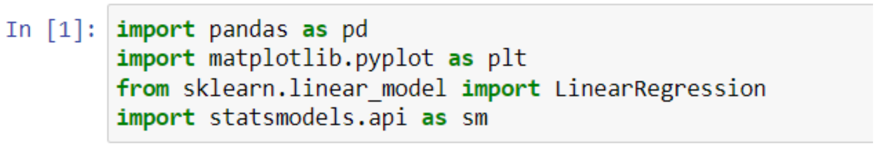
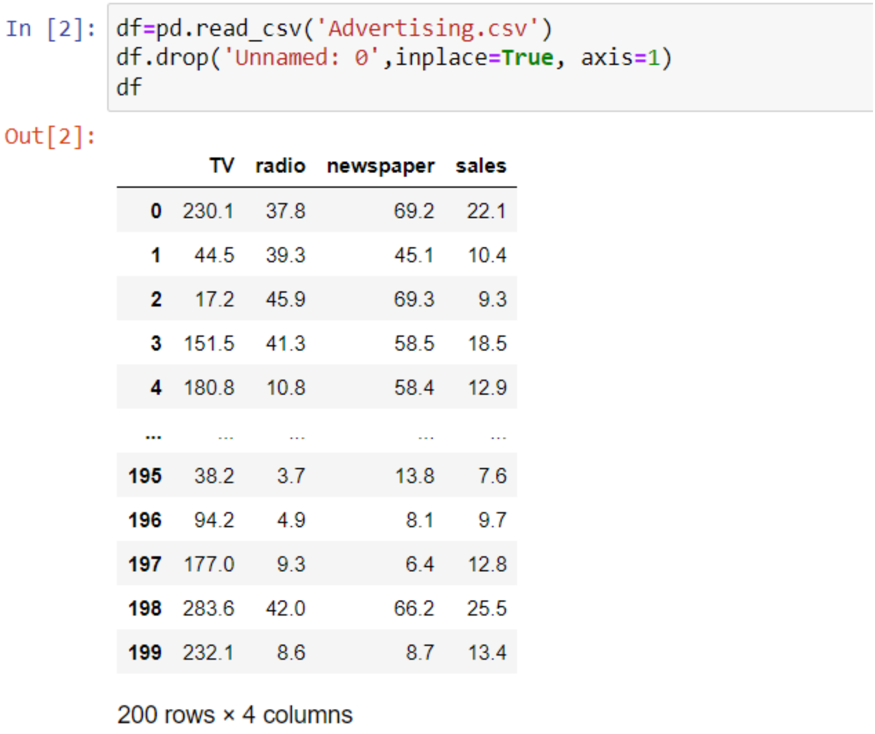
3.2 Example : Linear regression

In this example, we will use the dataset “Advertising and Sales”, which is available at www.kaggle.com. The datasets includes 4 features and 200 instances. The 3 features are “TV”, “radio”, “newspaper”, which refer to the amount of resources that are allocated in each media. The 4th feature is the target feature “sales”, which refers to the total sales number. For the purpose of this chapter, we will assume that the only feature that has strong impact on the target feature is “TV”. Thus, we ignore the other two features. In addition, we will use 160 out of the 200 instances. For this example we will use the programming language “python”.



```
In [1]: import pandas as pd
        import matplotlib.pyplot as plt
        from sklearn.linear_model import LinearRegression
        import statsmodels.api as sm
```



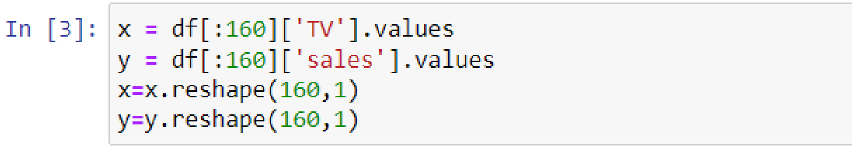
```
In [2]: df=pd.read_csv('Advertising.csv')
        df.drop('Unnamed: 0',inplace=True, axis=1)
        df
```

Out[2]:

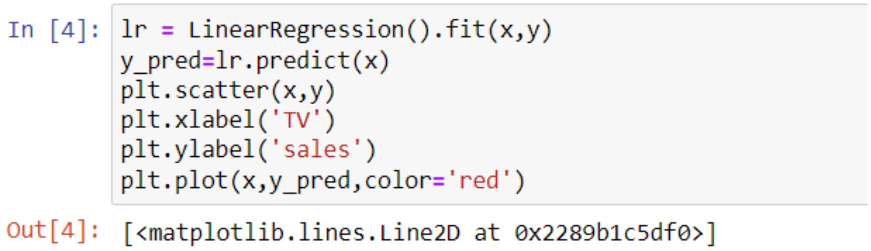
	TV	radio	newspaper	sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9
...	...	...	...	...
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	9.7
197	177.0	9.3	6.4	12.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	13.4

