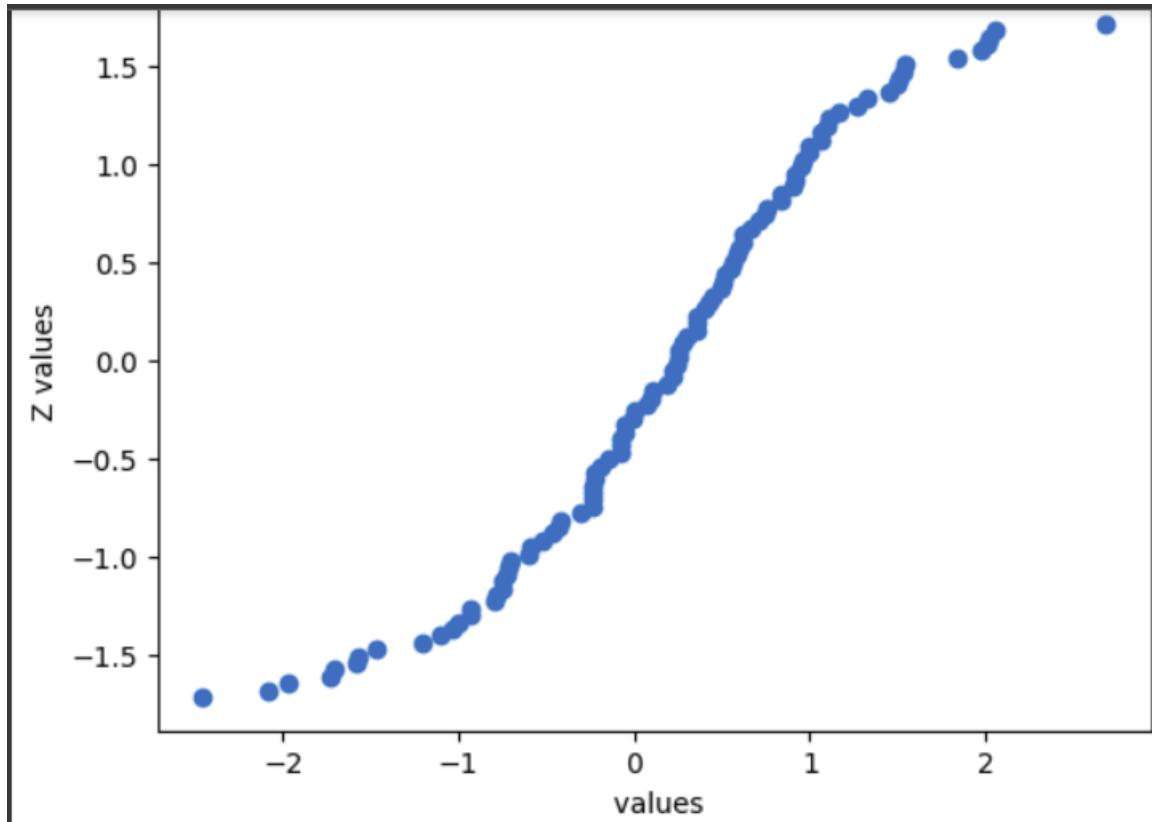


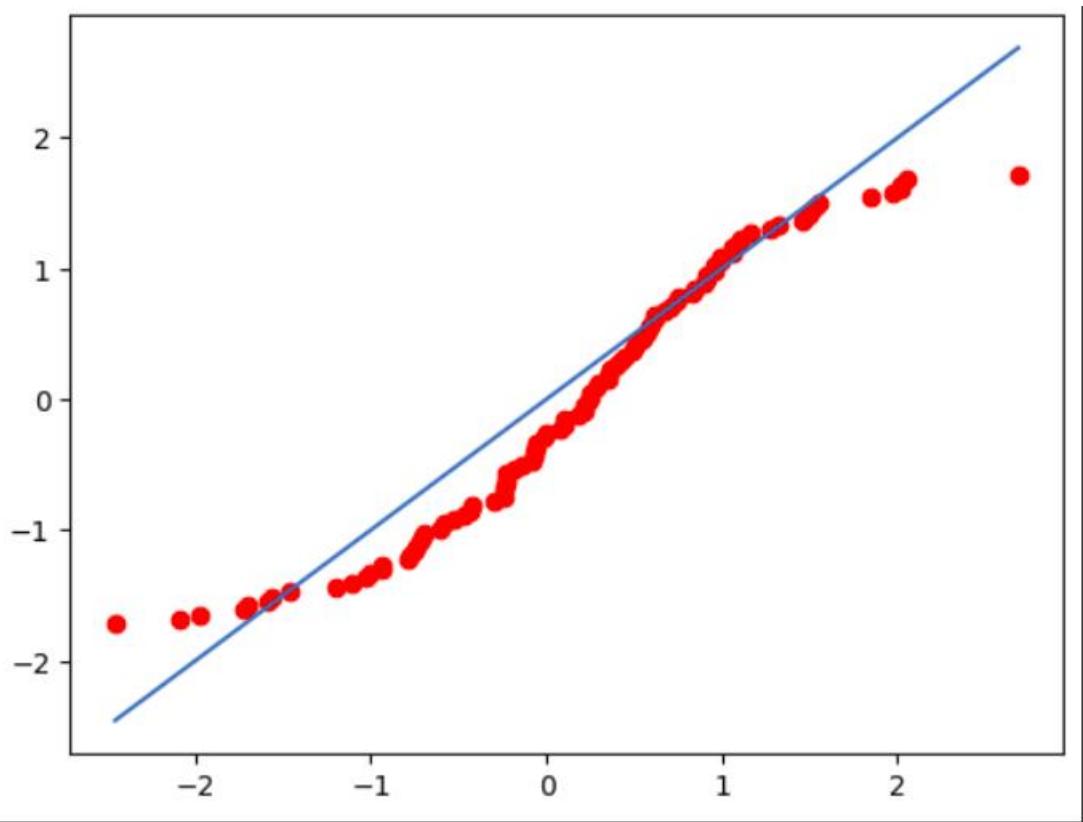
Program-1

```
❶ from scipy.stats import zscore
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
#its just an example npp of original values v/s theoretical values(z scores)
def npp(data):
    data = sorted(data)
    p = [(data.index(i)-0.5)/len(data) for i in data]
    z = zscore(p)

    plt.scatter(data, z)
    plt.ylabel('Z values')
    plt.xlabel('values')
    plt.show()
    plt.plot(data, z, 'ro', data, data)
    plt.show()

#n datapoints
n = 100
data = np.random.randn(n)
npp(data)
```





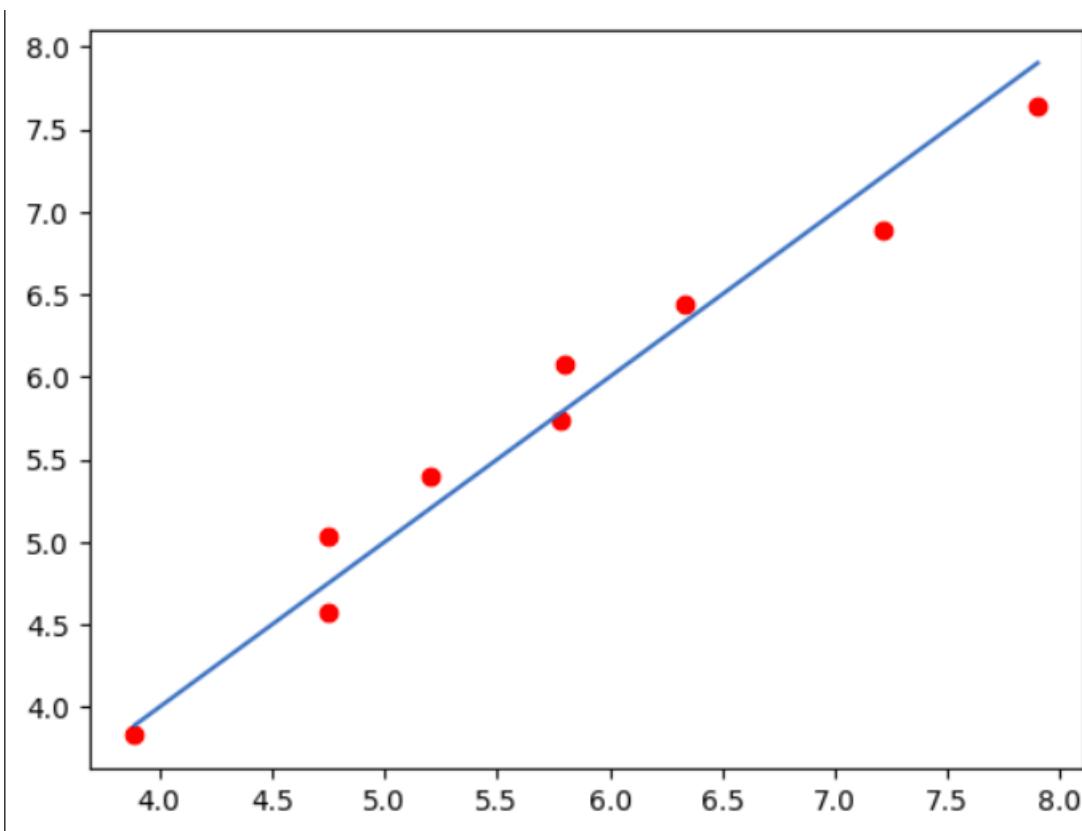
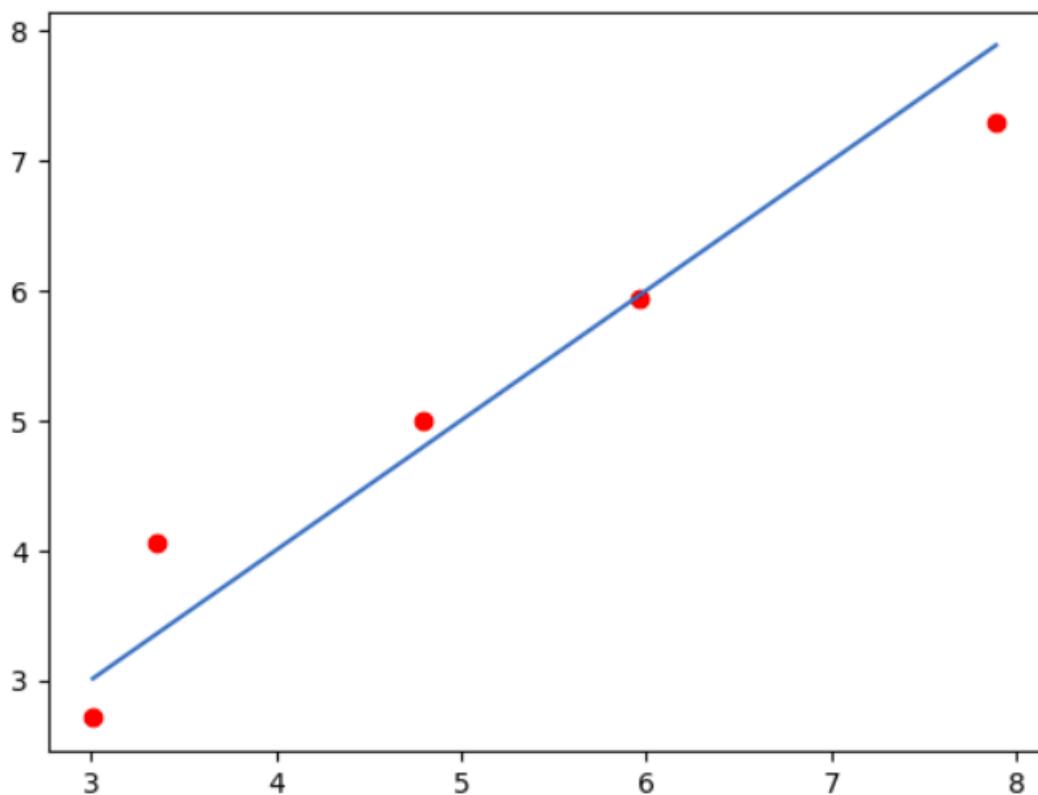
Program-2

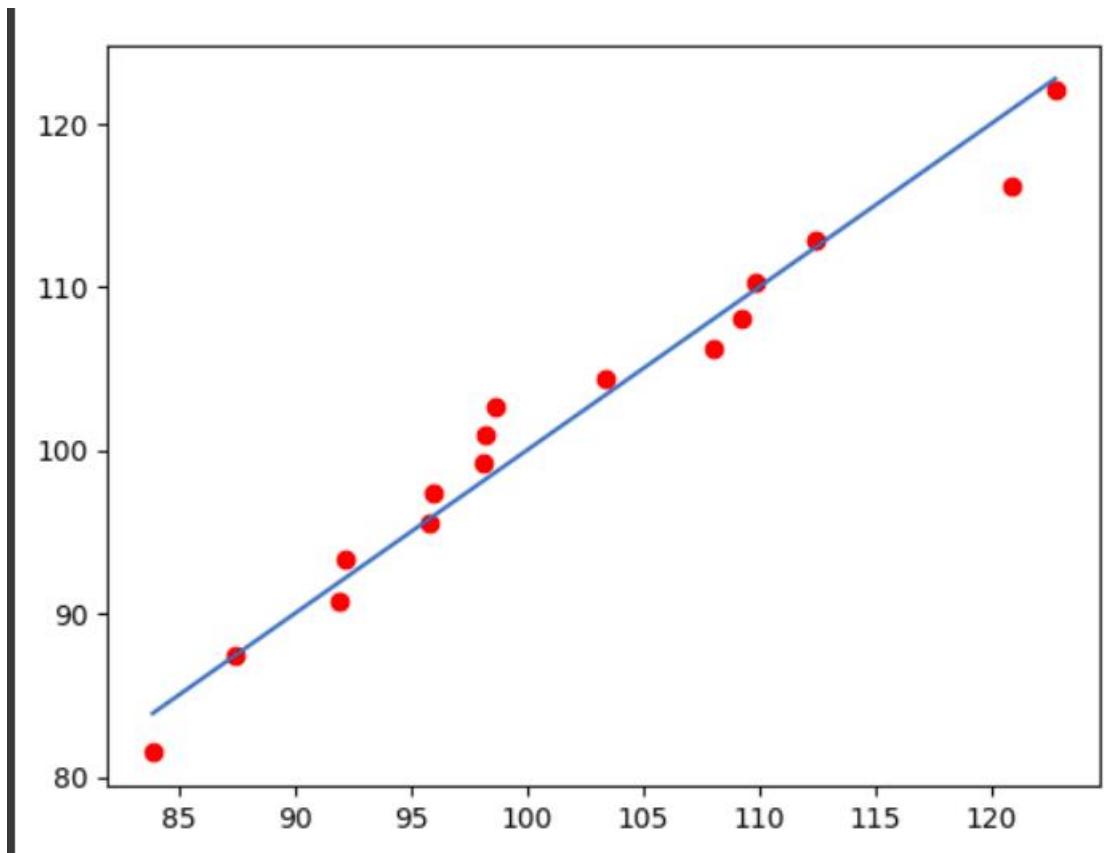
```

❶ X1 = [3.01, 3.35, 4.79, 5.96, 7.89]
X2 = [3.89, 4.75, 4.75, 5.20, 5.78, 5.80, 6.33, 7.21, 7.90]
X3 = [108.047, 109.249, 103.385, 112.454, 95.780, 122.734, 109.842 , 120.858,
