```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_percentage_error,r2_score
df=pd.read_csv('/content/drive/MyDrive/advertising.csv.xls')
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.head()
```

	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

```
df.tail()
```

	TV	Radio	Newspaper	Sales
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

```
df.columns
```

```
Index(['TV', 'Radio', 'Newspaper', 'Sales'], dtype='object')
```

```
df.dtypes
```

```
TV           float64
Radio        float64
Newspaper    float64
Sales        float64
dtype: object
```

```
df.isna().sum()
```

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

```
x=df.iloc[:,:-1]
y=df.iloc[:,-1]
```

x

	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

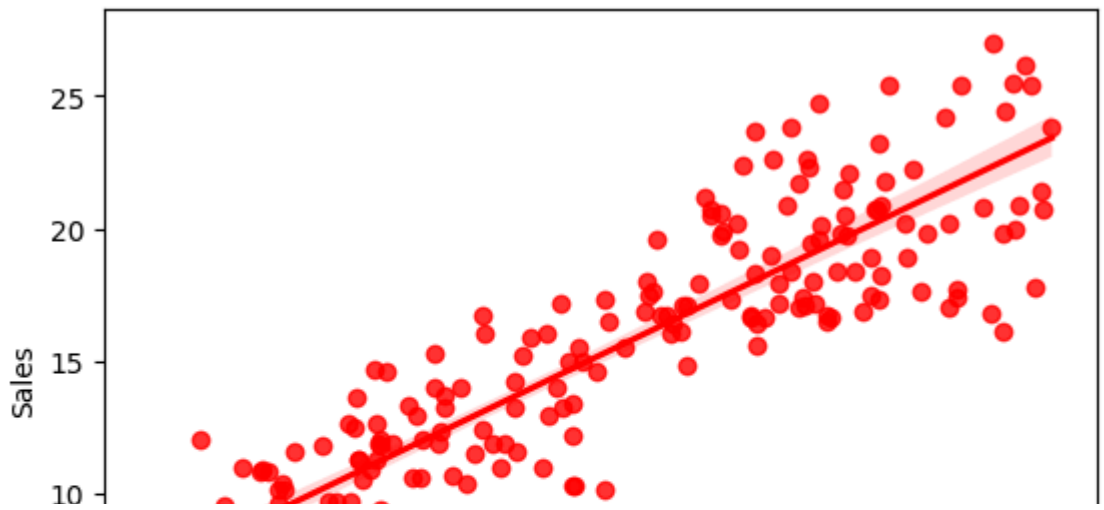
200 rows × 3 columns

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
```

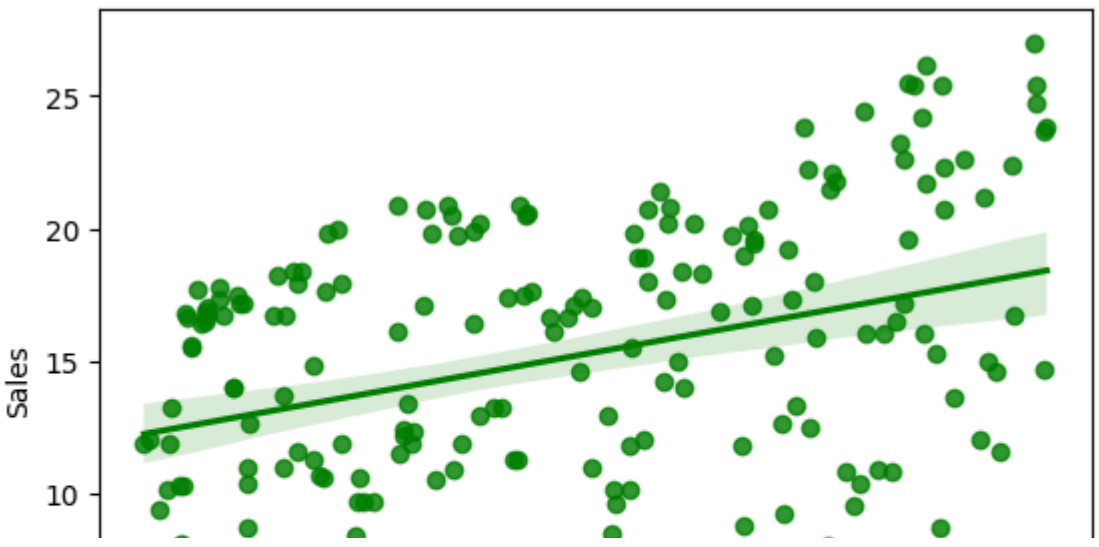
```
#regplot
import seaborn as sns
sns.regplot(x=df['TV'],y=y,color='r')
```

```
<Axes: xlabel='TV', ylabel='Sales'>
```



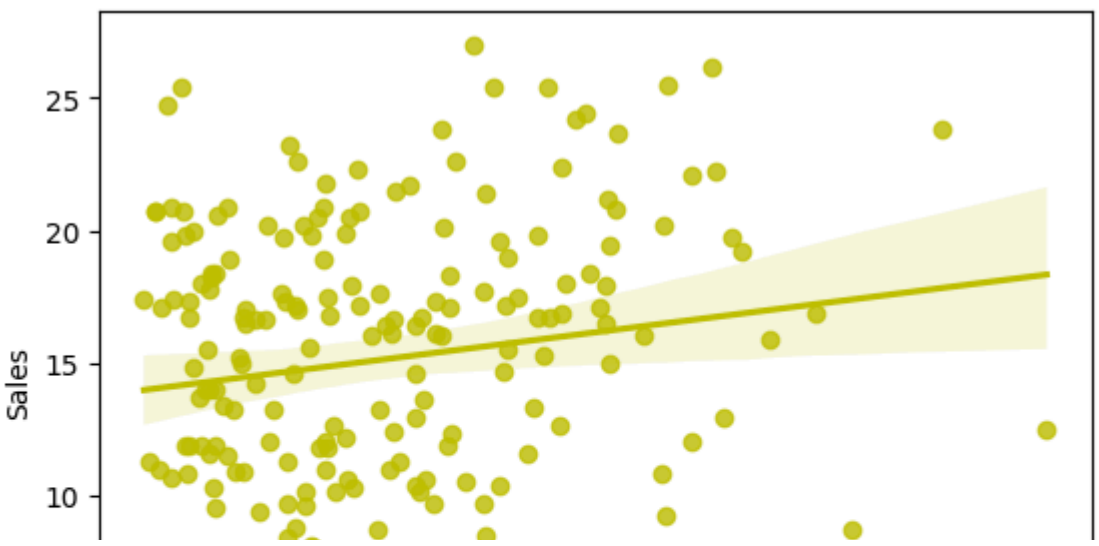

```
sns.regplot(x=df['Radio'],y=y,color='g')
```

```
<Axes: xlabel='Radio', ylabel='Sales'>
```




```
sns.regplot(x=df['Newspaper'],y=y,color='y')
```

```
<Axes: xlabel='Newspaper', ylabel='Sales'>
```




```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=42)
x_train
```

	TV	Radio	Newspaper
169	284.3	10.6	6.4
97	184.9	21.0	22.0
31	112.9	17.4	38.6
12	23.8	35.1	65.9
35	290.7	4.1	8.5
...	...	...	...
106	25.0	11.0	29.7
14	204.1	32.9	46.0
92	217.7	33.5	59.0
179	165.6	10.0	17.6
102	280.2	10.1	21.4

140 rows × 3 columns

```
model=LinearRegression()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
y_pred
```

```
array([17.15991908, 20.53369503, 23.68914396,  9.5191455 , 21.60736836,
       12.78101318, 21.08636345,  8.76054246, 17.11499951, 16.68789636,
        8.97584663,  8.57645026, 18.33212325,  8.17863567, 12.64605571,
       14.94486946,  8.34939536, 17.83858948, 11.12172174, 20.37740648,
       20.9483297 , 13.04035779, 11.01360656, 22.51142595,  9.40369784,
        7.98591291, 20.86943368, 13.77882255, 10.83407064,  8.00419229,
       15.88597618, 10.7027424 , 20.9521718 , 10.84679243, 21.50720813,
       21.07347295, 12.22673775, 22.85273767, 12.57698182,  6.54597206,
       11.93411853, 15.23490068, 10.07411153,  9.52159696, 17.11786382,
        7.28032677, 10.49404864, 15.24356754, 11.20742176, 11.78392665,
       14.01472163, 14.59884572, 10.82722434,  9.55839415,  9.03749681,
       12.51183313, 10.52551021, 25.01900824,  7.99334943, 15.73916263])
```

```
y_test
```

```
95     16.9
15     22.4
30     21.4
158     7.3
128    24.7
115    12.6
69     22.3
170     8.4
174    16.5
45     16.1
66     11.0
182     8.7
165    16.9
78      5.3
186    10.3
177    16.7
56      5.5
152    16.6
82     11.3
68     18.9
124    19.7
16     12.5
148    10.9
93     22.2
65     11.3
60      8.1
84     21.7
67     13.4
125    10.6
132     5.7
9      15.6
18     11.3
55     23.7
75      8.7
150    16.1
104    20.7
135    11.6
137    20.8
164    11.9
76      6.9
79     11.0
197    14.8
38     10.1
24      9.7
122    16.6
195     7.6
29     10.5
19     14.6
143    10.4
86     12.0
114    14.6
173    16.7
5       7.2
126     6.6
117     9.4
73     11.0
140    10.9
98     25.4
```

```
df1=pd.DataFrame({'Actual_value': y_test,'Predicted_value' :y_pred,'Difference':y_pred-y_test})
df1
```

	Actual_value	Predicted_value	Difference
95	16.9	17.159919	0.259919
15	22.4	20.533695	-1.866305
30	21.4	23.689144	2.289144
158	7.3	9.519146	2.219146
128	24.7	21.607368	-3.092632
115	12.6	12.781013	0.181013
69	22.3	21.086363	-1.213637
170	8.4	8.760542	0.360542
174	16.5	17.115000	0.615000
45	16.1	16.687896	0.587896
66	11.0	8.975847	-2.024153
182	8.7	8.576450	-0.123550
165	16.9	18.332123	1.432123
78	5.3	8.178636	2.878636
186	10.3	12.646056	2.346056
177	16.7	14.944869	-1.755131
56	5.5	8.349395	2.849395
152	16.6	17.838589	1.238589
82	11.3	11.121722	-0.178278
68	18.9	20.377406	1.477406
124	19.7	20.948330	1.248330
16	12.5	13.040358	0.540358
148	10.9	11.013607	0.113607
93	22.2	22.511426	0.311426
65	11.3	9.403698	-1.896302
60	8.1	7.985913	-0.114087
84	21.7	20.869434	-0.830566
67	13.4	13.778823	0.378823
125	10.6	10.834071	0.234071
132	5.7	8.004192	2.304192
9	15.6	15.885976	0.285976

```
print(model.predict([[310.3,12.3,21.4]]))
```

```
[22.80536855]
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but L
  warnings.warn(
```

```
print("slope is")
list(zip(x,model.coef_))
```

```
slope is
[('TV', 0.05358869132706914),
 ('Radio', 0.10270676778771287),
 ('Newspaper', 0.007931667677316324)]
```

```
print("constant is",model.intercept_)
```

```
constant is 4.743766701589685
```

197	14.8	15.234901	0.434901

```
#perfomance evaluation


print("MAPE IS",mean_absolute_percentage_error(y_test,y_pred))
print("R2 Score is",r2_score(y_test,y_pred))
```

```
MAPE IS 0.10536440823029307
R2 Score is 0.9091484341849799
```

143	10.4	11.207422	0.807422
86	12.0	11.783927	-0.216073
114	14.6	14.014722	-0.585278
173	16.7	14.598846	-2.101154
5	7.2	10.827224	3.627224
126	6.6	9.558394	2.958394