Task 4 (Sales Prediction using python)

Done by Cherukuri Krishna Lathvik

Importing the Libraries

In [2]:

```
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn import metrics
import matplotlib.pyplot as plt
```

In [4]:

```
df1 = pd.read_csv("advertising.csv")
df1.head()
```

Out[4]:

	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

In [5]:

```
df1.info
```

Out[5]:

```
<bound method DataFrame.info of          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 x 4 columns]>
```

In [6]:

```
df1.shape
```

Out[6]:

```
(200, 4)
```

In [7]:

```
df1.describe()
```

Out[7]:

	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

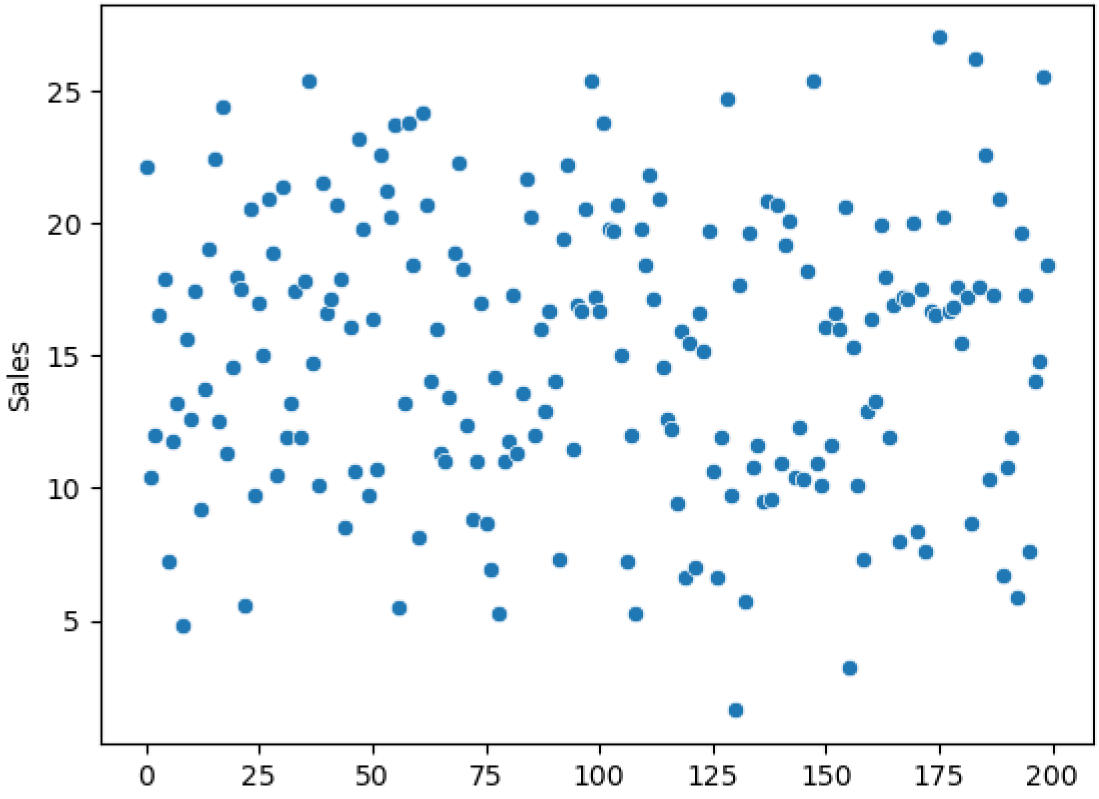
Visualising the Data

In [8]:

```
sns.scatterplot(df1["Sales"])
```

Out[8]:

```
<Axes: ylabel='Sales'>
```

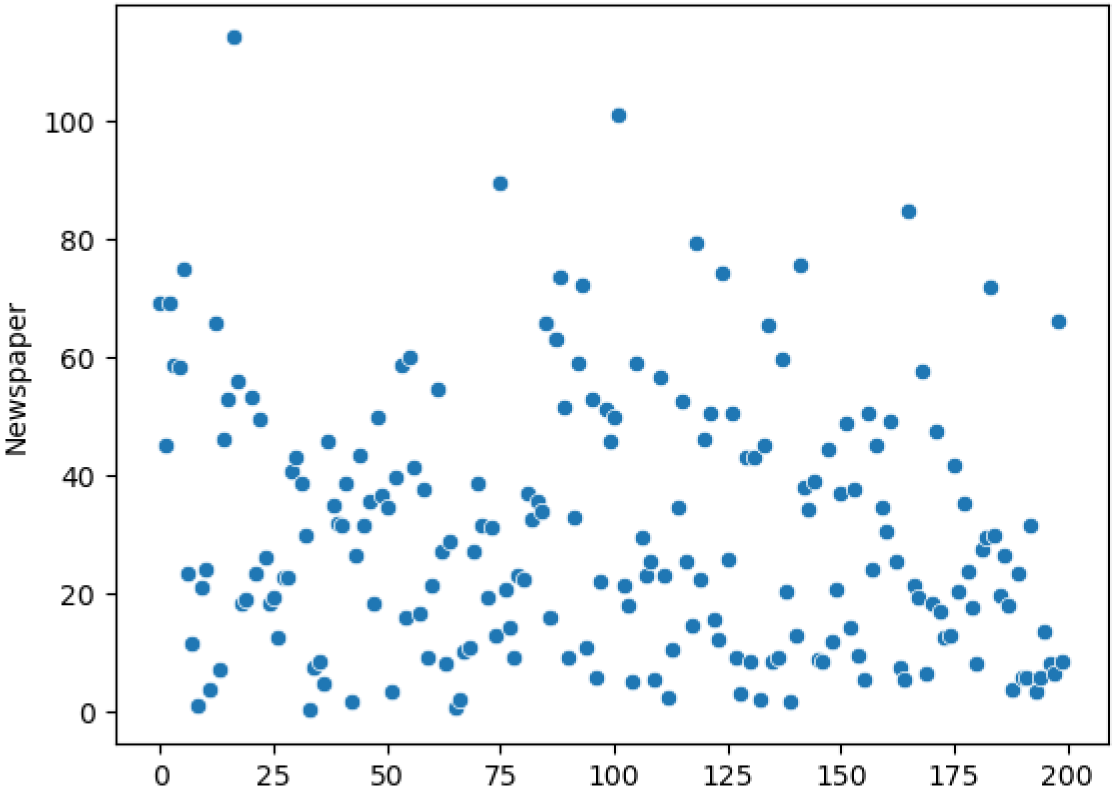


In [9]:

```
sns.scatterplot(df1["Newspaper"])
```

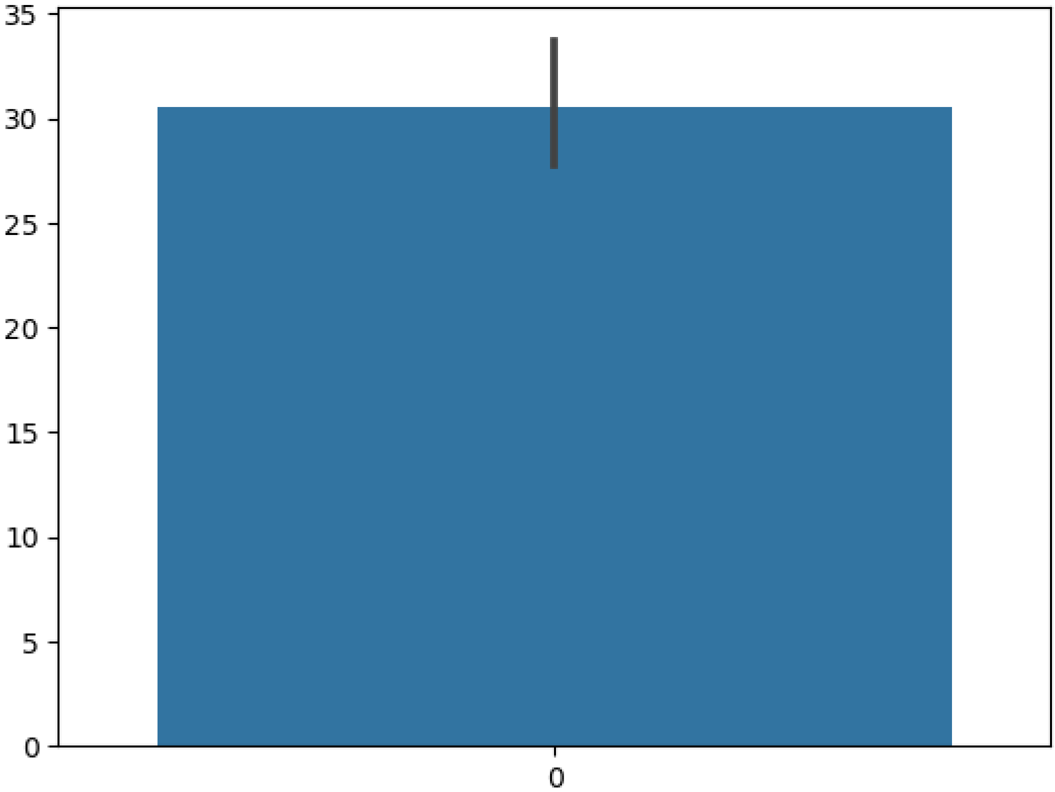
Out[9]:

```
<Axes: ylabel='Newspaper'>
```



