|  |  |
| --- | --- |
| 学号 | 10212818114 |



课 程 设 计

课程名称 python程序设计课程设计

|  |  |
| --- | --- |
| 题 目 | 数据分析的学习与实践-《绝地求生》 |
| 专 业 | 软件工程 |
| 班 级 | 软件1181 |
| 姓 名 | 张玮 |
| 成 绩 |  |
| 指 导 老 师 | \_\_\_\_\_\_\_\_\_\_\_徐勇\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

2021 年 1 月 11 日至 2021 年 1 月 22 日

武汉华夏理工学院信息工程学院

**课 程 设 计 任 务 书**

课程名称：python程序设计课程设计 指导教师：徐勇

班级名称： 软件1181-1182 开课院、系：计算机与网络工程系

**一、课程设计目的与任务**

“python程序设计课程设计”是一个综合性的学习实践型实验教学环节，将在“python程序设计”课程的授课基础上，对python的基础语法、python 对文件的操作、python 对数据的操作、数据处理基础、数据可视化等若干个知识点进行综合运用。

python是一种具有天然开源基因的编程语言，了解开源社区和广泛的使用开源工具，也是Python学习实践的重要环节。因此在本次课程设计中，要求掌握jupyter notebook、Git等常用工具，以及github等重要开源社区的使用。

**二、课程设计的内容与基本要求**

数据分析的基本技术和方法在“python程序设计”课程中已经进行完整的讲授，由于数据分析本身是基于业务场景的，因此本次课程设计环节更加偏重于实际的业务场景的实践。通过对近期互联网热点的调查，准备了15个具体的业务场景，用于本次课程设计的具体任务场景。具体包括如下：

1. 北上广深租房状况分析；
2. 蔡某坤粉丝数及转发数据真假状况分析；
3. 地震的数据分析；
4. 英文名字的数据分析；
5. 外籍英文老师收入虚高情况数据分析；
6. 我国城市空气污染和烟花燃放的关系分析；
7. 针对996工作，程序员群体的看法的分析；
8. 吴某凡微博热点的分析；
9. 节假日长假景点人满为患的数据分析；
10. 针对荔枝的品种、销售地等维度，进行价格数据分析；
11. 分析芒果TV《我是大侦探》的观众评论数据；
12. 针对当前儿科医生的缺乏，对相关数据进行分析；
13. 著名网游《绝地求生》的数据分析；
14. 实习岗位状况的数据分析；
15. 电影《流浪地球》的观众评价的数据分析。

每三个学生组建一个课程设计小组，最后的任务输出包括代码每小组一份、课程设计报告每人一份、答辩ppt每小组一份，并进行课程设计成果答辩。小组成员均参与前述工作，但是每个人的侧重点不同。

每个课程设计小组可以从上述15个场景中选取一个作为课程设计的选题，选题中提供了待分析的数据，和现有的分析方法。各小组，通过学习和实践现有的分析方法，理解实战分析的思维过程并锻炼实际动手能力，再此基础上可以扩展更多维度的分析和数据展现形式。

每个班每个选题最多只能被两个小组选中，先选先得。同时，如果各小组发现更有意思的场景，并能够获取到相关待分析的数据，也可以申请作为选题方向。

本次课程设计的目标是培养学生的团队协作能力、对python知识点的综合运用、对实际场景的理解和适应能力、针对答辩的表达能力等。注重过程，期待成果，但不强求结果的尽善尽美。

**三、学时分配进度安排**

|  |  |  |
| --- | --- | --- |
| 序号 | 设计内容 | 所用时间 |
| 1 | 下发任务书，学生查阅资料 | 1天 |
| 2 | 组建团队，并进行选题和团队匹配 | 1天 |
| 3 | 各小组对自己的选题开始进行研究 | 2天 |
| 4 | 各小组开始准备数据并形成初步处理意见 | 1天 |
| 5 | 代码和实现分析的动手实践 | 2天 |
| 6 | 形成初步的报告书和ppt | 1天 |
| 7 | 答辩并完成报告书 | 2天 |
| 合 计 | | 2周 |

**四、课程设计考核及评分标准**

**1.设计报告要求**

课程设计报告要求逻辑清晰、层次分明、书写整洁。课程设计报告为每人一份，同一个小组的各成员的整体报告内容结构一致，但个人侧重点不同，个人着重撰写自己的工作内容，其他人的内容只要体现文档结构的完整性即可。

课程设计考核将综合考虑学生考勤和参与度、团队协作能力，过程管理能力、成果达成情况等。

**2.过程要求**

整个过程要求通过github来进行过程化发布，即阶段性地在github上提交过程结果。

**3.评分标准**

|  |  |
| --- | --- |
| **评分依据** | **评分成绩** |
| 1．团队协作能力 | 25分 |
| 2．python综合运用能力 | 25分 |
| 3．态度认真、刻苦钻研、创新能力 | 10分 |
| 4．过程完成、对工具的使用、对github的运用 | 20分 |
| 5．课程设计答辩逻辑清晰，内容正确 | 10分 |
| 6. 课程设计期间的课堂考勤、遵守纪律 | 10分 |
| 总分 | 100分 |

注：按上述六项分别记分后求和，根据小组成员贡献率综合评定，记载个人最后成绩。

成绩等级：优（90分—100分）、良（80分—89分）、中（70分—79分）、及格（60分—69分）、60分以下为不及格。

**五、指导地点与时间**

本课程设计将安排在第20-21周，采用腾讯会议和QQ群的方式，以在线形式进行。具体安排如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 周次 | 星期一 | 星期二 | 星期三 | 星期四 | 星期五 |
| 第20周 | 第5-8节 | 第5-8节 |  | 第5-8节 |  |
| 第21周 |  | 第5-8节 |  | 第5-8节 |  |

执笔:徐 勇 日期：2021-1-8

审阅:钱小红 日期：2021-1-8

目录

[前言 6](#_Toc61701281)

[1编程环境与相关技术 7](#_Toc61701282)

[1.1硬件平台 7](#_Toc61701283)

[1.2项目托管平台-GitHub 8](#_Toc61701284)

[1.3开发平台-jupyter 8](#_Toc61701285)

[1.4编程语言-python 4](#_Toc61701286)

[2 Github详细介绍 1](#_Toc61701310)1

[2.1 Github简介 11](#_Toc61701312)

[2.2 Github注册与登录 1](#_Toc61701313)1

3 数据分析

[3.1数据分析库 1](#_Toc61701288)2

[3.1.1数据分析库导入见下图3.1-1 1](#_Toc61701289)3

[3.1.2数据分析库注解 1](#_Toc61701290)3

[3.2数据表 1](#_Toc61701291)4

[3.2.1数据表导入见下图3.2-1 1](#_Toc61701292)4

[3.2.2数据表说明 1](#_Toc61701293)4

[3.3数据分析图 15](#_Toc61701296)

[3.3.1数据提取代码见下图3.3-1 1](#_Toc61701297)5

[3.3.2生成数据分析图代码见下图3.3-2 1](#_Toc61701298)8

4 总结...............................................................................................................19

## 前言

《绝地求生》游戏简介

这座岛名字叫做Erangel，在游戏中，Erangel是位于克里米亚的一个小岛，在二战之后被苏联占领，占领之后，苏联人把这座岛从所有的地图中抹去，并将它打造成一个秘密研究基地，还在当地人身上进行生物和化学实验。随后这座岛被遗弃，因为一群反抗军炸掉了几个生物研究所，导致所有人都撤离了这座岛。

每一局游戏有100名玩家参与，玩家空投跳伞至地图的各个角落，赤手空拳寻找武器，车辆以及物资，并在多种多样的地形中展开战斗。想要取得胜利，策略与枪法同样重要。在游戏过程中，会有一个蓝色的圈驱逐玩家到一个较小的地方交火。这个蓝圈俗称为 "毒圈" ，然而这并不是毒气，开发者说这是由某种电力发射器释放的某种电场，本来是用于镇压Erangel当地人的反抗行动的。游戏的每一局比赛都会随机转换安全区，并且每个区域获得的武器、道具均是随机出现。

枪械

近战武器(如镰刀、平底锅等)、2把主武器(步枪、轻机枪、霰弹枪、冲锋枪、射手步枪、狙击枪)，1把副武器（手枪）

地图

游戏地图分为绝地岛、米拉苏、热带雨林三个，并且包含雨天，晴天，雾天等各种天气情况

载具

游戏里的载具有摩托车、吉普车、小轿车、沙漠越野车、快艇等，不同的载具可搭载不同的人数、不同的时速及不同的优点。

人物

游戏角色会有100点生命值和100点背包容量。玩家可进入到这些地图，搜集各种物资来维持生命值，或扩展自身的背包容量。角色身上可携带的物品数量，不能超出背包容量。

# 1编程环境与相关技术

## 1.1硬件平台

CPU：Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz ~2.70GHz

最低要求：64位7代i3处理器及以上

内存：4G

最低要求：4G

硬盘空间：SSD-120G

最低要求：机械硬盘-20G

## 1.2项目托管平台-GitHub

Github是一个基于git的代码托管平台，付费用户可以建私人仓库，我们一般的免费用户只能使用公共仓库，也就是代码要公开。

Github 由Chris Wanstrath, PJ Hyett 与Tom Preston-Werner三位开发者在2008年4月创办。迄今拥有59名全职员工，主要提供基于git的版本托管服务。

GitHub 于 2008 年 4 月 10 日正式上线，除了 [Git](https://baike.baidu.com/item/Git/12647237" \t "https://baike.baidu.com/item/github/_blank) 代码仓库托管及基本的 Web 管理界面以外，还提供了订阅、讨论组、文本渲染、在线文件编辑器、协作图谱（报表）、代码片段分享（Gist）等功能。目前，其注册用户已经超过 350 万，托管版本数量也是非常之多，其中不乏知名开源项目 [Ruby](https://baike.baidu.com/item/Ruby/11419" \t "https://baike.baidu.com/item/github/_blank) on Rails、[jQuery](https://baike.baidu.com/item/jQuery/5385065" \t "https://baike.baidu.com/item/github/_blank)、[python](https://baike.baidu.com/item/python/407313" \t "https://baike.baidu.com/item/github/_blank) 等。

## 1.3开发平台-jupyter

Jupyter Notebook（此前被称为 IPython notebook）是一个交互式笔记本，支持运行 40 多种编程语言。

Jupyter Notebook 的本质是一个 Web[应用程序](https://baike.baidu.com/item/%20%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F/5985445" \t "https://baike.baidu.com/item/Jupyter/_blank)，便于创建和共享文学化程序文档，支持实时代码，数学方程，可视化和 [markdown](https://baike.baidu.com/item/markdown/3245829" \t "https://baike.baidu.com/item/Jupyter/_blank)。 用途包括：数据清理和转换，数值模拟，统计建模，机器学习等等

简单来说Jupyter Notebooks 是一款开源的网络应用，我们可以将其用于创建和共享代码与文档。

其提供了一个环境，你无需离开这个环境，就可以在其中编写你的代码、运行代码、查看输出、可视化数据并查看结果。因此，这是一款可执行端到端的数据科学工作流程的便捷工具，其中包括数据清理、统计建模、构建和训练机器学习模型、可视化数据等等。

当你还处于原型开发阶段时，Jupyter Notebooks 的作用更是引人注目。这是因为你的代码是按独立单元的形式编写的，而且这些单元是独立执行的。这让用户可以测试一个项目中的特定代码块，而无需从项目开始处执行代码。很多其它 IDE 环境（比如 RStudio）也有其它几种方式能做到这一点，但我个人觉得 Jupyter 的单个单元结构是最好的。它们甚至允许你运行 Python 之外的其它语言，比如 R、SQL 等。因为它们比单纯的 IDE 平台更具交互性，所以它们被广泛用于以更具教学性的方式展示代码。

## 1.4编程语言-python

Python由荷兰数学和计算机科学研究学会的Guido van Rossum 于1990 年代初设计，作为一门叫做[ABC语言](https://baike.baidu.com/item/ABC%E8%AF%AD%E8%A8%80/334996" \t "https://baike.baidu.com/item/Python/_blank)的替代品。Python提供了高效的高级[数据结构](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E7%BB%93%E6%9E%84/1450" \t "https://baike.baidu.com/item/Python/_blank)，还能简单有效地[面向对象](https://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1/2262089" \t "https://baike.baidu.com/item/Python/_blank)编程。Python语法和动态类型，以及[解释型语言](https://baike.baidu.com/item/%E8%A7%A3%E9%87%8A%E5%9E%8B%E8%AF%AD%E8%A8%80/8888952" \t "https://baike.baidu.com/item/Python/_blank)的本质，使它成为多数平台上写脚本和快速开发应用的编程语言，随着版本的不断更新和语言新功能的添加，逐渐被用于独立的、大型项目的开发。

Python[解释器](https://baike.baidu.com/item/%E8%A7%A3%E9%87%8A%E5%99%A8/10418965" \t "https://baike.baidu.com/item/Python/_blank)易于扩展，可以使用C或[C++](https://baike.baidu.com/item/C++/99272" \t "https://baike.baidu.com/item/Python/_blank)（或者其他可以通过C调用的语言）扩展新的功能和[数据类型](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E7%B1%BB%E5%9E%8B/10997964" \t "https://baike.baidu.com/item/Python/_blank)。Python 也可用于可定制化软件中的扩展程序语言。Python丰富的标准库，提供了适用于各个主要系统平台的源码或[机器码](https://baike.baidu.com/item/%E6%9C%BA%E5%99%A8%E7%A0%81/86125" \t "https://baike.baidu.com/item/Python/_blank)。

Python的创始人为荷兰人吉多·范罗苏姆 Guido van Rossum）。1989年圣诞节期间，在[阿姆斯特丹](https://baike.baidu.com/item/%E9%98%BF%E5%A7%86%E6%96%AF%E7%89%B9%E4%B8%B9/2259975" \t "https://baike.baidu.com/item/Python/_blank)，Guido为了打发[圣诞节](https://baike.baidu.com/item/%E5%9C%A3%E8%AF%9E%E8%8A%82/127881" \t "https://baike.baidu.com/item/Python/_blank)的无趣，决心开发一个新的脚本解释程序，作为ABC语言的一种继承。之所以选中Python（大蟒蛇的意思）作为该编程语言的名字，是取自英国20世纪70年代首播的电视喜剧《蒙提·派森的飞行马戏团》（Monty Python's Flying Circus）。

ABC是由Guido参加设计的一种[教学](https://baike.baidu.com/item/%E6%95%99%E5%AD%A6" \t "https://baike.baidu.com/item/Python/_blank)语言。就Guido本人看来，ABC这种语言非常优美和强大，是专门为非专业程序员设计的。但是ABC语言并没有成功，究其原因，Guido认为是其非开放造成的。Guido决心在Python中避免这一错误。同时，他还想实现在ABC中闪现过但未曾实现的东西。

就这样，Python在Guido手中诞生了。可以说，Python是从ABC发展起来，主要受到了[Modula-3](https://baike.baidu.com/item/Modula-3/17009923" \t "https://baike.baidu.com/item/Python/_blank)（另一种相当优美且强大的语言，为小型团体所设计的）的影响。并且结合了[Unix shell](https://baike.baidu.com/item/Unix shell" \t "https://baike.baidu.com/item/Python/_blank)和[C](https://baike.baidu.com/item/C/7252092" \t "https://baike.baidu.com/item/Python/_blank)的习惯。

Python已经成为最受欢迎的[程序设计语言](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1%E8%AF%AD%E8%A8%80/2317999" \t "https://baike.baidu.com/item/Python/_blank)之一。自从2004年以后，python的使用率呈线性增长。Python 2于2000年10月16日发布，稳定版本是Python 2.7。Python 3于2008年12月3日发布，不完全兼容Python 2。2011年1月，它被[TIOBE](https://baike.baidu.com/item/TIOBE" \t "https://baike.baidu.com/item/Python/_blank)编程语言[排行榜](https://baike.baidu.com/item/%E6%8E%92%E8%A1%8C%E6%A6%9C/4895" \t "https://baike.baidu.com/item/Python/_blank)评为2010年度语言。

由于Python语言的[简洁](https://baike.baidu.com/item/%E7%AE%80%E6%B4%81" \t "https://baike.baidu.com/item/Python/_blank)性、易读性以及可扩展性，在国外用Python做科学计算的研究机构日益增多，一些知名大学已经采用Python来教授程序设计[课程](https://baike.baidu.com/item/%E8%AF%BE%E7%A8%8B" \t "https://baike.baidu.com/item/Python/_blank)。例如[卡耐基梅隆大学](https://baike.baidu.com/item/%E5%8D%A1%E8%80%90%E5%9F%BA%E6%A2%85%E9%9A%86%E5%A4%A7%E5%AD%A6" \t "https://baike.baidu.com/item/Python/_blank)的编程基础、[麻省理工学院](https://baike.baidu.com/item/%E9%BA%BB%E7%9C%81%E7%90%86%E5%B7%A5%E5%AD%A6%E9%99%A2/117999" \t "https://baike.baidu.com/item/Python/_blank)的计算机科学及编程导论就使用Python语言讲授。众多开源的科学计算软件包都提供了Python的调用[接口](https://baike.baidu.com/item/%E6%8E%A5%E5%8F%A3" \t "https://baike.baidu.com/item/Python/_blank)，例如著名的计算机视觉库[OpenCV](https://baike.baidu.com/item/OpenCV" \t "https://baike.baidu.com/item/Python/_blank)、三维可视化库VTK、医学图像处理库ITK。而Python专用的科学计算扩展库就更多了，例如如下3个十分经典的科学计算扩展库：NumPy、SciPy和matplotlib，它们分别为Python提供了快速数组处理、数值运算以及绘图功能。因此Python语言及其众多的扩展库所构成的开发环境十分适合[工程](https://baike.baidu.com/item/%E5%B7%A5%E7%A8%8B" \t "https://baike.baidu.com/item/Python/_blank)技术、科研人员处理实验数据、制作图表，甚至开发科学计算[应用程序](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F" \t "https://baike.baidu.com/item/Python/_blank)。2018年3月，该语言作者在邮件列表上宣布Python 2.7将于2020年1月1日终止支持。用户如果想要在这个日期之后继续得到与Python 2.7有关的支持，则需要付费给商业供应商。

Python在设计上坚持了清晰划一的风格，这使得Python成为一门易读、易维护，并且被大量用户所欢迎的、用途广泛的[语言](https://baike.baidu.com/item/%E8%AF%AD%E8%A8%80/2291095" \t "https://baike.baidu.com/item/Python/_blank)。

设计者开发时总的指导思想是，对于一个特定的问题，只要有一种最好的方法来解决就好了。这在由Tim Peters写的Python格言（称为The Zen of Python）里面表述为：There should be one-- and preferably only one --obvious way to do it. 这正好和[Perl](https://baike.baidu.com/item/Perl" \t "https://baike.baidu.com/item/Python/_blank)语言（另一种功能类似的高级[动态语言](https://baike.baidu.com/item/%E5%8A%A8%E6%80%81%E8%AF%AD%E8%A8%80" \t "https://baike.baidu.com/item/Python/_blank)）的中心思想TMTOWTDI（There's More Than One Way To Do It）完全相反。

Python的作者有意的设计限制性很强的语法，使得不好的编程习惯（例如[if语句](https://baike.baidu.com/item/if%E8%AF%AD%E5%8F%A5" \t "https://baike.baidu.com/item/Python/_blank)的下一行不向右缩进）都不能通过编译。其中很重要的一项就是Python的[缩进](https://baike.baidu.com/item/%E7%BC%A9%E8%BF%9B/7337492" \t "https://baike.baidu.com/item/Python/_blank)规则。

一个和其他大多数语言（如C）的区别就是，一个模块的界限，完全是由每行的首字符在这一行的位置来决定的（而C语言是用一对花括号[{}](https://baike.baidu.com/item/{}" \t "https://baike.baidu.com/item/Python/_blank)来明确的定出模块的边界的，与字符的位置毫无关系）。这一点曾经引起过争议。因为自从C这类的语言诞生后，语言的语法含义与字符的排列方式分离开来，曾经被认为是一种程序语言的进步。不过不可否认的是，通过强制程序员们[缩进](https://baike.baidu.com/item/%E7%BC%A9%E8%BF%9B" \t "https://baike.baidu.com/item/Python/_blank)（包括if，for和函数定义等所有需要使用模块的地方），Python确实使得程序更加清晰和美观。

Python在执行时，首先会将.py文件中的[源代码](https://baike.baidu.com/item/%E6%BA%90%E4%BB%A3%E7%A0%81/3969" \t "https://baike.baidu.com/item/Python/_blank)编译成Python的byte code（字节码），然后再由Python Virtual Machine（Python[虚拟机](https://baike.baidu.com/item/%E8%99%9A%E6%8B%9F%E6%9C%BA" \t "https://baike.baidu.com/item/Python/_blank)）来执行这些编译好的byte code。这种机制的基本思想跟Java，.NET是一致的。然而，Python Virtual Machine与Java或.NET的Virtual Machine不同的是，Python的Virtual Machine是一种更高级的Virtual Machine。这里的高级并不是通常意义上的高级，不是说Python的Virtual Machine比Java或.NET的功能更强大，而是说和Java 或.NET相比，Python的Virtual Machine距离真实机器的距离更远。或者可以这么说，Python的Virtual Machine是一种抽象层次更高的Virtual Machine。基于C的Python编译出的[字节码](https://baike.baidu.com/item/%E5%AD%97%E8%8A%82%E7%A0%81" \t "https://baike.baidu.com/item/Python/_blank)文件，通常是.pyc格式。除此之外，Python还可以以交互模式运行，比如主流操作系统Unix/[Linux](https://baike.baidu.com/item/Linux/27050" \t "https://baike.baidu.com/item/Python/_blank)、Mac、Windows都可以直接在命令模式下直接运行Python交互环境。直接下达操作指令即可实现交互操作。

# 2 Github详细介绍

## 2.1 Github简介

Github作为开源代码库以及版本控制系统，Github拥有超过900万开发者用户。随着越来越多的应用程序转移到了云上，Github已经成为了管理软件开发以及发现已有代码的首选方法。

如前所述，作为一个分布式的版本控制系统，在Git中并不存在主库这样的概念，每一份[复制](https://baike.baidu.com/item/%E5%A4%8D%E5%88%B6" \t "_blank)出的库都可以独立使用，任何两个库之间的不一致之处都可以进行合并。

GitHub可以托管各种git库，并提供一个web界面，但它与外国的[SourceForge](https://baike.baidu.com/item/SourceForge/6562141" \t "_blank)、[Google Code](https://baike.baidu.com/item/Google%20Code" \t "_blank)或中国的[coding](https://baike.baidu.com/item/coding/8921246" \t "_blank)的服务不同，GitHub的独特卖点在于从另外一个项目进行分支的简易性。为一个项目贡献代码非常简单：首先点击项目站点的“fork”的按钮，然后将代码检出并将修改加入到刚才分出的代码库中，最后通过内建的“pull request”机制向项目负责人申请代码合并。已经有人将GitHub称为代码玩家的MySpace。

**2.2 仓库的创建**

**如图2.3-1,2.3-2所示：**

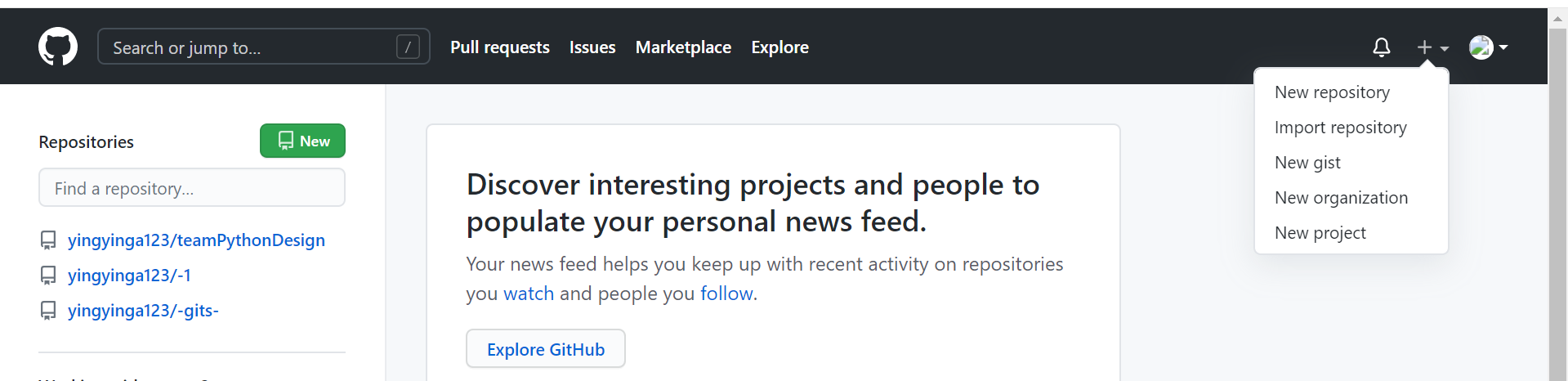
****

图2.3-1

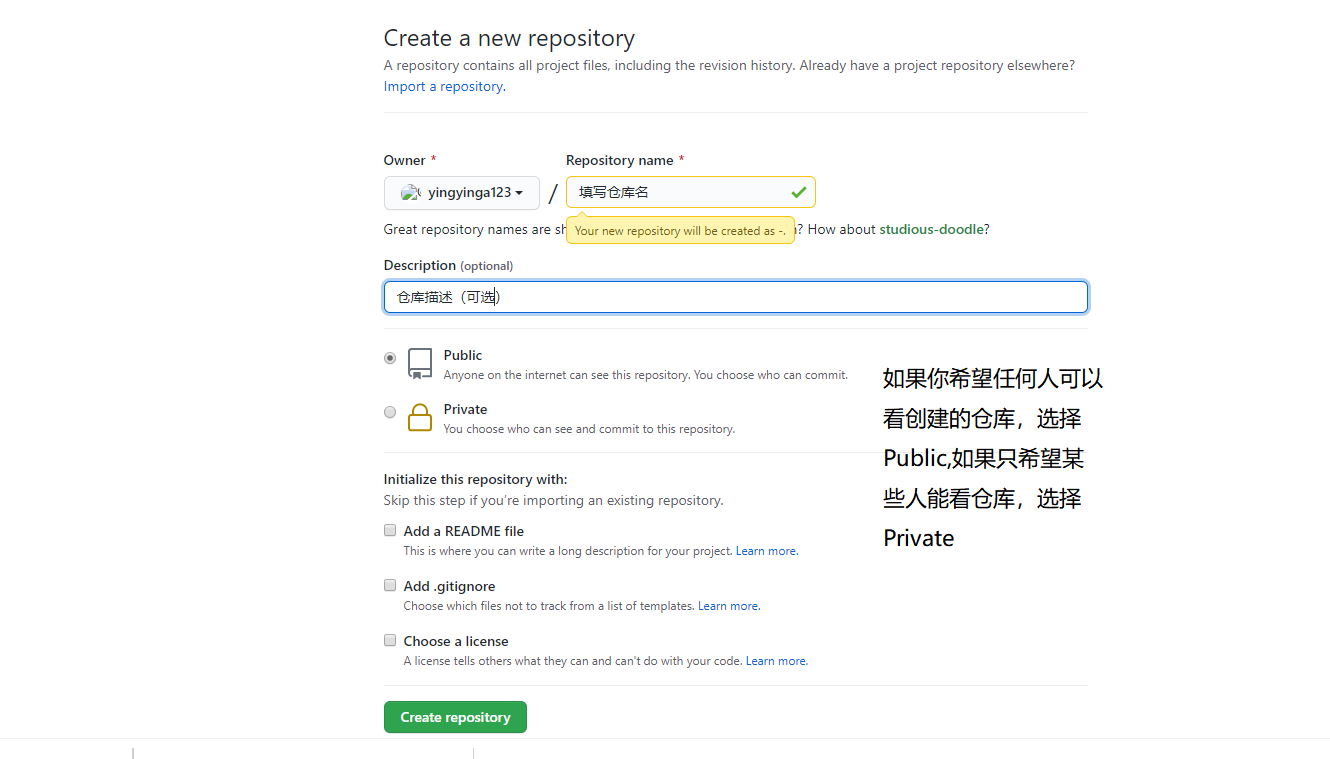


图2.3-2

# 3、程序设计

## 3.1数据分析库

### 3.1.1数据分析库导入见下图3.1-1

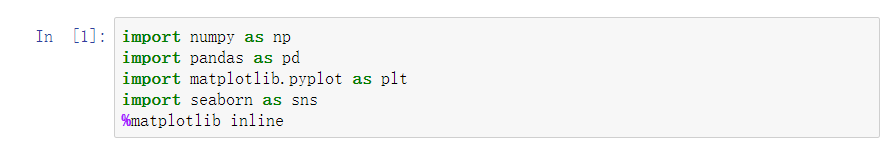


图3.1-1

