total (births per woman)	SP.DYN.TFRT.IN	4.820	4.655	4.471	4.271	4.059	3.8
1	Afghanistan	AFG	Fertility rate, total (births per woman)	SP.DYN.TFRT.IN	7.450	7.450	7.450	7.450	7.450	7.4
2	Angola	AGO	Fertility rate, total (births per woman)	SP.DYN.TFRT.IN	7.478	7.524	7.563	7.592	7.611	7.6

```
fertility.isna().sum()
```



```
Country Name    0
Country Code    0
Indicator Name   0
Indicator Code   0
1960            28
..
2012            16
2013            18
2014            18
2015            17
2016            18
Length: 61, dtype: int64
```

```
population.head()
```




	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1
0	Aruba	ABW	Population, total	SP.POP.TOTL	54211.0	55438.0	56225.0	5661
1	Afghanistan	AFG	Population, total	SP.POP.TOTL	8996351.0	9166764.0	9345868.0	953391
2	Angola	AGO	Population, total	SP.POP.TOTL	5643182.0	5753024.0	5866061.0	598041
3	Albania	ALB	Population, total	SP.POP.TOTL	1608800.0	1659800.0	1711319.0	176261
4	Andorra	AND	Population, total	SP.POP.TOTL	13411.0	14375.0	15370.0	1641

5 rows × 61 columns




```
population.isna().sum()
```





```
Country Name      0
Country Code      0
Indicator Name     0
Indicator Code     0
1960               4
...
2012               2
2013               2
2014               2
2015               2
2016               2
Length: 61, dtype: int64
```

```
life_expectancy.head()
```



	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964
0	Aruba	ABW	Life expectancy at birth, total (years)	SP.DYN.LE00.IN	65.662	66.074	66.444	66.787	67.111
1	Afghanistan	AFG	Life expectancy at birth, total (years)	SP.DYN.LE00.IN	32.292	32.742	33.185	33.624	34.061
2	Angola	AGO	Life expectancy at birth, total (years)	SP.DYN.LE00.IN	33.251	33.573	33.914	34.272	34.641
3	Albania	ALB	Life expectancy at birth, total (years)	SP.DYN.LE00.IN	62.279	63.298	64.187	64.911	65.461
4	Andorra	AND	Life expectancy at birth, total (years)	SP.DYN.LE00.IN	NaN	NaN	NaN	NaN	NaN

5 rows × 61 columns



```
# Function for data preprocecing
def preprocess_df(df, value_name):

    # create list for years
    years = [str(y) for y in range(1960, 2017)]

    # remove useless columns
    df.drop(['Country Name', 'Indicator Name', 'Indicator Code'], axis=1, inplace=True)


    # remove countries with missing value
    df.dropna(axis=0, inplace=True)

    # melt the dataframe to have years in one columns
    df = pd.melt(df,
                 id_vars='Country Code',
                 value_vars=years,
                 var_name='Year',
                 value_name=value_name)

    return df
```


```
country = country[['Country Code', 'Region']]
population = preprocess_df(population, 'Population')
fertility = preprocess_df(fertility, 'Fertility')
life_expectancy = preprocess_df(life_expectancy, 'Life Expectancy')
```

```
country.head()
```




	Country Code	Region
0	ABW	Latin America & Caribbean
1	AFG	South Asia
2	AGO	Sub-Saharan Africa
3	ALB	Europe & Central Asia
4	AND	Europe & Central Asia

```
fertility.head()
```




	Country Code	Year	Fertility
0	ABW	1960	4.820000
1	AFG	1960	7.450000
2	AGO	1960	7.478000
3	ALB	1960	6.489000
4	ARB	1960	6.948747

```
population.head()
```



	Country Code	Year	Population
0	ABW	1960	54211.0
1	AFG	1960	8996351.0
2	AGO	1960	5643182.0
3	ALB	1960	1608800.0
4	AND	1960	13411.0

```
life_expectancy.head()
```



	Country Code	Year	Life Expectancy
0	ABW	1960	65.662000
1	AFG	1960	32.292000
2	AGO	1960	33.251000
3	ALB	1960	62.279000
4	ARB	1960	46.825065

country



	Country Code	Region
0	ABW	Latin America & Caribbean
1	AFG	South Asia
2	AGO	Sub-Saharan Africa
3	ALB	Europe & Central Asia
4	AND	Europe & Central Asia
...
258	XKX	Europe & Central Asia
259	YEM	Middle East & North Africa
260	ZAF	Sub-Saharan Africa
261	ZMB	Sub-Saharan Africa
262	ZWE	Sub-Saharan Africa

263 rows × 2 columns

```
df = pd.merge(country, population, how='right', on='Country Code')
df
```



	Country Code	Region	Year	Population
0	ABW	Latin America & Caribbean	1960	54211.0
1	AFG	South Asia	1960	8996351.0
2	AGO	Sub-Saharan Africa	1960	5643182.0
3	ALB	Europe & Central Asia	1960	1608800.0
4	AND	Europe & Central Asia	1960	13411.0
...
14701	XKX	Europe & Central Asia	2016	1816200.0
14702	YEM	Middle East & North Africa	2016	27584213.0
14703	ZAF	Sub-Saharan Africa	2016	56015473.0
14704	ZMB	Sub-Saharan Africa	2016	16591390.0
14705	ZWE	Sub-Saharan Africa	2016	16150362.0

