In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import r2_score, mean_squared_error
from math import sqrt
import numpy
```

In [2]:
```
df=pd.read_csv('C:/Users/user/Downloads/Advertising.csv')
```

In [3]:
```
df
```

Out[3]:

	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

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

Out[4]:

	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

In [5]:

```
df.tail()
```

Out[5]:

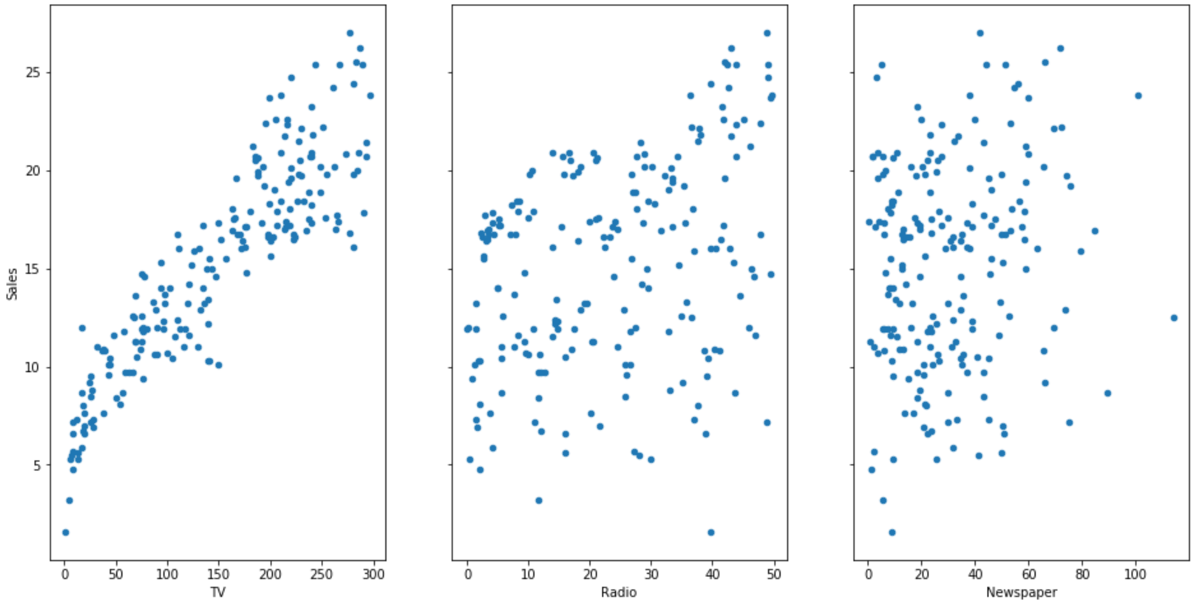
	TV	Radio	Newspaper	Sales
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	14.0
197	177.0	9.3	6.4	14.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	18.4

In [6]:

```
fig, axes=plt.subplots(nrows=1,ncols=3,sharey=True)
df.plot(kind='scatter',x='TV',y='Sales',ax=axes[0],figsize=(16,20))
df.plot(kind='scatter',x='Radio',y='Sales',ax=axes[1],figsize=(16,8))
df.plot(kind='scatter',x='Newspaper',y='Sales',ax=axes[2],figsize=(16,8))
```

Out[6]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2e0744b7fc8>
```



In [7]:

```
x=df['TV']
y=df.Sales
x=numpy.array(x)
y=numpy.array(y)
x=x.reshape(-1,1)
```

In [8]:

```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x,y)
```

Out[8]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=Fals
e)
```

In [9]:

```
print(lr.intercept_)
print(lr.coef_)
```

```
6.974821488229891
[0.05546477]
```

In [10]:

```
dff=pd.DataFrame({'TV':[df.TV.min(),df.TV.max()]})
```

In [11]:

```
pred=lr.predict(x)

pred=numpy.reshape(pred,(-1,1))
print(x.shape)
print(pred.shape)
```

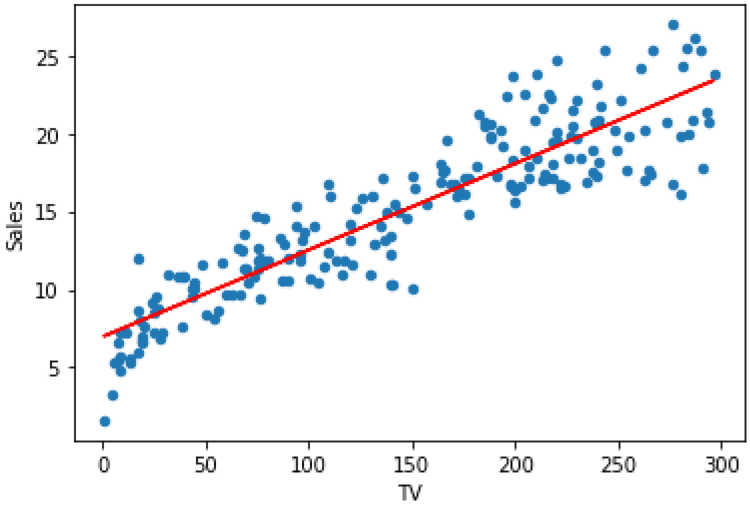
```
(200, 1)
(200, 1)
```

In [12]:

```
df.plot(kind='scatter',x='TV',y='Sales')
plt.plot(x,pred,c='red')
```

Out[12]:

```
[<matplotlib.lines.Line2D at 0x2e07465da88>]
```



In [13]:

```
r2_score(df.Sales,pred)
```

Out[13]:

```
0.8121757029987414
```

In [14]:

```
mean_squared_error(df.Sales,pred)/15.130500
```

Out[14]:

```
0.34484940337696235
```

In [15]:
```
df.describe()
```

Out[15]:

	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

In [16]:
```
df
```

Out[16]:

	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

In [17]:
```
h=df['Sales'].quantile(.25)
j=df['Sales'].quantile(.75)
```

In [18]:
```
i=j-h
i
print(j,h)
h
```

```
19.05 11.0
```

Out[18]:
```
11.0
```

In [19]:

```
df[(df.Sales<j) & (df.Sales>h)].Sales.value_counts
```

Out[19]:

```
<bound method IndexOpsMixin.value_counts of 2      12.0
3      16.5
4      17.9
6      11.8
7      13.2
       ... 
191    11.9
194    17.3
196    14.0
197    14.8
199    18.4
Name: Sales, Length: 98, dtype: float64>
```

In [20]:

```
df['asdf']=pd.cut(df.Sales, 4, labels=["best", "medium", "good","bad"])
```

In [21]:

```
df=df.drop(columns='asdf')
```

In [ ]:

```
```