

景德镇陶瓷大学科技艺术学院

本科生毕业设计（论文）

题目：基于 NumPy 的股市数据分析工具

学 号：	314040300122
姓 名：	朱敏
院（系）：	科技艺术学院
专 业：	计算机科学与技术
完成日期：	2018 年 06 月 06 日
指导老师：	徐远纯

分类号_____

学校代码_____

UDC _____ 密级 _____

学 号 314040300122

景德镇陶瓷大学科技艺术学院

本科生毕业设计（论文）

题目：基于 NumPy 的股市数据分析工具

朱敏

指导老师：徐远纯 教授 科技艺术学院 工程系

申请学位级别：工学学士 专业名称 计算机科学与技术

论文提交日期2018 年 06 月 06 日

论文答辩日期2018 年 06 月 08 日

学位授予单位：景德镇陶瓷大学科技艺术学院

答辩委员会主席：

论文评阅人：

2018 年 6 月 4 日

摘 要

现在的股市股票数目众多，根据现有的不完全统计，中国股市集中在深沪两市的股票已经有三千多只，每日股市的数据更新可以说是非常频繁，那么股市数据的复杂程度可见一斑。面对这么庞大的数据量，仅仅凭人工力量来操作加以分析将耗费巨大的人力、财力、物力、时间，这样显然是行不通的。为了解决股市数据庞大、多样、难懂的问题，我们需要开发能够针对股市数据进行分析的工具。

本课题开发的股市数据分析工具能够提供用户搜索股票数据和查看数据分析的功能。数据分析结合了数据可视化技术，使用 Python 编程语言来编写程序去实现股市数据的下载显示、股市数据分析、用户动态日志的显示和清空。NumPy 数值计算库的使用使得数据分析更方便。与此同时，Python 中的其它工具包也让系统的实现趋于便利。

关键词：股市数据分析 数据可视化 Python NumPy

Abstract

There are a large number of stocks in the stock market. According to the incomplete statistics available, there are already more than 3,000 Chinese stocks focused on Shenzhen and Shanghai stocks. The daily stock market data update can be said to be very frequent, and the complexity of the stock market data. It is evident. Faced with such a huge amount of data, simply manipulating and analyzing it with artificial forces will consume enormous manpower, financial resources, material resources, and time. This obviously does not work. In order to solve the huge, diverse and difficult problems of stock market data, we need to develop tools that can analyze the stock market data.

The stock market data analysis tool developed in this topic can provide users with the ability to search for stock data and view data analysis. Data analysis combined with data visualization technology, use Python programming language to write programs to achieve the download and display of stock market data, stock market data analysis, user dynamic log display and emptying. The NumPy numerical calculation library provides great convenience for data analysis. At the same time, other toolkits in Python make the system easier to implement.

Keywords: Stock Market Data Analysis Data Visualization Python
NumPy

目 录

摘 要	I
Abstract.....	II
第一章 绪论	1
1.1 课题研究背景及意义	1
1.2 研究目的	2
1.3 论文的结构安排	2
第二章 相关技术介绍	3
2.1 数据可视化技术简介	3
2.2 Python 编程语言及相关介绍	3
2.2.1 Python 编程语言	3
2.2.2 NumPy 数值计算库	3
2.2.3 Tushare 财经数据接口包.....	3
2.2.4 Matplotlib 图形绘制包.....	3
2.2.5 Tkinter 图形用户界面工具包.....	3
第三章 系统分析与建模	4
3.1 可行性分析	4
3.1.1 技术可行性	4
3.1.2 经济可行性	4
3.1.3 操作可行性	5
3.2 需求分析	5
3.2.1 功能需求分析	5
3.2.2 性能需求分析	8
3.2.3 软硬件需求分析	8
3.3 系统用例图	9
3.4 系统类功能模块	10
3.5 系统流程图	10
第四章 系统设计与实现	12
4.1 开发环境构建	12
4.1.1 Python 的安装	12
4.1.2 各类开发包的安装	12
4.1.3 各类开发包的引入	13
4.2 EventManager 模块	14
4.2.1 EventManager 模块设计	15
4.2.2 主界面可视化	15

4.3 数据显示模块	18
4.3.1 数据显示模块设计	18
4.3.2 数据显示模块实现	19
4.4 数据分析模块	21
4.4.1 数据分析模块设计	21
4.4.2 数据分析模块实现	21
4.5 动态日志模块	26
4.5.1 动态日志模块设计	26
4.5.2 动态日志模块实现	27
第五章 测试	29
5.1 测试目的与测试方法	29
5.1.1 测试目的	29
5.1.2 测试方法	29
5.2 系统测试设计	29
5.2.1 系统测试对象	29
5.2.2 测试执行	30
5.3 功能测试	30
5.3.1 功能测试	30
5.3.2 测试用例	30
5.4 测试结果	31
第六章 总结与展望	32
致谢	33
参考文献	34

第一章 绪论

1.1 课题研究背景及意义

现如今，随着大数据时代的到来，云计算、物联网等的兴起和数据库技术的成熟，当今社会中的各式各样的数据都正在以非常快的速度增长。科学家、工程师以及定量管理分析师在数据分析这一范围面临着诸多的挑战。因为传统的数据处理方式很难以从海量的数据当中发掘出隐藏的规律，必须要借助编程语言和数据分析软件进行数据挖掘并进行分析。数据科学家们都期望会有一个可以用较小的编程成本去实现在大数据集上进行较为准确的数值分析的方法，而且这些数据科学家们都期望自己编写的代码在前者的基础之上还能够具备运行速度较快、可读性较好、执行效率较高的特点，并且可以尽可能地贴近他们熟悉的一系列的数学概念。

本课题所要研究的股市数据仅仅是大数据集里面一个很小很小的分支，但是何不把它看作是大数据时代的一个缩影。在当代社会，人们的生活条件变得比以往好了，经济情况也相对的富裕了许多，那么人们在满足了自身或者家庭正常的日常开销之后，还富余的钱财也不会像以往一样存入银行生利息，更多的会去选择投资，而股票正是一个不错的投资选择。然而现在仅是中国深沪两市的股票数量就已达到了 3000 以上，每日的股市数据的更新量也是一个庞大的数字。单只股票数据包含了每日的开盘价、收盘价、最高价、最低价、成交量等指标，但是要分析股市数据，必定会监测股票在定量的一段时间内的数据更新。面对复杂和多维的股市数据，我们也有着和数据科学家们同样的期望。

在当今时代的科学计算领域中，为我们所知的就有很多符合数据科学家们的要求的解决方案。

Python 是一种现下流行的通用编程语言，在科学计算领域很受欢迎。虽然在这方面，C、C++、Fortran 等编程语言的优势可谓是各有千秋，但是这些编程语言都不是交互式语言，而且很多的人都觉得用这些编程语言来编写的代码过于复杂。除了这些，常见的编程类的商业产品还有 Matlab、Maple 和 Mathematica，它们提供给使用者强大的脚本语言，可是这类的编程产品和通用的编程语言相比较来说，功能是很局限的。另外，还有一些开源工具例如 Scilab、GNU Octave 和 R，它们和 Matlab 有些类似，作用都是很有限的。那么显而易见的，作为编程语言来说，Python 所能提供的功能比上述的编程工具都更为强大。Python 是面向对象语言，比 C 语言和 Fortran 更加高级。虽然 Python 本身不具备与 Matlab 等效的功能块，但是 Python 中的 NumPy 从某些意义上来说，是可以取代 Mathematica 和 Matlab 的部分功能的。原因是 NumPy 中涵括了很多的具备实用价

值的数学函数，能够很自然地使用矩阵和数组，并且 NumPy 对于多维数组的切片和索引的操作可以对数据进行更有针对性的分析。

1.2 研究目的

首先需要对本课题所要运用到的相关的计算机知识有一个整体的了解和把握，并且在这个层次上还需要进行一些基础的系统分析。系统分析是程序开发过程中一个必不可少的重要的环节，它的目的是用最小的代价在尽可能短的时间内确定问题是否能够解决，并且确定系统必须完成哪些工作，也就是对目标系统提出完整、准确、清晰、具体的要求。

本课题主要是运用 Python 语言中的 numpy 提供的强大的数据计算功能和整洁易读且出错最少的代码编写特点，编写代码来实现对于复杂和多维的股市数据的分析。并且还需要结合 Python 语言相关的图形绘制包 Matplotlib 来实现股市数据的数据可视化，使得数据更为直观、形象地呈现出来，让数据所要传递的信息更加明显易懂。

1.3 论文的结构安排

本论文一共分为七章，具体内容如下：

第一章主要是介绍了本文的研究背景及意义，研究目标和论文的结构安排。

第二章主要是介绍了股市数据分析工具的研发所涉及到的相关技术的介绍。

第三章主要是关于系统分析与建模。

第四章主要是关于系统的设计和实现，包含各个模块的实现流程图、运行效果图和关键代码。

第五章主要是对系统的测试。

第六章主要是对本课题的总结与展望。

第二章 相关技术介绍

2.1 数据可视化技术简介

数据可视化技术是指运用计算机图形学和图像处理技术，将数据转换为图形或者图像在屏幕上显示出来，并利用数据分析和开发工具发现其中未知信息的交互处理的理论、方法和技术^[1]，不再局限于通过关系数据表来观察和分析数据信息，还能以更直观方式看到数据及其结构关系。

数据可视化技术能够分析大量复杂和多维的数据，提供像人眼睛一样的直观的、交互的和反应灵敏的可视化环境。数据可视化的特点具有交互性、多维性、可视性。

2.2 Python 编程语言及相关介绍

2.2.1 Python 编程语言

Python 是通用的面向对象语言，它采用 GPL 兼容的版权，这就意味着使用 Python 时无需支付任何版权费用。其具有简单整洁，出错率低，可维护性较好，动态类型无需定义、不存在内存溢出错误、无需对代码进行编译的特性。

2.2.2 NumPy 数值计算库

NumPy (Numerical Python) 提供了 Python 对多维数组对象的支持, 其具有矢量运算能力, 快速、节省空间的特点。NumPy 支持高级大量的维度数组与矩阵运算, 此外也针对数组运算提供大量的数学函数库。

