8) Use Crime dataset

I) find the aggregations like all moments of business decisions for all columns,value counts.

II) do the plottings like plottings like histogram, boxplot, scatterplot, barplot, piechart,dot chart.

In [23]:

```
import pandas as pd
```

In [24]:

```
crime_df = pd.read_csv("crime_data (1).csv")
```

In [25]:

```
crime_df.count()
```

Out[25]:

```
Unnamed: 0    50
Murder        50
Assault       50
UrbanPop      50
Rape          50
dtype: int64
```

In [26]:

```
crime_df.head()
```

Out[26]:

	Unnamed: 0	Murder	Assault	UrbanPop	Rape
0	Alabama	13.2	236	58	21.2
1	Alaska	10.0	263	48	44.5
2	Arizona	8.1	294	80	31.0
3	Arkansas	8.8	190	50	19.5
4	California	9.0	276	91	40.6

In [27]:

```
crime_df.isnull().sum() #checking null values in whole data frame
```

Out[27]:

```
Unnamed: 0    0
Murder        0
Assault       0
UrbanPop      0
Rape          0
dtype: int64
```

In [28]:

```
crime_df = crime_df.rename(columns = {"Unnamed: 0":"City"}) #renaming first column.
```

In [29]:

```
crime_df.head()
```

Out[29]:

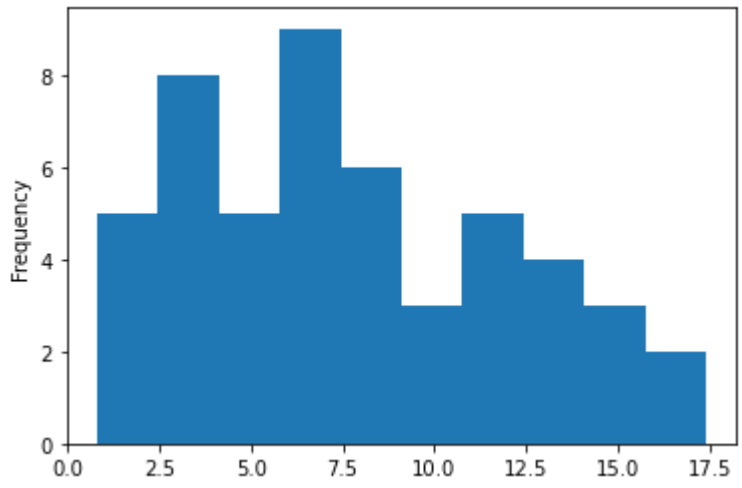
	City	Murder	Assault	UrbanPop	Rape
0	Alabama	13.2	236	58	21.2
1	Alaska	10.0	263	48	44.5
2	Arizona	8.1	294	80	31.0
3	Arkansas	8.8	190	50	19.5
4	California	9.0	276	91	40.6

In [30]:

```
#Histogram
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
crime_df1 = pd.read_csv('crime_data (1).csv')
crime_df1.head()
crime_df1['Murder'].plot(kind='hist')
```

Out[30]:

```
<AxesSubplot:ylabel='Frequency'>
```



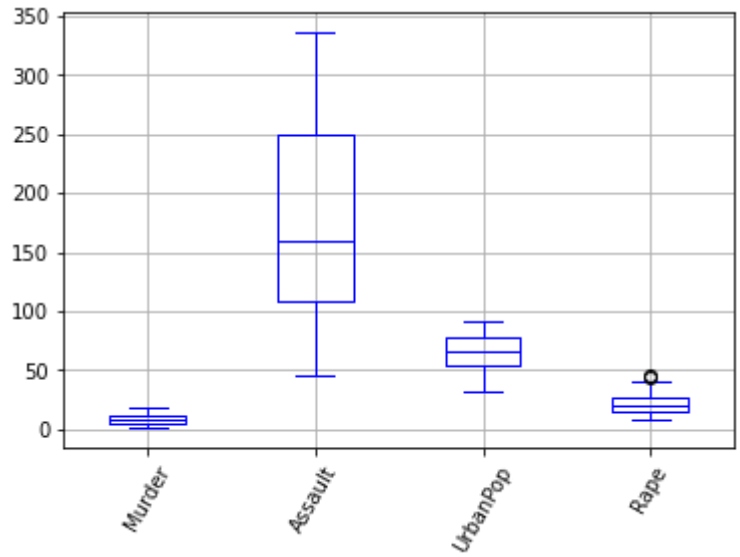
In [21]:

```
#Boxplot

crime_df1.boxplot(grid='false', color='blue', fontsize=10, rot=60)
```

