```
# Read the data from the device
```

In [3]:

```python
import pandas as pd
import numpy as np
df=pd.read_csv("E://Data Science//salesdataset.csv")
df
```

Out[3]:

```
        TV  Radio  Newspaper  Sales
  0  230.1   37.8       69.2   22.1
  1   44.5   39.3       45.1   10.4
  2   17.2   45.9       69.3   12.0
  3  151.5   41.3       58.5   16.5
  4  180.8   10.8       58.4   17.9
...    ...    ...        ...    ...
195   38.2    3.7       13.8    7.6
196   94.2    4.9        8.1   14.0
197  177.0    9.3        6.4   14.8
198  283.6   42.0       66.2   25.5
199  232.1    8.6        8.7   18.4

200 rows × 4 columns
```

Print the entire data from dataset

```
In [4]: df.to_string()
```

```
Out[4]: '        TV  Radio  Newspaper  Sales\n0    230.1   37.8       69.2   22.1
         \n1     44.5   39.3       45.1   10.4\n2     17.2   45.9       69.3   12.0
         \n3    151.5   41.3       58.5   16.5\n4    180.8   10.8       58.4   17.9
         \n5      8.7   48.9       75.0    7.2\n6     57.5   32.8       23.5   11.8
         \n7    120.2   19.6       11.6   13.2\n8      8.6    2.1        1.0    4.8
         \n9    199.8    2.6       21.2   15.6\n10    66.1    5.8       24.2   12.6
         \n11   214.7   24.0        4.0   17.4\n12    23.8   35.1       65.9    9.2
         \n13    97.5    7.6        7.2   13.7\n14   204.1   32.9       46.0   19.0
         \n15   195.4   47.7       52.9   22.4\n16    67.8   36.6      114.0   12.5
         \n17   281.4   39.6       55.8   24.4\n18    69.2   20.5       18.3   11.3
         \n19   147.3   23.9       19.1   14.6\n20   218.4   27.7       53.4   18.0
         \n21   237.4    5.1       23.5   17.5\n22    13.2   15.9       49.6    5.6
         \n23   228.3   16.9       26.2   20.5\n24    62.3   12.6       18.3    9.7
         \n25   262.9    3.5       19.5   17.0\n26   142.9   29.3       12.6   15.0
         \n27   240.1   16.7       22.9   20.9\n28   248.8   27.1       22.9   18.9
         \n29    70.6   16.0       40.8   10.5\n30   292.9   28.3       43.2   21.4
         \n31   112.9   17.4       38.6   11.9\n32    97.2    1.5       30.0   13.2
         \n33   265.6   20.0        0.3   17.4\n34    95.7    1.4        7.4   11.9
         \n35   290.7    4.1        8.5   17.8\n36   266.9   43.8        5.0   25.4
         \n37    74.7   49.4       45.7   14.7\n38    43.1   26.7       35.1   10.1
         \n39   228.0   37.7       32.0   21.5\n40   202.5   22.3       31.6   16.6
         \n41   177.0   33.4       38.7   17.1\n42   293.6   27.7        1.8   20.7
         \n43   206.9    8.4       26.4   17.9\n44    25.1   25.7       43.3    8.5
         \n45   175.1   22.5       31.5   16.1\n46    89.7    9.9       35.7   10.6
         \n47   239.9   41.5       18.5   23.2\n48   227.2   15.8       49.9   19.8
         \n49    66.9   11.7       36.8    9.7\n50   199.8    3.1       34.6   16.4
         \n51   100.4    9.6        3.6   10.7\n52   216.4   41.7       39.6   22.6
