#导包

import pandas as pd

import numpy as np

import linecache

#利用pandas直接获取数据

data = pd.read\_excel("./实验数据/Student.xlsx")

#删去C10列

data = data.drop(['C10'], axis = 1)

#用前一个数据进行填充NAN

data = data.fillna(method = 'pad')

Data

import pymysql

#连接数据库

conn = pymysql.connect(host = 'localhost', user = 'root', password = '09083645aA', database = 'test', charset = "utf8")

cursor = conn.cursor()

#创建student表

sql = "CREATE TABLE Student(ID int,Name varchar(20) ,City varchar(10),Gender varchar(10), Height float,C1 float,C2 float,C3 float,C4 float,C5 float,C6 float,C7 float,C8 float,C9 float,Constitution varchar(10));"

cursor.execute(sql)

conn.commit()

#插入数据

n = data['ID'].count()

for i in range(0, n) :

sql = "insert into student values(%d,'%s','%s','%s',%f,%f,%f,%f,%f,%f,%f,%f,%f,%f,'%s');" % (data.iloc[i][0], data.iloc[i][1], data.iloc[i][2], data.iloc[i][3], data.iloc[i][4], data.iloc[i][5], data.iloc[i][6], data.iloc[i][7], data.iloc[i][8], data.iloc[i][9], data.iloc[i][10], data.iloc[i][11], data.iloc[i][12], data.iloc[i][13], data.iloc[i][14])

cursor.execute(sql)

conn.commit()

#读取数据库数据

sql\_1 = "select \* from student"

data\_db = pd.read\_sql(sql\_1, conn)

#关闭连接

conn.close()

#查看数据库数据

data\_db

#data1读取文件第一行，用作字段名

data1 = linecache.getline(r'.\实验数据\Student.txt', 1).strip('\n').split(',')

#data2存储字段值

data2 = np.loadtxt(r'.\实验数据\Student.txt', dtype = str, delimiter = ',', skiprows = 1, encoding = 'UTF-8')

#用numpy读取txt文件并将C10列删掉

data\_txt = pd.DataFrame(data2, columns = data1).drop(['C10'], axis = 1)

#将空字符串类型转成NAN

data\_txt = data\_txt.replace(to\_replace = "", value = np.NaN)

#用前一个数据进行填充NAN

data\_txt = data\_txt.fillna(method = 'pad')

data\_txt

#数据处理, 消除不一致

#单位转换一致

data\_db.loc[:, 'ID'] = data\_db['ID'].astype(int) + 202000

data\_txt.loc[data\_txt.Gender == 'male', 'Gender'] = 'boy'

data\_txt.loc[data\_txt.Gender == 'female', 'Gender'] = 'girl'

data\_txt.loc[:, 'Height'] = data\_txt['Height'].astype(float) \* 100

#类型转换一致

data\_txt.loc[:, 'ID'] = data\_txt['ID'].astype(int)

data\_txt.loc[:, 'C1'] = data\_txt['C1'].astype(int)

data\_txt.loc[:, 'C2'] = data\_txt['C2'].astype(int)

data\_txt.loc[:, 'C3'] = data\_txt['C3'].astype(int)

data\_txt.loc[:, 'C4'] = data\_txt['C4'].astype(int)

data\_txt.loc[:, 'C5'] = data\_txt['C5'].astype(int)

data\_txt.loc[:, 'C6'] = data\_txt['C6'].astype(int)

data\_txt.loc[:, 'C7'] = data\_txt['C7'].astype(int)

data\_txt.loc[:, 'C8'] = data\_txt['C8'].astype(int)

data\_txt.loc[:, 'C9'] = data\_txt['C9'].astype(int)

data\_db.loc[:, 'C1'] = data\_db['C1'].astype(int)

data\_db.loc[:, 'C2'] = data\_db['C2'].astype(int)

data\_db.loc[:, 'C3'] = data\_db['C3'].astype(int)

data\_db.loc[:, 'C4'] = data\_db['C4'].astype(int)

data\_db.loc[:, 'C5'] = data\_db['C5'].astype(int)

data\_db.loc[:, 'C6'] = data\_db['C6'].astype(int)

data\_db.loc[:, 'C7'] = data\_db['C7'].astype(int)

data\_db.loc[:, 'C8'] = data\_db['C8'].astype(int)

data\_db.loc[:, 'C9'] = data\_db['C9'].astype(int)

data\_db.loc[:, 'Height'] = data\_db['Height'].astype(float)

#data1删去C10这个元素

data1.remove('C10')

#纵向合并数据

data = pd.DataFrame(np.vstack((data\_db, data\_txt)), columns = data1)

Data

#数据处理，去除冗余

data = data.drop\_duplicates(subset = ['ID'])

Data

#算C1平均值代码

#ts = 0

#for i in data['C1']:

# ts = ts + i

#t = ts / data['C1'].count()

#A1存储学生中家乡在Beijing的所有课程的平均成绩。

A1 = []

A1 = data[data['City'] == 'Beijing'].mean()

A1[1:12]

#学生中家乡在Guangzhou，课程1在80分以上，且课程9在9分以上的男同学

data[data['City'] == 'Guangzhou'][data['C1'] >= 80][data['C9'] >= 9][data['Gender'] == 'boy']

#A2存储学生中家乡在广州，课程1在80分以上，且课程9在9分以上的男同学的数量

A2 = data[data['City'] == 'Shenzhen'][data['C1'] >= 80][data['C9'] >= 9][data['Gender'] == 'boy']['ID'].count()

A2

#给体能增加一个数值化

data.loc[data.Constitution == 'excellent', 'Constitution\_int'] = 4

data.loc[data.Constitution == 'good', 'Constitution\_int'] = 3

data.loc[data.Constitution == 'general', 'Constitution\_int'] = 2

data.loc[data.Constitution == 'bad', 'Constitution\_int'] = 1

#由A3\_Guangzhou大于A3\_Shanghai可知广州女生的平均体能测试成绩更强

A3\_Guangzhou = data[data['City'] == 'Guangzhou']['Constitution\_int'].mean()

A3\_Guangzhou

A3\_Shanghai = data[data['City'] == 'Shanghai']['Constitution\_int'].mean()

A3\_Shanghai

#计算C1方差代码

#vs = 0

#for i in data['C1']:

# vs = vs + (i - data['C1'].mean())\*(i - data['C1'].mean())

#var = vs / (data['C1'].count() - 1)

#A4记录C1与体能测试成绩相关性

C1\_x = data['C1'].mean()#C1平均值

C1\_var = data['C1'].var()#C1方差

C1\_std = np.sqrt(C1\_var)#C1标准差

Con\_x = data['Constitution\_int'].mean()

Con\_var = data['Constitution\_int'].var()

Con\_std = np.sqrt(Con\_var)

#t1存储C1列，t2存储Constitution列

t1 = []

for i in data['C1']:

t1.append(i)

t2 = []

for i in data['Constitution\_int'] :

t2.append(i)

#t3存储t1与t2数乘列

t3 = np.multiply(t1, t2)

#转换成dataframe格式，方便计算

t = pd.DataFrame(t3, columns = ['C1\_Constitution'])

#计算协方差

cov = t['C1\_Constitution'].mean() - C1\_x \* Con\_x

#计算相关系数, | cor | ~[0, 1], 越接近1相关性越大，越接近0相关性越小

cor = cov / (C1\_std\*Con\_std)

np.abs(cor)

#将数据写进Student2

data.to\_csv('./实验数据/Student2.csv', index = False)