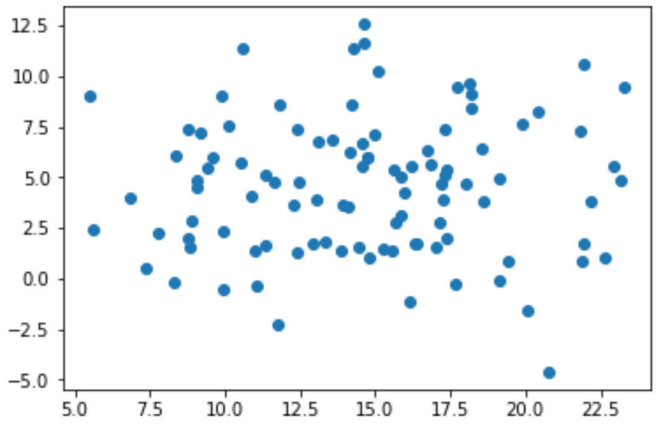
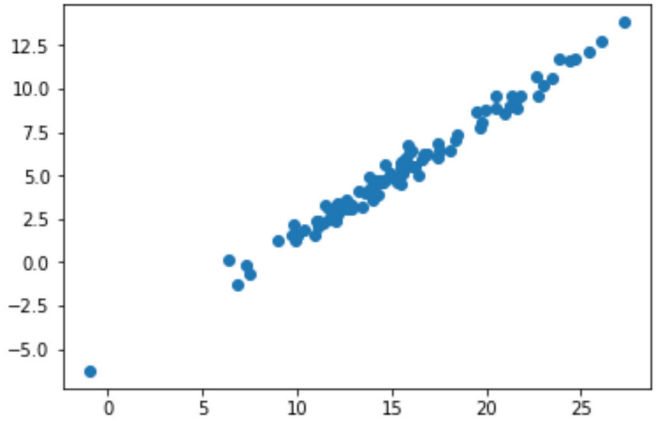
In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.mlab import bivariate_normal


# Question 2c

mu = [15, 5]
sigma = [[20, 0], [0, 10]]
samples = np.random.multivariate_normal(mu, sigma, size=100)
mu_estimate = np.mean(samples, axis=0).reshape(2,1)
sigma_estimate = 0
for i in range(100):
    deviation = samples[i, :].reshape(2,1) - mu_estimate
    sigma_estimate += np.matmul(deviation, deviation.T)
sigma_estimate /= 100
print("mu:\n", mu_estimate)
print("Sigma:\n", sigma_estimate)
plt.figure()
plt.scatter(samples[:, 0], samples[:, 1])
plt.show()

sigma = [[20, 14], [14, 10]]
samples = np.random.multivariate_normal(mu, sigma, size=100)
mu_estimate = np.mean(samples, axis=0).reshape(2,1)
sigma_estimate = 0
for i in range(100):
    deviation = samples[i, :].reshape(2,1) - mu_estimate
    sigma_estimate += np.matmul(deviation, deviation.T)
sigma_estimate /= 100
print("mu:\n", mu_estimate)
print("Sigma:\n", sigma_estimate)
plt.figure()
plt.scatter(samples[:, 0], samples[:, 1])
plt.show()

sigma = [[20, -14], [-14, 10]]
samples = np.random.multivariate_normal(mu, sigma, size=100)
mu_estimate = np.mean(samples, axis=0).reshape(2,1)
sigma_estimate = 0
for i in range(100):
    deviation = samples[i, :].reshape(2,1) - mu_estimate
    sigma_estimate += np.matmul(deviation, deviation.T)
sigma_estimate /= 100
print("mu:\n", mu_estimate)
print("Sigma:\n", sigma_estimate)
plt.figure()
plt.scatter(samples[:, 0], samples[:, 1])
plt.show()
```

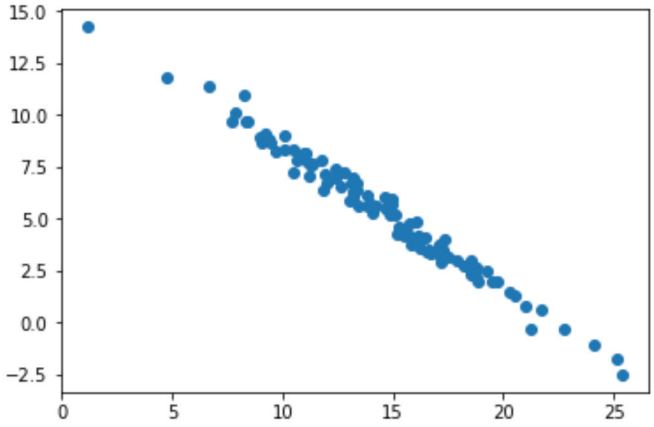
```
mu:
 [[ 14.62635369]
 [  4.53421849]]
Sigma:
 [[ 18.50848944   0.41721358]
 [  0.41721358  11.33299261]]
```



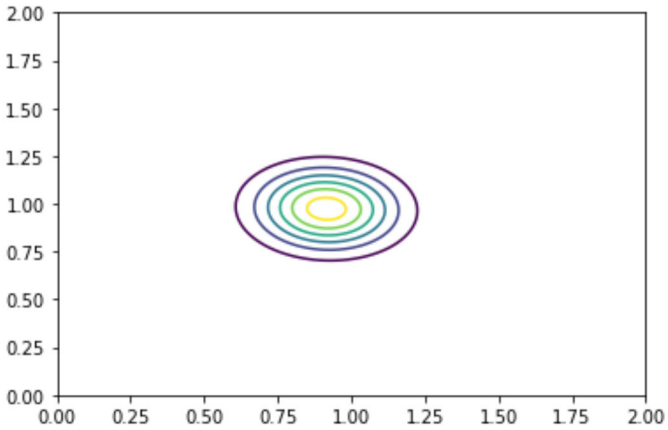

```
mu:
 [[ 15.35608833]
 [  5.22193569]]
Sigma:
 [[ 23.13563725  15.9046762 ]
 [ 15.9046762   11.12067857]]
```



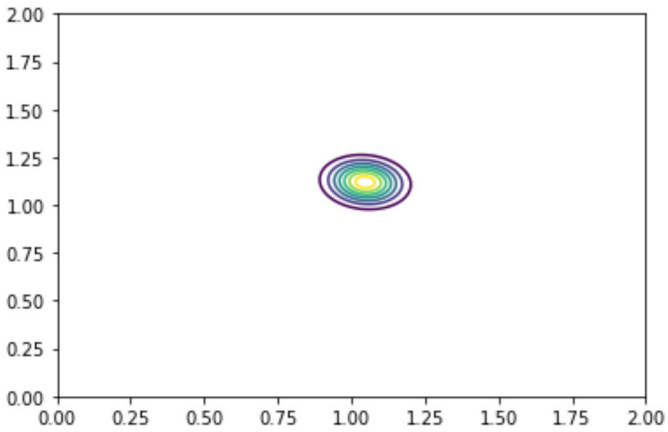

```
mu:
 [[ 14.54418001]
 [  5.3237607 ]]
Sigma:
 [[ 18.4894182  -12.85899727]
 [-12.85899727   9.11141132]]
```



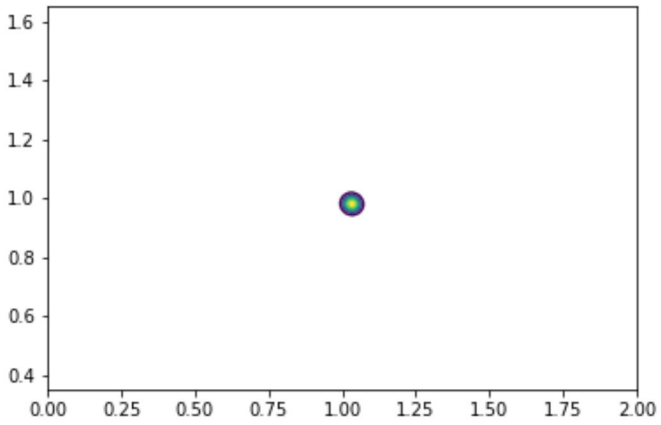
In [41]:
```
# Question 3d

mu_X = [0, 0]
sigma_X = [[5, 0], [0, 5]]
mu_w = [0, 0]
w_0 = np.linspace(0.5, 1.5, 100)
w_1 = np.linspace(0.5, 1.5, 100)
W0, W1 = np.meshgrid(w_0, w_1)

Sigma = [[1, 0], [0, 1]]
Prior = bivariate_normal(W0, W1, Sigma[0][0], Sigma[1][1], 0, 0, Sigma[0][1])
Posterior = np.ones( Prior.shape, "float" )
plt.figure()
n = 5
X = np.random.multivariate_normal(mu_X, sigma_X, size=n)
Z = np.random.normal(0, 1, size=n)
Y = X[:, 0] + X[:, 1] + Z
for i in range(len(w_0)):
    for j in range(len(w_1)):
        w = np.array([W0[i,j], W1[i,j]]).reshape(2, 1)
        likelihood = 1
        for k in range(n):
            X_vector = X[k, :].reshape(2, 1)
            likelihood *= np.exp(-0.5 * (Y[k] - np.matmul(w.T, X_vector)) ** 2)
        Posterior[i, j] = likelihood * Prior[i, j]
plt.contour(W0, W1, Posterior)
plt.axis((0, 2, 0, 2))
plt.show()
```



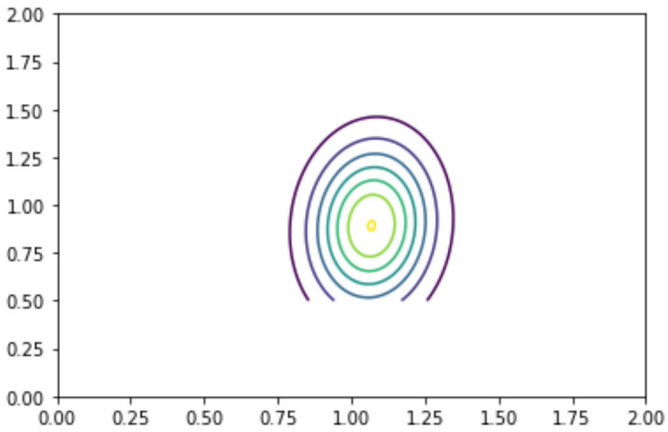
In [42]:
```
Sigma = [[1, 0], [0, 1]]
Prior = bivariate_normal(W0, W1, Sigma[0][0], Sigma[1][1], 0, 0, Sigma[0][1])
Posterior = np.ones( Prior.shape, "float" )
plt.figure()
n = 50
X = np.random.multivariate_normal(mu_X, sigma_X, size=n)
Z = np.random.normal(0, 1, size=n)
Y = X[:, 0] + X[:, 1] + Z
for i in range(len(w_0)):
    for j in range(len(w_1)):
        w = np.array([W0[i,j], W1[i,j]]).reshape(2, 1)
        likelihood = 1
        for k in range(n):
            X_vector = X[k, :].reshape(2, 1)
            likelihood *= np.exp(-0.5 * (Y[k] - np.matmul(w.T, X_vector)) ** 2)
        Posterior[i, j] = likelihood * Prior[i, j]
plt.contour(W0, W1, Posterior)
plt.axis((0, 2, 0, 2))
plt.show()
```



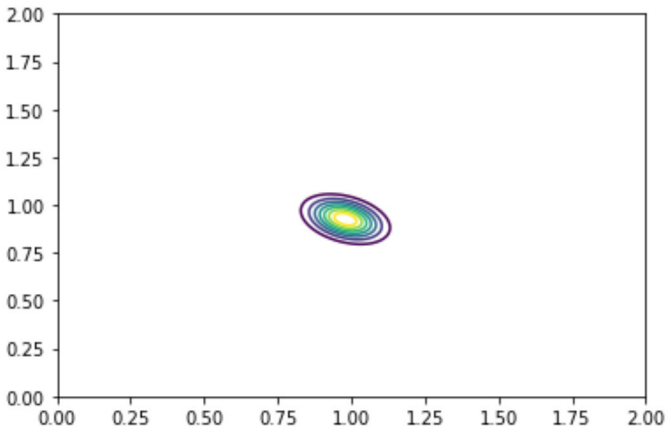
In [40]:
```
Sigma = [[1, 0], [0, 1]]
Prior = bivariate_normal(W0, W1, Sigma[0][0], Sigma[1][1], 0, 0, Sigma[0][1])
Posterior = np.ones( Prior.shape, "float" )
plt.figure()
n = 500
X = np.random.multivariate_normal(mu_X, sigma_X, size=n)
Z = np.random.normal(0, 1, size=n)
Y = X[:, 0] + X[:, 1] + Z
for i in range(len(w_0)):
    for j in range(len(w_1)):
        w = np.array([W0[i,j], W1[i,j]]).reshape(2, 1)
        likelihood = 1
        for k in range(n):
            X_vector = X[k, :].reshape(2, 1)
            likelihood *= np.exp(-0.5 * (Y[k] - np.matmul(w.T, X_vector)) ** 2)
        Posterior[i, j] = likelihood * Prior[i, j]
plt.contour(W0, W1, Posterior)
plt.axis((0, 2, 0, 2))
plt.show()
```



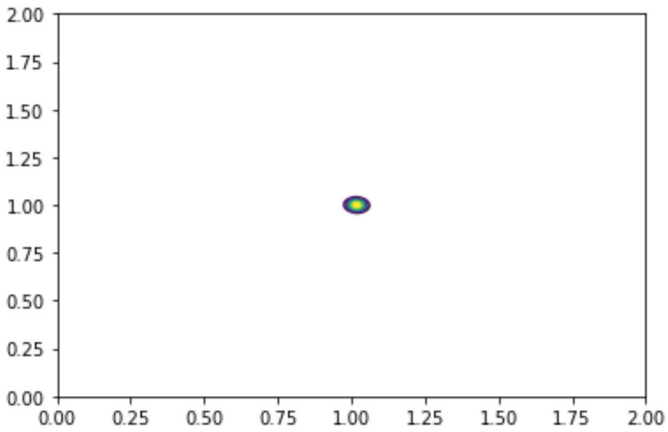
In [43]:
```
Sigma = [[1, 0.25], [0.25, 1]]
Prior = bivariate_normal(W0, W1, Sigma[0][0], Sigma[1][1], 0, 0, Sigma[0][1])
Posterior = np.ones( Prior.shape, "float" )
plt.figure()
n = 5
X = np.random.multivariate_normal(mu_X, sigma_X, size=n)
Z = np.random.normal(0, 1, size=n)
Y = X[:, 0] + X[:, 1] + Z
for i in range(len(w_0)):
    for j in range(len(w_1)):
        w = np.array([W0[i,j], W1[i,j]]).reshape(2, 1)
        likelihood = 1
        for k in range(n):
            X_vector = X[k, :].reshape(2, 1)
            likelihood *= np.exp(-0.5 * (Y[k] - np.matmul(w.T, X_vector)) ** 2)
        Posterior[i, j] = likelihood * Prior[i, j]
plt.contour(W0, W1, Posterior)
plt.axis((0, 2, 0, 2))
plt.show()
```



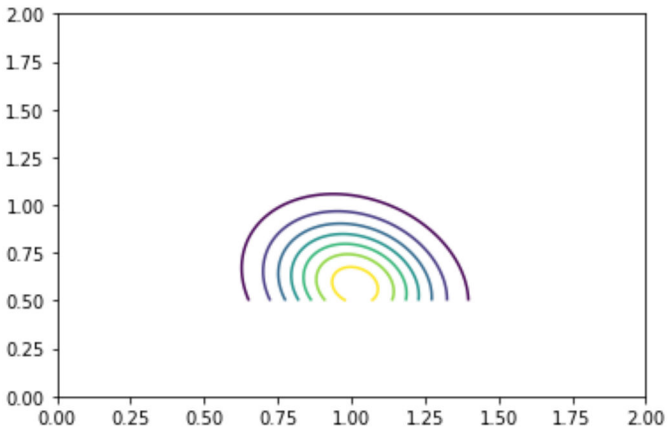
In [44]:
```
Sigma = [[1, 0.25], [0.25, 1]]
Prior = bivariate_normal(W0, W1, Sigma[0][0], Sigma[1][1], 0, 0, Sigma[0][1])
Posterior = np.ones( Prior.shape, "float" )
plt.figure()
n = 50
X = np.random.multivariate_normal(mu_X, sigma_X, size=n)
Z = np.random.normal(0, 1, size=n)
Y = X[:, 0] + X[:, 1] + Z
for i in range(len(w_0)):
    for j in range(len(w_1)):
        w = np.array([W0[i,j], W1[i,j]]).reshape(2, 1)
        likelihood = 1
        for k in range(n):
            X_vector = X[k, :].reshape(2, 1)
            likelihood *= np.exp(-0.5 * (Y[k] - np.matmul(w.T, X_vector)) ** 2)
        Posterior[i, j] = likelihood * Prior[i, j]
plt.contour(W0, W1, Posterior)
plt.axis((0, 2, 0, 2))
plt.show()
```



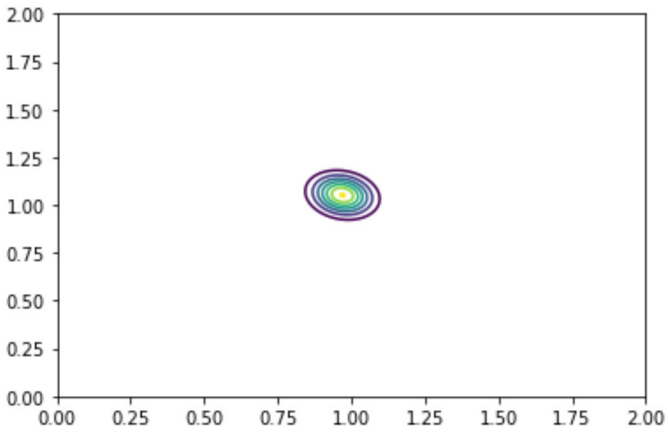
In [45]:
```
Sigma = [[1, 0.25], [0.25, 1]]
Prior = bivariate_normal(W0, W1, Sigma[0][0], Sigma[1][1], 0, 0, Sigma[0][1])
Posterior = np.ones( Prior.shape, "float" )
plt.figure()
n = 500
X = np.random.multivariate_normal(mu_X, sigma_X, size=n)
Z = np.random.normal(0, 1, size=n)
Y = X[:, 0] + X[:, 1] + Z
for i in range(len(w_0)):
    for j in range(len(w_1)):
        w = np.array([W0[i,j], W1[i,j]]).reshape(2, 1)
        likelihood = 1
        for k in range(n):
            X_vector = X[k, :].reshape(2, 1)
            likelihood *= np.exp(-0.5 * (Y[k] - np.matmul(w.T, X_vector)) ** 2)
        Posterior[i, j] = likelihood * Prior[i, j]
plt.contour(W0, W1, Posterior)
plt.axis((0, 2, 0, 2))
plt.show()
```



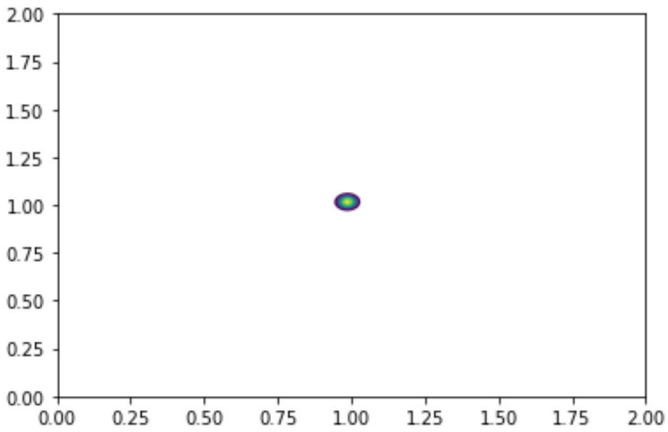
In [46]:
```
Sigma = [[1, 0.9], [0.9, 1]]
Prior = bivariate_normal(W0, W1, Sigma[0][0], Sigma[1][1], 0, 0, Sigma[0][1])
Posterior = np.ones( Prior.shape, "float" )
plt.figure()
n = 5
X = np.random.multivariate_normal(mu_X, sigma_X, size=n)
Z = np.random.normal(0, 1, size=n)
Y = X[:, 0] + X[:, 1] + Z
for i in range(len(w_0)):
    for j in range(len(w_1)):
        w = np.array([W0[i,j], W1[i,j]]).reshape(2, 1)
        likelihood = 1
        for k in range(n):
            X_vector = X[k, :].reshape(2, 1)
            likelihood *= np.exp(-0.5 * (Y[k] - np.matmul(w.T, X_vector)) ** 2)
        Posterior[i, j] = likelihood * Prior[i, j]
plt.contour(W0, W1, Posterior)
plt.axis((0, 2, 0, 2))
plt.show()
```



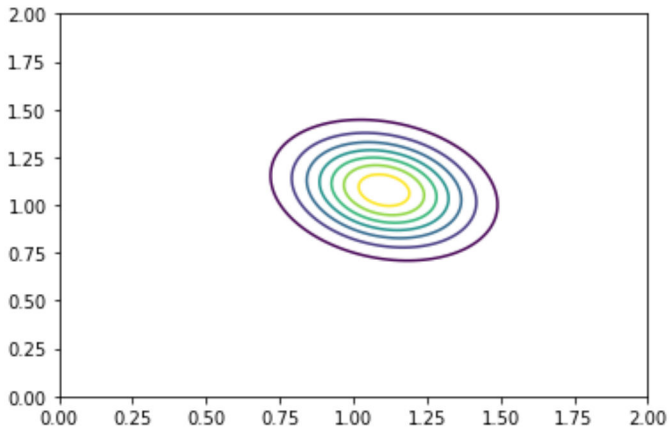
In [47]:
```
Sigma = [[1, 0.9], [0.9, 1]]
Prior = bivariate_normal(W0, W1, Sigma[0][0], Sigma[1][1], 0, 0, Sigma[0][1])
Posterior = np.ones( Prior.shape, "float" )
plt.figure()
n = 50
X = np.random.multivariate_normal(mu_X, sigma_X, size=n)
Z = np.random.normal(0, 1, size=n)
Y = X[:, 0] + X[:, 1] + Z
for i in range(len(w_0)):
    for j in range(len(w_1)):
        w = np.array([W0[i,j], W1[i,j]]).reshape(2, 1)
        likelihood = 1
        for k in range(n):
            X_vector = X[k, :].reshape(2, 1)
            likelihood *= np.exp(-0.5 * (Y[k] - np.matmul(w.T, X_vector)) ** 2)
        Posterior[i, j] = likelihood * Prior[i, j]
plt.contour(W0, W1, Posterior)
plt.axis((0, 2, 0, 2))
plt.show()
```



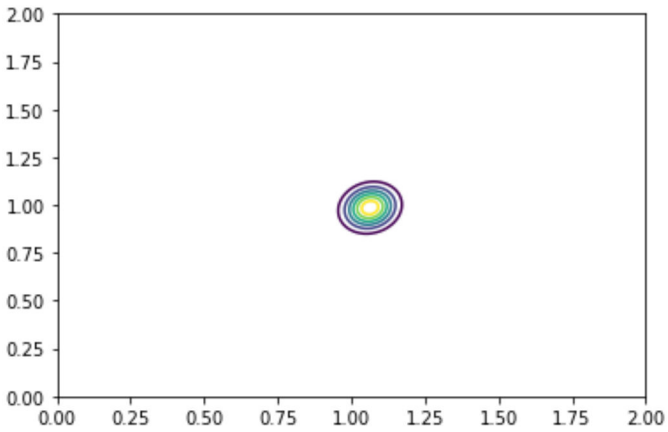
In [48]:
```
Sigma = [[1, 0.9], [0.9, 1]]
