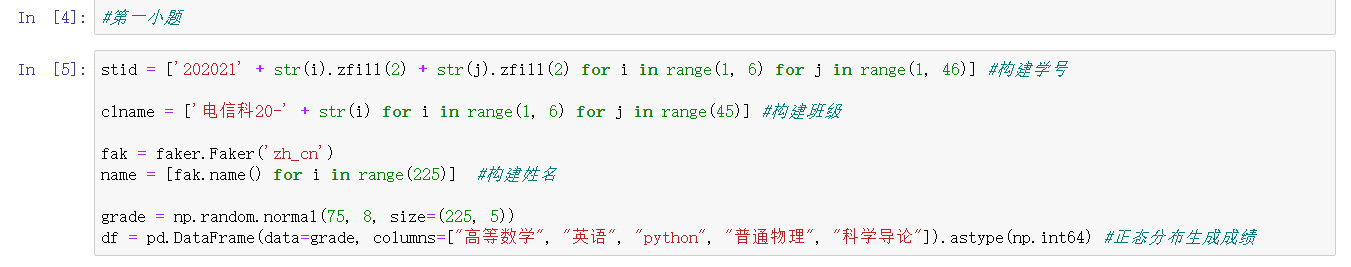
**第九次作业**

姓名：洪祎麟 班级：电信科20-1班 学号：2020217793

**第一题：**

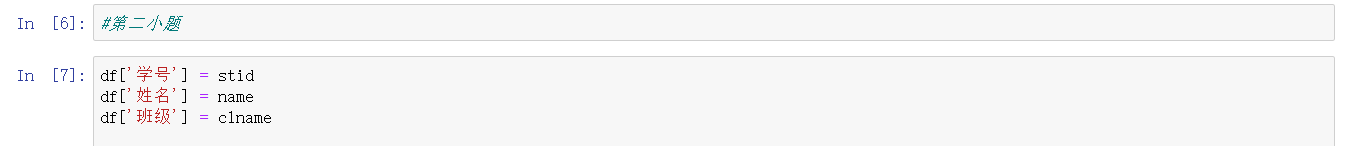
第一小题：

根据题意，第一小题需要生成学生们的学号，班级，姓名和五门课的成绩这些数据。故分别进行生成。对于学号，由于在for in range循环中出现的个位数不会显示“01,02…”，因此使用str的zfill方法，将其填充至两位，使得所有通过列表推导式生成的学号格式正确。对于班级名，同样也使用列表推导式生成。对于学生的姓名，需要引入Faker库，通过Fak.name（）生成5个班共225名学生的姓名。针对学生们5门科目的成绩，这里需要运用正态分布，使得整体的成绩更为准确。故这里使用np.random.normal函数，以8为σ，75为中间点，生成每位学生5门科目的成绩，并放在一个DataFrame中。具体第一题的代码截图如下：

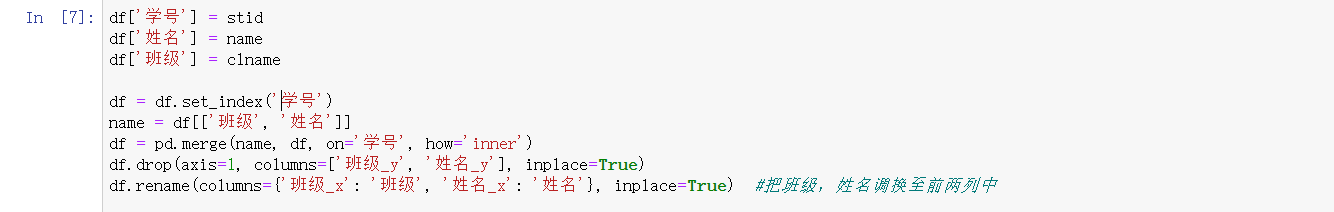


第二小题：

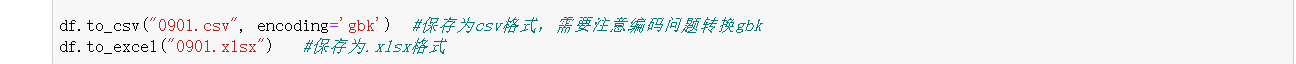
由于第一题已经创建了DataFrame，故第二题开始时需要将第一题所生成的班级，姓名和学号添加到第一题成绩所在的DataFrame表中。因此：



由于题目要求以学生的学号作为索引，各列数据的次序应该调整为：班级、姓名、高等数学，英语、Python、普通物理、科学导论。在这里，我使用了合并后在删除的方法，先提取出班级，姓名所在的两列，再将这两列与之前所创建的DataFrame表进行合并，最后删除掉重复出现的姓名和班级，获得的即为符合要求的DataFrame。



最后，将数据保存到.csv格式和.xlsx格式下。由于在csv格式下，若使用utf-8格式保存会出现乱码，故应该添加参数，使得以gbk模式保存。



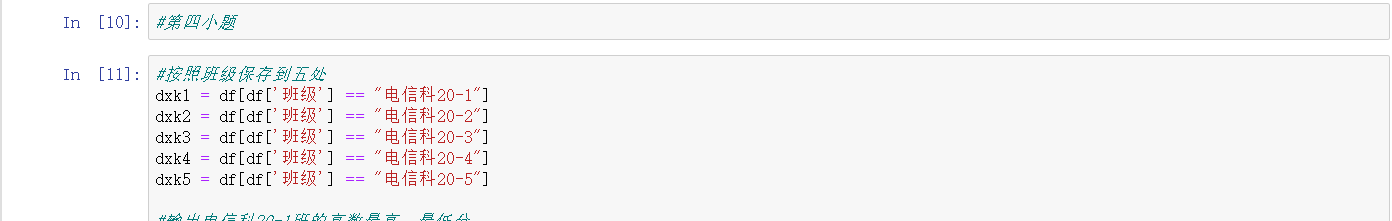
第三小题：

本题目需要输出每门课程的最高分、最低分和平均分。在这里可以使用panda库中所还有的describe方法，通过describe方法提取出数据的最大值，最小值和平均值。具体代码和输出结果如下：

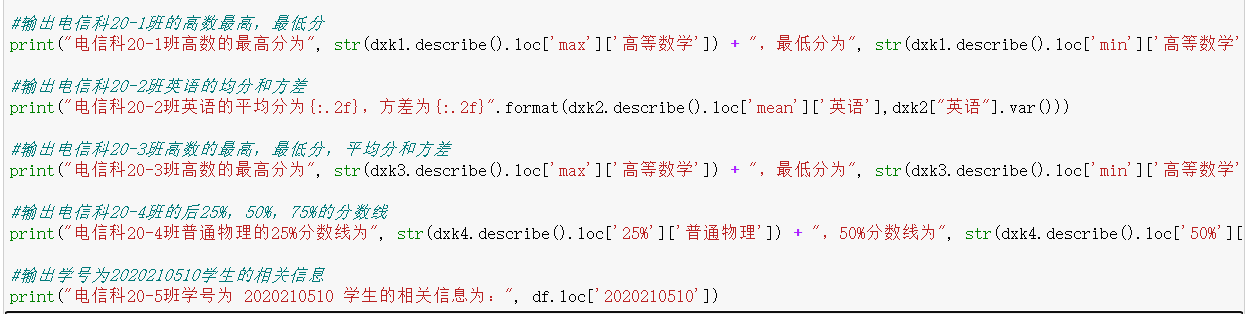


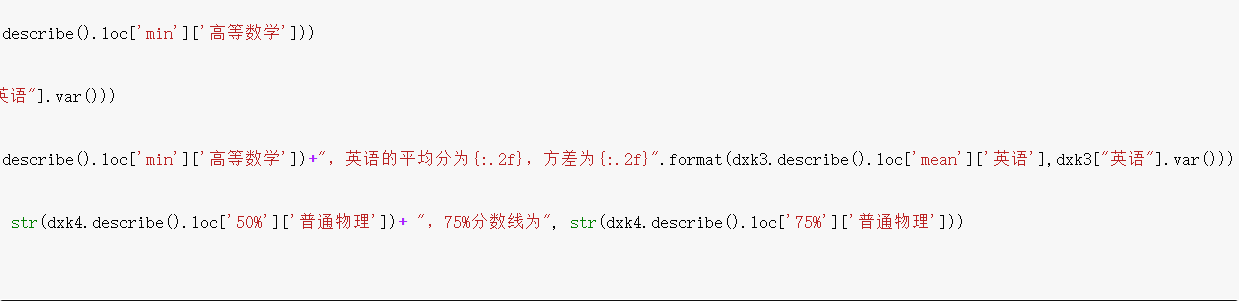
第四小题：

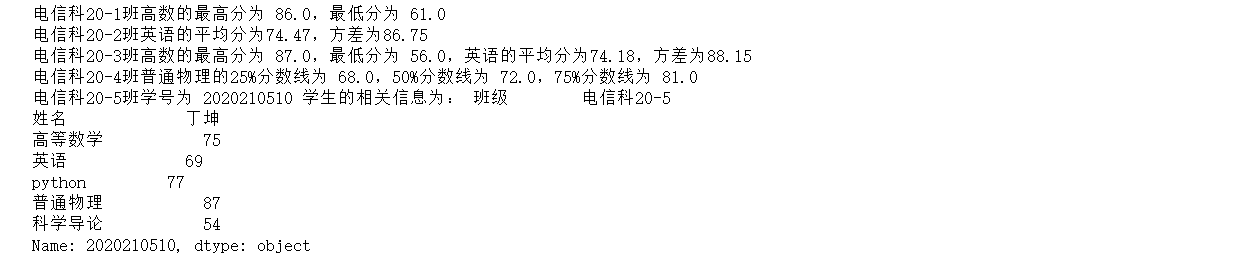
本题需要按班级分组来进行统计处理，求解输出所要求的数据。故这里，我先单独根据班级名称，提取出五个班级学生的所有数据。



再用panda库中所含有的describe和var方法，求出所要求的最大值，最小值，方差，平均值，25%，50%，75%。在这里，由于需要对数据进行精确抓取，故这里使用了.loc的方法，便于准确的找到所要统计的数据。对于方差和均分，这里保留了两位小数。具体的代码实现和输出结果如下：

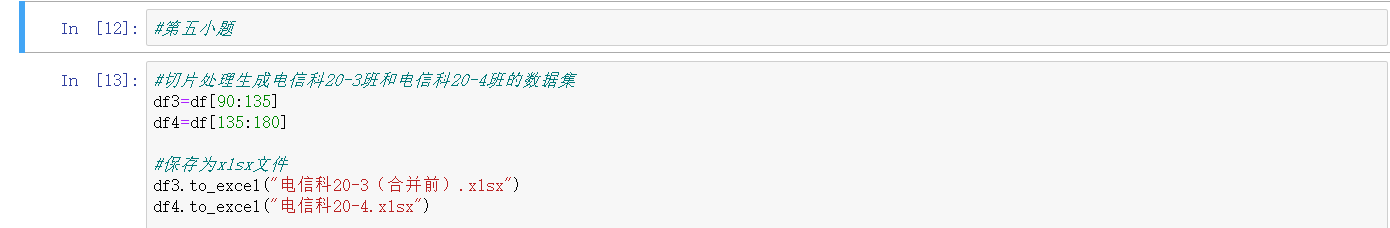






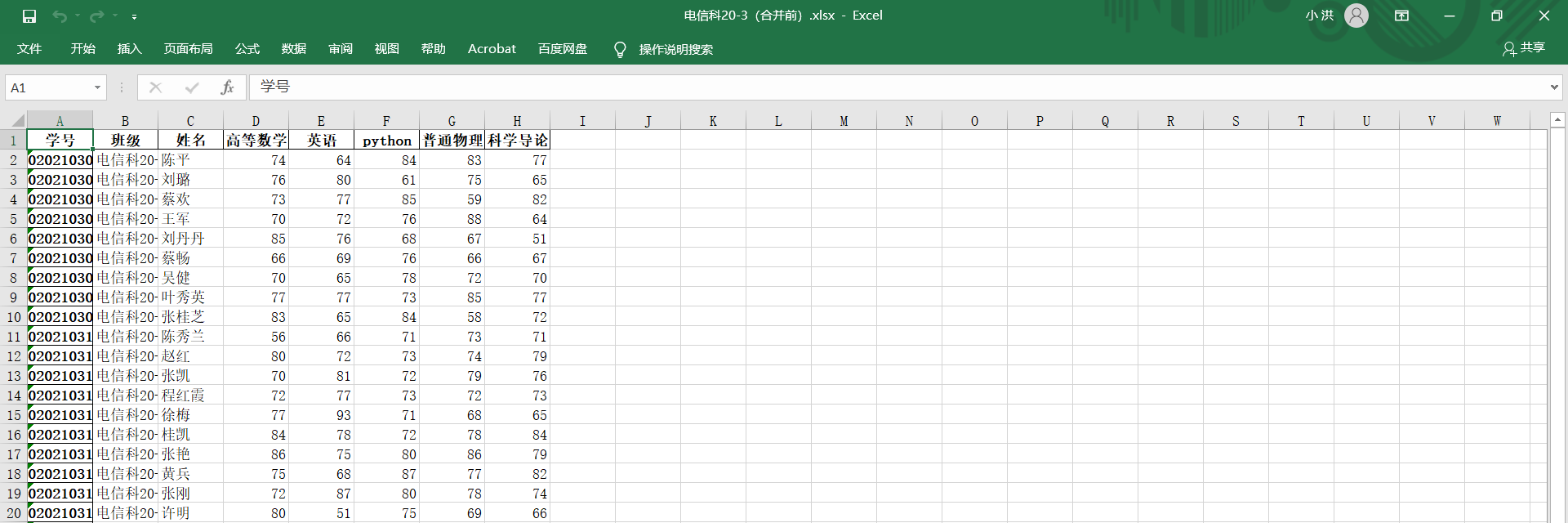
第五小题：

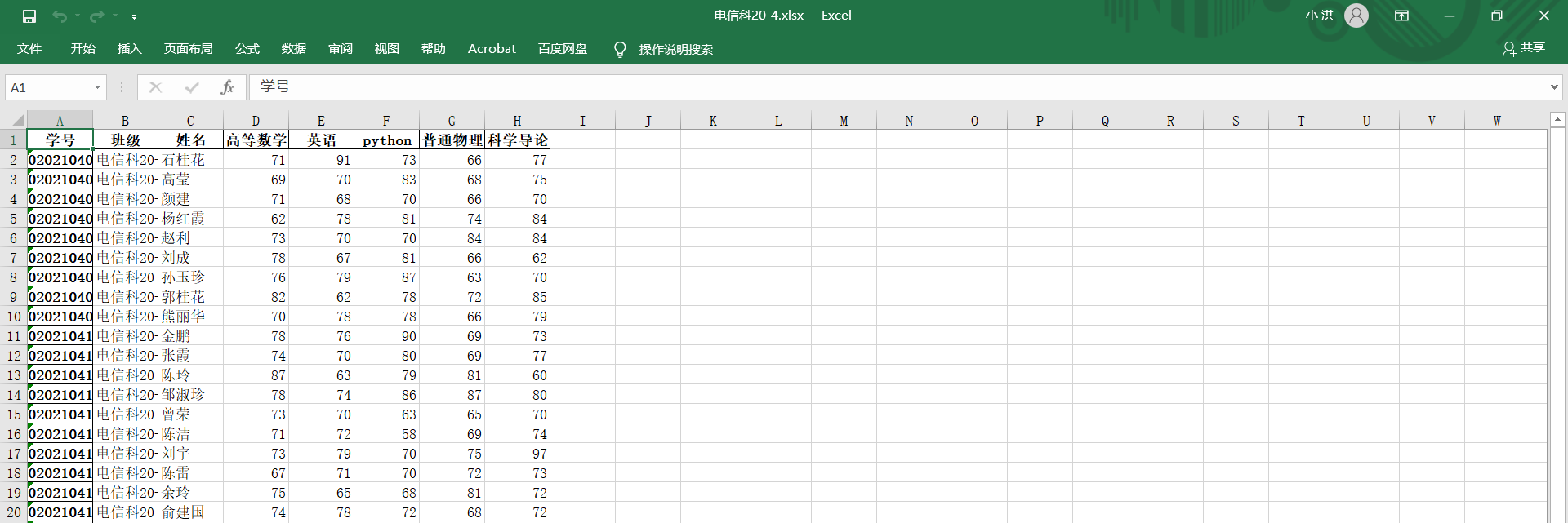
本题首先，需要运用切片处理生成一个电信科 20-3 和电信科 20-4 班学生的 DataFrame 数据集，在保存为xlsx文件。这里可以直接根据每班学生的人数灵活切片，代码如下：

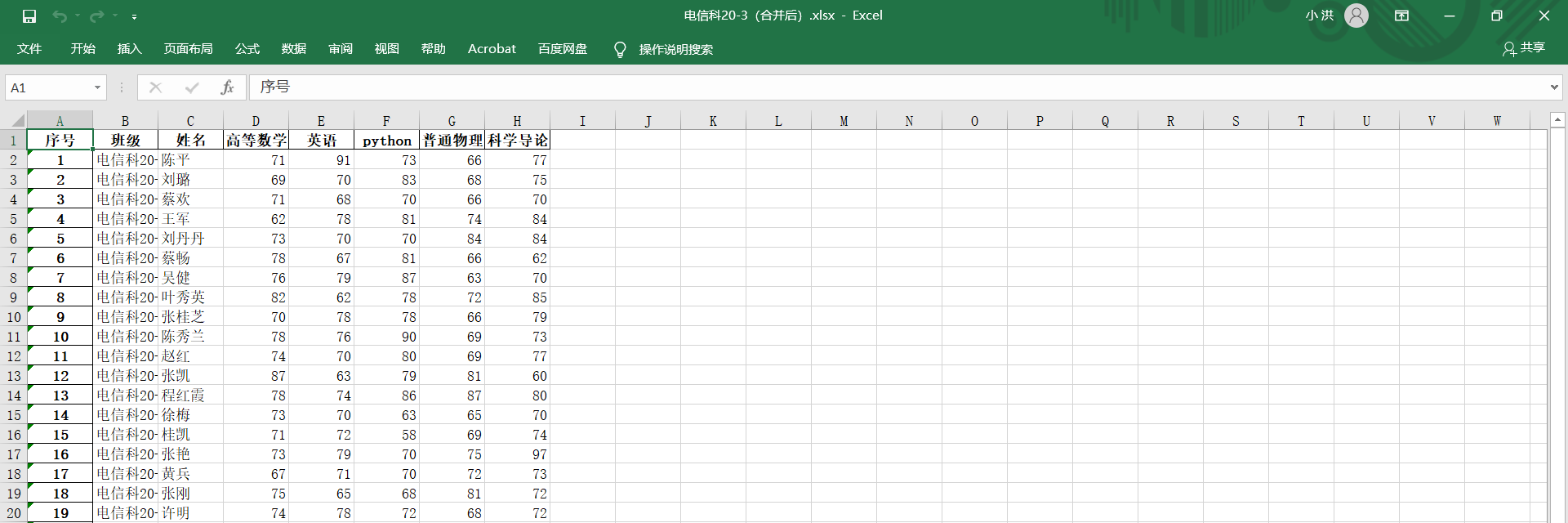


之后，需要以3班的姓名和4班的成绩合并为一个DataFrame数据集，保存为xlsx文件。这里，我们也可以跟第二小题一样，使用merge的方法。但由于两个数据集没有重复的一列，无法直接合并，因此可以人工生成一列“序号”，让两列根据“序号”进行合并。在这里，一定要使用.loc[:,“序号”]的模式，否则将会报出panda库的警告，提示我们最好不要这样做。在合并后，我们需要将代码转换为xlsx文件，为了与之前进行区分，这里将名字改为“电信科20-3（合并后）”。可以看出，合并后学生的成绩和合并前不同，与20-4班的学生成绩相同。本题的具体代码和生成的xlsx文件的部分截图如下：



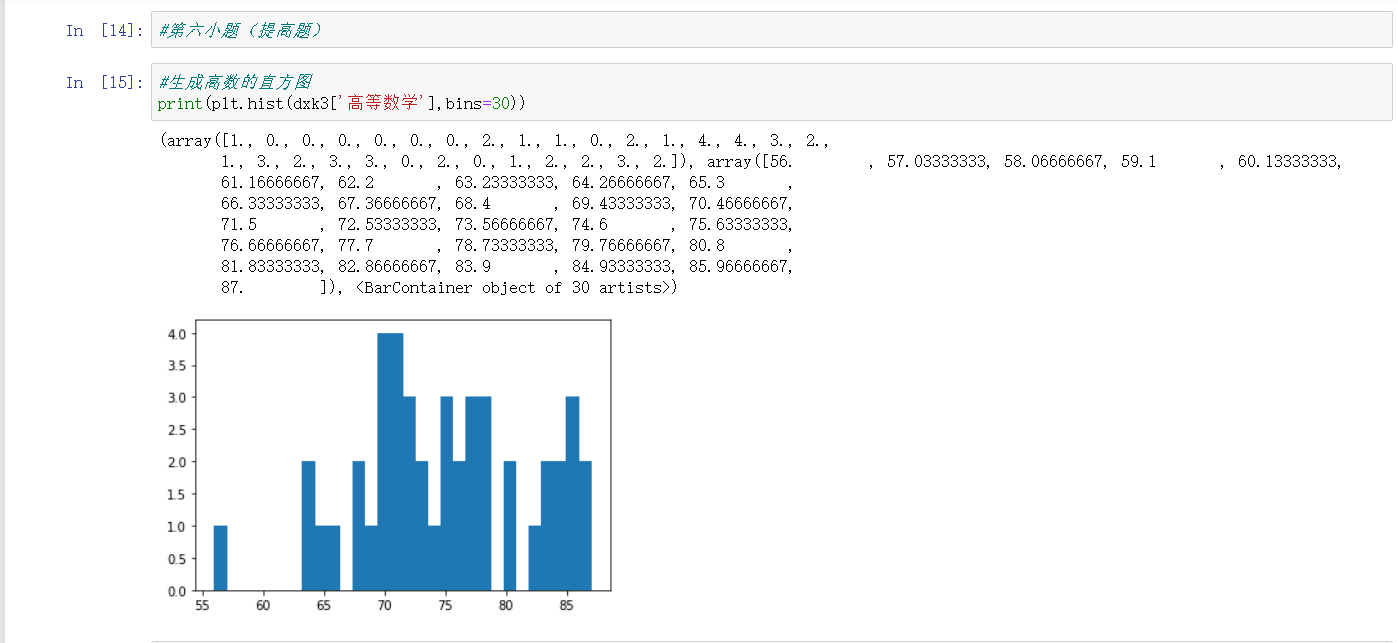


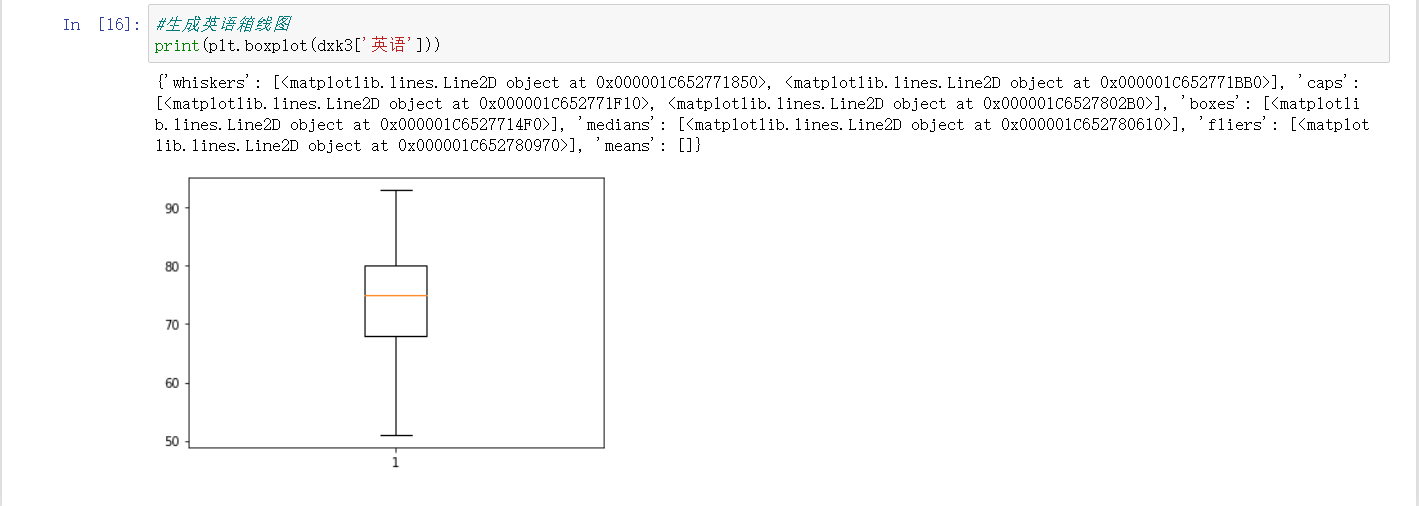




提高题：

本题需要我们根据matplotlib库，做出电信科20-3班学生的高等数学成绩直方图和英语课成绩的箱线图。根据matplotlib库中所含有的hist方法，可生成直方图；所含有的boxplot方法，可生成箱线图。从两个图表可以清楚的看出，成绩位于中等分数段的学生人数最多，成绩非常高的学生和成绩非常低的学生相对都较少。本题的代码和生成的图表截图如下：





**第二题：**

准备工作：

本题的案例是基于课件的。在进行做题之前，需要先定义好函数作为准备，便于后续题目中的数据分析。

首先需要引用所需的库包，再设计函数。

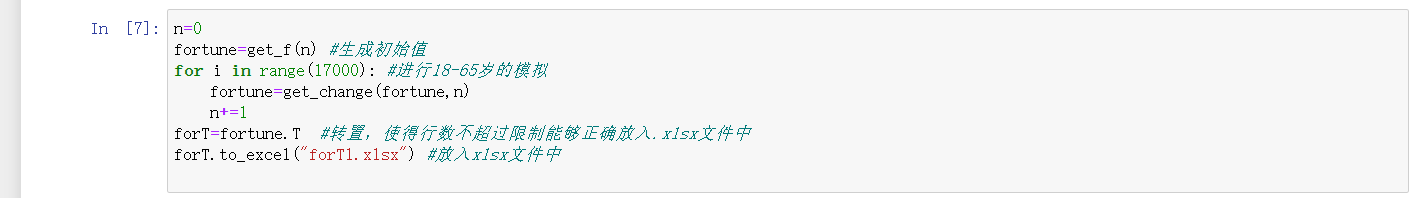
get\_f(n)函数主要是来生成初始的100个人，每个人的财富值都为100。其中通过列表推导式创建每个人的初始财富均为100，并将其索引换为每个人的编号，即可生成最初始的情况，便于后续的操作。

get\_change函数主要用于生成每次获得钱的情况。由于每个人获得钱的次数并不固定，因此需要运用count方法进行统计。在每一次获得钱之后，需要运用merge方法与原表格进行连接。为了方便连接，这里把索引改名为id，使得可以通过id来将两个表格连接。对于每个人每一次的情况，他的钱数即为他上一次的钱数-支出的一块钱+获得的钱数，将这个结果返回。

由于18-65岁我们近似看成是17000天，故我们需要模拟17000次，这里我们使用for循环进行模拟，由于excel中的行数是有限制的，17000远远超过了限制，故最后我们需要对生成的DataFrame数据集进行转置，这样才可以正确的放入.xlsx文件中。

本部分的具体代码实现如下：





第一小题：

在第一小题中，我们需要获得65岁时每个人拥有的钱数。由于之前的准备工作，我们的DataFrame的17000行就是65岁时每个人的数据。同时，我们可以用.iloc的限制条件筛选出在65岁时，财富大于100的每个人和总人数。用获得的总人数除以最开始的100人，则可以看出创业的成功率。具体实现代码和输出结果如下：



第二小题：

在40岁时，经历了（40-18）\*365=8030天，故我们需要对数据行索引为“8030”的数据进行分析，由于需要统计哪些人在此时财富小于0，故与第一题相同添加.loc的判断条件即可。具体代码实现和输出结果如下图：



第三小题：

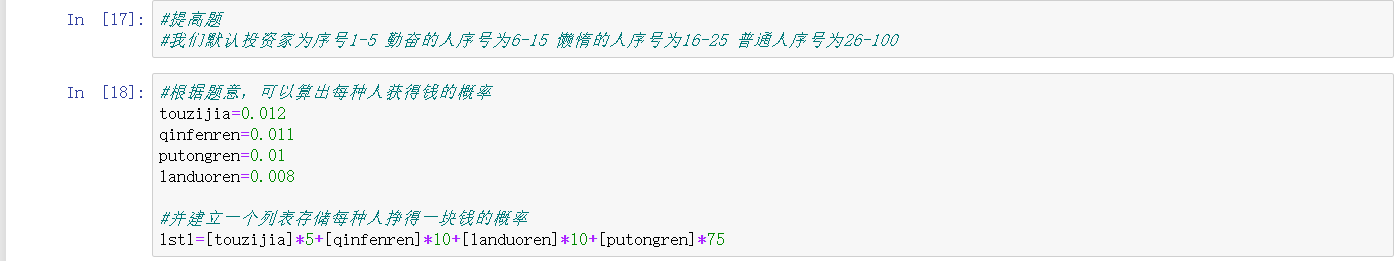
本题需要将第一小题的结果与第二小题的结果进行比较。在第二小题中，我们已经统计了所有在40岁时钱小于0的人，我们可以与第一题65岁时，财富数大于0和大于100的人相对比。这里，我采取了“&”的方式。最开始使用的是“and”，但由于报错，查询了资料才知道这里只能使用“&”表示且的关系。在用“&”后，由于剩下的结果只有True和False，因此需要将所有False的结果删去（此处可以用dropna()的方式处理），同时将所有为“True”结果的索引进行提取，放入列表中，与原数据集对应，输出他们序号所对应的65岁的财富值。这样，可以更加直观的与第二小题的结果进行对比，看出他们在40-65岁这25年间财富值发生的变化情况。根据所得出的结果，可以看出40岁-65岁间一逆负债情况变成不负债的人大约占4成，而不但逆转了负债情况，甚至开始致富的人几乎没有。由此可以清楚的看出，40岁开始创业很难，成功率并不高。本题的代码和输出的结果如下图：



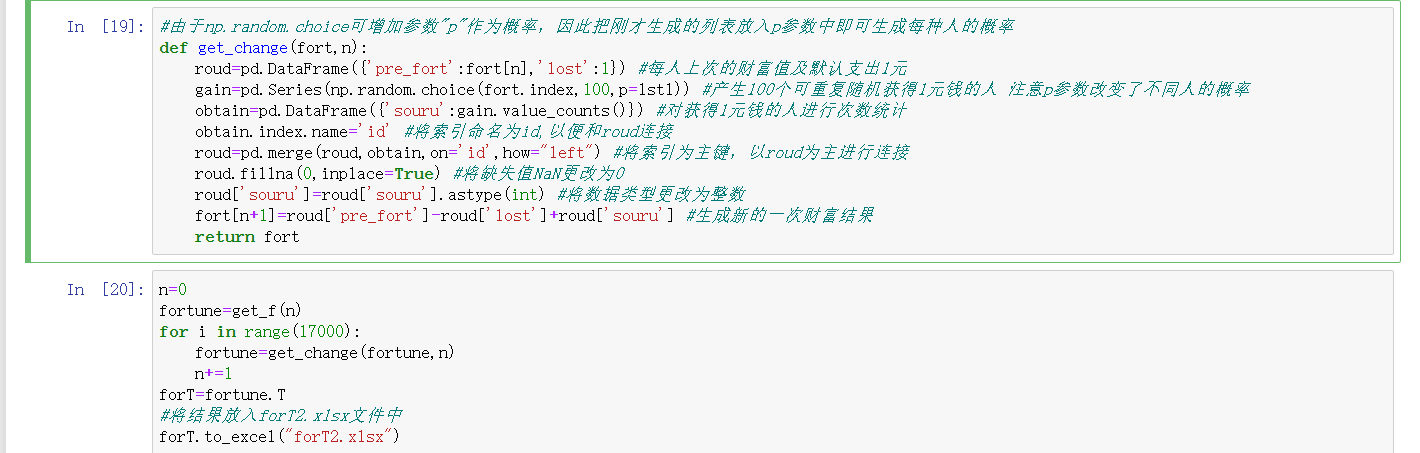
提高题：

根据题目的要求，本题分成了四种人。金融投资家，勤奋的人，懒惰的人和普通的人。为了方便，我们使用“touzijia”，“qinfenren”，“putongren”，“landuoren”来代替这四种人。其中，编号1-5为金融投资家，编号6-15为勤奋的人，编号16-25为懒惰的人，编号26-100为普通人。在本题中，我们需要把这四个人获得一块钱的概率进行调整和修正。由于np.random.choice方法中有参数“p”可以调整每个数据所对应的概率，故我们可以利用它来改变这四种人所对应的概率。

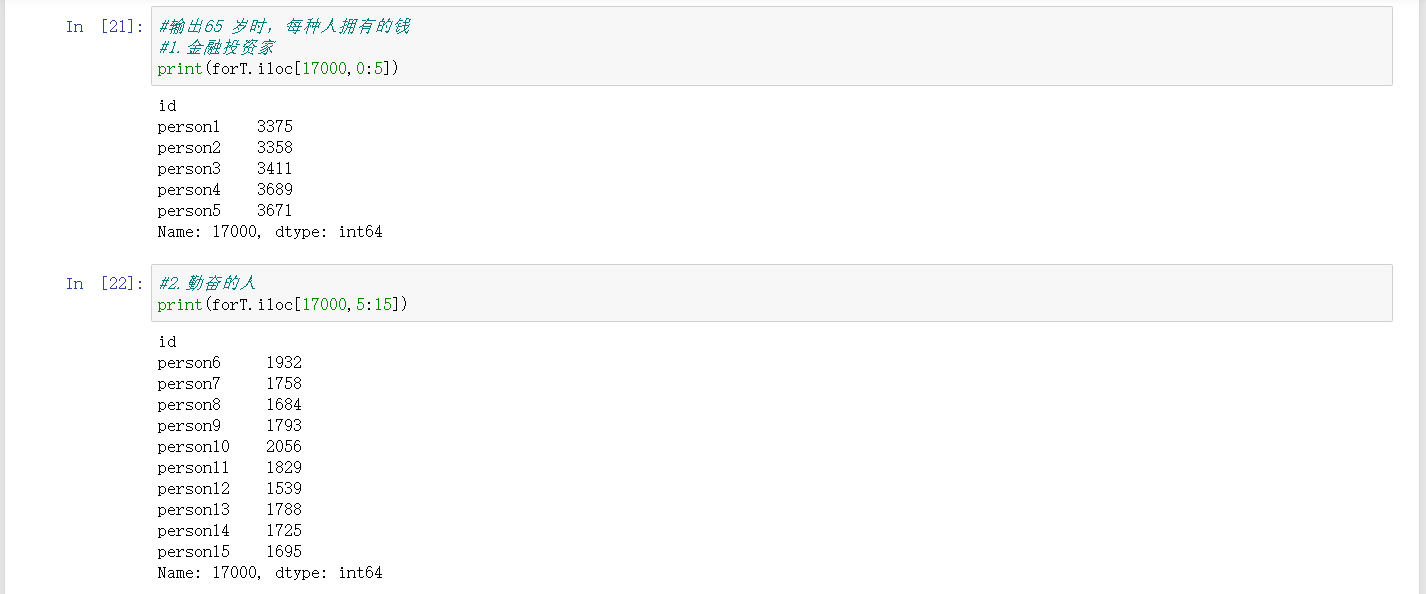
首先可以将对应的人对应的序号的概率生成放到列表中存储，便于之后“p”参数直接调用。

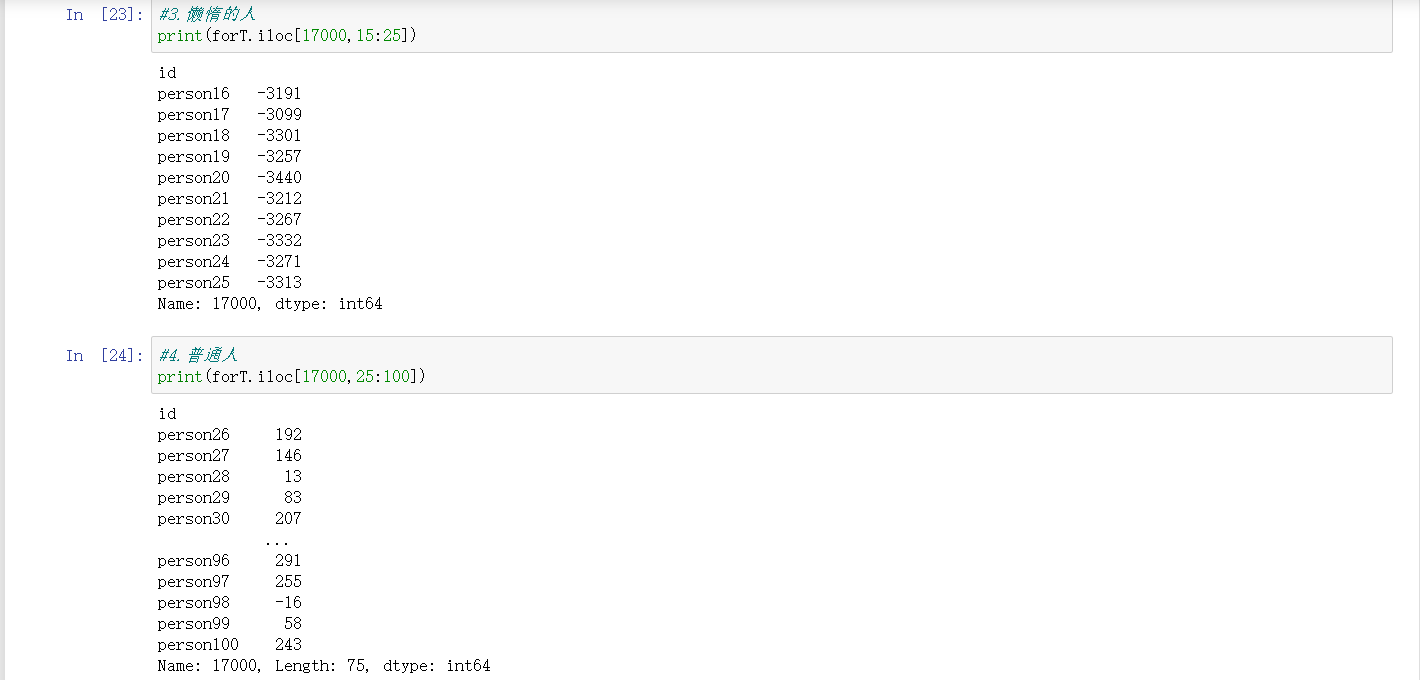


然后同样可以使用之前所使用的一系列方法，生成18-65岁的全部数据，具体代码如下：



然后，对四种人在65岁的结果分别输出：





从四种人65岁的结果可以看出，普通人基本上过了这么多年赚不到什么大钱，基本只能保本甚至本钱都拿不到；金融投资家各个都能致富，而且能翻本三十多倍；做一个勤奋的人也能致富，虽然没有投资家赚得多，但也能翻20倍左右；做一个懒惰的人只会亏钱，每天懒惰一点点到了65岁后基本上就负债累累了，各个亏了非常钱。

因此可以看出做人要勤奋，不能懒惰，否则赚不到什么大钱。要尽可能学习金融投资知识，争取获得更多收入。

