

# WorldBank\_Task

January 6, 2024

## 1 World Bank Data Task

### 1.1 1.Import Libraries

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

### 1.2 2. Load datasets

```
[593]: Country=pd.read_csv("/content/drive/MyDrive/Prepinsta intern/Metadata_Country.
↪csv")
population=pd.read_csv("/content/drive/MyDrive/Prepinsta intern/
↪country_population.csv")
fertility=pd.read_csv("/content/drive/MyDrive/Prepinsta intern/fertility_rate.
↪csv")
life_expectancy=pd.read_csv("/content/drive/MyDrive/Prepinsta intern/
↪life_expectancy.csv")
```

### 1.3 3. Data preprocessing

#### 1.3.1 3.1 Handle the country dataset and perform some operations...

```
[594]: Country
```

```
[594]:
```

	Country Code	Region	IncomeGroup \
0	ABW	Latin America & Caribbean	High income
1	AFG	South Asia	Low income
2	AGO	Sub-Saharan Africa	Lower middle income
3	ALB	Europe & Central Asia	Upper middle income
4	AND	Europe & Central Asia	High income
..	...	...	...
258	XKX	Europe & Central Asia	Lower middle income
259	YEM	Middle East & North Africa	Lower middle income
260	ZAF	Sub-Saharan Africa	Upper middle income

261	ZMB	Sub-Saharan Africa	Lower middle income
262	ZWE	Sub-Saharan Africa	Low income

	SpecialNotes	TableName \
0	SNA data for 2000-2011 are updated from offici...	Aruba
1	Fiscal year end: March 20; reporting period fo...	Afghanistan
2		NaN Angola
3		NaN Albania
4	WB-3 code changed from ADO to AND to align wit...	Andorra
..	...	...
258	WB-3 code changed from KSV to KXX to align wit...	Kosovo
259	Based on official government statistics and In...	Yemen, Rep.
260	Fiscal year end: March 31; reporting period fo...	South Africa
261	The base year is 2010. National accounts data ...	Zambia
262	Fiscal year end: June 30; reporting period for...	Zimbabwe

Unnamed: 5

0	NaN
1	NaN
2	NaN
3	NaN
4	NaN
..	...
258	NaN
259	NaN
260	NaN
261	NaN
262	NaN

[263 rows x 6 columns]

```
[595]: Country[["Country Code","Region"]]
```

```
[595]:
```

	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	KXX	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 x 2 columns]

```
[596]: Country=Country[["Country Code","Region"]]
```

```
[597]: Country
```

```
[597]:      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 x 2 columns]
```

```
[598]: Country.shape
```

```
[598]: (263, 2)
```

```
[599]: Country.info()
```

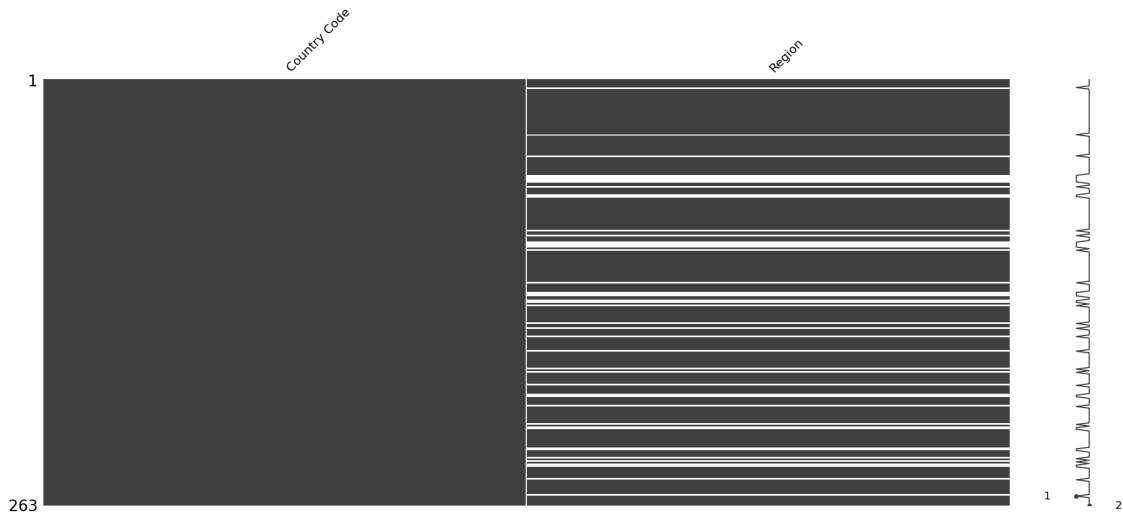
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 263 entries, 0 to 262
Data columns (total 2 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Country Code 263 non-null   object
1   Region       217 non-null   object
dtypes: object(2)
memory usage: 4.2+ KB
```

```
[600]: Country.isna().sum()
```

```
[600]: Country Code    0
Region            46
dtype: int64
```

```
[601]: mn.matrix(Country)
```

```
[601]: <Axes: >
```



### 3.1.1 Handle the missing values

```
[602]: Country.dropna(axis=0,inplace=True)
```

<ipython-input-602-82496ab87803>:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

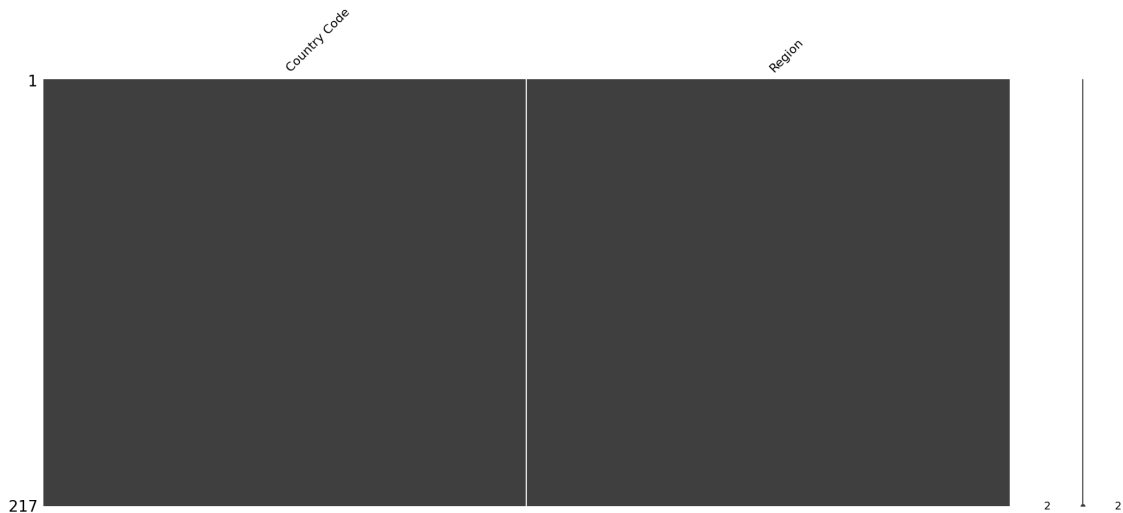
See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
[603]: Country.shape
```

```
[603]: (217, 2)
```

```
[604]: mn.matrix(Country)
```

```
[604]: <Axes: >
```



### 1.3.2 3.2 Handle the population dataset and perform some operations...

```
[605]: population.head()
```

```
[605]: Country Name Country Code Indicator Name Indicator Code 1960 \
0 Aruba ABW Population, total SP.POP.TOTL 54211.0
1 Afghanistan AFG Population, total SP.POP.TOTL 8996351.0
2 Angola AGO Population, total SP.POP.TOTL 5643182.0
3 Albania ALB Population, total SP.POP.TOTL 1608800.0
4 Andorra AND Population, total SP.POP.TOTL 13411.0

1961 1962 1963 1964 1965 ... 2007 \
0 55438.0 56225.0 56695.0 57032.0 57360.0 ... 101220.0
1 9166764.0 9345868.0 9533954.0 9731361.0 9938414.0 ... 26616792.0
2 5753024.0 5866061.0 5980417.0 6093321.0 6203299.0 ... 20997687.0
3 1659800.0 1711319.0 1762621.0 1814135.0 1864791.0 ... 2970017.0
4 14375.0 15370.0 16412.0 17469.0 18549.0 ... 82683.0

2008 2009 2010 2011 2012 2013 \
0 101353.0 101453.0 101669.0 102053.0 102577.0 103187.0
1 27294031.0 28004331.0 28803167.0 29708599.0 30696958.0 31731688.0
2 21759420.0 22549547.0 23369131.0 24218565.0 25096150.0 25998340.0
3 2947314.0 2927519.0 2913021.0 2905195.0 2900401.0 2895092.0
4 83861.0 84462.0 84449.0 83751.0 82431.0 80788.0

2014 2015 2016
0 103795.0 104341.0 104822.0
1 32758020.0 33736494.0 34656032.0
2 26920466.0 27859305.0 28813463.0
```

```

3    2889104.0    2880703.0    2876101.0
4      79223.0      78014.0      77281.0

```

```
[5 rows x 61 columns]
```

```
[606]: population.shape
```

```
[606]: (264, 61)
```

```
[607]: population.columns
```

```
[607]: 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'],
            dtype='object')
```

```
[608]: population.drop(['Country Name', 'Indicator Name', 'Indicator_
↳ Code'],axis=1,inplace=True)
```

```
[609]: population.head()
```

```
[609]:
```

	Country Code	1960	1961	1962	1963	1964	\
0	ABW	54211.0	55438.0	56225.0	56695.0	57032.0	
1	AFG	8996351.0	9166764.0	9345868.0	9533954.0	9731361.0	
2	AGO	5643182.0	5753024.0	5866061.0	5980417.0	6093321.0	
3	ALB	1608800.0	1659800.0	1711319.0	1762621.0	1814135.0	
4	AND	13411.0	14375.0	15370.0	16412.0	17469.0	

	1965	1966	1967	1968	...	2007	2008	\
0	57360.0	57715.0	58055.0	58386.0	...	101220.0	101353.0	
1	9938414.0	10152331.0	10372630.0	10604346.0	...	26616792.0	27294031.0	
2	6203299.0	6309770.0	6414995.0	6523791.0	...	20997687.0	21759420.0	
3	1864791.0	1914573.0	1965598.0	2022272.0	...	2970017.0	2947314.0	
4	18549.0	19647.0	20758.0	21890.0	...	82683.0	83861.0	

	2009	2010	2011	2012	2013	2014	\
0	101453.0	101669.0	102053.0	102577.0	103187.0	103795.0	
1	28004331.0	28803167.0	29708599.0	30696958.0	31731688.0	32758020.0	
2	22549547.0	23369131.0	24218565.0	25096150.0	25998340.0	26920466.0	
3	2927519.0	2913021.0	2905195.0	2900401.0	2895092.0	2889104.0	
4	84462.0	84449.0	83751.0	82431.0	80788.0	79223.0	

	2015	2016
0	104341.0	104822.0
1	33736494.0	34656032.0
2	27859305.0	28813463.0
3	2880703.0	2876101.0
4	78014.0	77281.0

[5 rows x 58 columns]

### 3.2.1 Handle the missing values

```
[610]: population.isna().sum()
```

```
[610]: Country Code    0
1960                  4
1961                  4
1962                  4
1963                  4
1964                  4
1965                  4
1966                  4
1967                  4
1968                  4
1969                  4
1970                  4
1971                  4
1972                  4
1973                  4
1974                  4
1975                  4
1976                  4
1977                  4
1978                  4
1979                  4
1980                  4
1981                  4
1982                  4
1983                  4
1984                  4
1985                  4
1986                  4
1987                  4
1988                  4
1989                  4
1990                  2
1991                  2
1992                  3
```

1993	3
1994	3
1995	2
1996	2
1997	2
1998	1
1999	1
2000	1
2001	1
2002	1
2003	1
2004	1
2005	1
2006	1
2007	1
2008	1
2009	1
2010	1
2011	1
2012	2
2013	2
2014	2
2015	2
2016	2

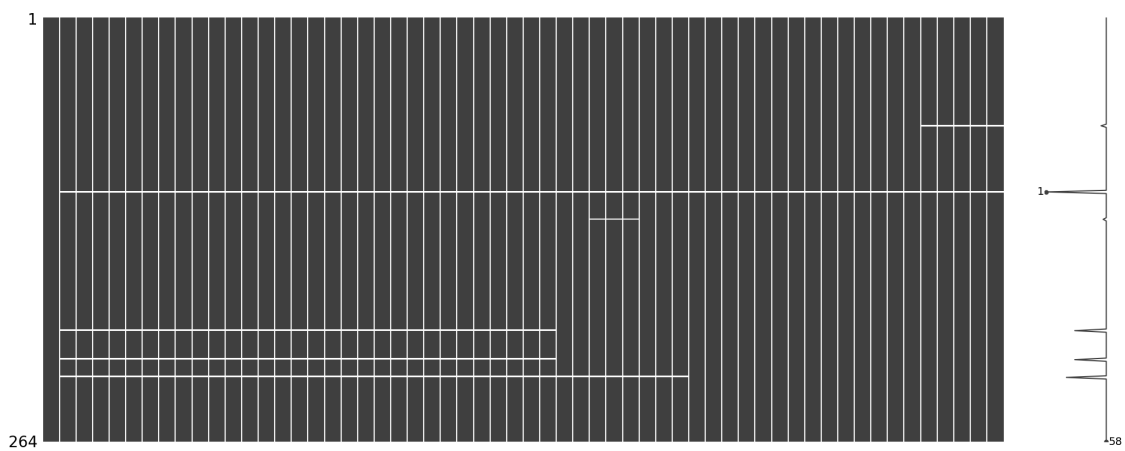
dtype: int64

```
[611]: population.shape
```

```
[611]: (264, 58)
```

```
[612]: mn.matrix(population)
```

```
[612]: <Axes: >
```





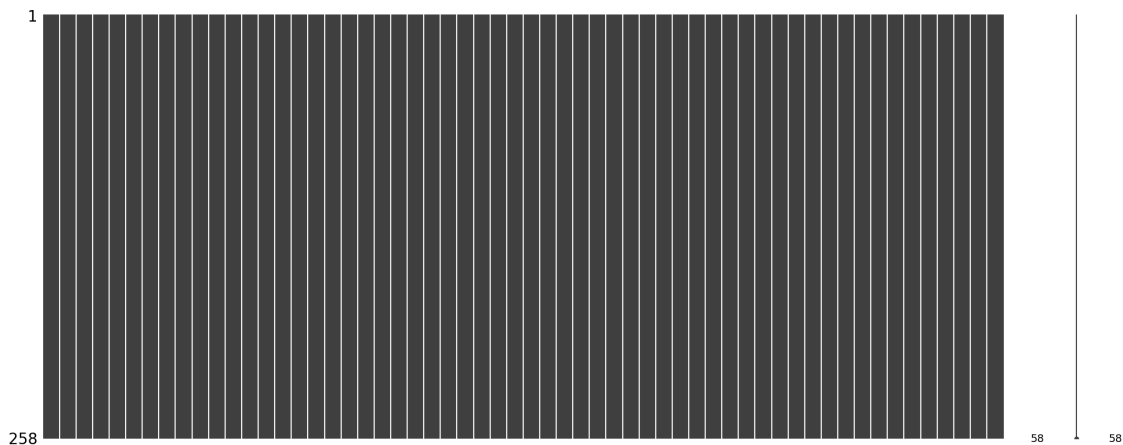
```
[613]: population.dropna(axis=0,inplace=True)
```

```
[614]: population.shape
```

```
[614]: (258, 58)
```

```
[615]: mn.matrix(population)
```

```
[615]: <Axes: >
```



**3.2.2 melt() is to be performed..** melt function is used to unpivot the ‘Course’ column while keeping ‘Name’ as the identifier variable.

```
[616]: years=[str(i) for i in range(1960,2017)]  
print(years)
```

```
['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']
```

```
[617]: df=pd.melt(population,  
                 id_vars='Country Code',  
                 value_vars=years,  
                 var_name='year',  
                 value_name='population')
```

```
[618]: df.shape
```

```
[618]: (14706, 3)
```

```
[619]: df.head()
```

```
[619]:   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
```

```
[620]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14706 entries, 0 to 14705
Data columns (total 3 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Country Code    14706 non-null  object
1   year            14706 non-null  object
2   population      14706 non-null  float64
dtypes: float64(1), object(2)
memory usage: 344.8+ KB
```

```
[621]: Country.head()
```

```
[621]:   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
```

```
[622]: df.head()
```

```
[622]:   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
```

**3.2.3 merge() is to be performed..** merge(): To combine the datasets on common column or index or both.

```
[623]: country_population_merge =pd.merge(Country,df,how='left',on='Country Code')
```

```
[624]: country_population_merge
```

```
[624]:      Country Code      Region  year  population
0          ABW  Latin America & Caribbean  1960      54211.0
1          ABW  Latin America & Caribbean  1961      55438.0
2          ABW  Latin America & Caribbean  1962      56225.0
3          ABW  Latin America & Caribbean  1963      56695.0
4          ABW  Latin America & Caribbean  1964      57032.0
...
12084      ZWE      Sub-Saharan Africa  2012  14710826.0
12085      ZWE      Sub-Saharan Africa  2013  15054506.0
12086      ZWE      Sub-Saharan Africa  2014  15411675.0
12087      ZWE      Sub-Saharan Africa  2015  15777451.0
12088      ZWE      Sub-Saharan Africa  2016  16150362.0
```

```
[12089 rows x 4 columns]
```

```
[625]: country_population_merge.shape
```

```
[625]: (12089, 4)
```

### 1.3.3 3.3 Handle the fertility dataset and perform some operations...

```
[626]: fertility.head()
```

```
[626]:      Country Name Country Code      Indicator Name \
0      Aruba      ABW  Fertility rate, total (births per woman)
1  Afghanistan      AFG  Fertility rate, total (births per woman)
2      Angola      AGO  Fertility rate, total (births per woman)
3  Albania      ALB  Fertility rate, total (births per woman)
4  Andorra      AND  Fertility rate, total (births per woman)
```

```
      Indicator Code  1960  1961  1962  1963  1964  1965  ...  2007 \
0  SP.DYN.TFRT.IN  4.820  4.655  4.471  4.271  4.059  3.842  ...  1.763
1  SP.DYN.TFRT.IN  7.450  7.450  7.450  7.450  7.450  7.450  ...  6.460
2  SP.DYN.TFRT.IN  7.478  7.524  7.563  7.592  7.611  7.619  ...  6.368
3  SP.DYN.TFRT.IN  6.489  6.401  6.282  6.133  5.960  5.773  ...  1.668
4  SP.DYN.TFRT.IN   NaN   NaN   NaN   NaN   NaN   NaN   ...  1.180
```

```
      2008  2009  2010  2011  2012  2013  2014  2015  2016
0  1.764  1.769  1.776  1.783  1.791  1.796  1.800  1.801  1.800
1  6.254  6.038  5.816  5.595  5.380  5.174  4.981  4.802  4.635
2  6.307  6.238  6.162  6.082  6.000  5.920  5.841  5.766  5.694
3  1.650  1.646  1.653  1.668  1.685  1.700  1.710  1.714  1.713
4  1.250  1.190  1.270   NaN   NaN   NaN   NaN   NaN   NaN
```

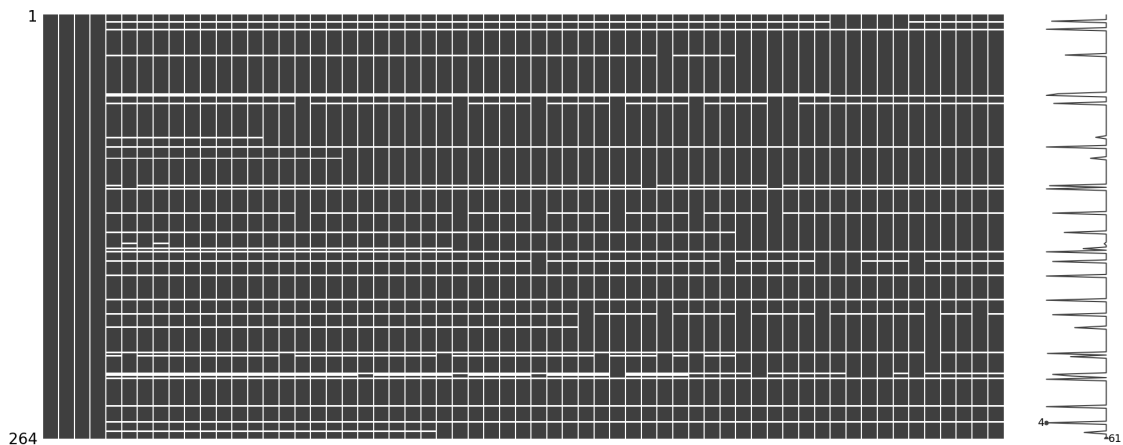
[5 rows x 61 columns]

```
[627]: fertility.isna().sum()
```

```
[627]: 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
```

```
[628]: mn.matrix(fertility)
```

```
[628]: <Axes: >
```



```
[629]: fertility.columns
```

```
[629]: 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'],
        dtype='object')
```

```
dtype='object')
```

```
[630]: fertility.shape
```

```
[630]: (264, 61)
```

### 3.3.1 Handle the missing values

```
[631]: fertility.isna().sum()
```

```
[631]: 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
```

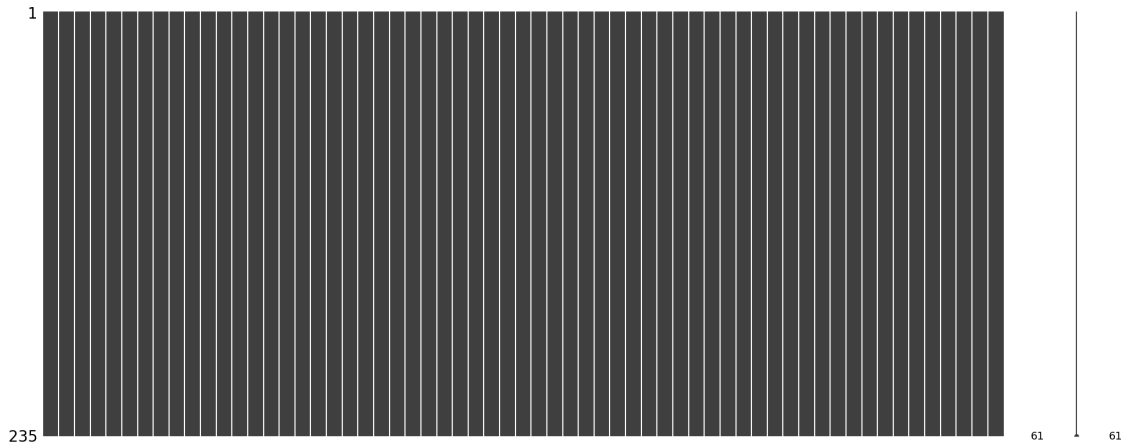
```
[632]: fertility.dropna(axis=0,inplace=True)
```

```
[633]: fertility.isna().sum()
```

```
[633]: Country Name      0
Country Code          0
Indicator Name        0
Indicator Code         0
1960                  0
..
2012                  0
2013                  0
2014                  0
2015                  0
2016                  0
Length: 61, dtype: int64
```

```
[634]: mn.matrix(fertility)
```

```
[634]: <Axes: >
```



```
[635]: fertility.shape
```

```
[635]: (235, 61)
```

```
[636]: fertility.drop(['Country Name', 'Indicator Name', 'Indicator_↵
↵Code'],axis=1,inplace=True)
```

```
[637]: fertility.head()
```

```
[637]: Country Code      1960      1961      1962      1963      1964      1965  \
0      ABW  4.820000  4.65500  4.47100  4.271000  4.059000  3.842000
1      AFG  7.450000  7.45000  7.45000  7.450000  7.450000  7.450000
2      AGO  7.478000  7.52400  7.56300  7.592000  7.611000  7.619000
3      ALB  6.489000  6.40100  6.28200  6.133000  5.960000  5.773000
5      ARB  6.948747  6.97137  6.99235  7.009265  7.020105  7.023159

      1966      1967      1968  ...      2007      2008      2009      2010  \
0  3.625000  3.417000  3.226000  ...  1.763000  1.764000  1.769000  1.776000
1  7.450000  7.450000  7.450000  ...  6.460000  6.254000  6.038000  5.816000
2  7.618000  7.613000  7.608000  ...  6.368000  6.307000  6.238000  6.162000
3  5.581000  5.394000  5.218000  ...  1.668000  1.650000  1.646000  1.653000
5  7.015824  6.998628  6.972205  ...  3.476805  3.470126  3.469736  3.471666

      2011      2012      2013      2014      2015      2016
0  1.783000  1.791000  1.796000  1.800000  1.80100  1.800000
1  5.595000  5.380000  5.174000  4.981000  4.80200  4.635000
2  6.082000  6.000000  5.920000  5.841000  5.76600  5.694000
3  1.668000  1.685000  1.700000  1.710000  1.71400  1.713000
5  3.470003  3.461347  3.442829  3.413299  3.37384  3.326532
```

```
[5 rows x 58 columns]
```