Prior = bivariate_normal(W0, W1, Sigma[0][0], Sigma[1][1], 0, 0, Sigma[0][1])
Posterior = np.ones( Prior.shape, "float" )
plt.figure()
n = 500
X = np.random.multivariate_normal(mu_X, sigma_X, size=n)
Z = np.random.normal(0, 1, size=n)
Y = X[:, 0] + X[:, 1] + Z
for i in range(len(w_0)):
    for j in range(len(w_1)):
        w = np.array([W0[i,j], W1[i,j]]).reshape(2, 1)
        likelihood = 1
        for k in range(n):
            X_vector = X[k, :].reshape(2, 1)
            likelihood *= np.exp(-0.5 * (Y[k] - np.matmul(w.T, X_vector)) ** 2)
        Posterior[i, j] = likelihood * Prior[i, j]
plt.contour(W0, W1, Posterior)
plt.axis((0, 2, 0, 2))
plt.show()
```



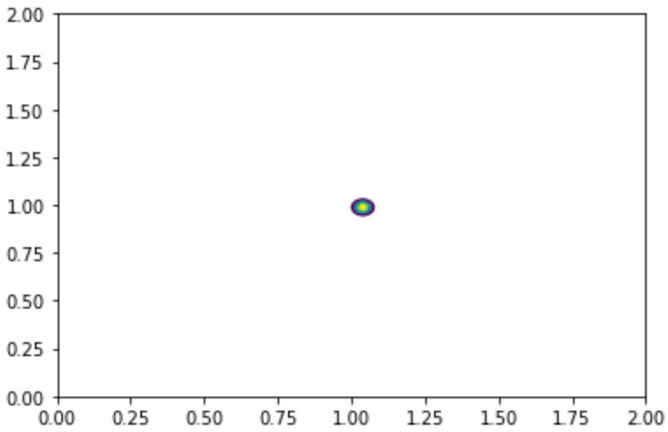
In [49]:
```
Sigma = [[1, -0.25], [-0.25, 1]]
Prior = bivariate_normal(W0, W1, Sigma[0][0], Sigma[1][1], 0, 0, Sigma[0][1])
Posterior = np.ones( Prior.shape, "float" )
plt.figure()
n = 5
X = np.random.multivariate_normal(mu_X, sigma_X, size=n)
Z = np.random.normal(0, 1, size=n)
Y = X[:, 0] + X[:, 1] + Z
for i in range(len(w_0)):
    for j in range(len(w_1)):
        w = np.array([W0[i,j], W1[i,j]]).reshape(2, 1)
        likelihood = 1
        for k in range(n):
            X_vector = X[k, :].reshape(2, 1)
            likelihood *= np.exp(-0.5 * (Y[k] - np.matmul(w.T, X_vector)) ** 2)
        Posterior[i, j] = likelihood * Prior[i, j]
plt.contour(W0, W1, Posterior)
plt.axis((0, 2, 0, 2))
plt.show()
```



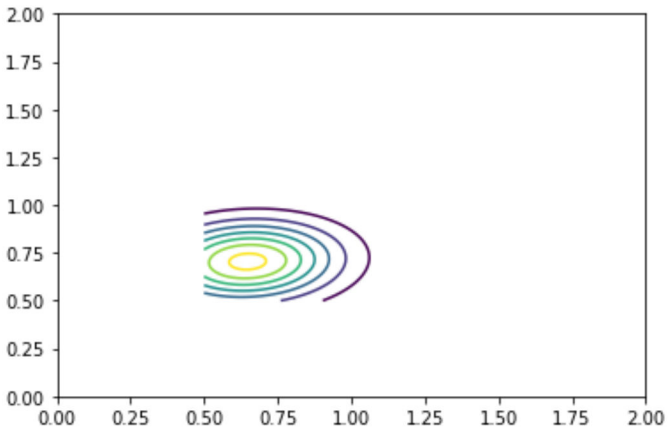
In [50]:
```
Sigma = [[1, -0.25], [-0.25, 1]]
Prior = bivariate_normal(W0, W1, Sigma[0][0], Sigma[1][1], 0, 0, Sigma[0][1])
Posterior = np.ones( Prior.shape, "float" )
plt.figure()
n = 50
X = np.random.multivariate_normal(mu_X, sigma_X, size=n)
Z = np.random.normal(0, 1, size=n)
Y = X[:, 0] + X[:, 1] + Z
for i in range(len(w_0)):
    for j in range(len(w_1)):
        w = np.array([W0[i,j], W1[i,j]]).reshape(2, 1)
        likelihood = 1
        for k in range(n):
            X_vector = X[k, :].reshape(2, 1)
            likelihood *= np.exp(-0.5 * (Y[k] - np.matmul(w.T, X_vector)) ** 2)
        Posterior[i, j] = likelihood * Prior[i, j]
plt.contour(W0, W1, Posterior)
plt.axis((0, 2, 0, 2))
plt.show()
```



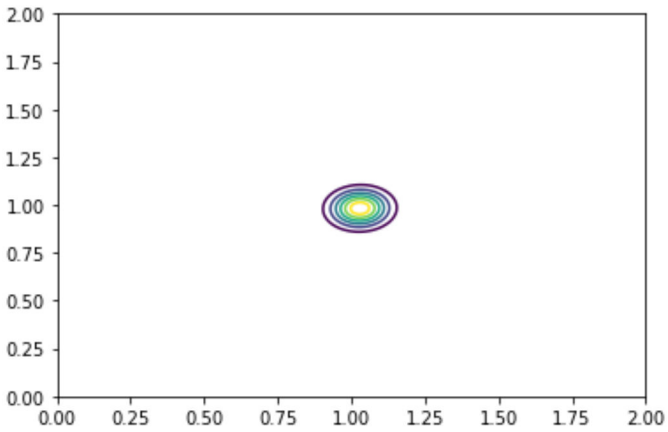
In [51]:
```
Sigma = [[1, -0.25], [-0.25, 1]]
Prior = bivariate_normal(W0, W1, Sigma[0][0], Sigma[1][1], 0, 0, Sigma[0][1])
Posterior = np.ones( Prior.shape, "float" )
plt.figure()
n = 500
X = np.random.multivariate_normal(mu_X, sigma_X, size=n)
Z = np.random.normal(0, 1, size=n)
Y = X[:, 0] + X[:, 1] + Z
for i in range(len(w_0)):
    for j in range(len(w_1)):
        w = np.array([W0[i,j], W1[i,j]]).reshape(2, 1)
        likelihood = 1
        for k in range(n):
            X_vector = X[k, :].reshape(2, 1)
            likelihood *= np.exp(-0.5 * (Y[k] - np.matmul(w.T, X_vector)) ** 2)
        Posterior[i, j] = likelihood * Prior[i, j]
plt.contour(W0, W1, Posterior)
plt.axis((0, 2, 0, 2))
plt.show()
```



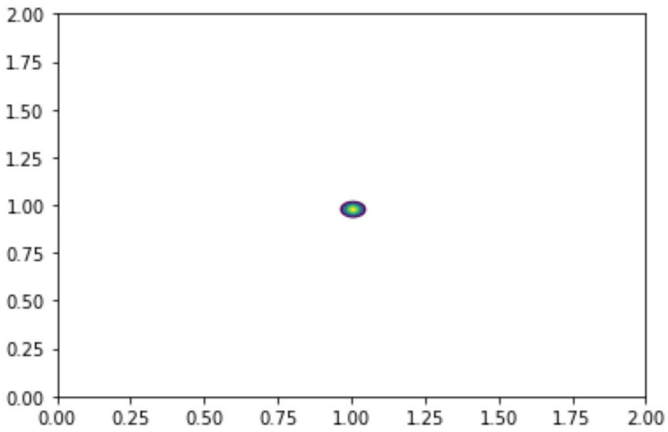
In [52]:
```
Sigma = [[1, -0.9], [-0.9, 1]]
Prior = bivariate_normal(W0, W1, Sigma[0][0], Sigma[1][1], 0, 0, Sigma[0][1])
Posterior = np.ones( Prior.shape, "float" )
plt.figure()
n = 5
X = np.random.multivariate_normal(mu_X, sigma_X, size=n)
Z = np.random.normal(0, 1, size=n)
Y = X[:, 0] + X[:, 1] + Z
for i in range(len(w_0)):
    for j in range(len(w_1)):
        w = np.array([W0[i,j], W1[i,j]]).reshape(2, 1)
        likelihood = 1
        for k in range(n):
            X_vector = X[k, :].reshape(2, 1)
            likelihood *= np.exp(-0.5 * (Y[k] - np.matmul(w.T, X_vector)) ** 2)
        Posterior[i, j] = likelihood * Prior[i, j]
plt.contour(W0, W1, Posterior)
plt.axis((0, 2, 0, 2))
plt.show()
```



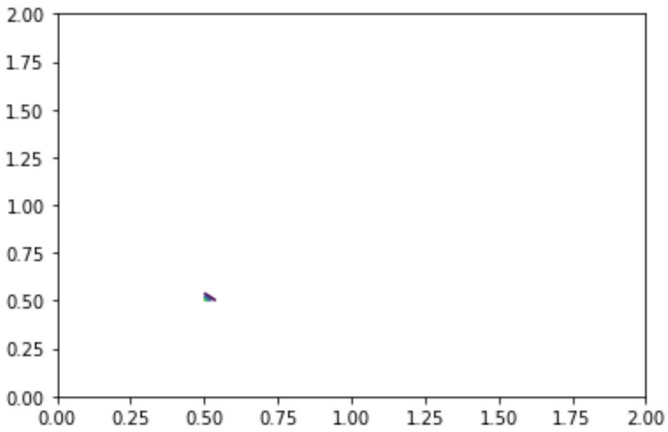
In [53]:
```
Sigma = [[1, -0.9], [-0.9, 1]]
Prior = bivariate_normal(W0, W1, Sigma[0][0], Sigma[1][1], 0, 0, Sigma[0][1])
Posterior = np.ones( Prior.shape, "float" )
plt.figure()
n = 50
X = np.random.multivariate_normal(mu_X, sigma_X, size=n)
Z = np.random.normal(0, 1, size=n)
Y = X[:, 0] + X[:, 1] + Z
for i in range(len(w_0)):
    for j in range(len(w_1)):
        w = np.array([W0[i,j], W1[i,j]]).reshape(2, 1)
        likelihood = 1
        for k in range(n):
            X_vector = X[k, :].reshape(2, 1)
            likelihood *= np.exp(-0.5 * (Y[k] - np.matmul(w.T, X_vector)) ** 2)
        Posterior[i, j] = likelihood * Prior[i, j]
plt.contour(W0, W1, Posterior)
plt.axis((0, 2, 0, 2))
plt.show()
```



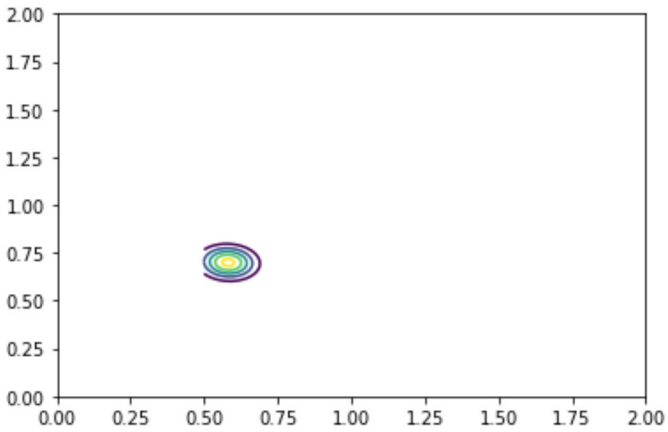
In [54]:
```
Sigma = [[1, -0.9], [-0.9, 1]]
Prior = bivariate_normal(W0, W1, Sigma[0][0], Sigma[1][1], 0, 0, Sigma[0][1])
Posterior = np.ones( Prior.shape, "float" )
plt.figure()
n = 500
X = np.random.multivariate_normal(mu_X, sigma_X, size=n)
Z = np.random.normal(0, 1, size=n)
Y = X[:, 0] + X[:, 1] + Z
for i in range(len(w_0)):
    for j in range(len(w_1)):
        w = np.array([W0[i,j], W1[i,j]]).reshape(2, 1)
        likelihood = 1
        for k in range(n):
            X_vector = X[k, :].reshape(2, 1)
            likelihood *= np.exp(-0.5 * (Y[k] - np.matmul(w.T, X_vector)) ** 2)
        Posterior[i, j] = likelihood * Prior[i, j]
plt.contour(W0, W1, Posterior)
plt.axis((0, 2, 0, 2))
plt.show()
```



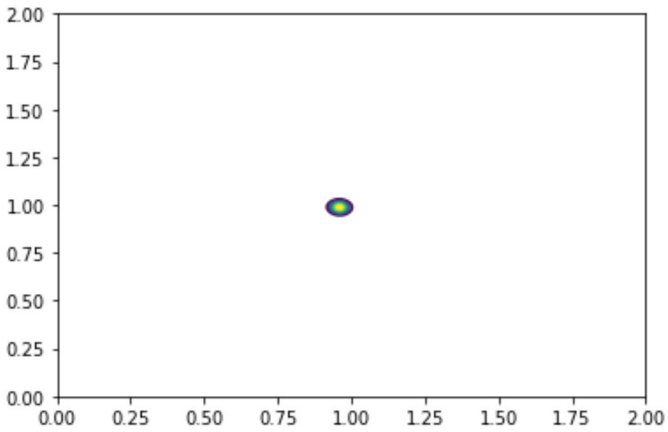
In [61]:
```
Sigma = [[0.1, 0], [0, 0.1]]
Prior = bivariate_normal(W0, W1, Sigma[0][0], Sigma[1][1], 0, 0, Sigma[0][1])
Posterior = np.ones( Prior.shape, "float" )
plt.figure()
n = 5
X = np.random.multivariate_normal(mu_X, sigma_X, size=n)
Z = np.random.normal(0, 1, size=n)
Y = X[:, 0] + X[:, 1] + Z
for i in range(len(w_0)):
    for j in range(len(w_1)):
        w = np.array([W0[i,j], W1[i,j]]).reshape(2, 1)
        likelihood = 1
        for k in range(n):
            X_vector = X[k, :].reshape(2, 1)
            likelihood *= np.exp(-0.5 * (Y[k] - np.matmul(w.T, X_vector)) ** 2)
        Posterior[i, j] = likelihood * Prior[i, j]
plt.contour(W0, W1, Posterior)
plt.axis((0, 2, 0, 2))
plt.show()
```



In [60]:
```
Sigma = [[0.1, 0], [0, 0.1]]
Prior = bivariate_normal(W0, W1, Sigma[0][0], Sigma[1][1], 0, 0, Sigma[0][1])
Posterior = np.ones( Prior.shape, "float" )
plt.figure()
n = 50
X = np.random.multivariate_normal(mu_X, sigma_X, size=n)
Z = np.random.normal(0, 1, size=n)
Y = X[:, 0] + X[:, 1] + Z
for i in range(len(w_0)):
    for j in range(len(w_1)):
        w = np.array([W0[i,j], W1[i,j]]).reshape(2, 1)
        likelihood = 1
        for k in range(n):
            X_vector = X[k, :].reshape(2, 1)
            likelihood *= np.exp(-0.5 * (Y[k] - np.matmul(w.T, X_vector)) ** 2)
        Posterior[i, j] = likelihood * Prior[i, j]
plt.contour(W0, W1, Posterior)
plt.axis((0, 2, 0, 2))
plt.show()
```



In [59]:
```
Sigma = [[0.1, 0], [0, 0.1]]
Prior = bivariate_normal(W0, W1, Sigma[0][0], Sigma[1][1], 0, 0, Sigma[0][1])
Posterior = np.ones( Prior.shape, "float" )
plt.figure()
n = 500
X = np.random.multivariate_normal(mu_X, sigma_X, size=n)
Z = np.random.normal(0, 1, size=n)
Y = X[:, 0] + X[:, 1] + Z
for i in range(len(w_0)):
    for j in range(len(w_1)):
        w = np.array([W0[i,j], W1[i,j]]).reshape(2, 1)
        likelihood = 1
        for k in range(n):
            X_vector = X[k, :].reshape(2, 1)
            likelihood *= np.exp(-0.5 * (Y[k] - np.matmul(w.T, X_vector)) ** 2)
        Posterior[i, j] = likelihood * Prior[i, j]
plt.contour(W0, W1, Posterior)
plt.axis((0, 2, 0, 2))
plt.show()
```



In [79]:
```
# Question 3e

mu_X = [0, 0]
sigma_X = [[5, 0], [0, 5]]
num = np.linspace(5, 200, 40, endpoint=True, dtype="int")
errors = np.zeros( num.shape, "float" )
errors_theor = np.zeros( num.shape, "float" )

X_test = np.random.multivariate_normal(mu_X, sigma_X, size=100)
Z_test = np.random.normal(0, 1, size=100)
Y_test = X_test[:, 0] + X_test[:, 1] + Z_test

for i in range(len(num)):
    X = np.random.multivariate_normal(mu_X, sigma_X, size=num[i])
    Z = np.random.normal(0, 1, size=num[i])
    Y = X[:, 0] + X[:, 1] + Z
    coeff = np.matmul( np.linalg.inv( np.matmul(X.T, X) ), np.matmul(X.T, Y) )
    errors_theor[i] = 5 * (coeff[0] - 1) ** 2 + 5 * (coeff[1] - 1) ** 2 + 1
    e = 0
    for j in range(100):
        e += (Y_test[j] - np.dot(X_test[j, :], coeff)) ** 2
    errors[i] = e / 100
plt.figure()
plt.plot(num, errors, label="MSE")
# plt.show()
# plt.figure()
plt.plot(num, errors_theor, label="theoretical error")
plt.legend()
plt.show()
```