      98.604, 98.122, 95.971, 98.173, 87.437 , 91.884, 92.193, 83.882]
def npp1(data):
    p = []
    t = []
    data = np.sort(np.array(data))
    p = [(i - 0.5)/len(data) for i in range(1, len(data)+1)]
    t = [norm.ppf(i, np.mean(data), np.std(data)) for i in p]
    plt.plot(data, t, 'ro', data, data)
    plt.show()

npp1(X1)
npp1(X2)
npp1(X3)

```



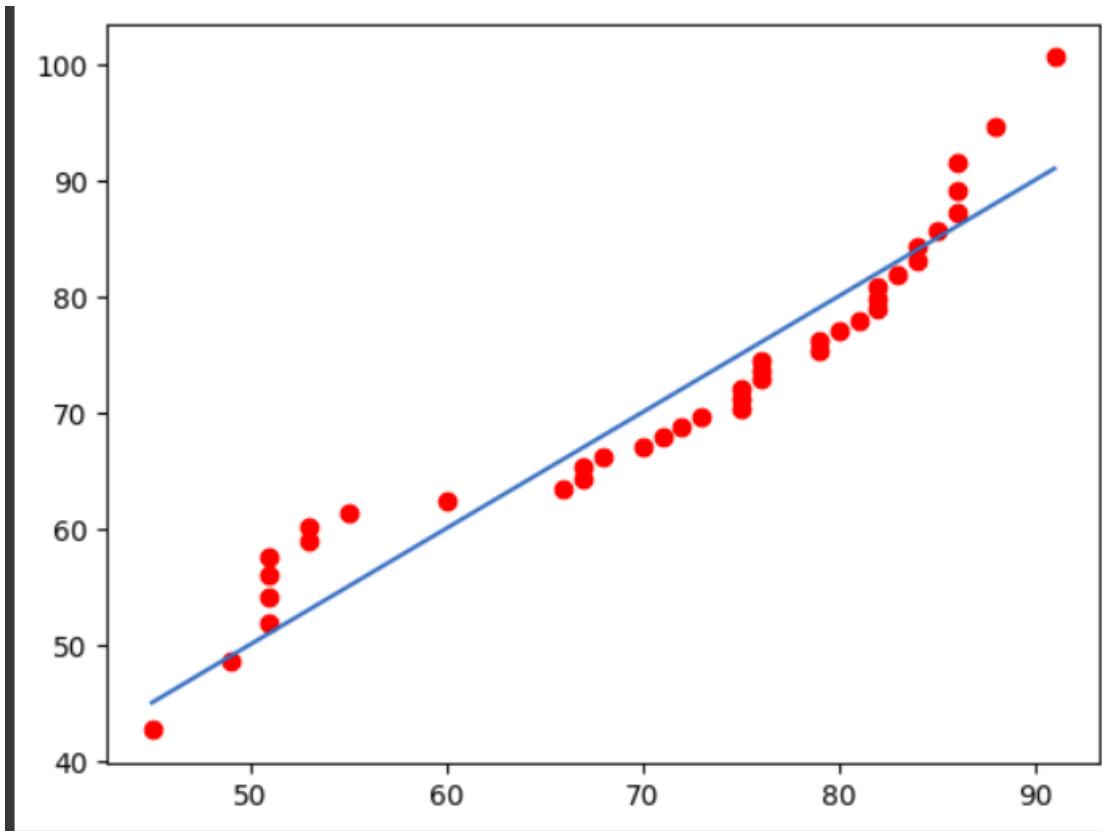


Program-3

```
[3] X1 = [91, 51, 79, 53, 82, 51, 76, 82, 84, 53,
     86, 51, 85, 45, 88, 51, 80, 49, 82, 75,
     73, 67, 68, 86, 72, 75, 75, 66, 84, 70,
     79, 60, 86, 71, 67, 81, 76, 83, 76, 55]

def npp1(data):
    p = []
    t = []
    data = np.sort(np.array(data))
    p = [(i - 0.5)/len(data) for i in range(1, len(data)+1)]
    t = [norm.ppf(i, np.mean(data), np.std(data)) for i in p]
    plt.plot(data, t, 'ro', data, data)
    plt.show()

npp1(X1)
```



Program-4

```
▶ X1 = [4.1, 1.8, 3.2, 1.9, 4.6, 2.0, 4.5, 3.9, 4.3, 2.3,
       3.8, 1.9, 4.6, 1.8, 4.7, 1.8, 4.6, 1.9, 3.5, 4.0,
       3.7, 3.7, 4.3, 3.6, 3.8, 3.8, 3.8, 2.5, 4.5, 4.1,
       3.7, 3.8, 3.4, 4.0, 3.4, 2.3, 4.4, 4.1, 4.3, 3.3, 2.0]

def npp1(data):
    p = []
    t = []
    data = np.sort(np.array(data))
    p = [(i - 0.5)/len(data) for i in range(1, len(data)+1)]
    t = [norm.ppf(i, np.mean(data), np.std(data)) for i in p]
    plt.plot(data, t, 'ro', data, data)
    plt.show()

npp1(X1)
```

