

BEIJINGINSTITUTEOFPETROCHEMICALTECHNOLOGY

数据采集与预处理实践

小组报告

League of Legends Pro League数据采集与预处理实验报告

|  |  |
| --- | --- |
| 团 队 组 号： | 8 |
| 组 长： | 张今朝 |
| 组 员： | 周一凡、刘宇恒、侯志昊 |
| 指 导 教 师： | 王晓伟 |
| 提 交 时 间： | 2022.7.1 |

目录

[第一部分：选题背景 1](#_Toc107547814)

[第二部分：数据源描述 2](#_Toc107547815)

[第三部分：具体实验过程 4](#_Toc107547816)

[3.1数据采集阶段 4](#_Toc107547817)

[3.2数据存储与预处理阶段 4](#_Toc107547818)

[3.3数据可视化阶段 4](#_Toc107547819)

[3.4数据分析阶段 4](#_Toc107547820)

[第四部分：实验总结 7](#_Toc107547821)

第一部分：选题背景

《英雄联盟》（League of Legends，简称LOL）是由美国拳头游戏（Riot Games）开发、中国内地由腾讯游戏代理运营的英雄对战MOBA竞技网游，自从2013年开始举办LPL春/夏季联赛后，英雄联盟的热度在国内不断提高，**关注人群规模庞大，**影响群体越来越多，同时随着更多资本入局，联赛奖金的上调以及场馆的建设和周边产品的推出，使得这一游戏已经成为了中国电竞行业的重要组成，这也让更多的赛事选择在中国举办，而中国战队在国际赛事上的表也是精彩纷呈，在S8、S9以及S11三次获得英雄联盟全球总决赛的冠军，中国选手的身价的不断上升，逐渐开始加盟外国战队走向世界，英雄联盟是划时代的产物，它现在的高度不止局限于游戏，它正在慢慢演变成一种潮流、一种文化、一种精神，这也从侧面佐证了我国电竞行业的崛起趋势。

2022年的赛季已经过半，距离世界赛也越来越近，本小组意通过课程所学，对2020年春季赛至2022年春季赛这5给赛季的数据进行数据采集与预处理，收集每个赛季的战队、选手以及英雄数据，从不同方面进行分析，优胜劣汰，最终由分析结果得出哪几支队伍能够成功晋级世界赛，即哪几支队伍能够获得2022年夏季赛的前四名。

当然英雄联盟这一游戏的影响因素有很多，例如每个版本都会有不同的强势以及弱势英雄，选手的状态以及战队的运营情况都会对最终成绩造成影响，所有本小组的目的是在不考虑外部因素以及不可抗力的情况下对数据进行分析，本次分析仅具有参考价值，结果以实际情况为准。

第二部分：数据源描述

本次我们组选择的数据源为英雄联盟赛事官网（腾讯游戏）。

其中2020年春季赛：

（<https://lpl.qq.com/es/data/rank.shtml?iGameId=134&sGameType=1,5>）

2020年夏季赛：

（<https://lpl.qq.com/es/data/rank.shtml?iGameId=134&sGameType=7,8>）

2021年春季赛：

（<https://lpl.qq.com/esnew/data/rank.shtml?iGameId=148&sGameType=1,5>）

2021年夏季赛：

（<https://lpl.qq.com/esnew/data/rank.shtml?iGameId=148&sGameType=7,8>）

2022年春季赛：

（<https://lpl.qq.com/esnew/data/rank.shtml?iGameId=167&sGameType=1,5>）

观察以上URL，URL繁多且规律不统一，因此小组决定使用selenium的By子模块来定位元素，并且模拟鼠标操作点击进行拟人爬取，此步骤简化了访问多个URL的复杂步骤。

来到LPL数据官网中，大家可以时刻关注赛季的进展以及自己喜爱的战队或者选手的整体数据，同时本小组可以找到适合我们的数据以及对比情况，并且数据全面、涵盖面广泛、分类清晰，有利于本小组之后进行数据采集与预处理。此平台的数据板块内容全面，存在近10万条战队、选手以及英雄数据，其中战队数据包含出场次数、胜负情况、胜率等10个维度，选手数据包含场上位置、总击杀、总助攻、总死亡、KDA等14个维度，英雄数据包含登场次数、pick比率、ban比率、胜率等9个维度。以上三类数据使用了JavaScript脚本来明文展示，但对网页元素进行观察后，所有数据均存在于本页面，因此使用BeautifulSoup模块定位元素进行爬取。同时，部分战队有重建、更名或解散的情况，这个问题我们在后期对所有数据进行筛选以及清洗，并以战队为单位进行整理。

网页整体的组织结构非常的工整，数据齐全，几乎没有空值或重复值，我们预计爬取的有效数据是1200条左右，由于数据规范化程度很高，因此经过数据清洗与数据规约后，有效数据数量几乎保持不变。

根据需求，我们的数据字段分为三类：

战队：trank，team，appearance，win\_lose，winRate，totalKill\_perGame，totalDie\_perGame，catchEye\_perGame，rowEye\_perGame，money\_perGame，baron\_perGame，dragon\_perGame；

选手：prank，player，position，appearance，modelViewPresenter，totalKill\_perGame，totalAssist\_perGame，totalDie\_perGame，killDieAssist，money\_perGame，creepScore\_perGame，catchEye\_perGame，rowEye\_perGame，attendanceRate\_perGame，contrapositionEconomicDifference，damage，gold；

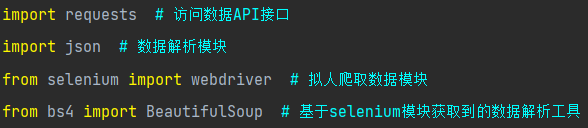
英雄：hrank，hero，appearance，pickRatio，banRatio，winRate，totalKill\_perGame，totalAssist\_perGame，totalDie\_perGame，killDieAssist，commonPlayers；

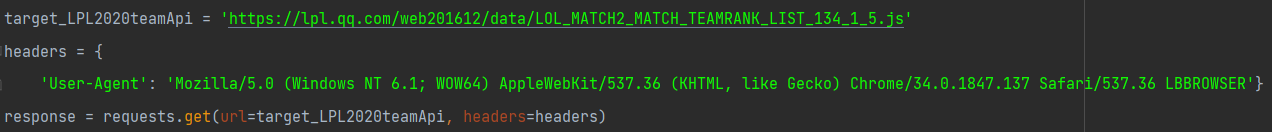
有了以上数据的支持，我们的最终分析结果将会更加具有可信度。

第三部分：具体实验过程

3.1数据采集阶段

在正式采集数据的阶段，本小组对数据源进行了考察，发现2020年的数据与其它年份的数据有结构上的不同，因此，本小组对2020年的数据使用了requests模块访问数据API接口进行了爬取，对其它年份的数据采用了selenium模块进行了爬取。

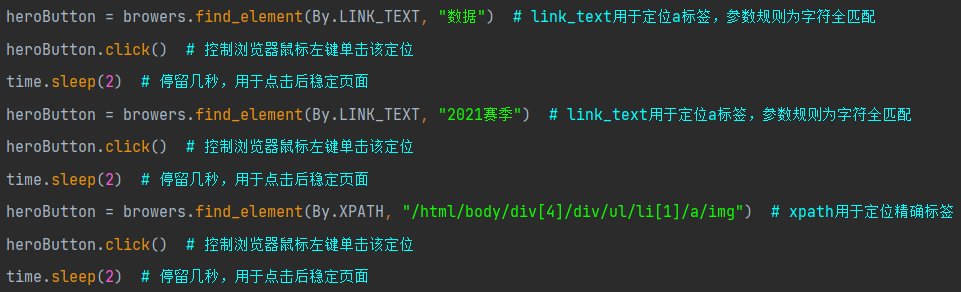






由于目标数据存放在很多不规则的URL中，因此需要依靠msedgedriver来模拟鼠标操作，同时，该手段需要依赖selenium.webdriver.common.by的子模块By来定位元素标签，使用browers.find\_element方法来定位元素，By所支持的定位器分类如下：CLASS\_NAME = 'class name'；CSS\_SELECTOR = 'css selector'；ID = 'id'；LINK\_TEXT = 'link text'；NAME = 'name'；PARTIAL\_LINK\_TEXT = 'partial link text'；TAG\_NAME = 'tag name'；XPATH = 'xpath'。

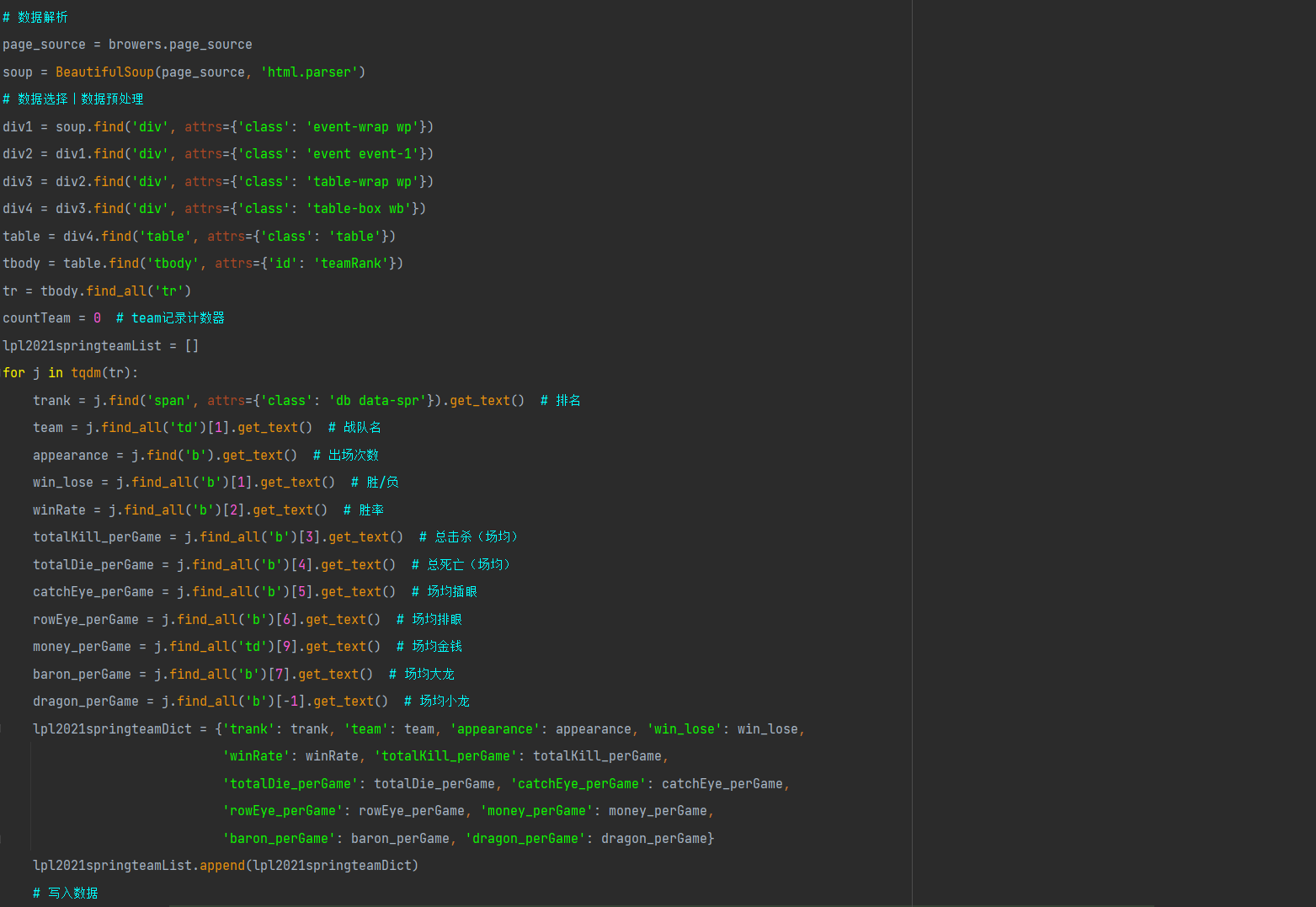
在多次踩坑之后，解决了模拟鼠标操作之后页面不稳定以及页面焦点不在新页面上的问题，即使用time.sleep方法稳定跳转后的页面。



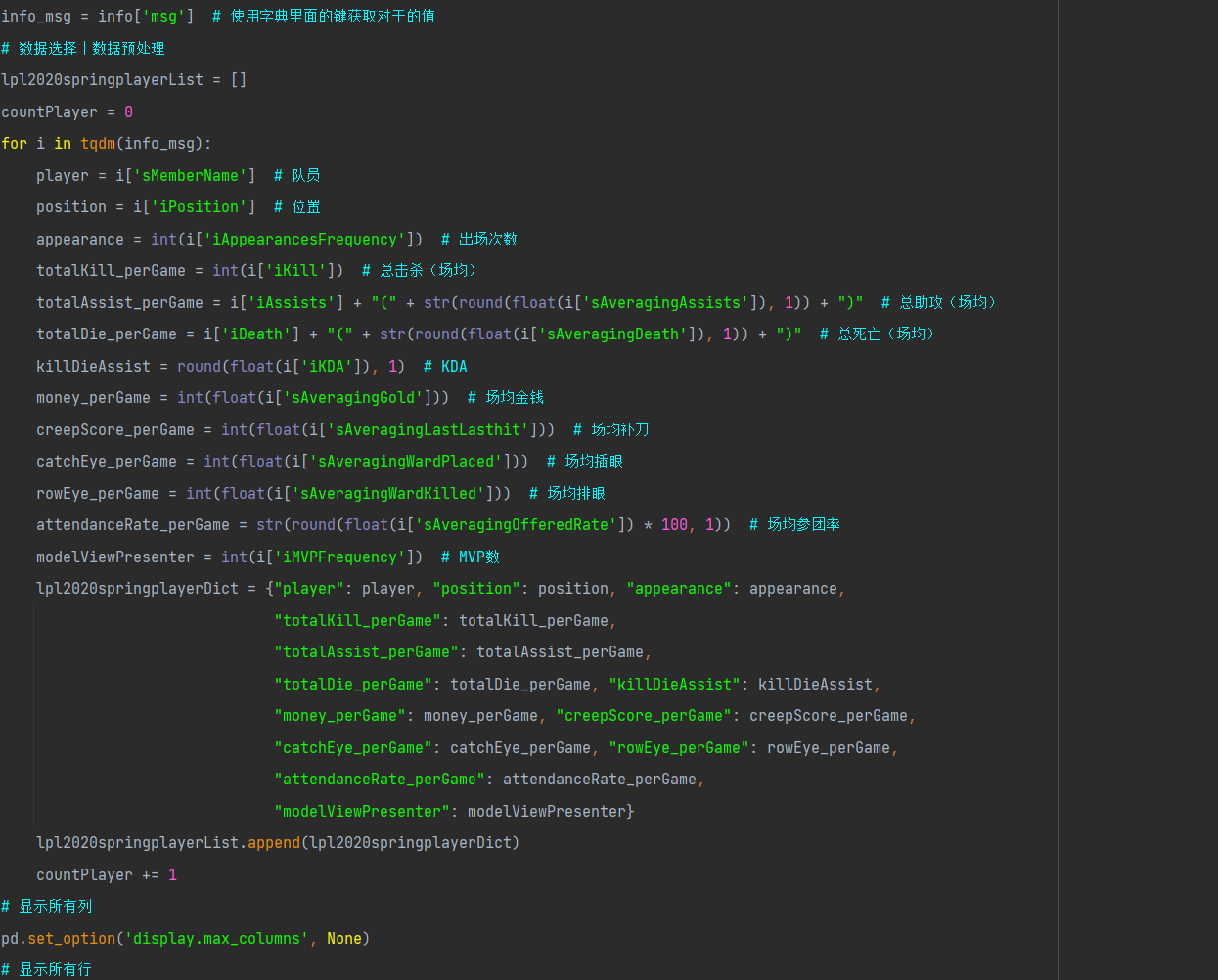
使用windows=browers.window\_handles获取父页面的句柄，为页面切换做准备；browers.switch\_to.window(windows[-1])，该方法用于新旧窗口之间的切换，控制浏览器将窗口切换为第-1个页面（即新页标签）。



在requests模块爬取过程中，搭配json模块进行数据解析；在selenium模块爬取过程中，使用BeautifulSoup的find与find＿all方法来查找元素。在所有年份的数据采集过程中，嵌套了tqdm进度条模块，使得爬取过程可视化，更加直观、清晰。使用time.perf\_counter方法记录程序开始与结束时间，用于统计程序执行耗费的时间。



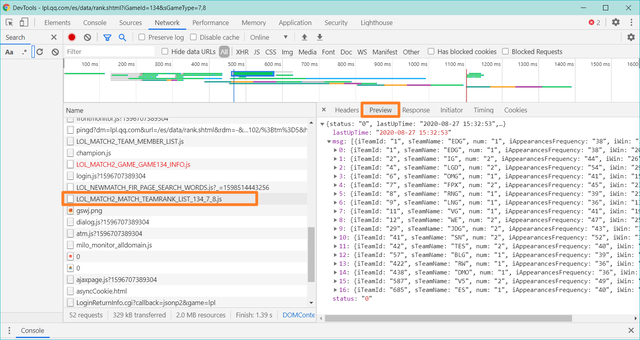




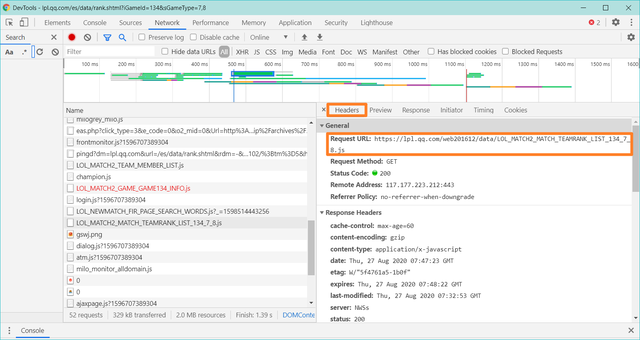
2020年数据：

在官网界面，右键点击检查（按F12打开开发者工具），点击【Network】，按CTRL+R 刷新之后，在【name】下方找到【LOL\_MATCH2\_MATCH\_TEAMRANK\_LIST\_134\_7\_8.js】，接着点击【Preview】，就会出现如下界面。





此时得到的数据是用json包装后的，利用Python简单处理一下就能获取到想要的数据。这些数据是用接口形式传输的，可以直接爬取API接口，点击旁边的【Headers】，复制【Request URL】后面的网址。

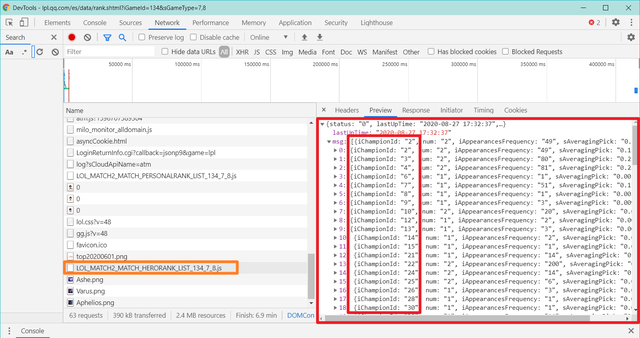


接下来打印请求的网址后的结果。

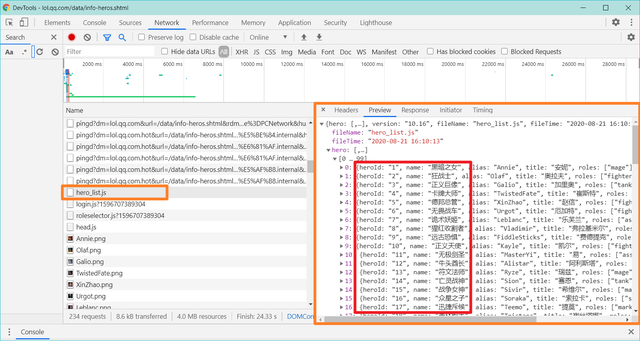


此时所有的战队信息已经获取成功，可是输出的内容是一个str（字符串）对象，需要使用json.loads()函数将str对象转换为dict（字典）对象。再使用字典和列表的方法就可以将想要的数据提取出来，并且存放到另外一个字典里面。个人数据的爬取过程同理。

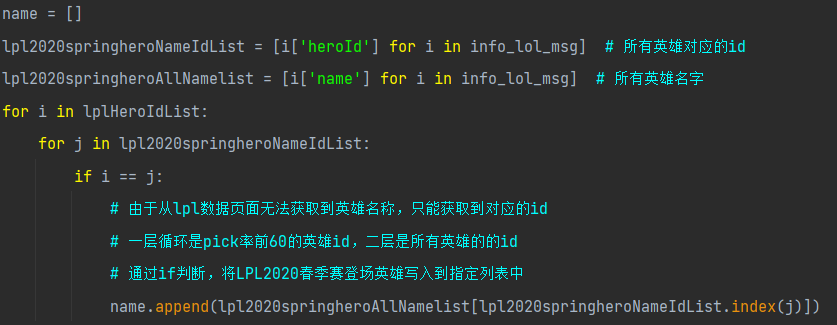
然而,获取英雄数据就不是那么顺利了,还是回到官网,点击【英雄数据】还是右键检查,找到【LOL\_MATCH2\_MATCH\_HERORANK\_LIST\_134\_7\_8.js】



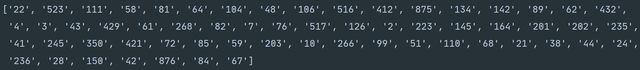
观察数据，没有英雄的名称，只有【iChampionId】和其他数据，想要查看英雄的名称,我们的第一反应就是访问英雄联盟官网，右键检查,找到【hero\_list.js】,结果非常尽人意！

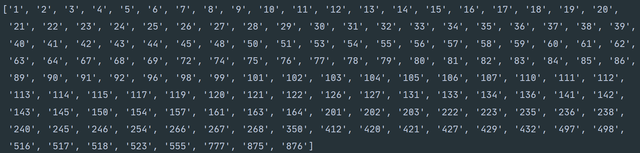


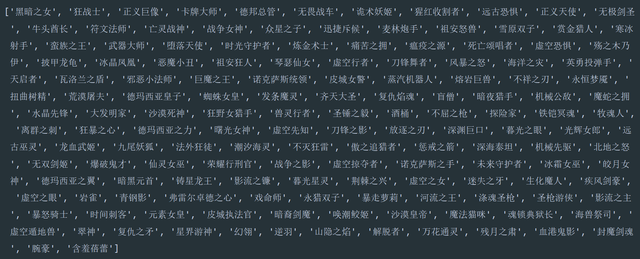
仔细观察,发现英雄名称前面有个【heroId】,再和刚才的【iChampionId】进行比对，发现二者是相对应的。从两个接口获取到的数据,需要保证【iChampionId】的英雄匹配到正确的名称，因此作出如下判断。

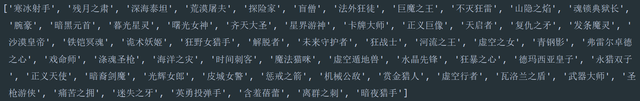


过程图如下：





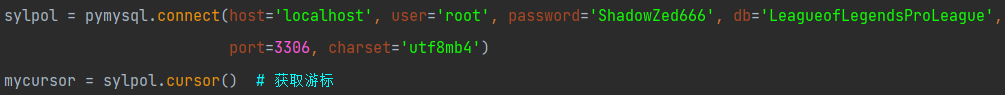


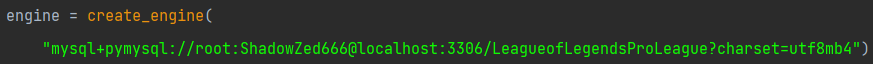


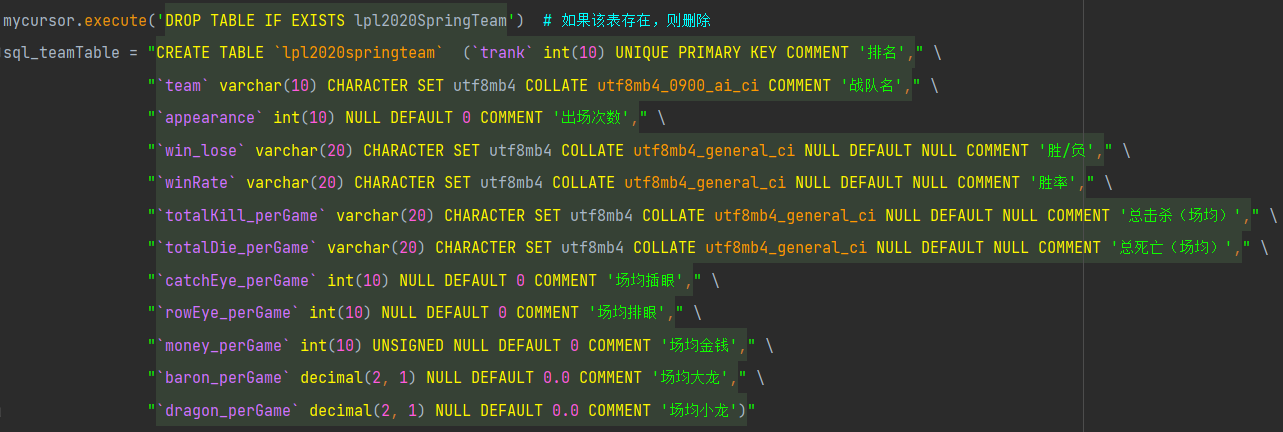
至此，数据采集阶段结束，所有数据已经准备就绪，下一步需要进行数据预处理与存储。

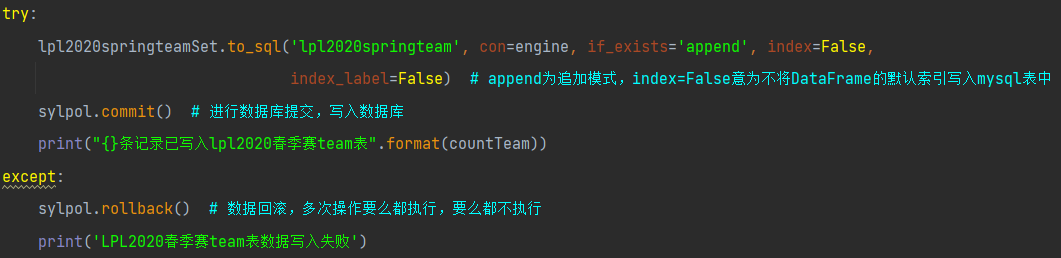
3.2数据存储与预处理阶段

对于爬取到的数据，本小组新建了一个名为LeagueofLegendsProLeague的数据库，为提高存储便捷性，使用了python代码编写了建表语句。同时，将两种不同结构的数据写入数据库中，分别使用了sqlalchemy模块的子模块create\_engine和pymysql模块。







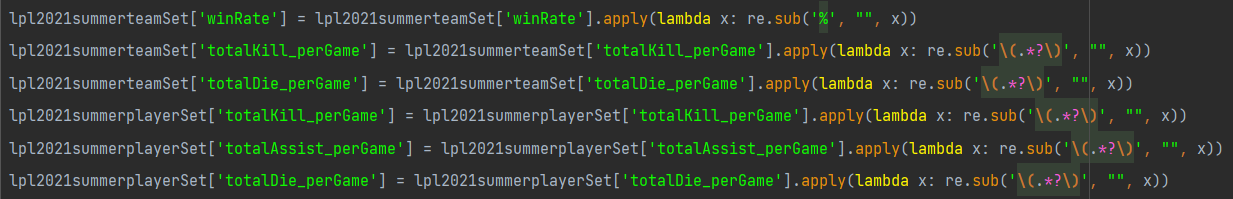


将采集到的数据转换为DataFrame类型，观察到在team数据帧中的winRate、totalKill\_perGame、totalDie\_perGame字段以及player数据帧中的totalAssist\_perGame字段为不规范的数据类型，因此引入re模块对以上字段进行数据清洗，同时，辅助以apply函数及匿名函数，具体实例演示：

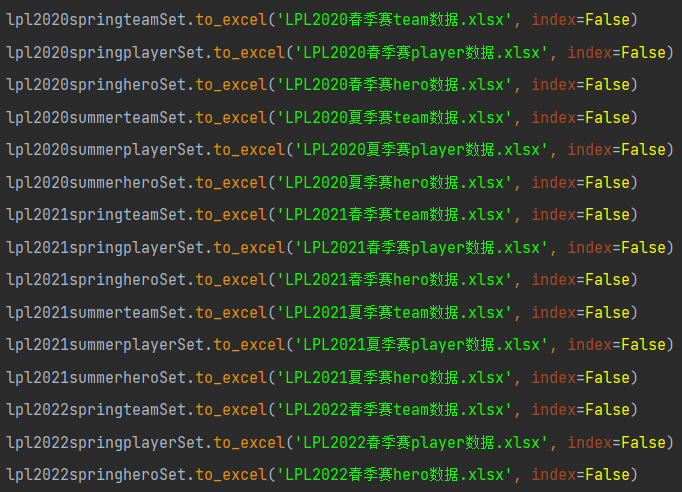




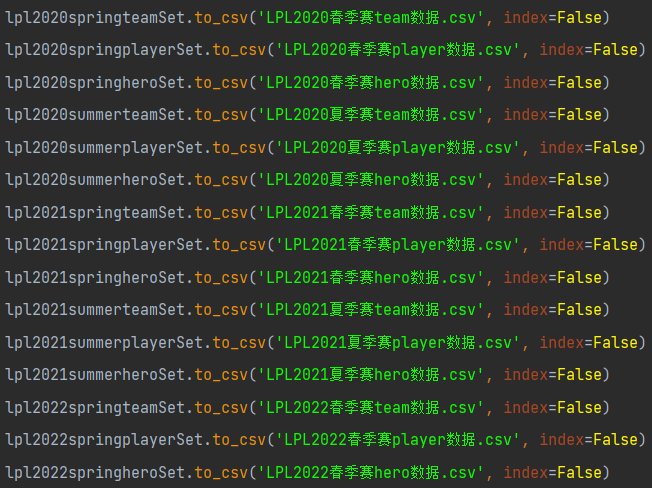




将以上所有DataFrame写入xlsx文件，以备后续数据分析使用。

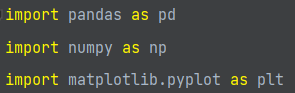


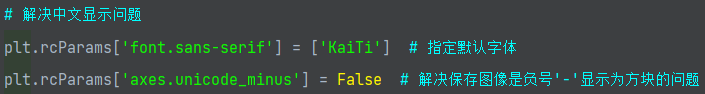
再将所有DataFrame写入csv文件，供给后续提交。



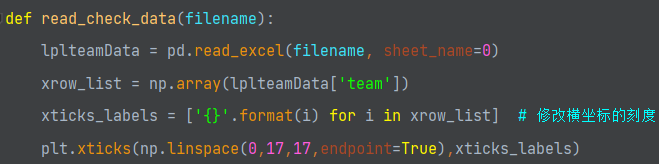
3.3数据可视化阶段

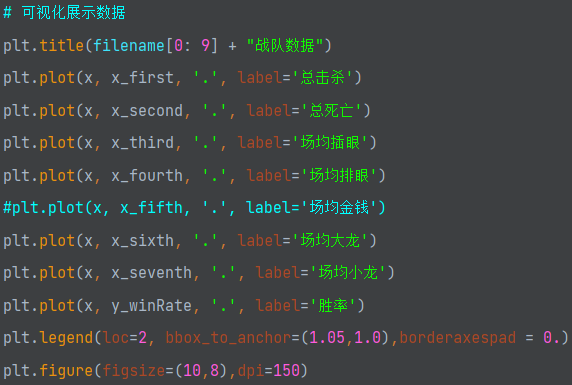
上述xlsx文件已经就绪，数据可视化需要读取以上文件，涉及到numpy、pandas模块进行提取Series，辅之matplotlib的子模块pyplot进行绘图。考虑到图片以及坐标轴刻度需要使用中文，因此先将字体与中文显示问题解决。

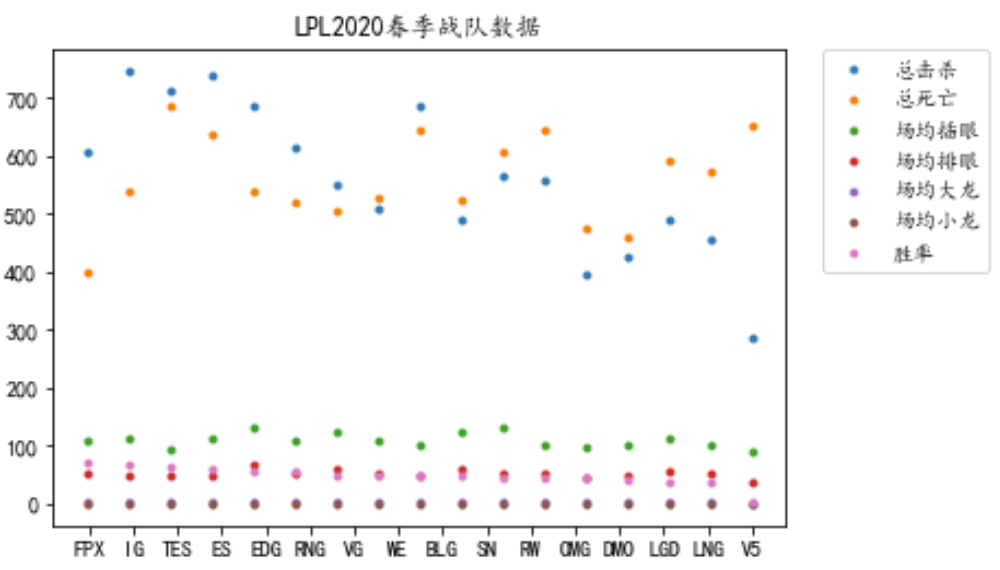


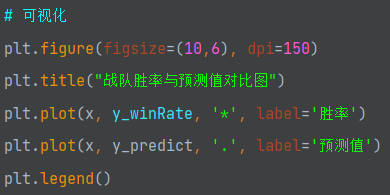


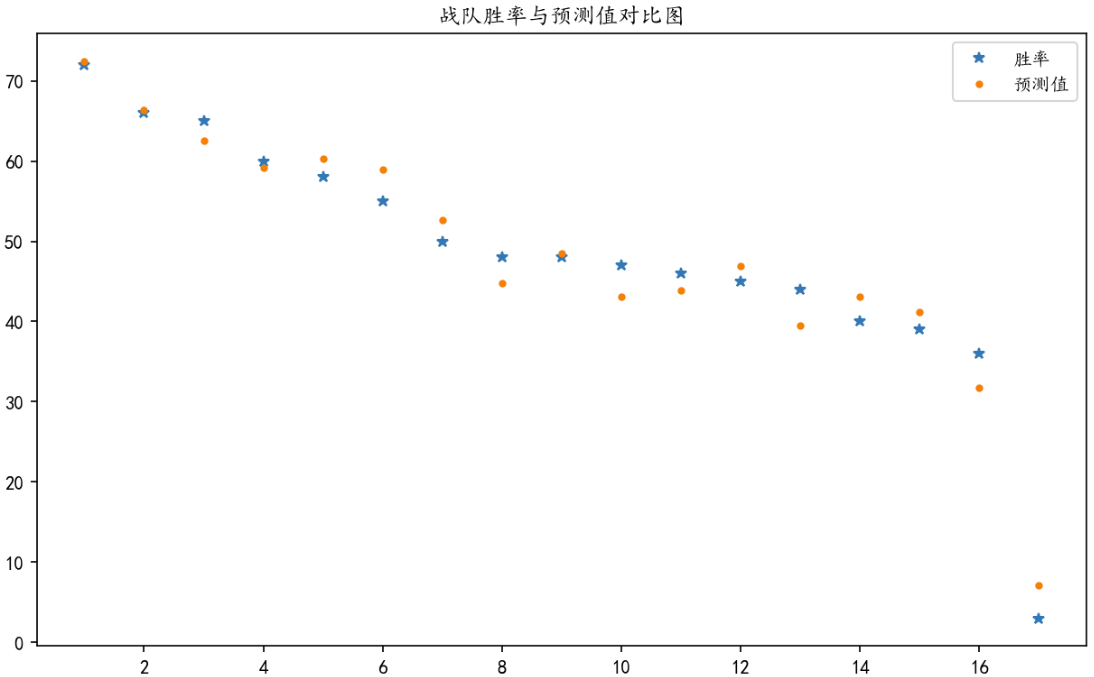
将x坐标轴的刻度改为战队名称，需要从DataFrame中提取team列。

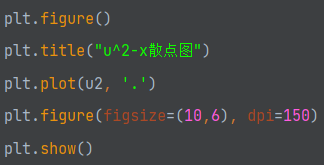


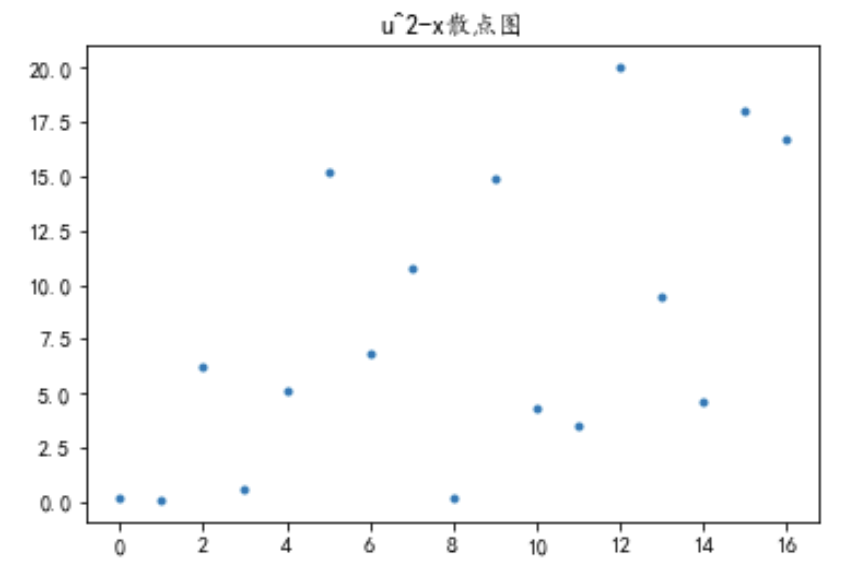


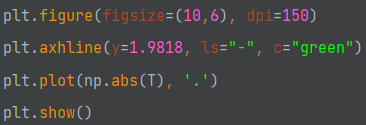


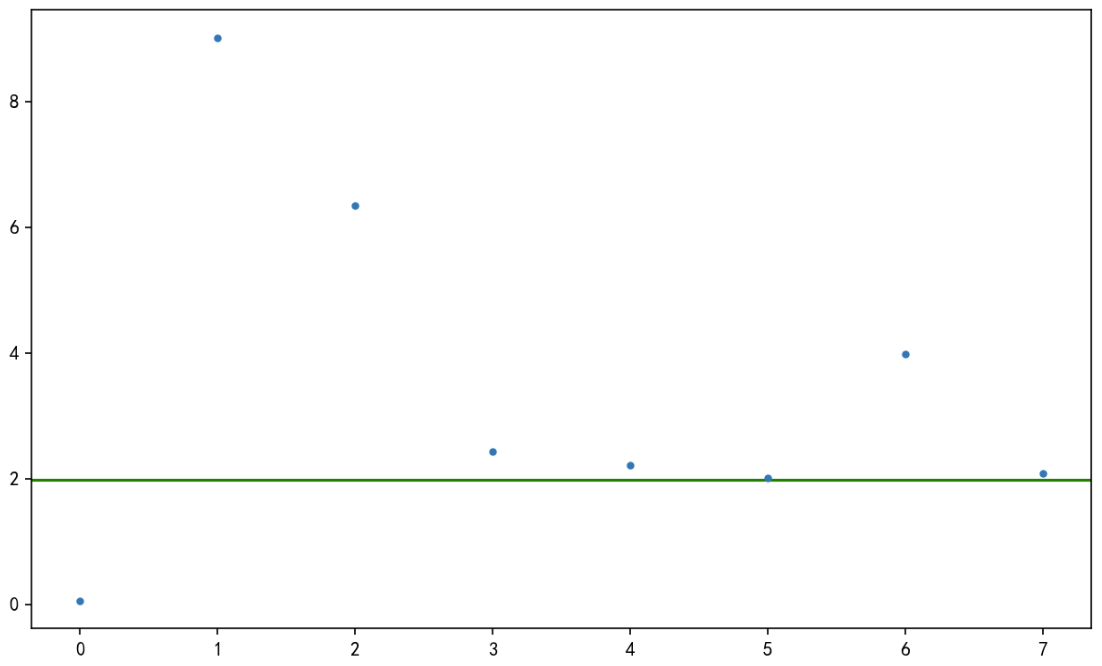








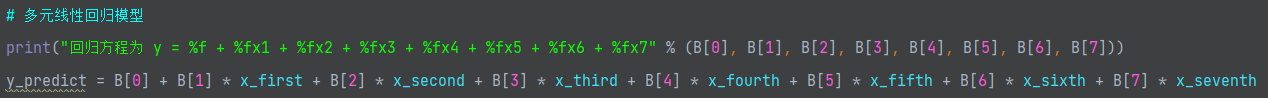




3.4数据分析阶段

本阶段使用statsmodels模块对以上数据进行多元线性回归分析、相关分析、显著性检验、拟合优度检验、异方差性检验，选取战队数据对胜率进行预测，即胜率作为因变量，选择总击杀、总死亡、场均插眼、场均排眼、场均金钱、场均大龙、场均小龙作为自变量，分别为x1，x2，x3……建立多元线性回归模型。

多元线性回归模型：



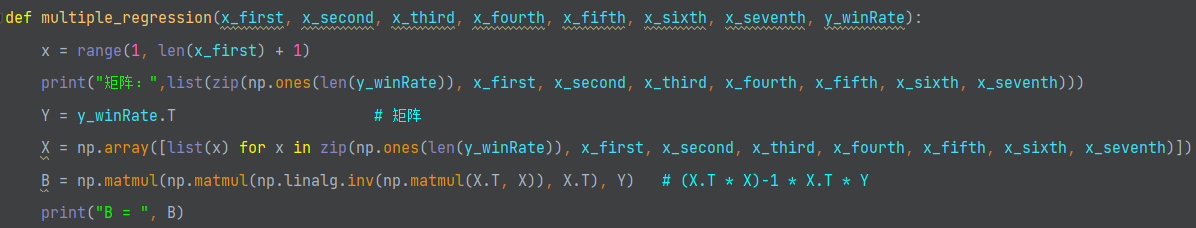
建立回归模型并求得方程如下图所示。



其中 y 为要预测的变量，x 为影响 y 值的变量，b 为回归系数，计算方式为：

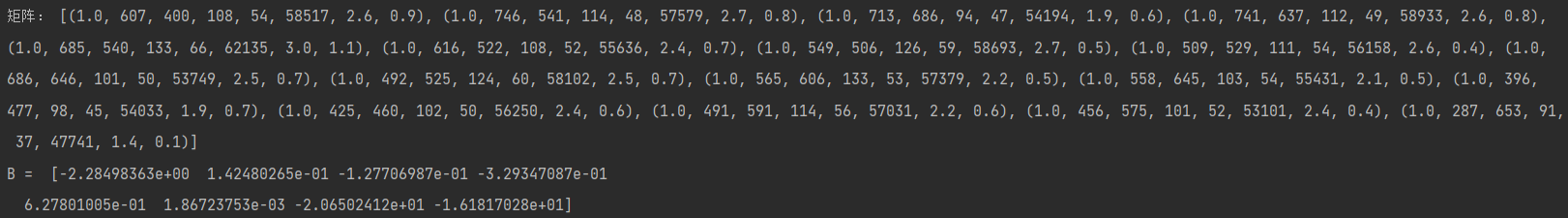
B=

计算结果为一个矩阵，分别对应b0，b1，b2，b3……



从下图中可以看出，胜率与总击杀、总死亡、场均大龙等数据呈现出较为一致的联动变化关系，除总死亡以外均呈现相同的变化趋势，根据相关知识，推测以上数据存在线性相关关系。

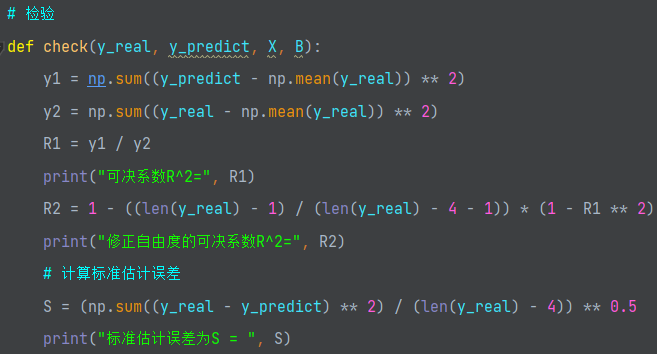
根据公式，求解出回归系数显著性检验的统计量如下图所示，通过显著性检验。



检验：

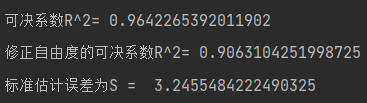
通过计算可决系数进行多元线性回归方程的拟合优度检验，可决系数是用来反映回归模式可靠程度的一个统计指标，可决系数越大，说明模型的预测值和观测值越接近，模型的拟合优度越好。计算方法如下：

其中，是回归方程计算的预测值，y为实际值，是实际值的平均值。



通过计算可决系数与修正自由度的可决系数进行多元线性回归方程的拟合优度检验如下图所示。可以看出以上数据存在高度相关性，并且该回归模型拟合度高。

从u^2散点图可以看出，u^2的值在0~20之间，大致呈现为一条斜率为正的直线，具有异方差性。根据公式求解出标准估计误差为下图（有一定的误差）。



后四个赛季的分析过程同上，分别绘图分析与回归分析，在得到回归方程后计算了标准估计误差，进行了拟合优度检验与回归系数检验，通过检验后进行预测。虽然数据间已经呈现高度相关性，但预测出来的结果仍有一定的偏差，但总的来说有不错的效果。此处不再赘述，可视化结果见ipynb文件。

通过对比本组的此次实践所爬取到的战队数据可以得出，RNG、V5以及TES这三支战队的综合实力相较于其他队伍要更强一些，所以我们预测这次2022年的夏季赛这三支战队将大概率进入到四强当中。而通过爬取到的数据显示EDG、JDG、WBG这三支战队的实力相当，所以我们推测四强当中的最后一个席位将由这三支战队中的其中一支占据。

在对EDG、JDG、WBG这三支战队进行分析的时候，由于实力相近我们决定采用对位分析的方法将战队里的选手进行逐一分析（对位分析即将两只战队中的同一位置的选手进行比较分析）。

通过我们的数据分析得出在其他选手的水平相当的情况下EDG战队中的中单和下路这两个位置的选手要比另外两只战队中的同对位选手的状态更好，发挥的更加稳定。并且EDG战队拥有丰富的大赛经验以及冠军基础（曾在2021年夺得全球总决赛冠军），再加上EDG战队的俱乐部成立于2013年，相比于另外两支队伍要早很多，基底要更加的深厚。

WBG战队有着顶尖水准的上单，但是由于队内的问题例如选手位置分配的不合理，在比赛当中选手资源分配不合理以及选手不契合战队体系等导致战队的整体水平略逊于EDG。

而JDG战队整体水平就比较稳定了，但这也是最大的问题，虽然发挥稳定但是过于的稳定以至于略显平庸了，并且在比赛当中JDG战队缺少亮点，缺乏选手的高光时刻。所以我们认为最后一个四强成员战队有很大的概率是EDG战队。

第四部分：实验总结

张今朝：

首先，我很庆幸和珍惜《数据采集与预处理实践》这门课程，它可以从心理以及实际行动上督促我捡起往学期的理论与实操知识，比如《Python语言》、《大数据管理与应用》、《数据采集与预处理》；又让我探索了新的知识，比如进度条模块tqdm、定位元素点击以及控制浏览器模拟鼠标操作的第三方模块By、依赖pandas库的DataFrame写入数据库模块sqlalchemy、time模块的新函数perf\_counter()、2D绘图模块matplotlib、基于统计学的数据分析模块statsmodels。

其次，在具体实验过程中，从提出需求、考察数据、尝试性编写代码、解决Bug、优化代码到最终的数据可视化、数据分析、预测结果、实验总结，我的收获颇多。浅举几个例子：需求不合理，改动了很多次；目标数据的结构不一致，被迫使用两个模块分别爬取；编写代码犯了很多常识性的错误，这也时刻提醒着我拾起从前的知识；由于使用新的第三方库非常的生涩，产生了无数个Bug，不过后来也都一一解决了，这种学习体验对于我来说是非常享受的；想方设法优化代码的可读性、美观程度以及面向用户深度，在各大网站阅读和理解资料的过程真的非常煎熬，但又何尝不是一种自我提升呢？

最后，依旧是很感谢《数据采集与预处理实践》这门课程，从各个方面都磨炼了我的能力，比如耐心、创新能力，代码编写流畅程度、速度、可读性、美观程度，当然这也离不开老师的系统性教学与深入浅出的点拨。从所有角度来看，我对该课程的实验过程付出非常巨大，诚然受益匪浅，我明显地可以感受到面对新事物不再那么恐惧、并且很少出现畏难情绪。这门课程让我重新认识了Python语言的魅力，见识到了认知里Python语言不曾出现的新高度，重新定义了我心目中的Python语言。

周一凡：

通过这次的数据采集与预处理实践课的学习与实践后，让我对整个过程更加熟悉，对数据采集与预处理有了更深的见解。从开始的数据爬取到后续的数据分析与预处理，让我了解到一套完整的数据采集与预处理该如何进行，我们组总共是进行了数据采集，数据预处理，数据分析，还有简单的数据可视化，回顾本次课程的学习，我体验了用网络爬虫从一个或若干初始网页的URL开始，获得各个网页上的内容，体验了用Python来进行数据的可视化以及回归分析。并顺利完成了本门课程。

通过这次的实践课复习到网络爬虫的基础知识、编程环境与第三方库、网络爬虫思路与实际操作，异步加载等理论知识。这些是想要爬取到数据的必须前置理论基础。数据的采集是挖掘数据的第一步，当数据量越来越大时，可发掘的有价值的信息也就更多，反应信息也就越加全面。只有更加充分地利用数据化处理平台，才可以保证分析结果的有效性和准确性。

在数据的预处理工作，可以使残缺的数据完整，并将错误的数据纠正、多余的数据去除，进而将所需的数据挑选出来，并且进行数据集成。

总而言之，数据采集与预处理是对数据进行提取，分析的强而有效的方法，通过这门课学会了这项技能，让我之后在工作时可以更加快速且有效率。

侯志昊：

通过本次数据采集与预处理的实践课程，使我对网络数据采集与预处理以及数据分析的过程和编程环境与第三方库有了更深的理解，这次实践课不仅时一次普通的实习，更是对网络爬虫的进阶学习，同时也对之前学过的相关课程的内容进行了再一次的系统性学习，小组在老师的带领下，分别进行了数据采集、数据预处理、数据分析以及相关内容的可视化处理等，顺利完成了本次实践课程。

在实践的过程中，首先要选择适合自己的数据，数据越全面、涵盖面越广泛、分类越清晰、数据的维度越丰富、直观，就会更加有利于小组之后进行的数据采集与预处理工作，当然在爬取到一份数据之后首先要进行数据的预处理工作，不要急于进行数据的分析，还要注意缺失值的处理与异常值的处理，若不进行处理可能会导致最后的分析结果有错误或者不准确的情况。

通过数据的预处理工作，可以使残缺的数据完整，并将错误的数据纠正、多余的数据去除，进而将所需的数据挑选出来，并且进行数据集成。数据预处理的常见方法有数据清洗、数据集成与数据变换。

在数据分析的过程中，将会面对很多的变量，但一般情况下只有少部分的变量与分析的结果有关，所以对数据进行准确的选取与筛选将有助于简化分析的过程，降低分析的难度，也可以提高最终预测结果与分析结果的准确度与可靠性。

刘宇恒：

通过这次的实践课我复习到了网络爬虫的基础知识、编程环境与第三方库、网络爬虫思路与实际操作，异步加载等理论知识。这些是想要爬取到数据的必须前置理论基础。

其次，我学到了MySQL数据库在爬取过程当中的使用，建立一个数据库作为之后爬取到的数据的容器。因为具有一定MySQL的知识基础，所以这方面的操作并没有过于难理解的地方。在对采集到的数据进行处理的时候要注意空值的填充以及异常值的处理。并且初识了sklearn，知道了它主要集成了数据预处理、数据特征选择。

在之后的课程当中又复习到了Pandas模块的基本操作及应用，其主要作用为进行数据分析以及数据处理。Pandas模块的进阶操作主要复习了CSV文件的创建以及读取。在之后的课程中老师带我们学习了jieba分词第三方模块的基本操作。其原理是利用一个中文分词词库，将带分词的内容和分词词库进行比对，通过图结构和动态规划方法找到最大概率词组。最后还学习到了线性回归的相关基础理论知识，通过代码的实现对采集到的数据进行线性回归分析。

这次的实践课不仅精进了有关网络爬虫的基础知识，还学习到了许多其他基于网络爬虫的进阶知识，甚至结合到了别的学科的相关知识。而在进行实践的时候为了解决一些代码上的问题要去自行去搜索学习相关的知识，这次的实践不仅是学习与兴趣的碰撞，更是一次学科之间的知识碰撞。