Multiple+Linear+Regression

October 2, 2024

1 Multiple Linear Regression

[2]:
```
import numpy as np, pandas as pd


import warnings
warnings.filterwarnings('ignore')


data = pd.read_csv("advertising.csv")
data.head()
```

[2]:

	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

[3]:
```
# adding more than 1 X variables into the model

X = data[['TV', 'Radio','Newspaper']]
y = data['Sales']
```

[4]:
```
from sklearn.model_selection import train_test_split
import statsmodels.api as sm
```

[5]:
```
X_train, X_test, y_train,y_test = train_test_split(X, y, test_size = 0.2,␣
 ↪random_state = 50)
```

[6]:
```
X_train_sm = sm.add_constant(X_train)
lr = sm.OLS(y_train, X_train_sm).fit()
```

[7]:
```
lr.params #  co-effs y = m1*TV + m2*Radio + m3*Newspaper + c
```

[7]:
```
const        4.511128
TV           0.055052
```

```
Radio        0.102899
Newspaper    0.004352
dtype: float64
```

[8]:
```
print(lr.summary()) # p values should be less than 0.05
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.902
Model:                            OLS   Adj. R-squared:                  0.900
Method:                 Least Squares   F-statistic:                     478.6
Date:                Wed, 02 Oct 2024   Prob (F-statistic):           1.98e-78
Time:                        04:04:13   Log-Likelihood:                -314.12
No. Observations:                 160   AIC:                             636.2
Df Residuals:                     156   BIC:                             648.5
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          4.5111      0.357     12.647      0.000       3.807       5.216
TV             0.0551      0.002     34.924      0.000       0.052       0.058
Radio          0.1029      0.010     10.307      0.000       0.083       0.123
Newspaper      0.0044      0.007      0.643      0.521      -0.009       0.018
==============================================================================
Omnibus:                       12.356   Durbin-Watson:                   1.873
Prob(Omnibus):                  0.002   Jarque-Bera (JB):               17.504
Skew:                          -0.451   Prob(JB):                     0.000158
Kurtosis:                       4.346   Cond. No.                         455.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[9]:
```
y_pred = lr.predict(X_train_sm)
print(y_pred)
```

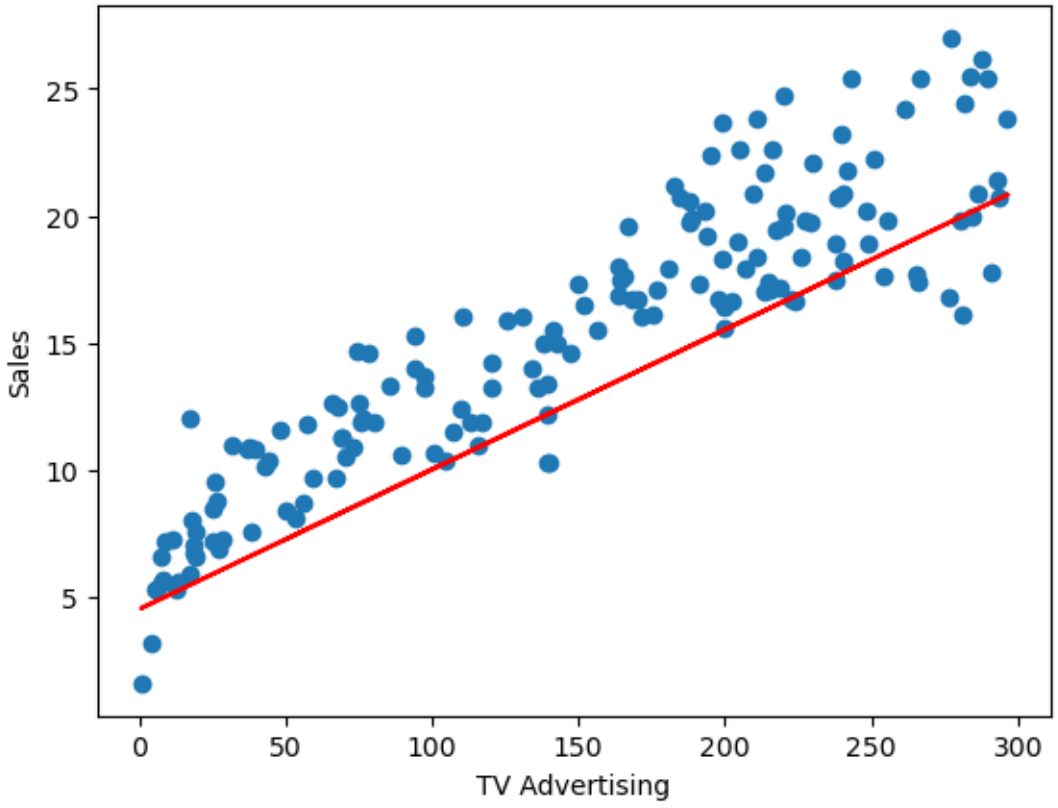
```
170     8.537408
183    25.081085
38      9.783996
153    18.190614
40     18.091245
         …
132     7.781541
33     21.192124
109    21.363227
139    19.214823
```