Out[21]:

<AxesSubplot:>

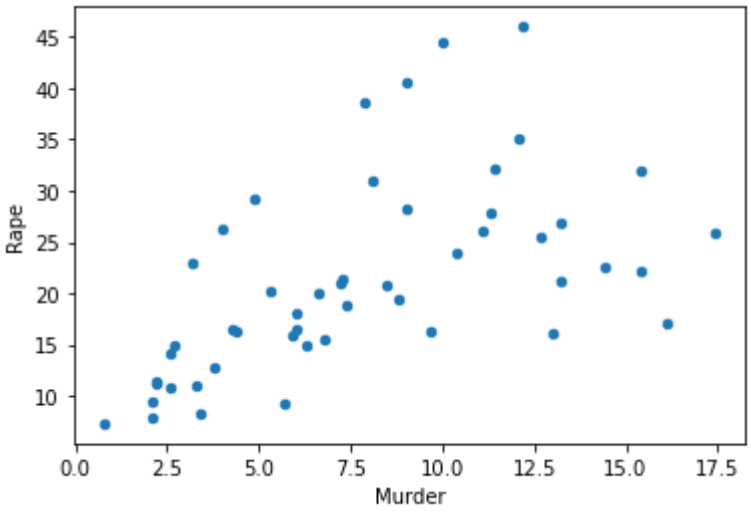


In [16]:

```
#scatterplot
crime_df1.plot(kind='scatter', x='Murder', y='Rape')
```

Out[16]:

```
<AxesSubplot:xlabel='Murder', ylabel='Rape'>
```



In [22]:

```
#piechart
crime_df1.plot.pie(y='Murder', figsize=(5, 4))
```

Out[22]:

```
<AxesSubplot:ylabel='Murder'>
```

9) use mtcars dataset.

A) delete/ drop rows-10 to 15 of all columns B)drop the VOL column

In [31]:

```
import pandas as pd
import numpy as np
data = pd.read_csv("mtcars (1).csv")
data.head()
```

Out[31]:

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

In [32]:

```
#delete/ drop rows-10 to 15 of all columns.
data = data.drop(data.index[range(12)])
data.head()
```

Out[32]:

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
12	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
13	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
14	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
15	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
16	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4

In [33]:

```
#drop the hp column
data = data.drop(columns=data.columns[3])
data.head()
```

Out[33]:

	mpg	cyl	disp	drat	wt	qsec	vs	am	gear	carb
12	17.3	8	275.8	3.07	3.730	17.60	0	0	3	3
13	15.2	8	275.8	3.07	3.780	18.00	0	0	3	3
14	10.4	8	472.0	2.93	5.250	17.98	0	0	3	4
15	10.4	8	460.0	3.00	5.424	17.82	0	0	3	4
16	14.7	8	440.0	3.23	5.345	17.42	0	0	3	4

In [34]:

```
#write the forloop to get value_counts of all cloumns
for col in data:
    print('-' * 40 + col + '-' * 40 , end=' - ')
display(data[col].value_counts().head(10))
```

```
----------------------------------------mpg---------------------------------
------- - ----------------------------------------cyl----------------------
---------------- - ----------------------------------------disp------------
---------------------------- - ----------------------------------------drat-
------------------------------------ - ---------------------------------
------wt---------------------------------------- - ------------------------
--------------qsec---------------------------------------- - --------------
-------------------------vs---------------------------------------- - -----
-----------------------------------am-----------------------------------------
- - ----------------------------------------gear-----------------------------
------------ - ----------------------------------------carb-----------------
---------------------- -

2    7
4    5
1    4
3    2
6    1
8    1
Name: carb, dtype: int64
```

10) Use Bank Dataset.

