In [6]:
```
import pandas as pd
```

In [7]:
```
from pandas import *
```

In [11]:
```
data = pd.read_csv("Advertising.csv")
```

In [12]:
```
data
```

Out[12]:

	ID	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 [13]:
```
data = pd.read_csv("Advertising.csv", index_col=0)
```

In [14]:
```
data
```

Out[14]:

ID	TV	Radio	Newspaper	Sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9
...	...	...	...	...
196	38.2	3.7	13.8	7.6
197	94.2	4.9	8.1	9.7
198	177.0	9.3	6.4	12.8
199	283.6	42.0	66.2	25.5
200	232.1	8.6	8.7	13.4

200 rows × 4 columns

In [15]:
```
data.head()
```

Out[15]:

ID	TV	Radio	Newspaper	Sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9

In [17]:
```
data
```

Out[17]:

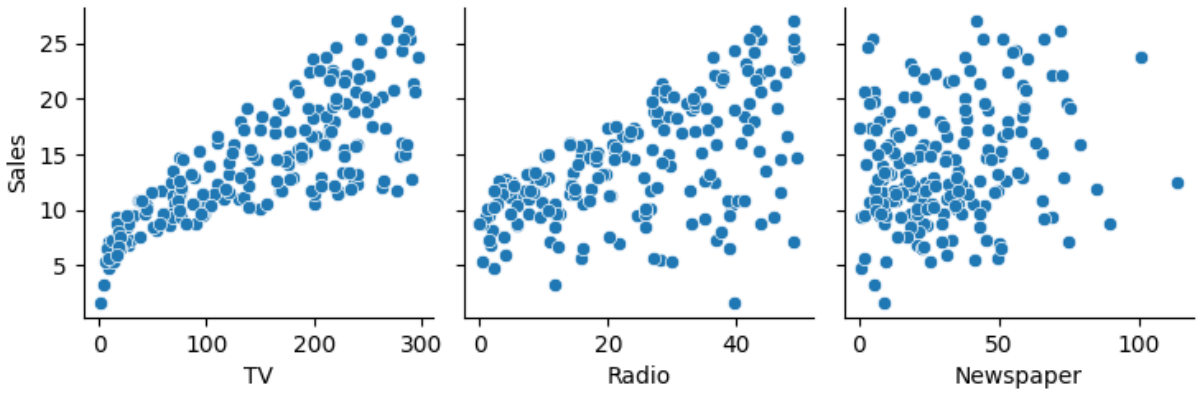
ID	TV	Radio	Newspaper	Sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9
...	...	...	...	...
196	38.2	3.7	13.8	7.6
197	94.2	4.9	8.1	9.7
198	177.0	9.3	6.4	12.8
199	283.6	42.0	66.2	25.5
200	232.1	8.6	8.7	13.4

200 rows × 4 columns

In [20]:
```
#Import libraries for plotting
import seaborn as sns
%matplotlib inline
```

In [22]:
```
# pairplot your data x&y
sns.pairplot(data,x_vars=['TV','Radio','Newspaper'], y_vars='Sales')
```

Out[22]:
```
<seaborn.axisgrid.PairGrid at 0x17d80a210>
```



In [28]:
```
# Extract the features in to x
features_cols = ['TV', 'Radio', 'Newspaper']


#### x = data[['TV', 'Radio', 'Newspaper']] this will have the same resuld
# and they both mean same. so either 1 can be used in python code.
```

In [24]:
```
x= data[features_cols]
```

In [26]:
```
x
```

Out[26]:

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

200 rows × 3 columns

In [27]:
```
x.head()
```

Out[27]:

ID	TV	Radio	Newspaper
1	230.1	37.8	69.2
2	44.5	39.3	45.1
3	17.2	45.9	69.3
4	151.5	41.3	58.5
5	180.8	10.8	58.4

In [29]:
```
# extract the result/response/dependent variable into y

y = data['Sales']
```

In [30]:
```
y
```

Out[30]:
```
ID
1      22.1
2      10.4
3       9.3
4      18.5
5      12.9
       ...
196     7.6
197     9.7
198    12.8
199    25.5
200    13.4
Name: Sales, Length: 200, dtype: float64
```

In [31]:
```
y.head()
```

Out[31]:
```
ID
1    22.1
2    10.4
3     9.3
4    18.5
5    12.9
Name: Sales, dtype: float64
```

In [33]:
```
# import libraries to split data into test and train data

from sklearn.model_selection import train_test_split
```

In [34]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,random_state=1)
```

In [35]:
```
print(x_train.shape)
```

```
(150, 3)
```

In [36]:
```
print(y_train.shape)
```

```
(150,)
```

In [37]:
```
print(x_test.shape)
```

```
(50, 3)
```

In [38]:
```
print(y_test.shape)
```

```
(50,)
```

In [39]:
```
### Now perform linear regression, so to do that import libraries

from sklearn.linear_model import LinearRegression
```

In [40]:
```
lr = LinearRegression()
lr.fit(x_train, y_train)
```

Out[40]:
```
▾  LinearRegression i ?

LinearRegression()
```

In [41]:
```
print(lr.intercept_)
```

```
2.87696662231793
```

In [42]:
```
print(lr.coef_)
```

```
[0.04656457 0.17915812 0.00345046]
```

In [ ]: