In [49]:
```
import pandas as pd
import numpy as np
import plotly.graph_objects as go
from sklearn.model_selection import train_test_split
import seaborn as sns
import matplotlib.pyplot as plt
```

In [3]:
```
df=pd.read_csv('Advertising.csv')
```

In [4]:
```
df
```

Out[4]:

	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 [5]:
```
df.shape
```

Out[5]:
```
(200, 5)
```

In [6]:
```
df.head()
```

Out[6]:

	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 [7]:
```
df.tail()
```

Out[7]:

	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 [8]:
```
df.describe
```

Out[8]:
```
<bound method NDFrame.describe of      Unnamed: 0     TV  Radio  Newspaper  S
ales
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 x 5 columns]>
```

In [9]:
```
df.info
```

Out[9]:
```
<bound method DataFrame.info of      Unnamed: 0     TV  Radio  Newspaper  Sal
es
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 x 5 columns]>
```

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
Unnamed: 0    0
TV            0
Radio         0
Newspaper     0
Sales         0
dtype: int64
```

In [11]:
```
df.duplicated().sum()
```

Out[11]:
```
0
```

In [14]:
```
fig.add_trace(go.Scatter(x=df['TV'], y=df['Sales'], mode='markers', marker=dic

fig.update_layout(title='TV Sales', xaxis_title='TV', yaxis_title='Sales')

fig.show()
```

In [15]:
```
fig.add_trace(go.Scatter(x=df['Radio'], y=df['Sales'], mode='markers', marker=
## Here we update these values under function attributes such as title,xaxis_t
fig.update_layout(title='Radio Sales', xaxis_title='Radio', yaxis_title='Sales
# Display the figure
fig.show()
```

In [16]:
```
fig.add_trace(go.Scatter(x=df['Newspaper'], y=df['Sales'], mode='markers', mar
## Here we update these values under function attributes such as title,xaxis_t
fig.update_layout(title='Newspaper Sales', xaxis_title='Newspaper', yaxis_titl
# Display the figure
fig.show()
```

In [18]:
```
X = df.drop('Sales', axis=1)
```

In [21]:
```
X
```

Out[21]:

	Unnamed: 0	TV	Radio	Newspaper
0	1	230.1	37.8	69.2
1	2	44.5	39.3	45.1
2	3	17.2	45.9	69.3
3	4	151.5	41.3	58.5
4	5	180.8	10.8	58.4
...	...	...	...	...
195	196	38.2	3.7	13.8
196	197	94.2	4.9	8.1
197	198	177.0	9.3	6.4
198	199	283.6	42.0	66.2
199	200	232.1	8.6	8.7

200 rows × 4 columns

In [22]:
```
X = X.drop(columns='Unnamed: 0')
X
```

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
...	...	...	...
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 × 3 columns

In [24]:
```
Y=df[['Sales']]
Y
```

Out[24]:

	Sales
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

200 rows × 1 columns

In [26]:
```
X_train, X_test, Y_train,Y_test = train_test_split(X,Y,test_size = 0.20, randc
```

In [27]:
```
from sklearn.linear_model import LinearRegression

# creating a Linear regression object
lr = LinearRegression()
lr
```

Out[27]:
```
LinearRegression()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [28]:
```
lr.fit(X_train, Y_train)
```

Out[28]:
```
LinearRegression()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [31]:
```
from sklearn import metrics
```

In [32]:
```
print('Mean Absolute Error: ',metrics.mean_absolute_error(Y_predict,Y_test))
```

```
Mean Absolute Error:  1.3617813502090275
```

In [34]:
```
print('Root Mean Squared Error:',np.sqrt(metrics.mean_squared_error(Y_predict,
```

```
Root Mean Squared Error: 2.0981225634956804
```

In [35]:
```
print('R-Squared: ',metrics.r2_score(Y_predict,Y_test))
```

```
R-Squared:  0.8330284237692487
```

Got 83% Accuracy

In [42]:
```
x=df[['TV','Radio','Newspaper']]
y=df['Sales']

x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.65, random_sta
```

In [43]:
```
model=LinearRegression()

model.fit(x_train,y_train)

y_predict=model.predict(x_test)
```

In [44]:
```
print(model.intercept_)
print(model.coef_)
```

```
2.848845981475902
[0.04441066 0.19656498 0.00357552]
```

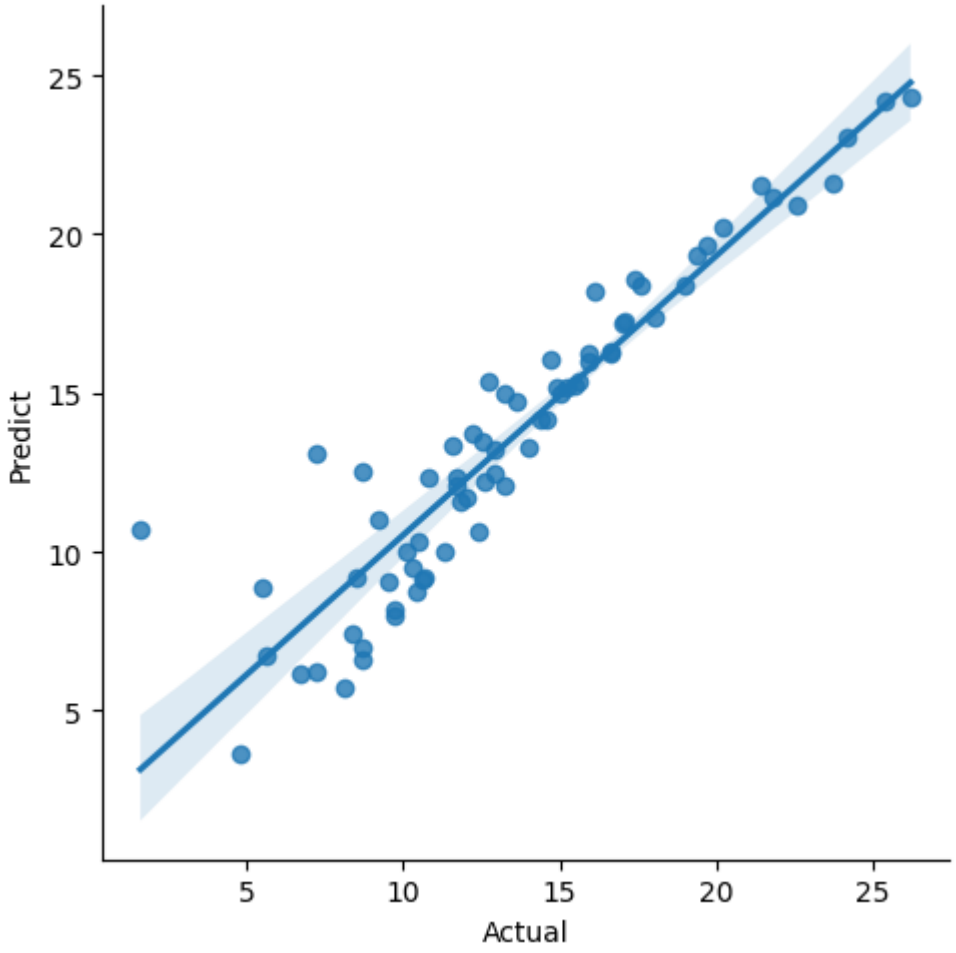
In [45]:
```
act_predict=pd.DataFrame({
    'Actual':y_test.values.flatten(),
    'Predict':y_predict.flatten()
})
act_predict.head(20)
```

Out[45]:

	Actual	Predict
0	11.3	10.017078
1	8.4	7.415322
2	8.7	7.005491
3	25.4	24.212379
4	11.7	12.066605
5	8.7	6.571338
6	7.2	13.115410
7	13.2	14.977876
8	9.2	11.040877
9	16.6	16.255128
10	24.2	23.042256
11	10.6	9.133528
12	10.5	10.344301
13	15.6	15.370656
14	11.8	11.569839
15	13.2	12.081157
16	17.4	18.576689
17	1.6	10.695014
18	14.7	16.040033
19	17.0	17.208418

In [46]:
```
sns.lmplot(data=act_predict,x='Actual',y="Predict")
```

Out[46]:
```
<seaborn.axisgrid.FacetGrid at 0x7f0edef0f790>
```

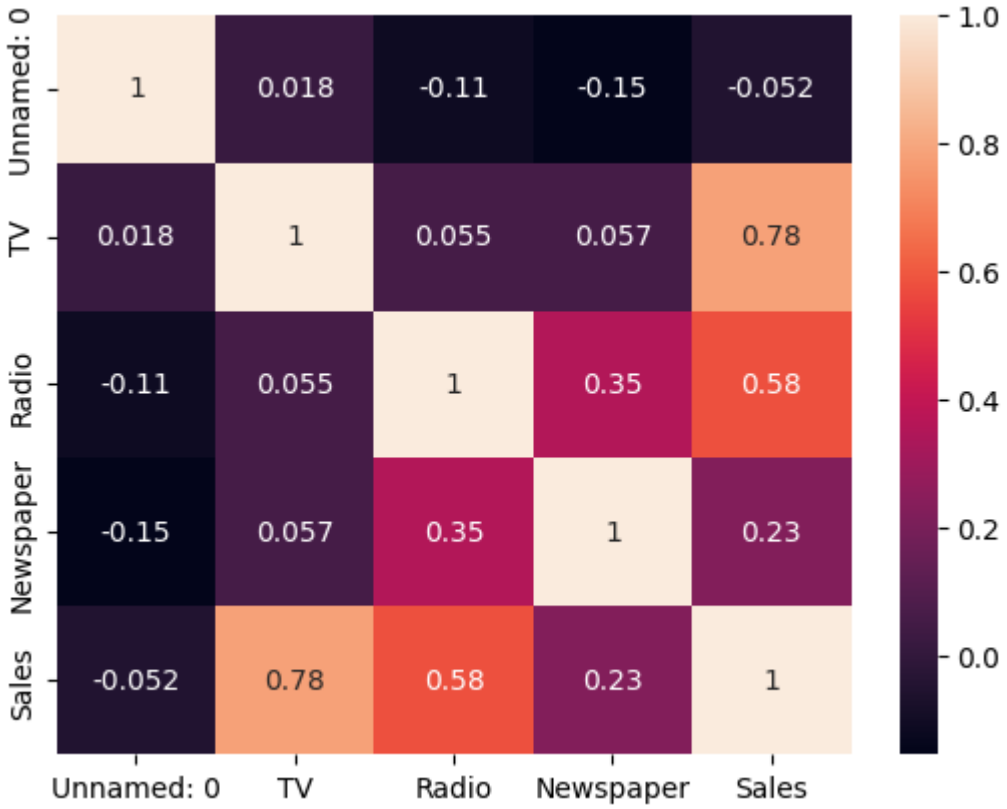


In [47]:
```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

```
print("Mean_absolute_error:",mean_absolute_error(y_test,y_predict))
print("Mean_squared_error:",mean_squared_error(y_test,y_predict))
print("Squre_Mean_absolute_error:",np.sqrt(mean_absolute_error(y_test,y_predict)))
print("r2_score:",r2_score(y_test,y_predict))
```

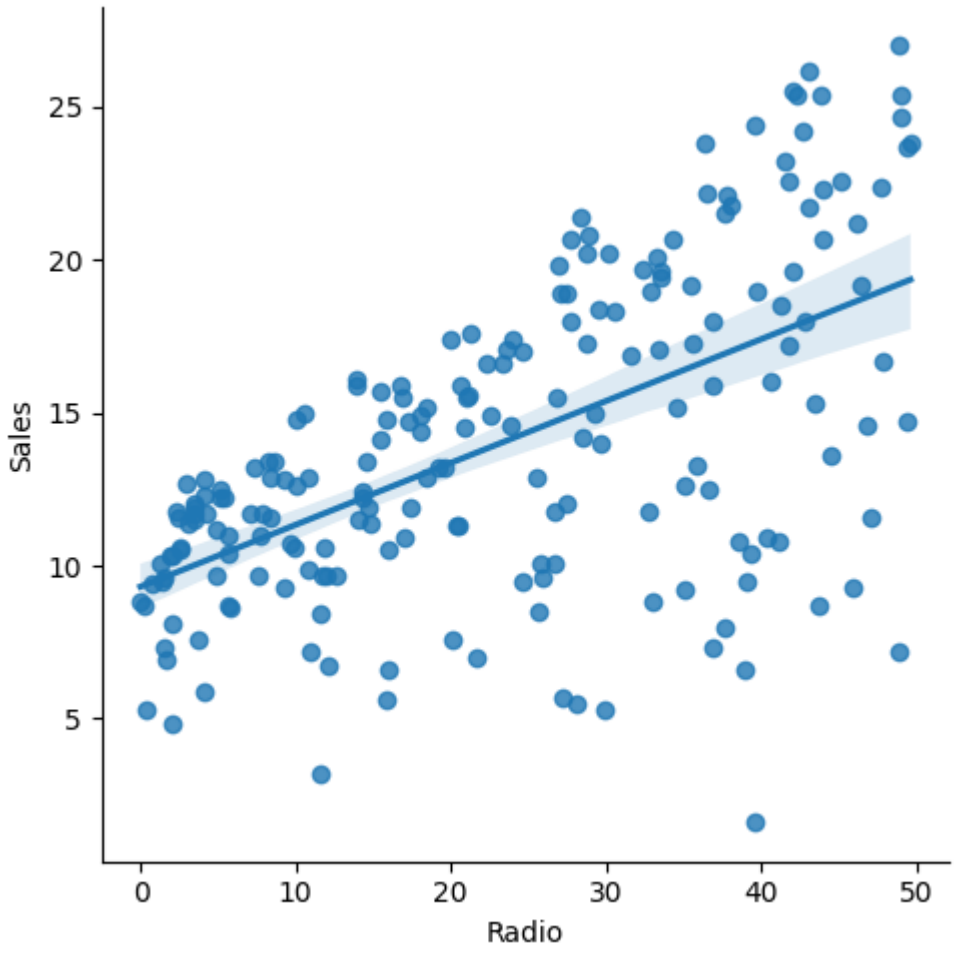
In [38]:
```
sns.heatmap(df.corr(),annot=True)
```

Out[38]: <Axes: >



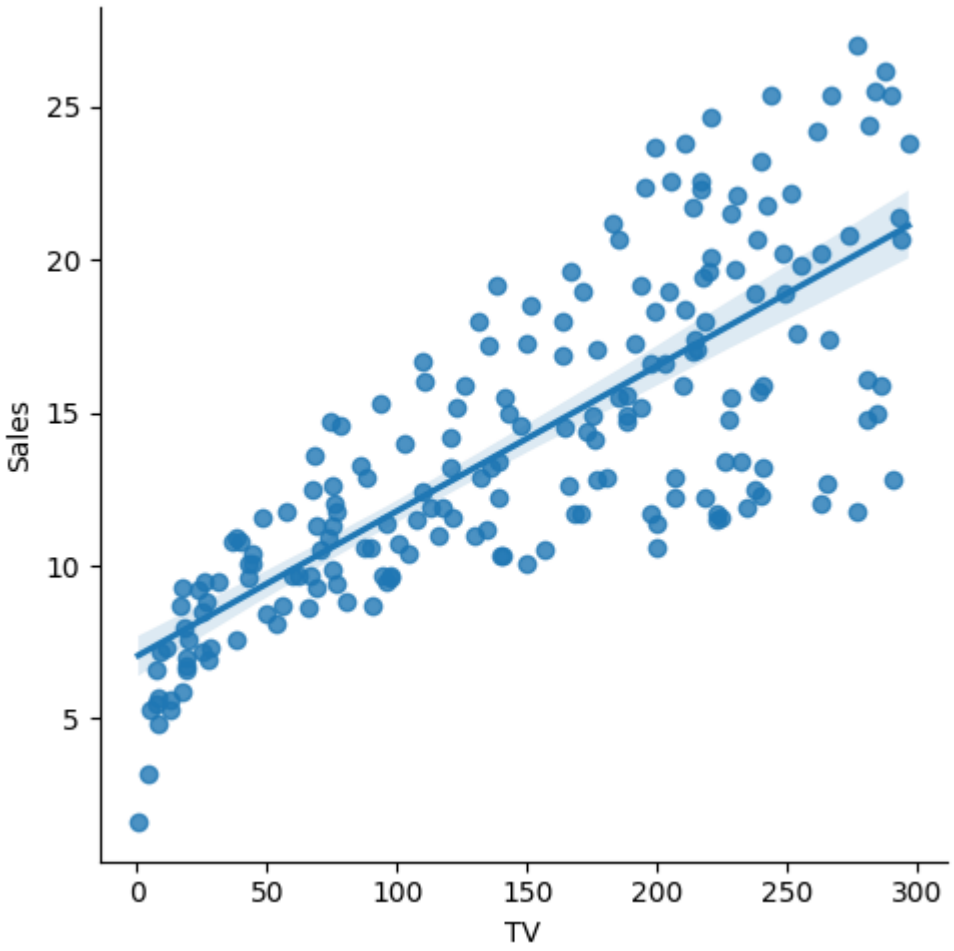
In [39]:
```
sns.lmplot(data=df,x='Radio',y="Sales")
```

Out[39]:
```
<seaborn.axisgrid.FacetGrid at 0x7f0ee3e01720>
```



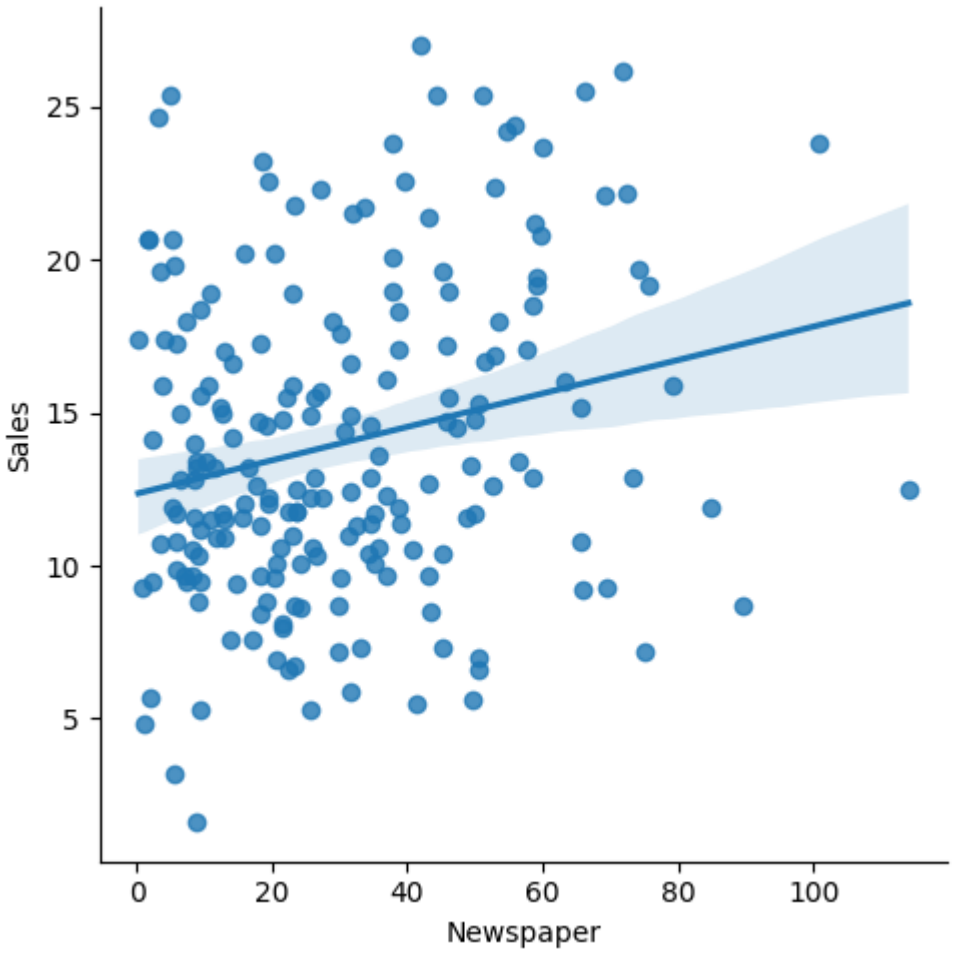
In [40]:
```
sns.lmplot(data=df,x='TV',y="Sales")
```

Out[40]:
```
<seaborn.axisgrid.FacetGrid at 0x7f0ee11d7ee0>
```



In [41]:
```
sns.lmplot(data=df,x='Newspaper',y="Sales")
```

Out[41]:
```
<seaborn.axisgrid.FacetGrid at 0x7f0edefbca30>
```



In [50]:
```
boxplot = df.boxplot(figsize = (5,5), rot = 90, fontsize= '8', grid = False)
```