```
176    21.381838
Length: 160, dtype: float64
```

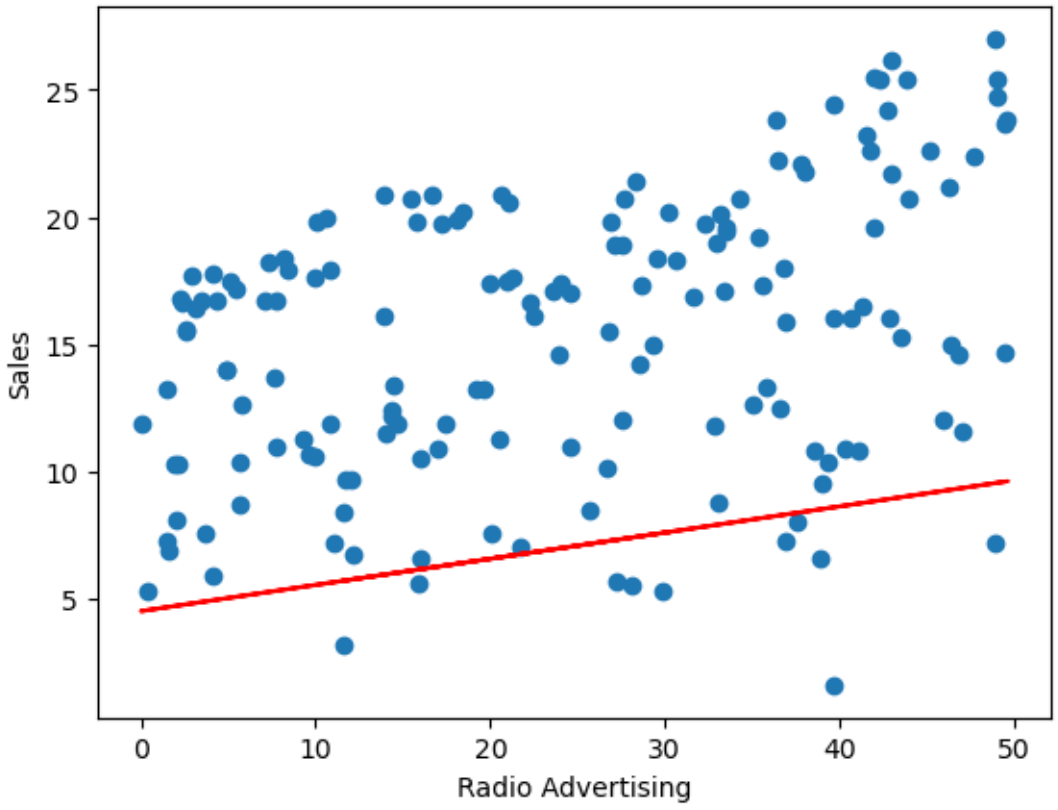
[10]:
```
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

[10]:
```
((160, 3), (40, 3), (160,), (40,))
```

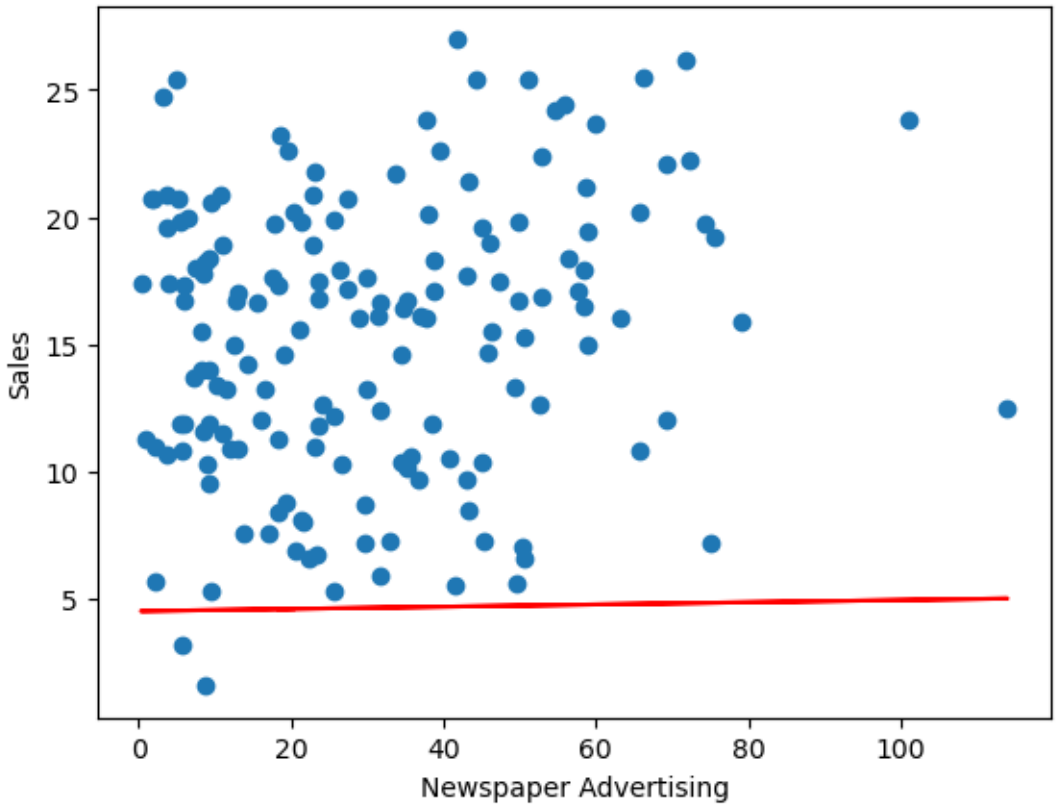
[11]:
```
import matplotlib.pyplot as plt, seaborn as sns


plt.scatter(X_train['TV'], y_train)
plt.plot(X_train['TV'],4.511128 + 0.055052 * X_train['TV'],'r')
plt.xlabel('TV Advertising')
plt.ylabel('Sales')
plt.show()
```



[12]:
```
plt.scatter(X_train['Radio'], y_train)
plt.plot(X_train['Radio'],4.511128 + 0.102899 * X_train['Radio'],'r')
plt.xlabel('Radio Advertising')
plt.ylabel('Sales')
plt.show()
```



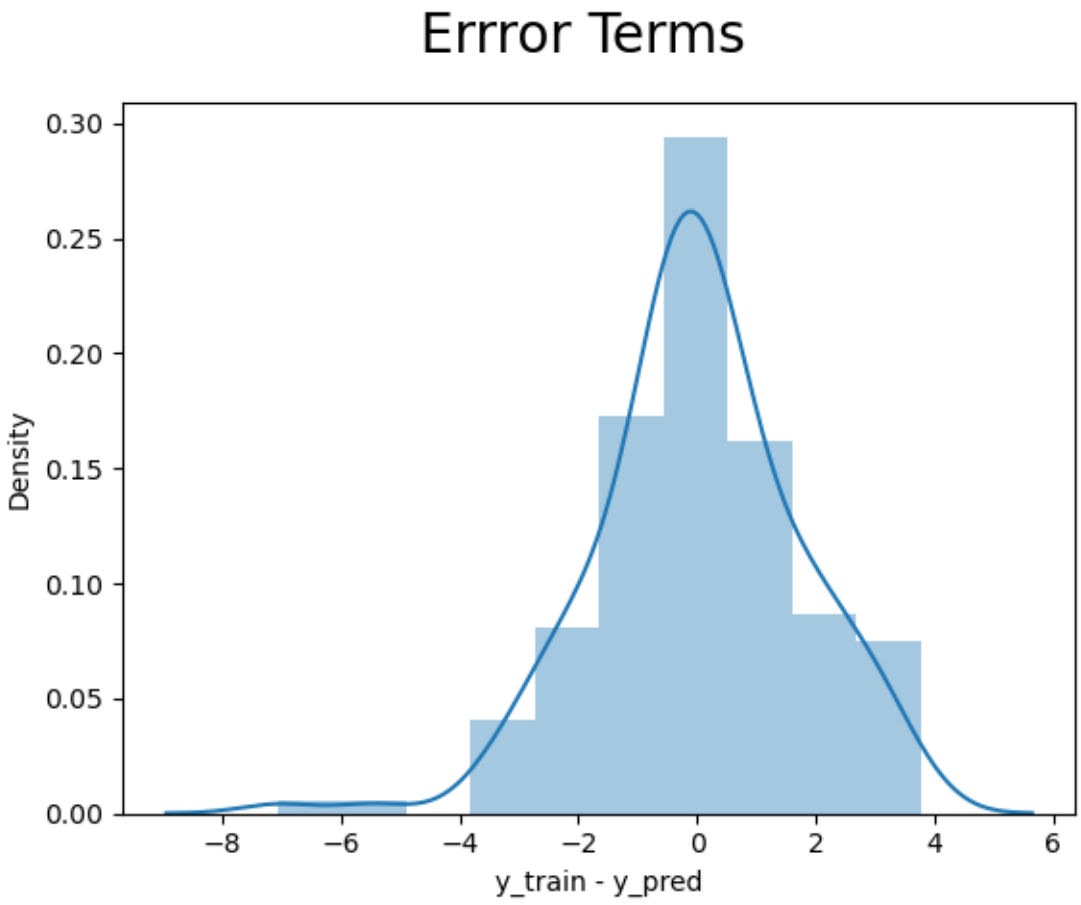
[13]:
```
plt.scatter(X_train['Newspaper'], y_train)
plt.plot(X_train['Newspaper'],4.511128 + 0.004352 * X_train['Newspaper'],'r')
plt.xlabel('Newspaper Advertising')
plt.ylabel('Sales')
plt.show()
```



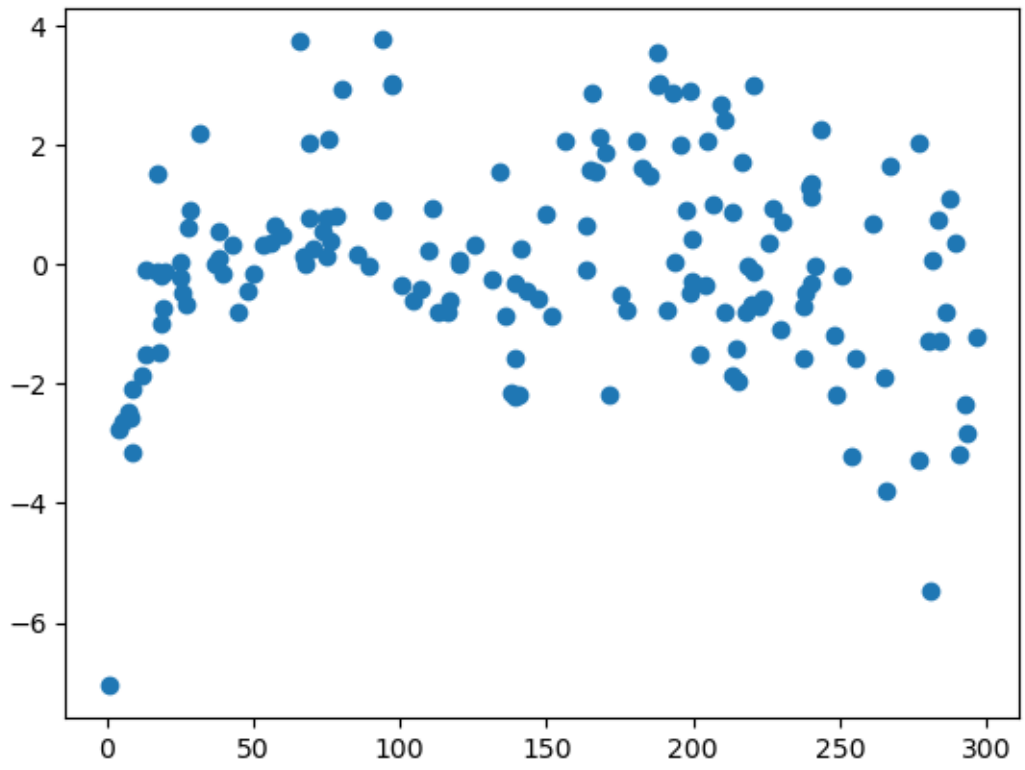
[14]:
```
res = (y_train - y_pred)
print(res)
```

```
170   -0.137408
183    1.118915
38     0.316004
153   -2.190614
40    -1.491245
         …
132   -2.081541
33    -3.792124
109   -1.563227
139    1.485177
176   -1.181838
Length: 160, dtype: float64
```