A) change all the categorical columns into numerical by creating Dummies and using label encoder.
B) rename all the column names in DF
C) Rename only one specific column in DF
D) After doing all the changes in bank dataset. save the file in your directory in Csv Format.(hint: by using .to_csv)

In [38]:

```
#Use Bank Dataset from LMS
from sklearn.preprocessing import LabelEncoder
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
```

In [45]:

```
df = pd.read_csv("bank-full (1).csv")
df.head()
```

Out[45]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	may
1	44	technician	single	secondary	no	29	yes	no	unknown	5	may
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may
4	33	unknown	single	unknown	no	1	no	no	unknown	5	may

In [46]:

```
#change all the categorical columns into numerical by creating Dummies and using label enco
new_DataM1 = df
data = pd.get_dummies(new_DataM1["marital"])
```

In [47]:

```
data.head(5)
```

Out[47]:

	divorced	married	single
0	0	1	0
1	0	0	1
2	0	1	0
3	0	1	0
4	0	0	1

In [48]:

```
df=df.iloc[:,:].values
```

In [49]:

```
df
```

Out[49]:

```
array([[58, 'management', 'married', ..., 0, 'unknown', 'no'],
       [44, 'technician', 'single', ..., 0, 'unknown', 'no'],
       [33, 'entrepreneur', 'married', ..., 0, 'unknown', 'no'],
       ...,
       [72, 'retired', 'married', ..., 3, 'success', 'yes'],
       [57, 'blue-collar', 'married', ..., 0, 'unknown', 'no'],
       [37, 'entrepreneur', 'married', ..., 11, 'other', 'no']],
      dtype=object)
```

In [50]:

```
marital =LabelEncoder()
```

In [51]:

```
df[:,2]=marital.fit_transform(df[:,2])
```

In [52]:

```
df
```

Out[52]:

```
array([[58, 'management', 1, ..., 0, 'unknown', 'no'],
       [44, 'technician', 2, ..., 0, 'unknown', 'no'],
       [33, 'entrepreneur', 1, ..., 0, 'unknown', 'no'],
       ...,
       [72, 'retired', 1, ..., 3, 'success', 'yes'],
       [57, 'blue-collar', 1, ..., 0, 'unknown', 'no'],
       [37, 'entrepreneur', 1, ..., 11, 'other', 'no']], dtype=object)
```

In [53]:

```
coltran=ColumnTransformer(transformers=[('encode',OneHotEncoder(),[2])],remainder='passthro
```

In [54]:

```
y=coltran.fit_transform(df)
```

In [55]:

```
y
```

Out[55]:

```
array([[0.0, 1.0, 0.0, ..., 0, 'unknown', 'no'],
       [0.0, 0.0, 1.0, ..., 0, 'unknown', 'no'],
       [0.0, 1.0, 0.0, ..., 0, 'unknown', 'no'],
       ...,
       [0.0, 1.0, 0.0, ..., 3, 'success', 'yes'],
       [0.0, 1.0, 0.0, ..., 0, 'unknown', 'no'],
       [0.0, 1.0, 0.0, ..., 11, 'other', 'no']], dtype=object)
```

In [56]:

```
y=pd.DataFrame(y)
y
```

Out[56]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0.0	1.0	0.0	58	management	tertiary	no	2143	yes	no	unknown	5	may	261
1	0.0	0.0	1.0	44	technician	secondary	no	29	yes	no	unknown	5	may	151
2	0.0	1.0	0.0	33	entrepreneur	secondary	no	2	yes	yes	unknown	5	may	76
3	0.0	1.0	0.0	47	blue-collar	unknown	no	1506	yes	no	unknown	5	may	92
4	0.0	0.0	1.0	33	unknown	unknown	no	1	no	no	unknown	5	may	198
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
45206	0.0	1.0	0.0	51	technician	tertiary	no	825	no	no	cellular	17	nov	977
45207	1.0	0.0	0.0	71	retired	primary	no	1729	no	no	cellular	17	nov	456
45208	0.0	1.0	0.0	72	retired	secondary	no	5715	no	no	cellular	17	nov	1127
45209	0.0	1.0	0.0	57	blue-collar	secondary	no	668	no	no	telephone	17	nov	508
45210	0.0	1.0	0.0	37	entrepreneur	secondary	no	2971	no	no	cellular	17	nov	361

45211 rows × 19 columns