** DONE BY ANIKET MUKHERJEE**

import your modules which is required for this project

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

read the data from the given data set

In [2]:

```
C=pd.read_csv("master.csv")
```

print the data set

In [3]:

```
print(C)
```

```
          country  year     sex          age  suicides_no  population  \
0         Albania  1987    male  15-24 years           21      312900
1         Albania  1987    male  35-54 years           16      308000
2         Albania  1987  female  15-24 years           14      289700
3         Albania  1987    male    75+ years            1       21800
4         Albania  1987    male  25-34 years            9      274300
...           ...   ...     ...          ...          ...         ...
27815  Uzbekistan  2014  female  35-54 years          107     3620833
27816  Uzbekistan  2014  female    75+ years            9      348465
27817  Uzbekistan  2014    male   5-14 years           60     2762158
27818  Uzbekistan  2014  female   5-14 years           44     2631600
27819  Uzbekistan  2014  female  55-74 years           21     1438935

       suicides/100k pop    country-year  gdp_for_year ($)   \
0                   6.71     Albania1987      2,156,624,900
1                   5.19     Albania1987      2,156,624,900
2                   4.83     Albania1987      2,156,624,900
3                   4.59     Albania1987      2,156,624,900
4                   3.28     Albania1987      2,156,624,900
...                  ...             ...                ...
27815               2.96  Uzbekistan2014     63,067,077,179
27816               2.58  Uzbekistan2014     63,067,077,179
27817               2.17  Uzbekistan2014     63,067,077,179
27818               1.67  Uzbekistan2014     63,067,077,179
27819               1.46  Uzbekistan2014     63,067,077,179

       gdp_per_capita ($)       generation
0                     796     Generation X
1                     796           Silent
2                     796     Generation X
3                     796  G.I. Generation
4                     796          Boomers
...                   ...              ...
27815                2309     Generation X
27816                2309           Silent
27817                2309     Generation Z
27818                2309     Generation Z
27819                2309          Boomers

[27820 rows x 11 columns]
```

show the index of the data set

In [4]:

```
C.index
```

Out[4]:

```
RangeIndex(start=0, stop=27820, step=1)
```

show the size of the data set

In [5]:

```
C.size
```

Out[5]:

```
306020
```

show all the row number and column number of the data set

In [6]:

```
C.shape
```

Out[6]:

```
(27820, 11)
```

show all the column name

In [7]:

```
C.columns
```

Out[7]:

```
Index(['country', 'year', 'sex', 'age', 'suicides_no', 'population',
       'suicides/100k pop', 'country-year', ' gdp_for_year ($) ',
       'gdp_per_capita ($)', 'generation'],
      dtype='object')
```

show the total number of columns

In [8]:

```
len(C.columns)
```

Out[8]:

```
11
```

show the memory consumption for all the columns

In [9]:

```
C.memory_usage()
```

Out[9]:

```
Index                   128
country              222560
year                 222560
sex                  222560
age                  222560
suicides_no          222560
population           222560
suicides/100k pop    222560
country-year         222560
 gdp_for_year ($)    222560
gdp_per_capita ($)   222560
generation           222560
dtype: int64
```

show the dimensions of the data set

In [10]:

```
C.ndim
```

Out[10]:

```
2
```

show the column names,data types all the information at a glance

In [11]:

```
C.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27820 entries, 0 to 27819
Data columns (total 11 columns):
country               27820 non-null object
year                  27820 non-null int64
sex                   27820 non-null object
age                   27820 non-null object
suicides_no           27820 non-null int64
population            27820 non-null int64
suicides/100k pop     27820 non-null float64
country-year          27820 non-null object
 gdp_for_year ($)     27820 non-null object
gdp_per_capita ($)    27820 non-null int64
generation            27820 non-null object
dtypes: float64(1), int64(4), object(6)
memory usage: 2.3+ MB
```

show the first 8 rows of the data set

In [12]:

```
C.head(8)
```

Out[12]:

	country	year	sex	age	suicides_no	population	suicides/100k pop	country-year	gdp_for_y
0	Albania	1987	male	15-24 years	21	312900	6.71	Albania1987	2,156,624,9
1	Albania	1987	male	35-54 years	16	308000	5.19	Albania1987	2,156,624,9
2	Albania	1987	female	15-24 years	14	289700	4.83	Albania1987	2,156,624,9
3	Albania	1987	male	75+ years	1	21800	4.59	Albania1987	2,156,624,9
4	Albania	1987	male	25-34 years	9	274300	3.28	Albania1987	2,156,624,9
5	Albania	1987	female	75+ years	1	35600	2.81	Albania1987	2,156,624,9
6	Albania	1987	female	35-54 years	6	278800	2.15	Albania1987	2,156,624,9
7	Albania	1987	female	25-34 years	4	257200	1.56	Albania1987	2,156,624,9

show the last 8 rows of the data set

In [13]:

```
C.tail(8)
```

Out[13]:

	country	year	sex	age	suicides_no	population	suicides/100k pop	country-year
27812	Uzbekistan	2014	male	15-24 years	347	3126905	11.10	Uzbekistan2014
27813	Uzbekistan	2014	male	75+ years	17	224995	7.56	Uzbekistan2014
27814	Uzbekistan	2014	female	25-34 years	162	2735238	5.92	Uzbekistan2014
27815	Uzbekistan	2014	female	35-54 years	107	3620833	2.96	Uzbekistan2014
27816	Uzbekistan	2014	female	75+ years	9	348465	2.58	Uzbekistan2014
27817	Uzbekistan	2014	male	5-14 years	60	2762158	2.17	Uzbekistan2014
27818	Uzbekistan	2014	female	5-14 years	44	2631600	1.67	Uzbekistan2014
27819	Uzbekistan	2014	female	55-74 years	21	1438935	1.46	Uzbekistan2014

show the rows between 76-85

In [14]:

```
C.head(86).tail(10)
```

Out[14]:

	country	year	sex	age	suicides_no	population	suicides/100k pop	country-year	gdp_for_
76	Albania	1995	male	15-24 years	11	241200	4.56	Albania1995	2,424,499
77	Albania	1995	male	75+ years	1	25100	3.98	Albania1995	2,424,499
78	Albania	1995	male	35-54 years	14	375900	3.72	Albania1995	2,424,499
79	Albania	1995	female	25-34 years	7	264000	2.65	Albania1995	2,424,499
80	Albania	1995	female	35-54 years	8	356400	2.24	Albania1995	2,424,499
81	Albania	1995	male	5-14 years	6	376500	1.59	Albania1995	2,424,499
82	Albania	1995	female	55-74 years	2	180400	1.11	Albania1995	2,424,499
83	Albania	1995	female	5-14 years	2	348700	0.57	Albania1995	2,424,499
84	Albania	1996	male	75+ years	2	25400	7.87	Albania1996	3,314,898
85	Albania	1996	male	15-24 years	17	243600	6.98	Albania1996	3,314,898

show the all unique suicides

In [15]:

```
np.unique(C["suicides_no"])
```

Out[15]:

```
array([    0,     1,     2, ..., 21262, 21706, 22338], dtype=int64)
```

show the 1st suicide year

In [16]:

```
np.unique(C["year"])[0]
```

Out[16]:

```
1985
```

show the last suicide year

In [17]:

```
np.unique(C["year"])[-1]
```

Out[17]:

```
2016
```

show the details of the 1st row

In [18]:

```
C.loc[0,:]
```

Out[18]:

```
country                    Albania
year                          1987
sex                           male
age                     15-24 years
suicides_no                      21
population                   312900
suicides/100k pop              6.71
country-year            Albania1987
 gdp_for_year ($)     2,156,624,900
gdp_per_capita ($)              796
generation              Generation X
Name: 0, dtype: object
```

show the data types of the all columns

In [19]:

```
C.dtypes
```

Out[19]:

```
country                object
year                    int64
sex                    object
age                    object
suicides_no             int64
population              int64
suicides/100k pop     float64
country-year           object
 gdp_for_year ($)      object
gdp_per_capita ($)      int64
generation             object
dtype: object
```

show the data type of year column

In [20]:

```
C["year"].dtypes
```

Out[20]:

```
dtype('int64')
```

show the total nos of every category

In [21]:

```
C.dtypes.value_counts()
```

Out[21]:

```
object     6
int64      4
float64    1
dtype: int64
```

show the total number of objects

In [22]:

```
C.dtypes.value_counts()[0]
```

Out[22]:

```
6
```

select the data set that contains all the data types except object

In [23]:

```
C.select_dtypes(exclude=["object"])
```

Out[23]:

	year	suicides_no	population	suicides/100k pop	gdp_per_capita ($)
0	1987	21	312900	6.71	796
1	1987	16	308000	5.19	796
2	1987	14	289700	4.83	796
3	1987	1	21800	4.59	796
4	1987	9	274300	3.28	796
...	...	...	...	...	...
27815	2014	107	3620833	2.96	2309
27816	2014	9	348465	2.58	2309
27817	2014	60	2762158	2.17	2309
27818	2014	44	2631600	1.67	2309
27819	2014	21	1438935	1.46	2309

27820 rows × 5 columns

show the all including object

In [24]:

```
C.select_dtypes(include=["object"])
```

Out[24]:

	country	sex	age	country-year	gdp_for_year ($)	generation
0	Albania	male	15-24 years	Albania1987	2,156,624,900	Generation X
1	Albania	male	35-54 years	Albania1987	2,156,624,900	Silent
2	Albania	female	15-24 years	Albania1987	2,156,624,900	Generation X
3	Albania	male	75+ years	Albania1987	2,156,624,900	G.I. Generation
4	Albania	male	25-34 years	Albania1987	2,156,624,900	Boomers
...	...	...	...	...	...	...
27815	Uzbekistan	female	35-54 years	Uzbekistan2014	63,067,077,179	Generation X
27816	Uzbekistan	female	75+ years	Uzbekistan2014	63,067,077,179	Silent
27817	Uzbekistan	male	5-14 years	Uzbekistan2014	63,067,077,179	Generation Z
27818	Uzbekistan	female	5-14 years	Uzbekistan2014	63,067,077,179	Generation Z
27819	Uzbekistan	female	55-74 years	Uzbekistan2014	63,067,077,179	Boomers

27820 rows × 6 columns

print all the year

In [25]:

```
np.unique(C["year"])
```

Out[25]:

```
array([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=int64)
```

check every cell is null or not

In [26]:

```
C.isnull()
```

Out[26]:

	country	year	sex	age	suicides_no	population	suicides/100k pop	country-year	gdp_for_
0	False	False	False	False	False	False	False	False	F
1	False	False	False	False	False	False	False	False	F
2	False	False	False	False	False	False	False	False	F
3	False	False	False	False	False	False	False	False	F
4	False	False	False	False	False	False	False	False	F
...	...	...	...	...	...	...	...	...	
27815	False	False	False	False	False	False	False	False	F
27816	False	False	False	False	False	False	False	False	F
27817	False	False	False	False	False	False	False	False	F
27818	False	False	False	False	False	False	False	False	F
27819	False	False	False	False	False	False	False	False	F

27820 rows × 11 columns

count the sum of all the columns

In [27]:

```
C.isnull().sum()
```

Out[27]:

```
country              0
year                 0
sex                  0
age                  0
suicides_no          0
population           0
suicides/100k pop    0
country-year         0
 gdp_for_year ($)    0
gdp_per_capita ($)   0
generation           0
dtype: int64
```

check the data types of above 5 columns

In [28]:

```
C["year"].dtypes
```

Out[28]:

```
dtype('int64')
```

In [29]:

```
C["suicides_no"].dtypes
```

Out[29]:

```
dtype('int64')
```

In [30]:

```
C["country"].dtypes
```

Out[30]:

```
dtype('O')
```

In [31]:

```
C["sex"].dtypes
```

Out[31]:

```
dtype('O')
```

In [32]:

```
C["age"].dtypes
```

Out[32]:

```
dtype('O')
```

show the count as per category from the country column

In [33]:

```
C["country"].value_counts()
```

Out[33]:

```
Iceland                   382
Austria                   382
Mauritius                 382
Netherlands               382
Belgium                   372
                         ...
Bosnia and Herzegovina     24
Macau                      12
Cabo Verde                 12
Dominica                   12
Mongolia                   10
Name: country, Length: 101, dtype: int64
```

create a copy of the above data set

In [34]:

```
A=C.copy()
```

print the copied data set

In [35]:

```
print(A)
```

```
          country  year     sex          age  suicides_no  population  \
0         Albania  1987    male  15-24 years           21      312900
1         Albania  1987    male  35-54 years           16      308000
2         Albania  1987  female  15-24 years           14      289700
3         Albania  1987    male    75+ years            1       21800
4         Albania  1987    male  25-34 years            9      274300
...           ...   ...     ...          ...          ...         ...
27815  Uzbekistan  2014  female  35-54 years          107     3620833
27816  Uzbekistan  2014  female    75+ years            9      348465
27817  Uzbekistan  2014    male   5-14 years           60     2762158
27818  Uzbekistan  2014  female   5-14 years           44     2631600
27819  Uzbekistan  2014  female  55-74 years           21     1438935

       suicides/100k pop    country-year  gdp_for_year ($)   \
0                   6.71     Albania1987      2,156,624,900
1                   5.19     Albania1987      2,156,624,900
2                   4.83     Albania1987      2,156,624,900
3                   4.59     Albania1987      2,156,624,900
4                   3.28     Albania1987      2,156,624,900
...                  ...             ...                ...
27815               2.96  Uzbekistan2014     63,067,077,179
27816               2.58  Uzbekistan2014     63,067,077,179
27817               2.17  Uzbekistan2014     63,067,077,179
27818               1.67  Uzbekistan2014     63,067,077,179
27819               1.46  Uzbekistan2014     63,067,077,179

       gdp_per_capita ($)       generation
0                     796     Generation X
1                     796           Silent
2                     796     Generation X
3                     796  G.I. Generation
4                     796          Boomers
...                   ...              ...
27815                2309     Generation X
27816                2309           Silent
27817                2309     Generation Z
27818                2309     Generation Z
27819                2309          Boomers

[27820 rows x 11 columns]
```

create a one way table of age by dropping nan values

In [36]:

```
pd.crosstab(index=C["age"],columns="count",dropna=True)
```

Out[36]:

col_0	count
age	
15-24 years	4642
25-34 years	4642
35-54 years	4642
5-14 years	4610
55-74 years	4642
75+ years	4642

create a two way table of age and sex

In [37]:

```
pd.crosstab(index=C["age"],columns=C["sex"],dropna=True)
```

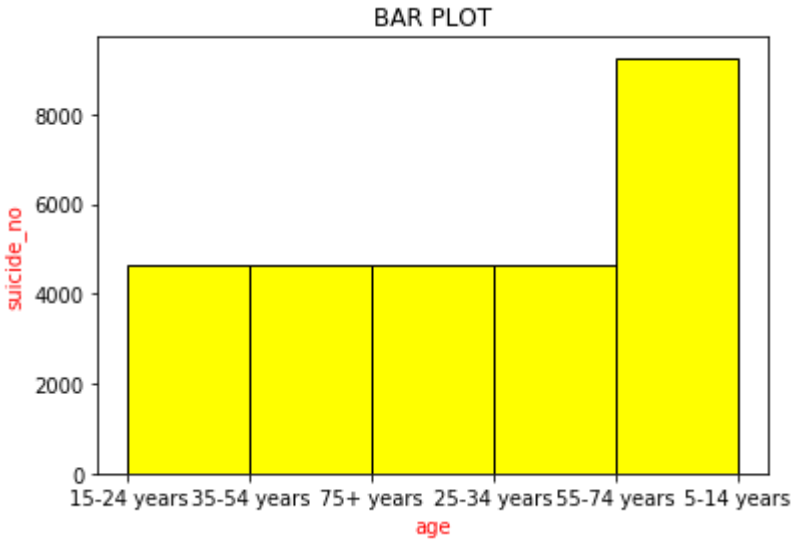
Out[37]:

sex	female	male
age		
15-24 years	2321	2321
25-34 years	2321	2321
35-54 years	2321	2321
5-14 years	2305	2305
55-74 years	2321	2321
75+ years	2321	2321

create a bar plot of age and suicide_no using matplotlib module

In [38]:

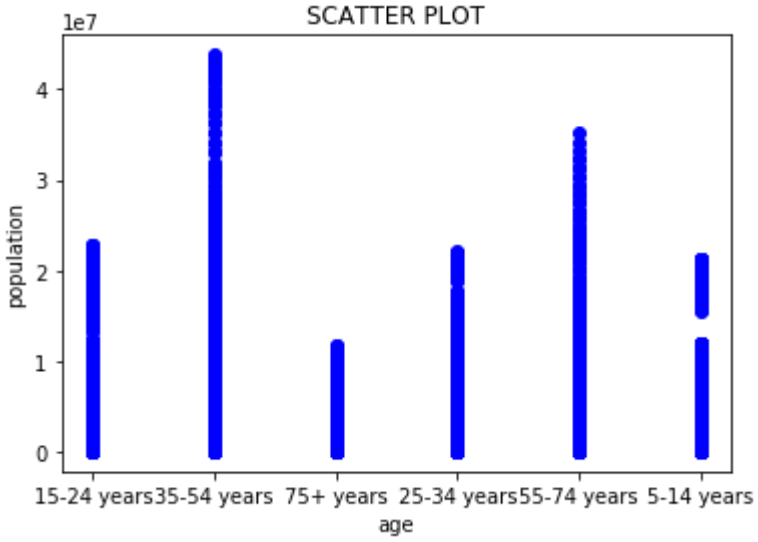
```
plt.hist(C["age"],bins=5,edgecolor="black",color="yellow")
plt.title("BAR PLOT")
plt.xlabel("age",c="red")
plt.ylabel("suicide_no",c="red")
plt.show()
```



scatter plot between population and age

In [39]:

```
plt.scatter(C["age"],C["population"],c="blue")
plt.title("SCATTER PLOT")
plt.xlabel("age")
plt.ylabel("population")
plt.show()
```