2.2.3 Tushare 财经数据接口包

Tushare 是 Python 的财经数据接口包。其功能对于股市数据等金融数据进行数据采集、数据清洗和数据存储。

2.2.4 Matplotlib 图形绘制包

Matplotlib 包是出色的 2D、3D 图形库，用来绘制图形。

2.2.5 Tkinter 图形用户界面工具包

Tkinter，也叫 Tk 接口，是 Python 的图形用户界面工具包，是一个轻量级的跨平台图形用户界面（GUI）开发工具。

第三章 系统分析与建模

3.1 可行性分析

3.1.1 技术可行性

技术可行性分析可以分为硬件可行性分析和软件可行性分析。

硬件可行性分析：本课题所需要的硬件要求方面并不存在很特殊的要求，所需要的硬件配置只要能够确保系统的正常工作，除此之外还需要拥有较高的效率，而这些要求实际上在我们日常使用的普通的硬件配置基础上就能够轻松的实现。假如拥有的硬件配置特别老旧低端，它将会导致系统的性能低下、运行的效率低，从而导致整个系统的运行不顺畅。以目前计算机普及程度广和硬件优化速度慢的特点来看，对于硬件配置的要求是十分容易实现的。通过上述分析，本课题的研究在硬件方面是可行的。

软件可行性分析：numpy 虽然代码编写方式在实现等价数据分析功能的前提下比纯 Python 代码编写方式更为整洁，但是 numpy 实质上是基于 Python 的，换言之，numpy 是不能脱离 Python 单独安装的。因此在安装 numpy 之前需要先安装 Python。现有的某些操作系统已经默认安装有 Python 环境，但是值得注意的是仍然需要检查 Python 的版本是否能够与要安装的 numpy 版本兼容。Numpy 在 windows、Mac OS X、以及各种 Linux 发行版上均有二进制安装包，这意味着开发人员能够安装包含源代码的版本，但是对于 Python 的版本有更高要求，需要在系统中安装 Python 2.4.x 或者更高的版本。在 Windows 操作系统中 Python 的安装程序可以在 www.python.org/download 下载。在完成 Python 的情况下，numpy 的安装程序应该能自动检测到。在 windows 上安装 numpy 是必需的，安装过程也并没有想象中的那么复杂，只需要从 SourceForge 网站下载 numpy 的 windows 的安装程序，运行后在安装向导的指导下便能完成安装流程。Matplotlib 和 Scipy 安装过程都不复杂。软件的获取安装过程都不具备难度，并且在数据分析领域里 Python 编程语言有着突出的表现，通过上述分析，并且结合本课题的实际情况来考虑，本课题的研究在软件方面也是可行的。

3.1.2 经济可行性

本课题的研究基于 windows 操作系统的平台，研究所需要的硬件配置，在现有市场上都很容易能够购买到。软件基于 Python 编程语言和 numpy 包，这些都是开源的，开源的所有益处都可以享受到，代码是面向所有人开放的，这意味着其中的价格方面是零成本，不必担心软件授权关心的问题。本课题所要研发的股市数据分析工具，主要是要实现能够从网络上下载股市数据、选择特定股票指标有针对性的

分析数据并实现股市数据可视化，例如 K 线图、布林线、趋势线等。程序并不复杂，所以在开发时对于人力的要求并不高。综合上述的分析，本课题的研究在经济可行性方面是可行的。

3.1.3 操作可行性

用户的计算机即使是不同的操作系统环境，在安装了相应的可兼容 Python 软件和数据包的条件之下，就能够进行一系列的操作，操作步骤简单易懂。本课题的研究在操作可行性上的分析可以证明在操作方面是可行的。

3.2 需求分析

3.2.1 功能需求分析

本课题研发的软件是股市数据分析工具，程序基于 Numpy 数据计算包来对股市数据进行计算分析，但是要让用户可以看见各类分析结果，那么必须要提供的是：可视化的面对用户界面；能够显示数据；支持调用各类分析图形；动态日志生成；能够清空日志。

(1) 数据下载需求分析

数据下载的需求：用户在单行文本框里输入任意一只股票代码，系统会去爬取数据；当用户点击了“数据显示”按钮之后，系统所获取到的数据将会全部显示在按钮下面的文本框里。该需求的用例描述如表 3.1 所示，相应的用例图如图 3.1 所示：

表 3.1 数据下载显示用例规约

用例名称	数据下载显示	
参与者	用户	
前置条件	主界面可视化成功	
后置条件	数据显示成功	
事件驱动		
参与者动作		系统行为
用户在文本框中输入任意一只股票代码		系统爬取代码相应的数据
用户点击“数据显示”按钮		对用户显示获取到的数据

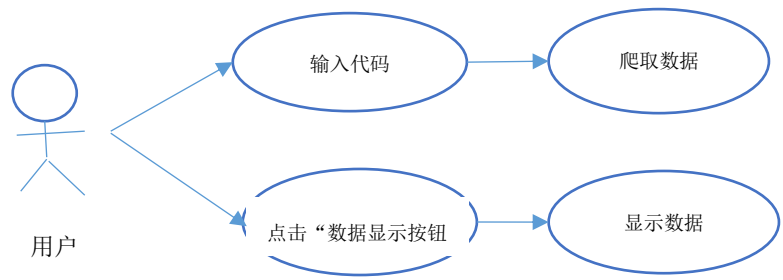


图 3.1 数据下载用例图

(2) 数据分析模块需求分析

数据分析的需求：用户通过点击“K线图”、“布林线”、“成交量”、“趋势线”、“价格直方图”这些按钮来调用查看各类分析图形，每当用户点击一个按钮后，系统都会弹出一个窗口来展示数据分析图形。该模块需求的分析的用例描述如表 3.2 所示，相应的用例图如图 3.2 所示：

表 3.2 数据分析用例规约

用例名称	数据分析	
参与者	用户	
前置条件	数据获取显示成功	
后置条件	成功调用了各类数据分析图形	
事件驱动		
参与者动作		系统行为
用户点击“K线图”按钮		弹出窗口展示 K 线图分析
用户点击“布林线”按钮		弹出窗口展示布林线分析
用户点击“趋势线”按钮		弹出窗口展示趋势线分析
用户点击“成交量”按钮		弹出窗口展示成交量分析
用户点击“价格直方图”按钮		弹出窗口展示股价分布直方图分析

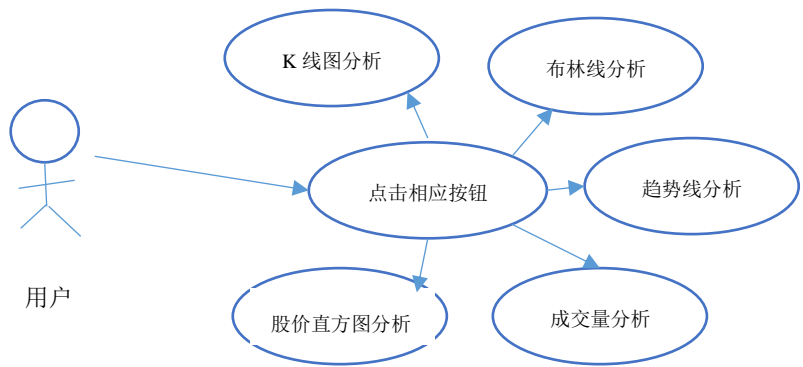


图 3.2 数据下载用例图

(3) 动态日志模块需求分析

日志生成的需求：在用户点击“数据显示”、“K线图”、“布林线”、“成交量”、“趋势线”、“价格直方图”这些按钮之后，激活相应的事件驱动，系统发送事件之后会将过程生成日志打印显示，提供给用户参考。该需求的分析的用例描述如表 3.3 所示，相应的用例图如图 3.3 所示：

表 3.3 日志生成用例规约

用例名称	日志生成	
参与者	用户	
前置条件	点击按钮激活了事件驱动	
后置条件	日志成功生成	
事件驱动		
参与者动作		系统行为
用户点击各类按钮，激活相应事件驱动		系统就过程生成日志打印显示

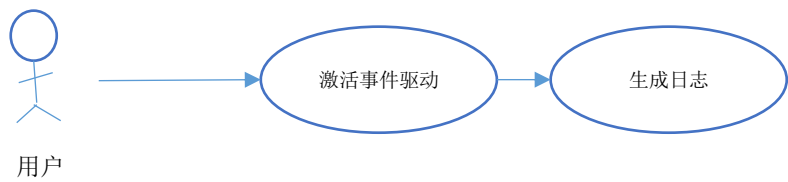


图 3.3 日志生成用例图

清空日志的需求：在动态日志生成之后，用户可以选择是否要进行清空日志的操作。用户点击“清空日志”按钮之后，系统会将日志框中生成的日志全部清空。该需求的分析的用例描述如表 3.4 所示，相应的用例图如图 3.4 所示：

表 3.4 清空日志用例规约

用例名称	清空日志	
参与者	用户	
前置条件	日志成功生成	
后置条件	日志内容全部清空	
事件驱动		
参与者动作		系统行为
用户点击“清空日志”按钮， 激活“清空日志”的事件驱动		系统清空所有生成的日志

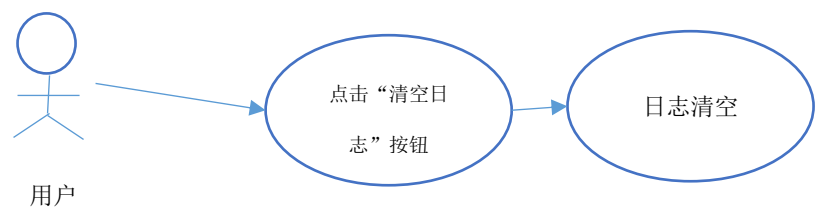


图 3.4 清空日志的用例图

3.2.2 性能需求分析

在系统实现之前，对系统性能的需求分析是必不可少的一个步骤，性能分析能够帮助我们发现系统性能在软硬件方面的优缺点，让我们最大限度的平衡系统资源，让我们开发的系统在软硬件上都能够达到合理的配置。

根据需求分析以及自己的生活体会，本系统需要具有各个控件有文字提示且摆放位置合理，方便用户调用，且操作便捷、功能相对完善的特点。

3.2.3 软硬件需求分析

(1) 硬件需求

处理器（CPU）:Intel（R） Core（TM） i5-4210U CPU @ 1.70Hz 2.40GHz
安装内存（RAM）: 4.00GB
系统类型: 64 位操作系统

(2) 软件需求

操作系统: Windows7

运行环境: IDLE

开发语言: Python

开发工具: Python3.6.5

3.3 系统用例图

根据以上的需求分析,要进行系统的建模。在整合了需求分析中的各个模块的用例图,在本小节中将呈现一个整体的系统用例图,清楚地描绘系统外在可见的需求情况。

本系统需实现数据下载、数据显示、调用数据分析结果、调用日志展示和清空日志五个基本功能。调用数据分析结果主要是将多维数据有针对性的切片分析后可可视化,具体有 K 线图分析、布林线分析、趋势线分析、成交量分析、股票价格直方图分析。系统结构用例图如图 3.4 所示:

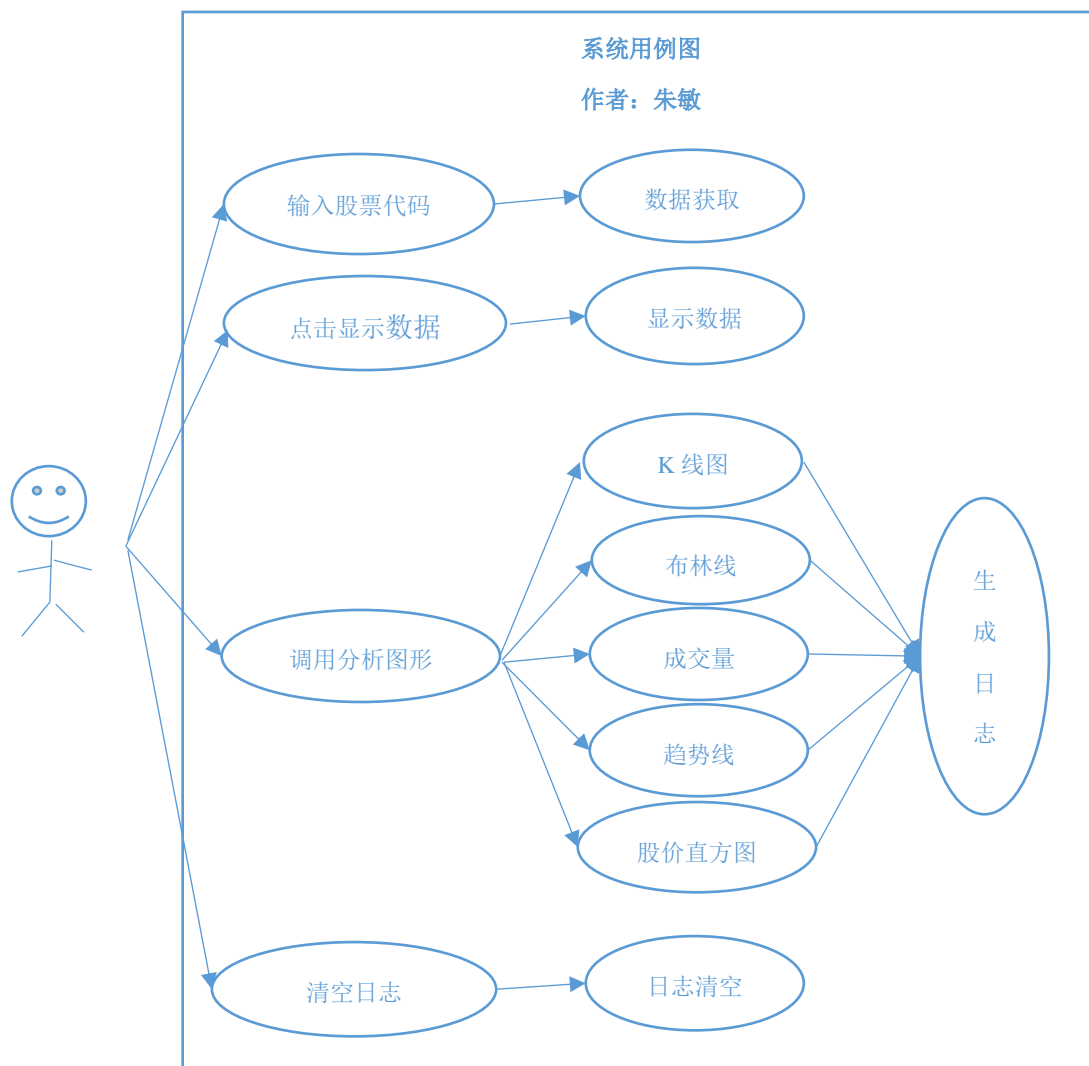


图 3.4 系统用例图

3.4 系统类功能模块

本课题所研发的股市数据分析工具包含了框架、标签、按钮、单行文本框和多行文本框这几个元素，调用方式和说明由表 3.5 给出：

表 3.5 元素调用方式和说明表

元素	tkinter 类	说明
框架	Frame	承载其他的元素
标签	Lable	显示不可编辑的文本或图标
按钮	Button	当用户激活它时执行一个动作
单行文本框	Entry	接受并显示一行文本
多行文本框	Text	接受并显示多行文本

本系统在定义了类之后，还需要定义并创建一些方法才能够使得可视化界面形成以及能够进行各类功能展示：

set_init_window: 负责定义可视化界面的根窗口，定义对象、事件以及事件处理器。

self.init_data_Text: 负责可视化界面里单行文本框和多行文本框的实现。

self.init_data_scrollbar_y: 负责可视化界面里纵向滚动条的实现，并与多行文本框进行事件绑定。

self.log_label: 标签，用于定义事件。

self.write_log_to_Text: 定义事件被处理之后用于进行相关日志动态打印。

show_content: 用于调用数据获取显示。

show_k_line: 用于调用 K 线图分析显示。

show_chengjiao: 用于调用成交量分析显示。

show_bulin: 用于调用布林线分析显示。

show_zhifangtu: 用于调用股票价格直方图分析显示。

show_zhangdie: 用于调用趋势线分析显示。

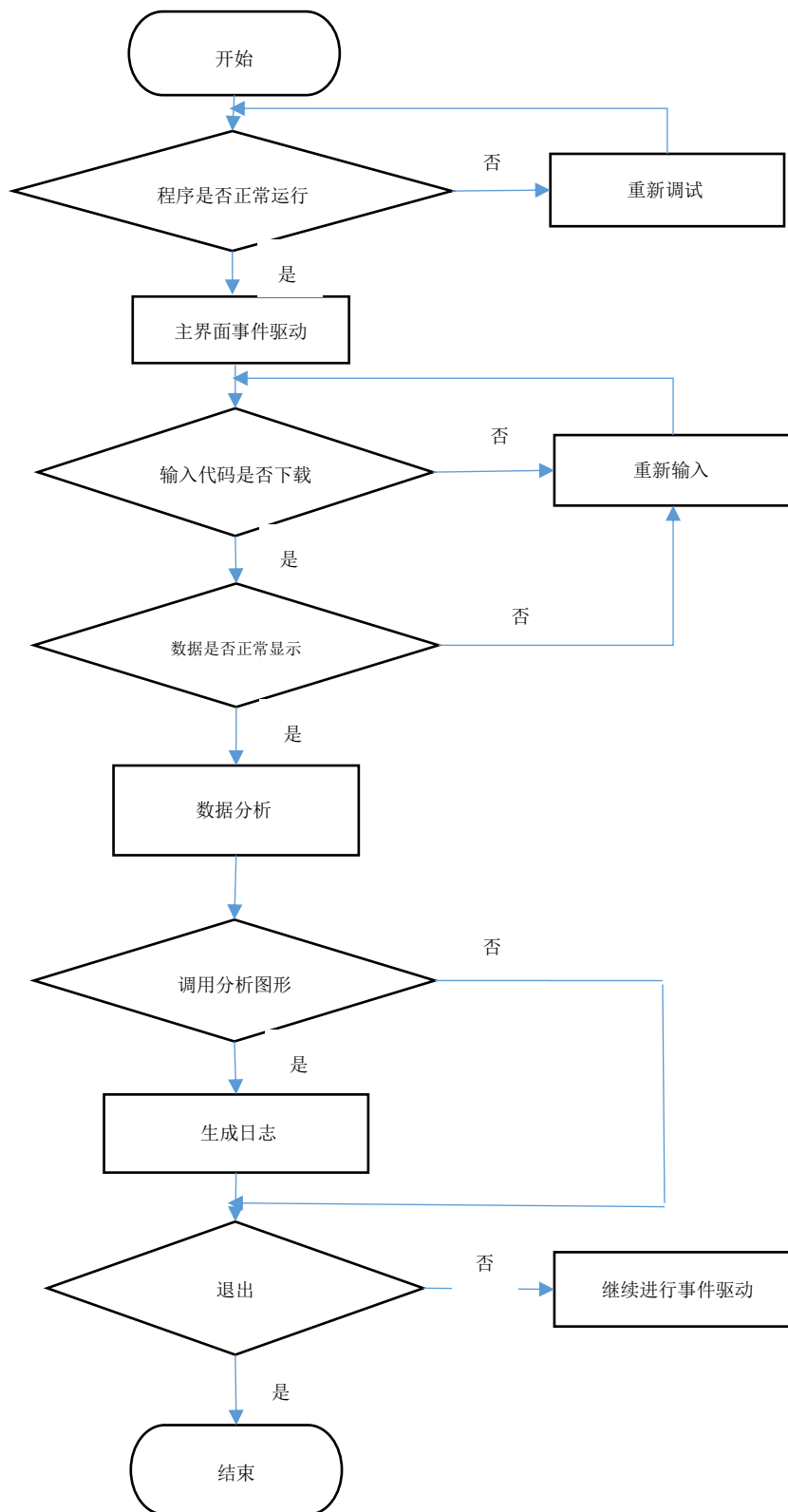
get_current_time: 获取日志生成的实时时间。

delete_Text: 清空文本内容。

delete_log_Text: 清空日志内容。

3.5 系统流程图

流程图是一个系统的整体运行流程图，一个系统的好与差很大程度上取决于你的系统的运行流程是不是符合功能需求，本系统的流程图如图 3.5 所示：



3.5 系统流程图

第四章 系统设计与实现

4.1 开发环境构建

本课题所要开发的程序所需要的环境：

1. 操作系统：Windows 系统。
2. 环境：Python。
3. 开发工具包：numpy、tushare、matplotlib、tkinter。

在开发本应用时，所使用的开发平台为 Win7 操作系统，所使用的开发环境为 Python3.6，所用到的开发工具包有 numpy、tushare、matplotlib、tkinter，都为相应版本。

4.1.1 Python 的安装

本课题所要开发的程序要用到的版本是 Python3.6.5。下载安装至操作系统中之后，假如被其它的版本覆盖，那么则需要配置环境变量，步骤如下：第一步：在“计算机”上点右键—>“属性”—>“高级系统设置”—>“环境变量”；第二步：在“系统变量”中将 path 中其它版本路径删除即可。在 cmd 命令里输入 python 测试版本，结果如图 4.1 的界面，则当前版本为可用版本。

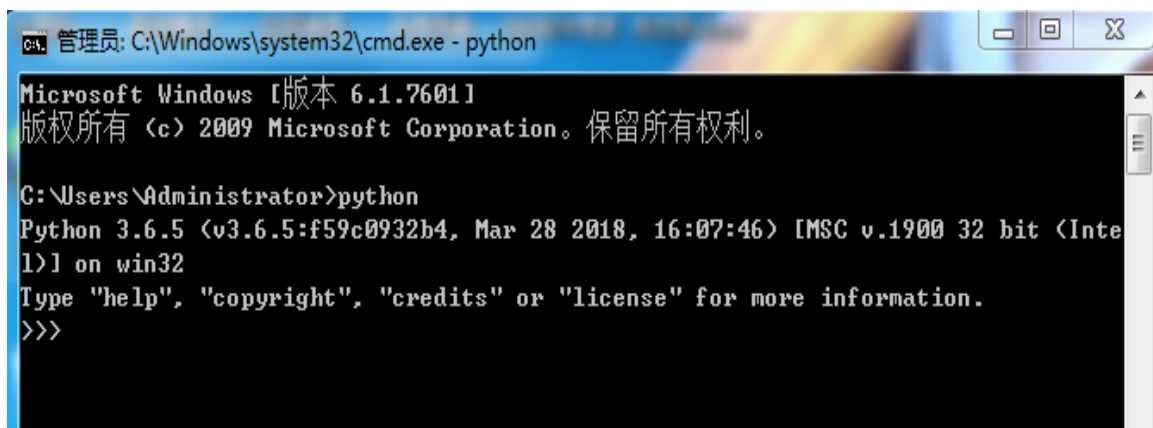


图 4.1 Python 测试图

4.1.2 各类开发包的安装

1. NumPy 数值计算扩展包的安装：我们已经下载安装好了 Python，并且设置好了路径，NumPy 的安装程序是能够自动检测到的。
2. Matplotlib 图形绘制包的安装：打开 cmd 命令，输入 pip3 install matplotlib，点击回车键。
3. Tushare 财经数据接口包的安装：打开 cmd 命令，输入 pip3 install tushare，点击回车键。

4. Tkinter 图形用户界面工具包的安装：打开 cmd 命令，输入 `pip3 install tkinter`，点击回车键。

工具包的安装过程都是大同小异的，下面就只给出 Tushare 包的安装实例，如图 4.2:

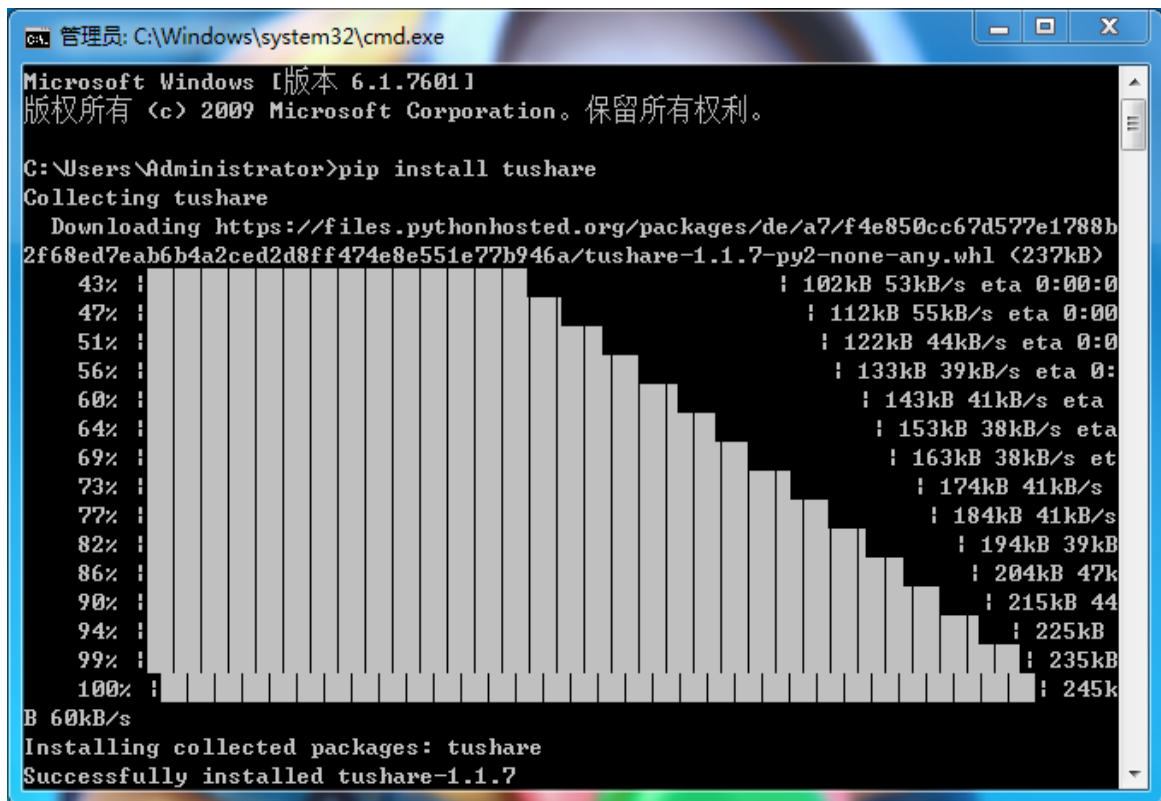


图 4.2 Tushare 工具包的安装

4.1.3 各类开发包的引入

因为 Python 本身并没有办法直接调用这些 numpy、tushare、matplotlib、tkinter 等的工具包，所以在编写代码的时候我们需要先将引入这些包的模块。代码在下面展示：

```
import pandas as pd
from pandas import *
import datetime
from tkinter import *
from sqlalchemy import Column, String, INTEGER
import tkinter.filedialog as filedialog
from tkinter.simpledialog import *
import tushare as ts
import matplotlib.pyplot as plt
import os
import time
import seaborn as sns
import seaborn.linearmodels as sns1
sns.set_style("whitegrid")
```

