**

**HUNAN UNIVERSITY**

《数据仓库技术与应用》

实验报告十二

|  |  |
| --- | --- |
| **报告名称：** | 厦门租房信息分析展示 |
| **学生姓名：** | 杨超然 |
| **学生学号：** | 202106060220 |
| **专业班级：** | 电商2102班 |
| **学 院：** | 工商管理学院 |
| **指导老师：** | 周中定 |
| **日 期：** | 2023.5.15 |

目录

[一、实验环境配置 3](#_Toc26692476)

[二、实验过程 3](#_Toc1943845556)

[（一）建立工程文件fishRent 3](#_Toc543537881)

[（二） 利用python爬取数据 3](#_Toc1993215276)

[（三） pyspark数据分析 6](#_Toc1371734179)

[（四） pyecharts数据可视化 7](#_Toc1499395908)

[（五） 启动程序与结果展示 7](#_Toc1773911858)

[三、易错点总结 10](#_Toc621126105)

[（一）Pycharm无法导入包 10](#_Toc348438668)

[（二） Pyspark无法连接 11](#_Toc20787707)

[四、学习小记 12](#_Toc1486640735)

[（一）实验心得 12](#_Toc2146083947)

[1.1网上的任何东西，都可以看作资源 12](#_Toc97562217)

[1.2爬虫，在当下时代有着无与伦比的价值 12](#_Toc1198158458)

[1.3数据清洗与数据分析，往往比单纯的挖掘与爬取更重要 12](#_Toc495045687)

[1.4数据，在当下已经愈发凸显出“价值化”特点 13](#_Toc881212931)

[（二）知识点笔记 13](#_Toc1498501605)

[2.1Pyecharts 13](#_Toc1775746866)

[2.2BeautifulSoup 14](#_Toc1397334503)

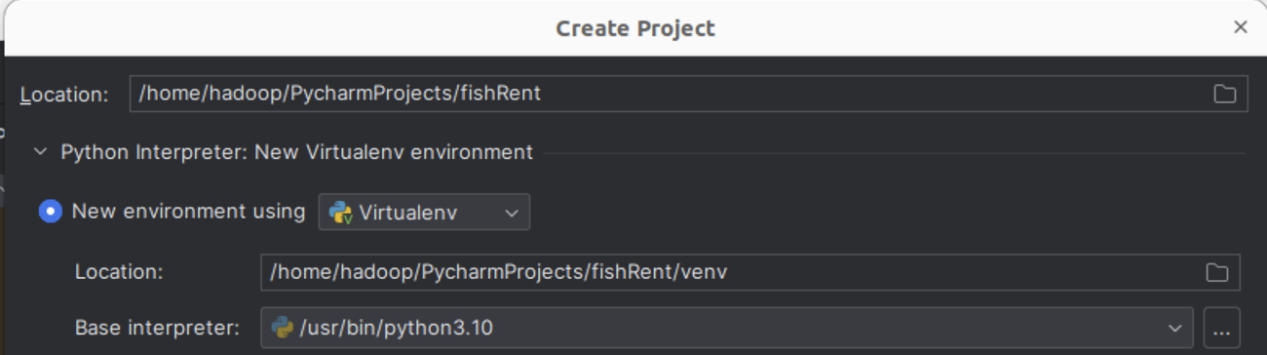
# 一、实验环境配置

本次实验环境：Linux：Ubuntu 22.04；Python：3.6；Pycharm community2023.1.1；

# 二、实验过程

## （一）建立工程文件fishRent

Pycharm-New project-Pure Python:工程名命名fishRent，python版本选择3.6（也可以直接使用jupyter notebook，但因为笔者重装了虚拟机，故直接安装了Pycharm）



## 利用python爬取数据

注：由于厦门小鱼网的网址可能会发生变化，因此指南上爬虫代码的网页不一定有效，建议还是先找到小鱼网房产的网址，进行比对之后再运行代码：



确认好网页地址后，进行代码爬取，该部分利用了BeautifulSoup这个库文件，工程文件命名为rentspider，部分代码展示如下：

·网页定位及数据记录部分：



·数据爬取部分：



运行爬虫函数部分：



在查看网页的元素，比如某一个具体的标签，如我们想爬取的list-word的标签。需要使用soup的方法，把html文件转化为一个dom的树型结构，所以在访问标签的时候可以使用find方法，也可以使用树型的方法，比如tag.em.a像这样。 然后小鱼网的访问有一个地方需要注意，你最多只能找到134页的搜索结果。也就是即使你信息量大于134页，也只能常规地看到134页，所以为了获取更多的数据量。我在实现中针对每一个区的租房信息进行爬取。经过人工测试，每个区的对应数量如下：思明（134），湖里（134），集美（27），同安（41），翔安（76），海沧（6）。爬完总计6000+条数据。



由于代码中有一个总启动入口，在此处就先不运行，但最终运行成功后，会有一个“rent.csv”文件：



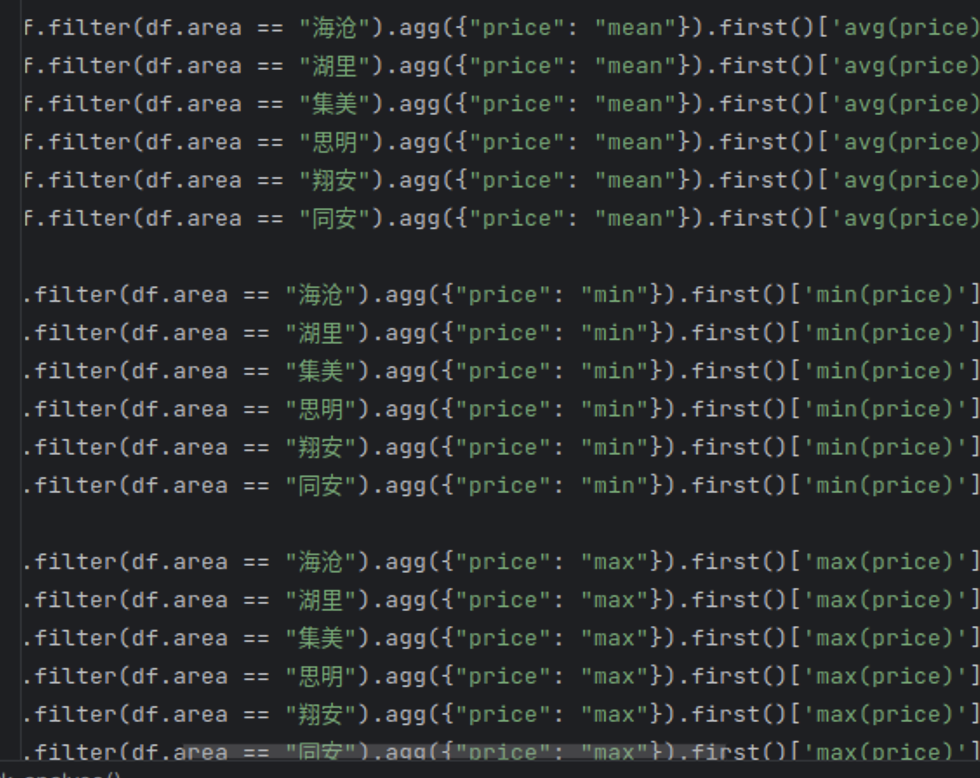
## pyspark数据分析

这一部分是对爬虫得到的“rent.csv”文件进行数据清洗与分析，获得数据中的最大值、最小值、均值、中值等数据，对应的工程文件为rent\_analyse,相关代码的展示如下：

·程序主入口、类型转换与数据存储部分：



·数据分析部分：此部分是一个被调用形式，返回值为一个list



## pyecharts数据可视化

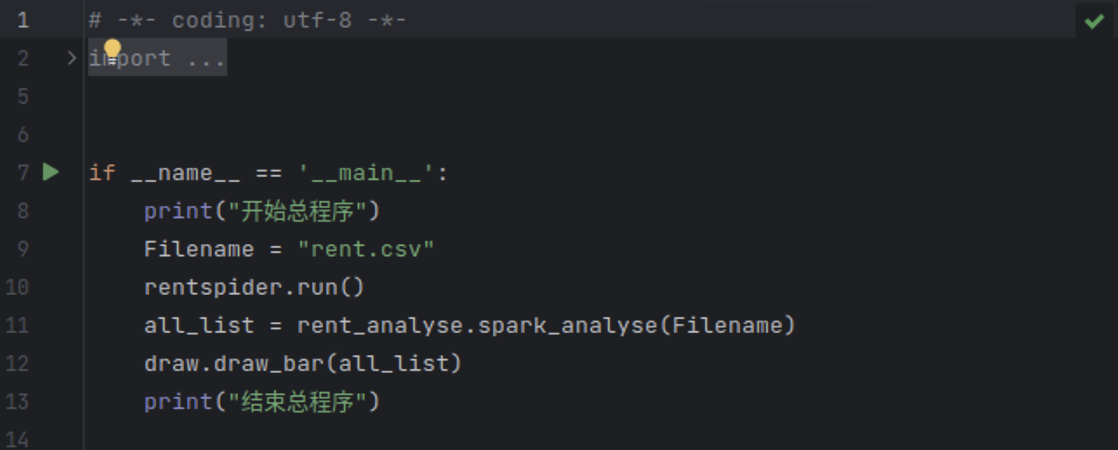
相关代码如下，此部分是进行柱状图（Bar）的展示：该代码通过传入的list读取由pyspark信息



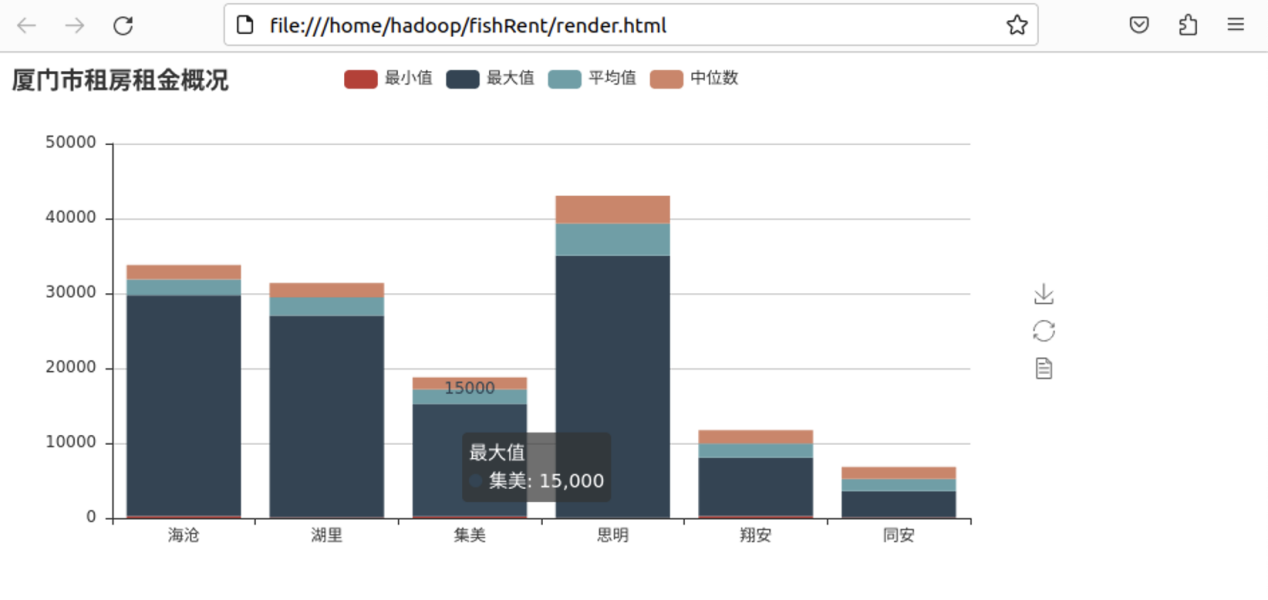
此外，需查看是否缺少snapshot插件，如果没有该插件的话最终启动的时候会报错，因此需要手动导入

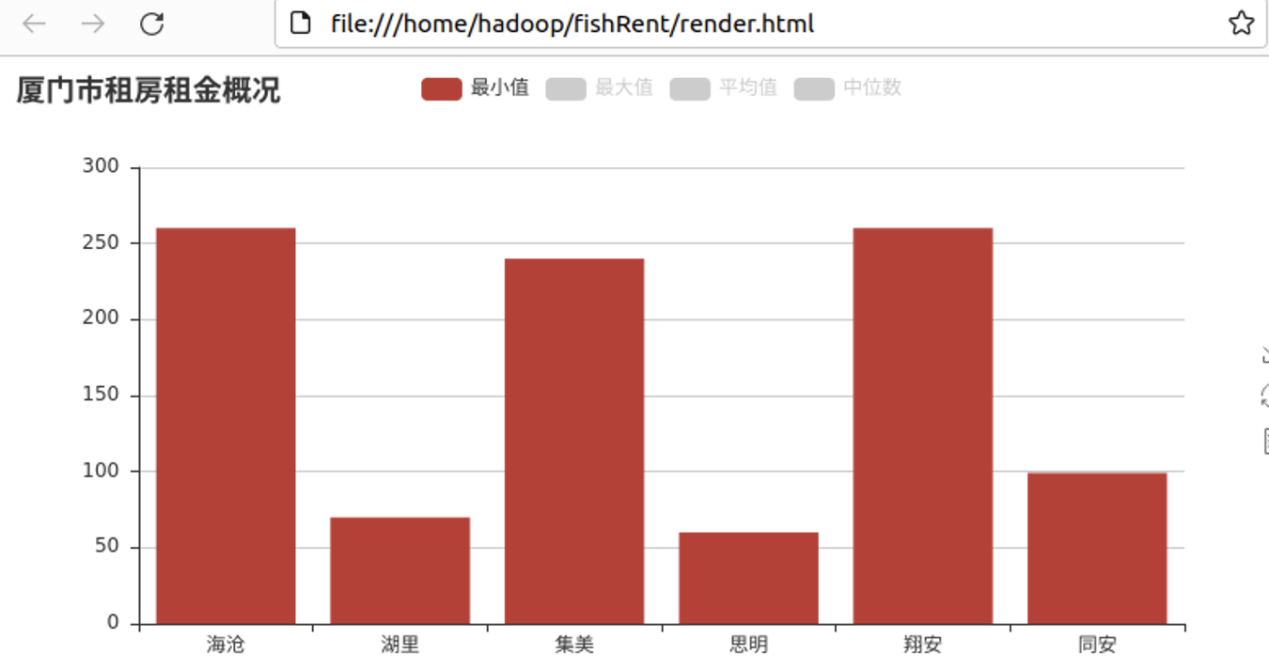
## 启动程序与结果展示

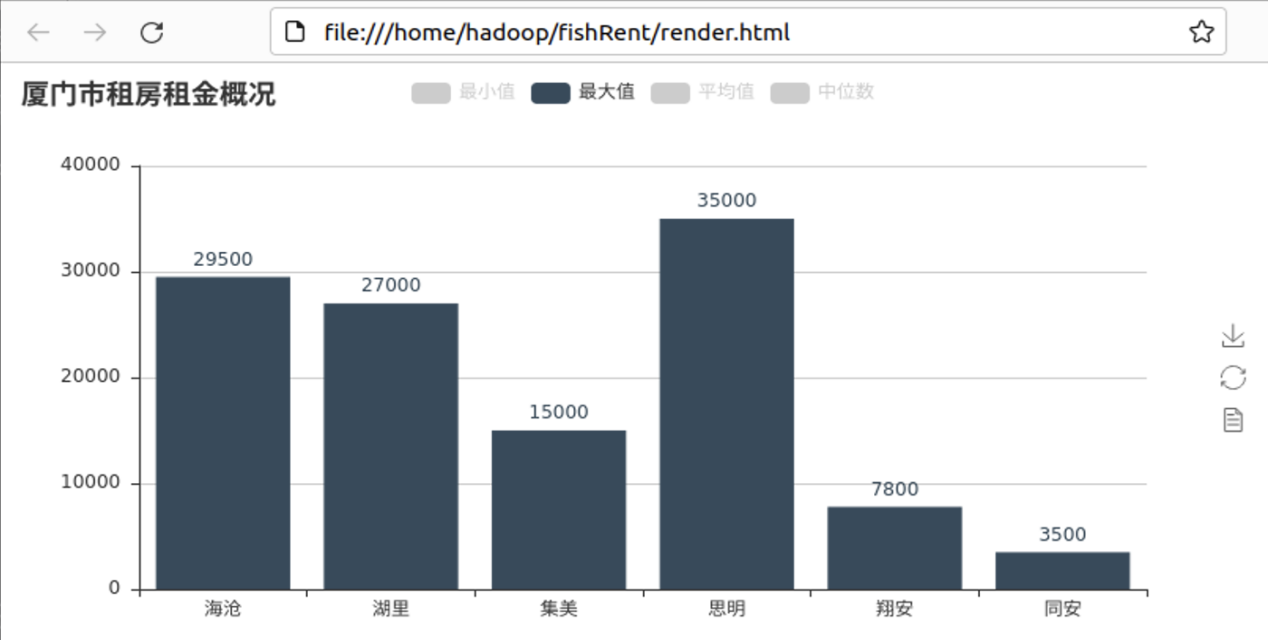
建立工程文件run.py，进行程序的启动：

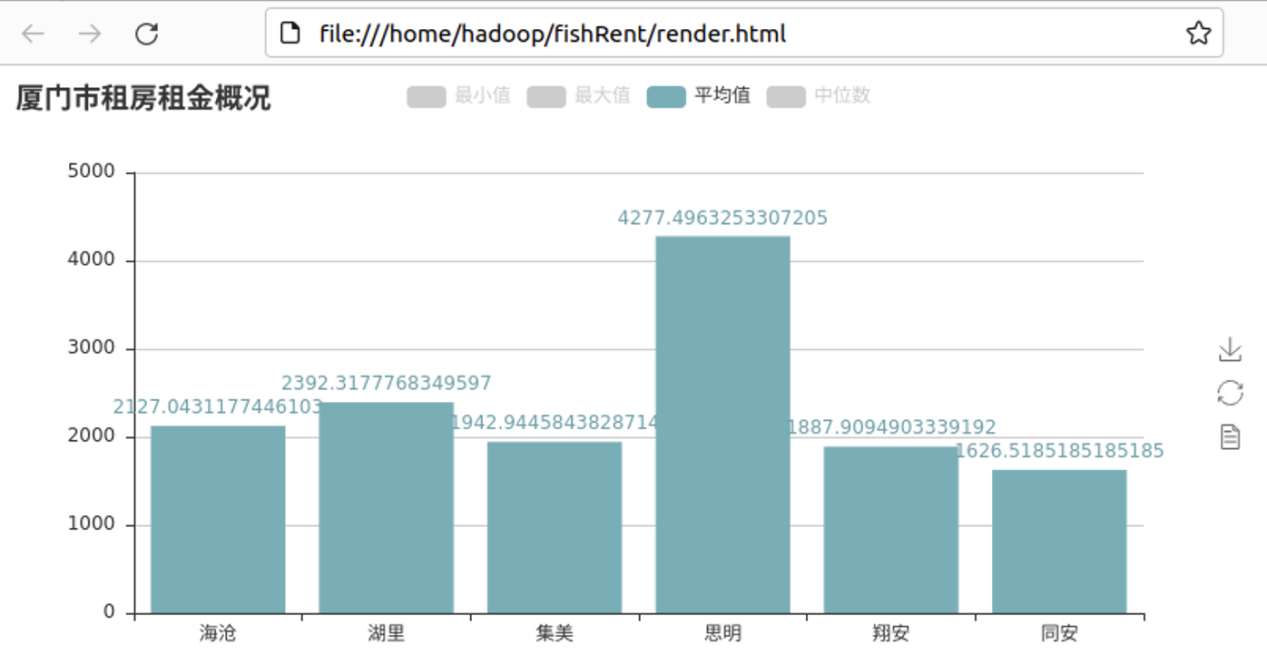


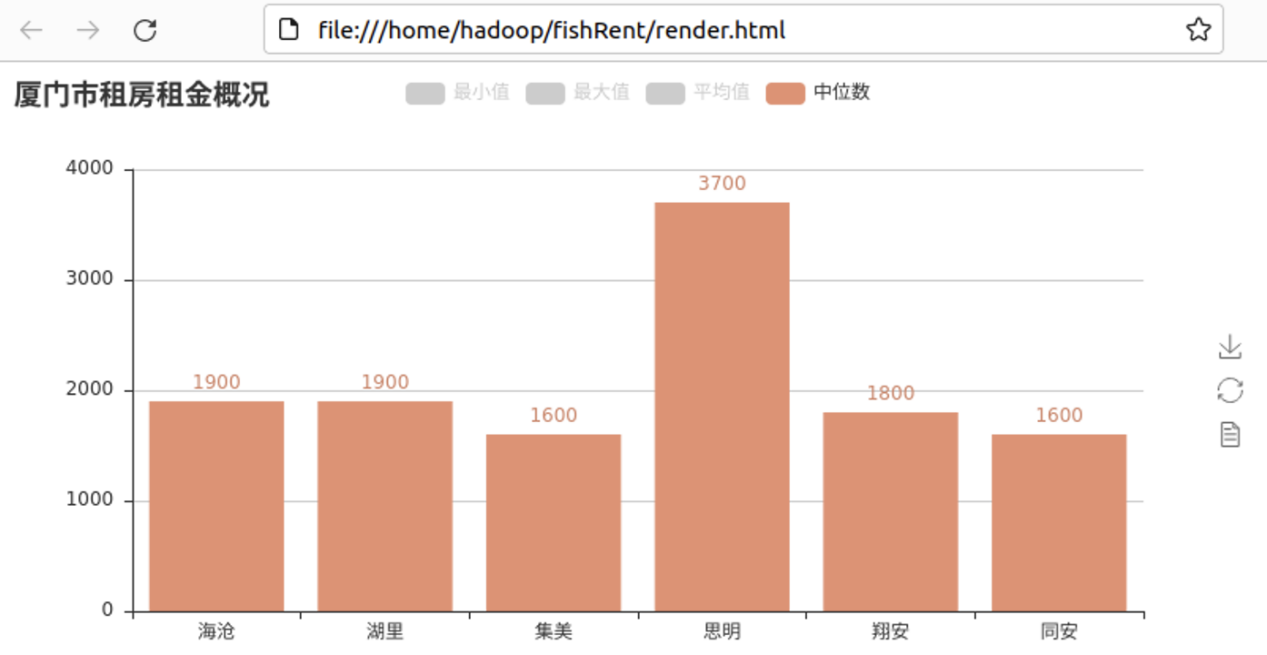
结果展示如下：

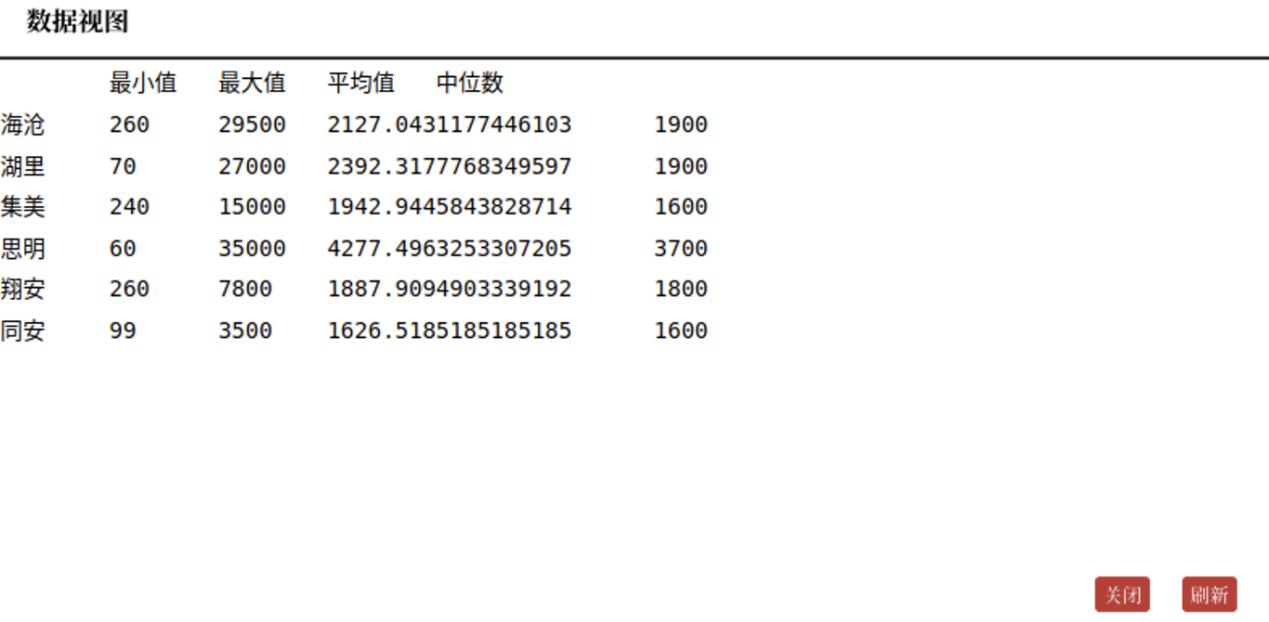








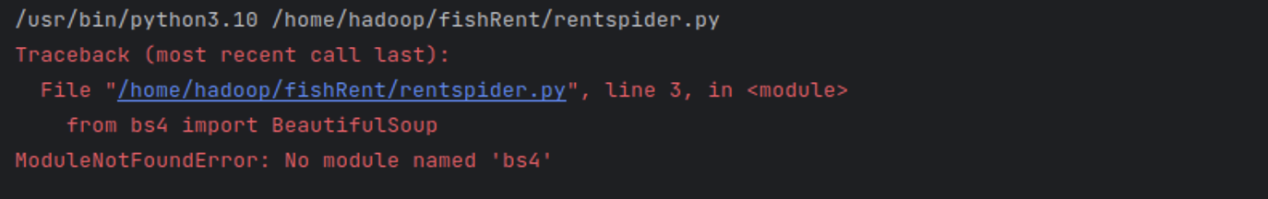




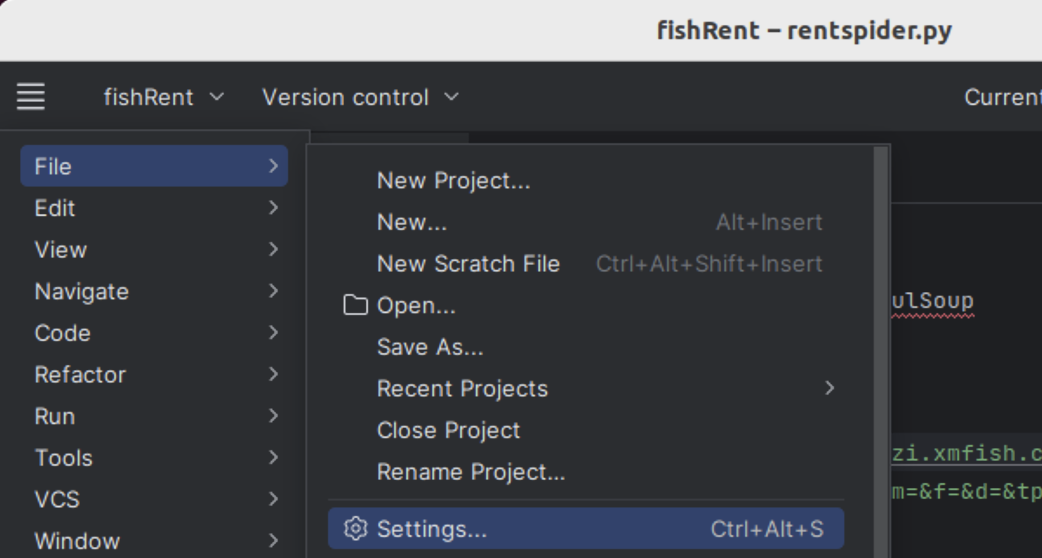
# 三、易错点总结

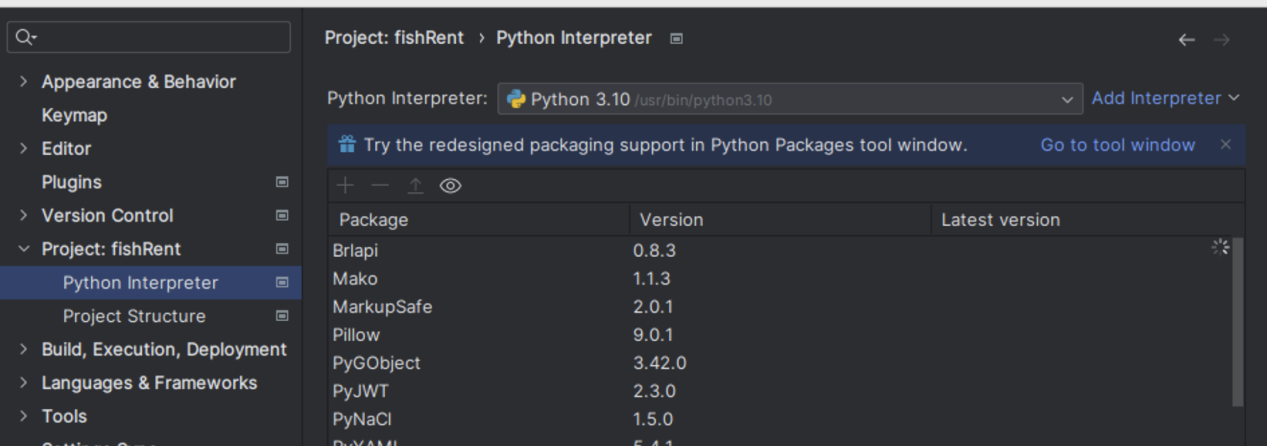
## （一）Pycharm无法导入包

在运行爬虫部分代码的时候，出现了未加载模块的错误



**解决方案：**由于Pycharm自带的包中没有beautifulsoup包，需要手动添加，具体位置为：file-setting-python interprter

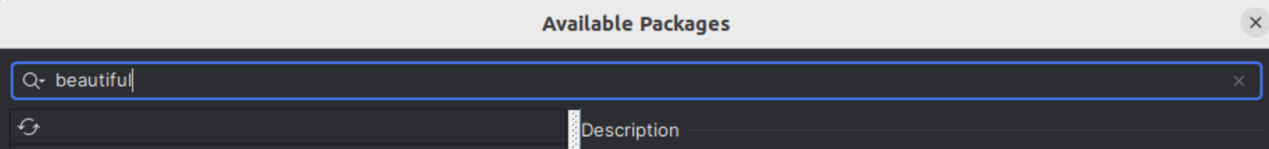




可能会让你安装一个导入包的工具，输入Hadoop的密码进行安装即可

截屏2023-05-15 08.34.26

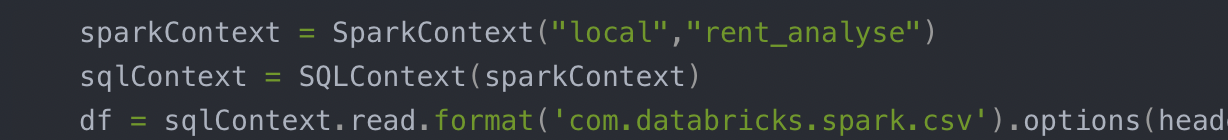
工具安装完毕后，就可以正常导入包了。

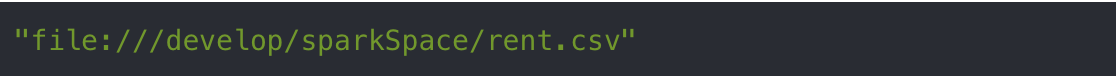


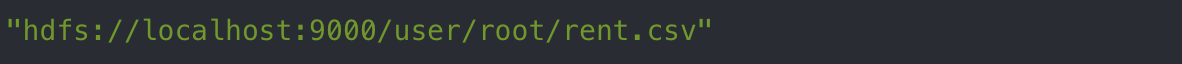
## Pyspark无法连接

在进行pyspark数据分析时，可能为由于csv文件格式读取不正确而报错

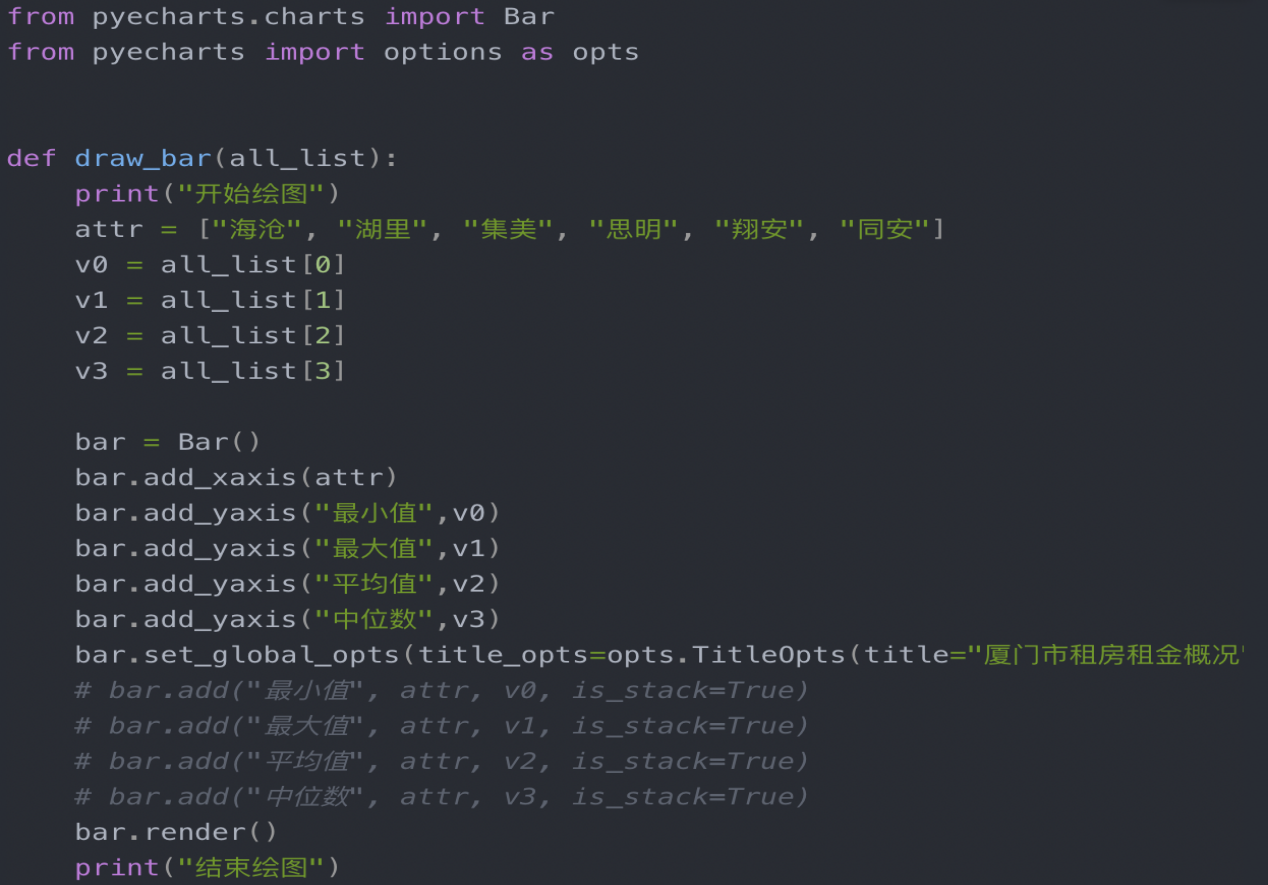
**解决方案：**改变spark读取csv文件的写法，并在读取文件时添加format参数。此外，还需要在文件路径前加上file:///，相关修改如下：







完成修改后的项目文件如下：



# 四、学习小记

## （一）实验心得

在经历了Kafka构建实时分析dashboard和淘宝双十一预测分析这两次令人头大的实验之后，这次的实验就显得轻松了很多，因为无论是从利用python爬取数据、到spark进行数据分析，再到pyecharts进行可视化展示，都是之前实验做过的东西（甚至爬虫我们在大一上学期的python课程设计中就有了初步的了解与实践），因此面对着熟悉的东西，操作起来就显得相对得心应手了，做这次实验甚至两天时间都没有用到，且几乎没有报错，完成地十分流畅，跟之前对比简直不要太过轻松。在这次实验（主要是爬虫）过程中，我得到的相关体会记录如下：

**1.1网上的任何东西，都可以看作资源**

在爬虫的过程中，我深切地体会到——网上可供你使用的资源远不止网页所展示的、可让你交互的界面。这些界面往往都是“想被你发现的”资源，但你能挖掘的远不止如此（当然，合法合规是重要前提）。因为一个网站可能就是一段html+css,一张图片可能就是某个地址下的xxx.jpg，大量放在互联网上的资源，都是可以通过url地址访问或是爬取的，因而可以说，供我们探寻的资源在当下已然越来越多。

**1.2爬虫，在当下时代有着无与伦比的价值**

由于如今网络资源越来越多，对于我们而言已经到了眼花缭乱的地步，想要通过传统手段去一条一条地寻找、总结几乎是天方夜谭。这时爬虫就凸显了自己的价值。其实从本质上来说爬虫就是一段程序代码，任何编程语言都是可以进行爬虫的，只是繁简程度不尽相同罢了。有了爬虫这一获取数据的手段，在当下我们就能更高效、更便捷地获取相关数据，因此只要熟练且合理的使用，爬虫就会在当下时代最大限度地发挥其价值。

**1.3数据清洗与数据分析，往往比单纯的挖掘与爬取更重要**

我们身处在信息爆炸的时代，无论你是否愿意，你每天都会接收到大量的信息，且实际上，许多信息都是对自己毫无意义、甚至是有误导作用的。因此，对数据进行清洗加工、洗掉其中的杂质后进行分析，才能更好地探究事物之间的规律。实践研究层面是如此，在生活当中也是如此。无论在学习过程中还是在人生道路上，对信息进行甄别和筛选，往往都比一昧地挖掘信息更重要。

**1.4数据，在当下已经愈发凸显出“价值化”特点**

其实做过一两次的爬虫，大家就会有所发觉，当下虽然说有很多数据，但是不是所有数据都是能够让你爬取或者获得的，直观点来说，许多网站设置的反爬虫机制就是有意地在防止数据“被获取”。究其根本，是当下数据“价值化”趋势已经越来越明显。从国家提出“数据作为要素充分流动”的概念开始，越来越多的企业就将自己手中的数据视为珍宝，甚至会自己搭建平台存储数据而不是借助第三方平台。因为这都是后续进行分析、交易的重要信息与筹码，又怎能让你轻易获得呢？因此，我们也只能在有限的框架内进行数据的爬取。扯远一点，之后要是我们开公司了，也得好好保管数据，因为这在数字化时代中，都是有价值的财富。

## （二）知识点笔记

**2.1Pyecharts**

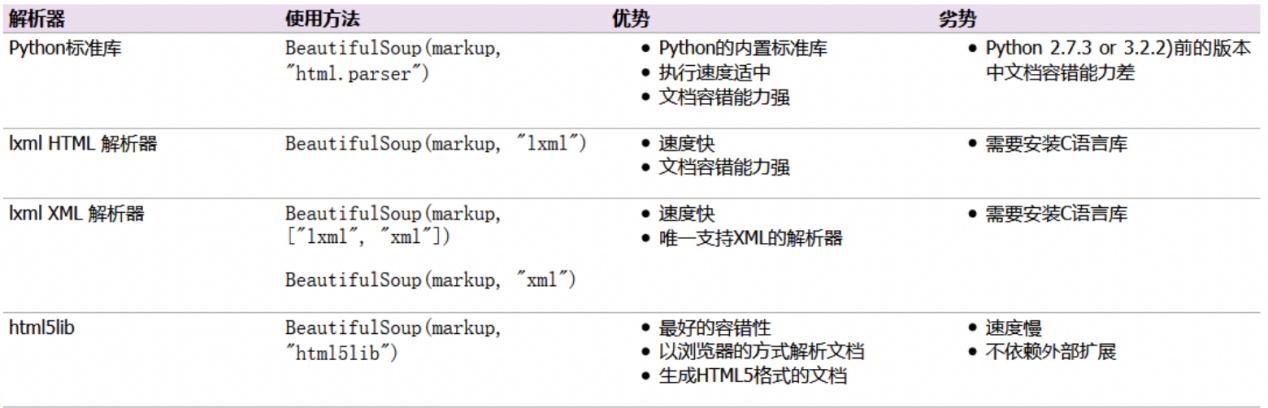
Pyecharts是一款将python与echarts结合的强大的数据可视化工具，使用pyecharts可以生成独立的网页，也可以在flask、Django中集成使用，包含以下图表：



**2.2BeautifulSoup**

BeautifulSoup是一个可以从HTML或XML文件中提取数据的Python库.它能够通过你喜欢的转换器实现惯用的文档导航,查找,修改文档的方式。其作为一个解析器，可以特定的解析出内容，省去了我们编写正则表达式的麻烦。

BeautifulSoup默认支持Python的标准HTML解析库，但是它也支持一些第三方的解析库：



相关的常用操作如下：

