Homework:

The enclosed two files contain three-dimensional, artificial data. You can read them in using

```
pop1 = loadtxt('data1.csv', delimiter=','),
pop2 = loadtxt('data1.csv', delimiter=',').
```

The first 50 data points in each of the two files are our training set. Calculate the sample mean and correlation. There are two ways to do this. You can do it by hand as in irisbayes.py or more elegantly, you create a Pandas dataframe and then use the Pandas corr() function in order to get the correlation matrix:

```
df1 = pd.DataFrame(pop1[:50],columns=['a', 'b', 'c'])
s1 = df1.corr()
m1 = df1.mean()
```

You can then calculate the probability of any data point using the multivariate normal function.

You can get it from

```
from scipy.stats import multivariate_normal.
```

This gives you the possibility to calculate the probability of a point given that the point belongs to population 1 or given that the point belongs to population 2. The second halves of our data sets are our test sets. For each point, calculate the two probabilities. The larger ones determines whether we predict the point to belong to population 1 or to population 2.

Example:
The means are m1 and m2 and the correlation matrices are s1 and s2. I got with different numbers

```
>>> m1
a    1.142192
b    0.343738
c   -0.090215
dtype: float64
```

```
>>> s1
          a         b         c
a  1.000000  0.216875 -0.524015
b  0.216875  1.000000 -0.808495
c -0.524015 -0.808495  1.000000
```

Assume that I am seeing the point (0,0,0). I calculate the predictors (assuming that both populations are equally likely, which is of course the case):

```
>>> point = [0,0,0]
>>> multivariate_normal(mean = m1, cov=s1).pdf(point)
0.03355989878033516
>>> multivariate_normal(mean = m2, cov=s2).pdf(point)
0.001104025192403398
```

Because the first value is larger, I will predict that the point is from population 1.

Task:
Find out how often you mis-predict.