



Tutorial No. 3

 jupyter

Untitled1 Last Checkpoint: 6 minutes ago

File Edit View Run Kernel Settings Help



```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

UsageError: Line magic function `%` not found.
```

```
[2]: df=pd.read_csv("Advertising.csv")
```

```
[3]: print(df.shape)

(200, 5)
```

```
[4]: print(df.head())
```

	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

```
[5]: print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Unnamed: 0  200 non-null   int64
1   TV          200 non-null   float64
2   radio       200 non-null   float64
3   newspaper   200 non-null   float64
4   sales       200 non-null   float64
dtypes: float64(4), int64(1)
memory usage: 7.9 KB
```

```
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Unnamed: 0    200 non-null    int64
1   TV             200 non-null    float64
2   radio          200 non-null    float64
3   newspaper      200 non-null    float64
4   sales          200 non-null    float64
dtypes: float64(4), int64(1)
memory usage: 7.9 KB
None
```

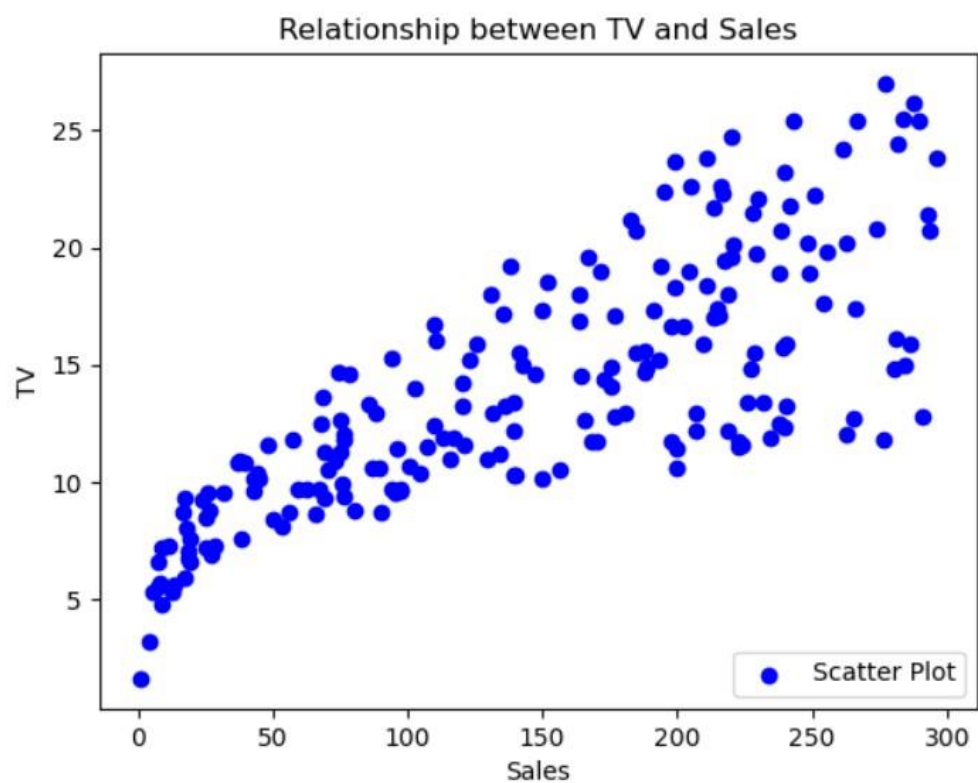
```
[6]: print(df.describe())
```

```
count      Unnamed: 0      TV      radio      newspaper      sales
mean    200.000000  100.500000  23.264000  30.554000  14.022500
std       57.879185   85.854236  14.846809  21.778621   5.217457
min        1.000000    0.700000   0.000000   0.300000   1.600000
25%       50.750000   74.375000   9.975000  12.750000  10.375000
50%      100.500000  149.750000  22.900000  25.750000  12.900000
75%      150.250000  218.825000  36.525000  45.100000  17.400000
max      200.000000  296.400000  49.600000 114.000000  27.000000
```

```
[7]: A=df['TV'].values
      B=df['radio'].values
      C=df['newspaper'].values
      D=df['sales'].values
```

```
[8]: plt.scatter(A,D,color='blue',label='Scatter Plot')
      plt.title('Relationship between TV and Sales')
      plt.xlabel('Sales')
      plt.ylabel('TV')
      plt.legend(loc=4)
      plt.show()
```

```
[8]: plt.scatter(A,D,color='blue',label='Scatter Plot')
plt.title('Relationship between TV and Sales')
plt.xlabel('Sales')
plt.ylabel('TV')
plt.legend(loc=4)
plt.show()
```



```
[9]: print(A.shape)
print(D.shape)
```

```
[9]: print(A.shape)
      print(D.shape)
```

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

```
[10]: A=A.reshape(-1,1)
      D=D.reshape(-1,1)
```

```
[11]: print(A.shape)
      print(D.shape)
```

```
(200, 1)
(200, 1)
```

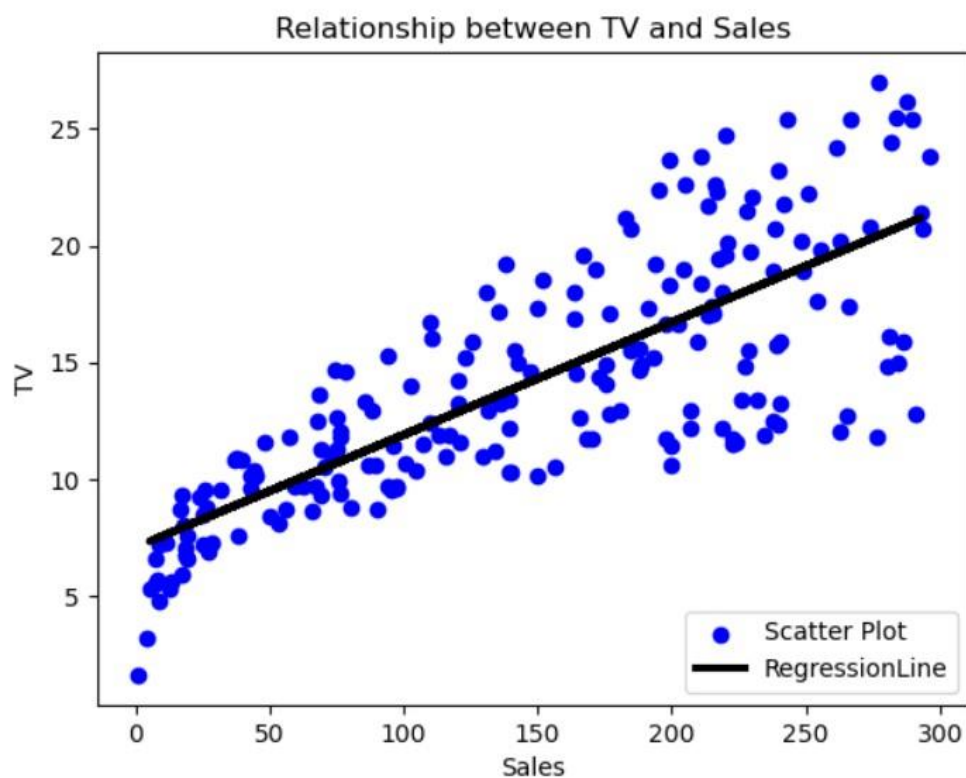
```
[12]: from sklearn.model_selection import train_test_split
      A_train,A_test,D_train,D_test=train_test_split(A,D,test_size=0.33,random_state=42)
      print(A_train.shape)
      print(D_train.shape)
      print(A_test.shape)
      print(D_test.shape)
```

```
(134, 1)
(134, 1)
(66, 1)
(66, 1)
```

```
[13]: from sklearn.linear_model import LinearRegression
      lm=LinearRegression()
      lm.fit(A_train,D_train)
      D_pred=lm.predict(A_test)
```

```
[14]: plt.scatter(A,D,color='blue',label='Scatter Plot')
      plt.plot(A_test,D_pred,color='black',linewidth=3,label='RegressionLine')
      plt.title('Relationship between TV and Sales')
      plt.xlabel('Sales')
```

```
[14]: plt.scatter(A,D,color='blue',label='Scatter Plot')
plt.plot(A_test,D_pred,color='black',linewidth=3,label='RegressionLine')
plt.title('Relationship between TV and Sales')
plt.xlabel('Sales')
plt.ylabel('TV')
plt.legend(loc=4)
plt.show()
```



[]: