《Big Data Analytics》实验报告

年级、专业、班级	2018 级机械一班	姓名	易弘睿	学号	20186103
实验题目	Pandas 数据转换				
实验时间	2022 年 3 月 29 日	实验地点	A 理 119		
学年学期	2021-2022(2)	实验性质	□验证性 ■设计性 □综合性		

一、实验目的

1) 异形数据文件的读写和解析；
2) 掌握 Pandas 的使用；
3) 掌握 Pandas 的数据转换；

二、实验项目内容

根据计算机学院 2022 年硕士研究生复试成绩公示中的成绩完成如下事宜：

1)分别统计和输出两个专业每个分数段的人数。每 10 分为一个分数段，最低分数段从复试分数线开始，最高分数段为全专业最高分所在的整 10 分数段。

2)分别统计和输出两个专业录取的考生中每个分数段的人数。每 10 分为一个分数段，最低分数段从复试分数线开始，最高分数段为全专业最高分所在的整 10 分数段。

3)在已有的 DataFrame 中添加一列“录取结果”用于表明考生是否录取。该列取值为“录取”和“待定”两类。普通考生根据录取指标确定考生是否录取，特殊考生（‘少高’，‘退役士兵’）复试成绩合格(>=60 分)即可录取。

4)在已有的 DataFrame 中添加一列“奖学金等级”用于表明考生的奖学金等级。该列的取值为 A、B、C 三等。A 等为全额奖学金，占比 40%，B 等为半额奖学金，占比 40%，C 等为无奖学金，占比 20%。普通考生按招生总指标计算各等奖学金指标数后进行分配。普通考生中的推免生优先占有 A 等和 B 等奖学金。特殊考生（‘少高’，‘退役士兵’）直接 A 等奖学金，且不占用奖学金指标。

三、实验过程和结果

实验过程：

```python
import pandas as pd
import numpy as np


def read_files(path):
    with open(path) as f:
        lines = f.readlines()
    return lines


def line_process(sub_line, headers, stud_dict):
    sub_line = sub_line.split()
    student_inf = sub_line[11:]
    saving_inf = [[] for _ in headers]
    for student_index in range(0, len(student_inf), 10):
        for inf_index, sub_inf in enumerate(saving_inf):
            if inf_index == len(saving_inf)-1:
                sub_inf.append(None)
            else:
                sub_inf.append(student_inf[student_index+inf_index])
    if len(stud_dict.get(headers[0], [])) == 0:
        new_dict = dict(zip(headers, saving_inf))
    else:
        for index_inf, (keys, values) in enumerate(stud_dict.items()):
            values.extend(saving_inf[index_inf])
        new_dict = stud_dict
    return new_dict
```

```python
def read_in_pandas(path):
    total_lines = read_files(path=path)
    sample_line = total_lines[0].split()
    header = sample_line[:11]

    student_dict = {}
    for line in total_lines:
        student_dict = line_process(line, header, student_dict)

    total_df = pd.DataFrame(student_dict)
    total_df = total_df.set_index("序号")
    total_df["初试总分"] = total_df["初试总分"].values.astype(int)
    total_df["复试总分"] = total_df["复试总分"].values.astype(float)
    total_df["综合成绩"] = total_df["综合成绩"].values.astype(float)
    # 返回学生表
    return total_df
```

```python
def Q1(stud_df, threshold):
    print("所有学生的成绩分布如下：")
    categories = set(stud_df["报考专业"])
    for category in categories:
        category_df = stud_df[stud_df["报考专业"] == category]
        print("专业：{} 总人数：{}".format(category, len(category_df)))
        category_low_grade = threshold[category]
        category_high_grade = category_df["初试总分"].max()
        for grade in range(category_low_grade, category_high_grade, 10):
            nums = len(category_df[(category_df["初试总分"] >= grade) & (category_df["初试总分"] < grade+10)])
            print("分数区间：{} - {} 人数：{}".format(grade, grade+10, nums))


def Q2(stud_df, threshold):
    print("已通过复试的成绩分布如下：")
    categories = set(stud_df["报考专业"])
    for category in categories:
        category_df = stud_df[(stud_df["报考专业"] == category) & (stud_df["录取结果"] == "录取")].copy(deep=True)
        print("专业：{} 通过人数：{}".format(category, len(category_df)))
        category_low_grade = threshold[category]
        category_high_grade = category_df["初试总分"].max()
        for grade in range(category_low_grade, category_high_grade, 10):
            nums = len(category_df[(category_df["初试总分"] >= grade) & (category_df["初试总分"] < grade+10)])
            print("分数区间：{} - {} 人数：{}".format(grade, grade+10, nums))
```

```python
def Q3(stud_df, enroll):
    stud_df["录取结果"] = "待定"
    categories = set(stud_df["报考专业"])
    for category in categories:
        category_df = stud_df[stud_df["报考专业"] == category].copy(deep=True)
        special_student = category_df[(category_df["专项计划"] != "无") & (category_df["复试总分"] >= 60)]
        for special_index in special_student.index:
            stud_df.loc[special_index, "录取结果"] = "录取"
            category_df = category_df.drop([special_index], axis=0)
        remain_nums = enroll[category] - len(special_student)
        category_df.sort_values(by=['综合成绩'], ascending=False, inplace=True)
        for change_index in category_df.iloc[:remain_nums].index:
            stud_df.loc[change_index, "录取结果"] = "录取"
    print(stud_df)
    return stud_df


def Q4(stud_df):
    stud_df["奖学金等级"] = None
    categories = set(stud_df["报考专业"])
    for category in categories:
        category_df = stud_df[(stud_df["报考专业"] == category) & (stud_df["录取结果"] == "录取")].copy(deep=True)
        category_df.sort_values(by=['综合成绩'], ascending=False, inplace=True)
        category_df["奖学金等级"] = pd.qcut(list(category_df["综合成绩"]), [0, 0.2, 0.6, 1], labels=["C", "B", "A"])
        special_student = category_df[category_df["专项计划"] != "无"]
        category_df.loc[special_student.index, "奖学金等级"] = "A"
        for changing_index in category_df.index:
            stud_df.loc[changing_index, "奖学金等级"] = category_df.loc[changing_index, "奖学金等级"]
    writer = pd.ExcelWriter("研究生奖学金.xlsx")
    stud_df.to_excel(writer)
    writer.save()
```

```python
file_path = "计算机学院2022年硕士研究生复试成绩公示.txt"
stu_df = read_in_pandas(file_path)
lowest_grade = {"081200计算机科学与技术": 330, "085400电子信息": 360}
enroll_nums = {"081200计算机科学与技术": 13, "085400电子信息": 70}
Q1(stu_df, lowest_grade)
stu_df = Q3(stu_df, enroll_nums)
Q2(stu_df, lowest_grade)
Q4(stu_df)
```

实验结果：

```
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['D:\\OneDrive - University of Cincinnati\\2022 Spring\\大数据\\pythonProject', 'D:/OneDrive

Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)]
所有学生的成绩分布如下：
专业：081200计算机科学与技术 总人数：16
分数区间：330 - 340 人数：6
分数区间：340 - 350 人数：3
分数区间：350 - 360 人数：2
分数区间：360 - 370 人数：4
专业：085400电子信息 总人数：104
分数区间：360 - 370 人数：29
分数区间：370 - 380 人数：36
分数区间：380 - 390 人数：20
分数区间：390 - 400 人数：8
分数区间：400 - 410 人数：7
分数区间：410 - 420 人数：2
                考生编号    姓名       报考学院             报考专业  ...   复试总分   综合成绩    备注   录取结果
序号                                                          ...
1    106112014080017   曾龙  014计算机学院  081200计算机科学与技术  ...  93.92  79.84  None    录取
2    106112014080015  陈宇泽  014计算机学院  081200计算机科学与技术  ...  93.14  79.60  None    录取
3    106112014080067  陈光华  014计算机学院  081200计算机科学与技术  ...  84.60  77.04  None    录取
4    106112014080077  晏文龙  014计算机学院  081200计算机科学与技术  ...  87.68  76.84  None    录取
5    106112014080005   王浩  014计算机学院  081200计算机科学与技术  ...  88.52  76.82  None    录取
..                ...  ...       ...              ...  ...    ...    ...   ...   ...
116  106112514080137   刘阳  014计算机学院       085400电子信息  ...  72.46  73.12  None    待定
117  106112514080263  刘青帝  014计算机学院       085400电子信息  ...  72.78  72.79  None    待定
118  106112514080422  陶彦辰  014计算机学院       085400电子信息  ...  70.26  72.46  None    待定
119  106112514080236   孔超  014计算机学院       085400电子信息  ...  72.98  72.43  None    待定
120  106112514080680  梁中豪  014计算机学院       085400电子信息  ...  81.22  66.79  None    录取

[120 rows x 11 columns]
已通过复试的成绩分布如下：
专业：081200计算机科学与技术 通过人数：13
In[3]:
```

四、实验总结与体会

本次实验，主要完成了 1)分别统计和输出两个专业每个分数段的人数；2)分别统计和输出两个专业录取的考生中每个分数段的人数；3)在已有的 DataFrame 中添加一列“录取结果”用于表明考生是否录取；4)在已有的 DataFrame 中添加一列“奖学金等级”用于表明考生的奖学金等级。该列的取值为 A、B、C 三等。

通过本实验，我更真切认识到不同的算法对于程序运行时间的影响，并通过额外增加一列用于表示是否被录取，初始化所有的录取结果都为待定，有效解决了 Q3 的问题。

实验报告填写说明：

1、第一、二部分由老师提供；

2、第三部分填写主要操作步骤（文字）与结果 3（截图）；

3、第四部分主要填写遇到的问题与解决问题的方法、总结和体会等；

4、报告规范：包含报告页眉、报告的排版、内容是否填写，命名是否规范等内容；

5、实验文件与实验报告命名：学号姓名序号.*（应用对应的扩展名），学号姓名(en).docx，例如学号 20161234 的张三同学，**他的第一次实验命名为：20161234 张三(e1)1.docx**