### 3.1.2数据分析库注解

pandas 是 python 的数据分析处理库 NumPy是Python语言的一个扩充程序库。支持高级大量的维度数组与bai矩阵运算，此外也针对数组运算提供大量的数学函数库。

Numpy内部解除了CPython的GIL（全局解释器锁），运行效率极好，是大量机器学习框架的基础库！

使用%matplotlib命令可以将matplotlib的图表直接嵌入到Notebook之中，或者使用指定的界面库显示图表，它有一个参数指定matplotlib图表的显示方式。inline表示将图表嵌入到Notebook中。

%matplotlib inline Seaborn是一种基于matplotlib的图形可视化python libraty。它提供了一种高度交互式界面，便于用户能够做出各种有吸引力的统计图表。

Seaborn其实是在matplotlib的基础上进行了更高级的API封装，从而使得作图更加容易，在大多数情况下使用seaborn就能做出很具有吸引力的图，而使matplotlib就能制作具有更多特色的图。

Matplotlib是一个Python的2D绘图库，它以各种硬拷贝格式和跨平台的交互式环境生成出版质量级别的图形。通过Matplotlib，开发者可以仅需要几行代码，便可以生成绘图。一般可绘制折线图、散点图、柱状图、饼图、直方图、子图等等。Matplot使用Numpy进行数组运算，并调用一系列其他的Python库来实现硬件交互。

## 3.2数据表

### 3.2.1数据表导入见下图3.2-1

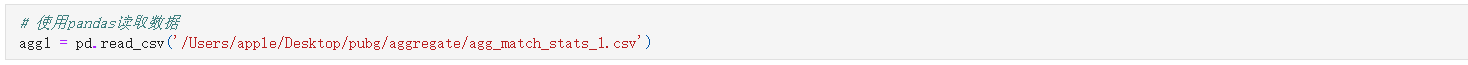
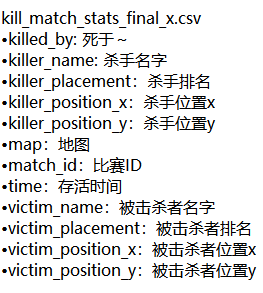


图3.2-1

### 3.2.2数据表说明

数据来源：老师提供的20G绝地求生比赛数据，主要分成两部分，一部分是玩家比赛的统计数据，以agg\_match\_stats开头，一部分是玩家被击杀的数据，以kill\_match\_stats开头

数据字段注释：

## 3.3数据分析图

### 3.3.1数据提取代码见下图3.3-1

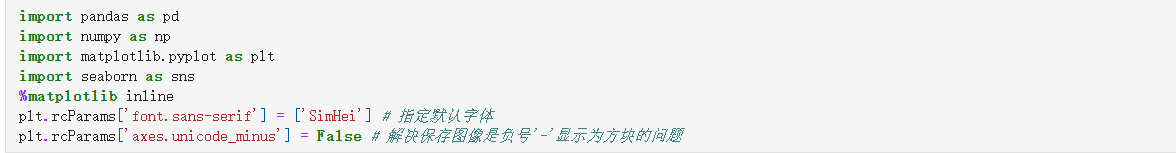




图3.3-1

### 3.3.2生成数据分析图代码见下图3.3-2——3.3-7



图3.3-2

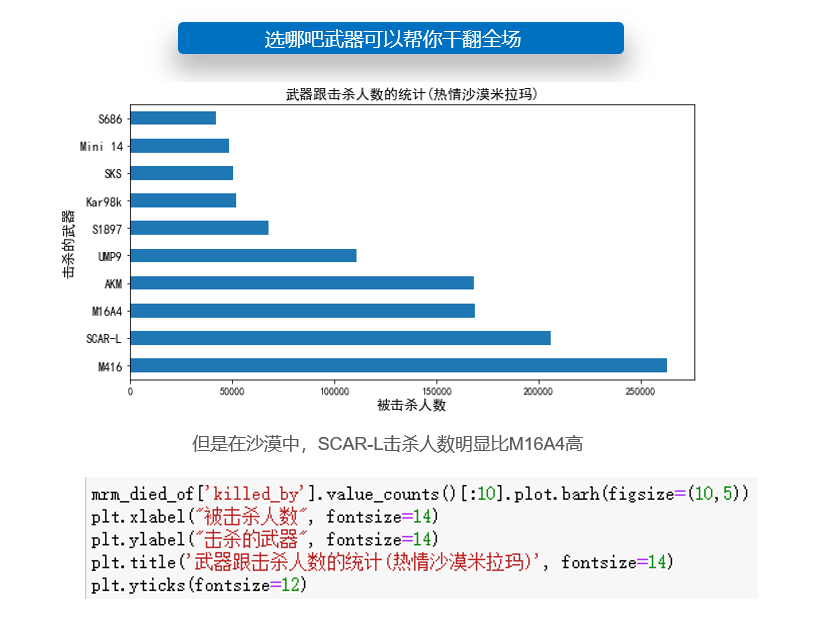


图3.3-3

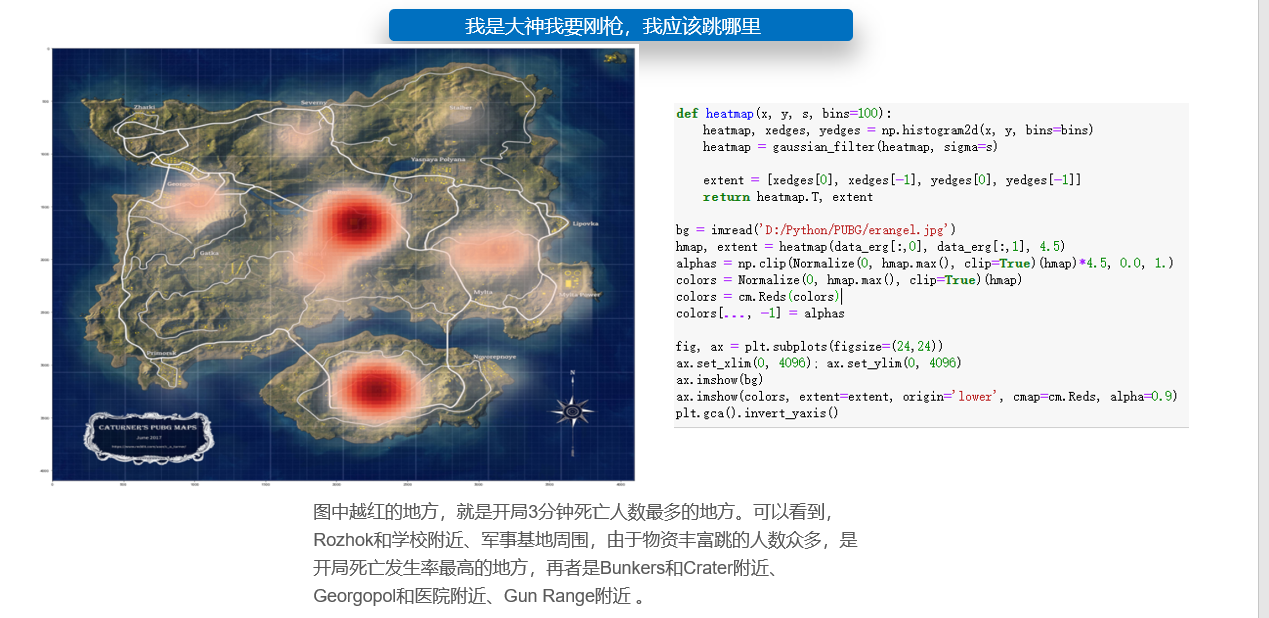


图3.3-4

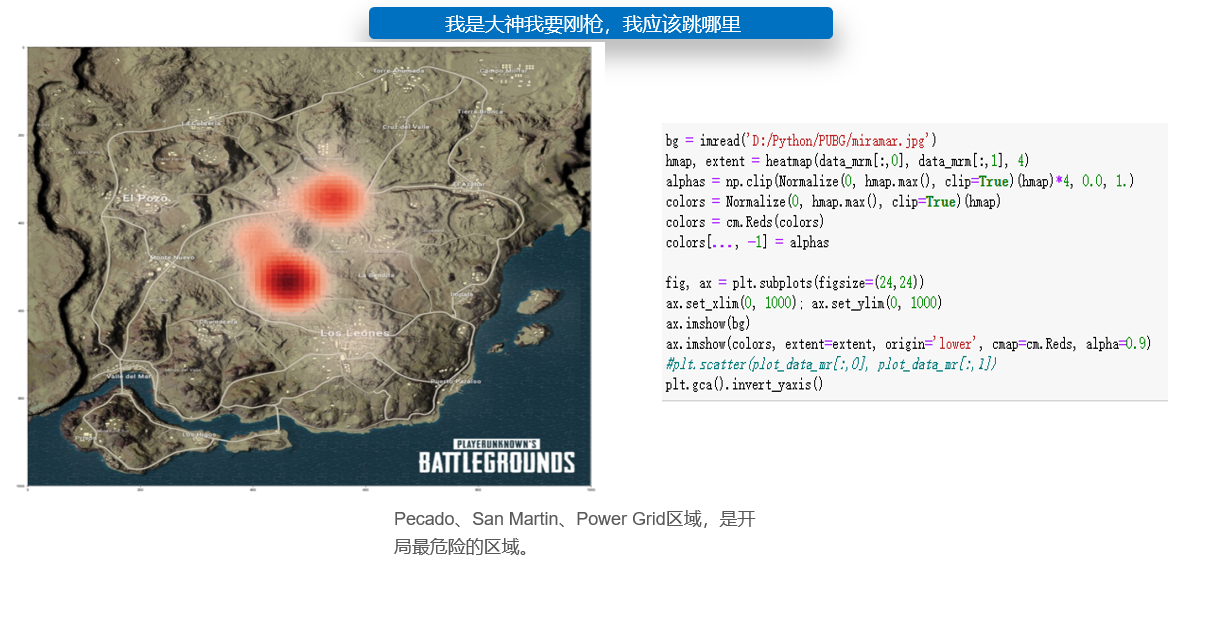


图3.3-5

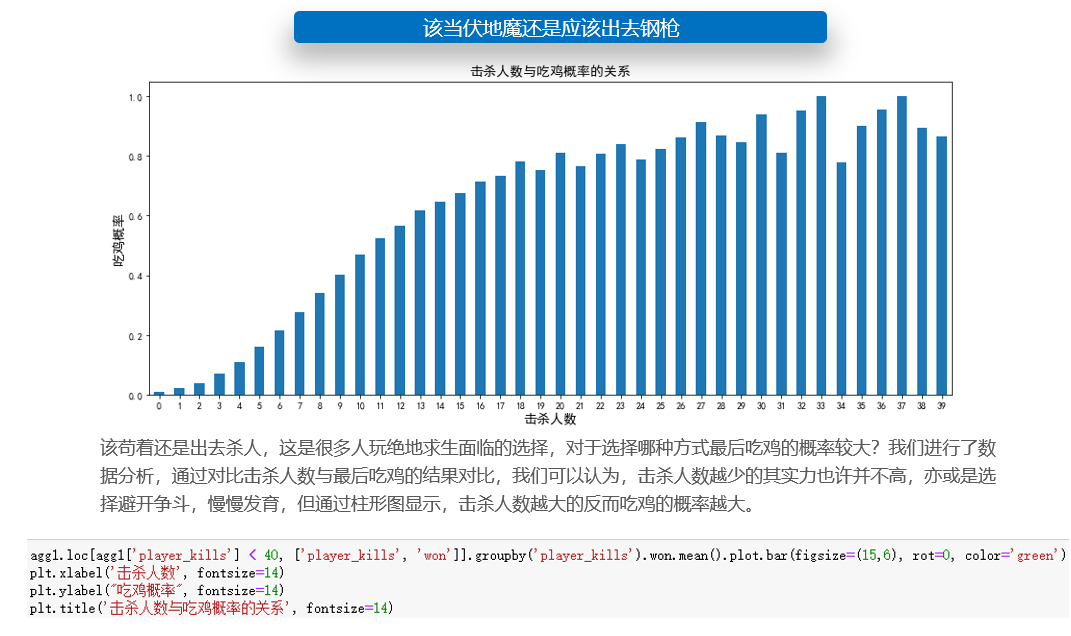


图3.3-6

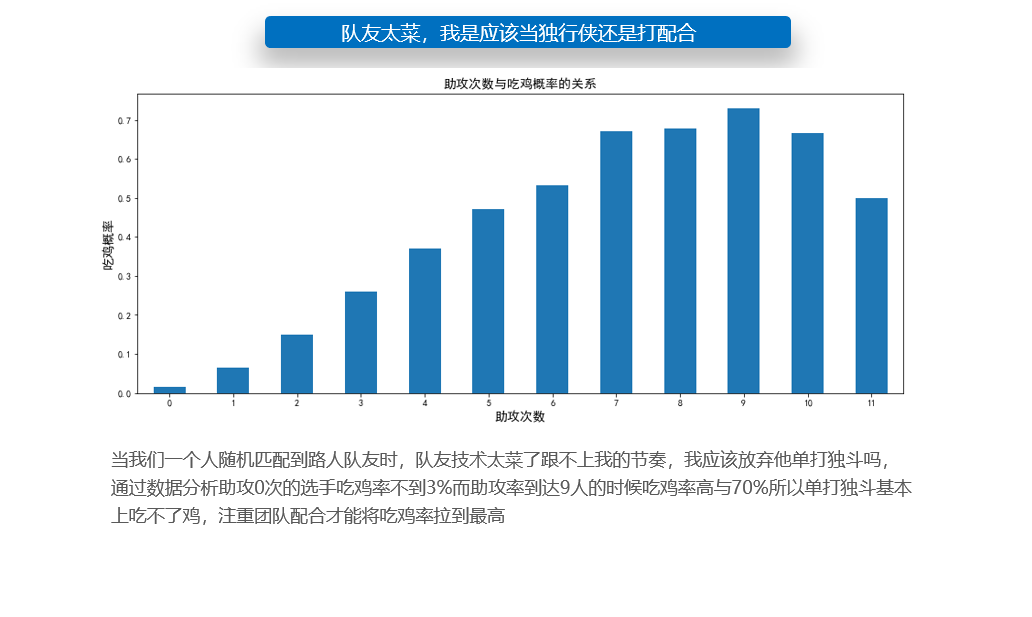


图3.3-7

## 4 总结

这次Python的课程设计和以往的有很大区别，以往的课程设计以制作软件为目的，这次则是以处理数据加上Python语言和其他语言编写格式的不同，让我对这次课程设计别有一番感受。首先就是，对于开头不在迷茫，俗话说“万事开头难”，但是这一次我却没有感到困难，反而在一开始就非常明确自己要干什么要怎么干需要用到哪些模块用到什么方法都非常的明确，而且在指定数据分析方向的时候也没有遇到太多的困难，有一种感觉和当时学习数据库时差不多，又好像觉得是编程版的Ece表格，筛选数据清洗数据最后以图形显示出来，一切都很熟悉，除了代码有一点生疏之外，其余的工作都感觉游刃有余，虽然我对代码这一个模块没有太多的参与大部分时间都在进行汇总和PPT的制作，但是对于每一个分析问题我都会去参与代码的实现帮助队员共同完成l。这次分组模式特别好，让我们每个人都清楚自己应该干什么，分工明确，有一种小组做项目的感觉，这种感觉实在别的课程设计上所没有的。

最后，这次Python课程设计让我有了很多收货，在复习了课程上的内容的同时也熟悉了团队配合，可以说是一次完美的体验。