PROJECT INTRODUCTION

Fossil fuel use is that the primary supply of carbonic acid gas. carbonic acid gas also can be emitted from direct human-induced impacts on biology and different land use, like through deforestation, land clearing for agriculture, and degradation of soils. Likewise, land also can take away carbonic acid gas from the atmosphere through rehabilitation, improvement of soils, and different activities.

Global carbon emissions from fossil fuels have considerably inflated since 1900. Since 1970, carbonic acid gas emissions have inflated by concerning ninetieth, with emissions from fuel combustion and industrial processes contributive concerning seventy eight of the overall greenhouse emission emissions increase from 1970 to 2011. Agriculture, deforestation, and different land-use changes are the second-largest contributors.

ANALYSIS OBJECTIVES

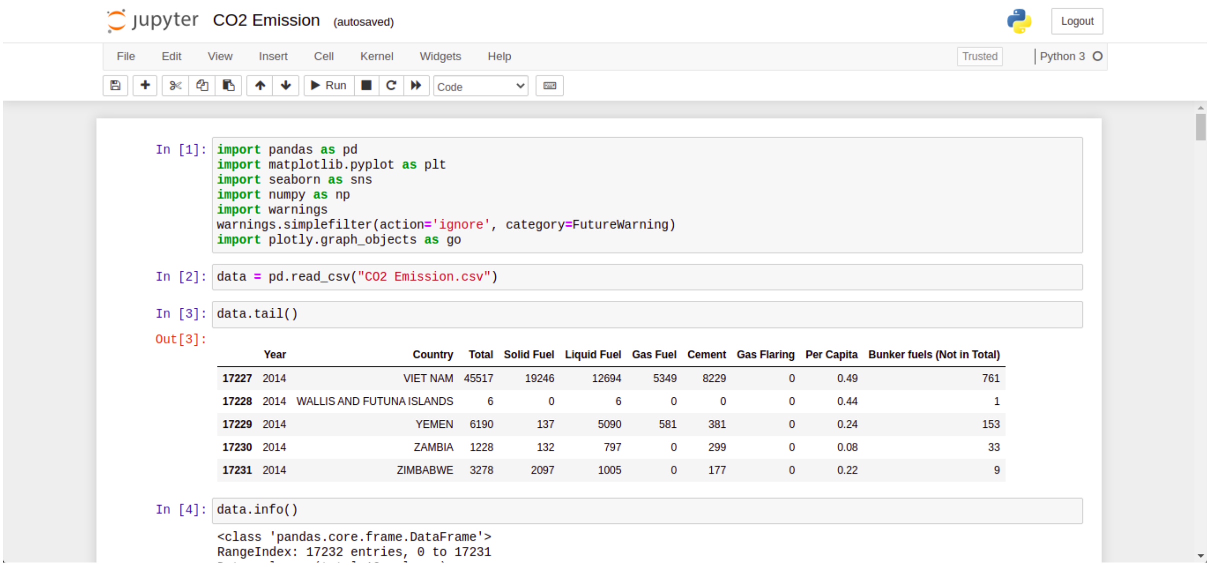
Here are the list of question whose analysis which be given below.

1. Data Analysis using statistical methods.
2. Data Analysis using conditional filtering when Country is Saudi Arabia.
3. Data Analysis using conditional filtering when Year is 2014.
4. Data Analysis by grouping the data on the basis of Country.
5. Data Analysis using sorting the data in ascending/descending order.
6. Data Analysis after sorting the data on the basis of Total, filtering ony Saudi Arabia and group by on their Year.
7. Data Analysis after sorting the data on the basis of Total, filtering ony 2014 data and group by on the Country.
8. Visualize data of 2014 on the basis of Country using a chart with proper headings and legends.
9. Visualize data of Saudi Arabia on the basis of Year using a chart with proper headings and legends.
10. Visualize data of USA on the basis of Year using a chart with proper headings and legends.

DATA ACQUISITION AND CLEANING

Code to read the data from Excel / CSV / HTML.

To read the dataset in xlsx format, we will load it into Pandas data frame but first let's import the pandas library and set an alias by typing **"import pandas as pd"**. After importing the library with the alias **"pd"**, let us load the .csv file using the following line of code:



```
In [1]: import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        import numpy as np
        import warnings
        warnings.simplefilter(action='ignore', category=FutureWarning)
        import plotly.graph_objects as go

In [2]: data = pd.read_csv("CO2 Emission.csv")

In [3]: data.tail()
Out[3]:
```

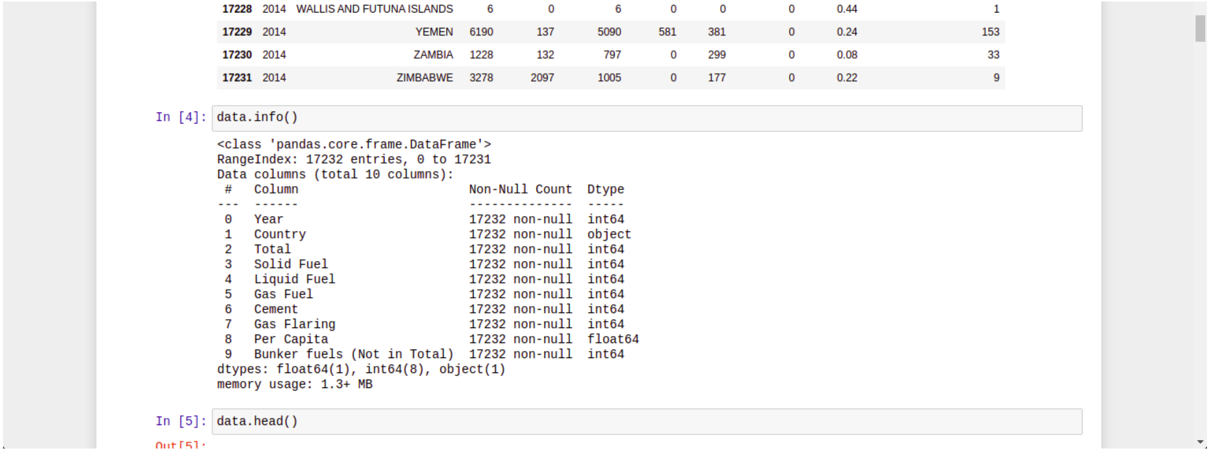
	Year	Country	Total	Solid Fuel	Liquid Fuel	Gas Fuel	Cement	Gas Flaring	Per Capita	Bunker fuels (Not in Total)
17227	2014	VIET NAM	45517	19246	12694	5349	8229	0	0.49	761
17228	2014	WALLIS AND FUTUNA ISLANDS	6	0	6	0	0	0	0.44	1
17229	2014	YEMEN	6190	137	5090	581	381	0	0.24	153
17230	2014	ZAMBIA	1228	132	797	0	299	0	0.08	33
17231	2014	ZIMBABWE	3278	2097	1005	0	177	0	0.22	9

```
In [4]: data.info()

        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 17232 entries, 0 to 17231
```

Here we have import our csv files and read through pandas library.

Here the xlsx file can be read through (Pandas library) and store in **data Dataframe**. The Dataframe can be shown through **.tail()**. The number of rows we want to show, that number we have to pass in head parentheses as an argument.

Now if we want to describe our dataframe for our better understanding to know the stats. and other parameter that our dataset should follow



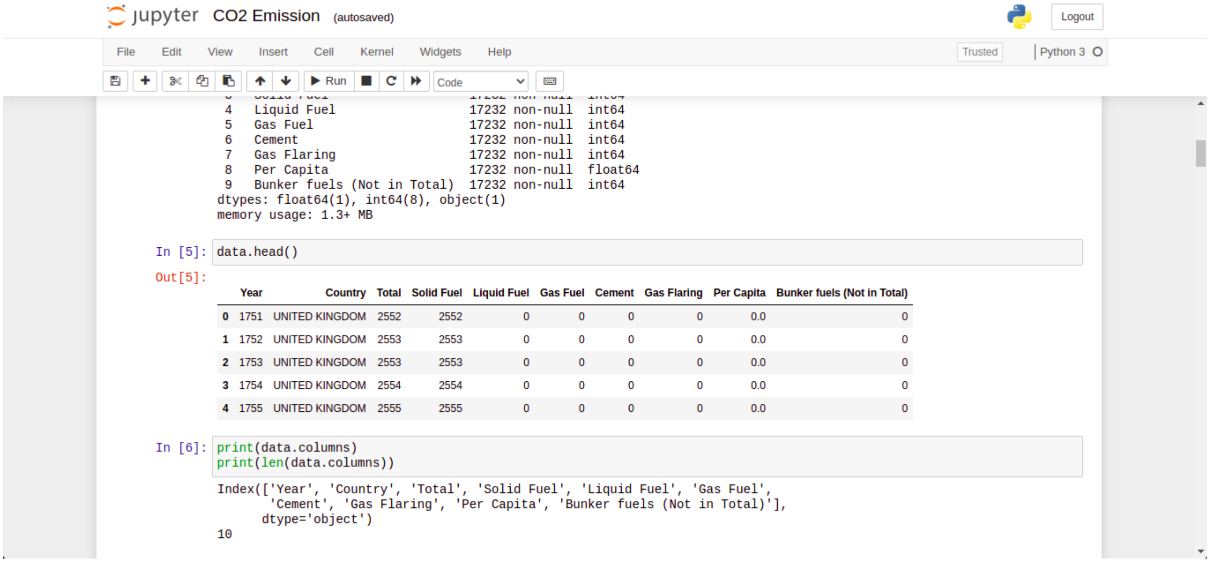
17228	2014	WALLIS AND FUTUNA ISLANDS	6	0	6	0	0	0	0.44	1
17229	2014	YEMEN	6190	137	5090	581	381	0	0.24	153
17230	2014	ZAMBIA	1228	132	797	0	299	0	0.08	33
17231	2014	ZIMBABWE	3278	2097	1005	0	177	0	0.22	9

In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17232 entries, 0 to 17231
Data columns (total 10 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Year                      17232 non-null  int64  
 1   Country                   17232 non-null  object 
 2   Total                     17232 non-null  int64  
 3   Solid Fuel                17232 non-null  int64  
 4   Liquid Fuel               17232 non-null  int64  
 5   Gas Fuel                  17232 non-null  int64  
 6   Cement                    17232 non-null  int64  
 7   Gas Flaring               17232 non-null  int64  
 8   Per Capita                17232 non-null  float64
 9   Bunker fuels (Not in Total)  17232 non-null  int64  
dtypes: float64(1), int64(8), object(1)
memory usage: 1.3+ MB
```

In [5]:
```
data.head()
```

Out[5]:



```
 4   Liquid Fuel               17232 non-null  int64  
 5   Gas Fuel                  17232 non-null  int64  
 6   Cement                    17232 non-null  int64  
 7   Gas Flaring               17232 non-null  int64  
 8   Per Capita                17232 non-null  float64
 9   Bunker fuels (Not in Total)  17232 non-null  int64  
dtypes: float64(1), int64(8), object(1)
memory usage: 1.3+ MB
```

In [5]:
```
data.head()
```

Out[5]:

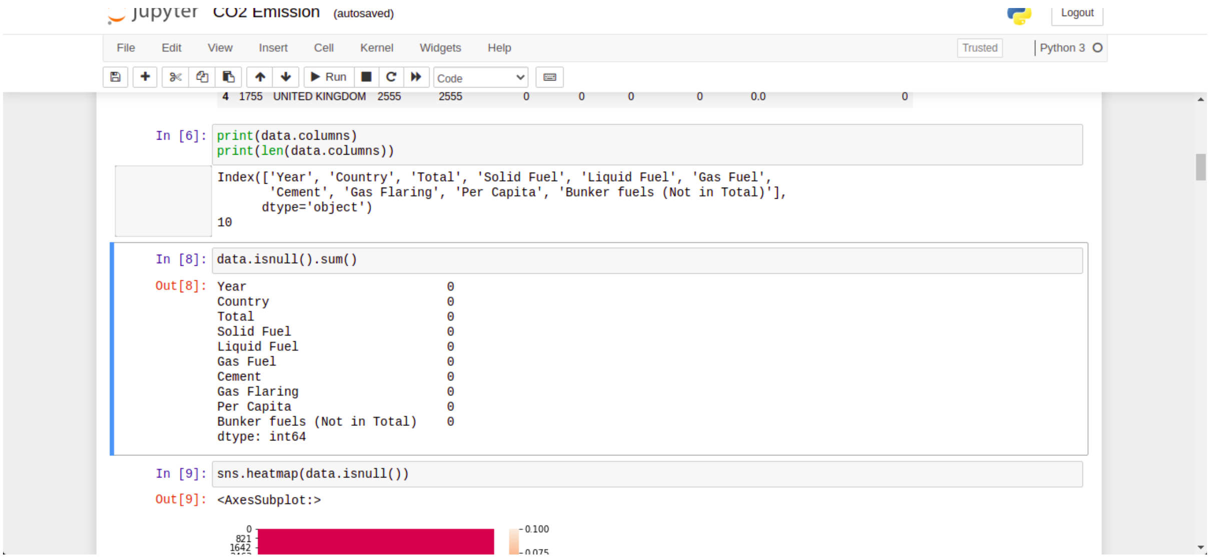
	Year	Country	Total	Solid Fuel	Liquid Fuel	Gas Fuel	Cement	Gas Flaring	Per Capita	Bunker fuels (Not in Total)
0	1751	UNITED KINGDOM	2552	2552	0	0	0	0	0.0	0
1	1752	UNITED KINGDOM	2553	2553	0	0	0	0	0.0	0
2	1753	UNITED KINGDOM	2553	2553	0	0	0	0	0.0	0
3	1754	UNITED KINGDOM	2554	2554	0	0	0	0	0.0	0
4	1755	UNITED KINGDOM	2555	2555	0	0	0	0	0.0	0

In [6]:
```
print(data.columns)
print(len(data.columns))
```

```
Index(['Year', 'Country', 'Total', 'Solid Fuel', 'Liquid Fuel', 'Gas Fuel',
       'Cement', 'Gas Flaring', 'Per Capita', 'Bunker fuels (Not in Total)'],
      dtype='object')
10
```

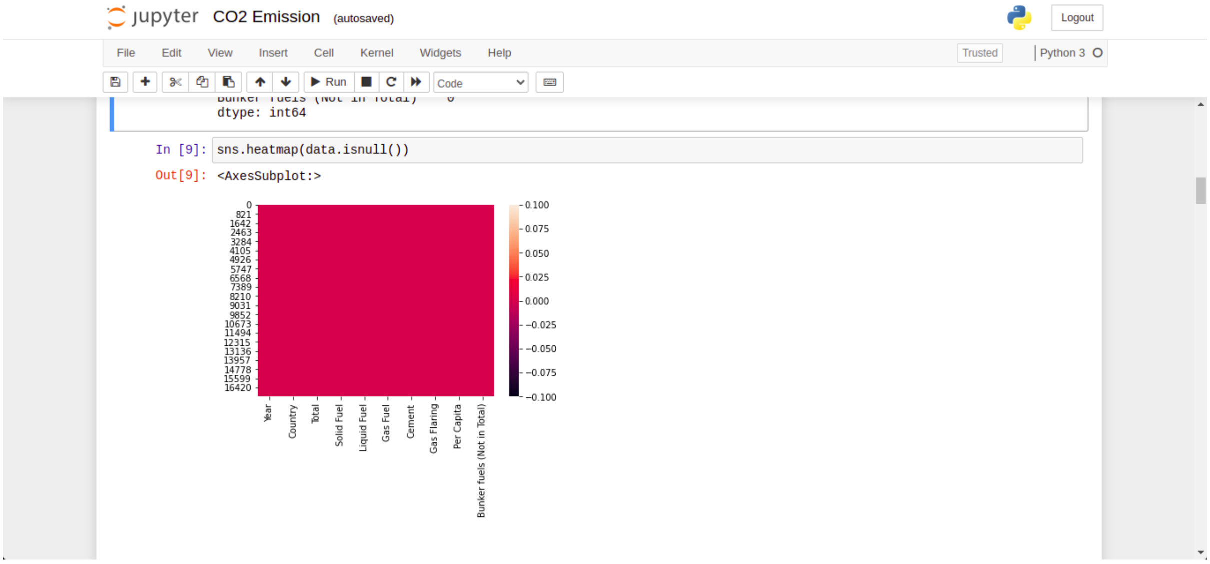
Clean the unnecessary data, by removing, replace the missing data and renaming the columns.

Dataset generally contains some null value, which is generally caused by misplacing some values. So its necessary to clean this mess from our dataset for better visualization



```
4   1755   UNITED KINGDOM   2555      2555       0        0       0        0       0.0                    0
```

```
In [6]: print(data.columns)
        print(len(data.columns))

        Index(['Year', 'Country', 'Total', 'Solid Fuel', 'Liquid Fuel', 'Gas Fuel',
               'Cement', 'Gas Flaring', 'Per Capita', 'Bunker fuels (Not in Total)'],
              dtype='object')
        10

In [8]: data.isnull().sum()

Out[8]: Year                           0
        Country                        0
        Total                          0
        Solid Fuel                     0
        Liquid Fuel                    0
        Gas Fuel                       0
        Cement                         0
        Gas Flaring                    0
        Per Capita                     0
        Bunker fuels (Not in Total)    0
        dtype: int64

In [9]: sns.heatmap(data.isnull())

Out[9]: <AxesSubplot:>
```



```
        dtype: int64

In [9]: sns.heatmap(data.isnull())

Out[9]: <AxesSubplot:>
```

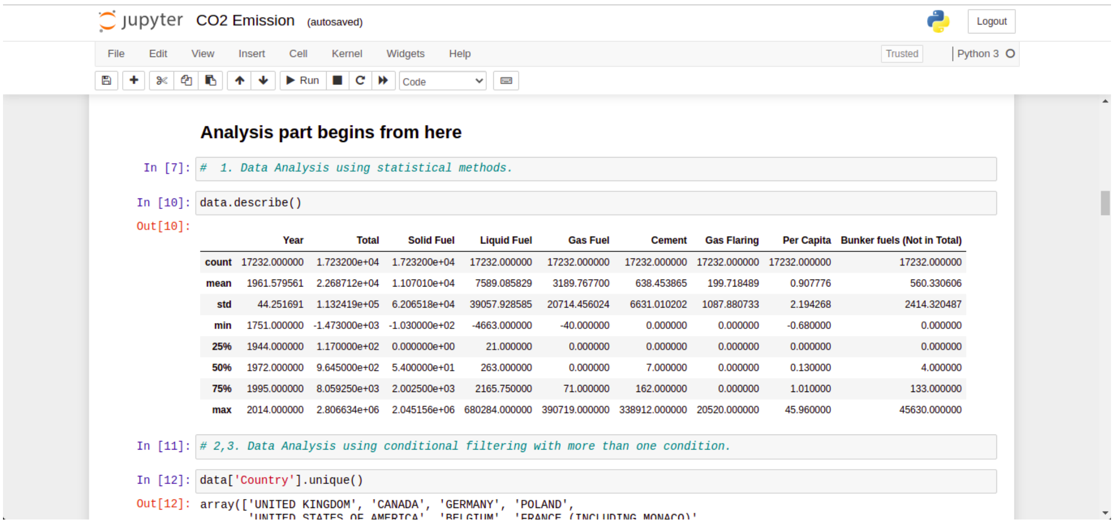
In the above diagram we see there is no columns of this dataset containing null value.

DATA AND EXPLORATORY ANALYSIS

Code and its output with Explanation

1. Data Analysis using statistical methods.

This data gives the estimated amount to various statistical methods that can be applied.



Analysis part begins from here

