#### Experiment Report

#### **第一题**

计算各门学科不及格人数、平均分、及成绩分散程度（标准差）。并将这些值作为新行添加到数据表中。

* 没有插入一行的方法，所以要先构造一行，再转换成df使用concat进行连接
* import pandas  
  from pandas import read\_csv  
  from pandas import DataFrame  
  from pandas import Series  
  missing\_values = [' ','\t',' ','作弊','缺考']  
  df = read\_csv("D:/programming/condaProgram/testScore.txt",sep = '\t',na\_values =missing\_values)  
  df.fillna(0,inplace = True) #  
  print(df)  
    
    
  #不及格人数汇总  
  p=[]  
  for item in ['英语', '体育','军训', '数分', '高代','解几']:  
   pa=pandas.cut(df[item],[-1,59.9],labels = ['不及格人数']) #左开右闭  
   p.append(pa.value\_counts())  
  print(p)  
    
  line=pandas.DataFrame({df.columns[4]:p[0],df.columns[5]:p[1],df.columns[6]:p[2],df.columns[7]:p[3]  
   ,df.columns[8]:p[4],df.columns[9]:p[5]})  
  print(line)  
  print(pandas.concat([df[0:],line])) #插入一行
* 依据以上思路即可得到后两行
* avg=[]  
  #插入平均分  
  for i in range(4,10):  
   avg.append(df.iloc[:,i].mean())  
  line=pandas.DataFrame({df.columns[4]:avg[0],df.columns[5]:avg[1],df.columns[6]:avg[2],df.columns[7]:avg[3]  
   ,df.columns[8]:avg[4],df.columns[9]:avg[5]},index=["平均分"])  
  print(pandas.concat([df[0:],line]))  
  '''  
  #标准差  
  std=[]  
  for i in range(4,10):  
   std.append(df.iloc[:,i].std())  
  line=pandas.DataFrame({df.columns[4]:std[0],df.columns[5]:std[1],df.columns[6]:std[2],df.columns[7]:std[3]  
   ,df.columns[8]:std[4],df.columns[9]:std[5]},index=["标准差"])  
  print(pandas.concat([df[0:],line]))
* 最后合并
* print(pandas.concat([df[0:],line,line1,line2]))
* **至此结束**

#### **第二题**

将数据表添加两列：每位同学的各科成绩总分（ score ）和每位同学的整体情况（类别 ）。 类别按照 学生总成绩分为“一般”、“较好”、“优秀”三种情况。计算输出各类别学生比例。

* 添加列：
  + insert方法df.insert(pos,'key',value)
  + reindex方法df.reindex(columns=[],fill\_value=)
  + concat方法pandas.concat([df,DataFrame()])
  + 用loc直接添加df.loc[:,'d']需注意loc用列名访问

import pandas  
from pandas import read\_csv  
from pandas import DataFrame  
from pandas import Series  
pandas.set\_option('display.max\_rows',None) #出现省略号可以更改不限行和列  
pandas.set\_option('display.max\_columns',None)  
missing\_values = [' ','\t',' ','作弊','缺考']  
df = read\_csv("D:/programming/condaProgram/testScore.txt",sep = '\t',na\_values =missing\_values)  
df.fillna(0,inplace = True) #  
  
#总成绩和等级比例  
s=[]  
for i in range(21):  
 s.append(df.iloc[i,4:10].sum())  
print(s)  
bins=[0,360,480,600]  
#360分以下为一般 360-480为较好，480以上为优秀  
pa = pandas.cut(s,bins,labels=['一般','较好','优秀'])  
print(pa)  
  
df.loc[:,'总分']=s  
df.loc[:,'评价']= pa  
print(df)  
  
print("比例：\n",df['评价'].value\_counts()/21)#value\_counts()只能Series用

* **第二题结束**

#### **第三题**

由于“军训”这门课程的成绩与其他科目成绩差异较大，并且给分较为随意，为了避免给同学评定奖学金带来不公平，请将每位同学的各科成绩标准化，再汇总，并标出“一般”“较好”“优秀”三种类别。计算输出各类别学生比例，并将其与（2）中得到的情况进行对比。

* 标准化
* #数据标准化1 新数据=(原数据-min)/(max-min) 0-1标准化  
  print((df1.a-df1.a.min())/(df1.a.max()-df1.a.min()))  
  #也可用lambda函数  
    
  F=lambda df1 : (df1.a-df1.a.min())/(df1.a.max()-df1.a.min())  
  print(F(df1))  
  #数据标准化2 新数据=（原数据-均值）/标准差 z-score 标准化 中心化  
  print((df1['a']-df1['a'].mean())/df1.a.std()) #df.var()方差 df.std()标准差  
  # 数据标准化3 用于稀疏数据 新数据=x/|max|  
  print(df1['a']/abs(df1['a'].max()))  
    
  #数据标准化4 针对离群点的RobustScaler 新数据=（原数据-median）/IQR （IQR=Q3-Q1）  
  #df1['a'].quantile([0.25,0.5,0.75])  
  print(df1['a'].quantile([0.25,0.5,0.75]))  
  res1=(df1['a']-df1['a'].median())  
  res2=df1['a'].quantile(0.75)-df1['a'].quantile(0.25)  
  print(res1/res2 )
* 各科先标准化再汇总是先对列进行标准化，在对行进行汇总,最后进行标准划分和输出
* import pandas  
  from pandas import read\_csv  
  from pandas import DataFrame  
  from pandas import Series  
  pandas.set\_option('display.max\_rows',None) #出现省略号可以更改不限行和列  
  pandas.set\_option('display.max\_columns',None)  
  missing\_values = [' ','\t',' ','作弊','缺考']  
  df = read\_csv("D:/programming/condaProgram/testScore.txt",sep = '\t',na\_values =missing\_values)  
  df.fillna(0,inplace = True)   
    
  #数据标准化1 新数据=(原数据-min)/(max-min) 0-1标准化  
  print(df)  
  m=[]  
  #标准化  
  for i in range(4,10):  
   n=df.iloc[:,i]  
   m.append((n-n.min())/(n.max()-n.min()))  
  #汇总   
  sum\_std=[]  
  df1=DataFrame(m)  
  for i in range(21):  
   sum\_std.append(sum(df1[i]))  
    
    
  bins=[-1,3.6,4.8,6]  
  labels=['一般','较好','优秀']  
  pa=pandas.cut(sum\_std,bins,labels=labels)  
  df['label']=pa  
  print(df)  
  print(df['label'].value\_counts()/21)
* **至此第三题结束**

#### **第四题**

按学号顺序输出各学生总分及排名

* 排序
  + Series和DataFrame可以用sort\_values()
  + 同时可以用rank来进行排名后者可指定调用的是行还是列
* # -\*- coding: utf-8 -\*-  
  """  
  Created on Tue Mar 29 22:08:37 2022  
    
  @author: lenovo  
  """  
    
  import pandas  
  from pandas import read\_csv  
  from pandas import DataFrame  
  from pandas import Series  
  pandas.set\_option('display.max\_rows',None) #出现省略号可以更改不限行和列  
  pandas.set\_option('display.max\_columns',None)  
  missing\_values = [' ','\t',' ','作弊','缺考']  
  df = read\_csv("D:/programming/condaProgram/testScore.txt",sep = '\t',na\_values =missing\_values)  
  df.fillna(0,inplace = True)  
    
  # 去重  
  df.drop\_duplicates(subset=['学号'],keep='first',inplace =True)  
    
    
  #用sort\_values()  
  print(df.sort\_values('学号',inplace=True)) #默认(axis=0)也即取列的值  
  print(df)  
  sum1=[]  
  for i in range(20):  
   sum1.append(df.iloc[i,4:10].sum())  
  print(sum1)  
  df['总分']=sum1  
  df['排名']=df['总分'].rank(method='first',ascending=False)  
  print(df)  
  df['排名']=[i for i in range(1,21)]  
  print(df)
* **至此第四题结束**

#### 第五题

使用画图工具（如excel等）将（1）~（3）的结果进行图形化对比分析展示：对比分析各科不及格人数、平均分及分散程度，对比分析各类别学生的比例。