CS289A_HW03_Prob2

February 27, 2017

```
In [18]: import math
         import numpy as np
         from matplotlib import pyplot as plt
         from mpl_toolkits.mplot3d import Axes3D
```

```
In [19]: def makesquarespace(xmin,xmax,ymin,ymax,delta):
         # Create arrays defining the space over which to evaluate density f'n Z

             xaxis = np.arange(xmin,xmax,delta)
             yaxis = np.arange(ymin,ymax,delta)
             X,Y = np.meshgrid(xaxis,yaxis)
             Z = np.empty_like(X)

             return(X,Y,Z)
```

```
In [20]: def calc_density_fn(X,Y,Z,mu,Sigma):
         # Given a space defined by X and Y, evaluate the density function Z for a
         # bivariate normal distribution with mean mu and covariance matrix Sigma
             n = len(mu)
             detCov = np.linalg.det(Sigma)
             SigInv = np.linalg.inv(Sigma)

             for i in range(len(X)):
                 for j in range(len(X[i])):
                     x = np.array([X[i,j],Y[i,j]])
                     Z[i,j] = 1/((2*math.pi)**(n/2)*detCov**(1/2))*math.exp(-0.5*np

             return Z
```

```
In [21]: def contourplots(X,Y,Z,filename=None):
         # Create contour plots with labeled isovalues

             fig = plt.figure(figsize=(10,10))
             cs = plt.contour(X,Y,Z)
             plt.contourf(X,Y,Z)
             cs = plt.clabel(cs, inline=1, fontsize=14, colors='white')
             plt.xlabel("$X_1$")
             plt.ylabel("$X_2$")
```

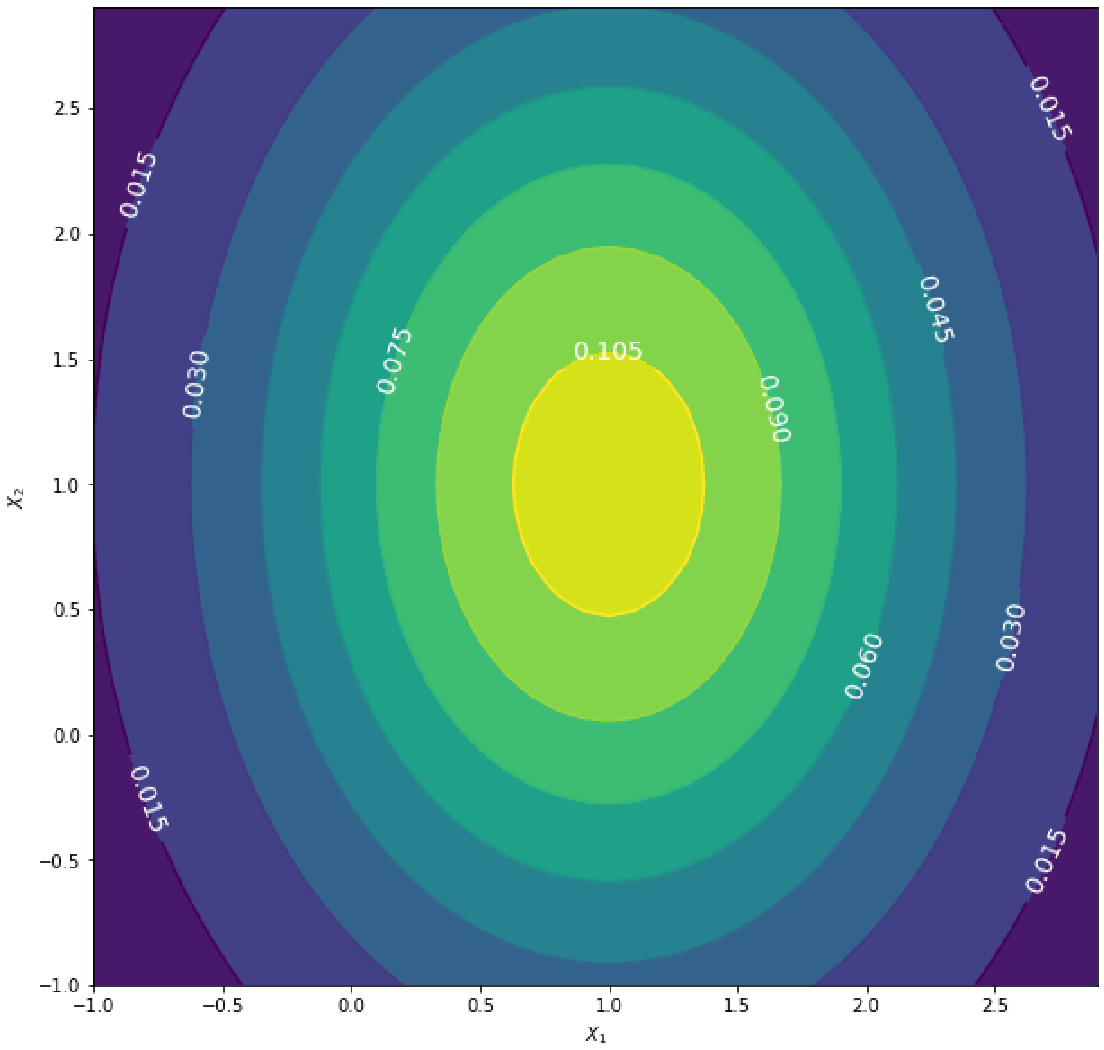
```
            if filename:
                plt.savefig(filename)
            plt.show()
```

```
In [22]: def SolveAndPlot(xmin,xmax,ymin,ymax,delta,mu,Sigma,figname=None):
             X1,X2,Z = makesquarespace(xmin,xmax,ymin,ymax,delta)
             Z = calc_density_fn(X1,X2,Z,mu,Sigma)
             contourplots(X1,X2,Z,figname)
```

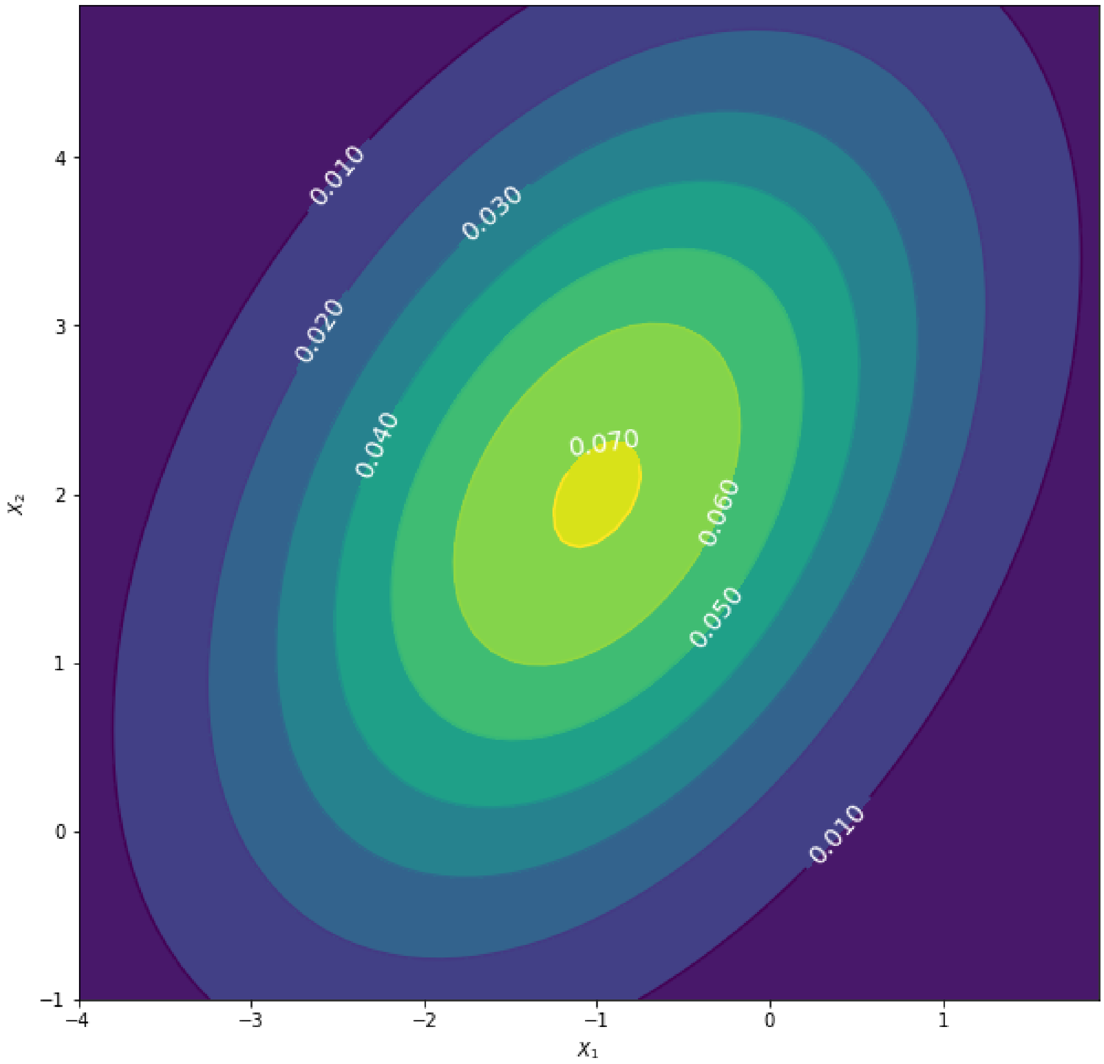
```
In [23]: def SolveSubAndPlot(xmin,xmax,ymin,ymax,delta,mu1,Sigma1,mu2,Sigma2,fignam
             X1,X2,Z = makesquarespace(xmin,xmax,ymin,ymax,delta)