14706 rows × 4 columns

```
df = pd.merge(df, life_expectancy, how='left', on=['Country Code', 'Year'])
df
```



	Country Code	Region	Year	Population	Life Expectancy
0	ABW	Latin America & Caribbean	1960	54211.0	65.662
1	AFG	South Asia	1960	8996351.0	32.292
2	AGO	Sub-Saharan Africa	1960	5643182.0	33.251
3	ALB	Europe & Central Asia	1960	1608800.0	62.279
4	AND	Europe & Central Asia	1960	13411.0	NaN
...
14701	XKX	Europe & Central Asia	2016	1816200.0	NaN
14702	YEM	Middle East & North Africa	2016	27584213.0	64.953
14703	ZAF	Sub-Saharan Africa	2016	56015473.0	62.774
14704	ZMB	Sub-Saharan Africa	2016	16591390.0	61.874
14705	ZWE	Sub-Saharan Africa	2016	16150362.0	61.163

14706 rows × 5 columns

```
df = pd.merge(df, fertility, how='left', on=['Country Code', 'Year'])
df
```



	Country Code	Region	Year	Population	Life Expectancy	Fertility
0	ABW	Latin America & Caribbean	1960	54211.0	65.662	4.820
1	AFG	South Asia	1960	8996351.0	32.292	7.450
2	AGO	Sub-Saharan Africa	1960	5643182.0	33.251	7.478
3	ALB	Europe & Central Asia	1960	1608800.0	62.279	6.489
4	AND	Europe & Central Asia	1960	13411.0	NaN	NaN
...
14701	XXK	Europe & Central Asia	2016	1816200.0	NaN	NaN
14702	YEM	Middle East & North Africa	2016	27584213.0	64.953	3.995
14703	ZAF	Sub-Saharan Africa	2016	56015473.0	62.774	2.458
14704	ZMB	Sub-Saharan Africa	2016	16591390.0	61.874	4.981
14705	ZWF	Sub-Saharan Africa	2016	16150362.0	61.163	3.760

```
df.dropna(axis=0, inplace=True)
```

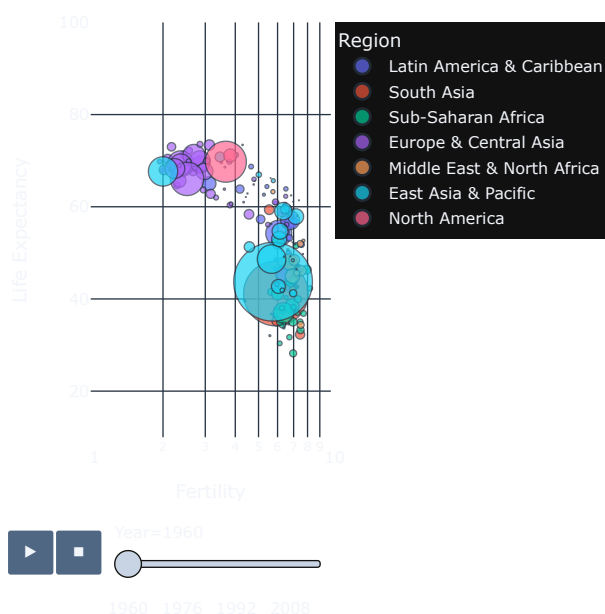
```
df.head()
```



	Country Code	Region	Year	Population	Life Expectancy	Fertility
0	ABW	Latin America & Caribbean	1960	54211.0	65.662	4.820
1	AFG	South Asia	1960	8996351.0	32.292	7.450
2	AGO	Sub-Saharan Africa	1960	5643182.0	33.251	7.478
3	ALB	Europe & Central Asia	1960	1608800.0	62.279	6.489

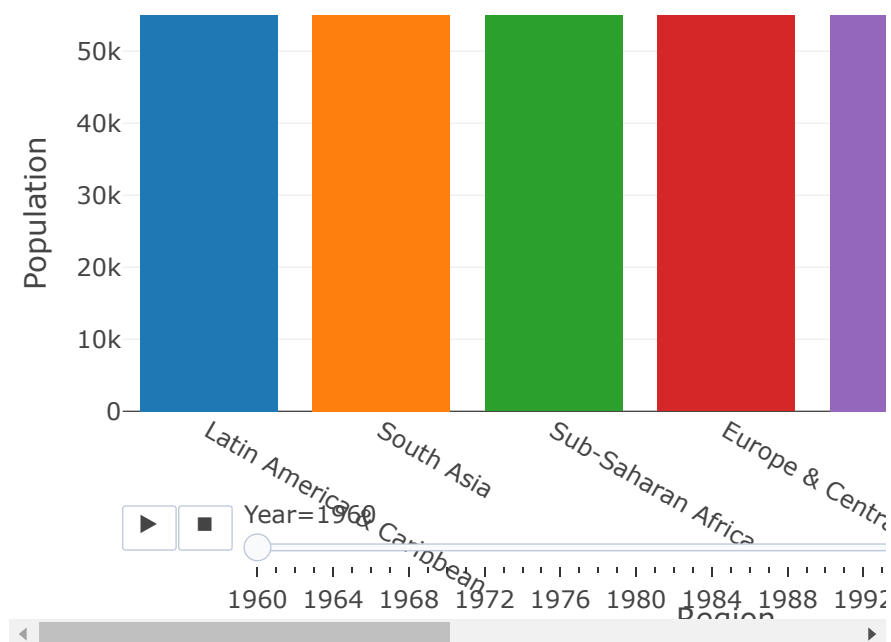
✓ SCATTER PLOT(LIFE_EXPECTANCY VS FERTILITY)

```
px.scatter(df,
            x="Fertility",
            y="Life Expectancy",
            animation_frame="Year",
            animation_group="Country Code",
            size="Population",
            color="Region",
            hover_name="Country Code",
            log_x=True,
            size_max=60,
            range_x=[1,10],
            range_y=[10,100],
            template="plotly_dark")
```



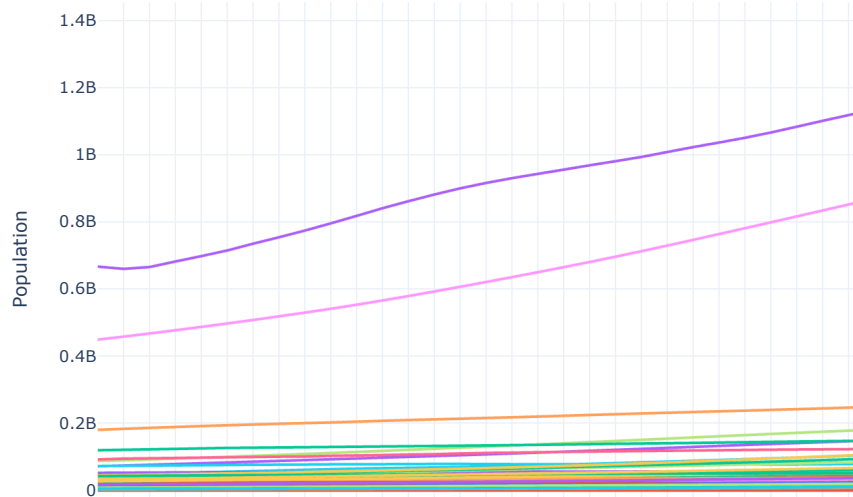
POPULATION AMONG REGIONS

```
px.bar(df,
       x="Region",
       y="Population",
       animation_frame="Year",
       animation_group="Country Code",
       color="Region",
       range_y=[0,55000],
       template="presentation")
```



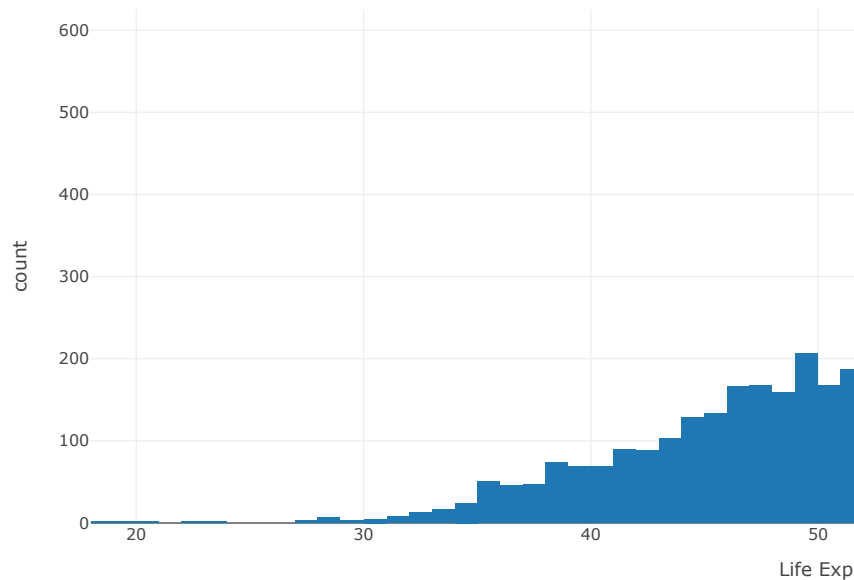
POPULATION COMPARISON AMONG YEARS

```
px.line(df,
        x='Year',
        y='Population',
        color="Country Code",
        # animation_frame="Year",
        # animation_group="Country Code",
        # range_y=[0,2500000000],
        template="plotly_white"
    )
```



✓ LIFE_EXPECTANCY DISTRIBUTION

```
px.histogram(df,
             x='Life Expectancy',
             template='gridon')
```



✓ FERTILITY RATE DISTRIBUTION

```
px.histogram(df,
             x='Fertility',
             template='ygridoff')
```