Sales_Prediction

In [1]:

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

In [2]:

```
df = pd.read_csv('Advertising.csv') #reading dataset
```

In [3]:

```
df.head() #give first 5 entriers
```

Out[3]:

	Unnamed: 0	TV	Radio	Newspaper	Sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9

In [4]:

```
df.tail() #give last 5 entriers
```

Out[4]:

	Unnamed: 0	TV	Radio	Newspaper	Sales
195	196	38.2	3.7	13.8	7.6
196	197	94.2	4.9	8.1	9.7
197	198	177.0	9.3	6.4	12.8
198	199	283.6	42.0	66.2	25.5
199	200	232.1	8.6	8.7	13.4

In [5]:

```
df.size
```

Out[5]:

```
1000
```

In [6]:

```
df.shape
```

Out[6]:

```
(200, 5)
```

In [7]:

```
df.info() #give information about dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Unnamed: 0  200 non-null    int64  
 1   TV          200 non-null    float64
 2   Radio       200 non-null    float64
 3   Newspaper   200 non-null    float64
 4   Sales       200 non-null    float64
dtypes: float64(4), int64(1)
memory usage: 7.9 KB
```

In [8]:

```
df.describe() #give  description about dataset
```

Out[8]:

	Unnamed: 0	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000	200.000000
mean	100.500000	147.042500	23.264000	30.554000	14.022500
std	57.879185	85.854236	14.846809	21.778621	5.217457
min	1.000000	0.700000	0.000000	0.300000	1.600000
25%	50.750000	74.375000	9.975000	12.750000	10.375000
50%	100.500000	149.750000	22.900000	25.750000	12.900000
75%	150.250000	218.825000	36.525000	45.100000	17.400000
max	200.000000	296.400000	49.600000	114.000000	27.000000

In [9]:

```
df.drop(columns=['Unnamed: 0'], axis=1, inplace=True)
```

In [10]:

```
df
```

Out[10]:

	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	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9
...	...	...	...	...
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	9.7
197	177.0	9.3	6.4	12.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	13.4

200 rows × 4 columns

In [11]:

```
df.isnull().sum() #finding null values
```

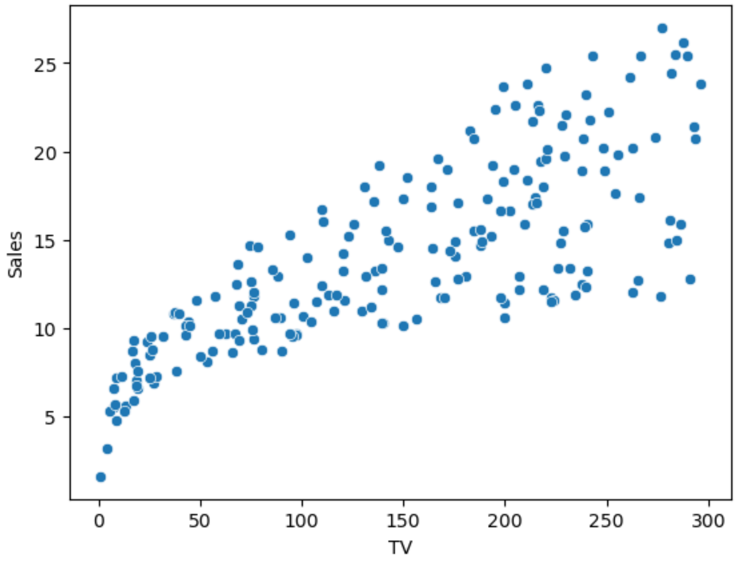
Out[11]:

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

Data Visualization

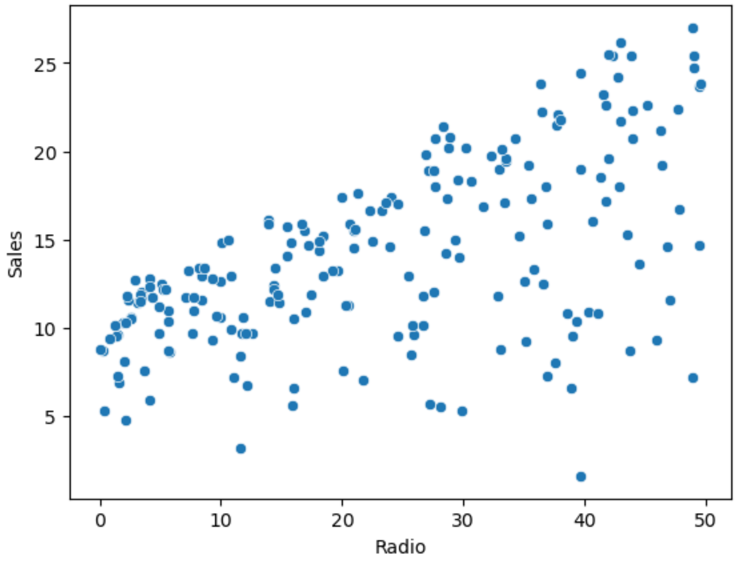
In [15]:

```
sns.scatterplot(data=df, x='TV', y='Sales')
plt.show()
```



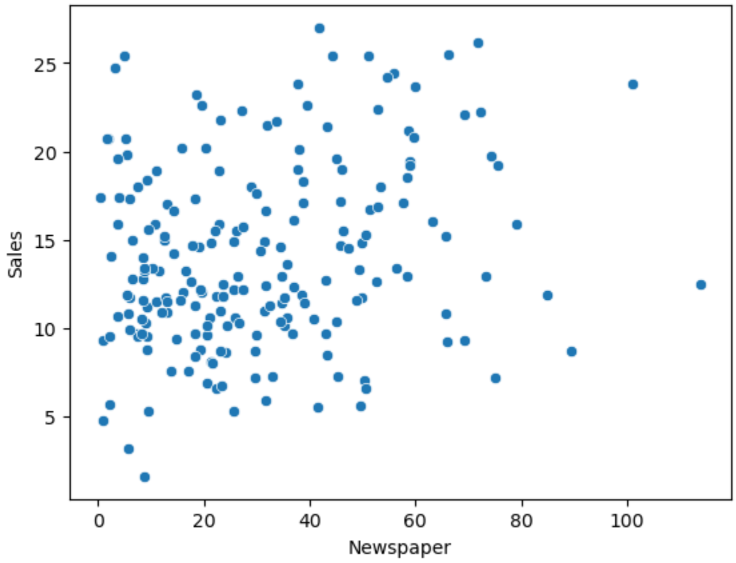
In [18]:

```
sns.scatterplot(data=df, x='Radio', y='Sales')
plt.show()
```



In [21]:

```
sns.scatterplot(data=df, x='Newspaper', y='Sales')
plt.show()
```



In [22]:

```
x=df.drop(['Sales'],1)
x.head()
```

```
C:\Users\baps\AppData\Local\Temp\ipykernel_19264\1013197829.py:1: FutureWarning: In a future version of pandas all argument
s of DataFrame.drop except for the argument 'labels' will be keyword-only.
  x=df.drop(['Sales'],1)
```

Out[22]:

	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

In [23]:

```
y=df['Sales']
y.head()
```

Out[23]:

```
0    22.1
1    10.4
2     9.3
3    18.5
4    12.9
Name: Sales, dtype: float64
```

In [24]:

```
from sklearn.model_selection import train_test_split
```

```
In [25]: _train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=100)   #splitting datset into train and 
```

```
In [26]: from sklearn.linear_model import LinearRegression   #creating model
```

```
In [27]: lr=LinearRegression()
```

```
In [28]: lr.fit(x_train,y_train)
```

```
Out[28]: ▾ LinearRegression
         LinearRegression()
```

```
In [29]: y_prediction=lr.predict(x_test)
         y_prediction
```

```
Out[29]: array([10.50948755, 20.05723558, 16.82665516, 19.09405782, 20.94383845,
                13.25348896, 11.87554518, 12.4882643 , 20.51420388, 20.91328754,
                10.90182094, 19.44307823,  6.5040246 , 15.3178433 ,  9.04147699,
                 8.03952046, 16.26581002, 12.10312044, 17.12086602, 11.28131376,
                17.07036157,  9.84452136, 20.70077487, 17.17479561, 15.19498306,
                21.96631309, 19.1587532 , 10.09216975, 19.40214482, 14.88180369,
                14.27844074,  7.69173892, 10.00183154, 14.82404891,  7.29177732,
                13.63453936,  7.59380446, 11.84106127, 13.61631336, 15.18810875])
```

```
In [30]: coefficiant=lr.coef_
         coefficiant
```

```
Out[30]: array([0.0455864 , 0.18569816, 0.00223281])
```

```
In [31]: intercept=lr.intercept_
         intercept
```

```
Out[31]: 2.8172751352950005
```

```
In [32]: from sklearn.metrics import r2_score
         r2_score(y_test,y_prediction)*100
```

```
Out[32]: 91.84369032278497
```

THANK YOU!

GitHub: https://github.com/anujtiwari21?tab=repositories (https://github.com/anujtiwari21?tab=repositories)