             Y1,Y2 = np.empty_like(Z),np.empty_like(Z)
             Y1 = calc_density_fn(X1,X2,Y1,mu1,Sigma1)
             Y2 = calc_density_fn(X1,X2,Y2,mu2,Sigma2)
             Z = Y1-Y2
             contourplots(X1,X2,Z,figname)
```

```
In [24]: delta = 0.1
```

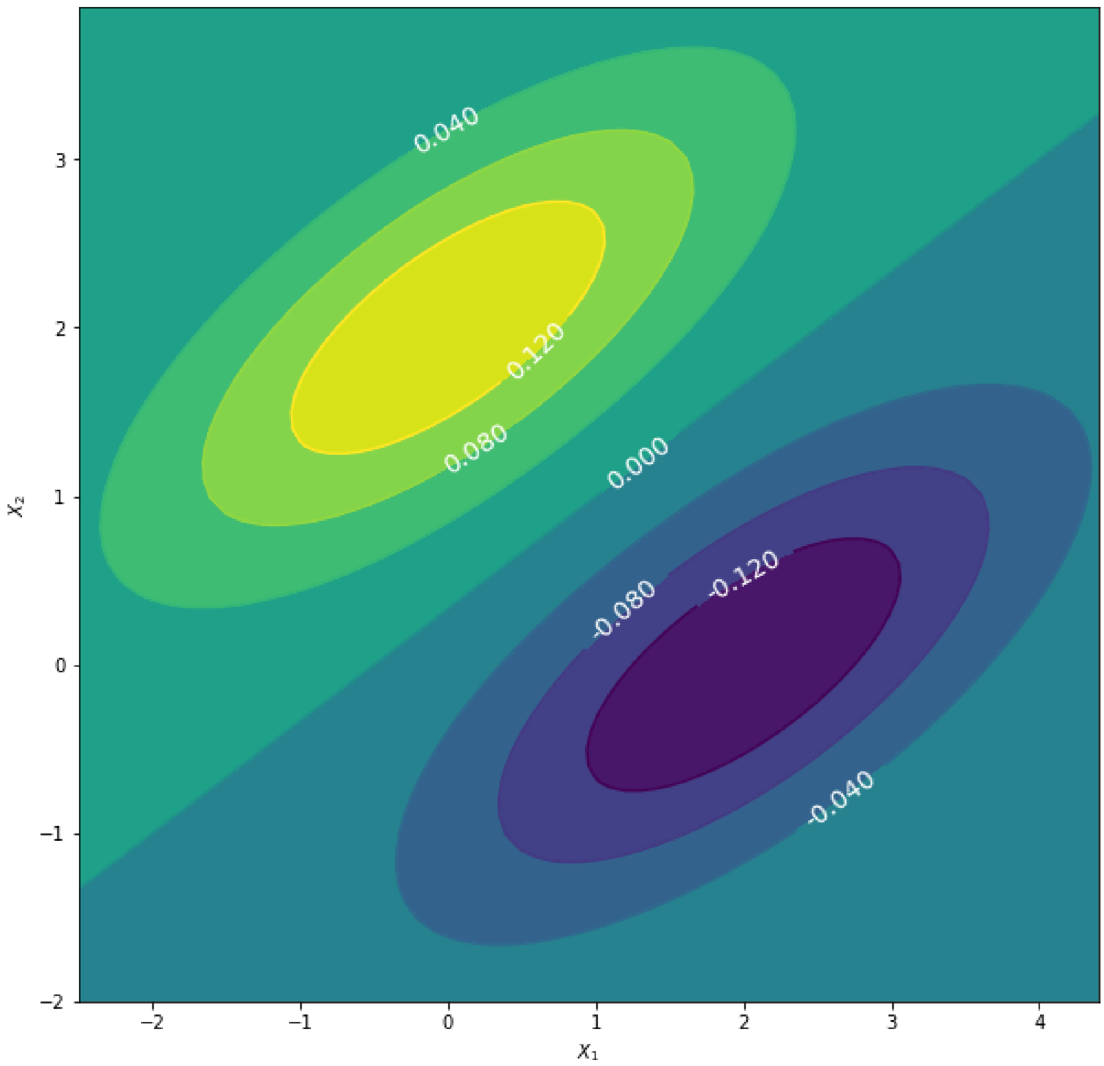
```
In [25]: # (2a)

         # Give the mean and covariance matrix (Sigma) of the multivariate distribu
         mu = np.array([1,1])
         Sigma = np.array([[1,0],[0,2]])

         figname = 'HW02_prob2a.jpg'
         SolveAndPlot(-1,3,-1,3,delta,mu,Sigma,figname)
```



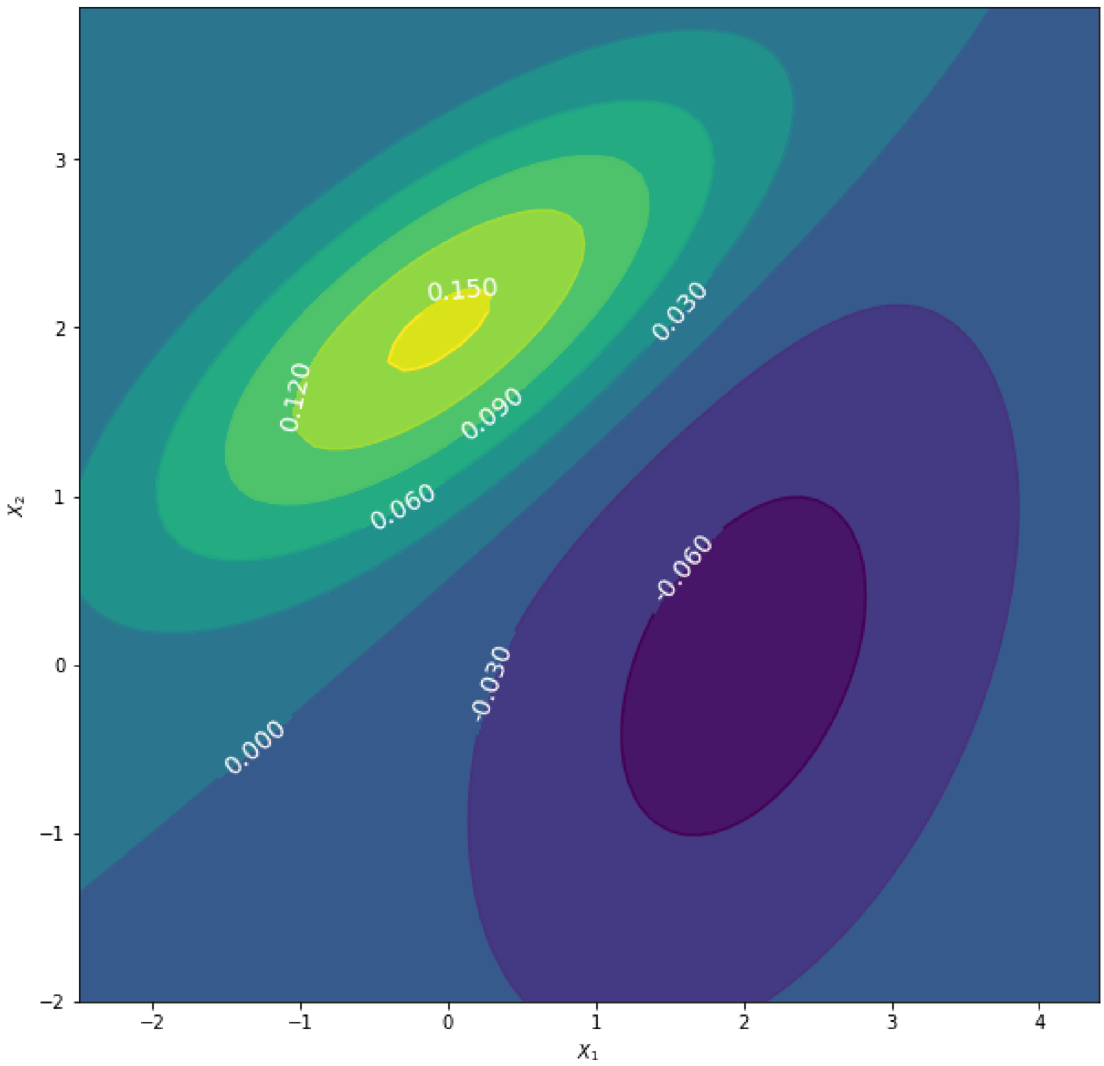
```
In [26]: # (2b)

         # Give the mean and covariance matrix (Sigma) of the multivariate distribu
         mu = np.array([-1,2])
         Sigma = np.array([[2,1],[1,3]])

         figname = 'HW02_prob2b.jpg'
         SolveAndPlot(-4,2,-1,5,delta,mu,Sigma,figname)
```



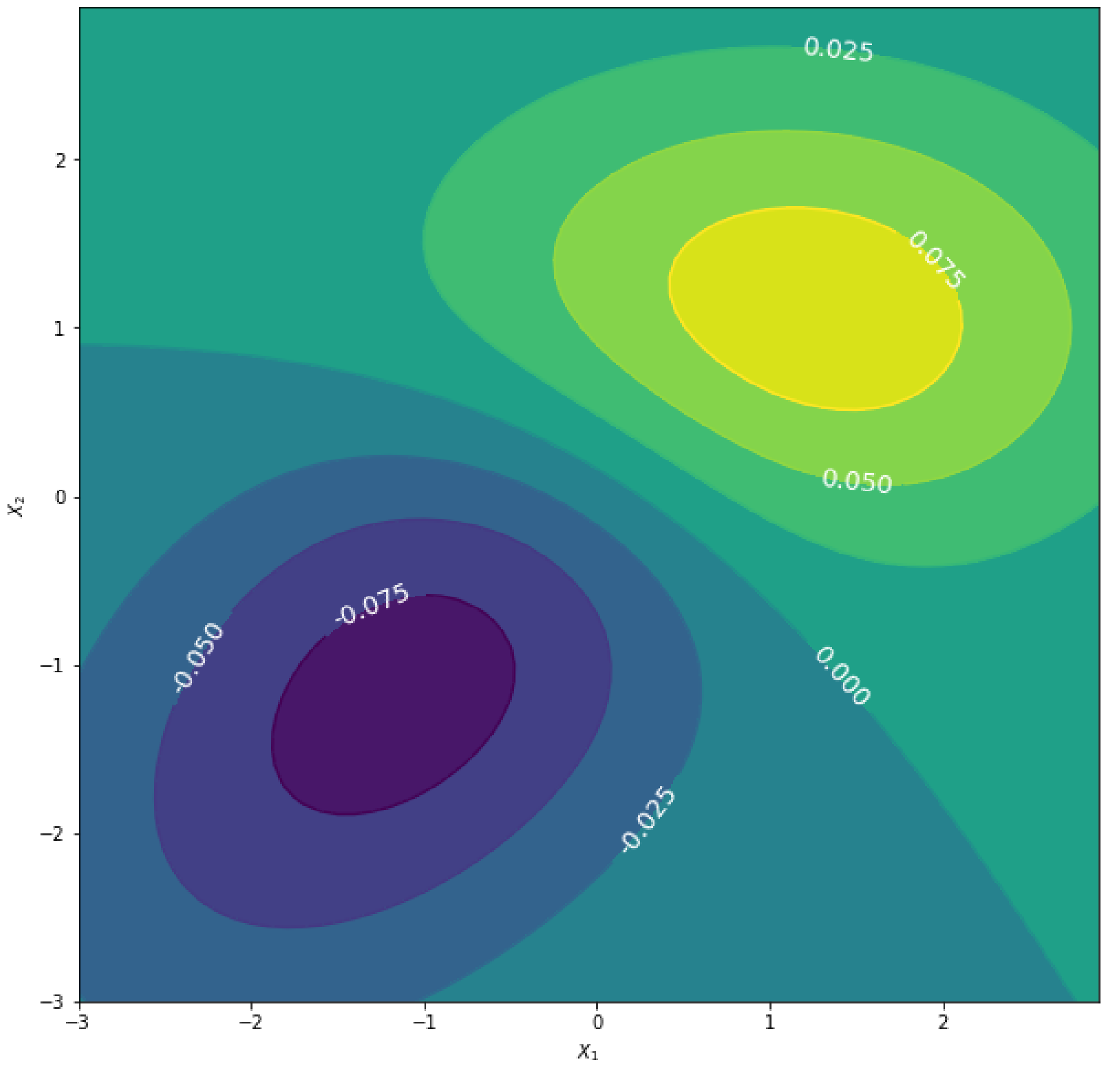
```
In [27]: # (2c)

         # Give the means and covariance matrices(Sigma1,Sigma2) of the multivariat
         mu1,mu2 = np.array([0,2]),np.array([2,0])
         Sigma1 = np.array([[2,1],[1,1]])
         Sigma2 = np.copy(Sigma1)

         figname = 'HW02_prob2c.jpg'
         SolveSubAndPlot(-2.5,4.5,-2,4,delta,mu1,Sigma1,mu2,Sigma2,figname)
```



```
In [28]: # (2d)

         # Give the means and covariance matrices(Sigma1,Sigma2) of the multivariat
         mu1,mu2 = np.array([0,2]),np.array([2,0])
         Sigma1,Sigma2 = np.array([[2,1],[1,1]]),np.array([[2,1],[1,3]])

         figname = 'HW02_prob2d.jpg'
         SolveSubAndPlot(-2.5,4.5,-2,4,delta,mu1,Sigma1,mu2,Sigma2,figname)
```



```
In [29]: # (2e)

         # Give the means and covariance matrices(Sigma1,Sigma2) of the multivariat
         mu1,mu2 = np.array([1,1]),np.array([-1,-1])
         Sigma1,Sigma2 = np.array([[2,0],[0,1]]),np.array([[2,1],[1,2]])

         figname = 'HW02_prob2e.jpg'
         SolveSubAndPlot(-3,3,-3,3,delta,mu1,Sigma1,mu2,Sigma2,figname)
```



```
In [ ]:
```