**温州大学计算机与人工智能学院**

**Python应用综合实践 实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **课程名称：** | Python应用综合实践 | | | | |
| **班 级：** | 21计算机2班 | **姓名：** | 唐登 | **学号：** | 21211357314 |
| **实验地点：** | 5B322 | **日期：** | 2023.10.9 | | |

**一、程序简介**

一种前后端分离的彩票爬取分析可视化python程序。后端基于Flask框架搭建，运用了threading库实现数据爬取功能和处理HTTP请求的功能并发执行。通过datatime库获取当前时间定时去爬取网站API得到彩票开奖数据并且存储到服务器的Mysql库中。服务器根据收到的不同HTTP请求，通过pymysql库编写代码操纵Mysql数据库得到数据并进行不同处理然后将处理结果封装成json格式发回给前端。拥有注册登录顶号功能。目前已经部署到腾讯云服务器上。前端通过tkinter库编写登录注册和功能页面，点击不同功能的按钮通过获取输入框的参数向后端发送HTTP请求，得到数据用jsonpath解包，根据结果来弹窗和用matplotlib库绘图呈现在功能页面的中央。

**二、运行效果、运行关键步骤截图**

**1.抓取彩票数据功能运行**

运行python后端程序，后端程序启动后会自动开始监控网站，在彩票开奖的时间后定时爬取相应的API得到数据存储到数据库当中。



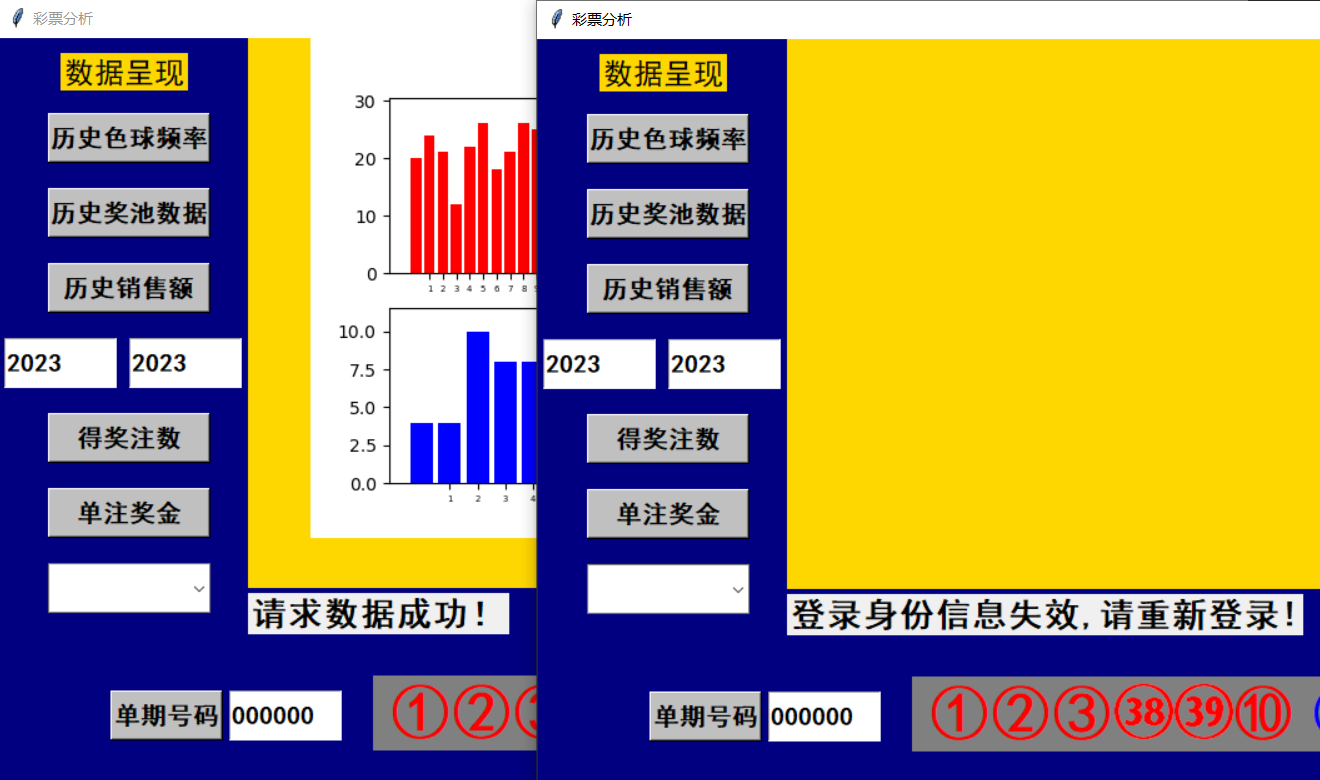
如图抓取到了2013年到2023年的双色球数据，按照期号-红球-蓝球-销售量-奖池数据-中奖注数-单注奖金的格式存储到mysql数据库当中。

**2.登录和注册**

运行前端前端程序，能通过向后端发送http请求的方式完成注册登录顶号等功能，顶号后无法正常向服务器请求数据。



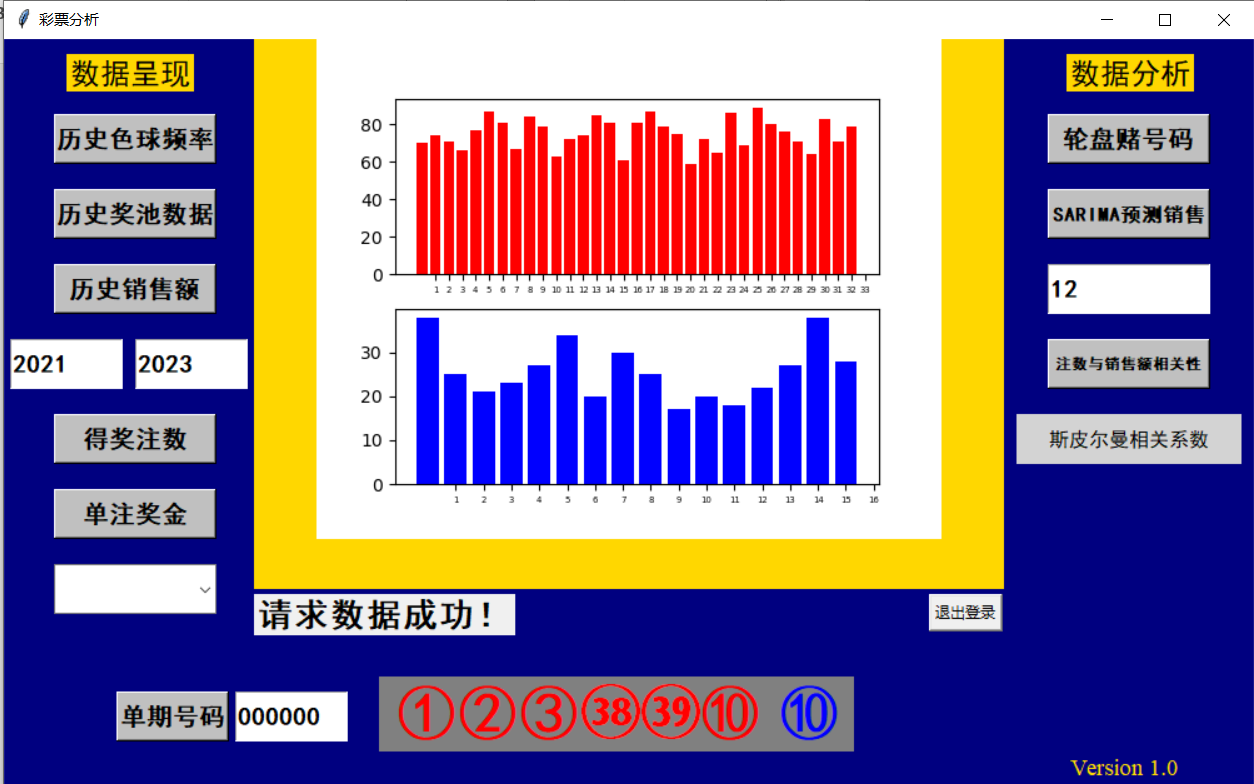




如图呈现了账号注册，登录，以及顶号的运行效果

**3.历史开奖号码频率的呈现**

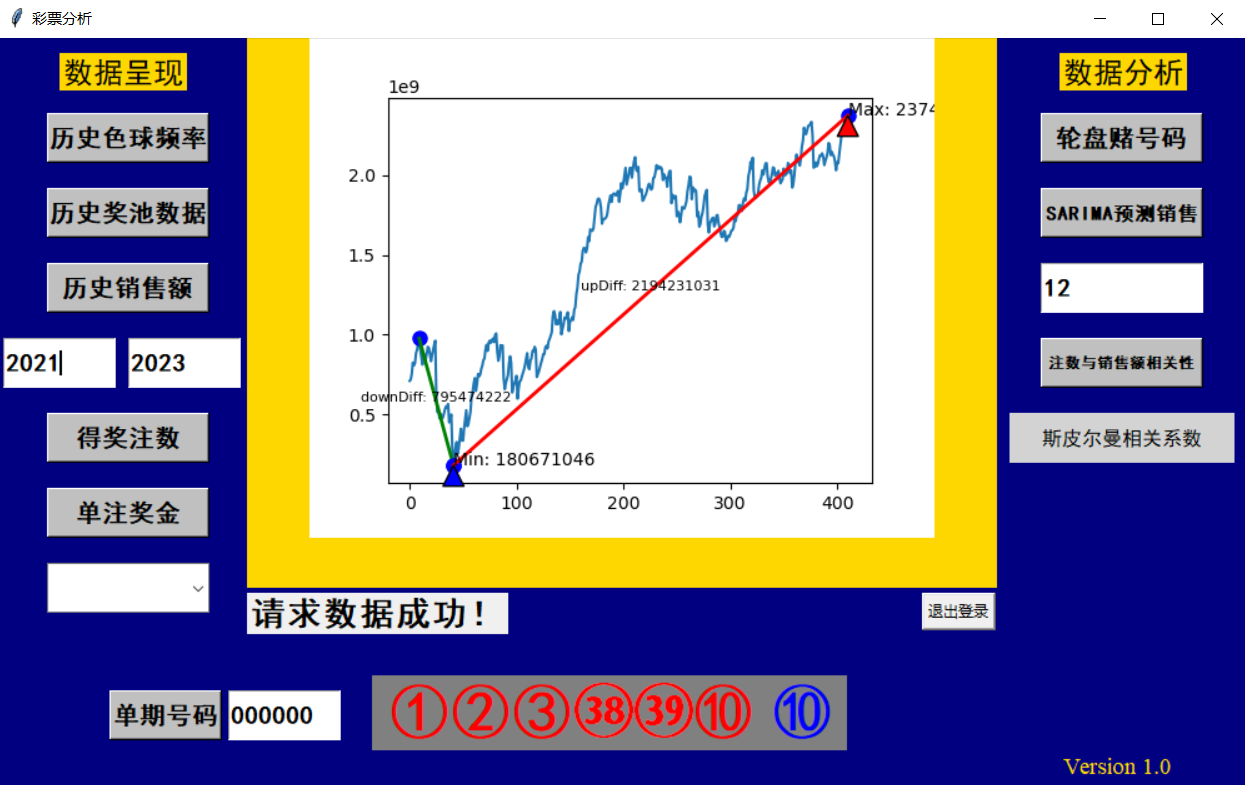
在输入框中输入需要查询的年份，就可以得到相应年份之间开奖的彩票号码的频率，上半部分显示红球下半部分显示蓝球。



如图呈现的就是2021年到2023年的红蓝各种号码开奖号码的频率数据。以柱状图得以呈现。

**4.历史奖池数据呈现**

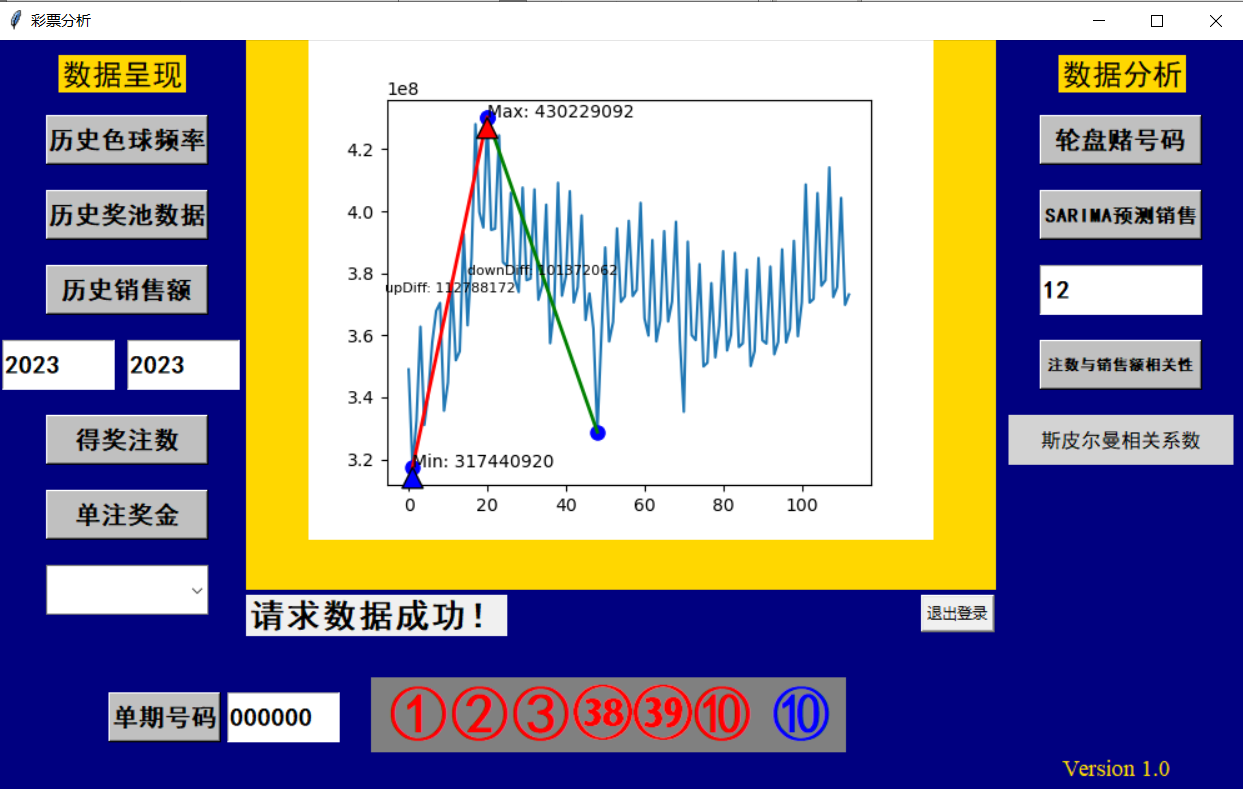
获取输入框中输入的年份之间的历史奖池数据，会以折线图的形式呈现在面前，还会画出沿着时间线进行的最大涨幅和最大回撤的相应折现，在最高点和最低点也会进行相应的描点，红点为最大值，绿点为最低值。



如图得到了2021年到2023年之间的奖池数据的折线图，可以通过折线图初步观察到最大涨幅是2194231031元，总体趋势是呈现递增趋势。

**5.历史销售额**

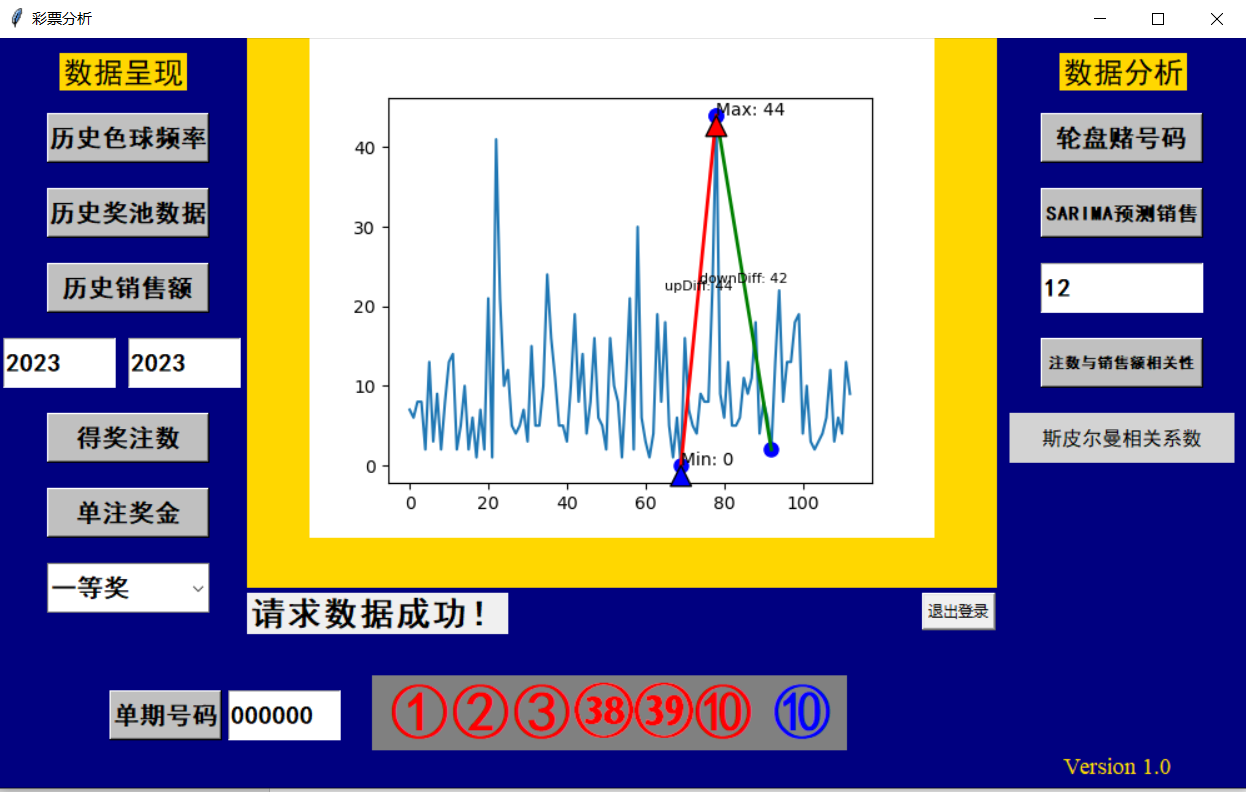
与历史奖池数据相似，输入相应的时间区间可以得到相应的数据折线图，并且进行机械基本的数据分析绘图。

****

如图可以看出，历史销售额非常的具有规律性，可以后续进行一些预测分析。

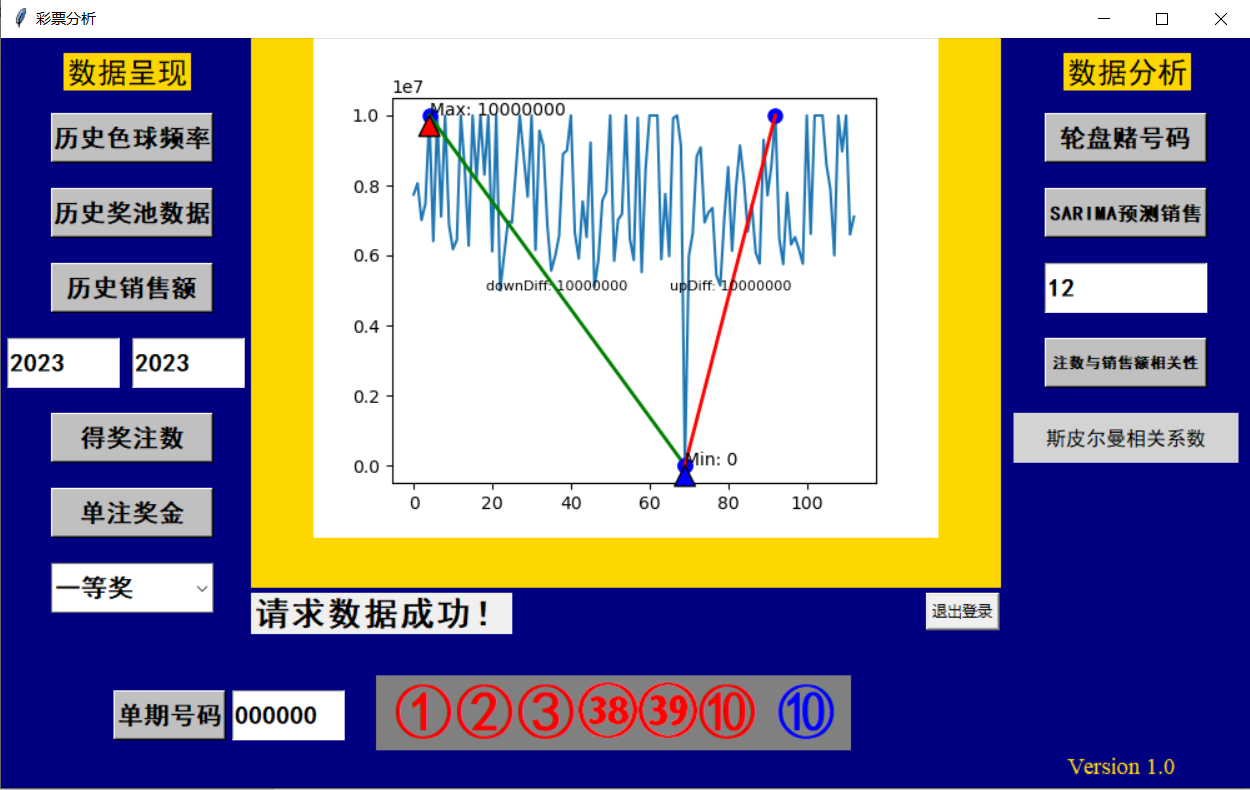
**6.得奖注数数据呈现**

输入区间可以查询到相应时间区间的得奖注数数据并且进行简单的分析

**** 如图得到了2023年中一等奖的得奖注数的相应数据。

**6.单注奖金数据呈现**

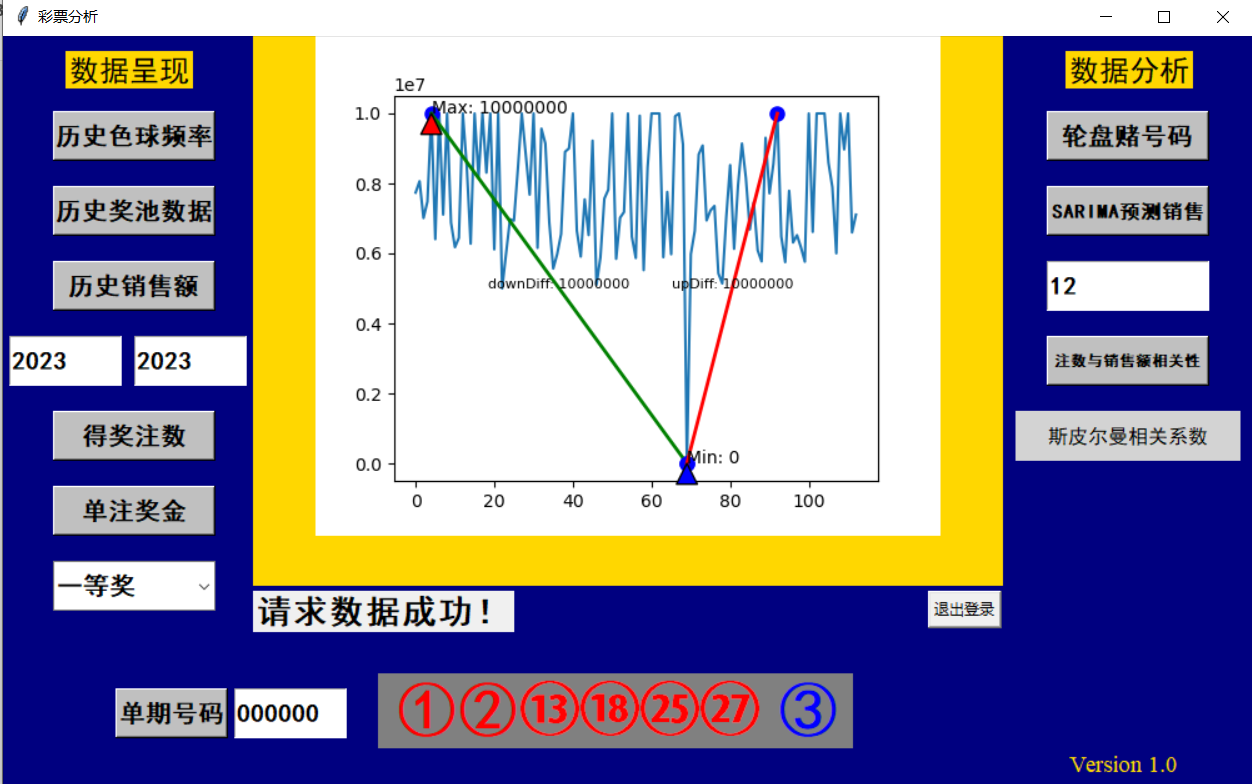
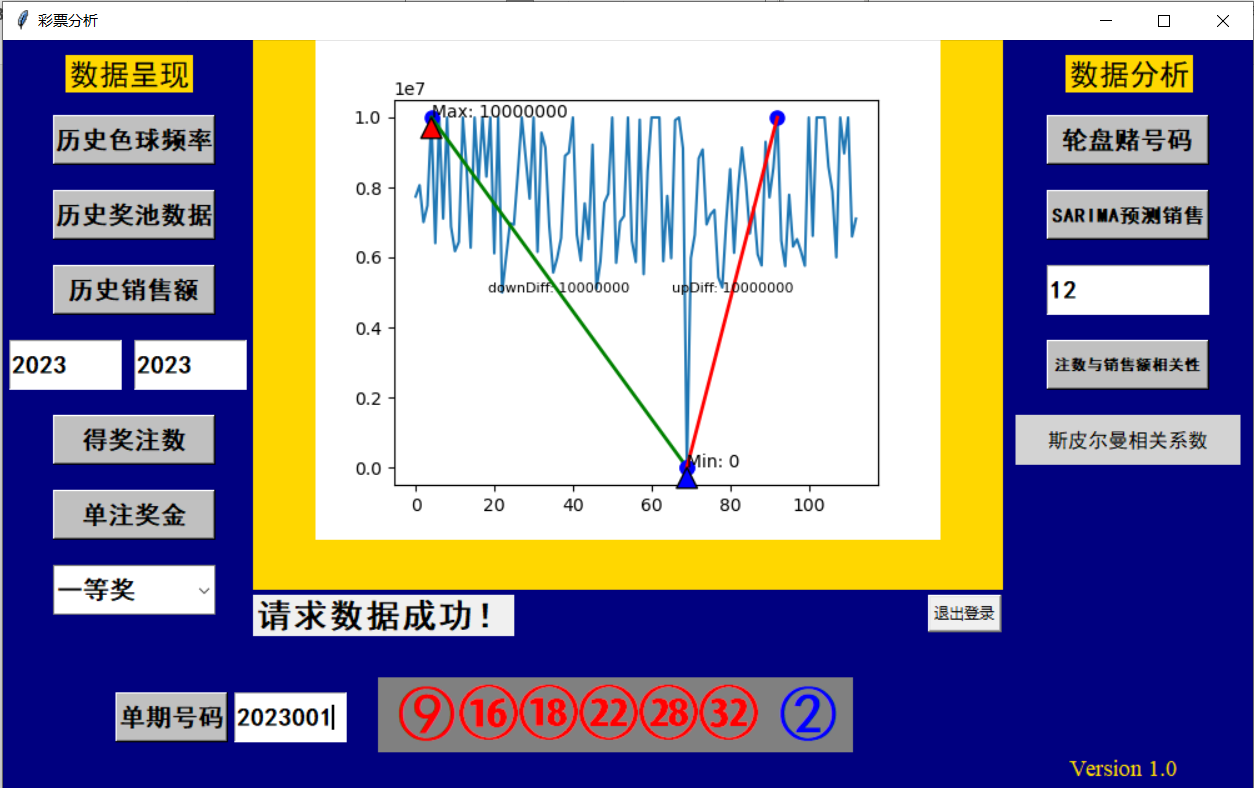
输入区间可以查询到相应时间区间的单注奖金的数据并且进行简单的分析



如图得到了2023年中一等奖的单注奖金的相应数据。

**7.通过期号查询中奖号码**

输入具体的期号可以查询相关的中奖号码，如果输入000000期号，就会查询到最新一期的彩票中奖号码。



如图，通过输入准确期号2023001查询到相应期号的中奖号码，也通过000000查询到了最新一期的开奖号码。

**8.轮盘赌号码生成**

彩票开奖是一个随机事件，当取得样本足够大的时候，各个号码的开奖频率应当是趋近于相同的，程序内部分析历史开奖频次，会赋予每个号码不同的被选中概率，历史被选中频率低的，被随机出来的概率会高反之会低，理论上来说所取的样本数目越多，轮盘赌的效果会好。

