In [5]:

```
#importing the packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [6]:

```
#Reading the csv file
data=pd.read_csv("Advertising.csv")
```

In [7]:

```
#displaying the dataset
data
```

Out[7]:

	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
...	...	...	...	...	...
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

200 rows × 5 columns

In [8]:

```
#diplaying first 5 rows
data.head()
```

Out[8]:

	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 [16]:

```
#displaying bottom 5 rows
data.tail()
```

Out[16]:

	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 [17]:

```
#displaying random 5 rows
data.sample(5)
```

Out[17]:

	Unnamed: 0	TV	Radio	Newspaper	Sales
88	89	88.3	25.5	73.4	12.9
51	52	100.4	9.6	3.6	10.7
188	189	286.0	13.9	3.7	15.9
8	9	8.6	2.1	1.0	4.8
3	4	151.5	41.3	58.5	18.5

In [10]:

```
data.shape
```

Out[10]:

```
(200, 5)
```

In [11]:

```
data.describe()
```

Out[11]:

	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 [12]:

```
data.info()
```

```
<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 [13]:

```
data.duplicated().sum()
```

Out[13]:

```
0
```

In [14]:

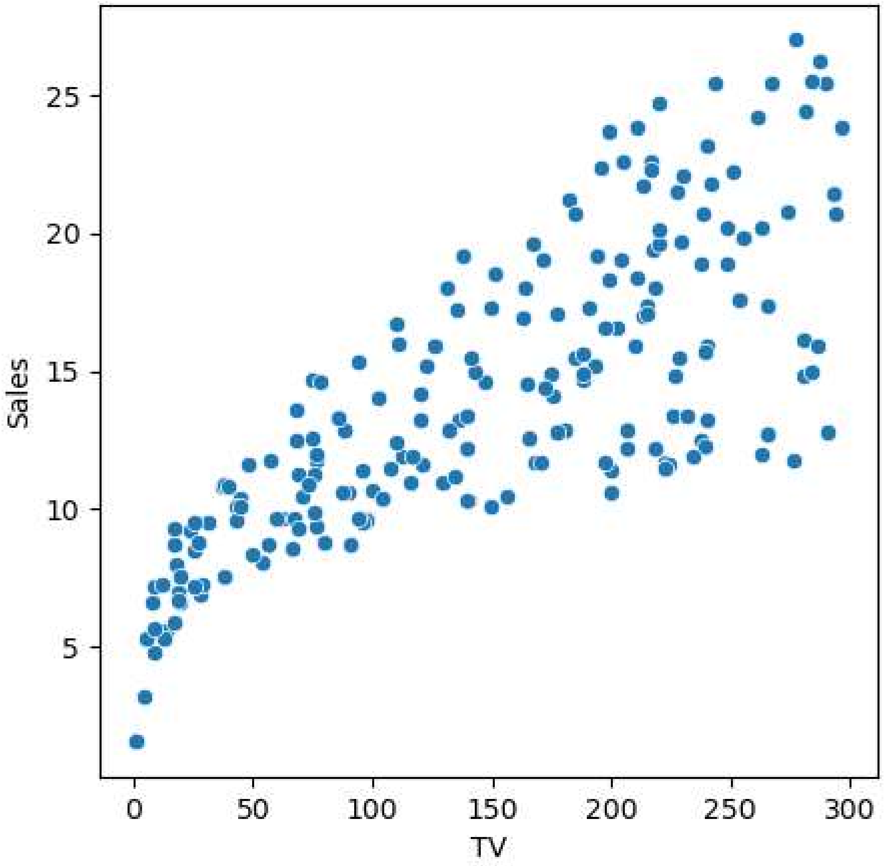
```
data.isnull().sum()
```

Out[14]:

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

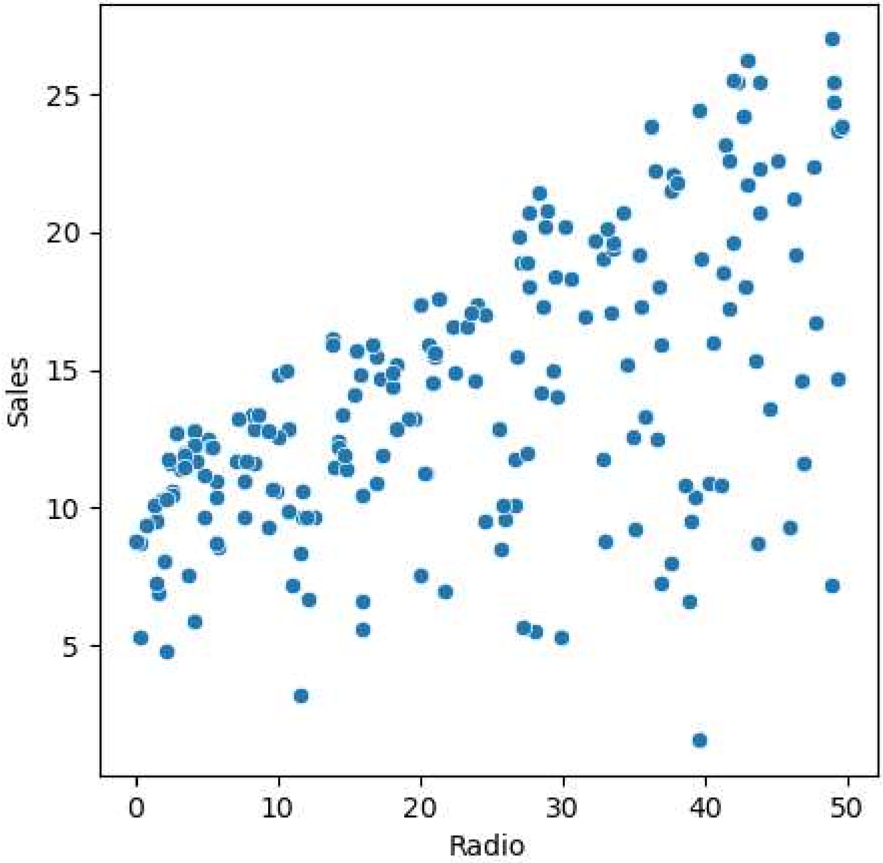
In [15]:

```
#scatterplot
plt.figure(figsize=(5,5))
sns.scatterplot(data=data,x=data['TV'],y=data['Sales'])
plt.show()
```



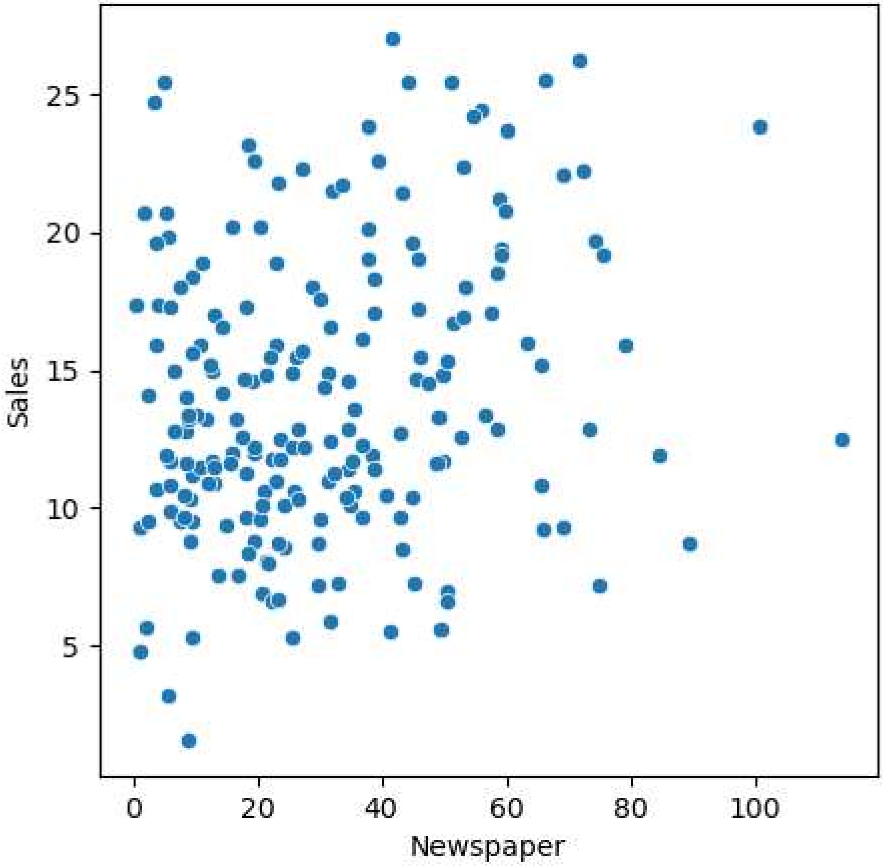
In [13]:

```
plt.figure(figsize=(5,5))
sns.scatterplot(data=data,x=data['Radio'],y=data['Sales'])
plt.show()
```



In [14]:

```
plt.figure(figsize=(5,5))
sns.scatterplot(data=data,x=data['Newspaper'],y=data['Sales'])
plt.show()
```



In [15]:

```
#Splitting the dataset into X(the attributes) and y(the target variable)
X=data.drop('Sales',axis=1)
```

In [16]:

```
y=data['Sales']
y
```

Out[16]:

```
0      22.1
1      10.4
2       9.3
3      18.5
4      12.9
       ... 
195     7.6
196     9.7
197    12.8
198    25.5
199    13.4
Name: Sales, Length: 200, dtype: float64
```

In [17]:

```
#splitting the data into train and test data
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.20, random_state=0)
```

In [19]:

```
from sklearn.linear_model import LinearRegression
sales=LinearRegression()
```

In [20]:

```
#fitting the model
sales.fit(X_train,y_train)
```

```
Out[20]: LinearRegression()
```

```
In [21]: #prediction
         prediction=sales.predict(X_test)
```

```
In [22]: prediction
```

```
Out[22]: array([10.24664397,  7.3067958 ,  7.01313011, 24.0629912 , 11.84538583,
                 6.3623126 , 12.97367539, 15.01392577, 10.95137181, 16.2302401 ,
                22.94764396,  9.07892877, 10.29830431, 15.372556  , 11.63358138,
                12.38729183, 18.91996453, 10.72601207, 16.03372676, 17.36589369,
                23.85225051,  9.51048184, 15.25366204, 12.2528663 ,  5.81346738,
                15.19998588, 12.12398331, 20.74762723, 13.36651757,  9.16548788,
                13.0430809 , 21.47977498, 18.01828076, 21.14170908,  6.74096317,
                 5.9527019 ,  7.82515133, 13.21661502, 14.72150697,  6.18172688])
```

```
In [23]: #Determining accuracy using different metrics
         from sklearn import metrics
         print('MAE:',metrics.mean_absolute_error(prediction,y_test))
         print('RMSE:',np.sqrt(metrics.mean_squared_error(prediction,y_test)))
         print('R-Squared',metrics.r2_score(prediction,y_test))
```

```
MAE: 1.4158480868317487
RMSE: 2.125203007295874
R-Squared 0.8283777385812958
```

```
In [ ]:
```