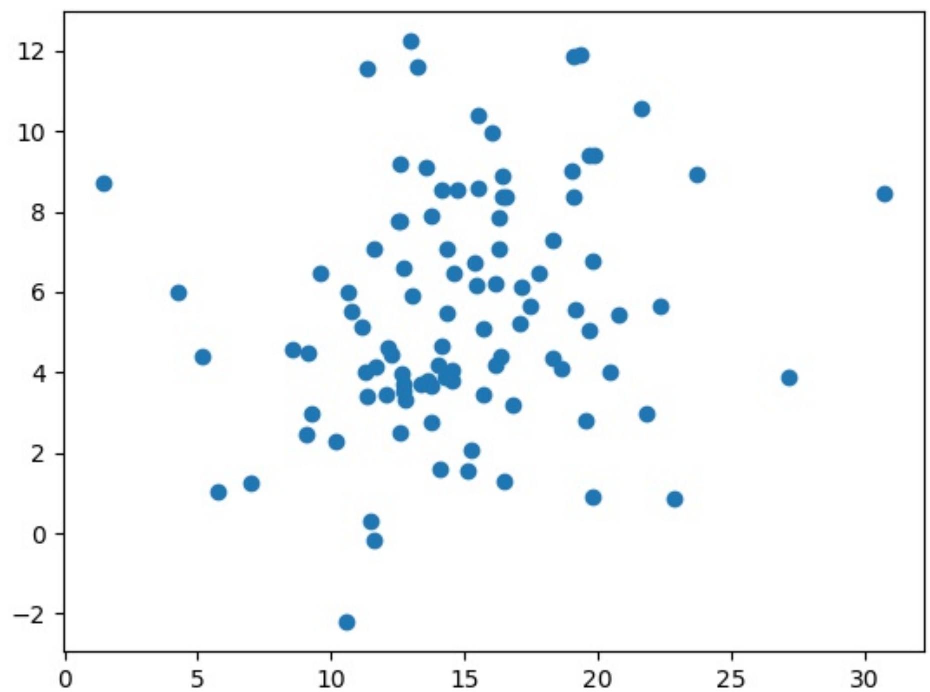
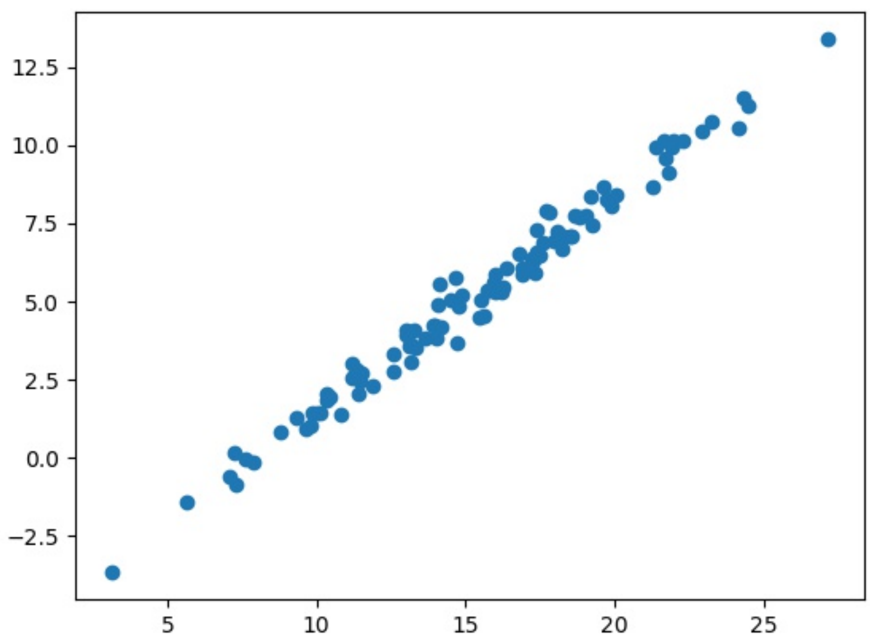
# 2c

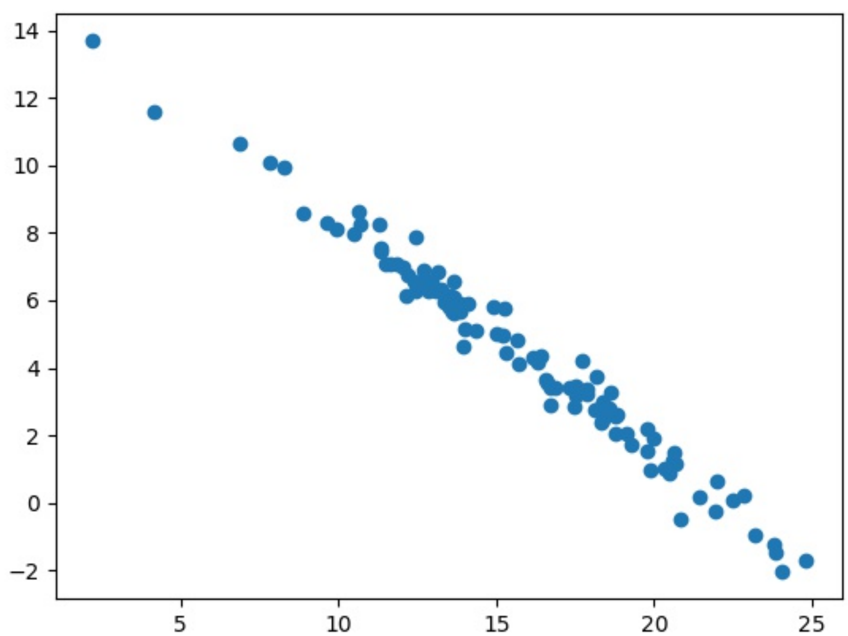
when simga = [[20, 0], [0, 10]], the MLE mean is [ 14.62972319   5.13279893], and MLE covariance is [[ 13.93459582   2.83012681]
 [  2.83012681   9.93336424]]
the singular values for sigma matrix is [ 20.  10.]



when simga = [[20, 14], [14, 10]], the MLE mean is [ 14.98469043   5.04061468],
and MLE covariance is [[ 20.46621341  14.35034222]
 [ 14.35034222  10.23210116]]
the singular values for sigma matrix is [ 29.86606875   0.13393125]



when simga = [[20, -14], [-14, 10]], the MLE mean is [ 14.35926801   5.38991362],
and MLE covariance is [[ 16.75876404 -11.60036104]
 [-11.60036104   8.29960636]]
the singular values for sigma matrix is [ 29.86606875   0.13393125]



```
import numpy as np
import matplotlib.pyplot as plt

def mean(samples):
	return np.average(samples, axis = 0)

def covariance(samples, mu_hat):
	n = samples.shape[0]
	d = samples.shape[1]
	mu_hat = mu_hat.reshape((d, 1))
	return (samples.T-mu_hat) @ (samples.T-mu_hat).T/n

if __name__ == '__main__':
	mu = [15, 5]
	sigmas = [[[20, 0], [0, 10]], [[20, 14], [14, 10]], [[20, -14], [-14, 10]]]
	for i, sigma in enumerate(sigmas):
		samples = np.random.multivariate_normal(mu, sigma, size=100)
		mu_hat = mean(samples)
		sigma_hat = covariance(samples, mu_hat)
		print('when simga = {}, the MLE mean is {}, and MLE covariance is
{}'.format(sigma, mu_hat, sigma_hat))
		print('the singular values for sigma matrix is
{}'.format(np.linalg.svd(sigma, compute_uv = False)))
		fig = plt.figure()
		plt.scatter(samples[:, 0], samples[:, 1])
		plt.savefig('sigma{}.jpg'.format(i))
```