|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名 | 学号 | 邮箱 | 练习题完成数量 | 分工 |
| 赵茁羽 | 171250006 | 171250006@smail.nju.edu.cn | 188 | 学生编程习惯分析 |
| 李轩 | 171250572 | 171250572@smail.nju.edu.cn | 139 | 编程伙伴推荐 |
| 王子卓 | 171830579 | 171830579@smail.nju.edu.cn | 144 | 学生编程能力雷达图，整体样本分析 |

1. 研究问题：构建学生编程能力画像，分析学生编程习惯和编程水平。

分析学生提交用例时间和得分，帮助同学了解自己的编程习惯，择优选择一天中效率最高的时间段进行编程活动，如若是面向 DDL 编程类型，我们希望提醒学生加以改进。同时按八种题型构建八边形学生编程水平雷达图，综合各题难度更准确地确定学生能力水平，让学生清晰地认识到自己的编程水平，查漏补缺。进一步，我们希望为学生推荐编程习惯相似、编程水平相补的学习伙伴。

1. 代码开源地址：

https://github.com/njuhhhh/Programming-feature-analysis

1. 研究方法
   1. 学生编程能力：

首先确定每种类型题目的数量，定义r为某类型题目的的提交完成度，r=（某同学该类型提交的题目数/该类型题目的总数），每道题目的得分为x1，x2,x3....,该类型题目的平均得分X=（（x1+x2+x3+...）/该类型题目总数）。某学生该类型的编程能力得分Y=X\*r。八类题型分别计算画出雷达图。学生的整体综合编程得分T=(各类型题目得分总和/8)。

* 1. 学生一天内效率最高的编程时间段：

将一天划分为多个时间段，非、规定每时间段效率=(该时间段非零分提交次数/非零分提交总次数)\*该时间段学生平均得分，得出学生一天中编程效率最高的时间段。

学生作业完成类型：计算每天完成的题数（100 分算完成），此数据为时间序列，进行ADF 检验测试，通过结果可以看出绝大部分数据都是平稳分布的，所以选择直接进行判断。

* + 1. 拖后完成：题目完成集中在 DDL 前几天
    2. 平稳完成：题目完成时间平均分配在任务期间
    3. 提前完成：题目完成集中在任务开始时
  1. 编程搭档推荐：

综合了前两块的内容。以编程水平互补为主要条件，编程习惯相似为次要条件推荐学习伙伴。

首先将学生分为3类。第一类是所有题目都完成了的学生，我们认为这类学生不需要编程搭档，他们可以帮助任何一个人。第二类是题目完成量较小的同学，我们将每个人得分最高的一个type的平均分计算出来，如果一位同学的优势type未能超过20分，我们认为这位学生编程能力较差，需要第一类同学，即全部完成的同学帮助。第三类则是大多数人，需要找到一个合适的编程搭档。

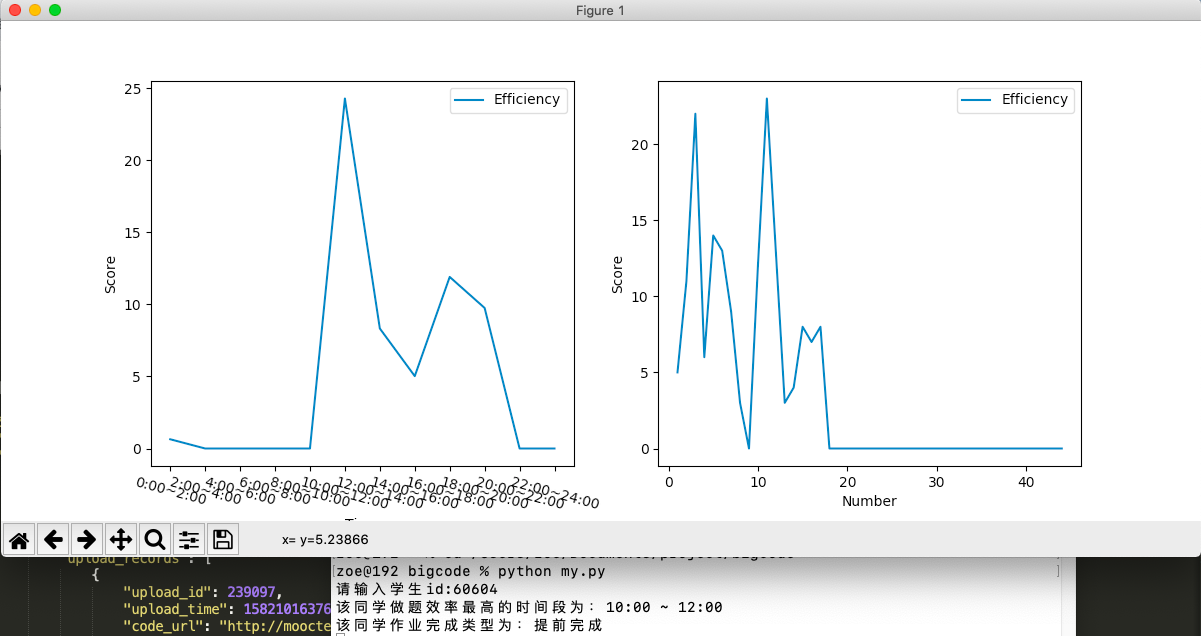
对于第三类人，我们采取了以下方法。首先，根据“学生编程能力”的研究方法，算出一位学生的弱项与其得分，再得出其强项与其得分。然后按照“其弱项是别人强项”及“其强项是别人弱项”的标准得到两个list，并取两个list的交集，得到一些搭档的候选人。最后，再根据第二块的“编程习惯”的研究算法，找到与其编程能力互补，编程习惯相似的最佳CP

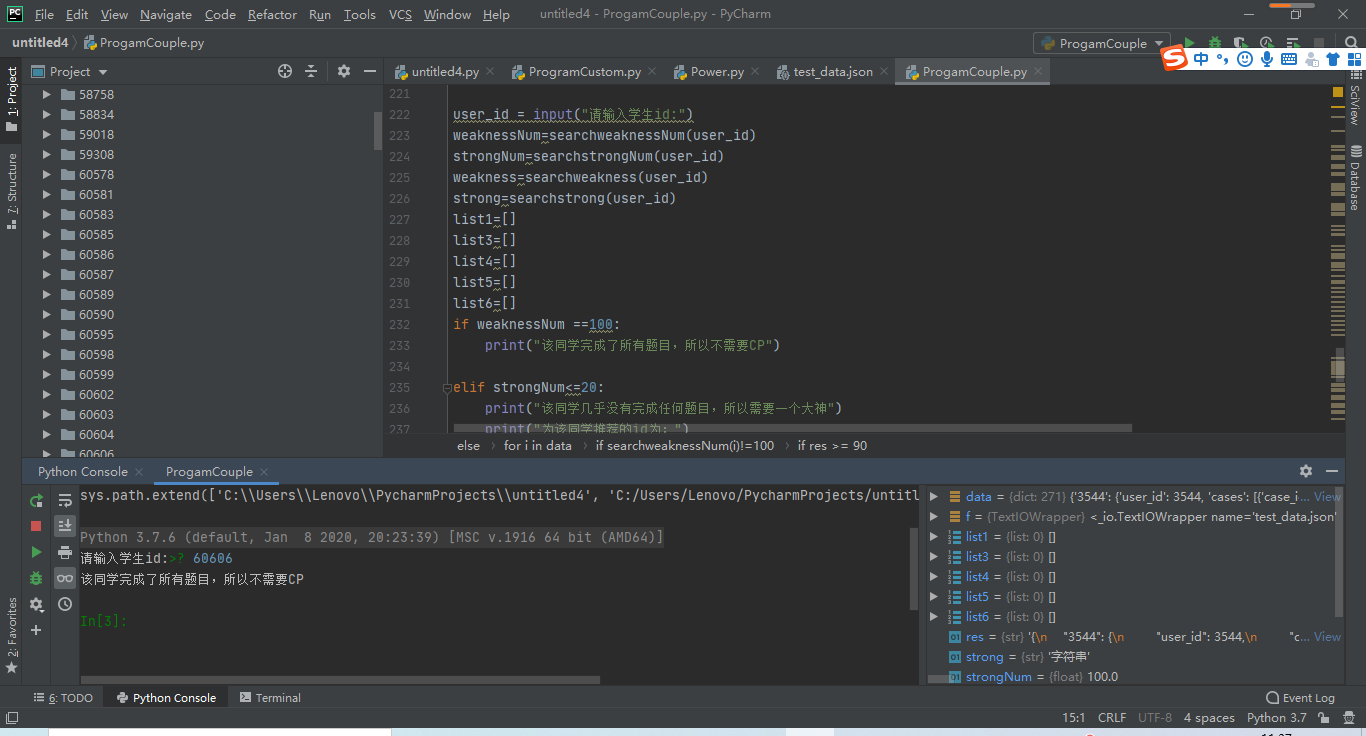
* 1. 整体样本分析

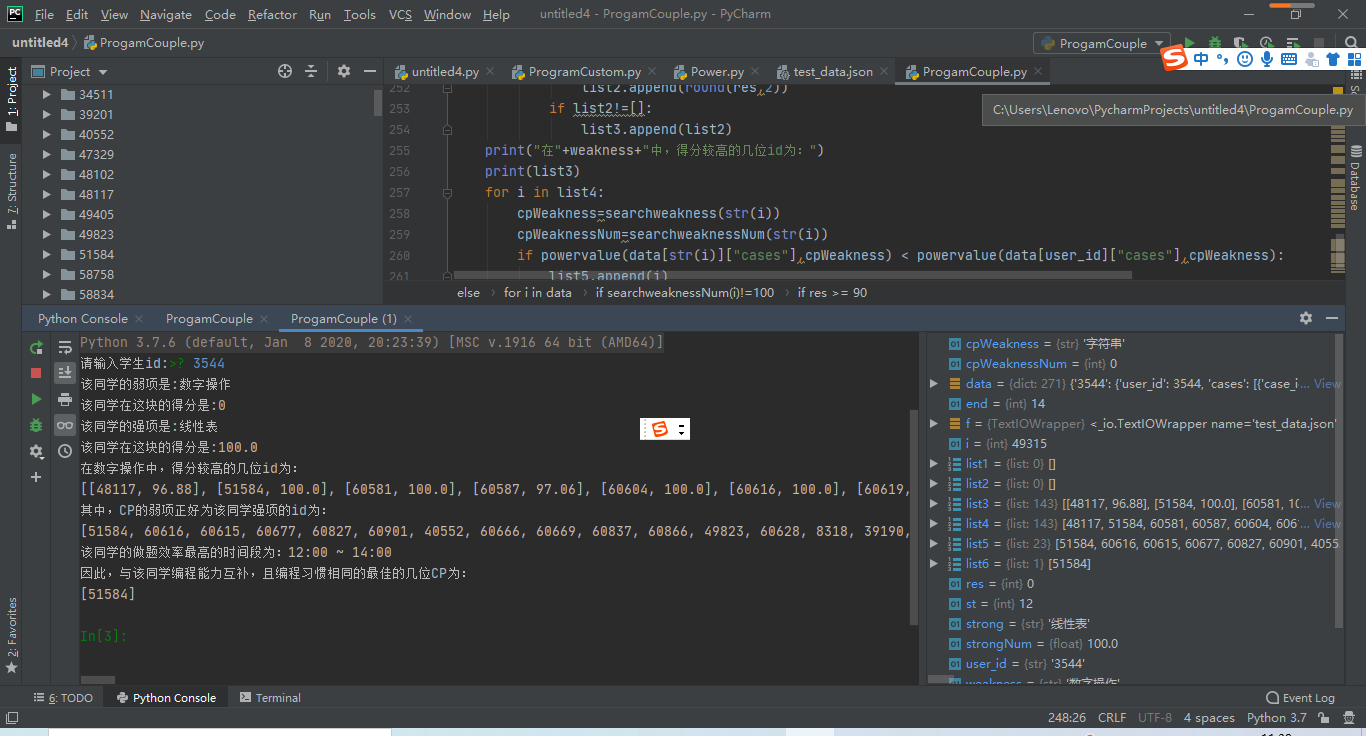
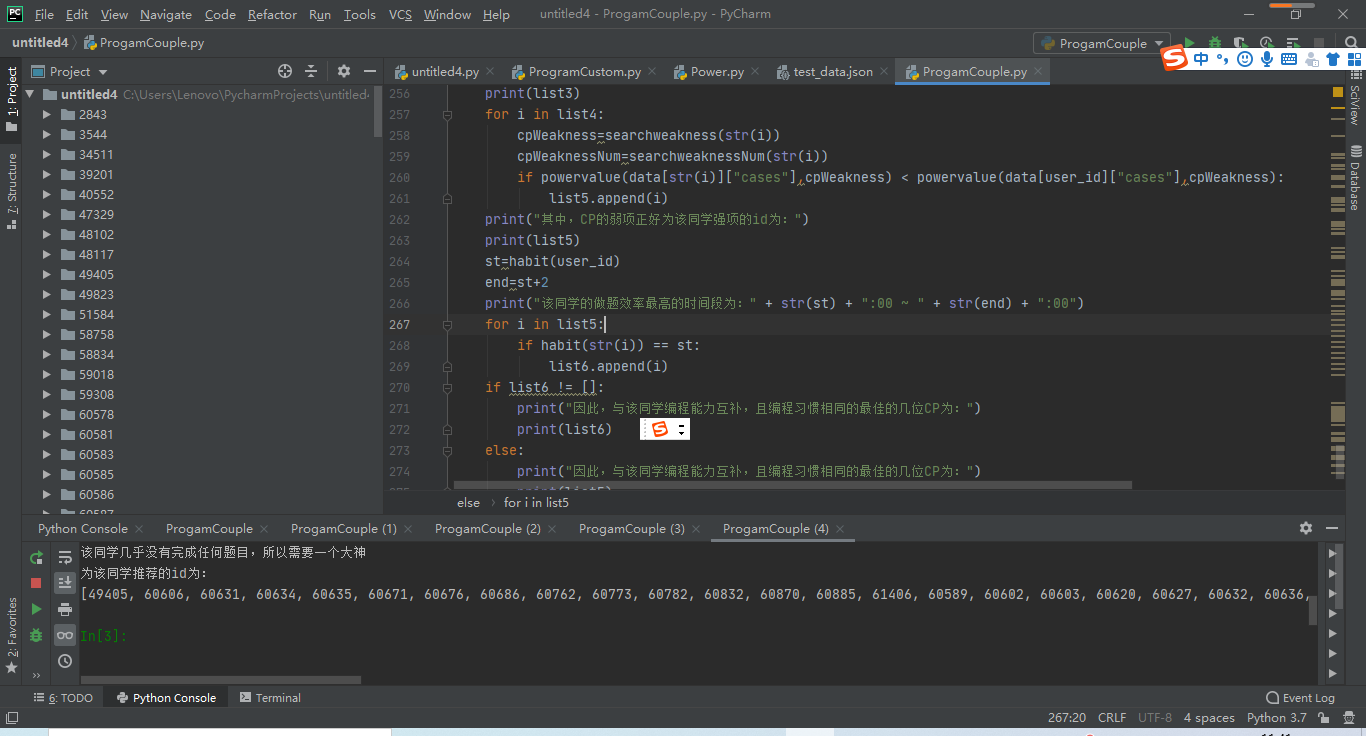
1. 基于研究方法1得到学生的综合编程能力，对所有样本进行统计分析，
2. 基于样本的分布结果进行进一步研究

四、案例分析

学生编程习惯分析代码实际操作如下：

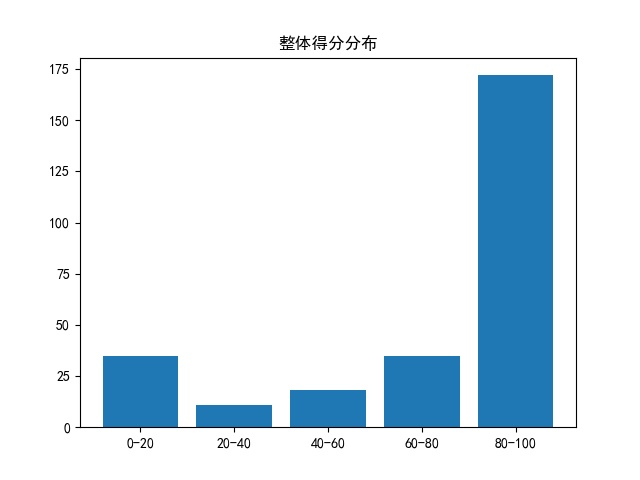






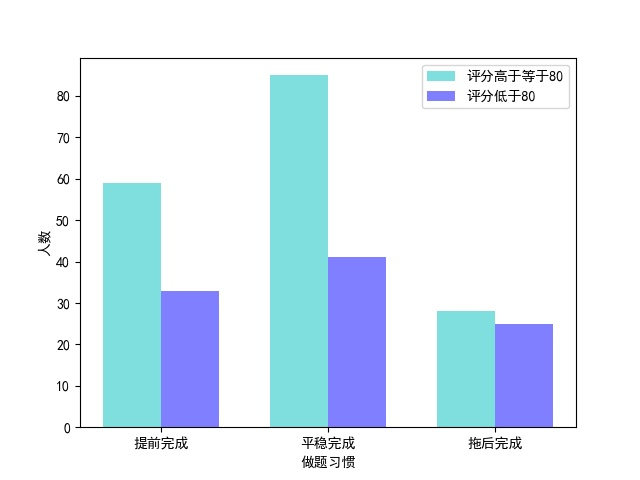
整体样本分析

基于研究方法1得到学生的综合编程能力，对所有样本进行统计分析，结果分数分布如下：



由图可知分数集中分布在0-80和80-100区间，再基于研究方法2得到学生的编程习惯类型（“提前完成”、“平稳完成”和“拖后完成”），对整体样本进行编程综合得分与编程习惯类型的关系研究，将总体样本分为6组类型（“提前完成且得分在0-80间”、“平稳完成且得分在0-80间”、“拖后完成且得分在0-80间”、“提前完成且得分在80-100间”、“平稳完成且得分在80-100间”和“拖后完成且得分在80-100间”）

结果如下图：



小结：

1、基于研究结果，评分高于80的同学大部分做题习惯属于“提前完成”和“平稳完成”，其中“平稳完成”的同学中高分人数最多，而“拖后完成”的同学得分情况不容乐观，造成低分的人数比例较高。

2、从本研究样本中抽取的个别样本出现了编程习惯是“提前完成”但是平均得分处于个位数的低分的情况，造成这种情况的一种可能原因是该同学在题目发布时积极进行了一段时间的练习，但未能坚持一直做题。这也可能是“提前完成”的同学中的高分比例比“平稳完成”的同学低的原因

3、综合以上结论得出，推荐同学们在编程练习时可以选择积极地提前做题，但更重要的是坚持做题，劳逸结合。不推荐将题目拖后完成，这样很可能因时间不足给自己带来巨大压力和造成最终得分较低的结果。

五、课程感想

目前本课程主要分为两部分，一部分是老师讲概率论知识，一部分是自主做 python 练习题，个人感觉两部分目前是完全割裂的，希望可以综合起来，除了大作业外再加些用python编程结合概率论知识解决实际问题的教学内容。