**3.3.2 melt() is to be performed..** melt function is used to unpivot the 'Course' column while keeping 'Name' as the identifier variable.

```
[638]: data=pd.melt(fertility,
                    id_vars='Country Code',
                    var_name='year',
                    value_name='fertility')
```

```
[639]: data
```

```
[639]:
```

	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
...	...	...	...
13390	WSM	2016	3.976000
13391	YEM	2016	3.995000
13392	ZAF	2016	2.458000
13393	ZMB	2016	4.981000
13394	ZWE	2016	3.760000

[13395 rows x 3 columns]

**3.3.3 merge() is to be performed..** merge(): To combine the datasets on common column or index or both.

```
[640]: df_fertility_merge=pd.
        merge(country_population_merge,data,how='left',on=['Country Code','year'])
```

```
[641]: df_fertility_merge.head()
```

```
[641]:
```

	Country Code	Region	year	population	fertility
0	ABW	Latin America & Caribbean	1960	54211.0	4.820
1	ABW	Latin America & Caribbean	1961	55438.0	4.655
2	ABW	Latin America & Caribbean	1962	56225.0	4.471
3	ABW	Latin America & Caribbean	1963	56695.0	4.271
4	ABW	Latin America & Caribbean	1964	57032.0	4.059

```
[642]: df_fertility_merge.shape
```

```
[642]: (12089, 5)
```

```
[643]: df_fertility_merge.isna().sum()
```

```
[643]: Country Code      0
      Region            0
      year              5
      population        5
      fertility        1430
      dtype: int64
```

```
[644]: df_fertility_merge.dropna(axis=0,inplace=True)
```

```
[645]: df_fertility_merge.isna().sum()
```

```
[645]: Country Code      0
      Region            0
      year              0
      population        0
      fertility          0
      dtype: int64
```

```
[646]: df_fertility_merge.shape
```

```
[646]: (10659, 5)
```

### 1.3.4 3.4 Handle the Life\_expectancy dataset and perform some operations...

```
[647]: life_expectancy.head()
```

```
[647]: Country Name Country Code Indicator Name \
0      Aruba      ABW Life expectancy at birth, total (years)
1  Afghanistan      AFG Life expectancy at birth, total (years)
2      Angola      AGO Life expectancy at birth, total (years)
3    Albania      ALB Life expectancy at birth, total (years)
4    Andorra      AND Life expectancy at birth, total (years)

Indicator Code  1960  1961  1962  1963  1964  1965  ... \
0  SP.DYN.LE00.IN  65.662  66.074  66.444  66.787  67.113  67.435  ...
1  SP.DYN.LE00.IN  32.292  32.742  33.185  33.624  34.060  34.495  ...
2  SP.DYN.LE00.IN  33.251  33.573  33.914  34.272  34.645  35.031  ...
3  SP.DYN.LE00.IN  62.279  63.298  64.187  64.911  65.461  65.848  ...
4  SP.DYN.LE00.IN   NaN   NaN   NaN   NaN   NaN   NaN   ...

      2007  2008  2009  2010  2011  2012  2013  2014  2015 \
0  74.576  74.725  74.872  75.016  75.158  75.299  75.440  75.582  75.725
1  59.694  60.243  60.754  61.226  61.666  62.086  62.494  62.895  63.288
2  55.096  56.189  57.231  58.192  59.042  59.770  60.373  60.858  61.241
3  75.656  75.943  76.281  76.652  77.031  77.389  77.702  77.963  78.174
4    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN
```



```

    2016
0  75.867
1  63.673
2  61.547
3  78.345
4      NaN

```

[5 rows x 61 columns]

```
[648]: life_expectancy.isna().sum()
```

```

[648]: Country Name      0
Country Code      0
Indicator Name     0
Indicator Code     0
1960              29
      ..
2012              17
2013              19
2014              19
2015              20
2016              19
Length: 61, dtype: int64

```

```
[649]: life_expectancy.shape
```

```
[649]: (264, 61)
```

```
[650]: life_expectancy.head()
```

```

[650]: Country Name Country Code      Indicator Name \
0      Aruba      ABW  Life expectancy at birth, total (years)
1  Afghanistan      AFG  Life expectancy at birth, total (years)
2      Angola      AGO  Life expectancy at birth, total (years)
3  Albania      ALB  Life expectancy at birth, total (years)
4  Andorra      AND  Life expectancy at birth, total (years)

      Indicator Code      1960      1961      1962      1963      1964      1965      ... \
0  SP.DYN.LE00.IN  65.662  66.074  66.444  66.787  67.113  67.435  ...
1  SP.DYN.LE00.IN  32.292  32.742  33.185  33.624  34.060  34.495  ...
2  SP.DYN.LE00.IN  33.251  33.573  33.914  34.272  34.645  35.031  ...
3  SP.DYN.LE00.IN  62.279  63.298  64.187  64.911  65.461  65.848  ...
4  SP.DYN.LE00.IN      NaN      NaN      NaN      NaN      NaN      NaN  ...

      2007      2008      2009      2010      2011      2012      2013      2014      2015 \
0  74.576  74.725  74.872  75.016  75.158  75.299  75.440  75.582  75.725
1  59.694  60.243  60.754  61.226  61.666  62.086  62.494  62.895  63.288

```

2	55.096	56.189	57.231	58.192	59.042	59.770	60.373	60.858	61.241
3	75.656	75.943	76.281	76.652	77.031	77.389	77.702	77.963	78.174
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

	2016
0	75.867
1	63.673
2	61.547
3	78.345
4	NaN

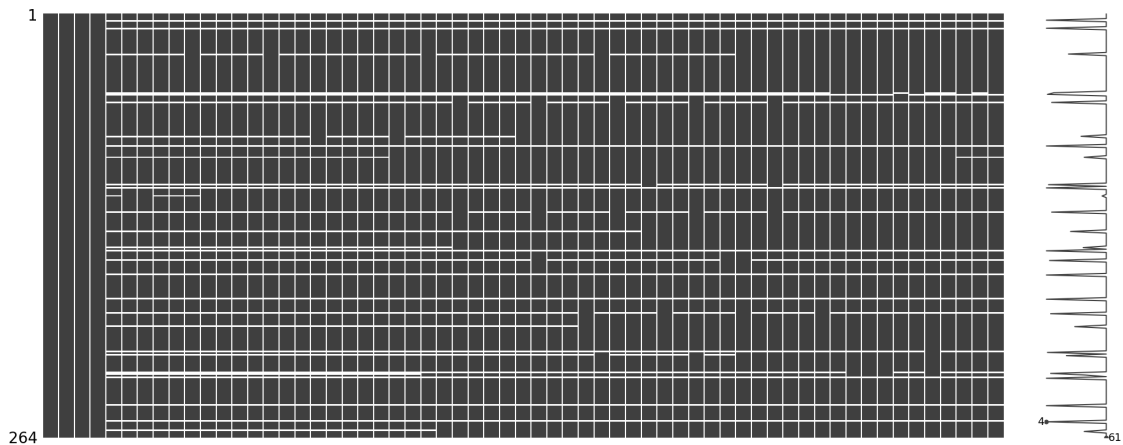
[5 rows x 61 columns]

```
[651]: life_expectancy.isna().sum()
```

```
[651]: Country Name      0
Country Code           0
Indicator Name         0
Indicator Code         0
1960                   29
..
2012                   17
2013                   19
2014                   19
2015                   20
2016                   19
Length: 61, dtype: int64
```

```
[652]: mn.matrix(life_expectancy)
```

```
[652]: <Axes: >
```



### 3.4.1 Handle the missing values

```
[653]: life_expectancy.shape
```

```
[653]: (264, 61)
```

```
[654]: life_expectancy.dropna(axis=0,inplace=True)
```

```
[655]: life_expectancy.isna().sum()
```

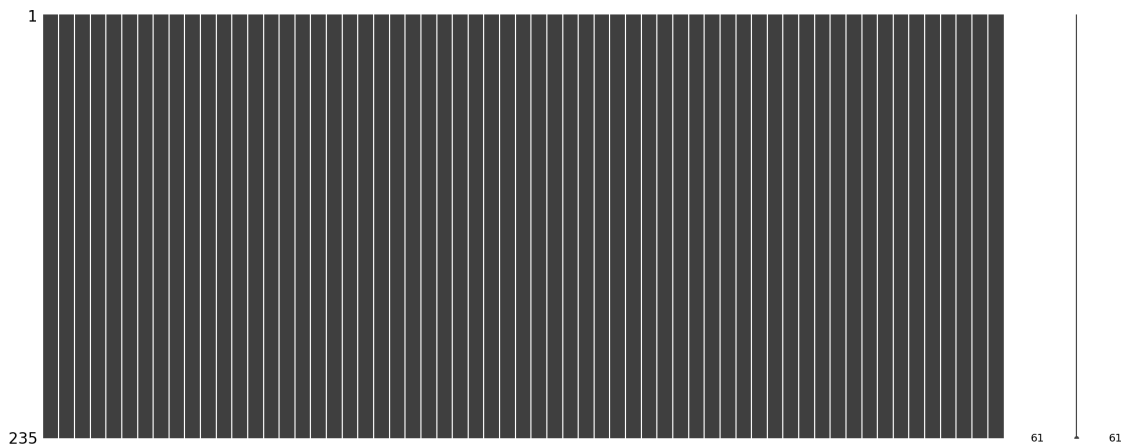
```
[655]: Country Name      0
      Country Code    0
      Indicator Name   0
      Indicator Code   0
      1960             0
           ..
      2012             0
      2013             0
      2014             0
      2015             0
      2016             0
      Length: 61, dtype: int64
```

```
[656]: life_expectancy.shape
```

```
[656]: (235, 61)
```

```
[657]: mn.matrix(life_expectancy)
```

```
[657]: <Axes: >
```



**3.4.2 melt() is to be performed..** melt function is used to unpivot the ‘Course’ column while keeping ‘Name’ as the identifier variable.

```
[658]: years=[str(i) for i in range(1960,2017)]
        print(years)
```

```
['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']
```

```
[659]: data_set=pd.melt(life_expectancy,
                        id_vars='Country Code',
                        value_vars=years,
                        var_name='year',
                        value_name='Expectancy')
```

```
[660]: data_set.head()
```

```
[660]:   Country Code  year  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
```

```
[661]: data_set.isna().sum()
```

```
[661]: Country Code    0
       year           0
       Expectancy     0
       dtype: int64
```

```
[662]: data_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13395 entries, 0 to 13394
Data columns (total 3 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Country Code    13395 non-null  object
1   year            13395 non-null  object
2   Expectancy      13395 non-null  float64
dtypes: float64(1), object(2)
memory usage: 314.1+ KB
```

**3.4.3 merge() is to be performed..** merge(): To combine the datasets on common column or index or both.

```
[663]: final_dataset=pd.merge(df_fertility_merge,data_set,how='left',on=['Country_↵
↵Code', 'year'])
```

```
[664]: final_dataset.head()
```

```
[664]:
```

	Country	Code	Region	year	population	fertility	\
0	ABW	Latin America & Caribbean	1960	54211.0	4.820		
1	ABW	Latin America & Caribbean	1961	55438.0	4.655		
2	ABW	Latin America & Caribbean	1962	56225.0	4.471		
3	ABW	Latin America & Caribbean	1963	56695.0	4.271		
4	ABW	Latin America & Caribbean	1964	57032.0	4.059		

	Expectancy
0	65.662
1	66.074
2	66.444
3	66.787
4	67.113

### 1.3.5 3.5 Final Dataset

```
[665]: final_dataset
```

```
[665]:
```

	Country	Code	Region	year	population	fertility	\
0	ABW	Latin America & Caribbean	1960	54211.0	4.820		
1	ABW	Latin America & Caribbean	1961	55438.0	4.655		
2	ABW	Latin America & Caribbean	1962	56225.0	4.471		
3	ABW	Latin America & Caribbean	1963	56695.0	4.271		
4	ABW	Latin America & Caribbean	1964	57032.0	4.059		
...	...	...	...	...	...		
10654	ZWE	Sub-Saharan Africa	2012	14710826.0	3.996		
10655	ZWE	Sub-Saharan Africa	2013	15054506.0	3.957		
10656	ZWE	Sub-Saharan Africa	2014	15411675.0	3.903		
10657	ZWE	Sub-Saharan Africa	2015	15777451.0	3.836		
10658	ZWE	Sub-Saharan Africa	2016	16150362.0	3.760		

	Expectancy
0	65.662
1	66.074
2	66.444
3	66.787
4	67.113
...	...
10654	56.516

```

10655      58.053
10656      59.360
10657      60.398
10658      61.163

```

```
[10659 rows x 6 columns]
```

```
[666]: final_dataset.columns
```

```
[666]: Index(['Country Code', 'Region', 'year', 'population', 'fertility',
          'Expectancy'],
          dtype='object')
```

```
[667]: final_dataset.shape
```

```
[667]: (10659, 6)
```

## 1.4 4.EDA for Visualization

```
[668]: final_dataset
```

```
[668]:
```

	Country Code	Region	year	population	fertility \
0	ABW	Latin America & Caribbean	1960	54211.0	4.820
1	ABW	Latin America & Caribbean	1961	55438.0	4.655
2	ABW	Latin America & Caribbean	1962	56225.0	4.471
3	ABW	Latin America & Caribbean	1963	56695.0	4.271
4	ABW	Latin America & Caribbean	1964	57032.0	4.059
...	...	...	...	...	...
10654	ZWE	Sub-Saharan Africa	2012	14710826.0	3.996
10655	ZWE	Sub-Saharan Africa	2013	15054506.0	3.957
10656	ZWE	Sub-Saharan Africa	2014	15411675.0	3.903
10657	ZWE	Sub-Saharan Africa	2015	15777451.0	3.836
10658	ZWE	Sub-Saharan Africa	2016	16150362.0	3.760

```

          Expectancy
0          65.662
1          66.074
2          66.444
3          66.787
4          67.113
...          ...
10654       56.516
10655       58.053
10656       59.360
10657       60.398
10658       61.163

```

[10659 rows x 6 columns]

```
[669]: import plotly.express as px
```

```
[670]: data_plot=px.scatter(data_frame=final_dataset, x="fertility",
                           y="Expectancy",
                           title="Fertility_rate vs Life_expectancy",
                           template='plotly_dark',
                           color="Region",
                           size="population",
                           size_max=50,
                           hover_data='Country Code',
                           animation_frame='year',
                           animation_group='Country Code',
                           range_x=[0,10],
                           range_y=[10,90])

data_plot.show()
```

```
[670]:
```