```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
```

Creating a Dataset

```
X=np.random.randint(1,100,100)
X=X.reshape(-1,1)
X
```

```
array([[43],
       [86],
       [59],
       [62],
       [98],
       [87],
       [71],
       [69],
       [21],
       [30],
       [73],
       [70],
       [44],
       [58],
       [43],
       [96],
       [15],
       [26],
       [35],
       [69],
       [ 8],
       [60],
       [56],
       [94],
       [54],
       [55],
       [30],
       [46],
       [51],
       [73],
       [29],
       [85],
       [73],
       [51],
       [47],
       [67],
       [77],
       [13],
       [12],
       [ 6],
       [16],
       [15],
       [12],
       [21],
       [36],
       [20],
       [90],
       [76],
       [21],
       [28],
       [41],
       [52],
       [85],
       [42],
       [53],
       [92],
       [60],
       [61],
```

Create Errors and add to y

```
err=np.random.rand(100)
err=err.reshape(-1,1)
err
```

```
array([[0.0847403 ],
       [0.65861218],
       [0.4772855 ],
       [0.73430669],
       [0.56450484],
       [0.7174708 ],
```

```
[0.92143322],
[0.10682987],
[0.96528252],
[0.56597248],
[0.23142304],
[0.85187746],
[0.81636823],
[0.66350869],
[0.0605795 ],
[0.26933085],
[0.78377649],
[0.87890487],
[0.84196591],
[0.66219466],
[0.56087308],
[0.03219164],
[0.70439104],
[0.33056653],
[0.75385337],
[0.1236444 ],
[0.2872219 ],
[0.44533002],
[0.25732365],
[0.16141242],
[0.64603882],
[0.41596631],
[0.82746091],
[0.12040676],
[0.65528035],
[0.63516401],
[0.09469582],
[0.89392777],
[0.70314239],
[0.96119862],
[0.71745855],
[0.97998839],
[0.11660079],
[0.26869413],
[0.3006044 ],
[0.45511645],
[0.15454835],
[0.68529609],
[0.41087612],
[0.0910359 ],
[0.91144231],
[0.92977831],
[0.58725193],
[0.11645415],
[0.69403651],
[0.94447958],
[0.93836521],
```

```
y=3*X+1+err*50
```

```
y=y.reshape(-1,1)
y
```

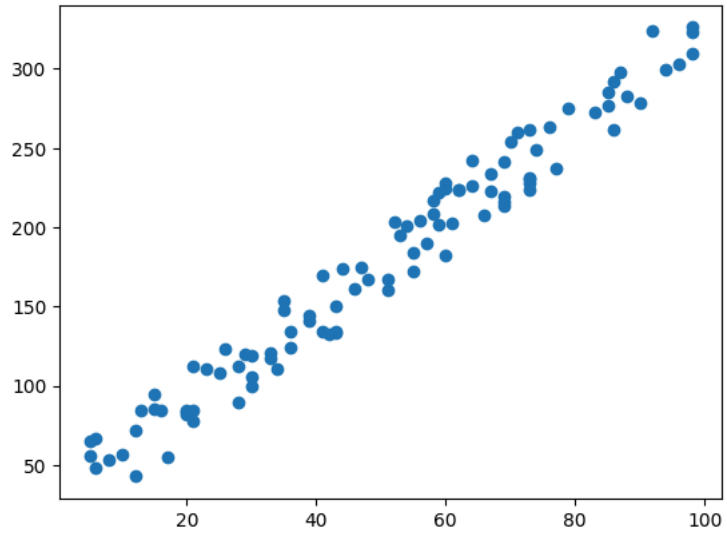
```
array([[134.23701478],
       [291.93060922],
       [201.86427508],
       [223.71533429],
       [323.22524203],
       [297.8735401 ],
       [260.07166086],
       [213.34149333],
       [112.26412604],
       [119.29862387],
       [231.57115184],
       [253.59387311],
       [173.81841165],
       [208.17543429],
       [133.02897487],
       [302.46654275],
       [ 85.18882458],
       [122.94524342],
       [148.09829571],
       [241.10973309],
       [ 53.04365394],
       [182.60958177],
       [204.21955192],
       [299.52832647],
       [200.69266863],
       [172.18221988],
       [105.36109478],
       [161.26650106],
       [166.86618233],
       [228.07062108],
       [120.30194123],
```

```
[276.79831574],
[261.37304537],
[160.02033777],
[174.76401764],
[233.7582004 ],
[236.73479123],
[ 84.69638825],
[ 72.15711972],
[ 67.05993079],
[ 84.87292766],
[ 94.99941959],
[ 42.83003957],
[ 77.43470653],
[124.03022016],
[ 83.75582237],
[278.72741733],
[263.26480457],
[ 84.54380601],
[ 89.55179478],
[169.57211531],
[203.48891536],
[285.36259635],
[132.82270741],
[194.70182527],
[324.22397882],
[227.91826025],
[202.70654509]
```

Visualizing the data

```
plt.scatter(X,y)
```

```
<matplotlib.collections.PathCollection at 0x7f7ad18b07c0>
```



Split the dataset

```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=46)
```

```
X_train.shape,X_test.shape
```

```
((80, 1), (20, 1))
```

```
y_train.shape,y_test.shape
```

```
((80, 1), (20, 1))
```

scale the features

```
sc=StandardScaler()
sc.fit(X_train)
```

```
StandardScaler
StandardScaler()
```

```
X_train_scaled=sc.transform(X_train)
X_train_scaled[:5]
```

```
array([[ 1.33401317],
       [ 1.25770208],
       [ 0.87614666],
       [-0.76454169],
       [ 0.34196905]])
```

```
X_test_scaled=sc.transform(X_test)
```

```
X_test_scaled[:5]
```

```
array([[ 0.53274677],
       [ 0.03672471],
       [-1.41318592],
       [ 0.87614666],
       [-0.53560843]])
```

Add column X0

```
X_train_scaled=np.insert(X_train_scaled,0,1,axis=1)
X_test_scaled=np.insert(X_test_scaled,0,1,axis=1)
```

```
X_train_scaled.shape
```

```
(80, 2)
```

```
X_test_scaled.shape
```

```
(20, 2)
```

```
X_train_scaled.min(),X_train_scaled.max()
```

```
(-1.7184302659385815, 1.8300352290886308)
```

Initialize the Parameters and Hyperpara-meters

```
alpha=0.05
n=len(X_train)
theta=np.array([0.5,0.5])
```

```
theta
```

```
array([0.5, 0.5])
```

Algorithm

```
# For a single column
# y_cap=np.dot(X_train_scaled[0],theta)
# theta-alpha*(y_cap-y[0]*X_train_scaled[0])
```

```
for epoch in range(5000):
    gradient=0
    for i in range(n):
        y_cap=np.dot(X_train_scaled[i],theta)
        gradient+=(y_cap-y_train[i])*X_train_scaled[i]
    gradient=(gradient)/n
    theta=theta-alpha*gradient
```

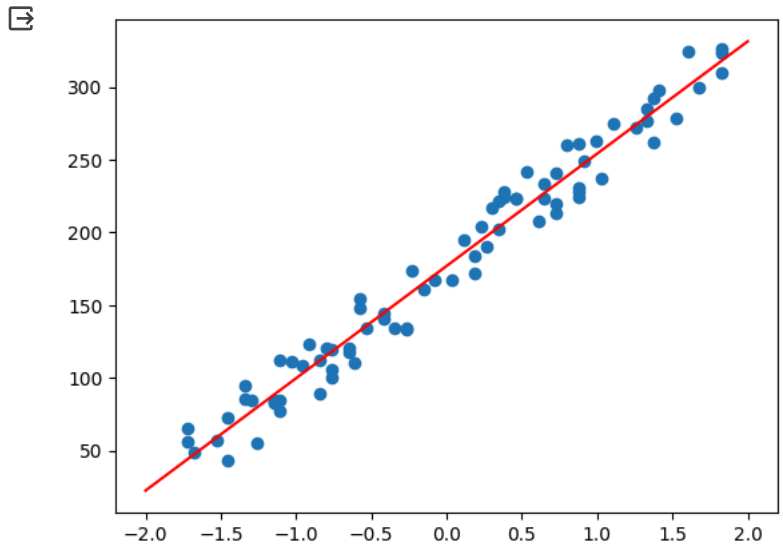
```
theta
```

```
array([176.9481513 ,  77.24699918])
```

Visualizing the new graph with the line plotted using the update theta

```
plt.scatter(X_train_scaled[:,1],y_train)
X_plot=np.linspace(-2,2,100)
y_plot=(theta[1]*X_plot)+theta[0]
plt.plot(X_plot,y_plot,'r-')
```

```
plt.show()
```



R^2 evaluation

```
y_pred=np.dot(X_test_scaled,theta)
y_pred
```

```
array([218.1012405 , 179.78502495,  67.78377952, 244.62785126,
       135.57400702, 167.99542017, 188.62722854, 288.83886919,
       232.83824647, 156.20581539, 150.311013  , 200.41683332,
       153.25841419, 209.25903691,  47.15197115, 182.73242615,
       235.78564767, 206.31163571,  53.04677355, 312.41807875])
```

```
y_test=y_test.flatten()
y_test
```

```
array([226.49423741, 160.02033777,  84.69638825, 231.57115184,
       124.03022016, 174.76401764, 200.69266863, 282.34438437,
       216.07768966, 150.45118274, 169.57211531, 208.17543429,
       132.82270741, 202.70654509,  67.05993079, 203.48891536,
       253.59387311, 182.60958177,  53.04365394, 302.46654275])
```

```
sstot=(y_test-np.mean(y_test))**2
ssres=(y_test-y_pred)**2
r2=1-(np.sum(ssres)/np.sum(sstot))
r2
```

```
0.9488622687901136
```

```
a=y_pred-y_test
(np.sqrt(np.abs(a)))
```

```
array([2.89706695, 4.44574934, 4.11249422, 3.61340552, 3.39761488,
       2.60165283, 3.47353424, 2.54842791, 4.0939659 , 2.39888154,
       4.38874724, 2.78542653, 4.520587  , 2.55978355, 4.46183366,
       4.55592902, 4.21997932, 4.86847553, 0.05585344, 3.15460552])
```

Calculating R^2 using Metrics

```
from sklearn.metrics import r2_score
```

```
r2_score(y_pred,y_test)
```

```
0.9540856405763084
```

Calculating r^2 using Scikit Model

```
from sklearn.linear_model import LinearRegression
model=LinearRegression()
```

```
model.fit(X_train_scaled,y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
y_pred=model.predict(X_test_scaled)
y_pred=y_pred.reshape(-1,1)
y_pred
```

```
array([[218.1012405 ],
       [179.78502495],
       [ 67.78377952],
       [244.62785126],
       [135.57400702],
       [167.99542017],
       [188.62722854],
       [288.83886919],
       [232.83824647],
       [156.20581539],
       [150.311013  ],
       [200.41683332],
       [153.25841419],
       [209.25903691],
       [ 47.15197115],
       [182.73242615],
       [235.78564767],
       [206.31163571],
       [ 53.04677355],
       [312.41807875]])
```

```
y_test=y_test.reshape(-1,1)
```

```
r_squared = model.score(X_test_scaled, y_test)
r_squared
```

```
0.9488622687901133
```

```
model.coef_
```

```
array([[ 0.        , 77.24699918]])
```