200 rows × 4 columns

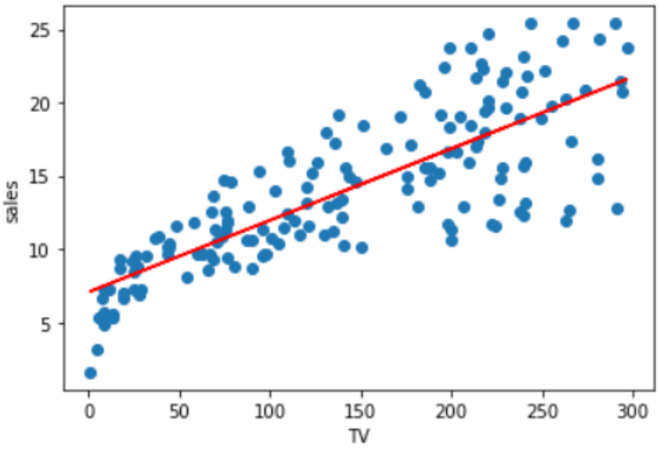
First, we will handle our data because they should have a specific form, before they are used as an input to our model. Then we will fit the model to our data and we will plot the results.

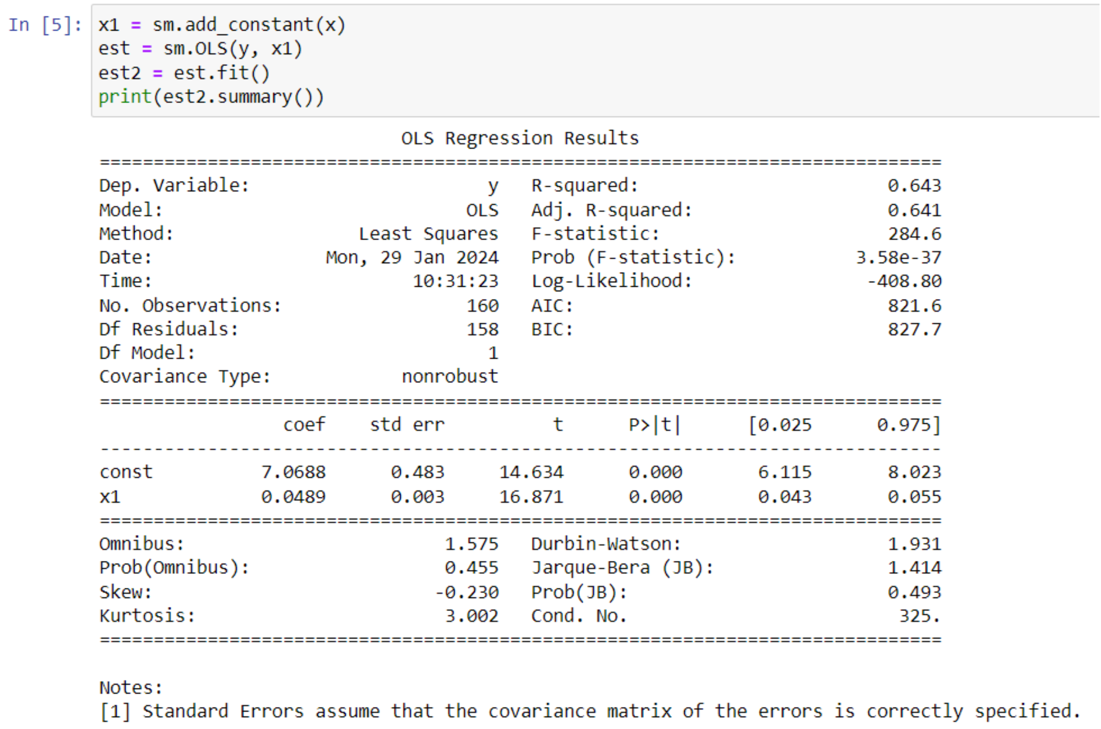


```
In [3]: x = df[:160]['TV'].values
        y = df[:160]['sales'].values
        x=x.reshape(160,1)
        y=y.reshape(160,1)
```



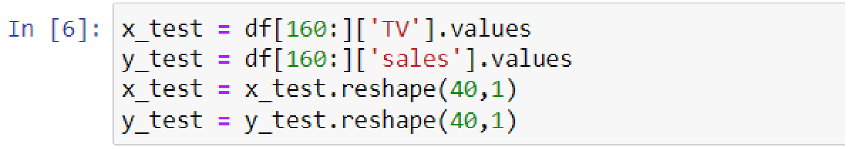
```
In [4]: lr = LinearRegression().fit(x,y)
        y_pred=lr.predict(x)
        plt.scatter(x,y)
        plt.xlabel('TV')
        plt.ylabel('sales')
        plt.plot(x,y_pred,color='red')

Out[4]: [<matplotlib.lines.Line2D at 0x2289b1c5df0>]
```



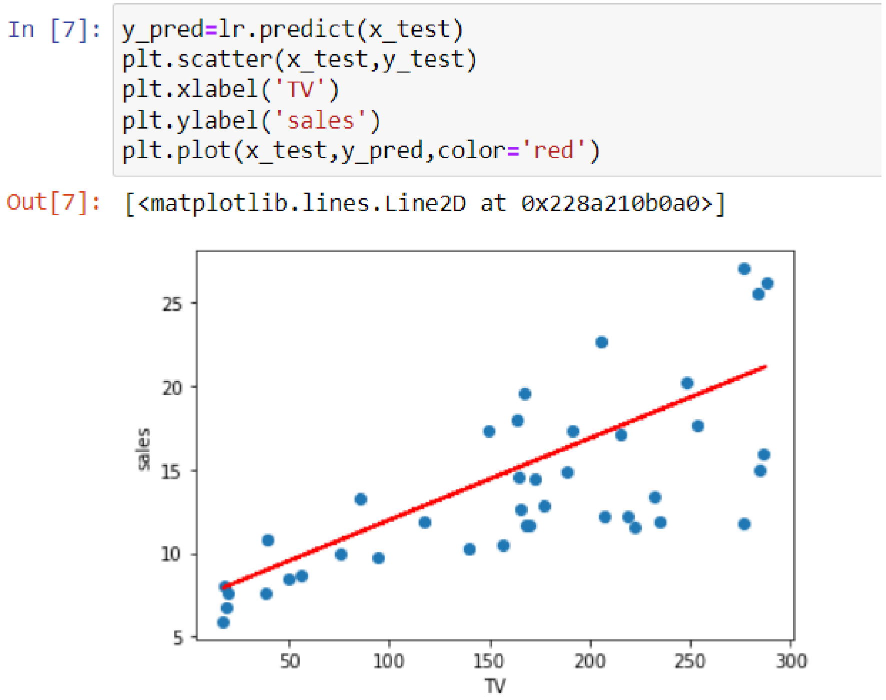


```
In [5]: x1 = sm.add_constant(x)
        est = sm.OLS(y, x1)
        est2 = est.fit()
        print(est2.summary())

                            OLS Regression Results                            
==============================================================================
Dep. Variable:                      y   R-squared:                       0.643
Model:                            OLS   Adj. R-squared:                  0.641
Method:                 Least Squares   F-statistic:                     284.6
Date:                Mon, 29 Jan 2024   Prob (F-statistic):           3.58e-37
Time:                        10:31:23   Log-Likelihood:                -408.80
No. Observations:                 160   AIC:                             821.6
Df Residuals:                     158   BIC:                             827.7
Df Model:                           1                                         
Covariance Type:            nonrobust                                         
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          7.0688      0.483     14.634      0.000       6.115       8.023
x1             0.0489      0.003     16.871      0.000       0.043       0.055
==============================================================================
Omnibus:                        1.575   Durbin-Watson:                   1.931
Prob(Omnibus):                  0.455   Jarque-Bera (JB):                1.414
Skew:                          -0.230   Prob(JB):                        0.493
Kurtosis:                       3.002   Cond. No.                         325.
==============================================================================

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

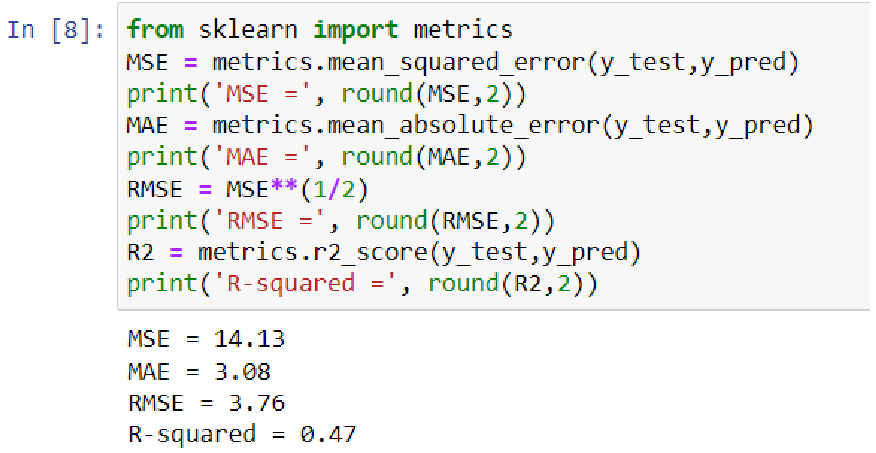
Evaluation



```
In [6]: x_test = df[160:]['TV'].values
        y_test = df[160:]['sales'].values
        x_test = x_test.reshape(40,1)
        y_test = y_test.reshape(40,1)
```



```
In [7]: y_pred=lr.predict(x_test)
        plt.scatter(x_test,y_test)
        plt.xlabel('TV')
        plt.ylabel('sales')
        plt.plot(x_test,y_pred,color='red')

Out[7]: [<matplotlib.lines.Line2D at 0x228a210b0a0>]
```



```
In [8]: from sklearn import metrics
        MSE = metrics.mean_squared_error(y_test,y_pred)
        print('MSE =', round(MSE,2))
        MAE = metrics.mean_absolute_error(y_test,y_pred)
        print('MAE =', round(MAE,2))
        RMSE = MSE**(1/2)
        print('RMSE =', round(RMSE,2))
        R2 = metrics.r2_score(y_test,y_pred)
        print('R-squared =', round(R2,2))

        MSE = 14.13
        MAE = 3.08
        RMSE = 3.76
        R-squared = 0.47
```