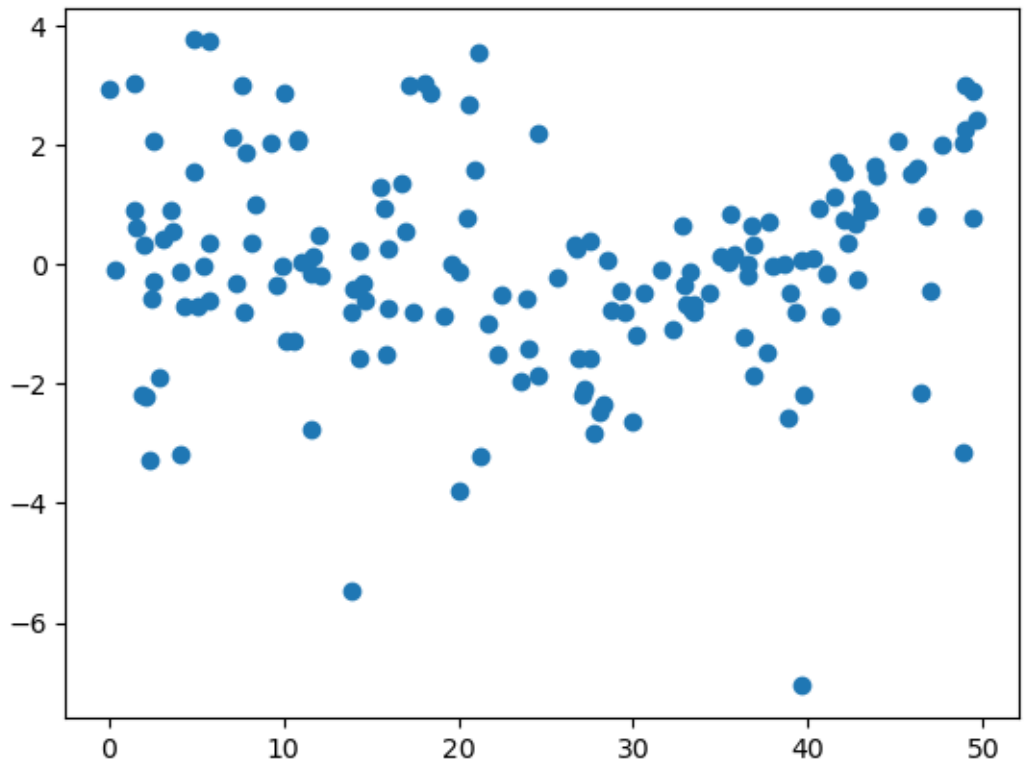
[15]:
```
fig = plt.figure()
sns.distplot(res, bins = 10)
fig.suptitle("Errror Terms", fontsize = 20)
plt.xlabel("y_train - y_pred", fontsize = 10)
plt.show()
```