```
In [7]: #  1. Data Analysis using statistical methods.
```

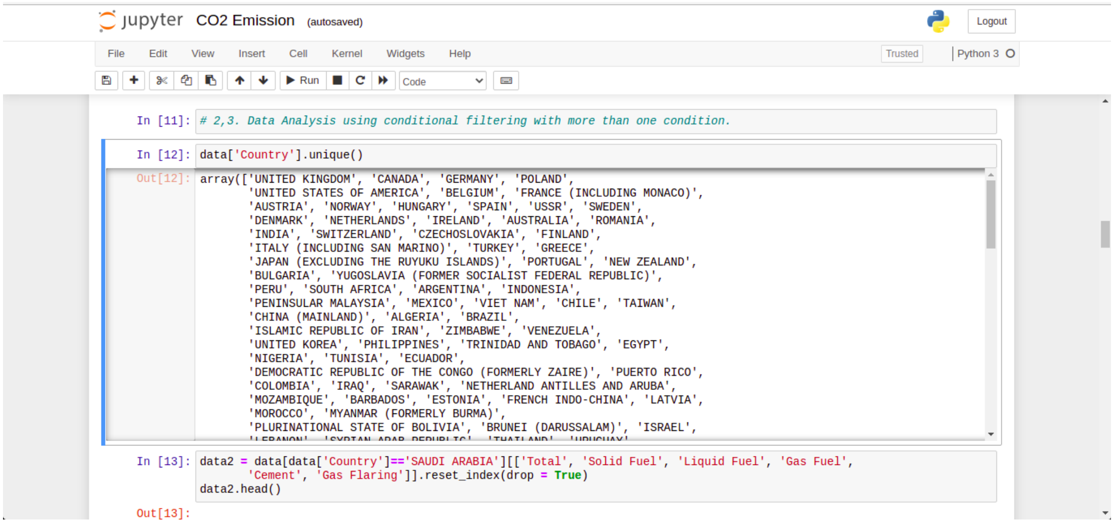
```
In [10]: data.describe()
```

Out[10]:

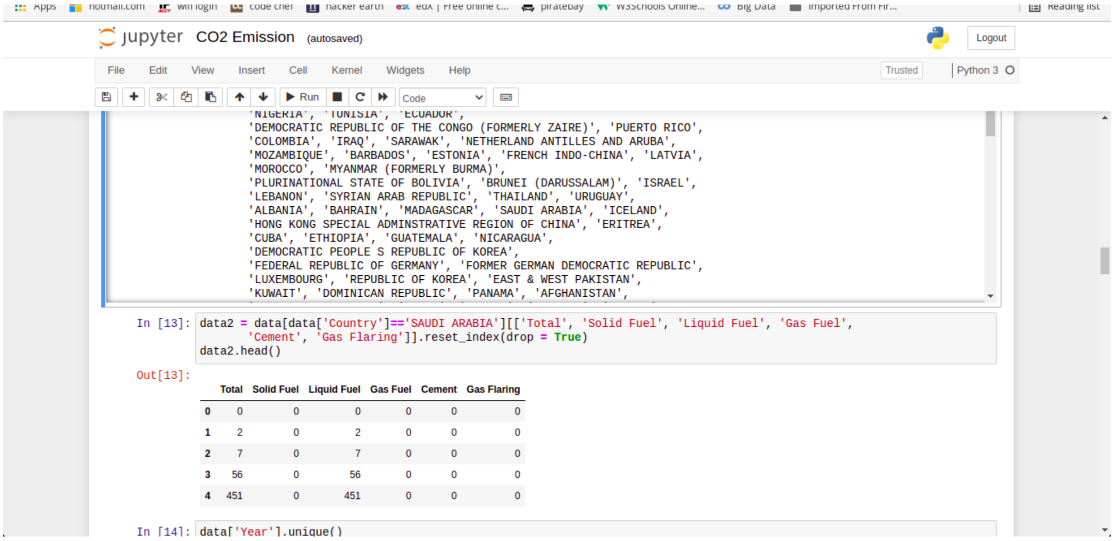
	Year	Total	Solid Fuel	Liquid Fuel	Gas Fuel	Cement	Gas Flaring	Per Capita	Bunker fuels (Not in Total)
count	17232.000000	1.723200e+04	1.723200e+04	17232.000000	17232.000000	17232.000000	17232.000000	17232.000000	17232.000000
mean	1961.579561	2.268712e+04	1.107010e+04	7589.085829	3189.767700	638.453865	199.718489	0.907776	560.330606
std	44.251691	1.132419e+05	6.206518e+04	39057.928585	20714.456024	6631.010202	1087.880733	2.194268	2414.320487
min	1751.000000	-1.473000e+03	-1.030000e+02	-4663.000000	-40.000000	0.000000	0.000000	-0.680000	0.000000
25%	1944.000000	1.170000e+02	0.000000e+00	21.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	1972.000000	9.645000e+02	5.400000e+01	263.000000	0.000000	7.000000	0.000000	0.130000	4.000000
75%	1995.000000	8.059250e+03	2.002500e+03	2165.750000	71.000000	162.000000	0.000000	1.010000	133.000000
max	2014.000000	2.806634e+06	2.045156e+06	680284.000000	390719.000000	338912.000000	20520.000000	45.960000	45630.000000

```
In [11]: # 2,3. Data Analysis using conditional filtering with more than one condition.

In [12]: data['Country'].unique()

Out[12]: array(['UNITED KINGDOM', 'CANADA', 'GERMANY', 'POLAND',
                'UNITED STATES OF AMERICA', 'BELGIUM', 'FRANCE (INCLUDING MONACO)',
```

2. Data Analysis using conditional filtering when Country is Saudi Arabia.



```
In [11]: # 2,3. Data Analysis using conditional filtering with more than one condition.

In [12]: data['Country'].unique()

Out[12]: array(['UNITED KINGDOM', 'CANADA', 'GERMANY', 'POLAND',
                'UNITED STATES OF AMERICA', 'BELGIUM', 'FRANCE (INCLUDING MONACO)',
                'AUSTRIA', 'NORWAY', 'HUNGARY', 'SPAIN', 'USSR', 'SWEDEN',
                'DENMARK', 'NETHERLANDS', 'IRELAND', 'AUSTRALIA', 'ROMANIA',
                'INDIA', 'SWITZERLAND', 'CZECHOSLOVAKIA', 'FINLAND',
                'ITALY (INCLUDING SAN MARINO)', 'TURKEY', 'GREECE',
                'JAPAN (EXCLUDING THE RUYUKU ISLANDS)', 'PORTUGAL', 'NEW ZEALAND',
                'BULGARIA', 'YUGOSLAVIA (FORMER SOCIALIST FEDERAL REPUBLIC)',
                'PERU', 'SOUTH AFRICA', 'ARGENTINA', 'INDONESIA',
                'PENINSULAR MALAYSIA', 'MEXICO', 'VIET NAM', 'CHILE', 'TAIWAN',
                'CHINA (MAINLAND)', 'ALGERIA', 'BRAZIL',
                'ISLAMIC REPUBLIC OF IRAN', 'ZIMBABWE', 'VENEZUELA',
                'UNITED KOREA', 'PHILIPPINES', 'TRINIDAD AND TOBAGO', 'EGYPT',
                'NIGERIA', 'TUNISIA', 'ECUADOR',
                'DEMOCRATIC REPUBLIC OF THE CONGO (FORMERLY ZAIRE)', 'PUERTO RICO',
                'COLOMBIA', 'IRAQ', 'SARAWAK', 'NETHERLAND ANTILLES AND ARUBA',
                'MOZAMBIQUE', 'BARBADOS', 'ESTONIA', 'FRENCH INDO-CHINA', 'LATVIA',
                'MOROCCO', 'MYANMAR (FORMERLY BURMA)',
                'PLURINATIONAL STATE OF BOLIVIA', 'BRUNEI (DARUSSALAM)', 'ISRAEL',
                'LEBANON', 'SYRIAN ARAB REPUBLIC', 'THAILAND', 'URUGUAY',

In [13]: data2 = data[data['Country']=='SAUDI ARABIA'][['Total', 'Solid Fuel', 'Liquid Fuel', 'Gas Fuel',
                'Cement', 'Gas Flaring']].reset_index(drop = True)
         data2.head()

Out[13]:
```



CO2 Emission (autosaved)

```
                 'NIGERIA', 'TUNISIA', 'ECUADOR',
                 'DEMOCRATIC REPUBLIC OF THE CONGO (FORMERLY ZAIRE)', 'PUERTO RICO',
                 'COLOMBIA', 'IRAQ', 'SARAWAK', 'NETHERLAND ANTILLES AND ARUBA',
                 'MOZAMBIQUE', 'BARBADOS', 'ESTONIA', 'FRENCH INDO-CHINA', 'LATVIA',
                 'MOROCCO', 'MYANMAR (FORMERLY BURMA)',
                 'PLURINATIONAL STATE OF BOLIVIA', 'BRUNEI (DARUSSALAM)', 'ISRAEL',
                 'LEBANON', 'SYRIAN ARAB REPUBLIC', 'THAILAND', 'URUGUAY',
                 'ALBANIA', 'BAHRAIN', 'MADAGASCAR', 'SAUDI ARABIA', 'ICELAND',
                 'HONG KONG SPECIAL ADMINSTRATIVE REGION OF CHINA', 'ERITREA',
                 'CUBA', 'ETHIOPIA', 'GUATEMALA', 'NICARAGUA',
                 'DEMOCRATIC PEOPLE S REPUBLIC OF KOREA',
                 'FEDERAL REPUBLIC OF GERMANY', 'FORMER GERMAN DEMOCRATIC REPUBLIC',
                 'LUXEMBOURG', 'REPUBLIC OF KOREA', 'EAST & WEST PAKISTAN',
                 'KUWAIT', 'DOMINICAN REPUBLIC', 'PANAMA', 'AFGHANISTAN',
```

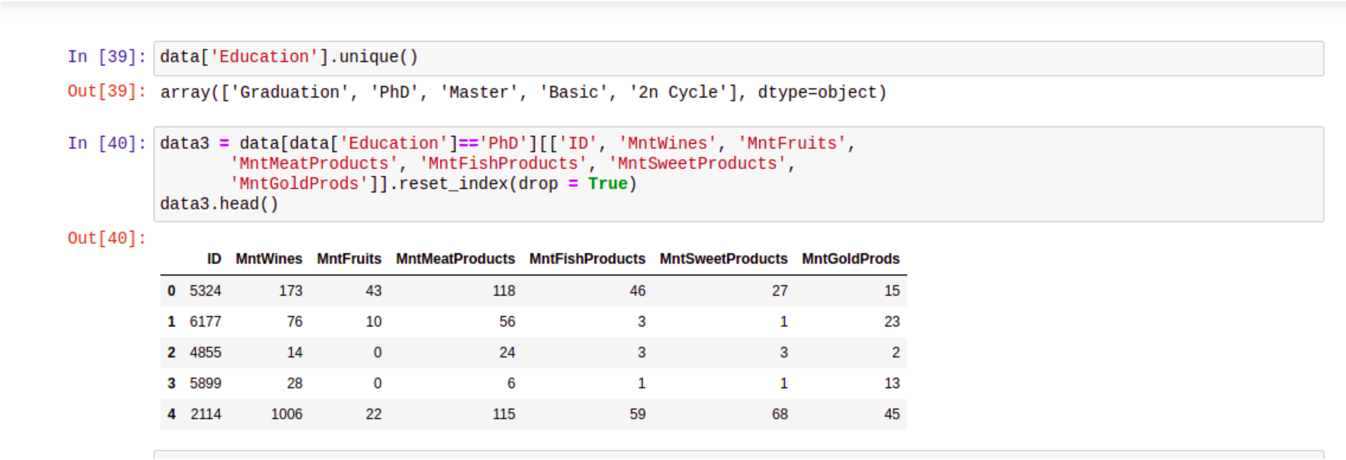
```
In [13]: data2 = data[data['Country']=='SAUDI ARABIA'][['Total', 'Solid Fuel', 'Liquid Fuel', 'Gas Fuel',
                'Cement', 'Gas Flaring']].reset_index(drop = True)
         data2.head()
```

Out[13]:

	Total	Solid Fuel	Liquid Fuel	Gas Fuel	Cement	Gas Flaring
0	0	0	0	0	0	0
1	2	0	2	0	0	0
2	7	0	7	0	0	0
3	56	0	56	0	0	0
4	451	0	451	0	0	0

```
In [14]: data['Year'].unique()
```

3. Data Analysis using conditional filtering when Year is 2014.

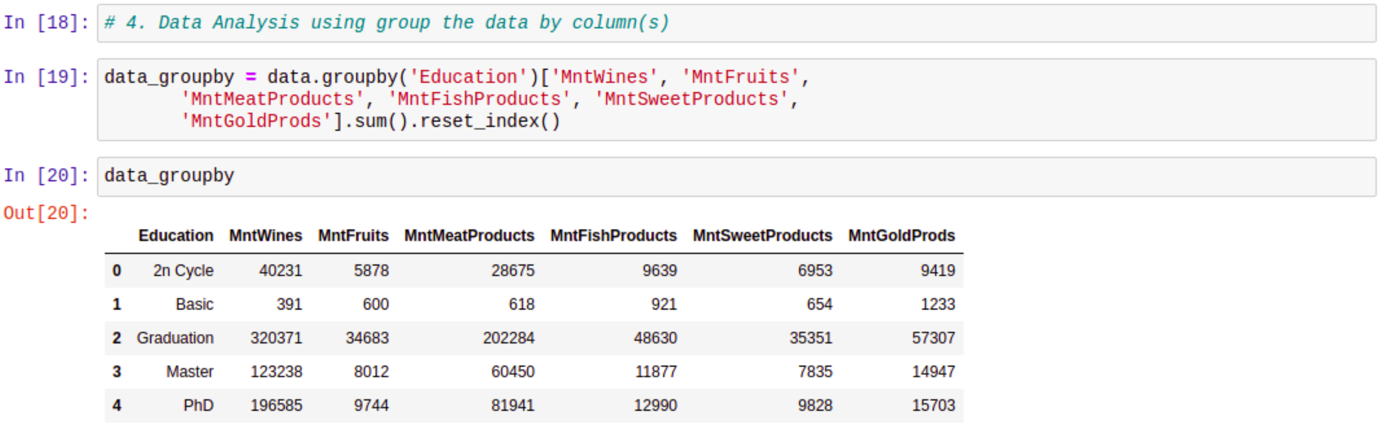


```
In [39]: data['Education'].unique()

Out[39]: array(['Graduation', 'PhD', 'Master', 'Basic', '2n Cycle'], dtype=object)

In [40]: data3 = data[data['Education']=='PhD'][['ID', 'MntWines', 'MntFruits',
                'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
                'MntGoldProds']].reset_index(drop = True)
         data3.head()
```

Out[40]:

	ID	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGoldProds
0	5324	173	43	118	46	27	15
1	6177	76	10	56	3	1	23
2	4855	14	0	24	3	3	2
3	5899	28	0	6	1	1	13
4	2114	1006	22	115	59	68	45

4. Data Analysis by grouping the data on the basis of Country.

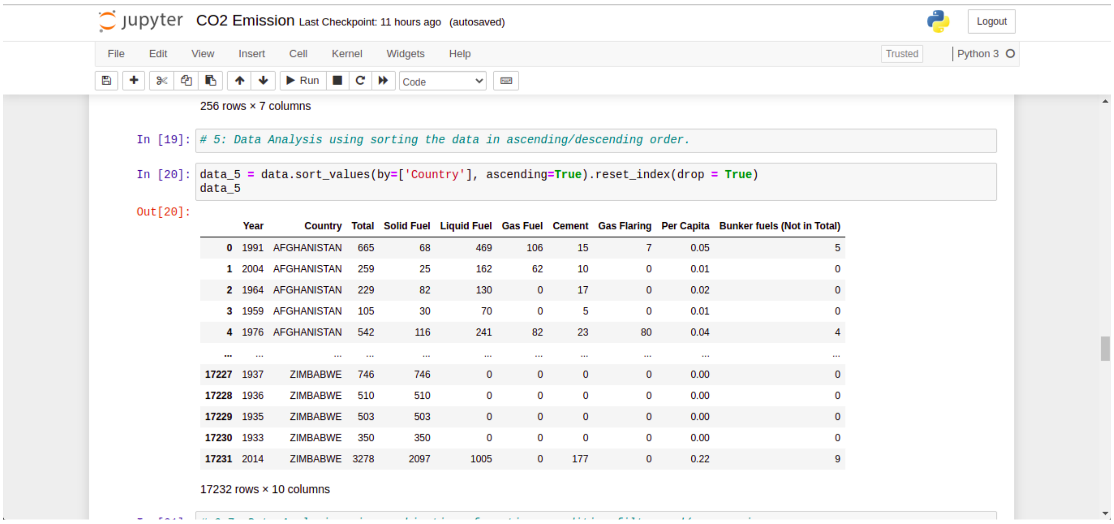


```
In [18]: # 4. Data Analysis using group the data by column(s)

In [19]: data_groupby = data.groupby('Education')['MntWines', 'MntFruits',
                'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
                'MntGoldProds'].sum().reset_index()

In [20]: data_groupby
```

Out[20]:

	Education	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGoldProds
0	2n Cycle	40231	5878	28675	9639	6953	9419
1	Basic	391	600	618	921	654	1233
2	Graduation	320371	34683	202284	48630	35351	57307
3	Master	123238	8012	60450	11877	7835	14947
4	PhD	196585	9744	81941	12990	9828	15703

5. **Data Analysis using sorting the data in ascending/descending order.**


jupyter CO2 Emission Last Checkpoint: 11 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

256 rows × 7 columns

```
In [19]: # 5: Data Analysis using sorting the data in ascending/descending order.
```

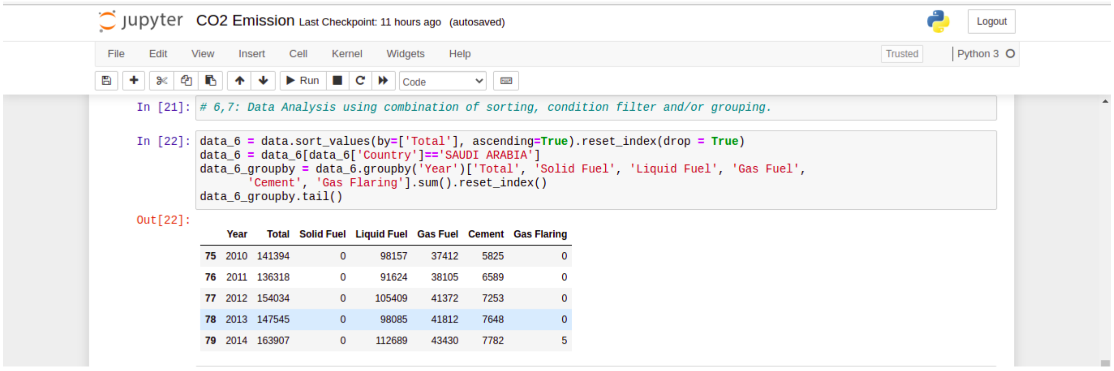
```
In [20]: data_5 = data.sort_values(by=['Country'], ascending=True).reset_index(drop = True)
         data_5
```

Out[20]:

	Year	Country	Total	Solid Fuel	Liquid Fuel	Gas Fuel	Cement	Gas Flaring	Per Capita	Bunker fuels (Not in Total)
0	1991	AFGHANISTAN	665	68	469	106	15	7	0.05	5
1	2004	AFGHANISTAN	259	25	162	62	10	0	0.01	0
2	1964	AFGHANISTAN	229	82	130	0	17	0	0.02	0
3	1959	AFGHANISTAN	105	30	70	0	5	0	0.01	0
4	1976	AFGHANISTAN	542	116	241	82	23	80	0.04	4
...	...	...	...	...	...	...	...	...	...	...
17227	1937	ZIMBABWE	746	746	0	0	0	0	0.00	0
17228	1936	ZIMBABWE	510	510	0	0	0	0	0.00	0
17229	1935	ZIMBABWE	503	503	0	0	0	0	0.00	0
17230	1933	ZIMBABWE	350	350	0	0	0	0	0.00	0
17231	2014	ZIMBABWE	3278	2097	1005	0	177	0	0.22	9

17232 rows × 10 columns

6. **Data Analysis after sorting the data on the basis of Total, filtering ony Saudi Arabia and group by on their Year.**


jupyter CO2 Emission Last Checkpoint: 11 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

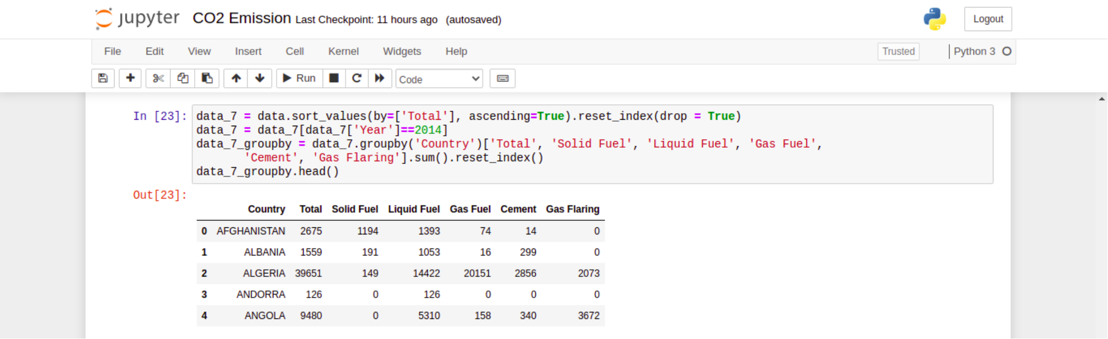
```
In [21]: # 6,7: Data Analysis using combination of sorting, condition filter and/or grouping.
```

```
In [22]: data_6 = data.sort_values(by=['Total'], ascending=True).reset_index(drop = True)
         data_6 = data_6[data_6['Country']=='SAUDI ARABIA']
         data_6_groupby = data_6.groupby('Year')['Total', 'Solid Fuel', 'Liquid Fuel', 'Gas Fuel',
                'Cement', 'Gas Flaring'].sum().reset_index()
         data_6_groupby.tail()
```

Out[22]:

	Year	Total	Solid Fuel	Liquid Fuel	Gas Fuel	Cement	Gas Flaring
75	2010	141394	0	98157	37412	5825	0
76	2011	136318	0	91624	38105	6589	0
77	2012	154034	0	105409	41372	7253	0
78	2013	147545	0	98085	41812	7648	0
79	2014	163907	0	112689	43430	7782	5

7. **Data Analysis after sorting the data on the basis of Total, filtering ony 2014 data and group by on the Country.**



```
In [23]: data_7 = data.sort_values(by=['Total'], ascending=True).reset_index(drop = True)
         data_7 = data_7[data_7['Year']==2014]
         data_7_groupby = data_7.groupby('Country')['Total', 'Solid Fuel', 'Liquid Fuel', 'Gas Fuel',
                'Cement', 'Gas Flaring'].sum().reset_index()
         data_7_groupby.head()
```

Out[23]:

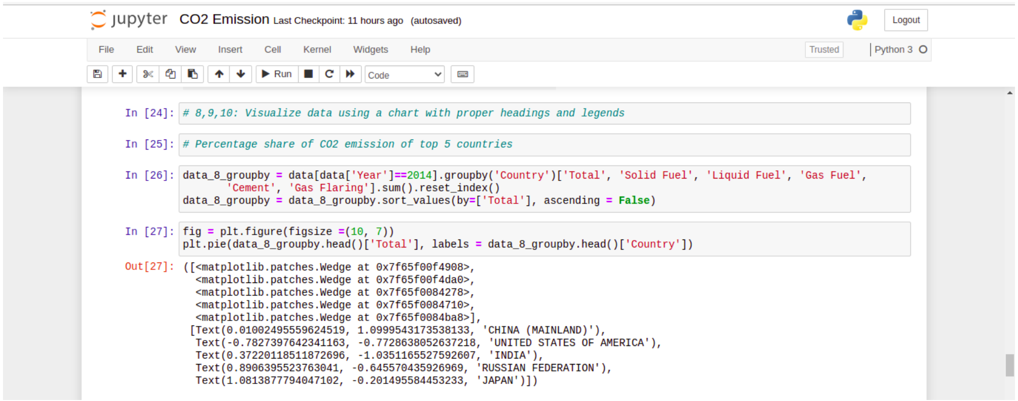
	Country	Total	Solid Fuel	Liquid Fuel	Gas Fuel	Cement	Gas Flaring
0	AFGHANISTAN	2675	1194	1393	74	14	0
1	ALBANIA	1559	191	1053	16	299	0
2	ALGERIA	39651	149	14422	20151	2856	2073
3	ANDORRA	126	0	126	0	0	0
4	ANGOLA	9480	0	5310	158	340	3672

DATA ANALYSIS – VISUALIZATION

Code and its output with vizualization

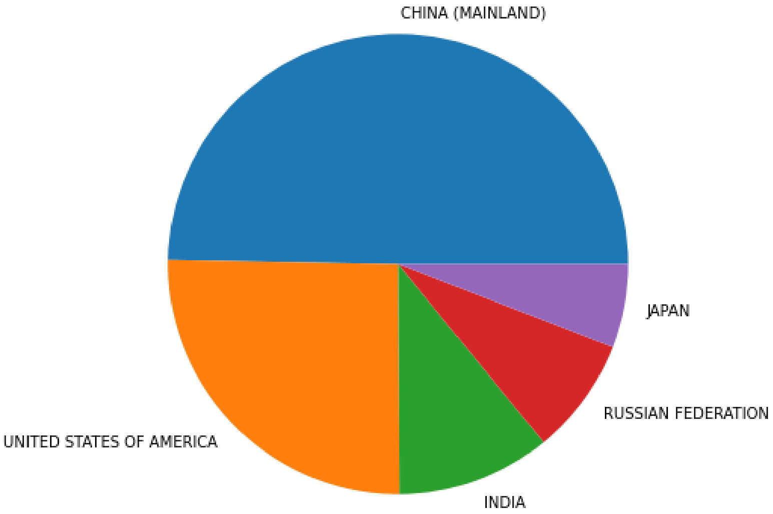
1. **Visualize data of 2014 on the basis of Country using a chart with proper headings and legends.**

Code:-



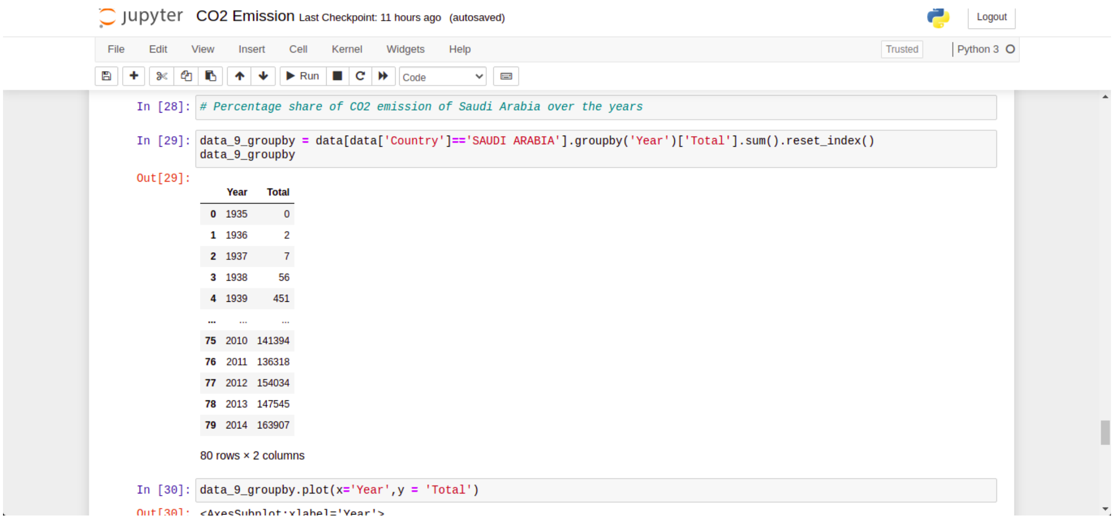
```
In [24]: # 8,9,10: Visualize data using a chart with proper headings and legends

In [25]: # Percentage share of CO2 emission of top 5 countries

In [26]: data_8_groupby = data[data['Year']==2014].groupby('Country')['Total', 'Solid Fuel', 'Liquid Fuel', 'Gas Fuel',
                'Cement', 'Gas Flaring'].sum().reset_index()
         data_8_groupby = data_8_groupby.sort_values(by=['Total'], ascending = False)

In [27]: fig = plt.figure(figsize =(10, 7))
         plt.pie(data_8_groupby.head()['Total'], labels = data_8_groupby.head()['Country'])

Out[27]: ([<matplotlib.patches.Wedge at 0x7f65f00f4908>,
           <matplotlib.patches.Wedge at 0x7f65f00f4da0>,
           <matplotlib.patches.Wedge at 0x7f65f0084278>,
           <matplotlib.patches.Wedge at 0x7f65f0084710>,
           <matplotlib.patches.Wedge at 0x7f65f0084ba8>],
          [Text(0.01002495559624519, 1.0999543173538133, 'CHINA (MAINLAND)'),
           Text(-0.7827397642341163, -0.7728638052637218, 'UNITED STATES OF AMERICA'),
           Text(0.37220118511872696, -1.0351165527592607, 'INDIA'),
           Text(0.8906395523763041, -0.645570435926969, 'RUSSIAN FEDERATION'),
           Text(1.0813877794047102, -0.201495584453233, 'JAPAN')])
```

Output:-



2. **Visualize data of Saudi Arabia on the basis of Year using a chart with proper headings and legends.**

Code:-



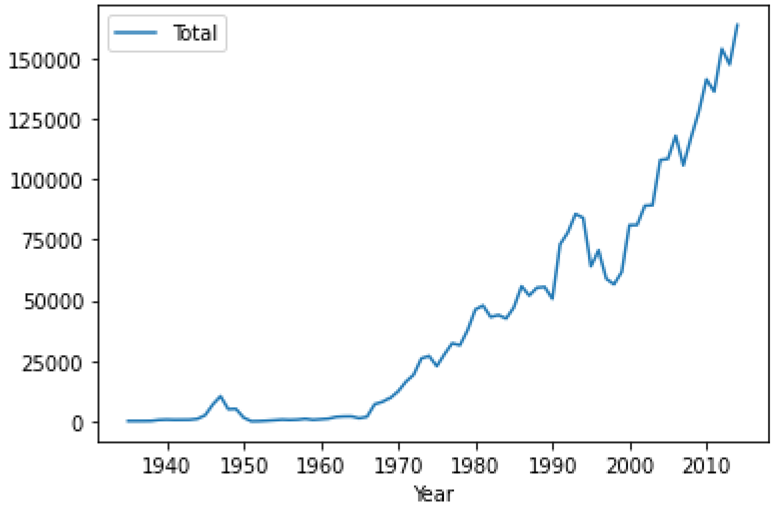
```
In [28]: # Percentage share of CO2 emission of Saudi Arabia over the years

In [29]: data_9_groupby = data[data['Country']=='SAUDI ARABIA'].groupby('Year')['Total'].sum().reset_index()
         data_9_groupby
Out[29]:
```

	Year	Total
0	1935	0
1	1936	2
2	1937	7
3	1938	56
4	1939	451
...	...	...
75	2010	141394
76	2011	136318
77	2012	154034
78	2013	147545
79	2014	163907

80 rows × 2 columns

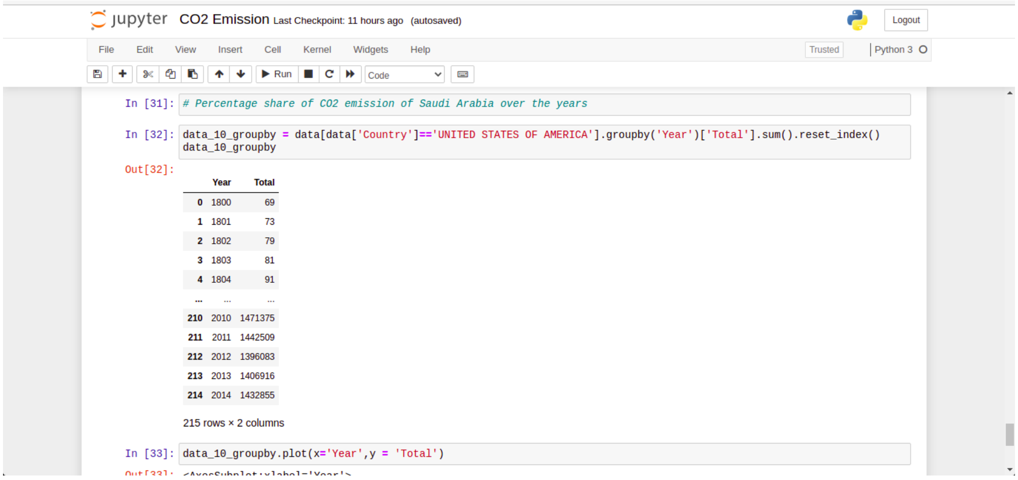
```
In [30]: data_9_groupby.plot(x='Year',y = 'Total')
Out[30]: <AxesSubplot:xlabel='Year'>
```

Output:-



3. **Visualize data of USA on the basis of Year using a chart with proper headings and legends.**

Code:-



```
In [31]: # Percentage share of CO2 emission of Saudi Arabia over the years

In [32]: data_10_groupby = data[data['Country']=='UNITED STATES OF AMERICA'].groupby('Year')['Total'].sum().reset_index()
         data_10_groupby
```

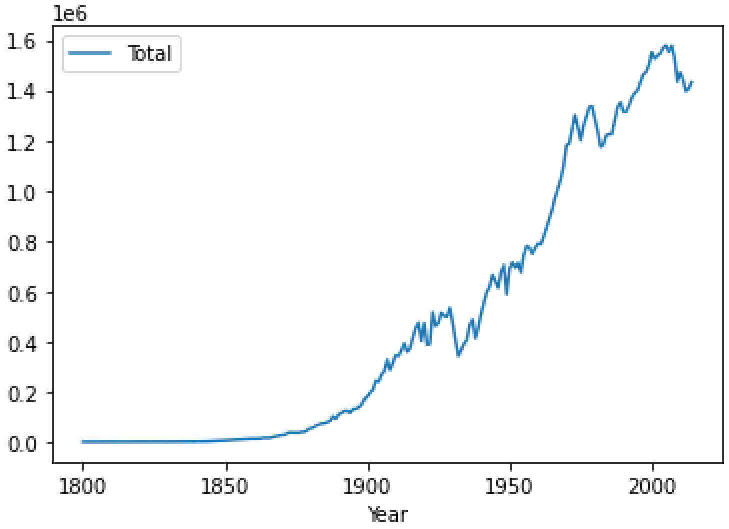
Out[32]:

	Year	Total
0	1800	69
1	1801	73
2	1802	79
3	1803	81
4	1804	91
...	...	...
210	2010	1471375
211	2011	1442509
212	2012	1396083
213	2013	1406916
214	2014	1432855

215 rows × 2 columns

```
In [33]: data_10_groupby.plot(x='Year',y = 'Total')
Out[33]: <AxesSubplot:xlabel='Year'>
```

Output:-



EXECUTIVE SUMMARY

CONCLUSION

In 2014, the highest dioxide (CO2) emitters were China, the us, the eu Union, India, the Russia, and Japan. These knowledge embrace carbon dioxide emissions from fuel combustion, additionally as cement producing and gas flaring. Together, these sources represent an outsized proportion of total international carbon dioxide emissions.

REFERENCES

- https://pandas.pydata.org/docs/
- https://www.kaggle.com/ggsri123/co2-emissions-from-fossil-fuels
- https://www.w3schools.com/python/pandas/pandas_plotting.asp