         \n53   182.6   46.2       58.7   21.2\n54   262.7   28.8       15.9   20.2
         \n55   198.9   49.4       60.0   23.7\n56     7.3   28.1       41.4    5.5
         \n57   136.2   19.2       16.6   13.2\n58   210.8   49.6       37.7   23.8
         \n59   210.7   29.5        9.3   18.4\n60    53.5    2.0       21.4    8.1
         \n61   261.3   42.7       54.7   24.2\n62   239.3   15.5       27.3   20.7
         \n63   102.7   29.6        8.4   14.0\n64   131.1   42.8       28.9   16.0
         \n65    69.0    9.3        0.9   11.3\n66    31.5   24.6        2.2   11.0
         \n67   139.3   14.5       10.2   13.4\n68   237.4   27.5       11.0   18.9
         \n69   216.8   43.9       27.2   22.3\n70   199.1   30.6       38.7   18.3
         \n71   109.8   14.3       31.7   12.4\n72    26.8   33.0       19.3    8.8
         \n73   129.4    5.7       31.3   11.0\n74   213.4   24.6       13.1   17.0
         \n75    16.9   43.7       89.4    8.7\n76    27.5    1.6       20.7    6.9
         \n77   120.5   28.5       14.2   14.2\n78     5.4   29.9        9.4    5.3
         \n79   116.0    7.7       23.1   11.0\n80    76.4   26.7       22.3   11.8
         \n81   239.8    4.1       36.9   17.3\n82    75.3   20.3       32.5   11.3
         \n83    68.4   44.5       35.6   13.6\n84   213.5   43.0       33.8   21.7
         \n85   193.2   18.4       65.7   20.2\n86    76.3   27.5       16.0   12.0
         \n87   110.7   40.6       63.2   16.0\n88    88.3   25.5       73.4   12.9
         \n89   109.8   47.8       51.4   16.7\n90   134.3    4.9        9.3   14.0
         \n91    28.6    1.5       33.0    7.3\n92   217.7   33.5       59.0   19.4
         \n93   250.9   36.5       72.3   22.2\n94   107.4   14.0       10.9   11.5
         \n95   163.3   31.6       52.9   16.9\n96   197.6    3.5        5.9   16.7
         \n97   184.9   21.0       22.0   20.5\n98   289.7   42.3       51.2   25.4
         \n99   135.2   41.7       45.9   17.2\n100  222.4    4.3       49.8   16.7
         \n101  296.4   36.3      100.9   23.8\n102  280.2   10.1       21.4   19.8
         \n103  187.9   17.2       17.9   19.7\n104  238.2   34.3        5.3   20.7
         \n105  137.9   46.4       59.0   15.0\n106   25.0   11.0       29.7    7.2
         \n107   90.4    0.3       23.2   12.0\n108   13.1    0.4       25.6    5.3
         \n109  255.4   26.9        5.5   19.8\n110  225.8    8.2       56.5   18.4
         \n111  241.7   38.0       23.2   21.8\n112  175.7   15.4        2.4   17.1
         \n113  209.6   20.6       10.7   20.9\n114   78.2   46.8       34.5   14.6
         \n115   75.1   35.0       52.7   12.6\n116  139.2   14.3       25.6   12.2
         \n117   76.4    0.8       14.8    9.4\n118  125.7   36.9       79.2   15.9
         \n119   19.4   16.0       22.3    6.6\n120  141.3   26.8       46.2   15.5
```

```
\n121   18.8   21.7       50.4     7.0\n122  224.0    2.4       15.6   16.6
\n123  123.1   34.6       12.4    15.2\n124  229.5   32.3       74.2   19.7
\n125   87.2   11.8       25.9    10.6\n126    7.8   38.9       50.6    6.6
\n127   80.2    0.0        9.2    11.9\n128  220.3   49.0        3.2   24.7
\n129   59.6   12.0       43.1     9.7\n130    0.7   39.6        8.7    1.6
\n131  265.2    2.9       43.0    17.7\n132    8.4   27.2        2.1    5.7
\n133  219.8   33.5       45.1    19.6\n134   36.9   38.6       65.6   10.8
\n135   48.3   47.0        8.5    11.6\n136   25.6   39.0        9.3    9.5
\n137  273.7   28.9       59.7    20.8\n138   43.0   25.9       20.5    9.6
\n139  184.9   43.9        1.7    20.7\n140   73.4   17.0       12.9   10.9
\n141  193.7   35.4       75.6    19.2\n142  220.5   33.2       37.9   20.1
\n143  104.6    5.7       34.4    10.4\n144   96.2   14.8       38.9   12.3
\n145  140.3    1.9        9.0    10.3\n146  240.1    7.3        8.7   18.2
\n147  243.2   49.0       44.3    25.4\n148   38.0   40.3       11.9   10.9
\n149   44.7   25.8       20.6    10.1\n150  280.7   13.9       37.0   16.1
\n151  121.0    8.4       48.7    11.6\n152  197.6   23.3       14.2   16.6
\n153  171.3   39.7       37.7    16.0\n154  187.8   21.1        9.5   20.6
\n155    4.1   11.6        5.7     3.2\n156   93.9   43.5       50.5   15.3
\n157  149.8    1.3       24.3    10.1\n158   11.7   36.9       45.2    7.3
\n159  131.7   18.4       34.6    12.9\n160  172.5   18.1       30.7   16.4
\n161   85.7   35.8       49.3    13.3\n162  188.4   18.1       25.6   19.9
\n163  163.5   36.8        7.4    18.0\n164  117.2   14.7        5.4   11.9
\n165  234.5    3.4       84.8    16.9\n166   17.9   37.6       21.6    8.0
\n167  206.8    5.2       19.4    17.2\n168  215.4   23.6       57.6   17.1
\n169  284.3   10.6        6.4    20.0\n170   50.0   11.6       18.4    8.4
\n171  164.5   20.9       47.4    17.5\n172   19.6   20.1       17.0    7.6
\n173  168.4    7.1       12.8    16.7\n174  222.4    3.4       13.1   16.5
\n175  276.9   48.9       41.8    27.0\n176  248.4   30.2       20.3   20.2
\n177  170.2    7.8       35.2    16.7\n178  276.7    2.3       23.7   16.8
\n179  165.6   10.0       17.6    17.6\n180  156.6    2.6        8.3   15.5
\n181  218.5    5.4       27.4    17.2\n182   56.2    5.7       29.7    8.7
\n183  287.6   43.0       71.8    26.2\n184  253.8   21.3       30.0   17.6
\n185  205.0   45.1       19.6    22.6\n186  139.5    2.1       26.6   10.3
\n187  191.1   28.7       18.2    17.3\n188  286.0   13.9        3.7   20.9
\n189   18.7   12.1       23.4     6.7\n190   39.5   41.1        5.8   10.8
\n191   75.5   10.8        6.0    11.9\n192   17.2    4.1       31.6    5.9
\n193  166.8   42.0        3.6    19.6\n194  149.7   35.6        6.0   17.3
\n195   38.2    3.7       13.8     7.6\n196   94.2    4.9        8.1   14.0
\n197  177.0    9.3        6.4    14.8\n198  283.6   42.0       66.2   25.5
\n199  232.1    8.6        8.7    18.4'
```

