



**《数据分析》实验报告**

**学 院**：

**班 级**：

**学 号**：

**姓 名**：

**指导教师**：

教 务 处

2022 年 4 月

题目1

**1. 回归分析**

**某销售公司将库存占用资金情况、广告投入的费用、员工薪酬以及销售额等方面的数据作了汇总,该公司试图根据这些数据找到销售额与其他变量之间的关系，以便进行销售额预测并为工作决策提供参考依据。**

1. **写入数据库文件，读取数据库文件数据 (略)**
2. **通过python\matlan\R读取数据库文件或csv文件 (pandas)**

**(3) 对获取得到的数据进行分析**

**1). 是否线性关系？请画出各自变量与因变量的散点图；**

**2). 建立并给出销售额的回归模型；给出各回归系数的95%的置信区间**

**3). 计算回归的拟合优度判定系数(多重可决系数)**

**4). 对回归方程做F检验，检验线性回归的显著性**

**5). 如果未来某月库存资金额为150万元，广告投入预算为45万元，员工薪酬总额为27万元，试根据建立的回归模型预测该月的销售额。**

**占用资金、广告投入、员工薪酬、销售额（单位：万元）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **第*n*个月** | **库存资金额(*x*1)** | **广告投入(*x*2)** | **员工薪酬总额(*x*3)** | **销售额(y)** |
| **1** | **75.2** | **30.6** | **21.1** | **1090.4** |
| **2** | **77.6** | **31.3** | **21.4** | **1133.7** |
| **3** | **80.7** | **33.9** | **22.9** | **1242.1** |
| **4** | **76** | **29.6** | **21.4** | **1003.2** |
| **5** | **79.5** | **32.5** | **21.5** | **1283.2** |
| **6** | **81.8** | **27.9** | **21.7** | **1012.2** |
| **7** | **98.3** | **24.8** | **21.5** | **1098.8** |
| **8** | **67.7** | **23.6** | **21** | **826.3** |
| **9** | **74** | **33.9** | **22.4** | **1003.3** |
| **10** | **151** | **27.7** | **24.7** | **1554.6** |
| **11** | **90.8** | **45.5** | **23.2** | **1199** |
| **12** | **102.3** | **42.6** | **24.3** | **1483.1** |
| **13** | **115.6** | **40** | **23.1** | **1407.1** |
| **14** | **125** | **45.8** | **29.1** | **1551.3** |
| **15** | **137.8** | **51.7** | **24.6** | **1601.2** |
| **16** | **175.6** | **67.2** | **27.5** | **2311.7** |
| **17** | **155.2** | **65** | **26.5** | **2126.7** |
| **18** | **174.3** | **65.4** | **26.8** | **2256.5** |

问题分析：

1.使用pandas库读取数据

2.通过sklearn库内置函数判断出每个自变量x都和y呈线性关系，使用matplotlib画出图。

3.通过statsmodels的线性模块建立模型并计算各评价参数。

4.输出回归系数的95%的置信区间，合优度判定系数，F检验参数，以及预测未来某月的销售额。

代码：

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

import statsmodels.api as sm

import statsmodels.stats.api as sms

import statsmodels.formula.api as smf

import scipy

plt.rcParams['font.sans-serif'] = ['SimHei']

plt.rcParams['axes.unicode\_minus'] = False

data1 = pd.read\_csv('1.csv', encoding='gbk')

data1 = data1.set\_index('第n个月')

# 输出线性相关图

plt.rcParams['axes.unicode\_minus']=False

sns.pairplot(data1, x\_vars=data1.columns[:-1], y\_vars=data1.columns[-1],kind="reg", size=5, aspect=0.7)

plt.show()

x = data1.iloc[:,0:-1]

y = data1.iloc[:,-1]

# 模型导入并运算

x\_model = sm.add\_constant(x)

model = sm.OLS(y,x\_model)

results = model.fit()

results.summary()

# 回归系数 y = a + bx1 + cx2 +dx3

results.params

# 95%回归系数置信区间

results.conf\_int(0.05)

# 拟合优度

results.rsquared\_adj

# f显著性检验

results.fvalue

results.f\_pvalue

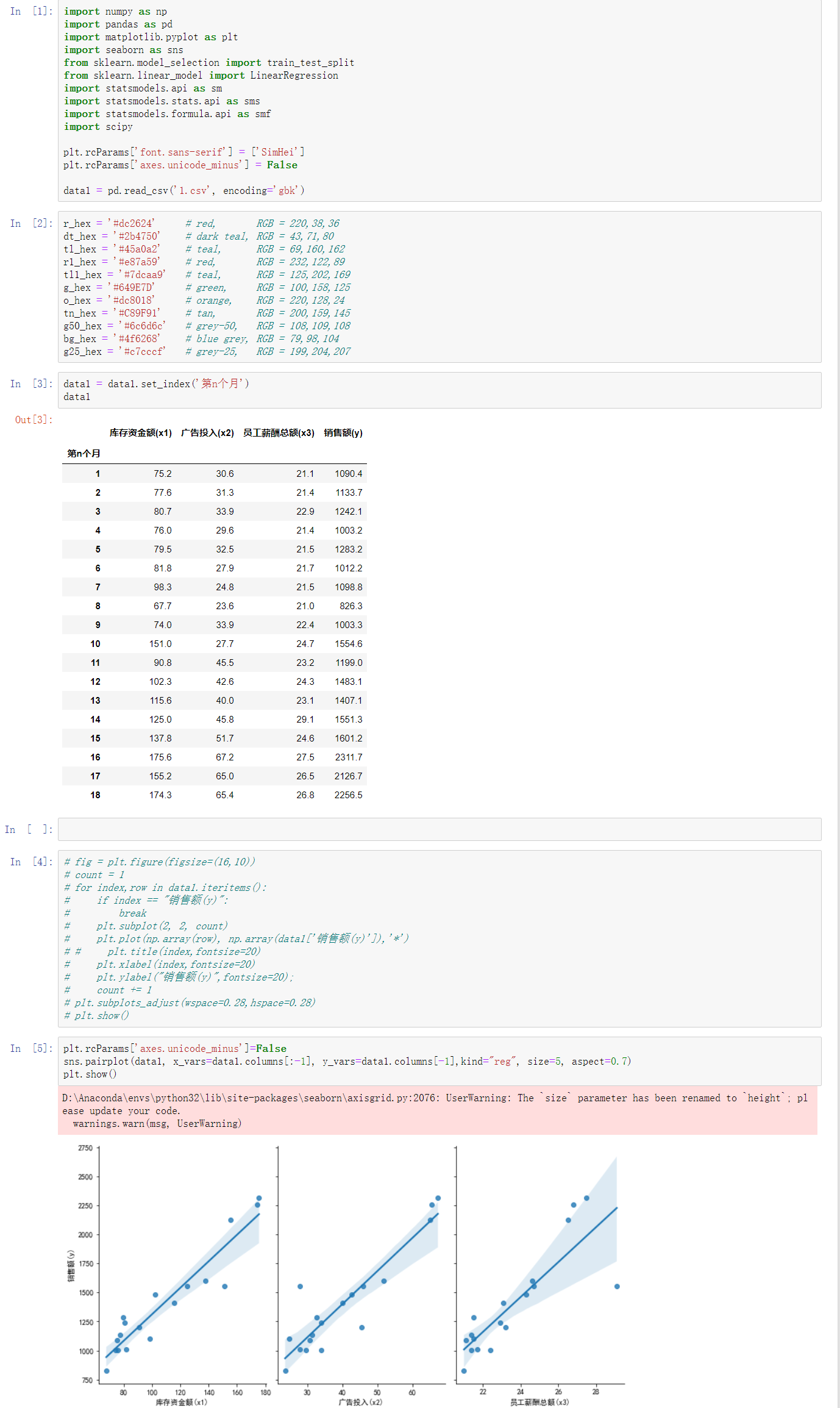
# 预测

x1,x2,x3 = 150,45,27

y = results.params[0]+results.params[1]\*x1+results.params[2]\*x2+results.params[3]\*x3

y

结果与分析：





实验过程记录：

比较顺畅，期间我尝试了使用sklearn机器学习进行训练建立模型，后来发现此方法无法给出模型的各种评价参数，后来改用了statsmodels进行多元线性回归，同时输出各种评价参数。最终结果当然是sklearn训练结果得到的更准确，但模型可解释性可评价性比不上后来用statsmodels进行多元线性回归得到的模型。

题目2

**1. 主成分分析**

**农业生态经济系统各区域单元的有关数据(x1—x9指标，21个样本)如下表1所示，请根据数据完成如下的数据分析任务，包括：**

1. **写入数据库文件，读取数据库文件数据 (略)**

**(2) 对获取得到的数据进行分析**

**1). 相关系数矩阵**

**2). 由相关系数矩阵计算特征值，****以及各个主成分的贡献率与累计贡献率**

**3). 按不低于85%的累计贡献率保留k个主成分，k=\_\_\_\_? 并给出k个主成分的在x1—x9指标下的表达式。**

**4). 如果认为第一主成分z1代表生态经济结构因素，请按照生态经济结构因素对**

**21个区域单元的样本进行由高到低的排名。**



1. **.定义评价系数： ，请按照综合评价**

**系数对样本进行排名。**

**表1 农业生态经济系统各区域单元的有关数据**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 样本序号 | x1：人口密度(人/km2) | x 2：人均耕地面积(ha) | x 3：森林覆盖率(%) | x 4：农民人均纯收入(元/人) | x 5：人均粮食产量 (kg/人) | x 6：经济作物占农作物播面比例(％) | x 7：耕地占土地面积比率(％) | x 8：果园与林地面积之比(％) | x 9：灌溉田占耕地面积之比(％) |
| 1 | 363.912 | 0.352 | 16.101 | 192.11 | 295.34 | 26.724 | 18.492 | 2.231 | 26.262 |
| 2 | 141.503 | 1.684 | 24.301 | 1752.35 | 452.26 | 32.314 | 14.464 | 1.455 | 27.066 |
| 3 | 100.695 | 1.067 | 65.601 | 1181.54 | 270.12 | 18.266 | 0.162 | 7.474 | 12.489 |
| 4 | 143.739 | 1.336 | 33.205 | 1436.12 | 354.26 | 17.486 | 11.805 | 1.892 | 17.534 |
| 5 | 131.412 | 1.623 | 16.607 | 1405.09 | 586.59 | 40.683 | 14.401 | 0.303 | 22.932 |
| 6 | 68.337 | 2.032 | 76.204 | 1540.29 | 216.39 | 8.128 | 4.065 | 0.011 | 4.861 |
| 7 | 95.416 | 0.801 | 71.106 | 926.35 | 291.52 | 8.135 | 4.063 | 0.012 | 4.862 |
| 8 | 62.901 | 1.652 | 73.307 | 1501.24 | 225.25 | 18.352 | 2.645 | 0.034 | 3.201 |
| 9 | 86.624 | 0.841 | 68.904 | 897.36 | 196.37 | 16.861 | 5.176 | 0.055 | 6.167 |
| 10 | 91.394 | 0.812 | 66.502 | 911.24 | 226.51 | 18.279 | 5.643 | 0.076 | 4.477 |
| 11 | 76.912 | 0.858 | 50.302 | 103.52 | 217.09 | 19.793 | 4.881 | 0.001 | 6.165 |
| 12 | 51.274 | 1.041 | 64.609 | 968.33 | 181.38 | 4.005 | 4.066 | 0.015 | 5.402 |
| 13 | 68.831 | 0.836 | 62.804 | 957.14 | 194.04 | 9.11 | 4.484 | 0.002 | 5.79 |
| 14 | 77.301 | 0.623 | 60.102 | 824.37 | 188.09 | 19.409 | 5.721 | 5.055 | 8.413 |
| 15 | 76.948 | 1.022 | 68.001 | 1255.42 | 211.55 | 11.102 | 3.133 | 0.01 | 3.425 |
| 16 | 99.265 | 0.654 | 60.702 | 1251.03 | 220.91 | 4.383 | 4.615 | 0.011 | 5.593 |
| 17 | 118.505 | 0.661 | 63.304 | 1246.47 | 242.16 | 10.706 | 6.053 | 0.154 | 8.701 |
| 18 | 141.473 | 0.737 | 54.206 | 814.21 | 193.46 | 11.419 | 6.442 | 0.012 | 12.945 |
| 19 | 137.761 | 0.598 | 55.901 | 1124.05 | 228.44 | 9.521 | 7.881 | 0.069 | 12.654 |
| 20 | 117.612 | 1.245 | 54.503 | 805.67 | 175.23 | 18.106 | 5.789 | 0.048 | 8.461 |
| 21 | 122.781 | 0.731 | 49.102 | 1313.11 | 236.29 | 26.724 | 7.162 | 0.092 | 10.078 |

问题分析：

1. 通过np.corrcoef()计算相关系数矩阵
2. 通过np.linalg.eig()计算矩阵特征向量和特征值以及各个主成分的贡献率与累计贡献率
3. 通过特征值的累计贡献度选出主成分
4. 按不低于85%的累计贡献率保留k个主成分

代码：

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn import preprocessing

plt.rcParams['font.sans-serif'] = ['SimHei']

plt.rcParams['axes.unicode\_minus'] = False

data1 = pd.read\_csv('2.csv', encoding='gbk')

data1 = data1.set\_index(['样本序号'])

data1

data2 = preprocessing.scale(data1)

data2

# 计算相关系数矩阵

data3 = np.around(np.corrcoef(data2.T),decimals=3)

data3

# 计算特征值，特征向量

featValue, featVec= np.linalg.eig(data3.T)

featValue, featVec

#求特征值的贡献度

gx = featValue/np.sum(featValue)

gx

#求特征值的累计贡献度

lg = np.cumsum(gx)

lg

#选出主成分 k = 3 三个特征值贡献率相加>85

k = [0, 1, 2]

k#选出主成分对应的特征向量矩阵

selectVec = np.matrix(featVec.T[k]).T

selectVe=selectVec\*(-1)

selectVec

df1=data1.values\*np.array(pd.DataFrame(selectVec).loc[:,0])

tf1=df1\*gx[0]/100

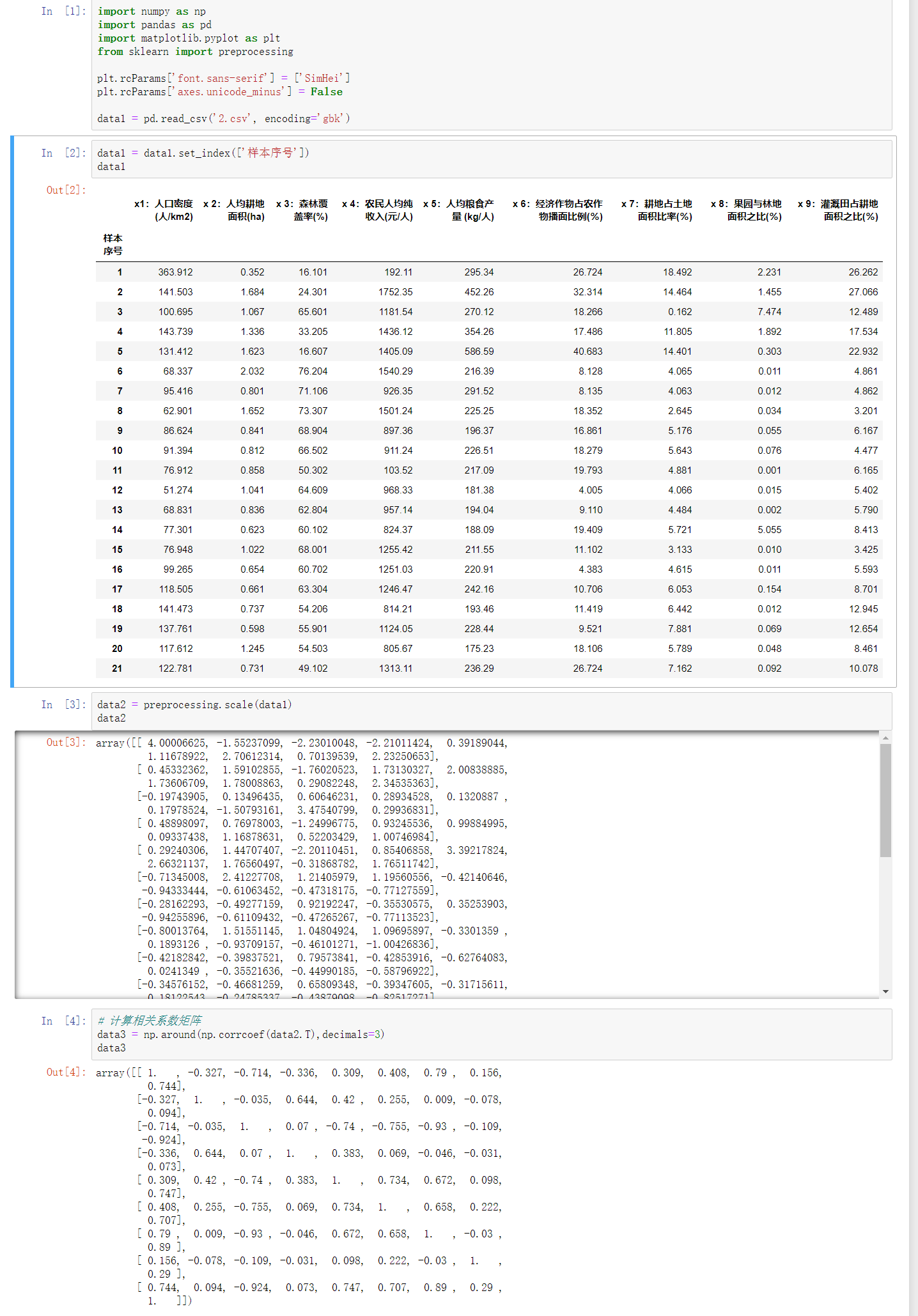
pd.DataFrame(tf1,index=data1.index, columns=data1.columns).sort\_values(["x1：人口密度(人/km2)"])

