**《时空数据处理与组织课程实习》**

**实验 2 实习报告**

**学 院: 遥感信息工程学院**

**班 级: 20F10**

**学 号: 2020302131249**

**姓 名: 马文卓**

**实习地点: 教学实验大楼101**

**指导教师: 李晓雷**

**2022 年 5 月 24 日**

1. **实验目的**
2. 熟悉编程工具PyCharm。
3. 熟悉Spark的RDD的基本操作。
4. 熟悉实际问题的解决方法。
5. **实验环境**

基于Windows的Spark环境，采用Python编程。

1. **实验内容**

根据给定的啤酒销售数据和去年同期销量数据，把xlsx文件转换为TXT文件，针对11月份啤酒销售数据，并通过编程进行数据处理和计算。

1. 去除整月销量为0的数据。
2. 转换数值格式，把销量数据中的引号、逗号等处理掉，并转换为数值。（基于提供的”实习2啤酒销量数据.tx”数据文件）
3. 有多少类型的啤酒？
4. 哪5种啤酒卖得最好？（销量最高）
5. 在去年销量大于500的区域中，哪个销售区域销售的啤酒同比去年增长最快？（按照增长率计算）
6. 统计每种啤酒的11月份前3周的销量。
7. 统计啤酒卖得最好的前三个区域的11月份前3周销量。
8. **步骤与结果**

在所有数据操作开始之前，指定编码方式为utf-8，初始化本机spark环境，导入所需要的第三方库。使用SparkContext创建其本地文件的对象，使用textFile方法读取txt数据，存储到data中。



1. **数据预处理**

数据预处理的步骤依次如下：

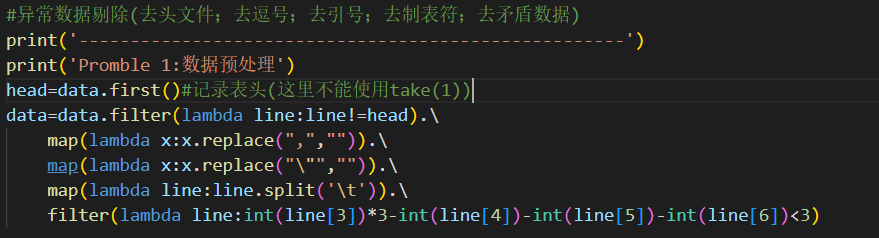
①去除表头：先使用first方法提取表头，然后使用filter方法进行剔除

②去除逗号、引号：使用map方法对每个逗号和引号进行替换，从而去除

③去除制表符：用map方法对每一行数据进行split操作，转换为一个个小元素

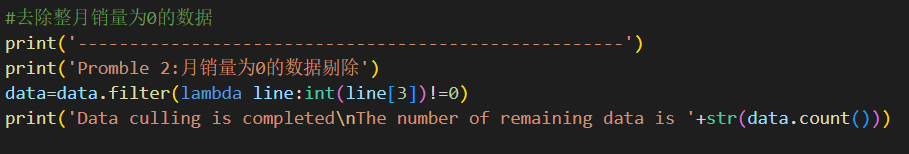
④去除矛盾数据：使用filter方法对每一行数据进行判断，如果三周平均销量\*3-三周销量之和<3则认为数据正确（这里由于避免平均数据四舍五入的影响，从而设置阈值为3）

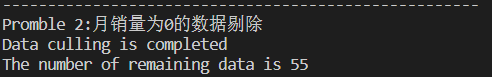
（值得注意的是，这里的第一行head不能够使用take（1）进行获取，经过试验，data.first()!=data.take(1)）



1. **去除月销售量为0的数据**

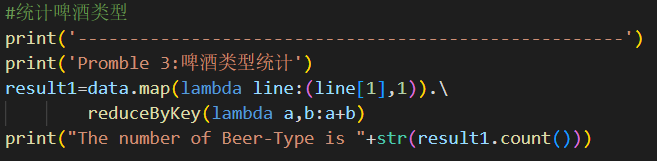
只要三周的平均销量为0，则可判断其月销量为0，使用filter方法进行筛选，得到剩余的数据条数是55.

