|  |  |
| --- | --- |
| 学号 | 10212818118 |



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-著名网游《绝地求生》的数据分析 |
| 专 业 | 软件工程 |
| 班 级 | 软件1181 |
| 姓 名 | 王昶 |
| 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

[1编程环境与相关技术 8](#_Toc26292)

[1.1硬件平台 8](#_Toc13292)

[1.2项目托管平台-GitHub 8](#_Toc30697)

[1.3开发平台-jupyter 8](#_Toc24629)

[1.4编程语言-python 9](#_Toc9461)

[2 Github详细介绍 11](#_Toc20090)

[2.1 Github简介 11](#_Toc22674)

[3 程序设计 16](#_Toc7297)

[3.1数据分析库 16](#_Toc1915)

[3.1.1数据分析库导入见下图3.1-1 16](#_Toc18923)

[3.1.2数据分析库注解 16](#_Toc16755)

[3.2数据表 17](#_Toc900)

[3.2.1数据表导入见下图3.2-1 17](#_Toc21444)

[3.2.2数据表说明 18](#_Toc11865)

[3.3该当伏地魔还是应该出去钢枪 18](#_Toc20623)

[3.3.1数据提取代码见下图3.3-1 18](#_Toc22595)

[3.3.2生成击杀人数与吃鸡概率关系分析图见下图3.3-2 19](#_Toc10644)

[3.4 应当单打独斗还是与队友一起配合 19](#_Toc8710)

[3.4.1生成助攻次数与吃鸡概率的关系代码分析图见下图3.4-1 19](#_Toc17003)

[3.4.1生成助攻次数与吃鸡概率关系分析图见下图3.4-1和3.4-2 19](#_Toc5949)

[3.5选哪把武器可以帮你carry全场 20](#_Toc15675)

[3.5.1武器与击杀人数的统计代码见下图3.5-1与3.5-2 20](#_Toc19094)

[3.5.2武器与击杀人数的统计图见下图3.5-3与3.5-4 20](#_Toc7794)

[3.6物资丰富战况猛烈or安全地区 21](#_Toc323)

[3.6.1物资丰富战况猛烈or安全地区代码见下图3.6-1与3.6-2 21](#_Toc1814)

[3.5.2物资丰富战况猛烈or安全地区生成图见下图3.6-3与3.6-4 21](#_Toc20092)

[4 问题与不足 23](#_Toc664)

[4.1 代码编写上的问题 23](#_Toc18569)

[4.1 Github使用上的问题 23](#_Toc3160)

[4.1 PPT汇总上的问题 23](#_Toc2739)

[5 总结 24](#_Toc21054)

前言

《绝地求生》这款著名网络FPS游戏想必大家并不陌生，其主要是在游戏中，所有玩家进入一座名为Erangel的小岛，并在小岛内搜集物资进行第一人称枪战的游戏，存活到最后便是赢家。

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。

每一局游戏至多有100名玩家参与，玩家自由选择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，会有一个蓝色的范围圈逐渐缩小将玩家驱赶到一个较小的地方交火。这个蓝圈玩家称之为 "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。

所以，在游戏中，除了运气成分，与队友配合，还有自身的枪法与战术部署、对地图的熟悉程度同样重要，想要赢得比赛，最后大吉大利，成功吃到“鸡”，这些缺一不可，而此次课程设计便是对大量的玩家再游戏之后产生的数据进行分析，从而得出，当我们或是新手进行游戏的时候，如何能更快的上手，更容易的吃到“鸡”！

# 

# 1编程环境与相关技术

## 1.1硬件平台

CPU：Intel(R) Core(TM) i7-8700H CPU @ 3.20GHz 3.19GHz

最低要求：64位7代i3处理器及以上

内存：16G

最低要求：4G

硬盘空间：SSD-250G+机械硬盘1T

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/ %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。

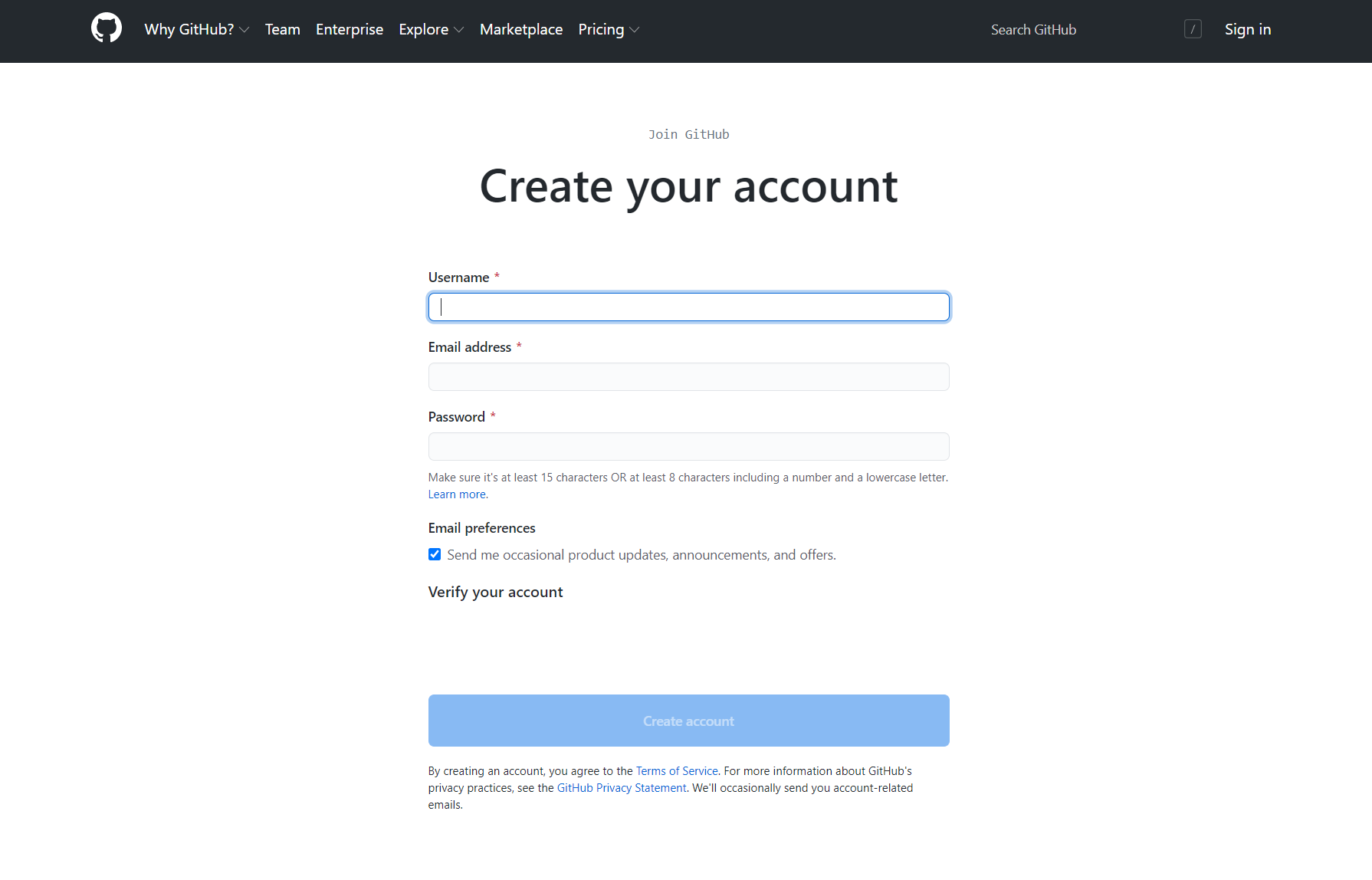
如前所述，作为一个分布式的版本控制系统，在Git中并不存在主库这样的概念，每一份[复制](https://baike.baidu.com/item/%E5%A4%8D%E5%88%B6" \t "_blank)出的库都可以独立使用，任何两个库之间的不一致之处都可以进行合并。

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 Github注册与登录

Github官网地址为：[https://github.com/](https://github.com/" \t "https://segmentfault.com/a/_blank)

Github注册如图2.2-1所示



用户名

邮箱地址

密码

图2.2-1

注册成功后如图2.2-2所示登录



用户名

密码

图2.2-2

2.3 仓库的创建

如图2.3-1,2.3-2所示：



图2.3-1

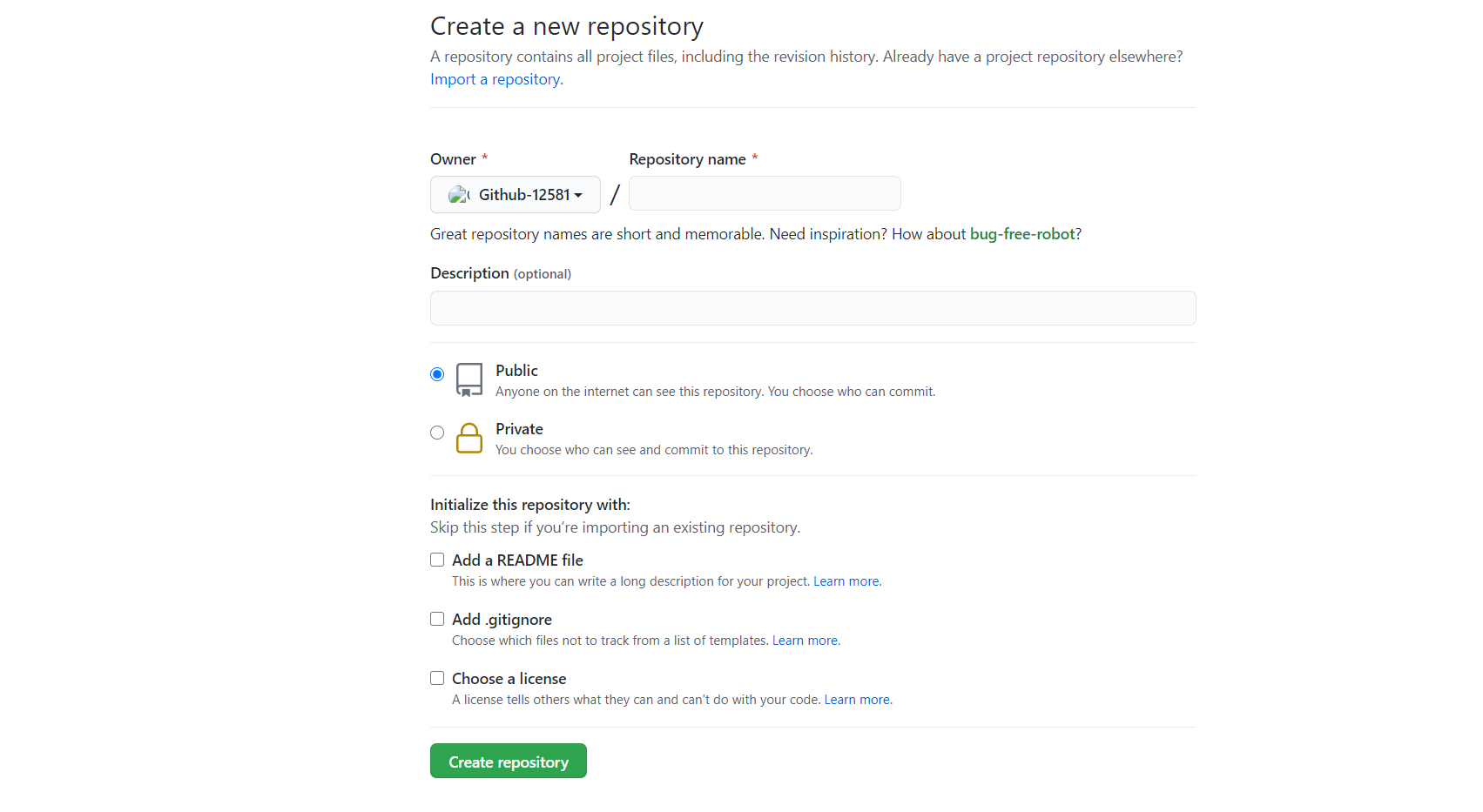


图2.3-2

2.4 Github与Git关系

Git 是一款免费、开源的分布式版本控制系统，他是著名的 Linux 发明者 Linus Torvalds 开发的。说到版本控制系统，估计很多人都用过 SVN ，只不过 Git 是新时代的产物，如果你还在用 SVN 来管理你的代码，那就真的有些落伍了。不管是学习 GitHub ，还是以后想从事编程行业，Git 都可以算是必备技能了，所以从现在开始建议你先去学习熟悉下 Git。  
而 GitHub 上面说了，主要提供基于 git 的版本托管服务。也就是说现在 GitHub 上托管的所有项目代码都是基于 Git 来进行版本控制的，所以 Git 只是 GitHub 上用来管理项目的一个工具而已，GitHub 的功能可远不止于此！

Github可用于学习优秀项目，多人协作，搭建博客、个人网站或者公司官网。

2.5 Ssh简介

当拥有了一个 GitHub 账号之后，就可以自由的 clone 或者下载其他项目，也可以创建自己的项目，但是你没法提交代码。仔细想想也知道，肯定不可能随意就能提交代码的，如果随意可以提交代码，那么 GitHub 上的项目岂不乱了套了，所以提交代码之前一定是需要某种授权的，而 GitHub 上一般都是基于 SSH 授权的。  
那么什么是 SSH 呢？ 简单点说，SSH是一种网络协议，用于计算机之间的加密登录。目前是每一台 Linux 电脑的标准配置。而大多数 Git 服务器都会选择使用 SSH 公钥来进行授权，所以想要在 GitHub 提交代码的第一步就是要先添加 SSH key 配置。

2.5.1 Ssh生成方法

Linux 与 Mac 都是默认安装了 SSH ，而 Windows 系统安装了 Git Bash 应该也是带了 SSH 的。大家可以在终端（win下在 Git Bash 里）输入 ssh 如图2.3.1-1证明你本机已经安装 SSH，紧接着输入 ssh-keygen -t rsa ，什么意思呢？就是指定 rsa 算法生成密钥，接着连续三个回车键（不需要输入密码），然后就会生成两个文件 id\_rsa 和 id\_rsa.pub ，而 id\_rsa 是密钥，id\_rsa.pub 就是公钥。这两文件默认分别在如下目录里生成：  
Linux/Mac 系统 在 ~/.ssh 下，win系统在 C:\Users\Username.ssh 下，都是隐藏文件，相信你们有办法查看的。

接下来要做的是把 id\_rsa.pub 的内容添加到 GitHub 上，这样你本地的 id\_rsa 密钥跟 GitHub 上的 id\_rsa.pub 公钥进行配对，授权成功才可以提交代码。

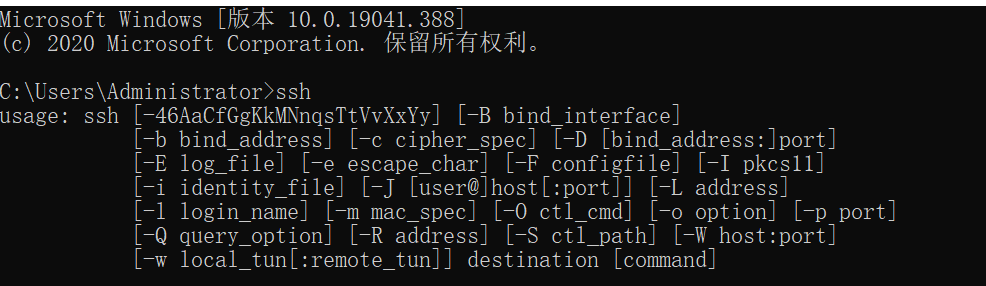


图2.5.1-1

2.6 Github基本概念

Repository：仓库的意思，即你的项目，你想在 GitHub 上开源一个项目，那就必须要新建一个 Repository，如果你开源的项目多了，你就拥有了多个 Repositories 。

Issue：问题的意思，举个例子，就是你开源了一个项目，别人发现你的项目中有bug，或者哪些地方做的不够好，他就可以给你提个 Issue ，即问题，提的问题多了，也就是 Issues ，然后你看到了这些问题就可以去逐个修复，修复ok了就可以一个个的 Close 掉。

Star：这个好理解，就是给项目点赞，但是在 GitHub 上的点赞远比微博、知乎点赞难的多，如果你有一个项目获得100个star都算很不容易了！

Fork：你开源了一个项目，别人想在你这个项目的基础上做些改进，然后应用到自己的项目中，这个时候他就可以 Fork 你的项目，这个时候他的 GitHub 主页上就多了一个项目，只不过这个项目是基于你的项目基础（本质上是在原有项目的基础上新建了一个分支，分支的概念后面会在讲解Git的时候说到），他就可以随心所欲的去改进，但是丝毫不会影响原有项目的代码与结构。

Pull Request：发起请求，这个其实是基于 Fork 的，还是上面那个例子，如果别人在你基础上做了改进，后来觉得改进的很不错，应该要把这些改进让更多的人收益，于是就想把自己的改进合并到原有项目里，这个时候他就可以发起一个 Pull Request（简称PR） ，原有项目创建人就可以收到这个请求，这个时候他会仔细review你的代码，并且测试觉得OK了，就会接受你的PR，这个时候你做的改进原有项目就会拥有了。

Watch：这个也好理解就是观察，如果你 Watch 了某个项目，那么以后只要这个项目有任何更新，你都会第一时间收到关于这个项目的通知提醒。

Gist：有些时候你没有项目可以开源，只是单纯的想分享一些代码片段，那这个时候 Gist 就派上用场了！

2.7 Github常见操作

当打开一个仓库后如图2.7-1所示：



图2.7-1

Code：这个好理解，就是你项目的代码文件而已，这里说明一下，每个项目通常都会有对该项目的介绍，只需要在项目的根目录里添加一个 README.md 文件就可以，使用 markdown 语法，GitHub 自动会对该文件进行渲染。

Issues：Issues 代表该项目的一些问题或者 bug，并不是说 Issues 越少越好，Issues 被解决的越多说明项目作者或者组织响应很积极，也说明该开源项目的作者很重视该项目。  
我们来看下Retrofit 的 Issues 主页，截至目前 close（解决） 了 1305 个 Issue，open （待解决）状态的有 37 个，这解决问题的比例与速度值得每位开源项目的作者学习。同样的，大家在使用一些开源项目有问题的时候都可以提 Issue，可以通过点击右上角的New Issue 来新建 Issue，需要添加一个标题与描述就可以了，这个操作很简单。

Pull requests：我们都知道 GitHub 的最大魅力在于人人都可参与，比如别人开源一个项目，我们每个人都可以一起参与开发，一起来完善，而这都通过 Pull requests 来完成，简称 PR。  
第一步登录你的 GitHub 账号，然后找到你想发起 PR 的项目点击右上角的 Fork 按钮，然后该项目就出现在了你自己账号的 Repository 里。  
第二步，把该项目 clone 到本地，然后修改的 bug 也好，想要新增的功能也好，总之把自己做的代码改动开发完，保存好。  
第三步，点击你 Fork 过来的项目主页的 Pull requests 页面，点击 New pull request 按钮,页面自动会比较该项目与原有项目的不同之处，然后最顶部可以方便直观的看到到底代码中做了哪些改动。  
第四步，写好标题和描述，然后我们点击中间的 Create pull request 按钮，至此我们就成功给该项目提交了一个 PR。  
然后就等着项目原作者 review 你的代码，并且决定会不会接受你的 PR，如果接受，那么恭喜你，你已经是该项目的贡献者之一了。

Projects：这个是最新 GitHub 改版新增的一个项目，这个项目就是方便你把一些 Issues、Pull requests进行分类，反正我觉得该功能很鸡肋，起码到目前为止基本没人用该功能，你们了解下就好。

Wiki:一般来说，我们项目的主页有 README.me 基本就够了，但是有些时候我们项目的一些用法很复杂，需要有详细的使用说明文档给开源项目的使用者，这个时候就用到了 Wiki。  
使用起来也很简单，直接 New Page ，然后使用 markdown 语法即可进行编写。

Pulse:Pulse 可以理解成该项目的活跃汇总。包括近期该仓库创建了多少个 Pull Request 或 Issue，有多少人参与了这个仓库的开发等，都可以在这里一目了然。  
根据这个页面，用户可以判断该项目受关注程度以及项目作者是否还在积极参与解决这些问题等。

Graphs:Graphs 是以图表的形式来展示该项目的一个整体情况。比如项目的全部贡献人，比如commits 的围度分析，比如某天代码提交的频繁程度等。

Settings:如果一个项目是自己的，那么你会发现会多一个菜单 Settings，这里包括了你对整个项目的设置信息，比如对项目重命名，比如删除该项目，比如关闭项目的 Wiki 和 Issues 功能等，不过大部分情况下我们都不需要对这些设置做更改。感兴趣的，可以自行看下这里的设置有哪些功能。

# 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

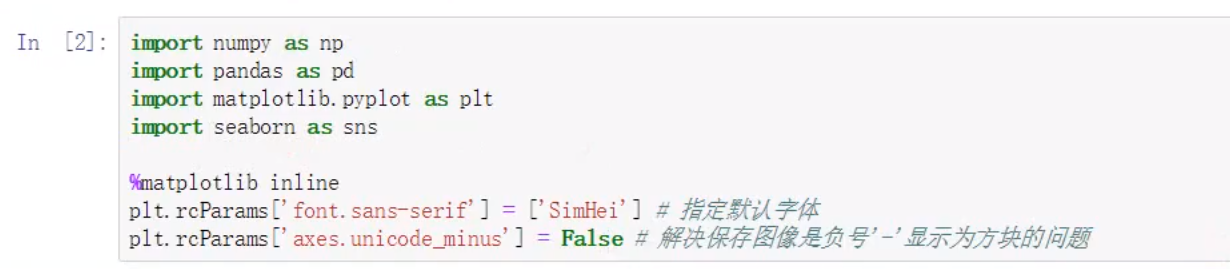


图3.1-1

### 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：老师给的20G 绝地求生比赛数据集其中小部分，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。

数据表注释如图3.2-2

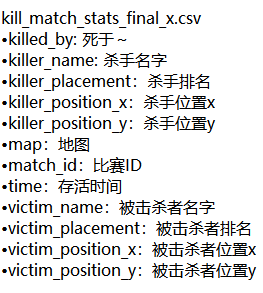


图3.2-1

## 3.3该当伏地魔还是应该出去钢枪

### 3.3.1数据提取代码见下图3.3-1

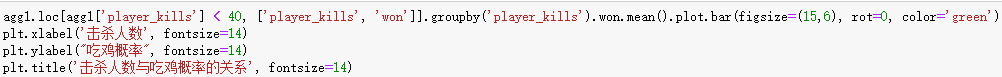


图3.3-1

图3.2-1

### 3.3.2生成击杀人数与吃鸡概率关系分析图见下图3.3-2

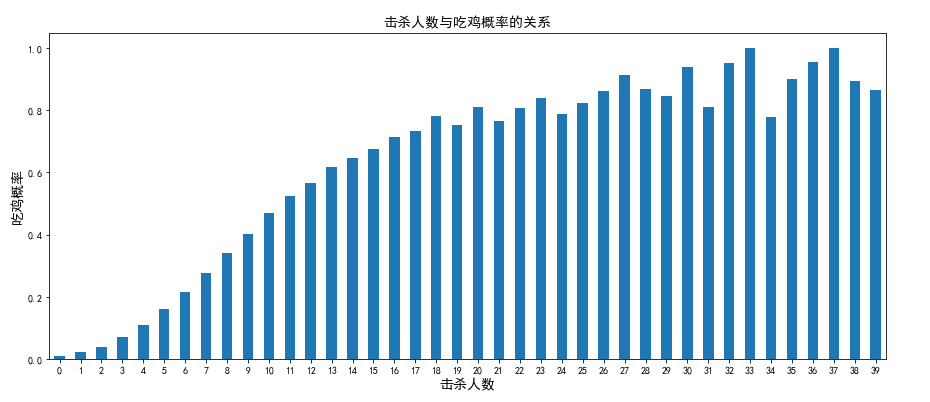


图3.3-2

## 3.4 应当单打独斗还是与队友一起配合

### 3.4.1生成助攻次数与吃鸡概率的关系代码分析图见下图3.4-1



图3.4-1

### 3.4.1生成助攻次数与吃鸡概率关系分析图见下图3.4-1和3.4-2

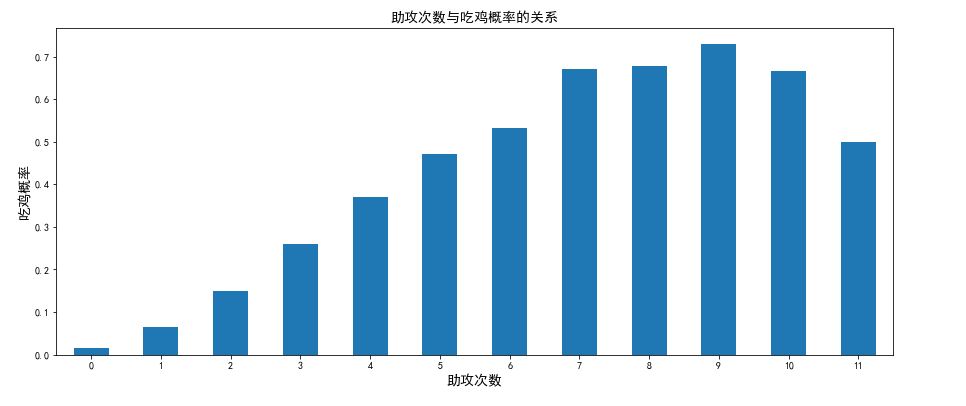


图3.4-2

## 3.5选哪把武器可以帮你carry全场

### 3.5.1武器与击杀人数的统计代码见下图3.5-1与3.5-2

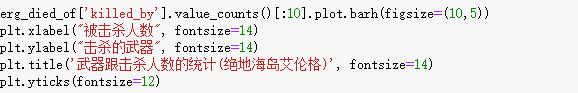


图3.5-1

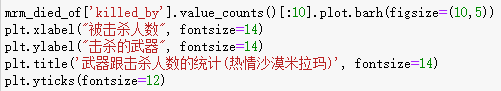


图3.5-2

### 3.5.2武器与击杀人数的统计图见下图3.5-3与3.5-4

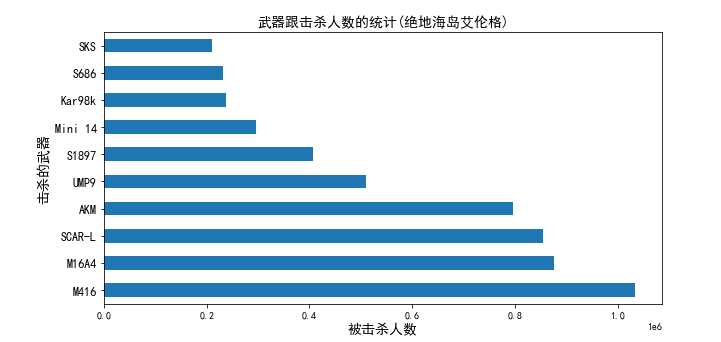


图3.5-3

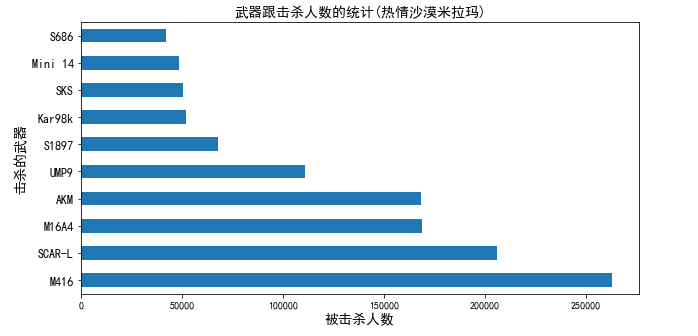


图3.5-4

## 3.6物资丰富战况猛烈or安全地区

### 3.6.1物资丰富战况猛烈or安全地区代码见下图3.6-1与3.6-2

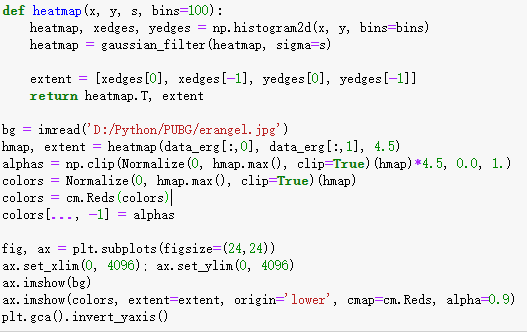


图3.6-1

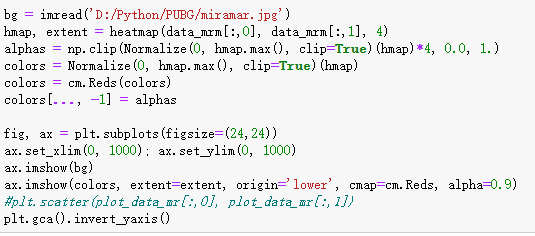


图3.6-2

### 3.5.2物资丰富战况猛烈or安全地区生成图见下图3.6-3与3.6-4

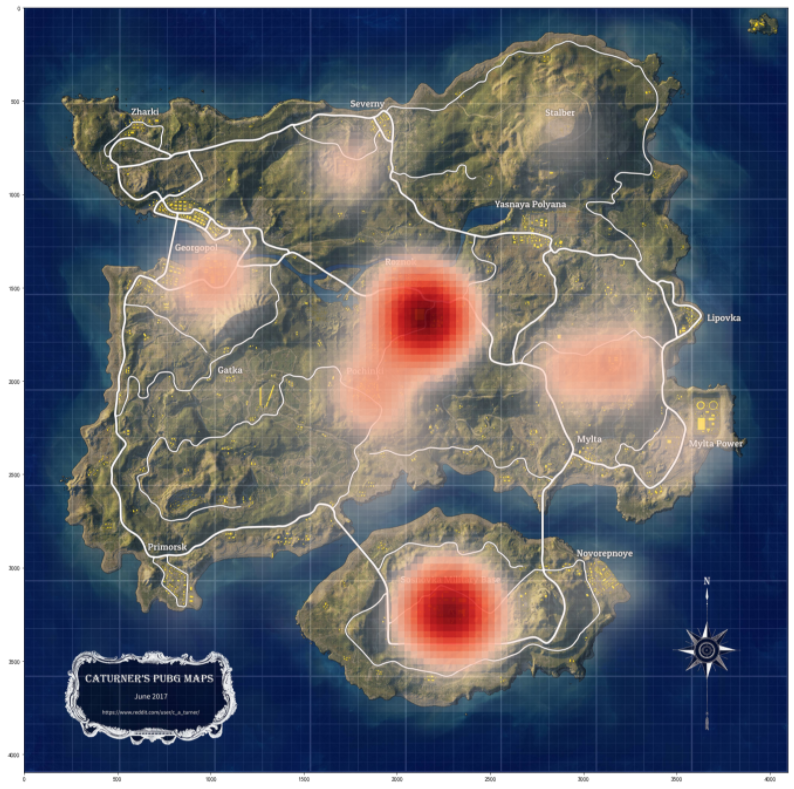


图3.6-3

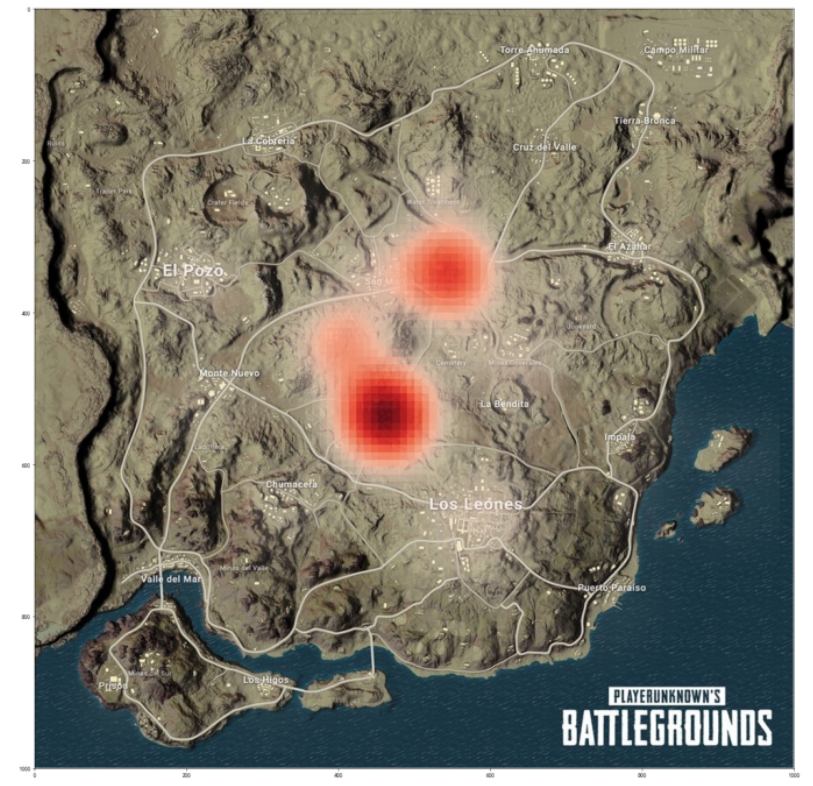


图3.6-4

# 4 问题与不足

## 4.1 代码编写上的问题

版本原因imread到了imageio包里面

mean() 函数定义：

numpy.mean(a, axis, dtype, out，keepdims )

mean()函数功能：求取均值

unique()是以 数组形式（numpy.ndarray）返回列的所有唯一值（特征的所有唯一值）

dropna()该函数主要用于滤除缺失数据。

## 4.1 Github使用上的问题

为什么在上传文件的时候要分主要分支和分支？因为在创建分支之后，如果有新的提交或者是新的改动，主分支的主要程序或是主要部分就不会受到影响，这样可以使得不同的任务组在不同的分支上进行开发，互相之间不会影响，同时加大开发进度，也保证了主线代码的完整性和可用性。

## 4.1 PPT汇总上的问题

当在PPT上汇总我们的学习成果时，因为很少参与到代码的编写，所以在写实验认识上的时候有些乏力，没有亲历的过程，无法完整的吧学习成果展现出来。

# 5 总结

通过这次作著名网游绝地求生的数据分析，我们受益良多，不仅加深了对python的认识理解，更让我们通过这次课设发现了其中的乐趣。

虽然在编写程序的过程中遇到了很多困难，比如第一次使用Github，第一次小团队的进行编写程序，互相之间的配合与分工也不是特别的默契，加上三个人每个人自己的时间有时候统一不起来，导致程序的编写断断续续，每个人也都有自己遇到的难题，但也正是这次课设的锻炼，我们的Python语言有了很大的提升，我们也有了相当的兴趣，觉得数据挖掘、大数据等等一系列无法分割的类别都有很深奥的学问。

同时经过python对绝地求生的数据分析之后，我们也了解了在游戏当中，如何才能提升自己最后“吃鸡”的概率，以后小伙伴们开黑打游戏胜利的概率也大大提高了。

**课程设计成绩评定表**

|  |  |  |  |
| --- | --- | --- | --- |
| 课程设计题目 | 数据分析的学习与实践 | | |
| 课程设计学生答辩或质疑记录： | | | |
| **评 分 依 据** | | **分 值** | **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、遵守纪律 | | 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、创新能力 | | 10分 |  |
| 总 分 | | 100分 |  |
| 最终评定等级为：  指导老师签字：  2021 年 1 月 15日 | | | |