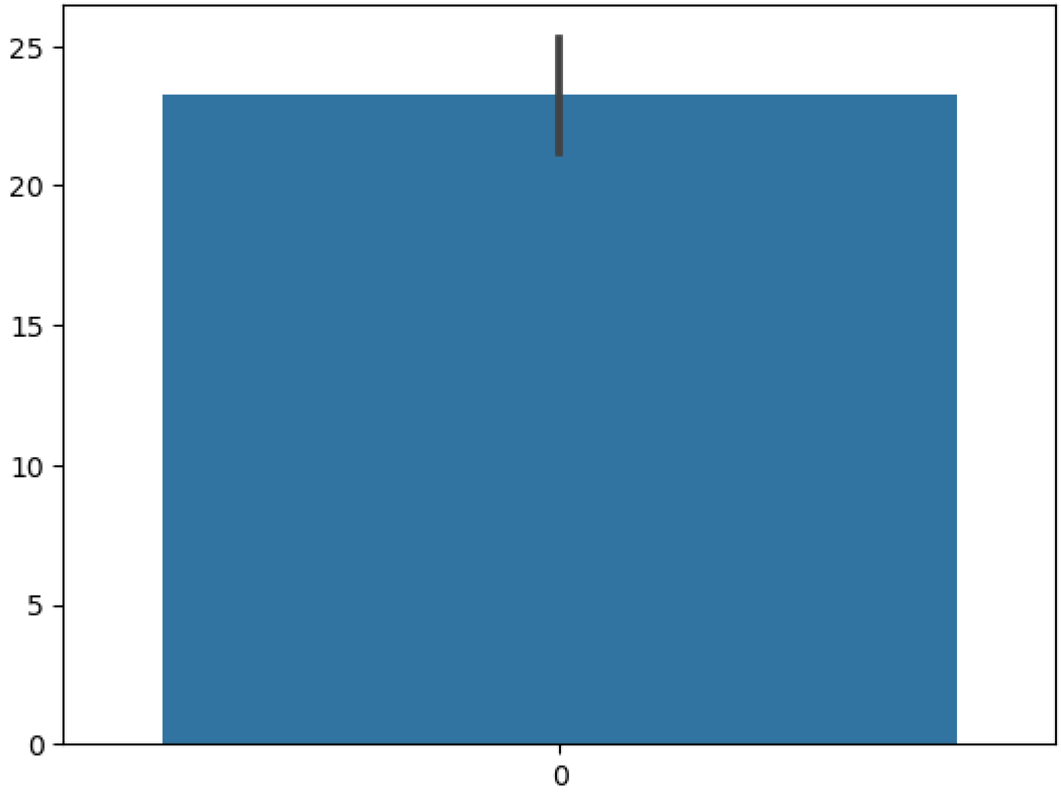
In [11]:
```
sns.barplot(df1["Newspaper"])
```

Out[11]:
```
<Axes: >
```



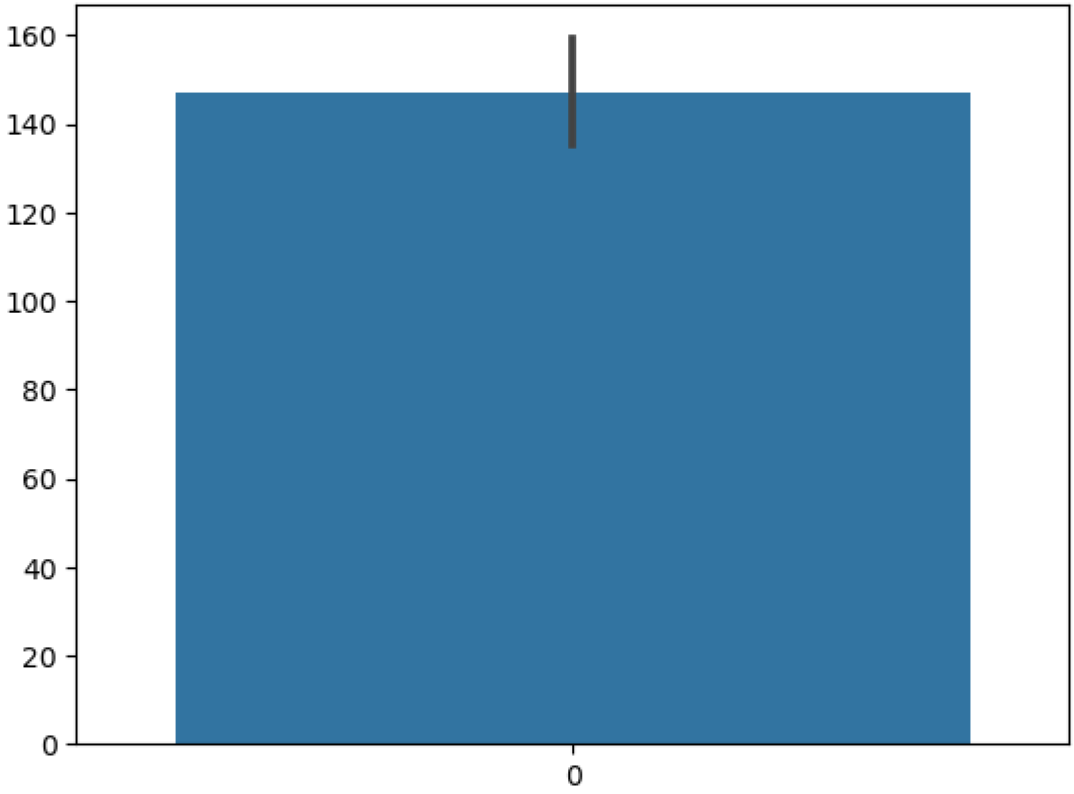
In [12]:
```
sns.barplot(df1["Radio"])
```

Out[12]:
```
<Axes: >
```



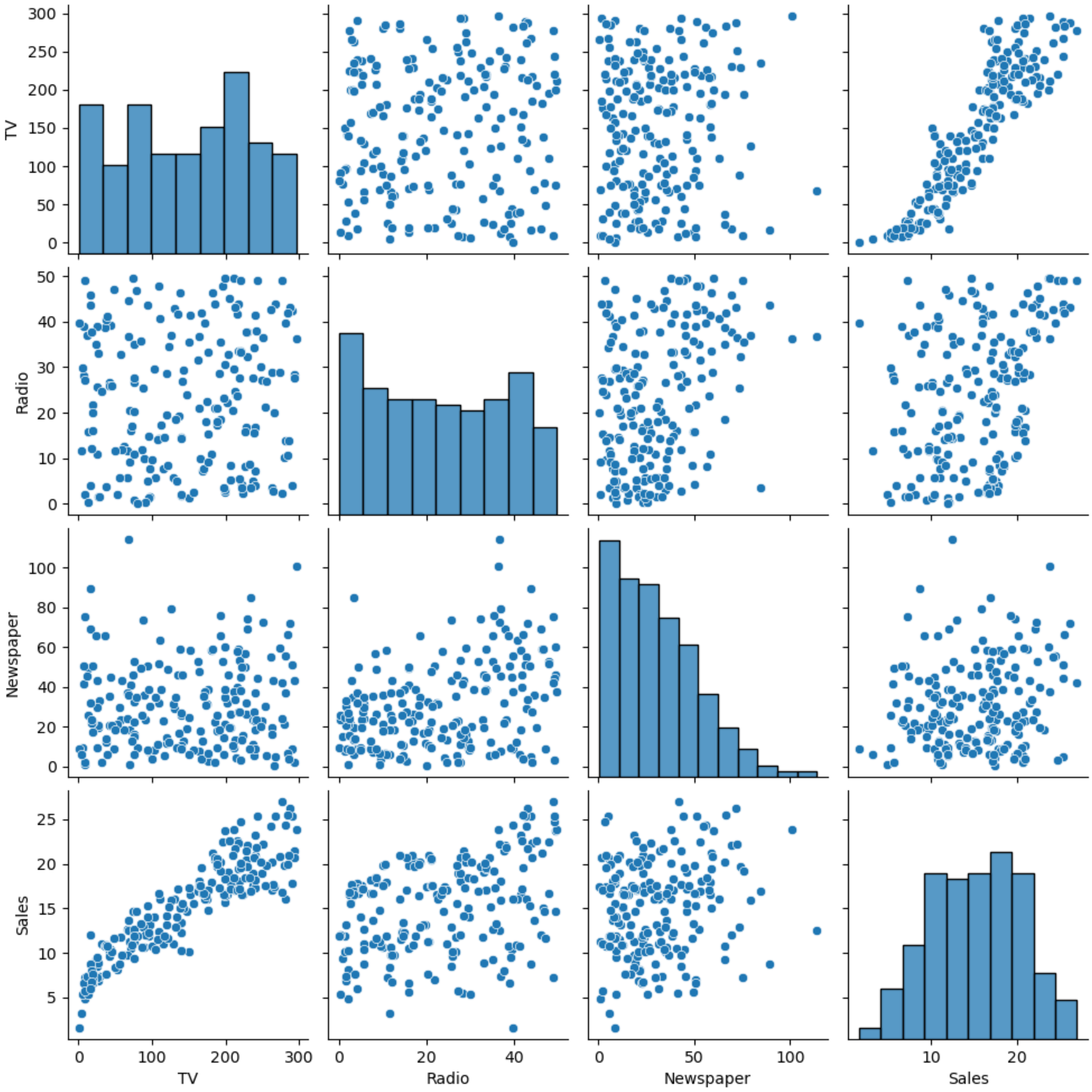
```
In [13]: sns.barplot(df1["TV"])
```

```
Out[13]: <Axes: >
```



```
In [14]: sns.pairplot(df1)
```

```
Out[14]: <seaborn.axisgrid.PairGrid at 0x1cb056a0690>
```



In [15]:
```
df1.isnull().any()
```

Out[15]:
```
TV           False
Radio        False
Newspaper    False
Sales        False
dtype: bool
```

Now splitting the data into Dependent and Independent variables

In [16]:
```
X = df1.iloc[:,0:3]
y = df1["Sales"]
```

In [17]:
```
X.head()
```

Out[17]:

	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 [18]:

```
y.head()
```

Out[18]:

```
0    22.1
1    10.4
2    12.0
3    16.5
4    17.9
Name: Sales, dtype: float64
```

In [19]:

```
X.shape
```

Out[19]:

```
(200, 3)
```

In [20]:

```
y.shape
```

Out[20]:

```
(200,)
```

Now splitting the data into Training and Testing datasets

In [22]:

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)
```

In [23]:

```
Lr = LinearRegression()
Lr.fit(X_train, y_train)
```

Out[23]:

```
▾ LinearRegression
LinearRegression()
```

In [24]:

```
Lr.intercept_
```

Out[24]:

```
4.714126402214131
```

In [25]:

```
Lr.coef_
```

Out[25]:

```
array([0.05450927, 0.10094536, 0.00433665])
```

In [26]:

```
y_pred=Lr.predict(X_test)
```

Evaluation

In [27]:

```
r2_score(y_test,y_pred)
```

```
Out[27]: 0.9059011844150826
```

```
In [28]: metrics.mean_squared_error(y_test,y_pred)
```

```
Out[28]: 2.9077569102710905
```

```
In [29]: np.sqrt(metrics.mean_squared_error(y_test,y_pred))
```

```
Out[29]: 1.7052146229349228
```

```
In [ ]:
```