4.2

April 8, 2016

1 Задача 2

In [3]:
```
%matplotlib inline
import numpy as np
import math as mt
import matplotlib
import matplotlib.pyplot as plt
from pylab import *
from scipy.stats import *
from mpl_toolkits.mplot3d import Axes3D
```

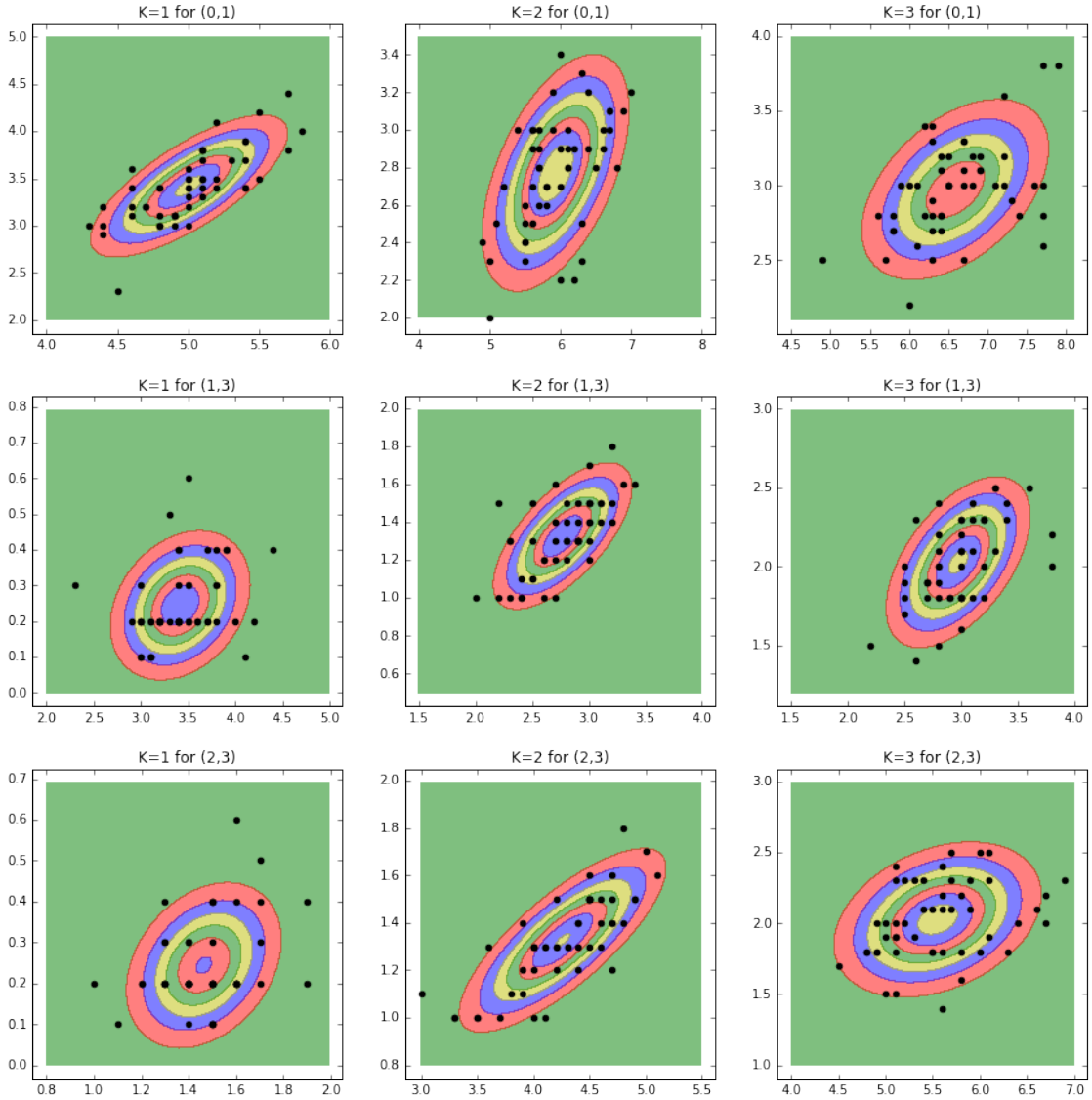
In [6]:
```
from sklearn.datasets import load_iris
data = load_iris()

means = []
sigmas = []

it = 0
for i in range(3):
    tmp = []
    while it < len(data['data']) and data['target'][it] == i:
        tmp.append(data['data'][it])
        it += 1
    means.append([mean(tmp[0]), mean(tmp[1]), mean(tmp[2]), mean(tmp[3])])
    sigmas.append(cov(tmp, rowvar=0))
means = np.array(means)
sigmas = np.array(sigmas)

print 'Матрицы ковариаций:'
print sigmas
print '\nСредние векторы:'
print means
```

```
Матрицы ковариаций:
[[[ 0.12424898  0.10029796  0.01613878  0.01054694]
  [ 0.10029796  0.14517959  0.01168163  0.01143673]
  [ 0.01613878  0.01168163  0.03010612  0.00569796]
  [ 0.01054694  0.01143673  0.00569796  0.01149388]]

 [[ 0.26643265  0.08518367  0.18289796  0.05577959]
  [ 0.08518367  0.09846939  0.08265306  0.04120408]
  [ 0.18289796  0.08265306  0.22081633  0.07310204]
  [ 0.05577959  0.04120408  0.07310204  0.03910612]]
```

```
 [[ 0.40434286  0.09376327  0.3032898   0.04909388]
  [ 0.09376327  0.10400408  0.07137959  0.04762857]
  [ 0.3032898   0.07137959  0.30458776  0.04882449]
  [ 0.04909388  0.04762857  0.04882449  0.07543265]]]

Средние векторы:
[[ 2.55   2.375  2.35   2.35 ]
 [ 4.075  3.9    4.1    3.275]
 [ 4.525  3.875  4.525  4.15 ]]
```

```
In [7]: coords = [[0,1], [1,3], [2,3]]  # Пары координат
        means_for_3 = np.zeros((3,3,2))  # Сюда запишу три вектора средних для
                                         # каждой пары координат
        sigma_for_3 = np.zeros((3,3,2,2))  # Сюда запишу три матрицы ковариации
                                           # для каждой компоненты для
                                           # каждой пары координат
        Xs = [[[],[],[]],[[],[],[]],[[],[],[]]]  # Сюда запишу массивы для каждой из координат
                                                 # и для каждой компоненты
        for i in range(3):
            ii, jj = coords[i]  # Номера первой и второй координат
            it = 0
            for j in range(3):
                while it < len(data['data']) and data['target'][it] == j:
                    Xs[i][j].append(data['data'][it][[ii,jj]])
                    it += 1
            Xs[i] = np.array(Xs[i])

            for j in range(3):
                # Здесь считаю матрицу ковариции и вектор средних
                sigma_for_3[i][j][0][0] = mean(Xs[i][j][:,0]*Xs[i][j][:,0])\
                    -(mean(Xs[i][j][:,0])*mean(Xs[i][j][:,0]))
                sigma_for_3[i][j][1][1] = mean(Xs[i][j][:,1]*Xs[i][j][:,1])\
                    -(mean(Xs[i][j][:,1])*mean(Xs[i][j][:,1]))
                sigma_for_3[i][j][0][1] = mean(Xs[i][j][:,0]*Xs[i][j][:,1])\
                    -(mean(Xs[i][j][:,0])*mean(Xs[i][j][:,1]))
                sigma_for_3[i][j][1][0] = sigma_for_3[i][j][0][1]

                means_for_3[i][j][0] = mean(Xs[i][j][:,0])
                means_for_3[i][j][1] = mean(Xs[i][j][:,1])

        # Рисую сетку графиков
        plt.close('all')
        ax = []
        f, ax = plt.subplots(3, 3)
        f.set_figheight(15)
        f.set_figwidth(15)

        # Здесь записаны пределы посроения сетки для расчета плостности
        limits = [[[4,6,2,5],[4,8,2,3.5],[4.5,8.1,2.1,4]],
                  [[2,5,0,0.8],[1.5,4,0.5,2],[1.5,4,1.2,3]],
                  [[0.8,2,0,0.7],[3,5.5,0.8,2],[4,7,1,3]]]

        # Рисую
```

```
for i in range(3):
    for j in range(3):
        x, y = np.mgrid[limits[i][j][0]:limits[i][j][1]:.01, \
                        limits[i][j][2]:limits[i][j][3]:.01]
        pos = np.empty(x.shape + (2,))
        pos[:, :, 0] = x; pos[:, :, 1] = y
        rv = multivariate_normal(means_for_3[i][j], sigma_for_3[i][j])

        ax[i][j].contourf(x, y, rv.pdf(pos), alpha=0.5, colors=('g','r','b','y'))
        ax[i][j].scatter(Xs[i][j][:,0],Xs[i][j][:,1], color='black')
        ax[i][j].set_title('K={} for ({},{})'.format(j+1,coords[i][0],\
                                                     coords[i][1]))
show()
```




```
In [18]: print len(data['data'])
         print hist(data['target'])
```

```
150
(array([ 50.,   0.,   0.,   0.,   0.,  50.,   0.,   0.,   0.,  50.]), array([ 0. ,  0.2,  0.4,  0.6,  0
```