**Python期末项目文档**

**一、程序概述**

本程序是一个多线程爬虫+数据分析+可视化的项目。程序爬取了豆瓣图书网站中计算机类的书籍信息，导出到csv文件中制成了一个书单。同时，对爬取下来的信息做了数据处理分析和可视化。

**二、程序结构与模块介绍**

1. 生成URL：（对应函数gen\_url）由于豆瓣图书的URL非常有规律，所以可以利用这个规律生成一个URL列表。这样通过分页就可以爬取“编程”标签下的所有书籍。

2. 下载HTML：（对应函数download）程序通过requests库，下载URL所对应的HTML代码。

3. 抽取信息：（对应函数extract）程序通过Beautifulsoup4库，将HTML代码中的书籍名称，评论数和评分抽取出来。

4. 多线程：通过创建多个download和extract函数的线程实现。

5. 数据处理：

(1) 根据书籍评分进行排序，并导出到douban.csv中

(2) 统计了书名中带有"C语言", "C++", "Python", "Java", "JavaScript", "Android", "算法", "Linux", "SQL", "C#", "Game"这些关键字的出现次数。（Java类书籍中不包括JavaScript）

(3) 计算出了每类书籍的平均评论数和平均评分

(4) 由于书籍之间分数差距较小，直接从分数的数值大小上不易看出各类书籍之间的评分差距，因此通过归一化计算出了相对评分。

(5) 定义了“人气指数”，即评论数均值 \* 评分均值。通过以此来衡量不同类书籍的受欢迎程度。由于评论数和评分在数值上相距过于悬殊，所以先将评论数均值和评分均值进行了归一化再进行相乘。

(6) 对以上数据，利用matplotlib库绘制了柱状图进行数据可视化分析。

**三、程序运行结果**

-------------------生成子页URL：------------------

100%[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*->]253.94ms

生成URL完成

创建线程完成

线程 <Thread(Thread-1, started 6024)> 开始！

线程 <Thread(Thread-2, stopped 19252)> 开始！

线程 <Thread(Thread-3, stopped 1064)> 开始！

线程 <Thread(Thread-4, stopped 23724)> 开始！

线程 <Thread(Thread-5, stopped 15580)> 开始！

线程 <Thread(Thread-6, started 1844)> 开始！

线程 <Thread(Thread-7, stopped 4064)> 开始！

爬取完成！用时共： 155.86200461967204 s.

各类书籍统计总书： [26. 47. 69. 60. 33. 17. 33. 28. 18. 12. 18.]

评论数均值： [234.46153846 359.5106383 265.85507246 270.26666667 335.6969697

147.58823529 399.93939394 248.64285714 257.05555556 120.91666667

122. ]

评分均值： [8.40769231 8.36170213 8.03188406 8.05833333 8.33939394 7.79411765

8.24242424 8.33214286 8.21111111 8.55833333 8.17777778]

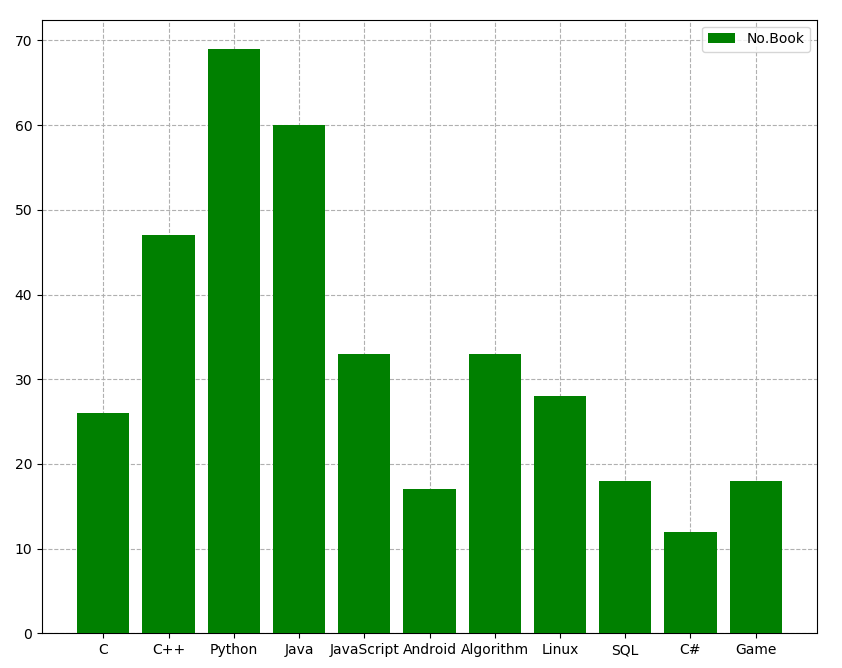
相对评分： [0.80288153 0.74270195 0.31112475 0.34573445 0.71351099 0.

0.58662313 0.70402273 0.54564892 1. 0.50203122]

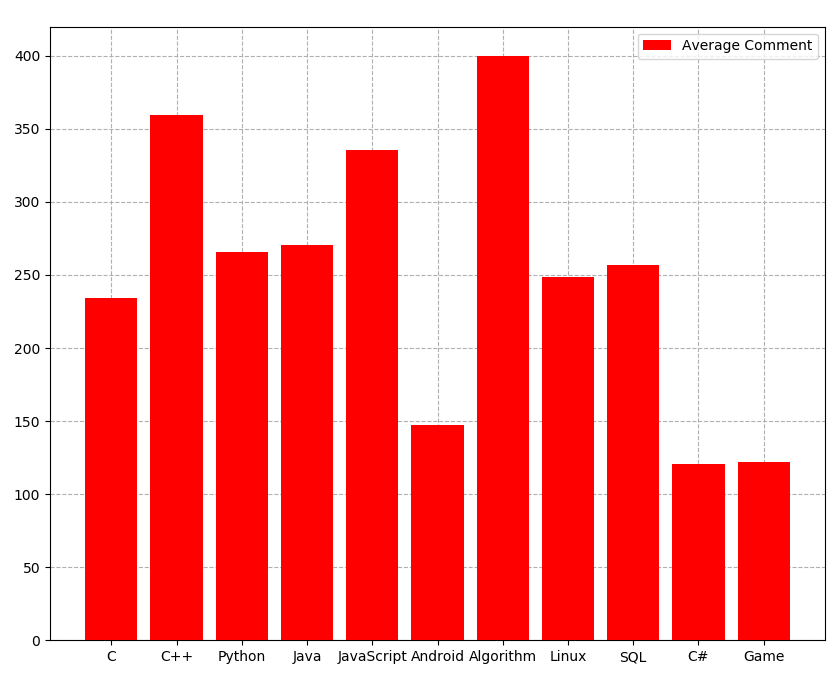
人气指数： [0.51445212 1. 0.25447434 0.29138947 0.86481077 0.

0.92368678 0.50744924 0.41920019 0. 0.00306915]

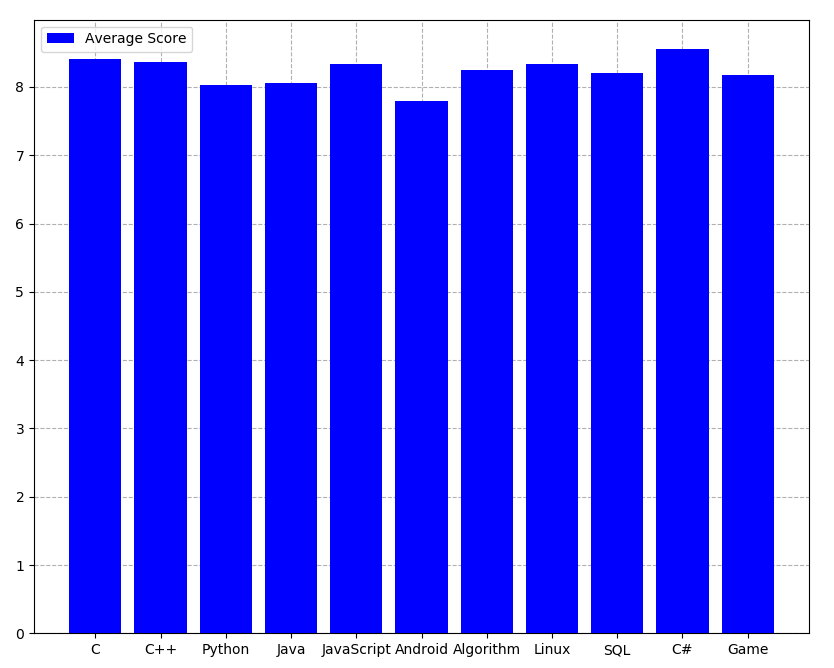
图一：书单中各类书籍的数量



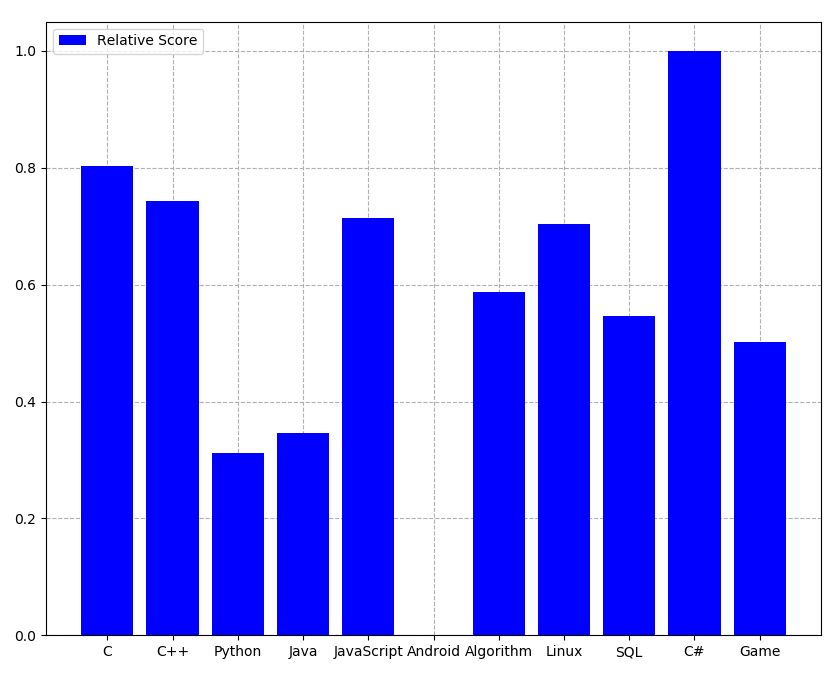
图二：平均评论数



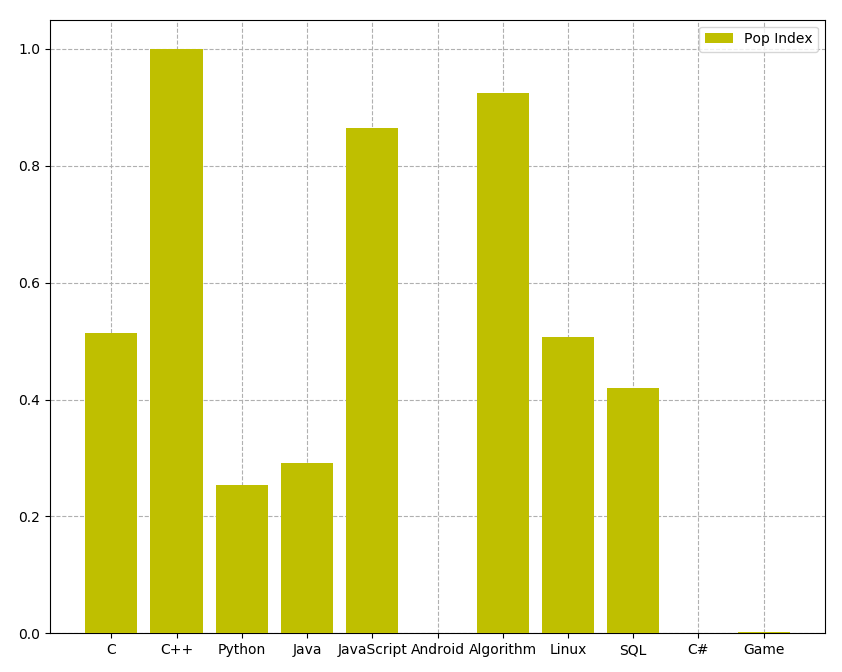
图三：平均评分



图四：相对评分



图五：人气指数



**四、有趣的结论**

通过以上的绘图分析，我们可以得出以下结论。

1. 在豆瓣编程目录下的书籍中，书籍数量的前三分别是Python, Java和C++, 最少的是C#. 安卓、游戏和SQL类书籍数目也相对较少。
2. 平均评论数的前三分别是算法，C++，JavaScript类书籍。C#, 游戏，安卓类书籍平均评论数量最少。书籍数量较少的SQL类书籍平均评论数却不低。
3. 平均评分各类书籍之间相差不大，在相对评分中出现了令人惊讶的结论：C#类书籍的得分最高，C语言、C++类书籍其次，而安卓的最低。
4. 人气书籍前三为C++，算法，JavaScript. C#相对评分虽高，却因为评论数过少而人气最低。此外，安卓和游戏类的人气也相对非常低。
5. 虽然书单中名字带有Python和Java的书籍最多，但这两类书籍的其他指标并不高。这也许侧面反映出了当前Python和Java类书籍多而杂的现状。

**五、不足之处**

程序的不足之处还是有的，不足和待改进的地方主要有以下方面：

1. 可利用数据库对已爬取的网页进行缓存。
2. 该爬虫只能爬取静态的HTML网页，限制较多。
3. 数据分析方面，书籍的名字并不能完全与该书籍所属的类别相对应。
4. 由于只是对书名进行关键字检索，所以在统计C语言类书籍的时候统计不完全。