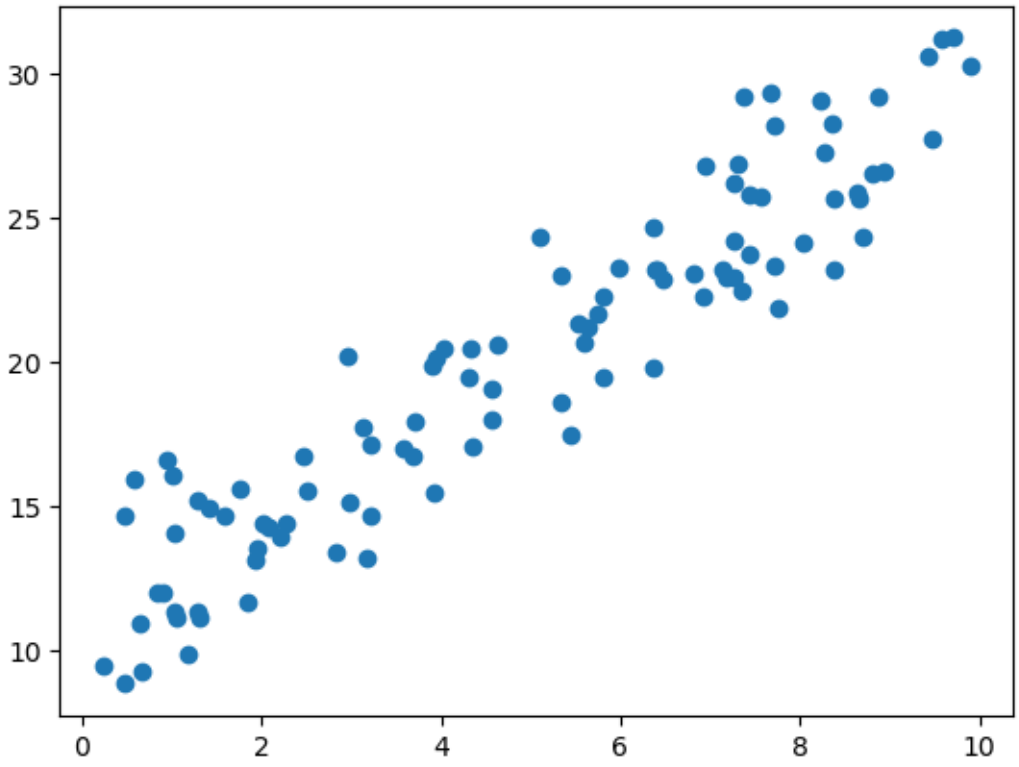
Untitled

September 2, 2023

[20]:
```
import numpy as np
from matplotlib import pyplot as plt
dataset1 = np.load("dataset_1.npy")
dataset2 = np.load("dataset_2.npy")
```

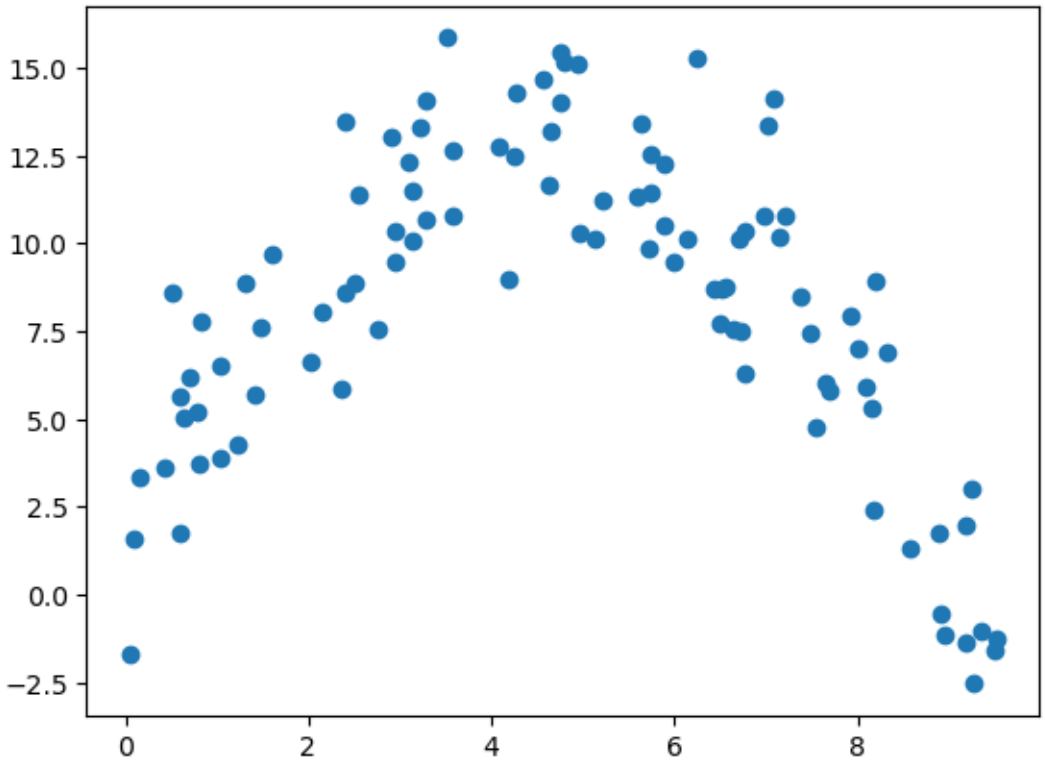
[21]:
```
plt.scatter(dataset1.T[0], dataset1.T[1])
```

[21]:
```
<matplotlib.collections.PathCollection at 0x7f82fc105be0>
```



[19]:
```
plt.scatter(dataset2.T[0], dataset2.T[1])
```

[19]:
```
<matplotlib.collections.PathCollection at 0x7f82fc18ac10>
```



[53]:
```
mean_x = sum(dataset1.T[0])/len(dataset1.T[0])
mean_x
mean_y = sum(dataset1.T[1])/len(dataset1.T[1])
mean_y
sum_vals = 0
n = len(dataset1.T[0])
for i in range(n):
    sum_vals += (dataset1.T[0][i] - mean_x)* (dataset1.T[1][i] - mean_y)
covariance = sum_vals/(n - 1)

std_0 = np.dot(dataset1.T[0] - mean_x, dataset1.T[0] - mean_x) / (n - 1)
std_1 = np.dot(dataset1.T[1] - mean_y, dataset1.T[1] - mean_y) / (n - 1)

correlation = covariance/(std_0*std_1)**(1/2)
correlation
```

[53]:
```
0.9390949620246674
```

[54]:
```
mean_x = sum(dataset2.T[0])/len(dataset2.T[0])
mean_x
mean_y = sum(dataset2.T[1])/len(dataset2.T[1])
mean_y
```

```
sum_vals = 0
n = len(dataset2.T[0])
for i in range(n):
    sum_vals += (dataset2.T[0][i] - mean_x)* (dataset2.T[1][i] - mean_y)
covariance = sum_vals/(n - 1)

std_0 = np.dot(dataset2.T[0] - mean_x, dataset2.T[0] - mean_x) / (n - 1)
std_1 = np.dot(dataset2.T[1] - mean_y, dataset2.T[1] - mean_y) / (n - 1)

correlation = covariance/(std_0*std_1)**(1/2)
correlation
```

[54]: -0.17930801518997463

[ ]: