第九章 课程设计指导

9.1 题目要求：学生成绩数据分析

已知某年级学生期末考试成绩“studentScore.csv”，包含 “考号”、“姓名”、“班级”、“语文”、“数学”、“英语”、“总分”、“班名次”、“升降幅度”9个字段。要求根据所学知识完成以下任务。

表9-1 某年级学生期末考试成绩统计表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 考号 | 姓名 | 班级 | 语文 | 数学 | 英语 | 总分 | 班名次 | 升降幅度 |
| 70605 | 张璐 | 一班 | 131 | 143 | 144 | 418 | 1 | ↑4 |
| 70603 | 王雪 | 二班 | 131 | 135 | 144 | 410 | 2 | ↑1 |
| 70609 | 李林霖 | 一班 | 127 | 139 | 142 | 408 | 3 | ↑6 |
| 70601 | 赵龙逸 | 二班 | 123 | 148 | 136 | 407 | 4 | ↓3 |
| 70606 | 周鉴学 | 一班 | 126 | 135 | 140 | 401 | 5 | ↑1 |
| 70604 | 武雨萌 | 二班 | 129 | 133 | 138 | 400 | 6 | ↓2 |
| 70602 | 郑帅 | 一班 | 116 | 143 | 140 | 399 | 7 | ↓5 |
| 70616 | 王惠雯 | 二班 | 114 | 142 | 139 | 395 | 8 | ↑8 |
| 70607 | 张钰婷 | 一班 | 115 | 139 | 135 | 389 | 9 | ↓2 |
| 70611 | 王世博 | 二班 | 116 | 142 | 129 | 387 | 10 | ↑1 |
| 70621 | 李希 | 一班 | 123 | 130 | 134 | 387 | 11 | ↑10 |
| 70608 | 张冲 | 二班 | 122 | 124 | 139 | 385 | 12 | ↓4 |
| 70612 | 许宇飞 | 一班 | 118 | 136 | 131 | 385 | 13 | ↓1 |
| 70623 | 樊一凡 | 二班 | 121 | 123 | 139 | 383 | 14 | ↑9 |
| 70610 | 李瑞鑫 | 一班 | 126 | 115 | 139 | 380 | 15 | ↓5 |
| 70633 | 武作鑫 | 二班 | 121 | 127 | 131 | 379 | 16 | ↑16 |
| 70620 | 张子翔 | 一班 | 111 | 139 | 128 | 378 | 17 | ↑3 |
| 70625 | 郑传禹 | 二班 | 119 | 129 | 130 | 378 | 18 | ↑7 |
| 70619 | 武雪桐 | 一班 | 124 | 108 | 144 | 376 | 19 | → |
| 70614 | 王姗 | 二班 | 124 | 128 | 122 | 374 | 20 | ↓6 |
| 70613 | 张柏坤 | 一班 | 121 | 123 | 128 | 372 | 21 | ↓8 |
| 70668 | 李永刚 | 一班 | 116 | 131 | 122 | 369 | 22 | → |
| 70636 | 王馨月 | 二班 | 114 | 124 | 122 | 360 | 23 | ↑13 |
| 70667 | 郑曦月 | 一班 | 116 | 123 | 119 | 358 | 24 | → |
| 70624 | 赵丁文 | 二班 | 116 | 122 | 118 | 356 | 25 | ↓1 |
| 70626 | 郑美欣 | 一班 | 118 | 126 | 111 | 355 | 26 | → |
| 70629 | 徐殿宇 | 二班 | 112 | 109 | 130 | 351 | 27 | ↑2 |
| 70646 | 李智文 | 一班 | 109 | 116 | 125 | 350 | 28 | ↑18 |
| 70649 | 张季 | 二班 | 114 | 117 | 118 | 349 | 29 | ↑20 |
| 70645 | 马加晖 | 一班 | 110 | 102 | 136 | 348 | 30 | ↑15 |
| 70635 | 徐旭 | 二班 | 114 | 113 | 120 | 347 | 31 | ↑4 |
| 70618 | 周智凯 | 一班 | 110 | 117 | 119 | 346 | 32 | ↓14 |
| 70643 | 李晓爽 | 二班 | 113 | 113 | 119 | 345 | 33 | ↑10 |
| 70637 | 张美慧 | 一班 | 117 | 121 | 106 | 344 | 34 | ↑3 |
| 70617 | 蒋埕镒 | 二班 | 112 | 105 | 126 | 343 | 35 | ↓18 |
| 70615 | 周俊文 | 一班 | 103 | 127 | 110 | 340 | 36 | ↓21 |
| 70634 | 何馨月 | 二班 | 108 | 110 | 119 | 337 | 1 | ↑3 |
| 70638 | 郑诗涵 | 一班 | 105 | 105 | 126 | 336 | 2 | → |
| 70622 | 黄宇 | 二班 | 115 | 111 | 106 | 332 | 3 | ↑17 |
| 70631 | 张俊焯 | 一班 | 101 | 112 | 106 | 319 | 4 | ↑9 |
| 70642 | 郑建志 | 二班 | 98 | 104 | 116 | 318 | 5 | ↑1 |
| 70627 | 王俊奎 | 一班 | 103 | 103 | 111 | 317 | 6 | ↑15 |
| 70641 | 何波 | 二班 | 96 | 130 | 89 | 315 | 7 | ↓2 |
| 70648 | 孙良 | 一班 | 105 | 102 | 105 | 312 | 8 | ↑4 |
| 70650 | 李春月 | 二班 | 118 | 82 | 112 | 312 | 9 | ↑5 |
| 70632 | 许明哲 | 一班 | 101 | 91 | 115 | 307 | 10 | ↑14 |
| 70630 | 王开羽 | 二班 | 112 | 90 | 104 | 306 | 11 | ↓7 |
| 70639 | 李建译 | 一班 | 109 | 68 | 126 | 303 | 12 | ↓9 |
| 70647 | 狄家硕 | 二班 | 107 | 67 | 129 | 303 | 13 | ↓2 |
| 70655 | 周娜 | 一班 | 94 | 98 | 104 | 296 | 14 | ↑5 |
| 70644 | 郑坤 | 二班 | 107 | 76 | 104 | 287 | 15 | ↓7 |
| 70628 | 张禹 | 一班 | 106 | 100 | 80 | 286 | 16 | ↓4 |
| 70651 | 袁治超 | 二班 | 105 | 95 | 82 | 282 | 17 | ↓2 |
| 70640 | 张云飞 | 一班 | 101 | 59 | 108 | 268 | 18 | ↓6 |
| 70656 | 何茂哲 | 二班 | 85 | 95 | 85 | 265 | 19 | ↑1 |
| 70669 | 袁婉艺 | 一班 | 95 | 71 | 85 | 251 | 20 | → |
| 70654 | 丁娇莹 | 二班 | 97 | 76 | 71 | 244 | 21 | ↓3 |
| 70661 | 李旭辉 | 一班 | 99 | 87 | 44 | 230 | 22 | ↑3 |
| 70659 | 王宇婷 | 二班 | 83 | 61 | 71 | 215 | 23 | → |
| 70652 | 吕明涛 | 一班 | 87 | 77 | 43 | 207 | 24 | ↓8 |
| 70657 | 赵文卓 | 二班 | 82 | 53 | 62 | 197 | 25 | ↓4 |
| 70653 | 李晟煜 | 一班 | 79 | 49 | 64 | 192 | 26 | ↓9 |
| 70660 | 李龙基 | 二班 | 99 | 21 | 67 | 187 | 27 | ↓3 |
| 70662 | 张森 | 一班 | 90 | 29 | 64 | 183 | 28 | ↓2 |
| 70663 | 王朝升 | 二班 | 78 | 45 | 47 | 170 | 29 | ↓2 |
| 70658 | 张忠浩 | 一班 | 86 | 32 | 46 | 164 | 30 | ↓8 |
| 70664 | 周禹志 | 二班 | 75 | 23 | 34 | 132 | 31 | ↓3 |
| 70665 | 李鸿涛 | 二班 | 66 | 23 | 34 | 123 | 32 | ↓3 |

【任务】

1. 用pandas读取“studentScore.csv”，将缺失值丢弃处理导出为新文件“studentScoreP.csv”，并查看前三行、后两行。
2. 选择预处理后的“studentScoreP.csv”文件中的列“姓名”、“考号”、“班级”、“语文”、“数学”、“英语”，导出为新的csv文件“studentScoreP\_new.csv”；然后，重新读取新的数据集“studentScoreP\_new.csv”，并选择字段“150”>=语文>=“100”、“150”>=数学>=“100”、“150”>=英语>=“100”的所有数据集，导出为新的文本文件“studentScoreP\_newGood.txt”，要求数据之间用逗号分隔，每行末尾包含换行符。
3. 重新读取文件“studentScoreP\_new.csv”，按照列“班级”分类汇总各班“语文”、“数学”、“英语”的平均成绩，并将分组计算结果导出到文本文件studentScoreP\_MeanGroup.txt中，要求分组名不作为列名。
4. 重新读取CSV文件“studentScoreP\_new.csv”，计算每位同学语文、数学、英语三科的平均成绩，并将平均成绩作为一个新的列“均值”添加到原始文件，并按照列“均值”降序排序，并将排序后结果转存到Excel文件studentScoreP\_Mean.xlsx中。
5. 读取新的数据集“studentScoreP\_Mean.xlsx”，统计字段“均值”的最大值maxValue、四分之三位数threeQuartersValue、中位数medianValue、四分之一位数quarterValue、最小值minValue，category = [minValue, quarterValue, medianValue, threeQuartersValue，maxValue]和labels = ['Poor', 'Moderate', 'Good', ‘Excellent’]将均值进行离散化，并根据离散化结果进行直方图统计，分别画出统计结果的柱状图和饼状图，并分别将柱状图和饼状图保存为“studentScoreP\_Mean\_bar.png”、“ studentScoreP\_Mean\_pie.png”，要求柱状图的x轴刻度以及饼状图均显示labels，图像分辨率不低于300dpi。
6. 重新读取文件“studentScoreP\_Mean.xlsx”，分别可视化显示每位学生的平均成绩，要求包括图例、图标题，x、y轴均显示刻度值且x轴刻度值以姓名显示，曲线颜色为红色，以png图片保存，分辨率为400dpi，png图片命名分别为“studentScoreP\_Mean.png”。此外，为显示美观，x轴姓名显示请选择倾斜显示或者间隔5名同学显示一次学生姓名（即将姓名每隔5个抽样显示）。

【要求】

1. 根据以上数据处理任务，设计并编程实现“数据分析与可视化系统”，要求① 各个任务选择用菜单实现（菜单可用字符串输出模拟，或者Tkinter形式实现）。

② 各个任务名称自己定义，须由独立的函数实现，且每个任务执行成功与否须给出必要的文字提示。

③ 数据输入和结果输出的文件名须由人工输入，且输出结果都要以文件形式保存。

④ 为保持程序的健壮性，各个任务执行过程中需要进行必要的判断（如文件是否存在、输入是否合法等）、程序异常控制等。

1. 根据以上统计结果，书写不少于300字的结果分析。

9.2 需求分析

根据题目要求，任务主要涉及到数据读取、数据查询、数据分类汇总、数据计算及排序、数据可视化、数据导出等常规的数据分析操作步骤，可以调用Pandas的文件读写、数据分析等功能模块实现；各任务要求用函数形式实现，则需要设计各函数之间用参数传递实现各操作步骤之间的松耦合，进行模块化程序设计；各操作步骤需要用菜单实现功能选择，以及提供必要的输入输出等人机交互操作；此外，在程序中应提供必要的异常控制代码，保证程序的健壮性。

9.3 概要设计

根据需求分析，可以将该系统设计为“数据读取及预处理”、“数据选择及导出”、“数据分类汇总”、“数据计算及排序”、“数据统计”、“数据可视化”六大功能模块，以及“功能选择”主菜单辅助模块。

图9-1 学生成绩数据分析系统概要设计

* 1. 详细设计

9.4.1 主函数详细设计

【分析】

在Python函数式编程中，主函数一般比较简洁，只提供函数调用。在本例中，主函数仅包含任务调用函数。

【关键代码】

##########################################################

#主函数

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

task()#调用功能选择函数

##########################################################

9.4.2.功能选择模块详细设计

【分析】

功能选择模块设计应比较简洁，只提供输入、功能处理和输出部分的函数调用。其中，各功能模块采用菜单式选择。另外，由于该实例中各任务之间存在先后顺序关系，后续任务的输入可能来源于前序任务的输出，因此在各步骤执行前还需要判断需要的数据源是否已经准备好，即前序任务是否已经执行完毕。

具体流程如下图所示：

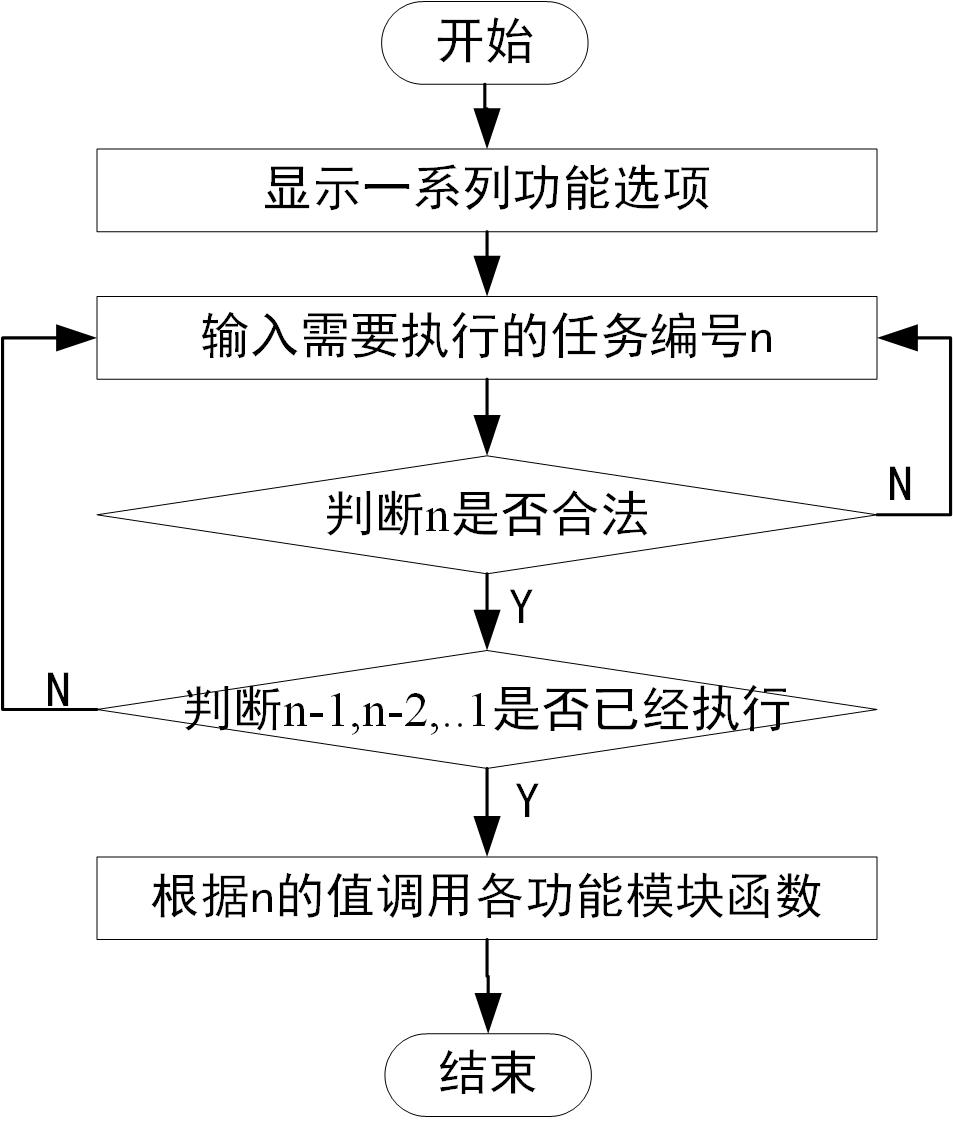


图9-2 功能选择模块程序流程图

【关键代码】

##########################################################

#系统主界面

def menu():

print('【任务选择】\n'

'＋－－－－学生成绩数据分析及可视化系统 －－－－－－－－－－＋\n'

'｜0、退出。 ｜\n'

'｜1、数据读取及预处理。 ｜\n'

'｜2、数据选择及导出。 ｜\n'

'｜3、数据分类汇总。 ｜\n'

'｜4、数据计算及排序。 ｜\n'

'｜5、数据统计。 ｜\n'

'｜6、数据可视化。 ｜\n'

'＋－－－－－－－－－－－－－－－－－－－－－－－－－-－－－＋')

###############################################################

#功能选择模块

def task():

while True:

menu()#打印系统主界面

num = input("请输入任务选项：")

if num == '1':

dataPreprocessing()

elif num == '2':

if os.path.exists('studentScoreP.csv'):

dataSelection()

else:

print('未能执行当前选项，请先执行前面的选项！')

elif num == '3':

if os.path.exists('studentScoreP\_new.csv'):

dataGroup()

else:

print('未能执行当前选项，请先执行前面的选项！')

elif num == '4':

if os.path.exists('studentScoreP\_new.csv'):

dataCalculate()

else:

print('未能执行当前选项，请先执行前面的选项！')

elif num == '5':

if os.path.exists('studentScoreP\_Mean.xlsx'):

dataDescribeVisualization()

else:

print('未能执行当前选项，请先执行前面的选项！')

elif num == '6':

if os.path.exists('studentScoreP\_Mean.xlsx'):

dataVisualization()

else:

print('未能执行当前选项，请先执行前面的选项！')

elif num == '0':

print('程序结束！')

break

else:

print('输入选项有误')

input("回车显示菜单")

##########################################################

9.4.3 数据读取及预处理模块详细设计

【分析】

（1）数据读取除了文件操作open函数外，更为便捷的是Pandas提供的read\_csv和read\_excel两个功能，因此本示例选择read\_csv来进行数据读取。

read\_csv和read\_excel用法及参数解释如下所示：

|  |
| --- |
| Pandas读取CSV文件功能：read\_csv |
| read\_csv(filepath\_or\_buffer, sep=',', delimiter=None, header='infer', names=None, index\_col=None, usecols=None, squeeze=False, prefix=None, mangle\_dupe\_cols=True, dtype=None, engine=None, converters=None, true\_values=None, false\_values=None, skipinitialspace=False, skiprows=None, nrows=None, na\_values=None, keep\_default\_na=True, na\_filter=True, verbose=False, skip\_blank\_lines=True, parse\_dates=False, infer\_datetime\_format=False, keep\_date\_col=False, date\_parser=None, dayfirst=False, iterator=False, chunksize=None, compression='infer', thousands=None, decimal=b'.', lineterminator=None, quotechar='"', quoting=0, escapechar=None, comment=None, encoding=None, dialect=None, tupleize\_cols=False, error\_bad\_lines=True, warn\_bad\_lines=True, skipfooter=0, skip\_footer=0, doublequote=True, delim\_whitespace=False, as\_recarray=False, compact\_ints=False, use\_unsigned=False, low\_memory=True, buffer\_lines=None, memory\_map=False, float\_precision=None)  参数解释   * filepath\_or\_buffer:（唯一一个必有参数）文件所在处的路径 * sep：指定分隔符，默认为逗号',' * delimiter : 定界符，备选分隔符（如果指定该参数，则sep参数失效） * header：指定哪一行作为表头。默认设置为0（即第一行作为表头），如果没有表头的话，要修改参数，设置header=None。 * names：指定列的名称，用列表表示。一般我们没有表头，即header=None时，这个用来添加列名就很有用啦！ * index\_col:指定哪一列数据作为行索引，可以是一列，也可以多列。多列的话，会看到一个分层索引 * prefix：给列名添加前缀。如prefix="x",会出来"x1"、"x2"、"x3"酱纸 * nrows : 需要读取的行数（从文件头开始算起） * encoding：乱码的时候用这个就是了，官网文档看看用哪个： * skiprows : 需要忽略的行数（从文件开始处算起），或需要跳过的行号列表（从0开始）。 |

|  |
| --- |
| Pandas读取Excel文件功能：read\_excel |
| read\_excel(io,sheetname=0,header=0,skiprows=None,index\_col=None,names=None, arse\_cols=None,date\_parser=None,na\_values=None,thousands=None, convert\_float=True,has\_index\_names=None,converters=None,dtype=None, true\_values=None,false\_values=None,engine=None,squeeze=False,\*\*kwds)  参数解释   * io ：excel 路径； * sheetname：默认是sheetname为0，返回多表使用sheetname=[0,1]，若sheetname=None是返回全表 。注意：int/string返回的是dataframe，而none和list返回的是dict of dataframe。 * header ：指定作为列名的行，默认0，即取第一行，数据为列名行以下的数据；若数据不含列名，则设定 header = None； * skiprows：省略指定行数的数据 * skip\_footer：省略从尾部数的行数据 * index\_col ：指定列为索引列，也可以使用 u’string’ * names：指定列的名字，传入一个list数据 |

（2）数据预处理主要包含缺失值丢弃处理、缺失值补充、重复值丢弃、异常值丢弃或规约等处理方式。本示例要求将缺失值丢弃处理，则直接利用Pandas中的dropna()功能即可。

【关键代码】

###########################################################

#读取数据

df = pd.read\_csv('studentScore.csv', encoding='cp936')

#丢弃缺失值

df = df.dropna()

df.to\_csv('studentScoreP.csv', encoding='cp936',index=False)

##查看前三行

print(df.head(3))

##查看后两行

print(df.tail(2))

##########################################################

9.4.4 数据选择及导出模块详细设计

【分析】

1. Pandas提供了直接索引、基于标签索引、基于位置索引等数据选择功能。

###############################################################################

>>>import pandas as pd

>>>import numpy as np

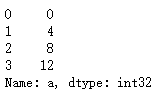
>>>df=pd.DataFrame(np.arange(16).reshape((4,4)),index=list(range(4)),columns=['a','b','c','d'])

>>>df

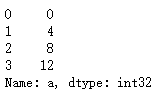


#直接索引

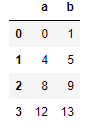
>>>df[‘a’]



>>>df.a

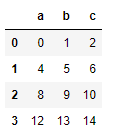


>>>df[['a', 'b']]

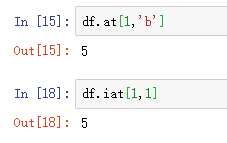


#基于标签索引

>>>df.loc[0:3,‘a’:‘c’] #选择一个区间



>>>df.at[1,‘b’] #选择单个值

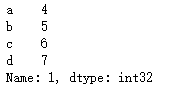


#基于位置索引

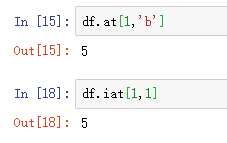
>>>df.iloc[0:3,0:4]



>>>df.iloc[1]#如果iloc方括号中直接给定一个数字或者一个slice的话，默认索引的是行。其中数字的情况会返回一个Series



>>>df.iat[1,1] #选择单个值



###############################################################################

1. 数据导出除了文件操作的write(s)和writelines(s)之外，更为便捷的是Pandas提供的to\_csv和to\_excel两个功能。

to\_csv和to\_excel用法及参数解释如下所示：

|  |
| --- |
| Pandas导出CSV文件功能：to\_csv |
| to\_csv(path\_or\_buf=None, sep=', ', na\_rep='', float\_format=None, columns=None, header=True, index=True, index\_label=None, mode='w', encoding=None, compression=None, quoting=None, quotechar='"', line\_terminator='\n', chunksize=None, tupleize\_cols=None, date\_format=None, doublequote=True, escapechar=None, decimal='.')  参数解释   * path\_or\_buf=None： 字符串或文件句柄，默认无文件路径或对象，如果没有提供，结果将返回为字符串。 * sep : 默认字符 ‘ ，’。输出文件的字段分隔符。 * na\_rep : 字符串，默认为 ‘’。浮点数格式字符串 * float\_format :字符串，默认为 None。浮点数格式字符串 * columns :顺序，可选列写入 * header : 字符串或布尔列表，默认为true。写出列名。如果给定字符串列表，则假定为列名的别名。 * index : 布尔值，默认为Ture。写入行名称（索引） * index\_label : 字符串或序列，或False,默认为None。如果需要，可以使用索引列的列标签。如果没有给出，且标题和索引为True，则使用索引名称。如果数据文件使用多索引，则应该使用这个序列。如果值为False，不打印索引字段。在R中使用index\_label=False 更容易导入索引. * mode : 模式：值为‘str’，字符串。Python写模式，默认“w” * encoding :编码：字符串，可选。表示在输出文件中使用的编码的字符串，Python 3上默认为“UTF-8”。 * compression : 字符串，可选项。表示在输出文件中使用的压缩的字符串，允许值为“gzip”、“bz2”、“xz”，仅在第一个参数是文件名时使用。 * line\_terminator : 字符串，默认为 ‘\n’。在输出文件中使用的换行字符或字符序列 * quoting : CSV模块的可选常量。默认值为to\_csv.QUOTE\_MINIMAL。如果设置了浮点格式，那么浮点将转换为字符串，因此csv.QUOTE\_NONNUMERIC会将它们视为非数值的。 * quotechar : 字符串（长度1），默认“”。用于引用字段的字符 * doublequote : 布尔，默认为Ture。控制一个字段内的quotechar * escapechar : 字符串（长度为1），默认为None。在适当的时候用来转义sep和quotechar的字符 * chunksize : int或None。一次写入行 * date\_format : 字符串，默认为None。字符串对象转换为日期时间对象 * decimal: 。字符串，默认’。’。字符识别为小数点分隔符。例如。欧洲数据使用 ​​’，’。 |

|  |
| --- |
| Pandas导出Excel文件功能：to\_excel |
| to\_excel(excel\_writer, sheet\_name='Sheet1', na\_rep='', float\_format=None, columns=None, header=True, index=True, index\_label=None, startrow=0, startcol=0, engine=None, merge\_cells=True, encoding=None, inf\_rep='inf', verbose=True, freeze\_panes=None)  参数解释   * excel\_writer : 字符串或ExcelWriter 对象。文件路径或现有的ExcelWriter * sheet\_name :字符串,默认“Sheet1”。将包含DataFrame的表的名称。 * na\_rep : 字符串,默认‘ ’。缺失数据表示方式 * float\_format : 字符串,默认None。格式化浮点数的字符串 * columns : 序列,可选。要编写的列 * header : 布尔或字符串列表，默认为Ture。写出列名。如果给定字符串列表，则假定它是列名称的别名。 * index :布尔,默认的Ture。写行名（索引） * index\_label : 字符串或序列，默认为None。如果需要，可以使用索引列的列标签。如果没有给出，标题和索引为true，则使用索引名称。如果数据文件使用多索引，则需使用序列。 * startrow :左上角的单元格行来转储数据框 * startcol :左上角的单元格列转储数据帧 * engine : 字符串,默认没有。使用写引擎 - 您也可以通过选项io.excel.xlsx.writer，io.excel.xls.writer和io.excel.xlsm.writer进行设置。 * merge\_cells : 布尔,默认为Ture。编码生成的excel文件。 只有xlwt需要，其他编写者本地支持unicode。 * inf\_rep : 字符串,默认“正”。无穷大的表示(在Excel中不存在无穷大的本地表示) * freeze\_panes : 整数的元组(长度2)，默认为None。指定要冻结的基于1的最底部行和最右边的列。 |

【关键代码】

###########################################################

df\_new = df.loc[:,['姓名','考号','班级','语文','数学','英语']]

df\_new.to\_csv('studentScoreP\_new.csv', encoding='cp936',index=False)

#重新读取

df\_new = pd.read\_csv('studentScoreP\_new.csv', encoding='cp936')

df\_newGood = df\_new[df\_new['语文']>=100]

df\_newGood = df\_newGood[df\_new['语文']<=150]

df\_newGood = df\_newGood[df\_new['数学']>=100]

df\_newGood = df\_newGood[df\_new['数学']<=150]

df\_newGood = df\_newGood[df\_new['英语']>=100]

df\_newGood = df\_newGood[df\_new['英语']<=150]

df\_newGood.to\_csv('studentScoreP\_newGood.txt', encoding='cp936',index=False)

##########################################################

9.4.5 数据分类汇总模块详细设计

【分析】

在进行数据处理和分析时，经常需要按照某一列对原始数据进行分类汇总，即该列数值相同的行中其他列进行求和、求平均等操作，这可以通过groupby()、sum()方法和mean()方法等来实现。

本示例即按照字段“班级”分类汇总各班“语文”、“数学”、“英语”的班级总平均成绩。

【关键代码】

###########################################################

df\_new = pd.read\_csv('studentScoreP\_new.csv', encoding='cp936')

df\_new = df\_new.loc[:,['班级','语文','数学','英语']]

df\_new\_groupby = df\_new.groupby(['班级'], as\_index=False).mean()

print(df\_new\_groupby)

df\_new\_groupby.to\_csv('studentScoreP\_newGroup.txt', encoding='cp936',index=False)

##########################################################

9.4.6 数据计算及排序模块详细设计

【分析】

数据计算除了简单的四则运算外，还包括更为复杂的数据归一化、数据变换、数据滤波等机器学习算法。Pandas作为基于Numpy的工具，通过提供Series、DataFrame、Panels三种数据结构，为数据分析与计算提供了丰富的算法接口。

本示例需要计算三个字段的均值，直接在数据选择的基础上进行均值计算即可。

此外，本示例需要按照新字段“均值”对计算结果排序，则可以直接用Pandas的sort\_values函数。

【关键代码】

###########################################################

df\_new = pd.read\_csv('studentScoreP\_new.csv', encoding='cp936')

df\_new['均值'] = (df\_new['语文'] + df\_new['数学'] + df\_new['英语'])/3.0

df\_new.sort\_values(by='均值',ascending=False) # #按列对数据进行降序排序

print(df\_new.head())

df\_new.to\_excel('studentScoreP\_Mean.xlsx', encoding='cp936',index=False)

##########################################################

9.4.7 数据统计模块详细设计

【分析】

Pandas数据统计功能，可以直接调用describe()函数查看DataFrame数组的平均值、标准差、最大值、四分之三位数、中位数、四分之一位数、最小值等统计信息。

###########################################################################

>>> df.describe() #平均值、标准差、最小值、最大值等信息

A C D

count 4.000000 4.000000 4.0

mean 59.750000 2.500000 3.0

std 27.354159 1.290994 0.0

min 36.000000 1.000000 3.0

25% 42.750000 1.750000 3.0

50% 52.500000 2.500000 3.0

75% 69.500000 3.250000 3.0

max 98.000000 4.000000 3.0

###########################################################################

【关键代码】

###########################################################

df\_mean = pd.read\_excel('studentScoreP\_Mean.xlsx', encoding='cp936')

df\_mean\_describe = df\_mean.describe()

print(type(df\_mean\_describe))#<class 'pandas.core.frame.DataFrame'>

print(df\_mean\_describe)

maxValue = df\_mean\_describe.at['max','均值']

threeQuartersValue = df\_mean\_describe.at['75%','均值']

medianValue = df\_mean\_describe.at['50%','均值']

quarterValue = df\_mean\_describe.at['25%','均值']

minValue = df\_mean\_describe.at['min','均值']

category = [minValue, quarterValue, medianValue, threeQuartersValue,maxValue]

labels = ['Poor', 'Moderate', 'Good', 'Excellent']

mean\_cut = pd.cut(df\_mean['均值'],category, right=False, labels=labels)

print(mean\_cut)

print(type(mean\_cut))#<class 'pandas.core.series.Series'>

mean\_cut\_counts = mean\_cut.value\_counts()

print(mean\_cut\_counts)

print(type(mean\_cut\_counts))#<class 'pandas.core.series.Series'>

##########################################################

9.4.8 数据可视化模块详细设计

【分析】

数据可视化，是数据分析的重要组成部分。在Python生态系统中，包括诸如matplotlib、Seaborn、HoloViews、Altair、PyQtGraph、ggplot、Bokeh、pygal、VisPy、NetworkX、Plotly、geoplotlib、folium、Gleam、vincent、mpld3、python-igraph、missingno、Mayavi2、Leather等等众多的扩展库实现数据可视化功能。

不同的可视化库在不同方面具有不同的偏重，如Seaborn、Altair更偏重于专业统计图表绘制，PyQtGraph、VisPy、Mayavi2更适合于数学、工程等领域制图，NetworkX、python-igraph更适合于网络研究和分析制图，geoplotlib、folium适合于绘制地图，等等。

然而，在诸多的第三方可视化库中，matplotlib是Python中最为著名的绘图系统，很多其他的绘图系统如seaborn也是由其封装而来。此外，Pandas也提供了自己的绘图接口，并结合matplotlib实现更为丰富的功能。

|  |
| --- |
| Pandas—Series绘图 |
| 用法：Series.plot(kind='line', ax=None, figsize=None, use\_index=True, title=None, grid=None, legend=False, style=None, logx=False, logy=False, loglog=False, xticks=None, yticks=None, xlim=None, ylim=None, rot=None, fontsize=None, colormap=None, table=False, yerr=None, xerr=None, label=None, secondary\_y=False, \*\*kwds)  参数：  （1）kind : str类型，包含‘line’ : line plot (折线图，default)；‘bar’ : vertical bar plot（垂直柱状图）；‘barh’ : horizontal bar plot（水平柱状图）；‘hist’ : histogram（直方图）；‘box’ : boxplot（箱形图）；‘kde’ : Kernel Density Estimation plot（密度图）；‘density’ : same as ‘kde’（密度图）；‘area’ : area plot(面积图)；‘pie’ : pie plot（饼图）  （2）ax : matplotlib axes object。If not passed, uses gca() |

|  |
| --- |
| Pandas—DataFrame绘图 |
| 用法：DataFrame.plot(x=None, y=None, kind='line', ax=None, subplots=False, sharex=None, sharey=False, layout=None, figsize=None, use\_index=True, title=None, grid=None, legend=True, style=None, logx=False, logy=False, loglog=False, xticks=None, yticks=None, xlim=None, ylim=None, rot=None, fontsize=None, colormap=None, table=False, yerr=None, xerr=None, secondary\_y=False, sort\_columns=False, \*\*kwds)  参数：  （1）x和y即指的是指数据框列的标签或位置参数。  （2）kind : str类型，包含‘line’ : line plot (折线图，default)；‘bar’ : vertical bar plot（垂直柱状图）；‘barh’ : horizontal bar plot（水平柱状图）；‘hist’ : histogram（直方图）；‘box’ : boxplot（箱形图）；‘kde’ : Kernel Density Estimation plot（密度图）；‘density’ : same as ‘kde’（密度图）；‘area’ : area plot(面积图)；‘pie’ : pie plot（饼图）；scatter（散点图）；  ‘hexbin’ : hexbin plot.  （3）ax : matplotlib axes object。If not passed, uses gca() |

本示例中将主要采用Pandas提供的绘图接口，并借用matplotlib扩展库实现坐标轴美观、图保存等功能。

【关键代码】

###########################################################################

#任务5绘图（柱状图和饼状图）

#柱状图

plt.figure()

mean\_cut\_counts.plot(kind='bar',figsize=(12,8))

plt.xticks(rotation=0,fontsize=16)

plt.yticks(fontsize=16)

plt.title("平均成绩离散化统计柱状图")

plt.savefig('studentScoreP\_Mean\_bar.png',dpi=400)

plt.show()

#饼状图

plt.figure()

mean\_cut\_counts.plot(kind='pie',figsize=(12,8))

plt.title("平均成绩离散化统计饼状图")

plt.savefig('studentScoreP\_Mean\_pie.png',dpi=400)

plt.show()

#任务6绘图

df\_mean = pd.read\_excel('studentScoreP\_Mean.xlsx', encoding='cp936')

df\_mean = df\_mean.loc[:,['姓名','均值']]

#利用matplotlib绘图

plt.figure(figsize=(12,8))

plt.plot(df\_mean['姓名'],df\_mean['均值'],label = '均值',color='red')

plt.xlabel('姓名',fontsize=12)

#姓名每隔5个抽样显示

xlength = len(df\_mean)

print('xlength=',xlength)#65

#构建xticks显示位置

xticksloc = [i for i in range(xlength) if i%5==0]

print('xticksloc=',xticksloc)#[0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60]

#构建xticks显示标签

xtickslabels = df\_mean['姓名'].values[::5]

print('xtickslabels',xtickslabels)

plt.xticks(xticksloc,xtickslabels,rotation=30)#倾斜30度显示

plt.ylabel('平均分',fontsize=12)

plt.legend(fontsize=16)#显示图例并设置字号

plt.title("学生平均成绩",fontsize=16)

plt.savefig('studentScoreP\_Mean.png',dpi=400)

plt.show()

###########################################################################

* 1. 完整参考代码

###########################################################################

"""

Created on Sun Apr 12 16:00:42 2020

@author: Jining Yan

运行环境：Windows 10系统，Python 3.7

"""

import pandas as pd

import matplotlib.pyplot as plt

import os

#防止中文出现乱码

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

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

#【任务1】

def dataPreprocessing():

while True:

#读取数据

fileName = input('请输入要打开的文件名studentScore.csv:')

try:

df = pd.read\_csv(fileName, encoding='cp936')

#丢弃缺失值

df = df.dropna()

df.to\_csv('studentScoreP.csv', encoding='cp936',index=False)

##查看前三行

print(df.head(3))

##查看后两行

print(df.tail(2))

print("任务1执行成功！")

break

except: #打开文件失败时执行的代码

print('文件不存在,请重新输入文件名')

#【任务2】

def dataSelection():

while True:

#读取数据

fileName = input('请输入要打开的文件名studentScoreP.csv:')

try:

df = pd.read\_csv(fileName, encoding='cp936')

df\_new = df.loc[:,['姓名','考号','班级','语文','数学','英语']]

try:

df\_new.to\_csv('studentScoreP\_new.csv', encoding='cp936',index=False)

df\_new = pd.read\_csv('studentScoreP\_new.csv', encoding='cp936')

df\_newGood = df\_new[df\_new['语文']>=100]

df\_newGood = df\_newGood[df\_new['语文']<=150]

df\_newGood = df\_newGood[df\_new['数学']>=100]

df\_newGood = df\_newGood[df\_new['数学']<=150]

df\_newGood = df\_newGood[df\_new['英语']>=100]

df\_newGood = df\_newGood[df\_new['英语']<=150]

df\_newGood.to\_csv('studentScoreP\_newGood.txt', encoding='cp936',index=False)

print("任务2执行成功！")

break

except:#打开文件失败时执行的代码

print('文件导出失败！')

except:#打开文件失败时执行的代码

print('文件不存在,请重新输入文件名')

#【任务3】

def dataGroup():

while True:

#读取数据

fileName = input('请输入要打开的文件名studentScoreP\_new.csv:')

try:

df\_new = pd.read\_csv(fileName, encoding='cp936')

df\_new = df\_new.loc[:,['班级','语文','数学','英语']]

df\_new\_groupby = df\_new.groupby(['班级'], as\_index=False).mean()

print(df\_new\_groupby)

df\_new\_groupby.to\_csv('studentScoreP\_newGroup.txt', encoding='cp936',index=False)

print("任务3执行成功！")

break

except: #打开文件失败时执行的代码

print('文件不存在,请重新输入文件名')

#【任务4】

def dataCalculate():

while True:

#读取数据

fileName = input('请输入要打开的文件名studentScoreP\_new.csv:')

try:

df\_new = pd.read\_csv(fileName, encoding='cp936')

df\_new['均值'] = (df\_new['语文'] + df\_new['数学'] + df\_new['英语'])/3.0

df\_new.sort\_values(by='均值',ascending=False) #按列对数据进行降序排序

df\_new.to\_excel('studentScoreP\_Mean.xlsx', encoding='cp936',index=False)

print("任务4执行成功！")

break

except: #打开文件失败时执行的代码

print('文件不存在,请重新输入文件名')

#【任务5】

def dataDescribeVisualization():

while True:

#读取数据

fileName = input('请输入要打开的文件名studentScoreP\_Mean.xlsx:')

try:

df\_mean = pd.read\_excel(fileName, encoding='cp936')

df\_mean\_describe = df\_mean.describe()

print(type(df\_mean\_describe))#<class 'pandas.core.frame.DataFrame'>

print(df\_mean\_describe)

maxValue = df\_mean\_describe.at['max','均值']

threeQuartersValue = df\_mean\_describe.at['75%','均值']

medianValue = df\_mean\_describe.at['50%','均值']

quarterValue = df\_mean\_describe.at['25%','均值']

minValue = df\_mean\_describe.at['min','均值']

category = [minValue, quarterValue, medianValue, threeQuartersValue,maxValue]

labels = ['Poor', 'Moderate', 'Good', 'Excellent']

mean\_cut = pd.cut(df\_mean['均值'],category, right=False, labels=labels)

print(mean\_cut)

print(type(mean\_cut))#<class 'pandas.core.series.Series'>

mean\_cut\_counts = mean\_cut.value\_counts()

print(mean\_cut\_counts)

print(type(mean\_cut\_counts))#<class 'pandas.core.series.Series'>

#柱状图

plt.figure()

mean\_cut\_counts.plot(kind='bar',figsize=(12,8))

plt.xticks(rotation=0,fontsize=16)

plt.yticks(fontsize=16)

plt.title("平均成绩离散化统计柱状图")

plt.savefig('studentScoreP\_Mean\_bar.png',dpi=400)

plt.show()

#饼状图

plt.figure()

mean\_cut\_counts.plot(kind='pie',figsize=(12,8))

plt.title("平均成绩离散化统计饼状图")

plt.savefig('studentScoreP\_Mean\_pie.png',dpi=400)

plt.show()

print("任务5执行成功！")

break

except: #打开文件失败时执行的代码

print('文件不存在,请重新输入文件名')

#【任务6】

def dataVisualization():

while True:

#读取数据

fileName = input('请输入要打开的文件名studentScoreP\_Mean.xlsx:')

try:

df\_mean = pd.read\_excel(fileName, encoding='cp936')

df\_mean = df\_mean.loc[:,['姓名','均值']]

#利用matplotlib绘图

plt.figure(figsize=(12,8))

plt.plot(df\_mean['姓名'],df\_mean['均值'],label = '均值',color='red')

plt.xlabel('姓名',fontsize=12)

#姓名每隔5个抽样显示

xlength = len(df\_mean)

print('xlength=',xlength)#65

#构建xticks显示位置

xticksloc = [i for i in range(xlength) if i%5==0]

print('xticksloc=',xticksloc)#[0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60]

#构建xticks显示标签

xtickslabels = df\_mean['姓名'].values[::5]

print('xtickslabels',xtickslabels)

plt.xticks(xticksloc,xtickslabels,rotation=30)#倾斜30度显示

plt.ylabel('平均分',fontsize=12)

plt.legend(fontsize=16)#显示图例并设置字号

plt.title("学生平均成绩",fontsize=16)

plt.savefig('studentScoreP\_Mean.png',dpi=400)

plt.show()

print("任务6执行成功！")

break

except: #打开文件失败时执行的代码

print('文件不存在,请重新输入文件名')

#系统主界面

def menu():

print('【任务选择】\n'

'＋－－－－学生成绩数据分析及可视化系统 －－－－－－－－－－＋\n'

'｜0、退出。 ｜\n'

'｜1、数据读取及预处理。 ｜\n'

'｜2、数据选择及导出。 ｜\n'

'｜3、数据分类汇总。 ｜\n'

'｜4、数据计算及排序。 ｜\n'

'｜5、数据统计。 ｜\n'

'｜6、数据可视化。 ｜\n'

'＋－－－－－－－－－－－－－－－－－－－－－－－－－-－－－＋')

#功能选择模块

def task():

while True:

menu()#打印系统主界面

num = input("请输入任务选项：")

if num == '1':

dataPreprocessing()

elif num == '2':

if os.path.exists('studentScoreP.csv'):

dataSelection()

else:

print('未能执行当前选项，请先执行前面的选项！')

elif num == '3':

if os.path.exists('studentScoreP\_new.csv'):

dataGroup()

else:

print('未能执行当前选项，请先执行前面的选项！')

elif num == '4':

if os.path.exists('studentScoreP\_new.csv'):

dataCalculate()

else:

print('未能执行当前选项，请先执行前面的选项！')

elif num == '5':

if os.path.exists('studentScoreP\_Mean.xlsx'):

dataDescribeVisualization()

else:

print('未能执行当前选项，请先执行前面的选项！')

elif num == '6':

if os.path.exists('studentScoreP\_Mean.xlsx'):

dataVisualization()

else:

print('未能执行当前选项，请先执行前面的选项！')

elif num == '0':

print('程序结束！')

break

else:

print('输入选项有误')

input("回车显示菜单")

#主函数

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

task()#调用功能选择函数

###########################################################################

* 1. 结果分析

1. 通过统计一班二班语文、数学、英语三科成绩均高于100分的情况，发现共42名同学三科均高于100分，约占学生总数的62%（42/68=0.62）。
2. 通过按照班级分类汇总语文、数学、英语三科平均成绩，可以发现一班的语文班级平均成绩108.76与二班的语文平均成绩107.23基本持平；一班的数学平均成绩105.76明显高于二班的数学平均成绩99.88；一般的英语平均成绩111.17明显高于二班的英语平均成绩106.82。单纯语文、数学、英语三科班级平均成绩来看，一班稍优于二班。
3. 通过计算两个班级各同学语文、数学、英语三科平均成绩来看，三科最高平均成绩139分，三科最低平均成绩41。其中三科平均成绩高于130分的共8位同学，一班二班各占4位；三科平均成绩低于60分的共4位同学，一班一位二班3位。