```
import numpy as np
import pandas as pd
import scipy.stats as sp
import statistics as stats
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.model_selection import train_test_split
import statsmodels.api as stats
```

```
df=pd.read_csv('advertising.csv')
```

```
df
```

	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

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   TV         200 non-null    float64
 1   Radio      200 non-null    float64
 2   Newspaper  200 non-null    float64
 3   Sales      200 non-null    float64
dtypes: float64(4)
memory usage: 6.4 KB
```

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

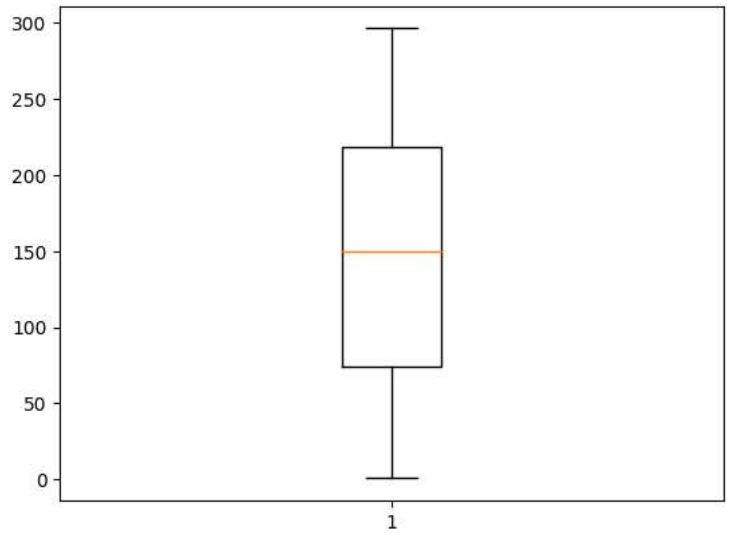
```
TV           0
Radio        0
Newspaper    0
Sales        0
dtype: int64
```

```
df.describe()
```

	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

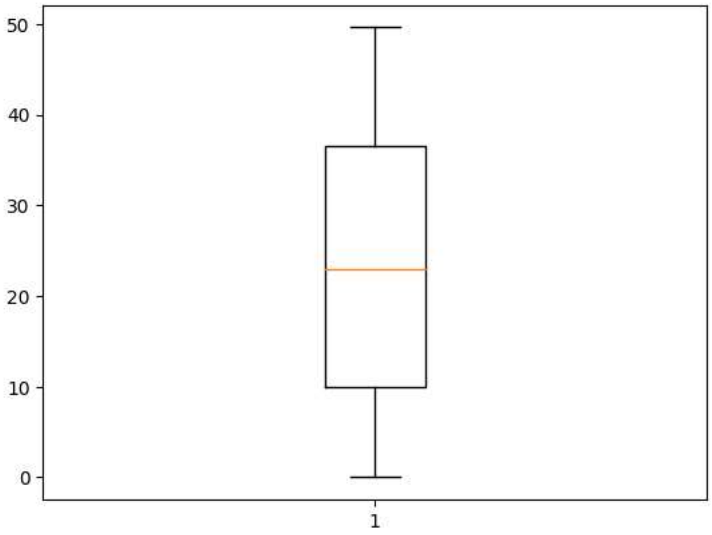
```
plt.boxplot(df['TV'])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x7d1e055e43d0>,
  <matplotlib.lines.Line2D at 0x7d1e443fcd00>],
 'caps': [<matplotlib.lines.Line2D at 0x7d1e052e8d30>,
  <matplotlib.lines.Line2D at 0x7d1e052ea560>],
 'boxes': [<matplotlib.lines.Line2D at 0x7d1e055e6290>],
 'medians': [<matplotlib.lines.Line2D at 0x7d1e052e8be0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7d1e052e9240>],
 'means': []}
```



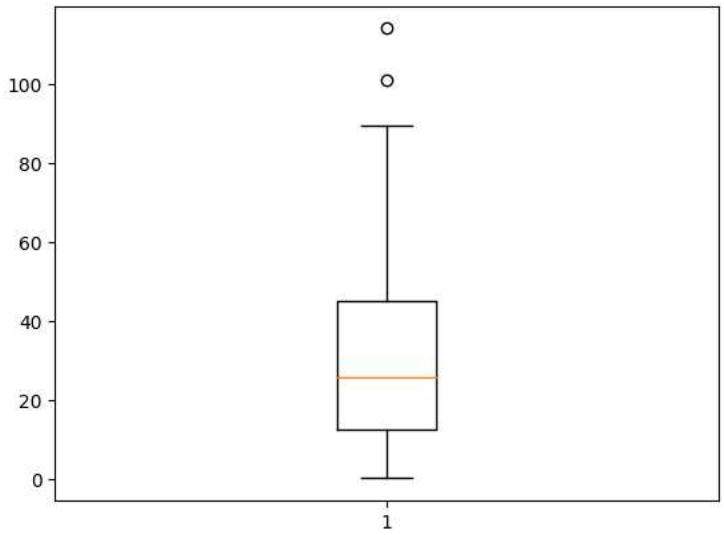

```
plt.boxplot(df['Radio'])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x7d1e0529c3a0>,
  <matplotlib.lines.Line2D at 0x7d1e0529e230>],
 'caps': [<matplotlib.lines.Line2D at 0x7d1e053eaec0>,
  <matplotlib.lines.Line2D at 0x7d1e053ebd30>],
 'boxes': [<matplotlib.lines.Line2D at 0x7d1e0529e320>],
 'medians': [<matplotlib.lines.Line2D at 0x7d1e053e8be0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7d1e053e8c10>],
 'means': []}
```



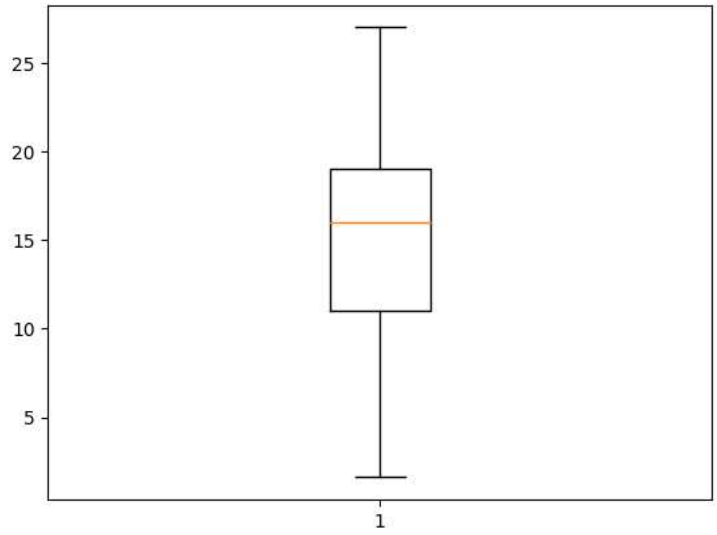

```
plt.boxplot(df['Newspaper'])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x7d1e053b4e50>,
  <matplotlib.lines.Line2D at 0x7d1e053b6b00>],
 'caps': [<matplotlib.lines.Line2D at 0x7d1e053b5960>,
  <matplotlib.lines.Line2D at 0x7d1e053b7790>],
 'boxes': [<matplotlib.lines.Line2D at 0x7d1e053b6050>],
 'medians': [<matplotlib.lines.Line2D at 0x7d1e053b77f0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7d1e053b5390>],
 'means': []}
```




```
plt.boxplot(df['Sales'])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x7d1e02a3fc70>,
  <matplotlib.lines.Line2D at 0x7d1e02a3ca60>],
 'caps': [<matplotlib.lines.Line2D at 0x7d1e02a3c310>,
  <matplotlib.lines.Line2D at 0x7d1e02a3d4b0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7d1e02a3cf40>],
 'medians': [<matplotlib.lines.Line2D at 0x7d1e02a3fcd0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7d1e02a3fee0>],
 'means': []}
```

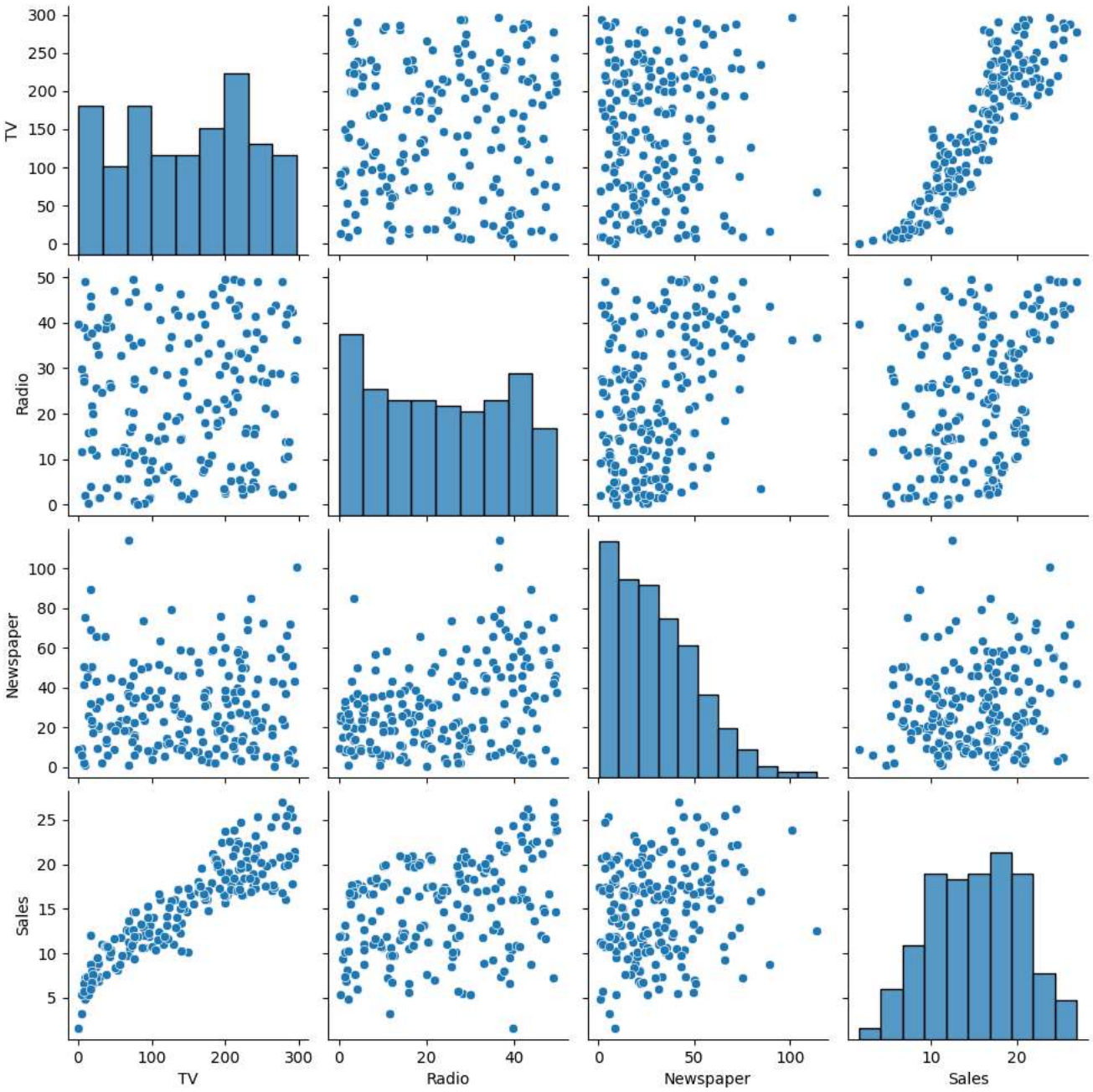



```
df[df['Newspaper'] >=100]
```

	TV	Radio	Newspaper	Sales
16	67.8	36.6	114.0	12.5
101	296.4	36.3	100.9	23.8

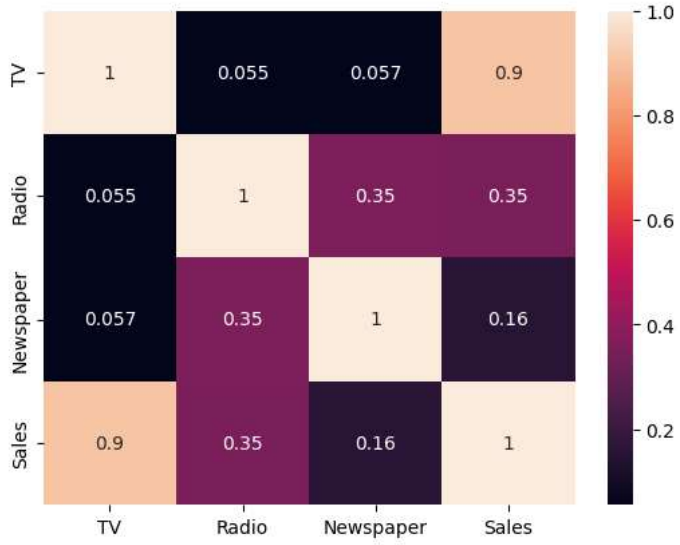
```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7d1e028fc850>
```




```
sns.heatmap(df.corr(),annot= True)
```

```
<Axes: >
```




```
X= df[['TV','Newspaper','Radio']]
X
```

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

200 rows × 3 columns

```
y=df['Sales']
y
```

```
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
```

```
LR= LinearRegression()
```

```
LR.fit(X,y)
```

```
▾ LinearRegression
LinearRegression()
```

```
y_predict= LR.predict(X)
y_predict
```

```
array([21.22097209, 11.26824775, 10.49620897, 17.31244651, 15.64413685,
       10.35633677, 11.2732847 , 13.27062458,  5.31839603, 15.78871013,
        8.8527202 , 18.88400523,  9.69879662, 10.74921373, 19.27328852,
       20.38554445, 12.27105794, 24.20214502, 10.5924398 , 15.20872794,
       19.49794066, 18.10414656,  7.06177654, 18.87221072,  9.37145421,
       19.31996936, 15.54479137, 19.49216302, 21.07865408, 10.19471067,
       23.61492832, 12.64683045, 10.08782551, 21.2260486 ,  9.98787085,
       20.89407055, 23.84503494, 13.99342411,  9.8404516 , 21.08344935,
       18.04712878, 17.84885818, 23.57494339, 16.79762772,  8.75617872,
       16.57668108, 10.58020572, 22.13342743, 18.70257411,  9.53181337,
       15.84670856, 11.11990058, 20.88243422, 19.53008343, 22.01500291,
       20.76038994,  8.04320903, 14.10063486, 21.4222098 , 19.25650785,
        7.75915886, 23.43911942, 19.32168181, 13.3867616 , 16.35231896,
        9.37729644,  8.97313482, 13.7643628 , 20.49677834, 21.13545307,
       18.75250649, 12.14402868,  9.62178972, 12.29082115, 18.88048094,
       10.25121926,  6.30053312, 14.24014196,  8.1216232 , 11.77249775,
       11.64919966, 18.13231302, 10.90792515, 13.12271953, 20.86169623,
       17.13492417, 11.72724142, 15.01773541, 12.18585509, 15.73518229,
       12.46462002,  6.35385195, 20.08231542, 22.21538326, 11.97427675,
       16.91511512, 15.76009495, 16.94655913, 24.94140428, 16.4635515 ,
       17.21068667, 24.68086584, 20.96872721, 16.70191561, 21.26603007,
       17.117858  ,  7.17325114,  9.58691025,  5.38975714, 21.41075554,
       17.81535602, 21.85850312, 15.83987218, 18.24477648, 13.90202178,
       12.47673434, 13.74268711,  8.87535042, 15.4438881 ,  7.40087704,
       15.20145315,  7.98754856, 17.08301809, 15.03380429, 20.60147616,
       10.64410416,  9.22913323,  8.99476371, 21.86366378,  9.16857418,
        8.90340499, 19.38888188,  7.99360692, 20.19198592, 10.78643994,
       12.28676609, 10.19510558, 22.63930843,  9.74450543, 19.3900734 ,
       10.44479522, 18.98449095, 20.19558086, 10.94160634, 11.45948342,
       12.47019032, 18.48158513, 23.12426769, 11.01020756,  9.8263967 ,
       21.40779104, 12.12822036, 17.88150523, 18.21228932, 17.11095629,
        6.09147928, 14.40908701, 12.92836006,  9.22565677, 13.77606971,
       15.96404812, 13.13831936, 16.82802417, 17.46713823, 12.58090014,
       17.78492754,  9.63019994, 16.4474296 , 18.89730805, 21.24042066,
        8.59480345, 15.8336908 ,  7.84869225, 14.55779863, 17.10206692,
       24.94755122, 21.38770686, 14.73822063, 19.94432942, 14.71726516,
       13.43232243, 17.10853074,  8.30485298, 24.90888356, 20.73265903,
       20.61884334, 12.45394152, 18.10675693, 21.68517626,  6.94582943,
       11.1754297 ,  9.89340771,  6.01090333, 18.20194019, 16.58691507,
        7.10548951, 10.28094143, 15.25928683, 24.58221952, 18.18512048])
```

```
sns.regplot(x=y, y=y_predict)
plt.xlabel('Actual Sales')
plt.ylabel('Predicted Sales')
plt.title('Actual vs Predicted Sales')
plt.show()
```