## 实验一 《多源数据集成、清洗和统计》

**组长：卢学文**

**组员：卢学文、 夏旭、 李鸿洋**

#### 题目

广州大学某班有同学100人，现要从两个数据源汇总学生数据。第一个数据源在数据库中，第二个数据源在txt文件中，两个数据源课程存在缺失、冗余和不一致性，请用C/C++/Java程序实现对两个数据源的一致性合并以及每个学生样本的数值量化。

* 数据库表：ID (int), 姓名(string), 家乡(string:限定为Beijing / Guangzhou / Shenzhen / Shanghai), 性别（string:boy/girl）、身高（float:单位是cm)）、课程1成绩（float）、课程2成绩（float）、...、课程10成绩(float)、体能测试成绩（string：bad/general/good/excellent）；其中课程1-课程5为百分制，课程6-课程10为十分制。
* txt文件：ID(string：6位学号)，性别（string:male/female）、身高（string:单位是m)）、课程1成绩（string）、课程2成绩（string）、...、课程10成绩(string)、体能测试成绩（string：差/一般/良好/优秀）；其中课程1-课程5为百分制，课程6-课程10为十分制。

#### 参考

数据库中Stu表数据

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Name | City | Gender | Height | C1 | ... | C10 | Constitution |
| 1 | Sun | Beijing | boy | 160 | 87 |  | 9 | good |
| 2 | Zhu | Shenzhen | girl | 177 | 66 |  | 8 | excellent |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |

student.txt中

|  |
| --- |
| ID Name City Gender Height C1 。。。 C10 Constitution  202001 Sun Beijing male 180 87 。。。 9 good  202003 Tang Hanghai male 156 91 。。。 10 general  ... ... ... .. ... .. ... ... ... |

两个数据源合并后读入内存，并统计：

1. 学生中家乡在Beijing的所有课程的平均成绩。
2. 学生中家乡在广州，课程1在80分以上，且课程9在9分以上的男同学的数量。(备注：该处做了修正，课程10数据为空，更改为课程9)
3. 比较广州和上海两地女生的平均体能测试成绩，哪个地区的更强些？
4. 学习成绩和体能测试成绩，两者的相关性是多少？（九门课的成绩分别与体能成绩计算相关性）

#### 提示

参考数据结构：

Student{

int id;

string id;

vector<float> data;

}

可能用到的公式：

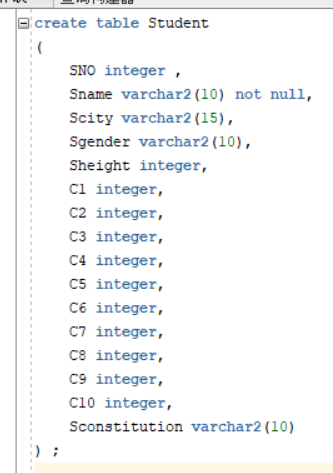
|  |  |
| --- | --- |
| 均值公式 |  |
| 协方差公式 |  |
| z-score规范化 |  |
| 数组A和数组B的相关性 | 这里A=[a1, a2,...ak,..., an],  B=[b1, b2,...bk,..., bn],  mean(A)代表A中元素的平均值  std是标准差，即对协方差的开平方。  点乘的定义： |

注意：计算部分不能调用库函数；画图/可视化显示可以用可视化API或工具实现。

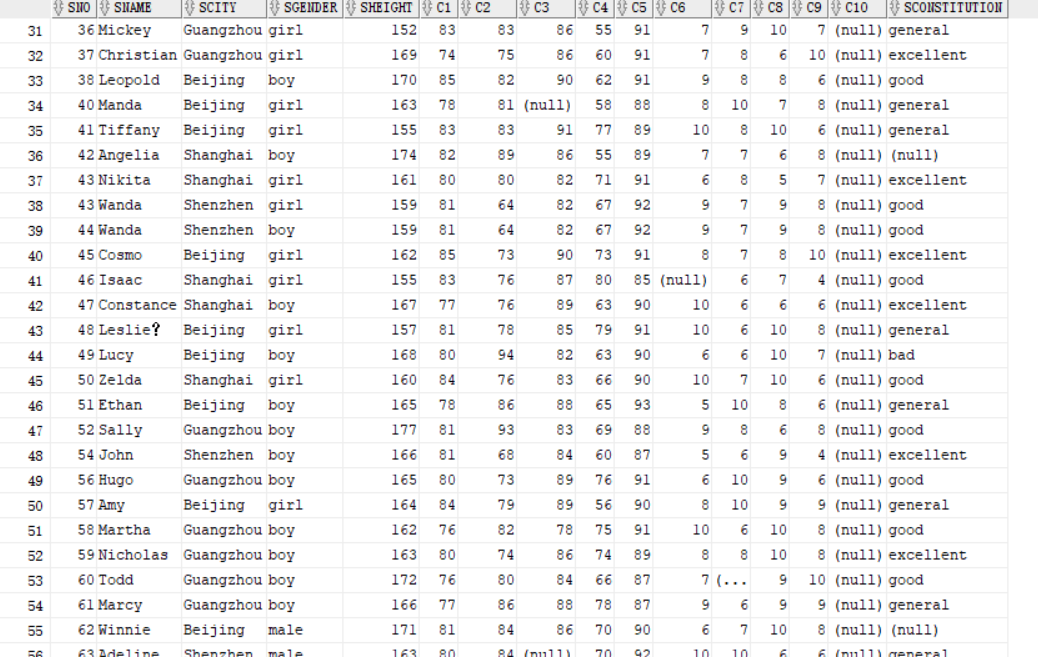
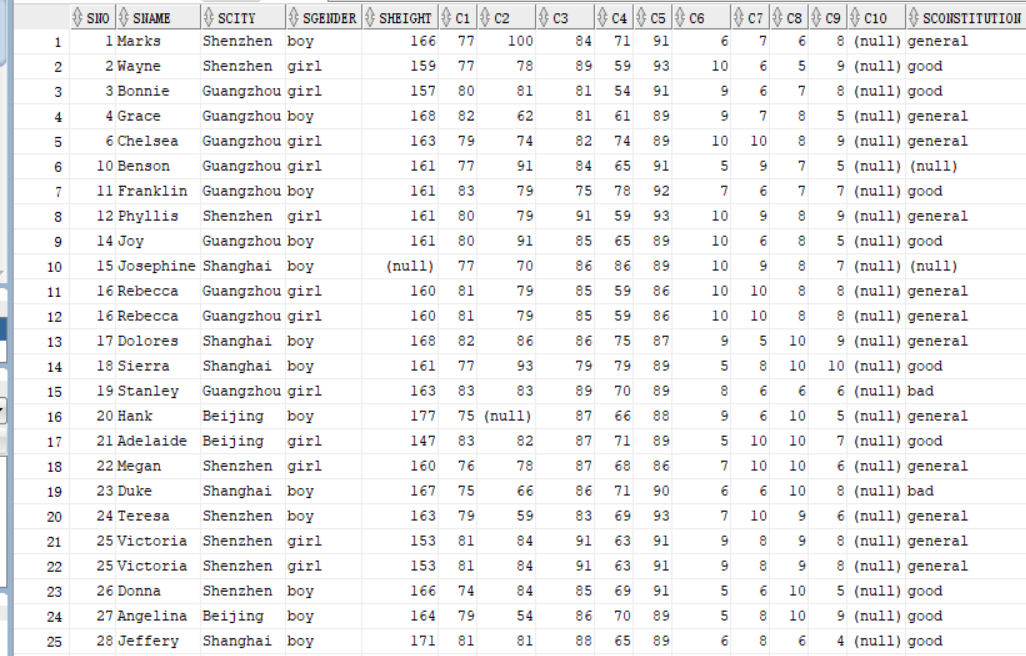
### 数据来源：

源于老师所给的两个数据源，一个为txt文档，一个为excel文档（导入数据库）

在数据库中，创建新表后导入excel文档



数据库存放结果显示：



之后在python中将两份文件调用，以下为全部实现代码

import cx\_Oracle

import pandas as pd

import numpy as np

from math import sqrt

import pickle

import random

#读取数据库数据

def readDB():

db\_conn = cx\_Oracle.connect("cc","ccpassword","localhost/OurCourse")

db\_cur = db\_conn.cursor()

result = db\_cur.execute("select \* from STUDENT")

return result,db\_cur,db\_conn

#读取txt数据

def readTxt():

txtData\_set = pd.read\_csv('data.txt',sep=',')

txtData = np.array(txtData\_set)

return txtData

# 数据单位一致（以数据库为准）

def uniformity(txtData,title):

for i in range(0,txtData.shape[1]):

for j in range(0,txtData.shape[0]):

if i == title.index('SHEIGHT'):

txtData[j][i] = txtData[j][i]\*100#txt文件里单位是米，换算为cm

if i == title.index('SNO'):

txtData[j][i] = txtData[j][i]%1000#sno只保留后三位

if i == title.index('SGENDER'):

if txtData[j][i] == 'male':

txtData[j][i] = 'boy'

if txtData[j][i] == 'female':

txtData[j][i] = 'girl'

#处理数据冗余

def redundancy(txtData\_row,title,txtData):

dataLabel = {}

i = 0

while(i<txtData\_row):

if txtData[i][title.index('SNO')] not in dataLabel:

dataLabel[txtData[i][title.index('SNO')]] = 0

dataLabel[txtData[i][title.index('SNO')]]+=1

if dataLabel[txtData[i][title.index('SNO')]]>1:

print("出现数据冗余,行数为%d"%(i))

txtData = np.delete(txtData,i,0)

i-=1

txtData\_row = txtData.shape[0]

i+=1

return txtData

#课程C10随机生成5到10的随机分数

def arr\_Column(arr,Cnum):

size=len(arr)

arr[:,Cnum]=np.random.randint(5, 10, size)

return arr

#数据合并

def complementary(db\_cur,txtData\_row,title,txtData):

i: int = 0

while(i<txtData\_row):

num = i+1

if num != txtData[i][title.index('SNO')]:

# print(num)#从数据库中，按学号提取每一行

arr = db\_cur.execute('select \* from STUDENT where SNO = %s and rownum<2'%num)

testData = []

for re in arr:

re = list(re)

testData.append(re)

testData = np.array(testData)

testData = testData.flatten()

if testData.shape == ():

# txtData = np.append(txtData,testData, axis=2)

txtData = np.row\_stack((txtData,testData))

txtData\_row = txtData.shape[0]

i+=1

# txtData[txtData[:,1].argsort()]

i = 0

dataLabel = {}

#判断是否有重复人

while(i<txtData\_row):

if txtData[i][title.index('SNO')] not in dataLabel:

dataLabel[txtData[i][title.index('SNO')]] = 0

dataLabel[txtData[i][title.index('SNO')]]+=1

#如果有重复的人就删除第二个重复的

if dataLabel[txtData[i][title.index('SNO')]]>1:

txtData = np.delete(txtData,i,0)

i-=1

txtData\_row = txtData.shape[0]#更新

i+=1

txtData = txtData[txtData[:,0].argsort()]

test = []

for i in range(txtData.shape[0]):

for j in range(txtData.shape[1]):

if pd.isna(txtData[i][j]) or txtData[i][j] == None:

id = txtData[i][title.index('SNO')]

res = db\_cur.execute("select %s from STUDENT where SNO = %s"%(title[j],id))

for re in res:

re = list(re)

if re!=None:

txtData[i][j] = re[0]

print(txtData)

np.savetxt('result1.txt',txtData,fmt='%s')#合并后文件保存为result1

return txtData

#回答问题

def Caculate(txtData,title):

#1. 学生中家乡在Beijing的所有课程的平均成绩。

#2. 学生中家乡在广州，课程1在80分以上，且课程9在9分以上的男同学的数量。

#3. 比较广州和上海两地女生的平均体能测试成绩，哪个地区的更强些？

#4. 每门学习成绩和体能测试成绩，两者的相关性是多少？

Snum=0

total=0

for i in txtData:

if i[title.index('SCITY')]=='Beijing':

Snum+=1

# if i[1]=='Maxine':

# print(1)

for j in range(title.index('C1'),title.index('C9')):

if True!=np.isnan(i[j]):

total+=i[j]

# if True==np.isnan(total):

# print(1)

ans1=total/Snum

Snum = 0

Gnum=0

for i in txtData:

if i[title.index('SCITY')] == 'Guangzhou' and i[title.index('C1')]>80 and i[title.index('C9')]>=9 and i[title.index('SGENDER')]=='BOY':

Snum+=1

ans2=Snum

G\_girl=txtData[txtData[:,title.index('SCITY')]=='Guangzhou',title.index('SCONSTITUTION')]

S\_girl=txtData[txtData[:,title.index('SCITY')]=='Shanghai',title.index('SCONSTITUTION')]

G\_score=0

S\_score=0

score={'bad':1,'general':2,'good':4,'excellent':8}

for i in G\_girl:

G\_score +=score[i]

Gnum+=1

for i in S\_girl:

S\_score +=score[i]

Snum += 1

if S\_score/Snum>G\_score/Gnum:

ans3='Shanghai'

else:

ans3='Guangzhou'

Ax={5:[],6:[],7:[],8:[],9:[],10:[],11:[],12:[],13:[]}

Ax\_sum = {5: 0, 6:0, 7:0, 8:0, 9:0, 10:0, 11:0, 12:0, 13:0}

B\_sum = 0

Bx=[]

Snum=0

for i in txtData:

for x in range(title.index('C1'),title.index('C10')):

if np.isnan(i[x]) == True:

i[x]=68

Ax\_sum[x]+=i[x]

Ax[x].append(i[x])

if type(i[title.index('SCONSTITUTION')])==str:

B\_sum+=score[i[title.index('SCONSTITUTION')]]

Bx.append(score[i[title.index('SCONSTITUTION')]])

else:

B\_sum +=0

Bx.append(3)

Snum+=1

#这里对空值的处理是，替换成不计空值成员前的平均值取整（通过excel计算）

#替换后再重新计算平均值和方差等数值

A\_mean= {5: 0, 6:0, 7:0, 8:0, 9:0, 10:0, 11:0, 12:0, 13:0}

for k in Ax\_sum:

A\_mean[k]=(Ax\_sum[k]/Snum)

B\_mean =B\_sum/Snum

A\_variance={5: 0, 6:0, 7:0, 8:0, 9:0, 10:0, 11:0, 12:0, 13:0}

B\_variance=0

for i in Ax:

for j in Ax[i]:

A\_variance[i]+=(j-A\_mean[i])\*\*2

A\_variance[i]\*=1/(Snum-1)

for i in Bx:

B\_variance+=(i-B\_mean)\*\*2

B\_variance\*=1/(Snum-1)

coefficient={'C1': 0, 'C2':0, 'C3':0, 'C4':0, 'C5':0, 'C6':0, 'C7':0, 'C8':0, 'C9':0}

for i,j in zip(Ax,Bx):

for z in Ax[i]:

coefficient[title[i]]+=((z-A\_mean[i])/sqrt(A\_variance[i]))\*((j-B\_mean)/sqrt(B\_variance))

ans4=coefficient

print('问题1：',ans1,'\n'

'问题2: ',ans2,'\n'

'问题3：',ans3,'\n'

'问题4：',ans4,'\n')

def main():

result,db\_cur,db\_conn=readDB()#调用数据库并保存

title = [i[0] for i in db\_cur.description]#以数据库为准保存表头

print(title)

txtData=readTxt()#调用txt并保存

uniformity(txtData,title)#维持txt的数据一致性

txtData\_row = txtData.shape[0]

txtData=arr\_Column(txtData,title.index('C10'))

txtData=redundancy(txtData\_row,title,txtData)

txtData=complementary(db\_cur, txtData\_row, title,txtData)

Caculate(txtData,title)

db\_cur.close()

db\_conn.close()

if \_\_name\_\_ == '\_\_main\_\_':

main()

#### 实验运行结果截图