Data preprocessing

In [5]:

```
print(df.head())
df.describe()
```

```
      TV  Radio  Newspaper  Sales
0  230.1   37.8       69.2   22.1
1   44.5   39.3       45.1   10.4
2   17.2   45.9       69.3   12.0
3  151.5   41.3       58.5   16.5
4  180.8   10.8       58.4   17.9
```

Out[5]:

	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	15.130500
std	85.854236	14.846809	21.778621	5.283892
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	11.000000
50%	149.750000	22.900000	25.750000	16.000000
75%	218.825000	36.525000	45.100000	19.050000
max	296.400000	49.600000	114.000000	27.000000

In [6]:

```
df.isnull().sum()
```

Out[6]:

```
TV           0
Radio        0
Newspaper    0
Sales        0
dtype: int64
```

In [7]:

```
df.duplicated()
```

Out[7]:

```
0      False
1      False
2      False
3      False
4      False
       ...  
195    False
196    False
197    False
198    False
199    False
Length: 200, dtype: bool
```

Plotting the data using scatter plot from pandas package

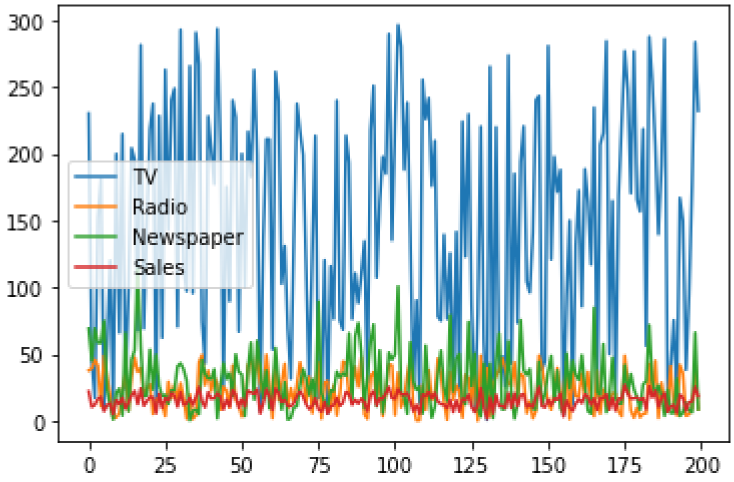
In [8]:

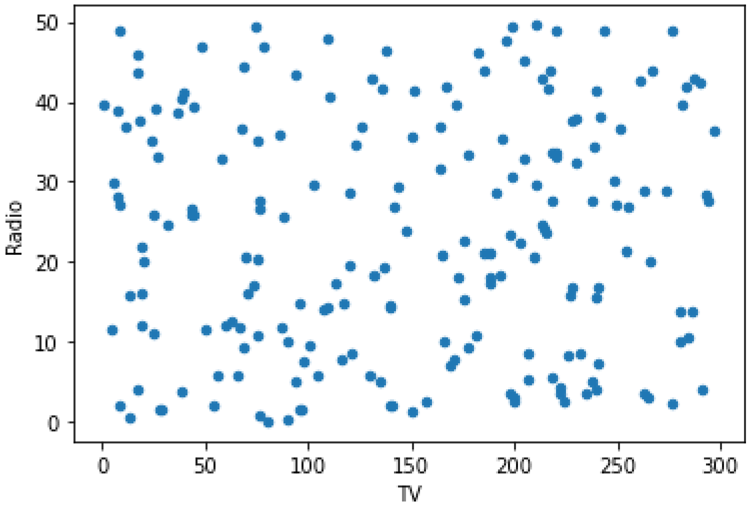
```
print(df.plot())
df.plot(kind='scatter',x='TV',y='Radio')
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
```

Out[8]:

```
<AxesSubplot:xlabel='TV', ylabel='Radio'>
```





In [9]:

```
df.corr()
```

Out[9]:

	TV	Radio	Newspaper	Sales
TV	1.000000	0.054809	0.056648	0.901208
Radio	0.054809	1.000000	0.354104	0.349631
Newspaper	0.056648	0.354104	1.000000	0.157960
Sales	0.901208	0.349631	0.157960	1.000000

Training and Testing the data using certain packages

In [10]:

```
from sklearn.model_selection import train_test_split
X=df.drop('Sales',axis=1)
Y=df['Sales']
print(X)
print(Y)
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_sta
```

```
        TV  Radio  Newspaper
0    230.1   37.8       69.2
1     44.5   39.3       45.1
2     17.2   45.9       69.3
3    151.5   41.3       58.5
4    180.8   10.8       58.4
..     ...    ...        ...
195   38.2    3.7       13.8
196   94.2    4.9        8.1
197  177.0    9.3        6.4
198  283.6   42.0       66.2
199  232.1    8.6        8.7

[200 rows x 3 columns]
0      22.1
1      10.4
2      12.0
3      16.5
4      17.9
       ...
195     7.6
196    14.0
197    14.8
198    25.5
199    18.4
Name: Sales, Length: 200, dtype: float64
```

