Step 1: Import packages and classes

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from sklearn.linear_model import LinearRegression
```

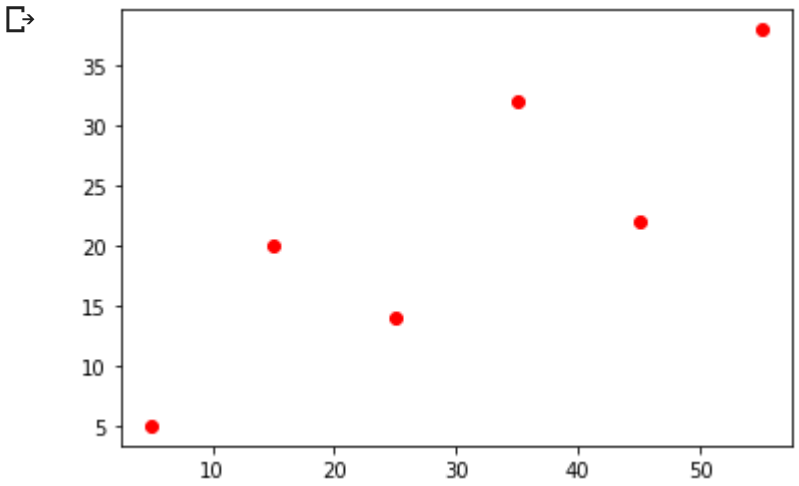
```
1 x = np.array([5, 15, 25, 35, 45, 55])
2 print(x.shape)
3 x = np.array([5, 15, 25, 35, 45, 55]).reshape((-1, 1))
4 print(x.shape)
```

```
(6,)
(6, 1)
```

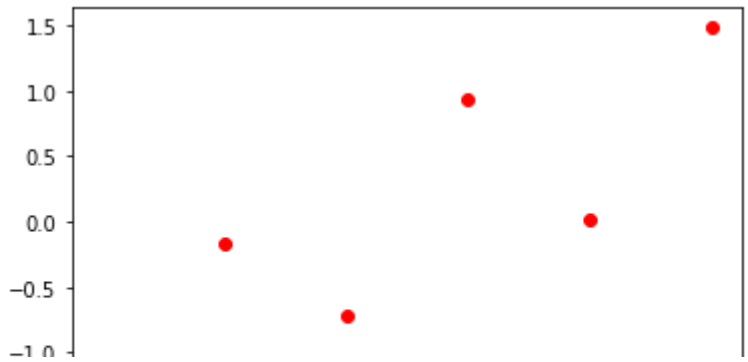
Step 2: Provide data

```
1 x = np.array([5, 15, 25, 35, 45, 55]).reshape((-1, 1))
2 y = np.array([5, 20, 14, 32, 22, 38]).reshape((-1,1))
```

```
1 plt.scatter(x, y, color='r')
2 plt.show()
```




```
1 from sklearn.preprocessing import StandardScaler
2 sc_X = StandardScaler()
3 sc_y = StandardScaler()
4 x = sc_X.fit_transform(x)
5 y = sc_y.fit_transform(y)
6
7 plt.scatter(x, y, color='r')
8 plt.show()
```



Step 3: Create a model and fit it

The next step is to create a linear regression model and fit it using the existing data.

Create an instance of the class LinearRegression, which will represent the regression model:

```
1 model = LinearRegression()
```

```
1 model.fit(x, y)
```

```
   LinearRegression()
```

```
1 model = LinearRegression().fit(x, y)
```

Step 4: Get results

```
1 r_sq = model.score(x, y)
2 print(f"coefficient of determination: {r_sq}")
```

```
   coefficient of determination: 0.7158756137479542
```

```
1 print(f"intercept: {model.intercept_}")
2 print(f"slope: {model.coef_}")
```

```
   intercept: 5.633333333333329
   slope: [0.54]
```

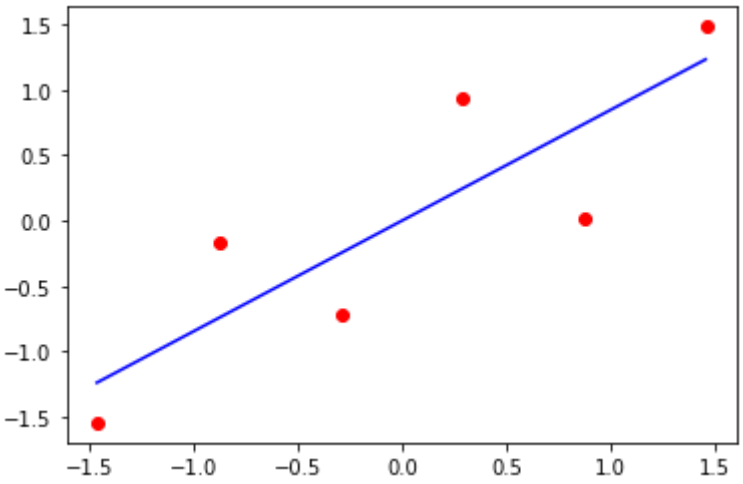
Step 5: Predict response

```
1 y_pred = model.predict(x)
2 print(f"predicted response:\n{y_pred}")
```

```
   predicted response:
   [ 8.33333333 13.73333333 19.13333333 24.53333333 29.93333333 35.33333333]
```

```
1 X_grid = np.arange(min(x), max(x), 0.01) #this step required because data is fe
2 X_grid = X_grid.reshape((len(X_grid), 1))
3 plt.scatter(x, y, color = 'red')
```

```
4 plt.plot(X_grid, model.predict(X_grid), color = 'blue')
5 #plt.xlabel()
6 #plt.ylabel()
7 plt.show()
```



Colab paid products - Cancel contracts here

✓ 0s completed at 1:24 PM