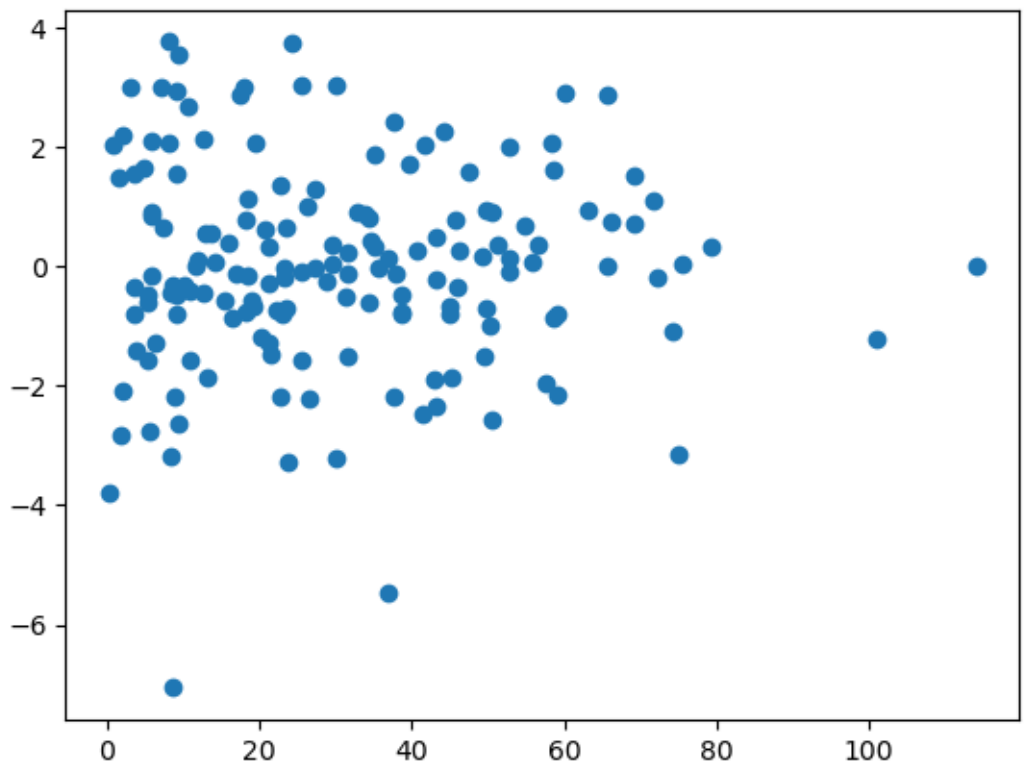
[16]:
```
plt.scatter(X_train['TV'],res)
plt.show()
```



[17]:
```
plt.scatter(X_train['Radio'],res)
plt.show()
```



[18]:
```
plt.scatter(X_train['Newspaper'],res)
plt.show()
```



[19]:
```
X_test_sm = sm.add_constant(X_test)
X_test_sm
```

[19]:

	const	TV	Radio	Newspaper
112	1.0	175.7	15.4	2.4
165	1.0	234.5	3.4	84.8
12	1.0	23.8	35.1	65.9
73	1.0	129.4	5.7	31.3
144	1.0	96.2	14.8	38.9
20	1.0	218.4	27.7	53.4
199	1.0	232.1	8.6	8.7
8	1.0	8.6	2.1	1.0
39	1.0	228.0	37.7	32.0
88	1.0	88.3	25.5	73.4
81	1.0	239.8	4.1	36.9
197	1.0	177.0	9.3	6.4
69	1.0	216.8	43.9	27.2
160	1.0	172.5	18.1	30.7
25	1.0	262.9	3.5	19.5
99	1.0	135.2	41.7	45.9
151	1.0	121.0	8.4	48.7
23	1.0	228.3	16.9	26.2

```
138    1.0   43.0   25.9       20.5
159    1.0  131.7   18.4       34.6
89     1.0  109.8   47.8       51.4
82     1.0   75.3   20.3       32.5
24     1.0   62.3   12.6       18.3
174    1.0  222.4    3.4       13.1
137    1.0  273.7   28.9       59.7
83     1.0   68.4   44.5       35.6
107    1.0   90.4    0.3       23.2
34     1.0   95.7    1.4        7.4
97     1.0  184.9   21.0       22.0
167    1.0  206.8    5.2       19.4
123    1.0  123.1   34.6       12.4
157    1.0  149.8    1.3       24.3
75     1.0   16.9   43.7       89.4
152    1.0  197.6   23.3       14.2
117    1.0   76.4    0.8       14.8
149    1.0   44.7   25.8       20.6
63     1.0  102.7   29.6        8.4
54     1.0  262.7   28.8       15.9
125    1.0   87.2   11.8       25.9
80     1.0   76.4   26.7       22.3
```