In [11]:

```
from sklearn.linear_model  import LinearRegression
f1=LinearRegression()
f1.fit(X_train,Y_train)
y=f1.predict(X_test)
l1=f1.score(X_test,Y_test)*100
print("Accuracy of Sales prediction using LinearRegression is",l1)
```

```
Accuracy of Sales prediction using LinearRegression is 90.59011844150827
```

In [12]:

```
import matplotlib.pyplot as plt
a=df['TV'].head()
b=df['Radio'].head()
print(a)
b
```

```
0    230.1
1     44.5
2     17.2
3    151.5
4    180.8
Name: TV, dtype: float64
```

Out[12]:

```
0    37.8
1    39.3
2    45.9
3    41.3
4    10.8
Name: Radio, dtype: float64
```

In [13]:

```
arr1=np.array(a)
arr2=np.array(b)
print(arr1)
arr2
```

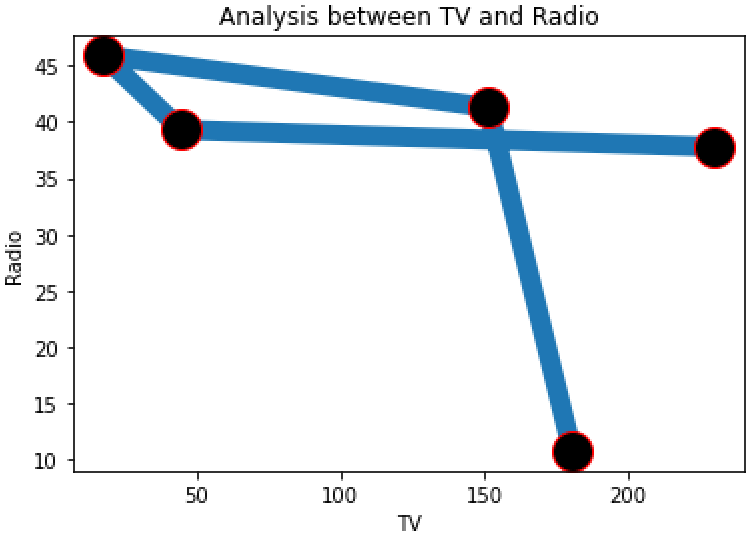
```
[230.1  44.5  17.2 151.5 180.8]
```

Out[13]:

```
array([37.8, 39.3, 45.9, 41.3, 10.8])
```

In [14]:

```
plt.plot(a,b,marker='o',mec='r',mfc='k',ls='-',markersize=20,linewidth=10)
plt.xlabel("TV")
plt.ylabel("Radio")
plt.title("Analysis between TV and Radio")
plt.show()
```



# Using Flatten() convert the 2d into 1d

In [20]:

```
import pandas as pd
data=pd.DataFrame(data={"Predicted Sales":y.flatten()})
print(data)
```

```
    Predicted Sales
0         17.034772
1         20.409740
2         23.723989
3          9.272785
4         21.682719
5         12.569402
6         21.081195
7          8.690350
8         17.237013
9         16.666575
10         8.923965
11         8.481734
12        18.207512
13         8.067507
14        12.645510
15        14.931628
16         8.128146
17        17.898766
18        11.008806
19        20.478328
20        20.806318
21        12.598833
22        10.905183
23        22.388548
24         9.417961
25         7.925067
26        20.839085
27        13.815209
28        10.770809
29         7.926825
30        15.959474
31        10.634909
32        20.802920
33        10.434342
34        21.578475
35        21.183645
36        12.128218
37        22.809533
38        12.609928
39         6.464413
```