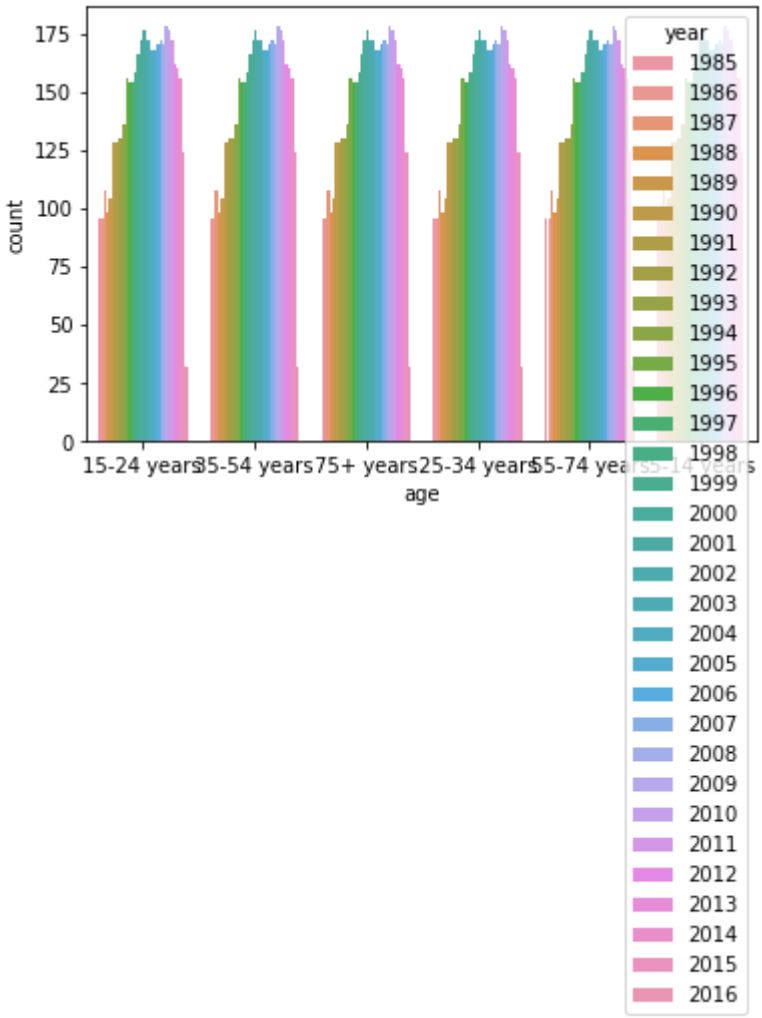
count plot of age based on year

In [40]:

```
sns.countplot(x="age",data=C,hue="year")
```

Out[40]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f6cc4bbcc8>
```



show the correlation matrix

In [41]:

```
C.corr()
```

Out[41]:

	year	suicides_no	population	suicides/100k pop	gdp_per_capita ($)
year	1.000000	-0.004546	0.008850	-0.039037	0.339134
suicides_no	-0.004546	1.000000	0.616162	0.306604	0.061330
population	0.008850	0.616162	1.000000	0.008285	0.081510
suicides/100k pop	-0.039037	0.306604	0.008285	1.000000	0.001785
gdp_per_capita ($)	0.339134	0.061330	0.081510	0.001785	1.000000

show the description of the data set

In [42]:

```
C.describe()
```

Out[42]:

	year	suicides_no	population	suicides/100k pop	gdp_per_capita ($)
count	27820.000000	27820.000000	2.782000e+04	27820.000000	27820.000000
mean	2001.258375	242.574407	1.844794e+06	12.816097	16866.464414
std	8.469055	902.047917	3.911779e+06	18.961511	18887.576472
min	1985.000000	0.000000	2.780000e+02	0.000000	251.000000
25%	1995.000000	3.000000	9.749850e+04	0.920000	3447.000000
50%	2002.000000	25.000000	4.301500e+05	5.990000	9372.000000
75%	2008.000000	131.000000	1.486143e+06	16.620000	24874.000000
max	2016.000000	22338.000000	4.380521e+07	224.970000	126352.000000

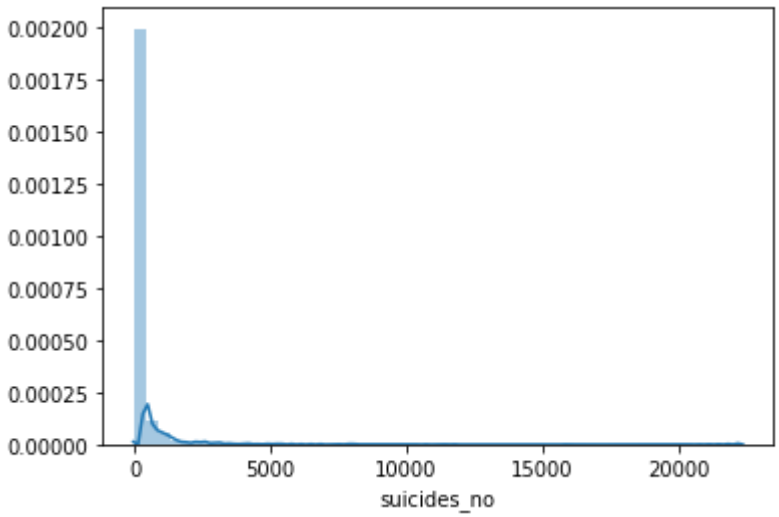
show the histogram of suicides_no with the kernel density estimate

In [43]:

```
sns.distplot(C["suicides_no"])
```

Out[43]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f6cc4c5ac8>
```



show the mean,median and mode of the data set

In [44]:

```
C.mean()
```

Out[44]:

```
year                 2.001258e+03
suicides_no          2.425744e+02
population           1.844794e+06
suicides/100k pop    1.281610e+01
gdp_per_capita ($)   1.686646e+04
dtype: float64
```

In [45]:

```
C.median()
```

Out[45]:

```
year                  2002.00
suicides_no             25.00
population          430150.00
suicides/100k pop        5.99
gdp_per_capita ($)    9372.00
dtype: float64
```

In [46]:

```
C.mode()
```

Out[46]:

	country	year	sex	age	suicides_no	population	suicides/100k pop	country-year
0	Austria	2009.0	female	15-24 years	0.0	24000.0	0.0	Albania1987
1	Iceland	NaN	male	25-34 years	NaN	NaN	NaN	Albania1988
2	Mauritius	NaN	NaN	35-54 years	NaN	NaN	NaN	Albania1989
3	Netherlands	NaN	NaN	55-74 years	NaN	NaN	NaN	Albania1992
4	NaN	NaN	NaN	75+ years	NaN	NaN	NaN	Albania1993
...	...	...	...	...	...	...	...	...
2300	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Uzbekistan2010
2301	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Uzbekistan2011
2302	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Uzbekistan2012
2303	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Uzbekistan2013
2304	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Uzbekistan2014

2305 rows × 11 columns

show the value which has the highest counts in the column population

In [47]:

```
A["population"].mode()
```

Out[47]:

```
0    24000
dtype: int64
```

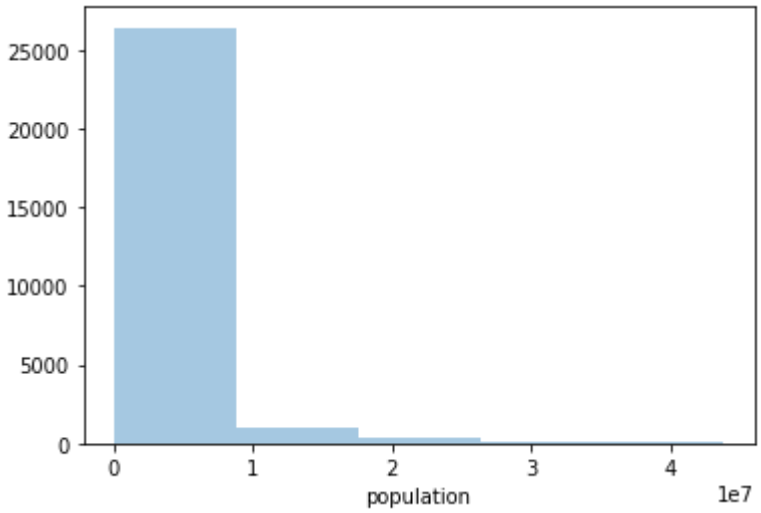
show the distplot of population where KDE is false

In [48]:

```
sns.distplot(C["population"],kde=False,bins=5)
```

Out[48]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f6cc8f7d48>
```



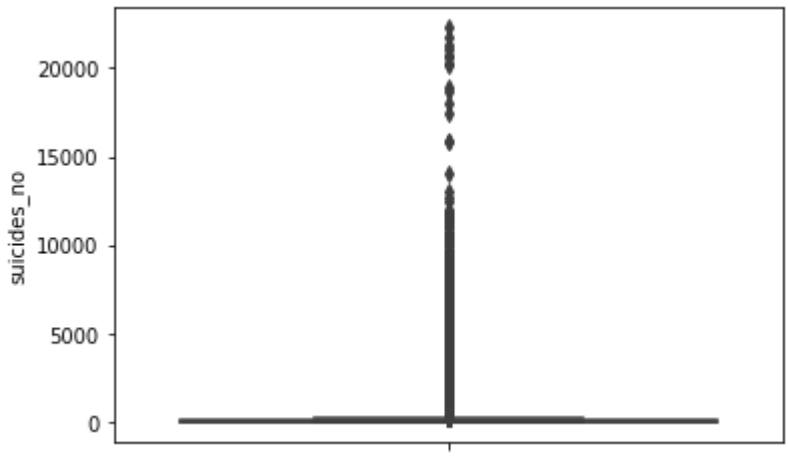
show the boxplot of suicides_nos

In [49]:

```
sns.boxplot(y=C["suicides_no"])
```

Out[49]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f6ccf134c8>
```



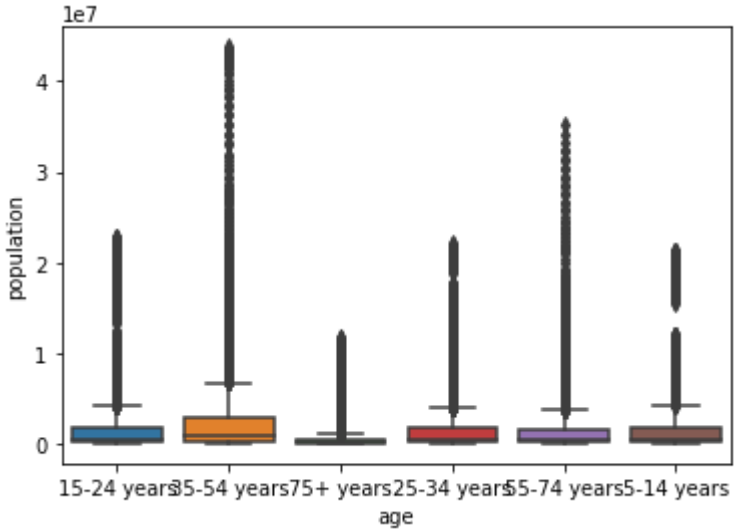
show the boxplot between age and population

In [50]:

```
sns.boxplot(x=C["age"],y=C["population"])
```

Out[50]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f6cce69bc8>
```



insert a new column in the given data set

In [51]:

```
C.insert(11,"current rating",0)
```

In [52]:

```
C
```

Out[52]:

	country	year	sex	age	suicides_no	population	suicides/100k pop	country-year
0	Albania	1987	male	15-24 years	21	312900	6.71	Albania1987
1	Albania	1987	male	35-54 years	16	308000	5.19	Albania1987
2	Albania	1987	female	15-24 years	14	289700	4.83	Albania1987
3	Albania	1987	male	75+ years	1	21800	4.59	Albania1987
4	Albania	1987	male	25-34 years	9	274300	3.28	Albania1987
...	...	...	...	...	...	...	...	...
27815	Uzbekistan	2014	female	35-54 years	107	3620833	2.96	Uzbekistan2014
27816	Uzbekistan	2014	female	75+ years	9	348465	2.58	Uzbekistan2014
27817	Uzbekistan	2014	male	5-14 years	60	2762158	2.17	Uzbekistan2014
27818	Uzbekistan	2014	female	5-14 years	44	2631600	1.67	Uzbekistan2014
27819	Uzbekistan	2014	female	55-74 years	21	1438935	1.46	Uzbekistan2014

27820 rows × 12 columns

show the marginal probability

In [59]:

```
pd.crosstab(index=A['suicides_no'],columns=A['population'],normalize=True,dropna=True,margi
```

Out[59]:

1	293	294	297	302	304	...	42957716	42992076	42997878	4300247
6	0.000036	0.000036	0.000036	0.000036	0.000036	...	0.000000	0.000000	0.000000	0.00000
0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.00000
0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.00000
0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.00000
0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.00000
..	...	...	...	...	...	...	...	...	...	.
0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.00000
0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.00000
0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.00000
0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.00000
6	0.000036	0.000036	0.000036	0.000036	0.000036	...	0.000036	0.000036	0.000036	0.00003

show the joint probability

In [54]:

```
pd.crosstab(index=A['suicides_no'],columns=A['population'],normalize=True,dropna=True)
```

Out[54]:

1	293	294	297	302	304	...	42932194	42957716	42992076	4299787
6	0.000036	0.000036	0.000036	0.000036	0.000036	...	0.0	0.0	0.0	0.
0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0	0.0	0.0	0.
0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0	0.0	0.0	0.
0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0	0.0	0.0	0.
0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0	0.0	0.0	0.
..	...	...	...	...	...	...	...	...	...	.
0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0	0.0	0.0	0.
0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0	0.0	0.0	0.
0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0	0.0	0.0	0.
0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0	0.0	0.0	0.
0	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0	0.0	0.0	0.

THANK YOU