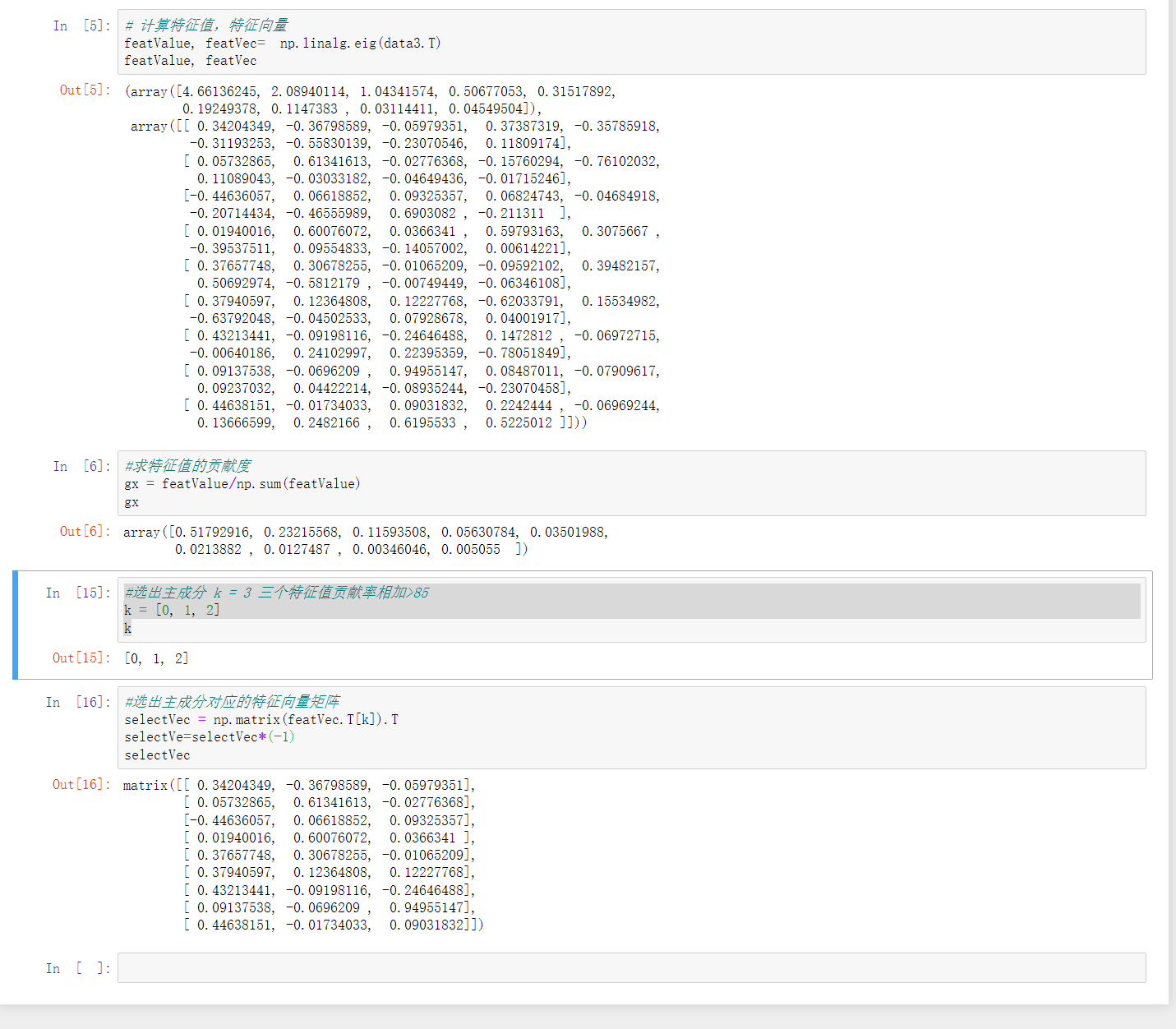
df=data2\*np.array(pd.DataFrame(selectVec).loc[:,0])+data2\*np.array(pd.DataFrame(selectVec).loc[:,1])+data2\*np.array(pd.DataFrame(selectVec).loc[:,2])

tf=df\*gx/sum(gx[0:3])

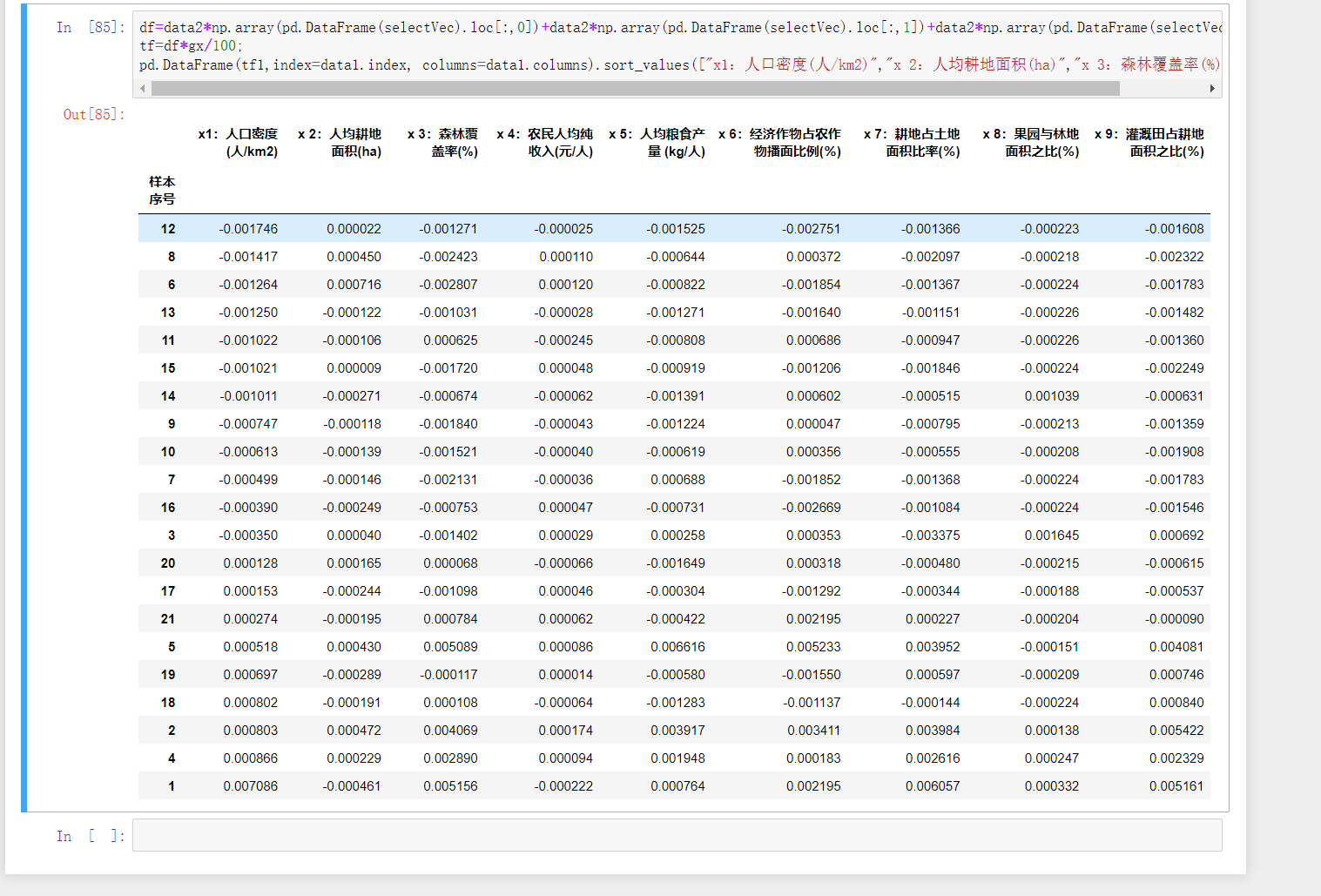
pd.DataFrame(tf1,index=data1.index, columns=data1.columns).sort\_values(["x1：人口密度(人/km2)","x 2：人均耕地面积(ha)","x 3：森林覆盖率(%)"])

结果与分析：









实验过程记录：

较为顺畅，在计算特征值之前进行了标准化。

题目3

1. **正态总体的Bayesian决策分析**

**某医院利用心电图检测来对人群进行划分，数据见表. “g=1”表示健康人，“g=2”表示主动脉硬化患者，“g=3”表示冠心病患者，X1, X2表示测得的心电图中表明心脏功能的两项不相关的指标. 某受试者心电图该两项指标的数据分别为380.20，9.08. 设先验概率按比例分配，进行Bayes判别，判定其归属.**

**提示：首先按照*α=*5%的对三个正态总体的协方差矩阵是否相等进行显著性检验**

**表 24人心电图数据**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **编号** | ***X*1** | ***X*2** | **g** | **编号** | ***X*1** | ***X*2** | **g** |
| **1**  **2**  **3**  **4**  **5**  **6**  **7**  **8**  **9**  **10**  **11**  **12** | **261.01**  **185.39**  **249.58**  **137.13**  **231.34**  **231.38**  **260.25**  **259.51**  **273.84**  **303.59**  **231.03**  **308.90** | **7.36**  **5.99**  **6.11**  **4.35**  **8.79**  **8.53**  **10.02**  **9.79**  **8.79**  **8.53**  **6.15**  **8.49** | **1**  **1**  **1**  **1**  **1**  **1**  **1**  **1**  **1**  **1**  **1**  **2** | **13**  **14**  **15**  **16**  **17**  **18**  **19**  **20**  **21**  **22**  **23**  **24** | **258.69**  **355.54**  **476.69**  **316.12**  **274.57**  **409.42**  **330.34**  **331.47**  **352.50**  **347.31**  **189.59**  **380.20** | **7.16**  **9.43**  **11.32**  **8.17**  **9.67**  **10.49**  **9.61**  **13.72**  **11.00**  **11.19**  **5.46**  **9.08** | **2**  **2**  **2**  **2**  **2**  **2**  **3**  **3**  **3**  **3**  **3**  **待判** |

问题分析：

1.根据g列分类出三类人

2.根据三类人数据计算三个正态总体的协方差矩阵

3.判断协方差矩阵是否相等，显著性检验

4. 设先验概率按比例分配，进行Bayes判别，判定其归属

代码：

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from math import log

from numpy.linalg import \*

from scipy.stats.distributions import chi2

data1 = pd.read\_excel("心电图.xlsx")

A = data1[0:-1]

A

B = np.array(data1[-1:])[0][1:-1]

B

g1 = A[0:11]

g2 = A[11:18]

g3 = A[18:]

n = 23

k = 3

p = 2

n1 = 11

n2 = 7

n3 = 5

f = p\*(p+1)\*(k-1)/2

d = (2\*pow(p, 2)+3\*p-1)\*(1/(n-1)+1/(n2-1)+1/(n3-1)-1/(n-k))/(6\*(p+1)\*(k-1))

p1 = n1/n

p2 = n2/n

p3 = n3/n

m1 = np.mean(g1)

m2 = np.mean(g2)

m3 = np.mean(g3)

s1 = np.cov(np.array(g1["X1"]),np.array(g1["X2"]))

s2 = np.cov(np.array(g2["X1"]),np.array(g2["X2"]))

s3 = np.cov(np.array(g3["X1"]),np.array(g3["X2"]))

s = ((n1-1)\*s1+(n2-1)\*s2+(n3-1)\*s3)/(n-k)

M = (n-k)\*log(det(s))-((n1-1)\*log(det(s1))+(n2-1)\*log(det(s2))+(n3-1)\*log(det(s3)))

T = (1-d)\*M

C = chi2.ppf(0.95, f)

if T < C:

print("协方差相等")

else:

print("协方差不相等")

x = B

x

m1 = [m1.loc["X1"],m1.loc["X2"]]

m2 = [m2.loc["X1"],m2.loc["X2"]]

m3 = [m3.loc["X1"],m3.loc["X2"]]

tp1 = m1\*inv(s)\*x-m1\*inv(s)\*np.array(m1).reshape(1,2)/2+log(p1)

tp2 = m2\*inv(s)\*x-m2\*inv(s)\*np.array(m2).reshape(1,2)/2+log(p2)

tp3 = m3\*inv(s)\*x-m3\*inv(s)\*np.array(m3).reshape(1,2)/2+log(p3)

tps = np.array([tp1, tp2, tp3])

tps

i = 1

for tp in tps:

if tp.all() == np.max(tps.all(), axis = 0):

print(i)

i += 1

结果与分析：





实验过程记录：

比较顺畅，通过python将数学公式表达出来进行运算，锻炼了我各类python函数的应用能力。

题目4

**k-means 彩色图像前景背景分割**

**图片：一只遥望大海的小狗，此图为m x n像素的彩色图片，每个像素可以表示为三维向量（分别对应彩色图像中的红色、绿色和蓝色通道）**

**(1) 使用K-means算法对图像进行分割, 将图片分割为合适的背景区域（三个）和前景区域（小狗）。**

**(2) 尝试改变K的取值，并给出分割结果，总结K对分割结果的影响。**

**(3) 尝试不同的初始聚类中心的选取方式，在实验报告上呈现出你认为比较好的分割结果**





问题分析：

需要将图像分成Iab空间中的a和b分量，然后划分分割区域为4，再合并分割后的图像。

代码：

A1=imread('1.png');

I\_lab=rgb2lab(A1);

I\_rgb=A1;

A2=imread('2.png');

a1=double(rgb2lab(A1));

a2=double(rgb2lab(A2));

%进行K-mean聚类将图像分割成2个区域

ab =double(I\_lab(:,:,2:3));

%取出lab空间的a分量和b分量

nrows= size(ab,1);

ncols= size(ab,2);

ab =reshape(ab,nrows\*ncols,2);

nColors= 4; %分割的区域个数为4

[cluster\_idx,cluster\_center] =kmeans(ab,nColors,'distance','sqEuclidean','Replicates',2); %重复聚类2次

pixel\_labels= reshape(cluster\_idx,nrows,ncols);

%显示分割后的各个区域

segmented\_images= cell(1,4);

rgb\_label= repmat(pixel\_labels,[1 1 3]);

for k= 1:nColors

color = I\_rgb;

color(rgb\_label ~= k) = 0;

segmented\_images{k} = color;

end

figure(),imshow(segmented\_images{1}),title('分割结果——区域1');

figure(),imshow(segmented\_images{2}),title('分割结果——区域2');

figure(),imshow(segmented\_images{3}),title('分割结果——区域3');

figure(),imshow(segmented\_images{4}),title('分割结果——区域4');

%使分割后的图像在一个图中显示

m=uint8(rgb\_label);

for i=1:69

for j=1:97

if m(i,j,1)==1

m(i,j,1)=255;

m(i,j,2)=0;

m(i,j,3)=0;

end

if m(i,j,1)==2

m(i,j,1)=256;

m(i,j,2)=256;

m(i,j,3)=0;

end

if m(i,j,1)==3

m(i,j,1)=0;

m(i,j,2)=0;

m(i,j,3)=255;

end

if m(i,j,1)==4

m(i,j,1)=0;

m(i,j,2)=128;

m(i,j,3)=0;

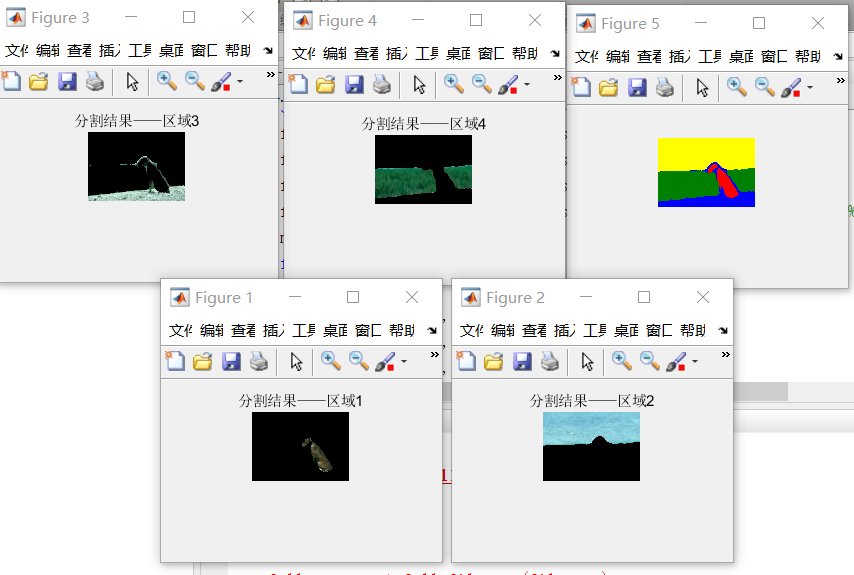
end

end

end

figure(),imshow(m)

结果与分析：



实验过程记录：

实验过程较为顺畅，图像处理方面python需要多个第三方库配合，在matlab中直接使用内置函数即可，更为方便，所以最终使用matlab完成此项实验。