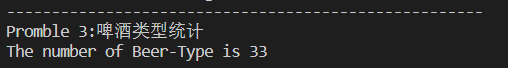




1. **统计啤酒类型**

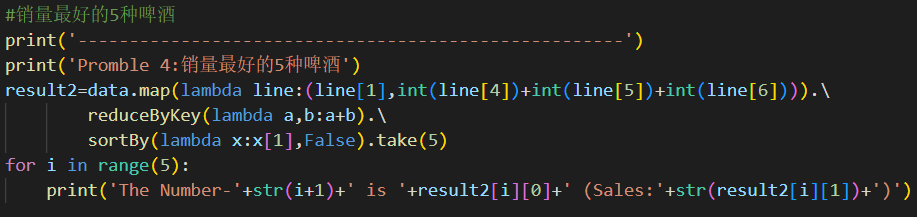
使用map方法生成以啤酒类型为关键字、值均为1的键值对，然后使用reduceByKey进行按关键字融合（值相加），最后使用count方法统计类型数目。

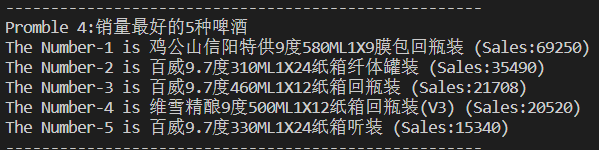




1. **统计销量最高的5种啤酒**

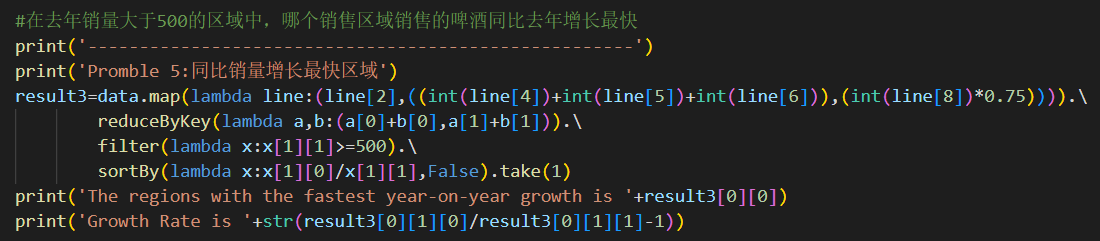
使用map方法生成以啤酒类型为关键字、三周销量之和为值的键值对，然后使用reduceByKey按类型融合（值相加得到该类型的总销量），最后使用sortBy按照总销量（值）降序排列，使用take方法获取前5个数据输出。

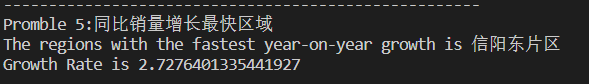




1. **统计同比增长最快的区域**

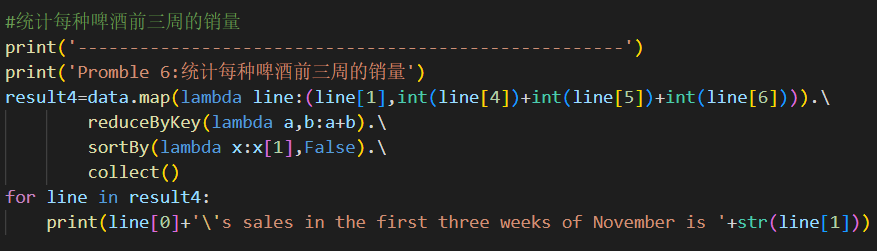
首先使用map方法生成以区域为关键字、三周销量之和和去年销量的0.75形成的元组为值的键值对，然后使用reduceByKey按照区域融合（值对应相加）,然后使用filter筛选出销量大于500的区域数据，最后按照倍数进行降序排列。

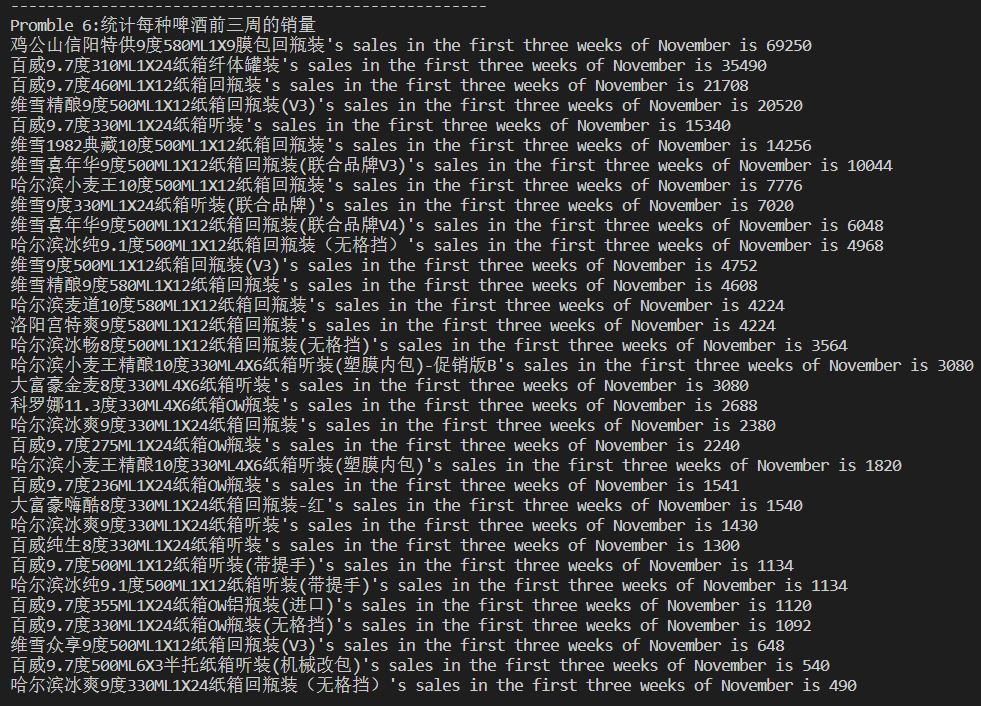




1. **统计每种啤酒的11月份前3周的销量**

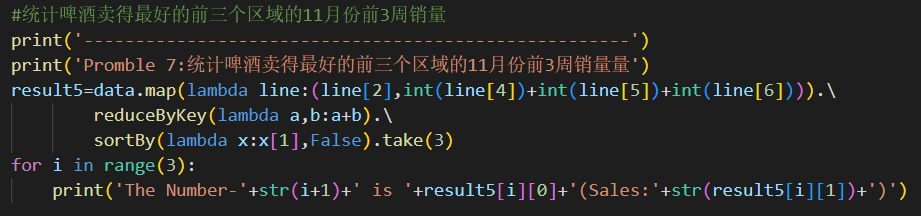
使用map生成以类型为关键字、前三周销量为值的键值对，使用reduceByKey按照关键字进行融合（值相加得到啤酒类型的总销量），使用sortBy进行降序排列，输出结果。

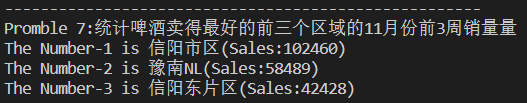




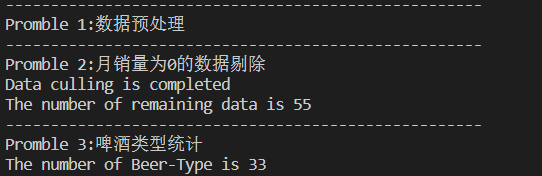
1. **统计啤酒卖得最好的前三个区域的11月份前3周销量**

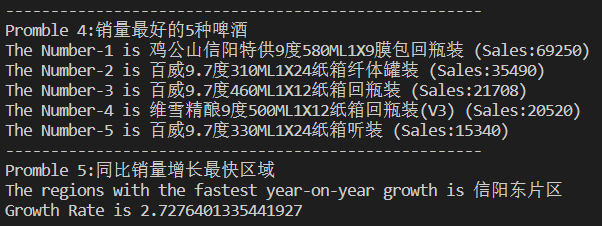
使用map生成以区域为关键字、三周销量之和为值的键值对，然后使用reduceByKey按区域融合（值相加得到各区域的总销量），使用sortBy按照销量进行降序排列，使用take方法取前三进行输出。

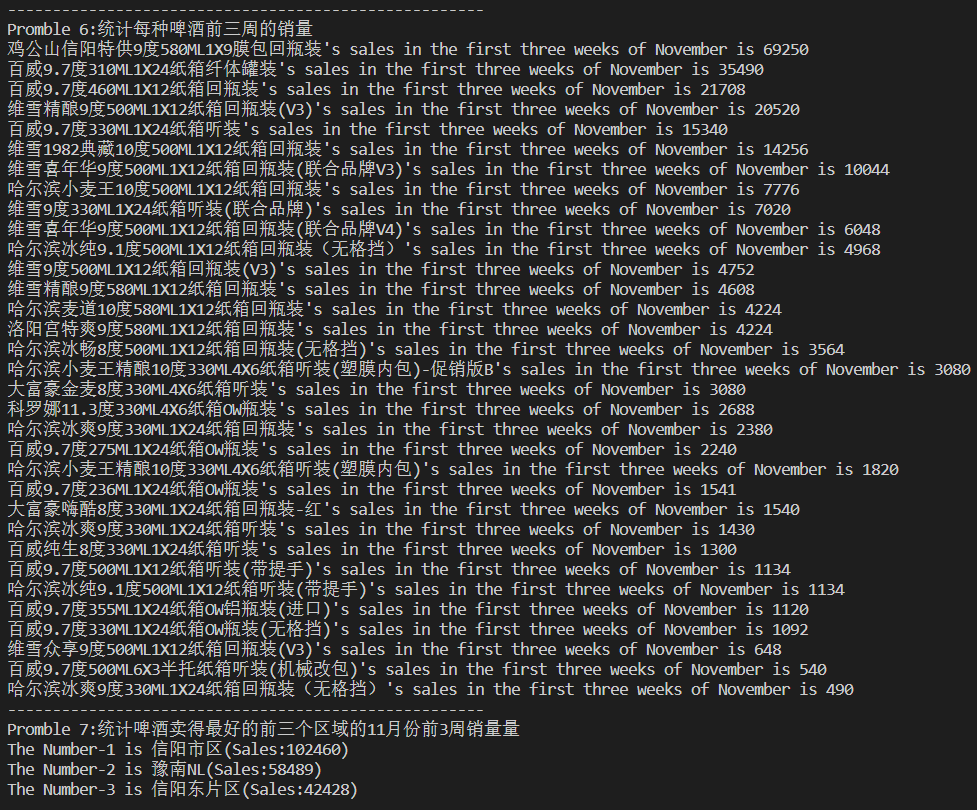




1. **结果汇总**







1. **感悟**

本次实验的最主要收获即为熟悉了spark环境以及RDD对象的操作。

从最开始对于大数据环境的一无所知，到现在能够使用其解决一些实际问题，这无疑是收获巨大的，也同时告诉我们其实大数据没有想象的那么难操作，只要有学习的决心，实际上其和普通的编程没有什么区别。

但是实验的时候依然遇到不少问题：

①RDD对象的数据格式：在对RDD对象不太了解的情况下，有时会把其当当做DataFrame类型进行操作进而报错，实际上解决此类问题的方法我认为是--输出中间结果。事实上无论任何一个新的结构方法你面前，都要试着写一行代码输出一行，这样可以快速帮助你熟悉其距离细节。

②代码的整体美观性：最开始不太注意代码美观，导致很多错误出现，还增加了错误的寻找时间。事实上，代码美观带来的好处就是代码结构清晰、逻辑性强，这样给原本以缩进来控制的python语言纠正错误带来了很好的便利。