[20]:
```
y_test_pred = lr.predict(X_test_sm)
y_test_pred
```

[20]:
```
112    15.778786
165    18.139634
12      9.719884
73     12.357547
144    11.499283
20     19.617089
199    18.211406
8       5.205011
39     21.081440
88     12.315529
81     18.294983
197    15.240080
69     21.081944
160    16.003604
25     19.429215
99     16.444731
151    12.248662
23     18.932425
138     9.632635
159    13.805339
89     15.698036
```

```
82     10.886794
24      9.317007
174    17.161481
137    22.812340
83     13.010573
107     9.619630
34      9.955833
97     16.946790
167    16.515309
123    14.902240
157    12.997384
75     10.327223
152    17.848668
117     8.863801
149     9.716368
63     13.247286
54     22.005871
125    10.638548
80     11.561512
dtype: float64
```

[21]:
```
from sklearn.metrics import mean_squared_error,r2_score
```

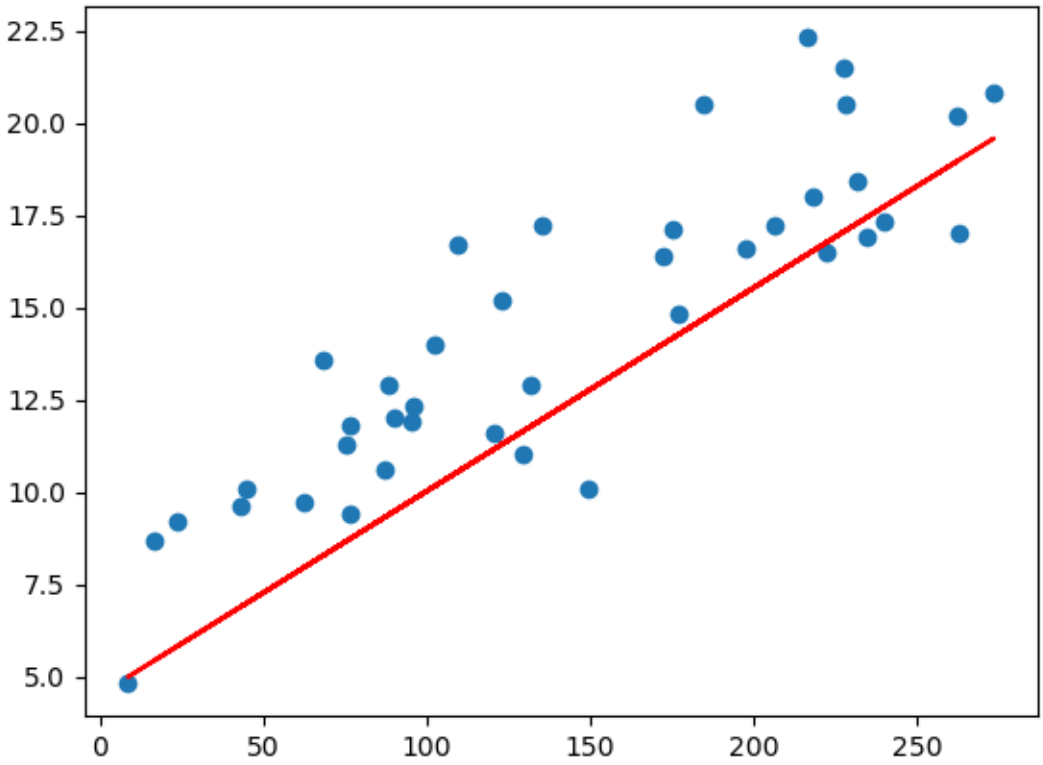
[22]:
```
np.sqrt(mean_squared_error(y_test, y_test_pred))
```

[22]:
```
1.304511191229726
```

[23]:
```
r2_score(y_test, y_test_pred)
```

[23]:
```
0.9006409689782175
```

[24]:
```
plt.scatter(X_test['TV'],y_test)
plt.plot(X_test['TV'],4.511128 + 0.055052 * X_test['TV'],'r')
plt.show()
```


22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
0
50
100
150
200
250

[ ]: