```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#importing dataset
df=pd.read_csv('/content/advertising.csv')
```

```
df.head(10)
```

	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
5	8.7	48.9	75.0	7.2
6	57.5	32.8	23.5	11.8
7	120.2	19.6	11.6	13.2
8	8.6	2.1	1.0	4.8
9	199.8	2.6	21.2	15.6

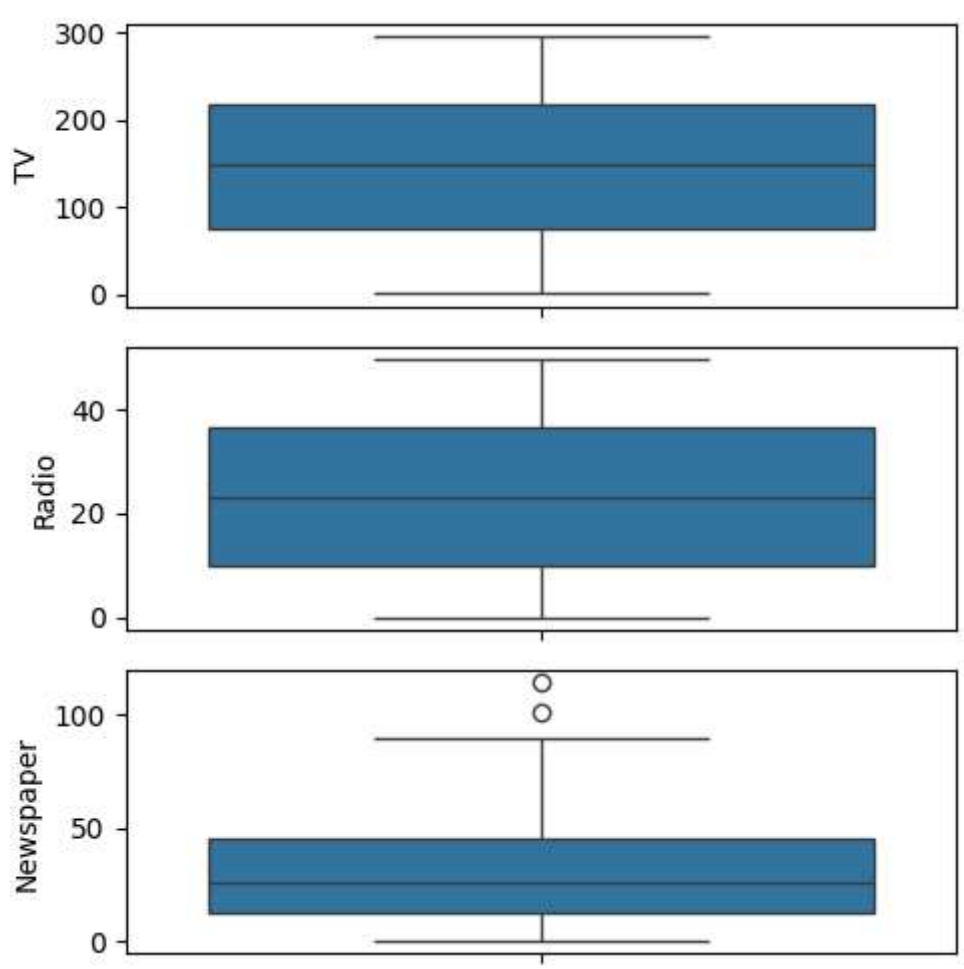
```
df.shape
```

```
(200, 4)
```

```
df.describe()
```

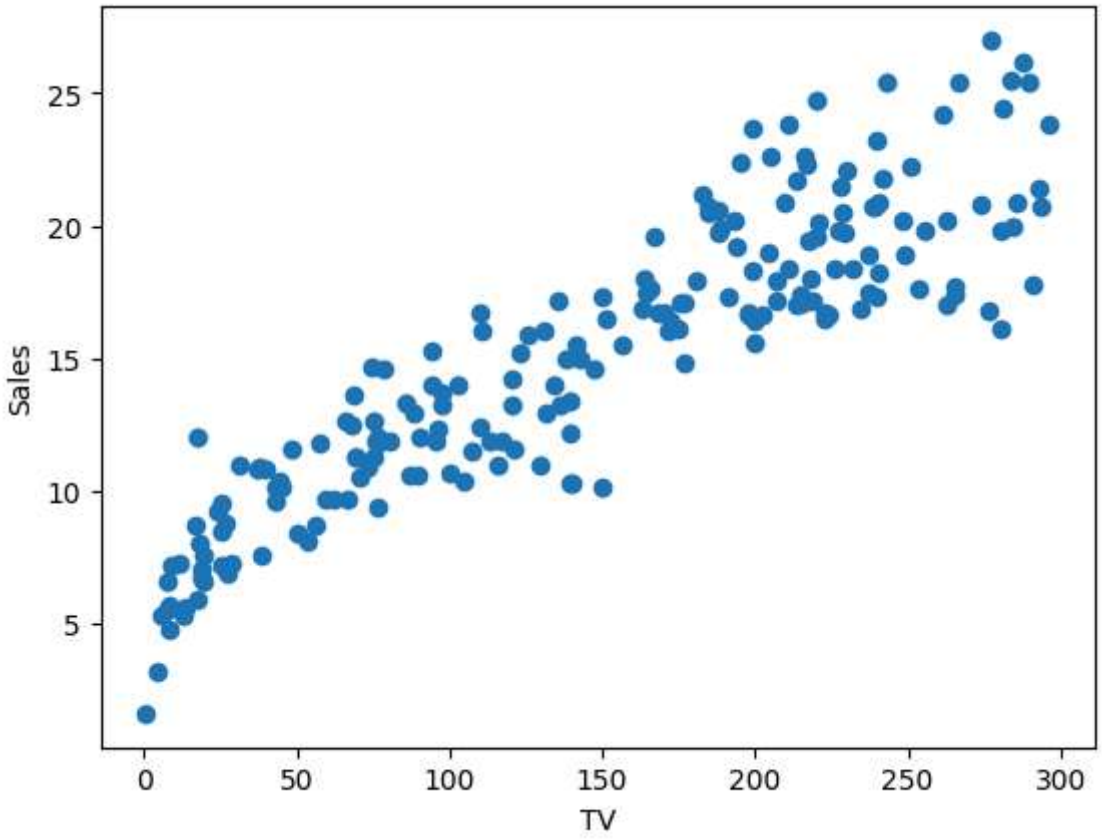
	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	15.130500
std	85.854236	14.846809	21.778621	5.283892
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	11.000000
50%	149.750000	22.900000	25.750000	16.000000
75%	218.825000	36.525000	45.100000	19.050000
max	296.400000	49.600000	114.000000	27.000000

```
fig, axs = plt.subplots(3, figsize = (5,5))
plt1=sns.boxplot(df['TV'],ax=axs[0])
plt2=sns.boxplot(df['Radio'],ax=axs[1])
plt3=sns.boxplot(df['Newspaper'],ax=axs[2])
plt.tight_layout()
```



```
x=df['TV']
y=df['Sales']
plt.scatter(x,y)
plt.xlabel('TV')
plt.ylabel('Sales')
```

```
Text(0, 0.5, 'Sales')
```

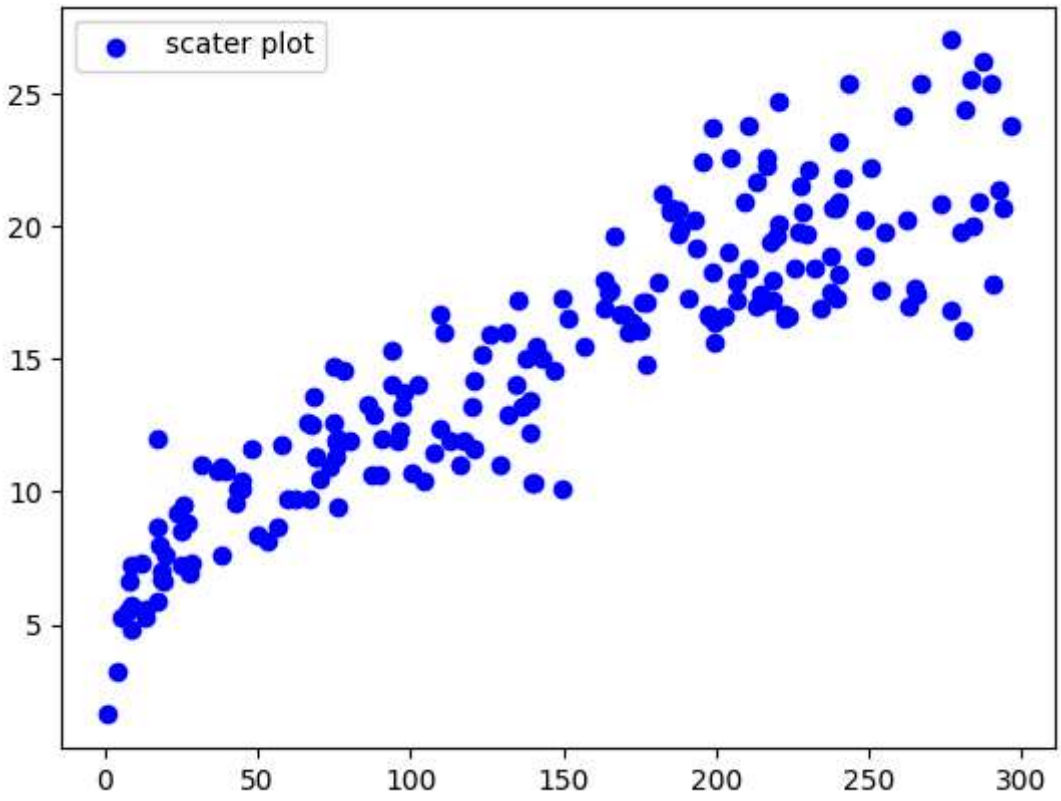



```
df.info()
```

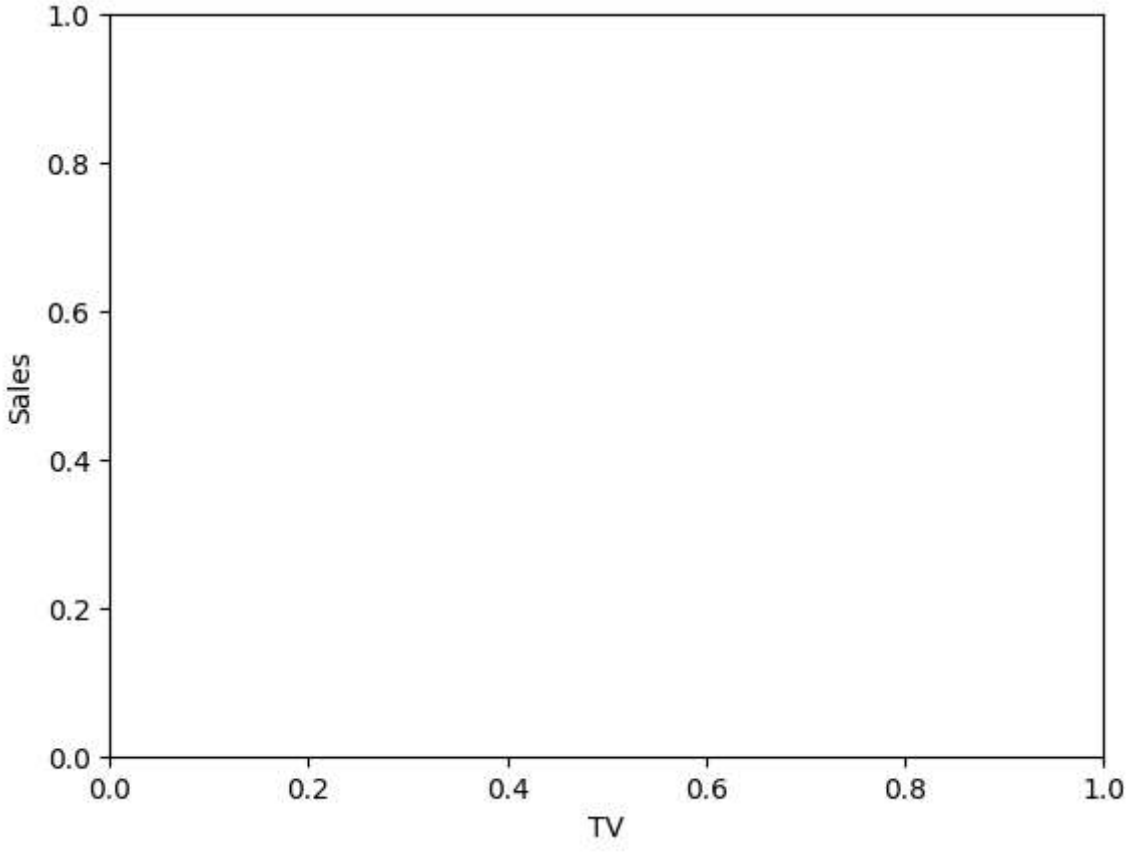
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   TV         200 non-null    float64
 1   Radio      200 non-null    float64
 2   Newspaper  200 non-null    float64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

```
x=df['TV']
y=df['Sales']
plt.scatter(x,y,color='blue',label='scater plot')
plt.legend()
plt.show()
plt.xlabel('TV')
```

```
plt.ylabel('Sales')
```




```
Text(0, 0.5, 'Sales')
```



```
print(x.shape)
print(y.shape)
```

```
(200,)
(200,)
```

```
X=x.values.reshape(-1,1)
Y=y.values.reshape(-1,1)
print(X)
print(Y)
```

```
[[230.1]
 [ 44.5]
 [ 17.2]
 [151.5]
 [180.8]
 [  8.7]
 [ 57.5]
 [120.2]
 [  8.6]
 [199.8]
 [ 66.1]
 [214.7]
 [ 23.8]
 [ 97.5]
 [204.1]
 [195.4]
 [ 67.8]
 [281.4]
 [ 69.2]
 [147.3]
 [218.4]
 [237.4]
 [ 13.2]
 [228.3]
 [ 62.3]
 [262.9]
 [142.9]
 [240.1]
 [248.8]
 [ 70.6]
 [292.9]
 [112.9]
 [ 97.2]
 [265.6]
 [ 95.7]
 [290.7]
 [266.9]
 [ 74.7]
 [ 43.1]
 [228. ]
 [202.5]
 [177. ]
 [293.6]
 [206.9]
```

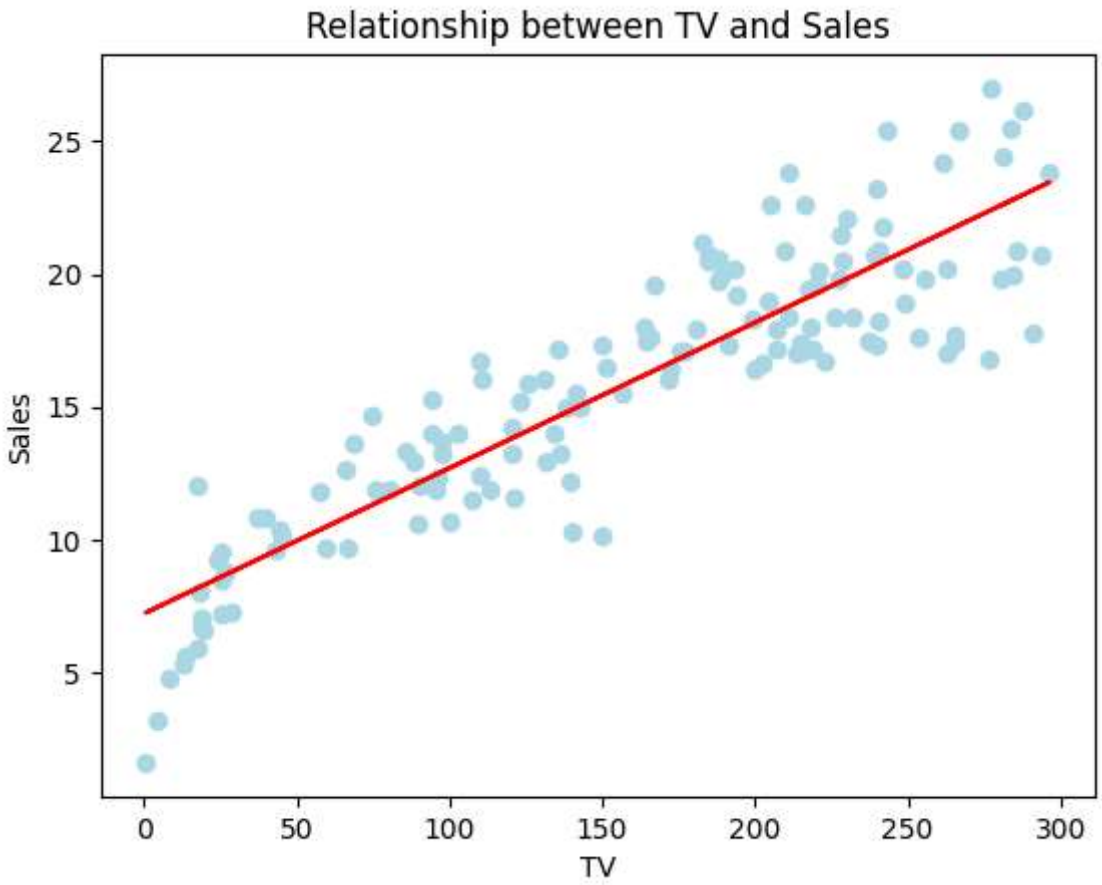
```
[ 25.1]
[175.1]
[ 89.7]
[239.9]
[227.2]
[ 66.9]
[199.8]
[100.4]
[216.4]
[182.6]
[262.7]
[198.9]
[  7.3]
[136.2]
```

```
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)
print(X_train.shape)
print(Y_train.shape)
print(X_test.shape)
print(Y_test.shape)
```

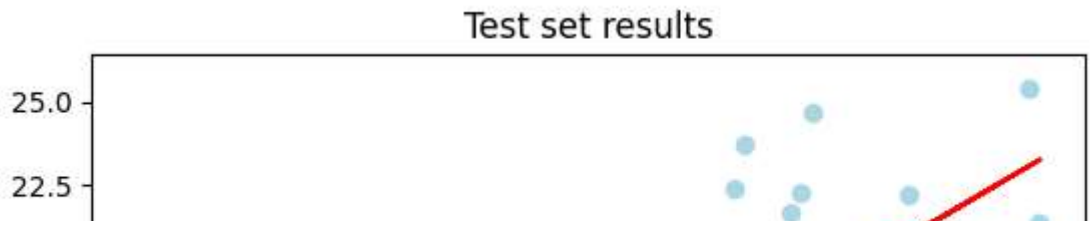
```
(140, 1)
(140, 1)
(60, 1)
(60, 1)
```

```
from sklearn.linear_model import LinearRegression
lm=LinearRegression()
lm.fit(X_train,Y_train)
y_pred=lm.predict(X_test)
```

```
plt.scatter(X_train,Y_train,color='lightblue')
plt.plot(X_train,lm.predict(X_train),color='red')
plt.title('Relationship between TV and Sales')
plt.xlabel('TV')
plt.ylabel('Sales')
plt.show()
```

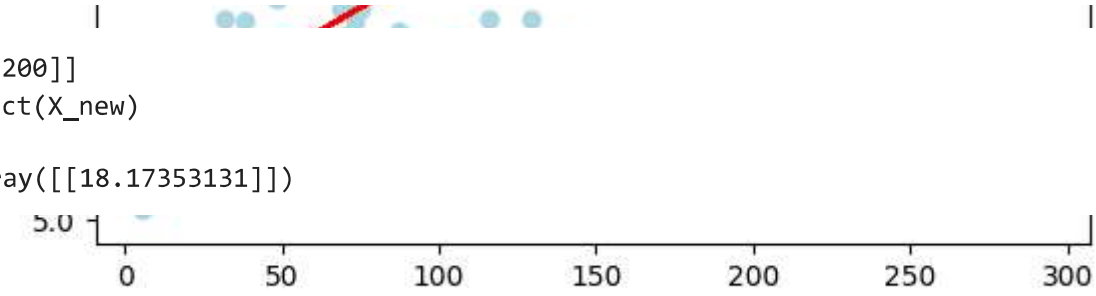



```
plt.scatter(X_test,Y_test,color='lightblue')
plt.plot(X_test,lm.predict(X_test),color='red')
plt.title('Test set results')
plt.xlabel('TV')
plt.ylabel('Sales')
plt.show()
```




```
slope=lm.coef_
intercept=lm.intercept_,
print("estimated model slope",slope)
print("estimated model intercept",intercept)
```

```
estimated model slope [[0.05483488]]
estimated model intercept (array([7.20655455]),)
```



```
X_new=[[200]]
lm.predict(X_new)
```

```
array([[18.17353131]])
```