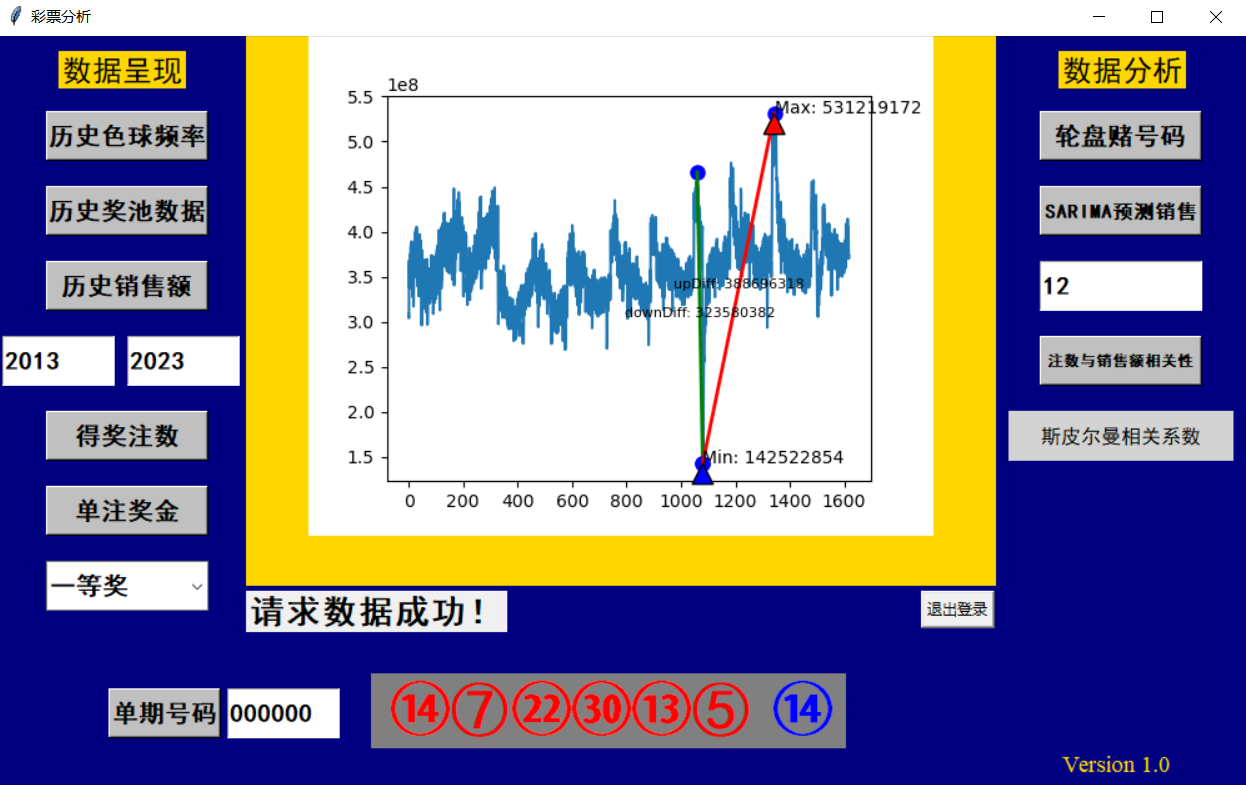


****

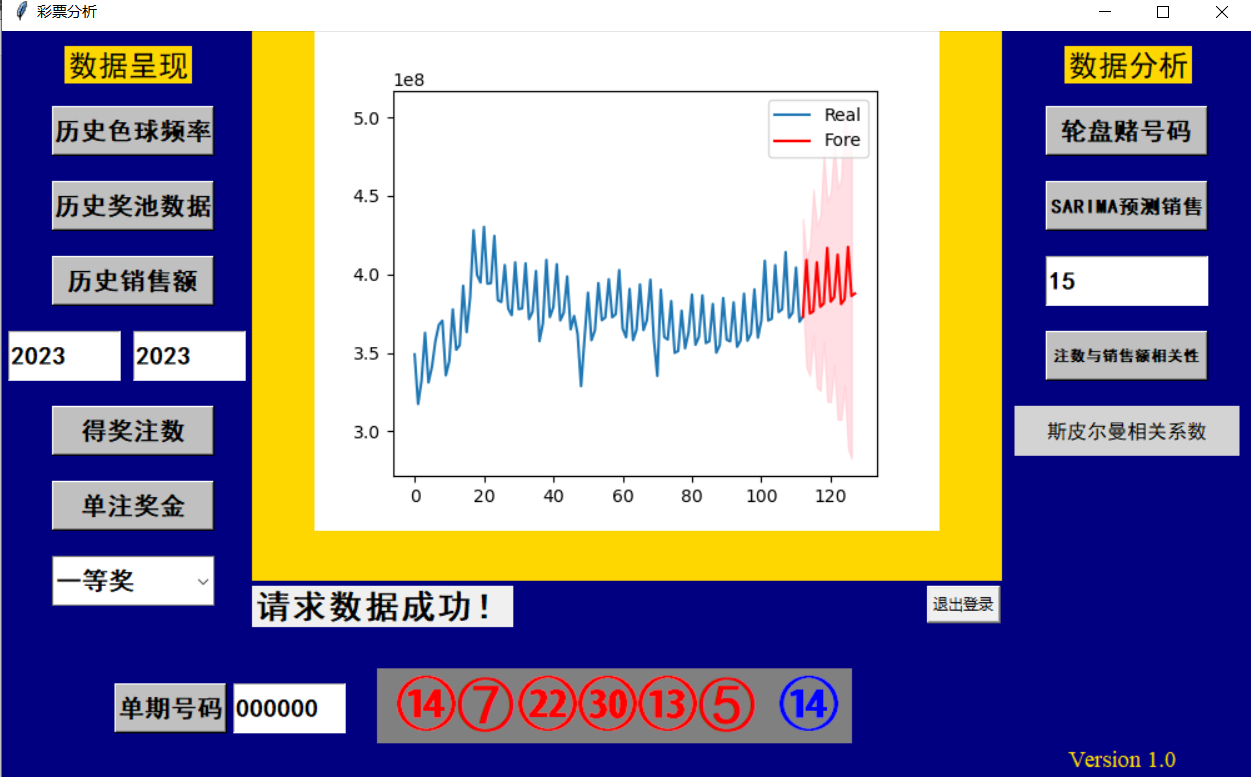
如图根据2013到2023年之间的数据来生成了两期轮盘赌的号码。

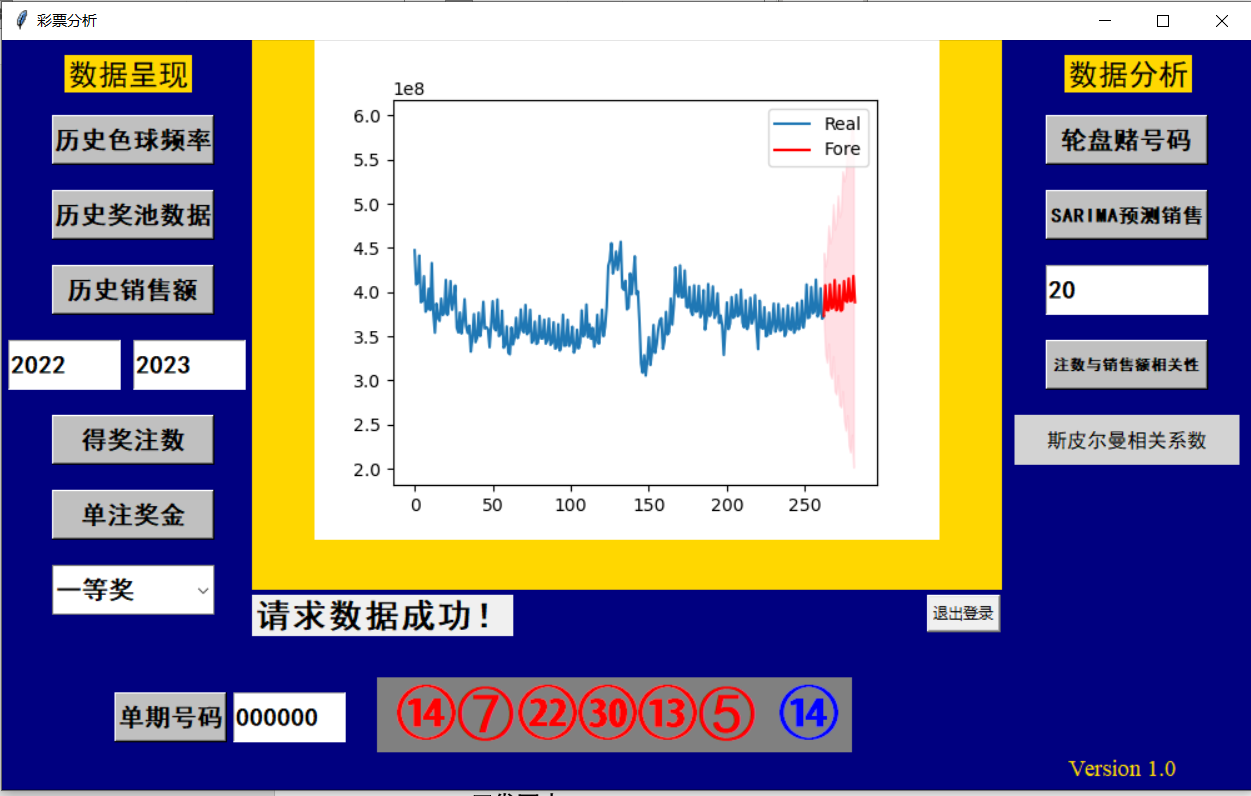
**9.SARIMA回归模型预测历史销售额**

可以通过呈现销售额的历史数据可以发现，销售额的曲线非常有规律甚至呈现出某种周期性，如下图所示。

****

这种数据非常适合用SARIMA进行预测分析，能得到相对可靠的预测结果，在右方输入框中输入需要预测的天数，左侧两个框提供相应的训练数据的区间就可以得到相应的预测结果和相应的置信区间

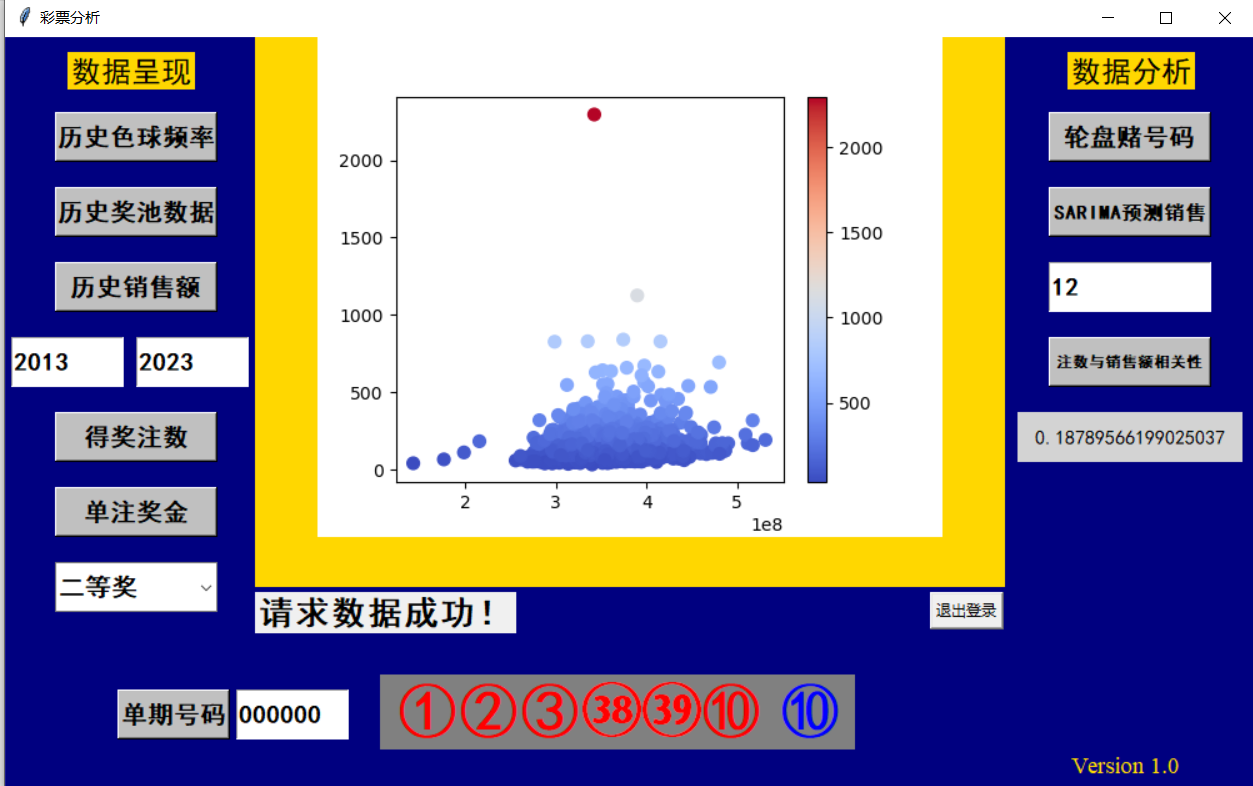
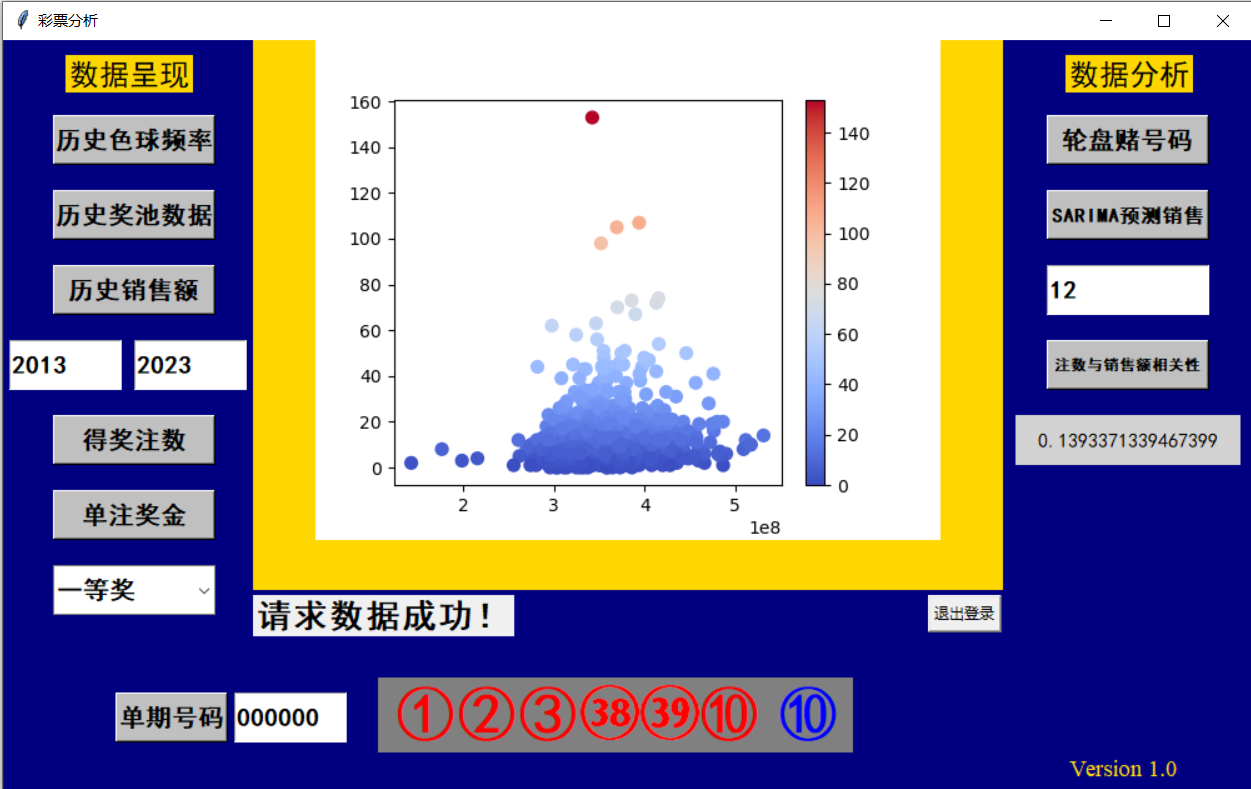
****

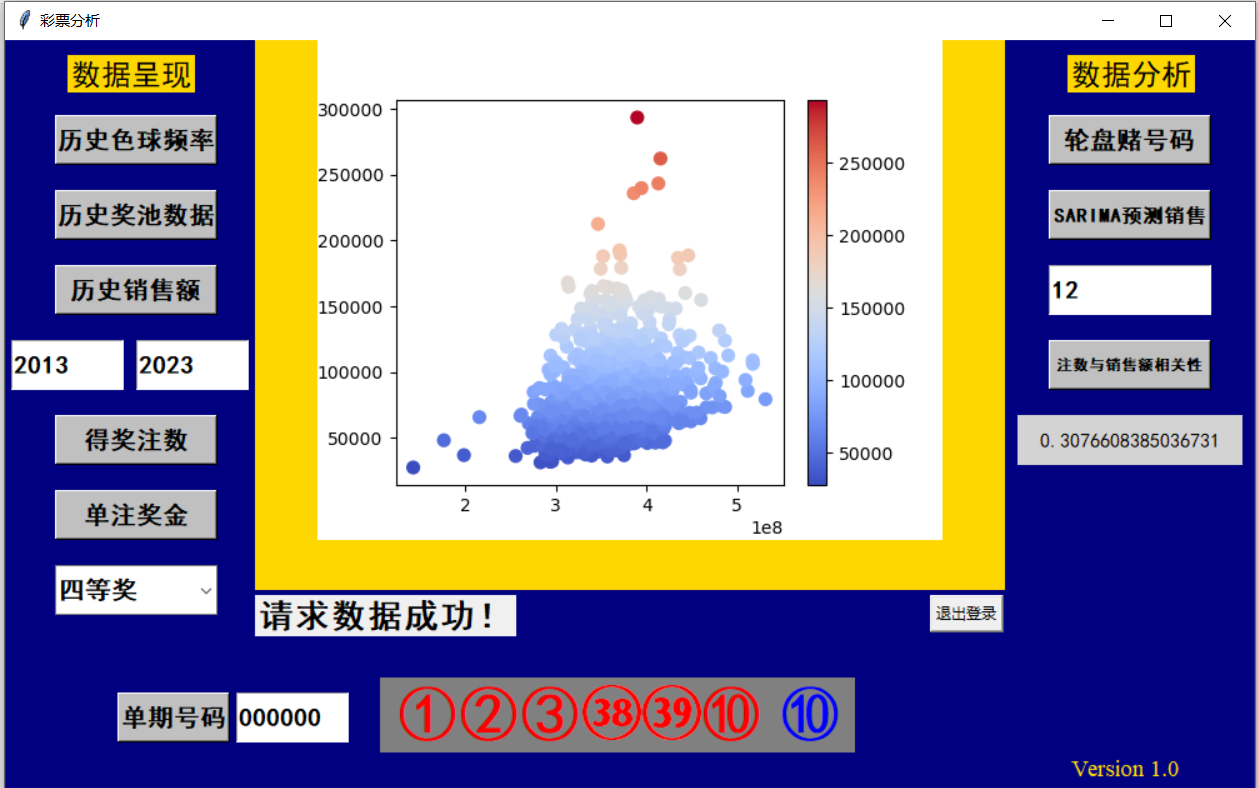
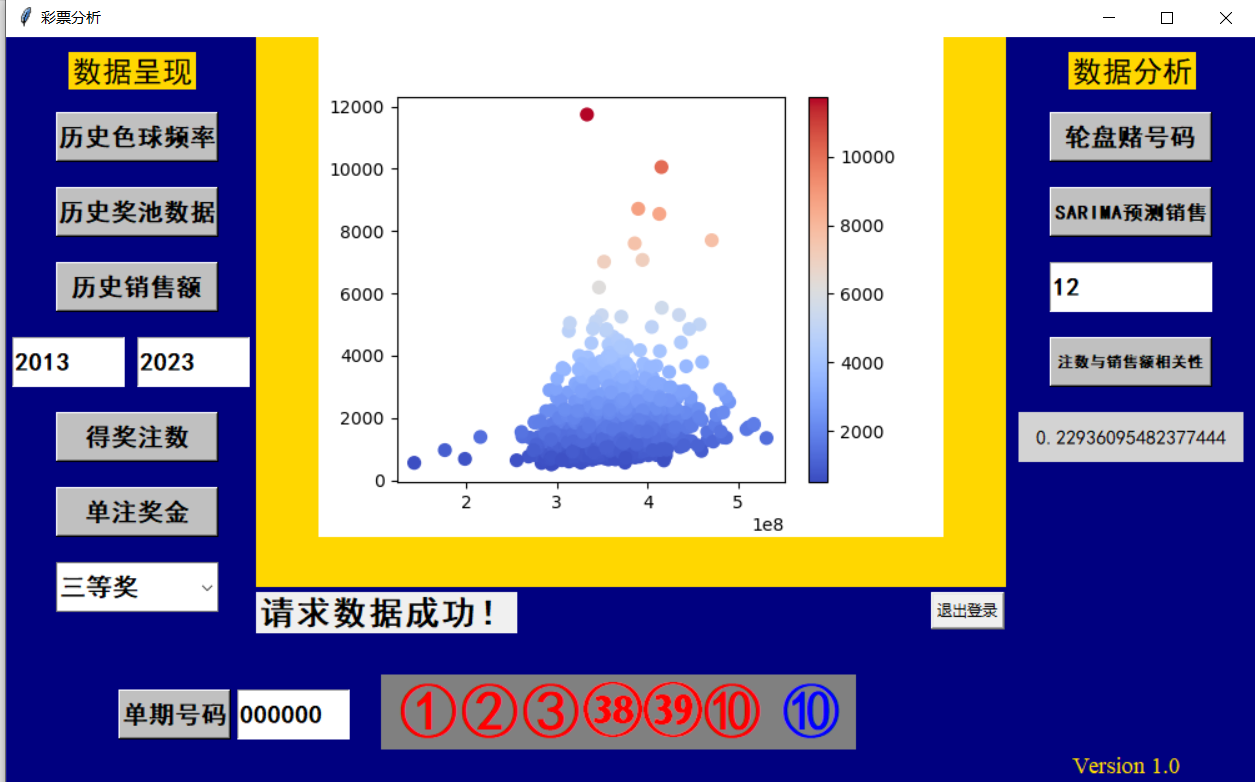
****

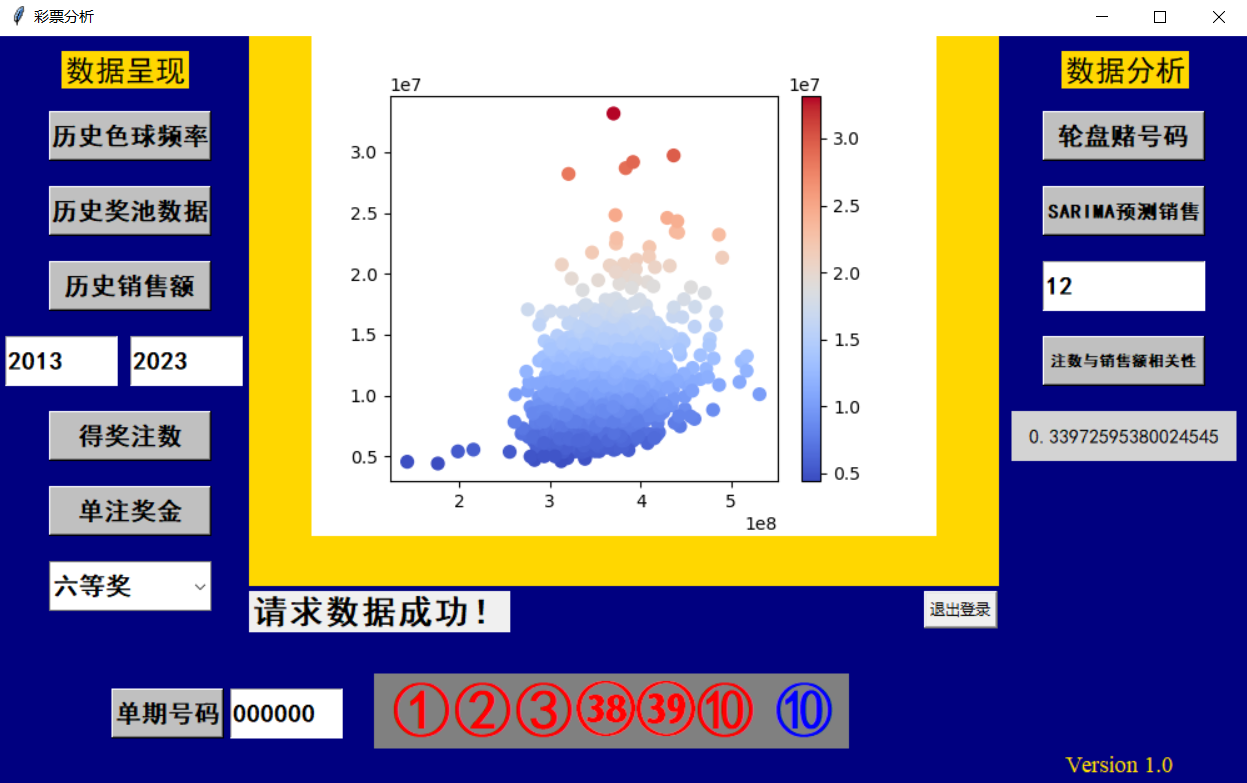
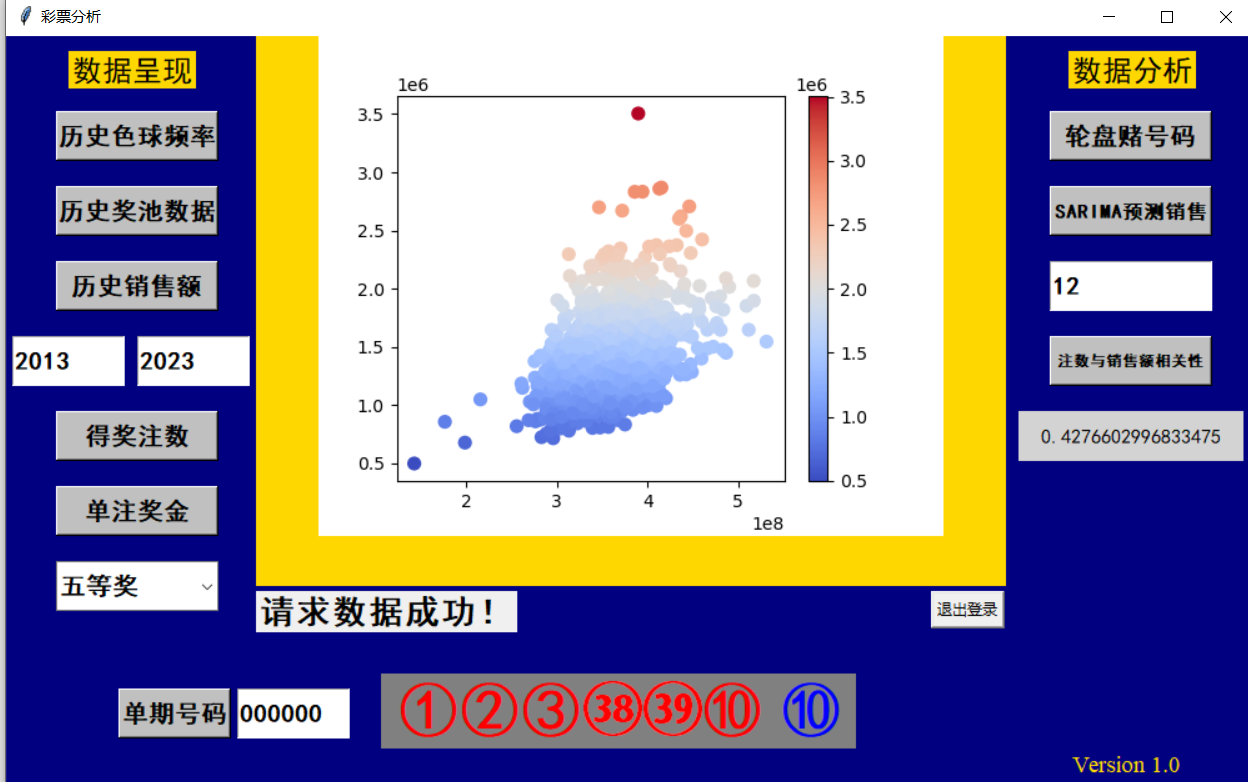
如图给出了两份预测的结果图，红色为预测趋势，淡红色为置信区间。

**10.销售额和中奖注数的散点热力图呈现和相关性求解**

比较好奇彩票的中奖注数和销售额是否有着相关性，随着销售额的增加彩票的中奖注数会不会也会相应的增加呢。通过查阅资料，斯皮尔曼相关系数非常适合这种非线性的数据，斯皮尔曼的数据区间在-1到1之间。趋近于-1是负相关，趋近于1是正相关。我们接下来分析一下2013年到2023年间所有开奖数据的相关性情况。







通过观察分析六个奖项的相关性，发现整体呈现正相关，而且彩票的等级越低，斯皮尔曼系数越高正相关性越高，最高来到5等奖的0.42。但是六等奖无论数据样本多少都趋于0.3，是因为六等奖每一期发放量都巨大，所以数据趋于稳定在0.3附近。

**三、开发要点**

**1. 爬虫API分析**

首先我们进入中国福利彩票官网，初步打算用request爬取网页html，然后通过xpath定位元素爬取相关信息。

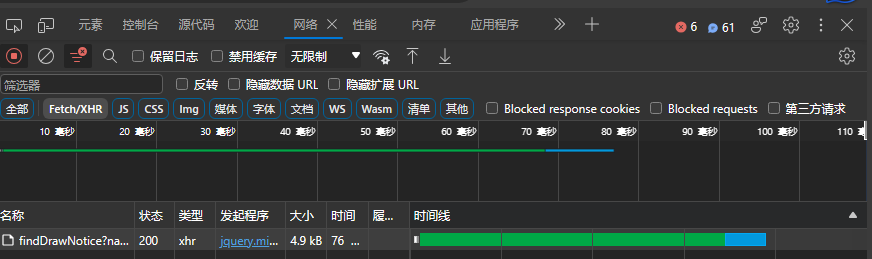




但是通过python解析xpath地址后获取到的结果是一个”-“字符，说明这个网页是通过动态加载出来的，不能直接通过request爬取数据，但是要爬也是可以爬的，可以用selenium库模拟访问浏览器来获取页面信息爬取到动态更新的内容，但是这样做只能获取到一组内容，不能获取到历史数据。所以我们在看看其他网页有没有相关数据。



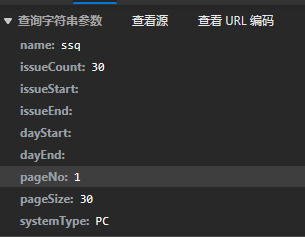
在往期开奖这里我们发现了大量的数据，这种需要更新，数据量大的网页，极有可能是服务器把数据封装成包发过来，页面根据数据包内容来做信息的填充，所以接下来我们尝试去找一找数据的API。



打开F12,查询了最近30期彩票的开奖信息，发现多了一个数据包叫findDrawNotice。



我们打开包预览了一下，发现就是我们需要的数据，接下来只需要将请求参数搞定就能获取到我们想要的数据了。



Name和ststemType两个参数不用管过多后者应该是请求设备类型，到时候直接就这两个发过去。issueCount应该是期号数目，结合我们查询了最近30期数据，所以填写多少这个就可以获取到最近多少期的数据。初步判断issuseStart和issueEnd是查询期号起点和终点之间的数据。而dayStart和dayEnd应该是查询日期起始之间的数据。但是实际操作发现issueCount不管填多大，都只能获得30组数据。后面结合页面再分析，应该是被pageSize限制了，最终发来包里数据的总数多少还是取决于pageSize的大小。甚至issueCount都可以不用给值，发送多少pageSize就会得到最近多少组数据。结合我们的需求，我们最终只需要给定这几个参数就能得到数据。

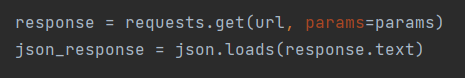


No是一个变量，他取多少，就能获得到最近多少组的数据。

**API："http://www.cwl.gov.cn/cwl\_admin/front/cwlkj/search/kjxx/findDrawNotice"**

**2.数据提取**

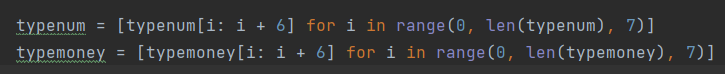
得到数据后先用json.loads方法处理一下，作用是把JSON格式的字符串转换成python的数据结构。

****

接下来就是用jsonpath把红球蓝球编号等数据全部提取出来。



然后用列表推导式，每隔7项提取前6个数据，为什么要这样处理呢，因为我们发现发来的数据每隔6个有用数据就会有一个空字符串，这里猜测是为了区分每一组数据，所以我们要把他给处理掉。







然后处理红球数据，每项用逗号作为分界组成列表，也是因为发来的数据一组数据是按照字符串发送来的，我们要把它变成列表。



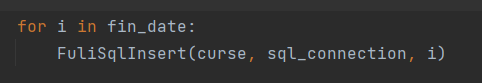


然后整合数据用zip同索引的数据结合在一起用tuple函数变成元组方便后续存储数据库。第二个列表推导式是对每一组数据进一步处理，因为有些数据存进去还是列表的形式，我们希望有列表的数据把列表解掉。



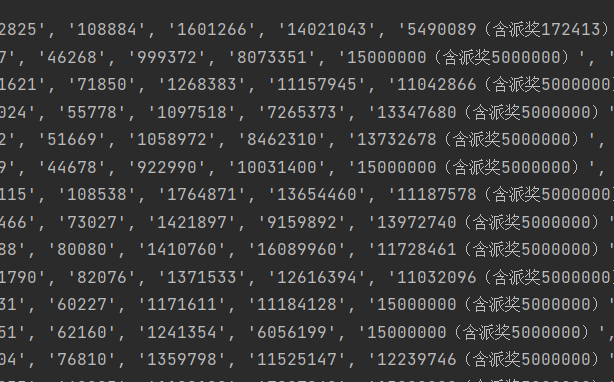


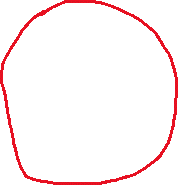
随后我们通过自编的FuliSqlInsert函数将需要存储的数据存储到数据库当中。



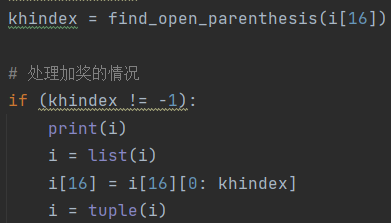
原本是将这样初步处理的数据直接存储到数据库当中,但是发现数目总是对不上发回数据包里彩票数据的数目，总是有组数被漏掉，让我排查了好久，最终发现彩票还有加奖的说法，在数据库设计中这一行是int类型，但是有加奖的话就变成了字符串，就会插入数据失败，所以我们要做一些处理再存入数据。



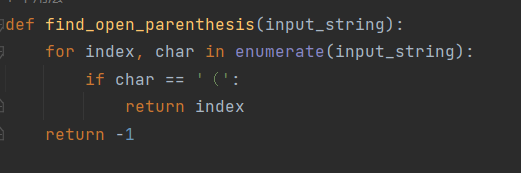




我的处理方式是找第一个左括号的位置，然后把左括号以及后面的数据全部删去，这样就剩下了纯数字也就不会有错误了。



Find\_open\_parenthsis是自编的找括号函数

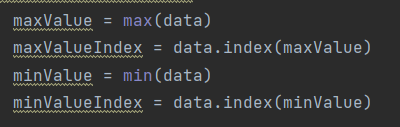


但是这样处理完了以后发现，数据依然有相当数量的缺失，而且缺的还是那几组，反复检查代码确认逻辑无误，排查了好久，最终发现是括号惹的祸，发来的括号是中文括号，而我输入的是一直是英文括号，最终导致了排查错误。最终向数据库插入相关数据，如果插入不成功就回滚。

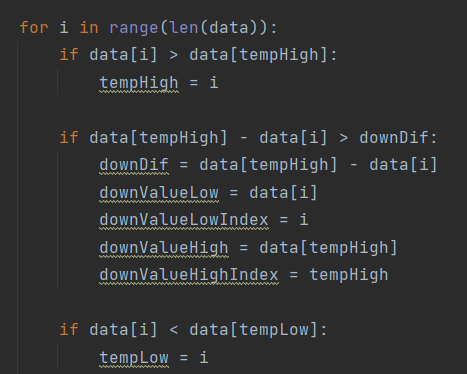


**3.最大回撤和涨幅计算**

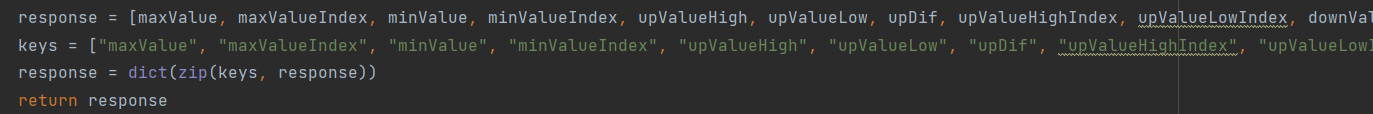
彩票数据中的销售额，奖池数据，注数等会随着时间波动是一种时序数据，所以计算随着时间线的最大涨幅和最大回撤是有价值的，可以观察到他的最大波动情况。所以我写了一个计算一个数据最大涨幅和回撤的函数。输入参数是一个列表，输出参数是最大值，最小值，最大回撤值，最大涨幅值以及他们的相应的下标位置。最大值最小值好处理，直接调用内置max,min函数就能完成计算。



这里介绍一下计算最大涨幅的方法，最大回撤类似，从左往右遍历整个列表，设置一个局部最大值变量，当碰到比局部最大值大的值时，刷新最大值的值，每个数据都和局部最大值做差，如果大于当前最大涨幅，就刷新相关设置的局部变量，注意记录值一定要是下标而不是具体值，有了下标可以得到值，但是有值后续回溯下标，如果有多个相同的值回溯出来的下标就会有问题会错误。



最后封装成一个字典返回



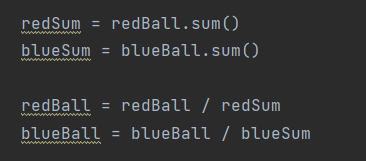
**4.轮盘赌生成号码**

彩票开奖是独立重复事件，每一期号码每一个号码的开奖概率是相同的。当样本足够大的时候，每一个号码出现的频率应该是相同的，我们根据统计每一个号码出现的频率，频率小的应当出现概率会更大，根据这个用轮盘赌的方式设置概率最终来生成一组号码，当然生成号码仅供娱乐，预测是做不到的。

通过自编函数获取到数据，提取红球和蓝球的开奖数据，将他每个值取倒数，这样大小关系就反过来了。



随后对这些数据进行归一化，就可以进行轮盘赌了。



以红球为例，轮盘赌的过程就是生成一个0-1的随机数，然后遍历累加，直到大于这个随机数为止的那个下标作为选中值，所以占比越大的数被选中概率也就越高。

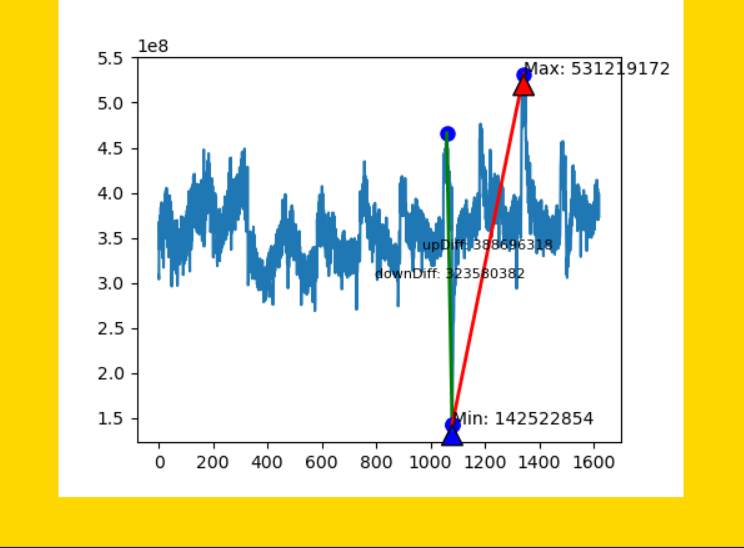


最后用json.dumps方法将结果封装成json格式返回，后面还有代码会将它发回给请求的客户端。



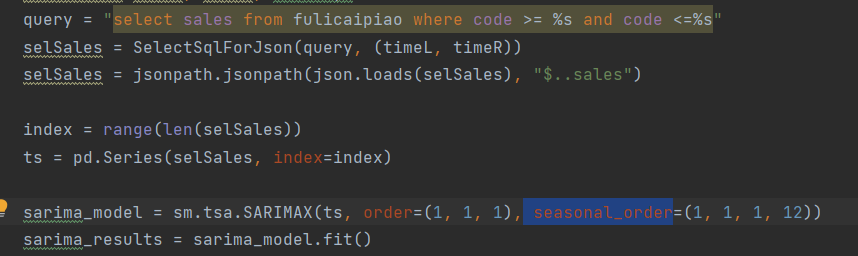
**5.SARIMA预测销售数据**

根据销售数据的呈现情况，我们可以发现数据整体十分有稳定，似乎有着周期性，看着有规律可循，这种比较稳定似乎存在某种规律的数据非常适合用SARIMA季节回归预测模型来做。



这种预测类模型在python中有做好的库statsmodels.api。

导入相关数据，填入相关的参数，seasonal\_order 参数用于定义季节性差分的阶数和季节周期。建立相关的模型调用fit函数进行拟合。



输入我们需要预测的天数，用训练好的模型进行预测计算，随后提取出预测的值和相应的置信区间的值，封装JSON格式发回，整体预测就完成了。



**6.斯皮尔曼相关系数计算**

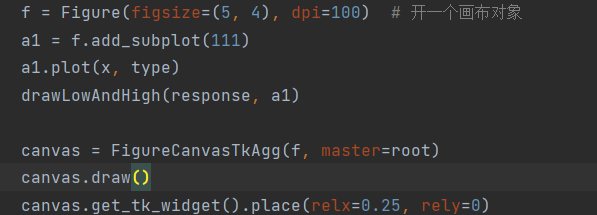
需要计算斯皮尔曼相关系数来反应两个值的相关度，所以写了一个输入量是时间起点与终点，还有中奖的等级情况。输出是斯皮尔曼相关系数。在scipy.stats库中有写好的计算斯皮尔曼系数的函数，所以我们这里只需要把数据切分好递交给函数就能算出相关的系数，最后封装成JSON格式发回



**7.各种绘图的呈现**

本次作业主要涉及到了柱状图，折线图和散点热力图。用matplotlib库绘制，呈现在tkinter界面上。首先说明如何将matplotlib的图呈现在TK的界面上。

其实只要用FigureCanvasTkAgg方法，将matplot创建的图形对象f传进去生成一个新的canvas对象，设置横纵坐标，再将他呈现在TK界面上就行了。

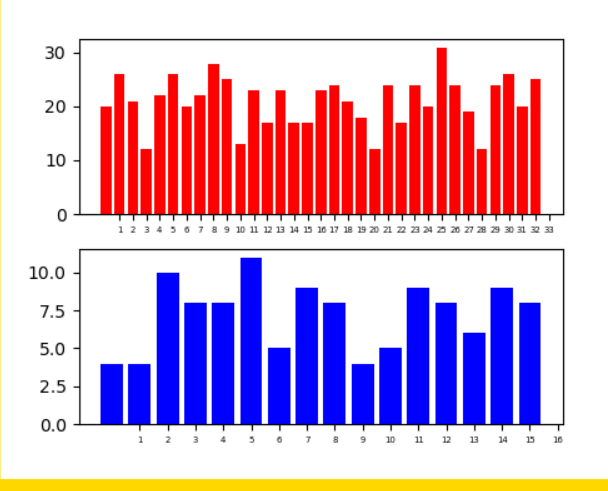


柱状图：

以红球为例，首先创建一个1-33的值的列表作为下标使用。Figsize是画布的大小dpi是像素密度。f.add\_subplot(211)其中211的意思是按照两行一列来划分，当前选中的是第一个区域，也就是第一行第一列，下面蓝球就是212第二个区域也就是第二行第一列。

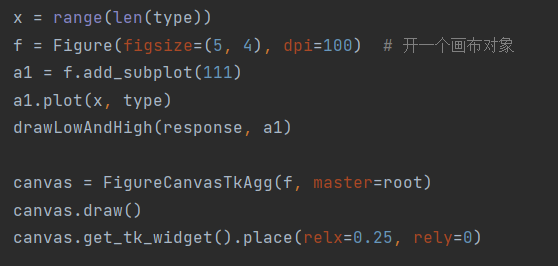
随后用bar方法输入下标列表和值的列表再设置颜色就能画出来图形了。

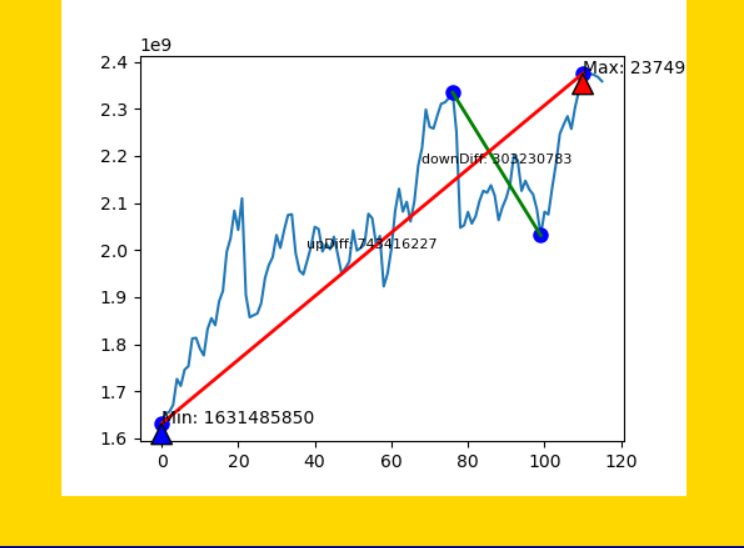




折线图：

折线图比较简单，只需要给定横坐标和纵坐标的列表值，直接用plot就能绘制出来了。

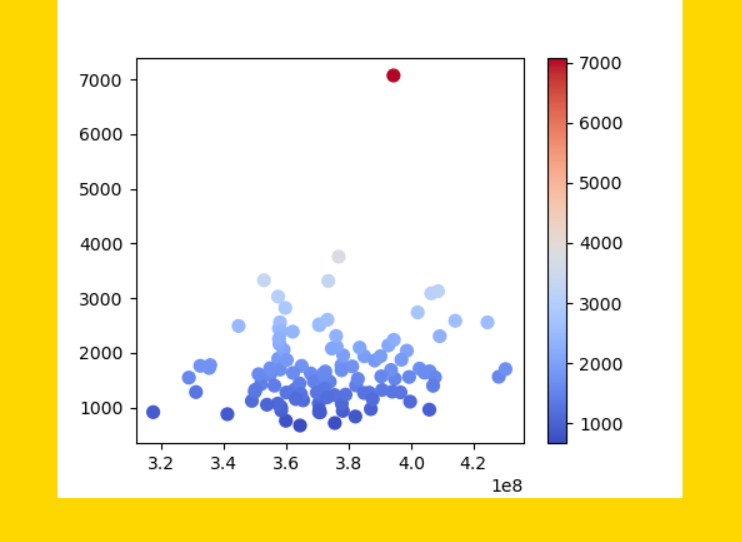




散点热力图：

主要用了scatter方法来绘制点，其中cmap是指定颜色的变化情况，我选择的是coolwarm,就是值相对较小的点化的是蓝色冷色调，值比较高的点绘制的是红色暖色调，colorbar是绘制边上的颜色参考柱子用的。





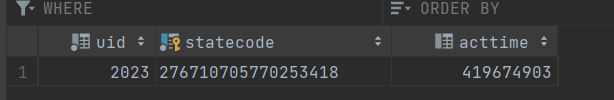
**7.flask搭建后端**

后端是用flask框架搭建编写的，这是一个轻量级框架用起来比较方便功能比较简洁。用[@app.rounte](mailto:我们需要先用@app.rounte)给定一个注解,比如/regist就是请求的注解，随后用request.values.get就可以提取到请求头中的相关信息了，当然用post会更加的安全。但是对于初学者来说用get可以用浏览器写url发送数据就能观察后端程序是否运行正常，用post还得写个发送程序调试起来不方便。

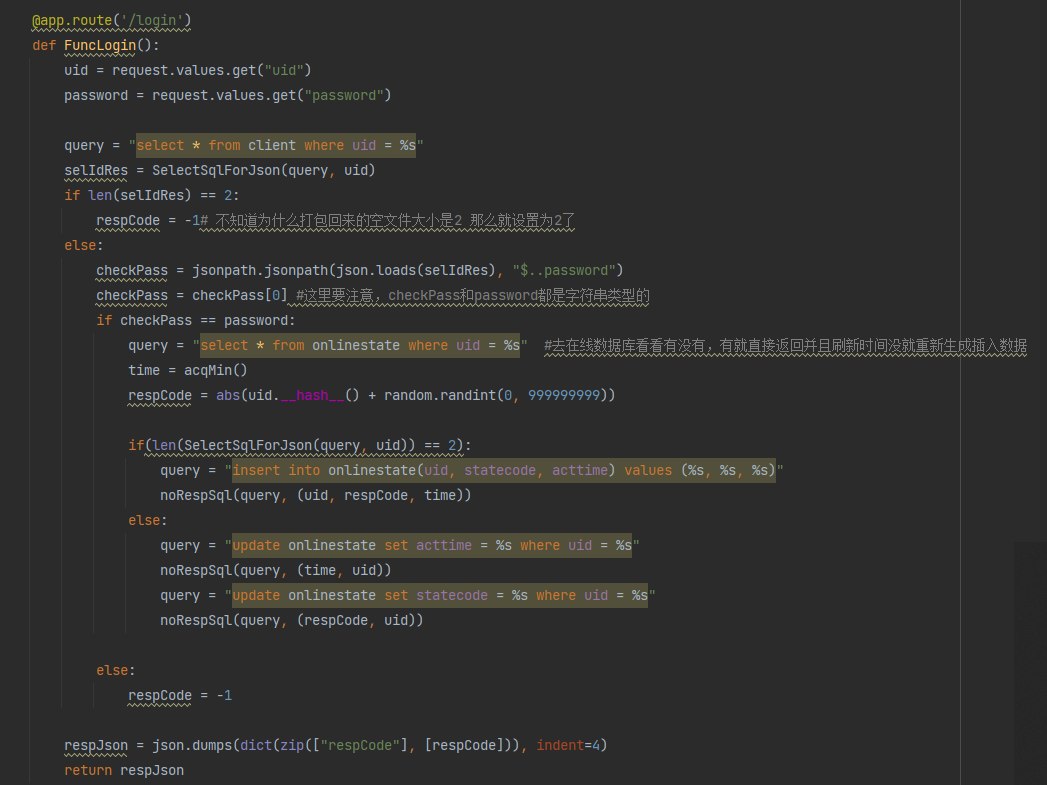


在注册模块中，就是将客户端发来的注册号码去数据库中搜一遍，如果搜到了，那么返回的列表值就是一个大于2的数字，我也不知道为什么查不到返回的空列表为什么是2，但是根据就这就能判断这个账号有没有注册，如果注册了就返回一个respCode为-1的JSON包，前端收到后会检车respCode，如果是-1就会弹窗注册失败，否则就会弹窗注册成功。

接下来讲一讲登录，这里设计了一个登录状态的数据库，其中有uid和statecode还有acttime，statecode是一个登录的状态校验码，acttime是转化成分钟形式的时间，比如0000年1月1日1时0分，那就是60。这样方便后续计算。

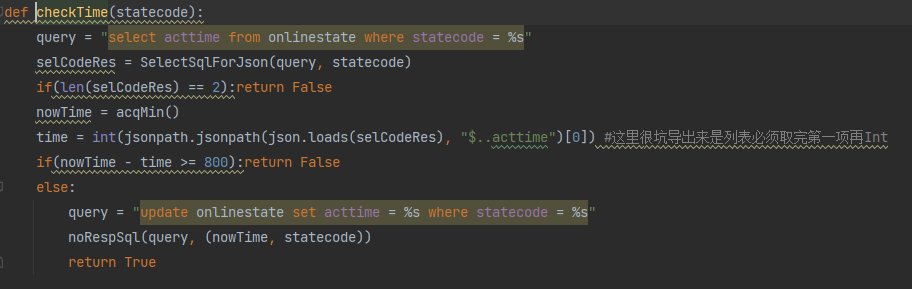


用户在登录后会检查状态数据库，如果值存在，就将statecode更新,acttime也更新，然后将statecode发回给客户端。如果没有值就当场生成一组数据插入，随后将statecode发回。Statecode用hash函数生成然后加一个随机值扰动一下。



这个statecode的作用是用来校验身份的，客户端登录成功后手上就握有一个statecode，他后续的所有数据请求，都必须在请求头中加入statecode，服务器接收到请求的第一件事就是验证statecode

，去数据库检查这个statecode是否存在，如果不存在就直接返回一个respCode为-1的JSON包，如果存在的话再验证一下acttime,如果超过设置的时间，就是登录状态过期。也返回-1.如果客户端接收到这个包就会弹出身份信息过期的提示。这里编写了一个checkTime函数专门用于校验身份状态。



客户端这里也写了一个函数来专门校验返回包的信息，如果发现有respCode = -1就会返回-1下面接收到直接跳出函数不绘图。



**四、第三方库使用情况**

1. requests

用于请求API，爬取相关数据，也要用与客户端向服务端发送http请求。

2. random

用于获取随机数，主要用于轮盘赌

3. threading

多线程库，用于实现服务器一边检测API数据一边相应HTTP请求，边爬边响应。

4. statsmodels.api

调用其中的函数建立SARIMA模型做回归预测

5. flask

用于搭建python后端程序

6. json

主要用了loads和dumps方法，前者将JSON格式转化为python数据结构，后者将字典封装成JSON格式。

7. datetime

获取当前时间，用于API监测中使用，实现定时爬取

8. pymysql

用于操纵Mysql数据库，增删改查

9. jsonpath

定位查找数据，将json loads下来的数据查找需要key下的所有value

10. numpy

是一个很强大的库，这里简单的运用了一下，用到了求和等简单函数。

11.scipy.stats

计算得到斯皮尔曼相关系数

12. tkinter

编写前端的界面，太老了本来想用PYQT但是报错打不开

13. matplotlib

绘制折线图，散点图，柱状图等可视化图形