4.2 EventManager 模块

本课题所开发的股市数据分析工具在实现主界面可视化之后，可以分为三个模块，分别是数据下载模块、数据分析模块、动态日志模块。系统完成之后可在 windows 操作系统下运行，用户可以通过点击界面中的按钮来实现数据下载显示、调用数据分析以及看到自己的动态日志。股市数据分析工具的系统结构功能图如图 4.3 所示：

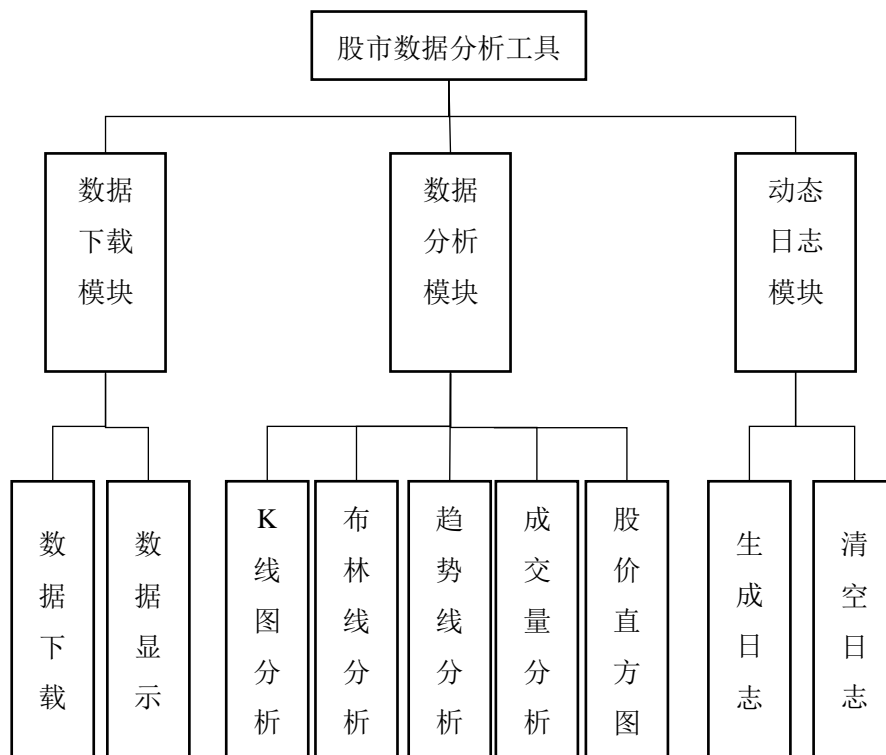


图 4.3 系统结构功能图

4.2.1 EventManager 模块设计

EventManager 模块实际上是主界面显示及所有控件显示的一个整合，换言之，这个模块所要实现的就是主界面的可视化，能够给用户提供一个可以查看下载的数据和调用数据分析图形的界面。EventManager 模块实现流程图如图 4.4 所示：

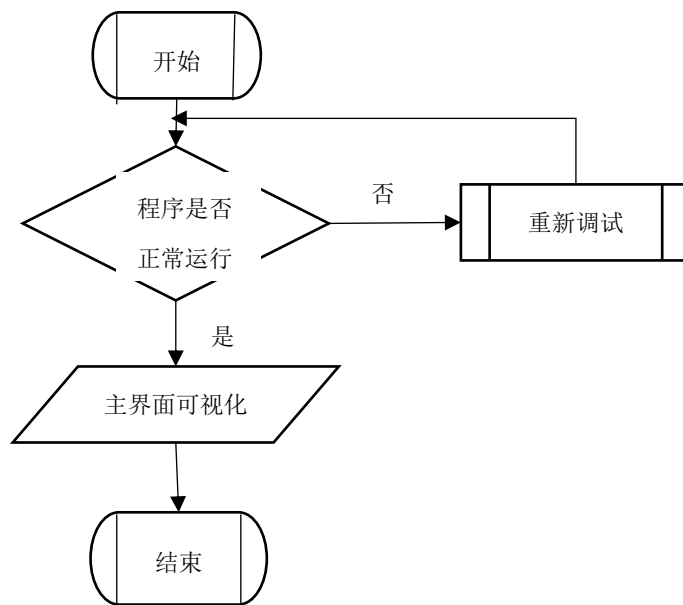


图 4.4 EventManager 模块实现流程图

4.2.2 主界面可视化

EventManager 可视化主界面设计主要运用到如下几个 GUI 元素：

Frame: 框架，用于承载其他的 GUI 元素。定义 Frame 对象，在构造方法中（init 方法）指定 parent 为 None，代表是顶层窗口，title 为窗口标题，size 为大小，pos 为位置。

Lable: 标签，用于显示不可编辑的文本或者图标。

Button: 按钮，当用户激活它时执行一个动作。用于定义数据显示、各类分析图形调用、日志清空等按钮。

Entry: 单行文本框，接受并显示一行文本。用于接受用户输入的股票代码。

Text: 多行文本框，接受并显示多行文本。用于显示获取到的股票数据和生成的日志动态。

系统设计好的可视化主界面正常运行如图 4.5 所示：

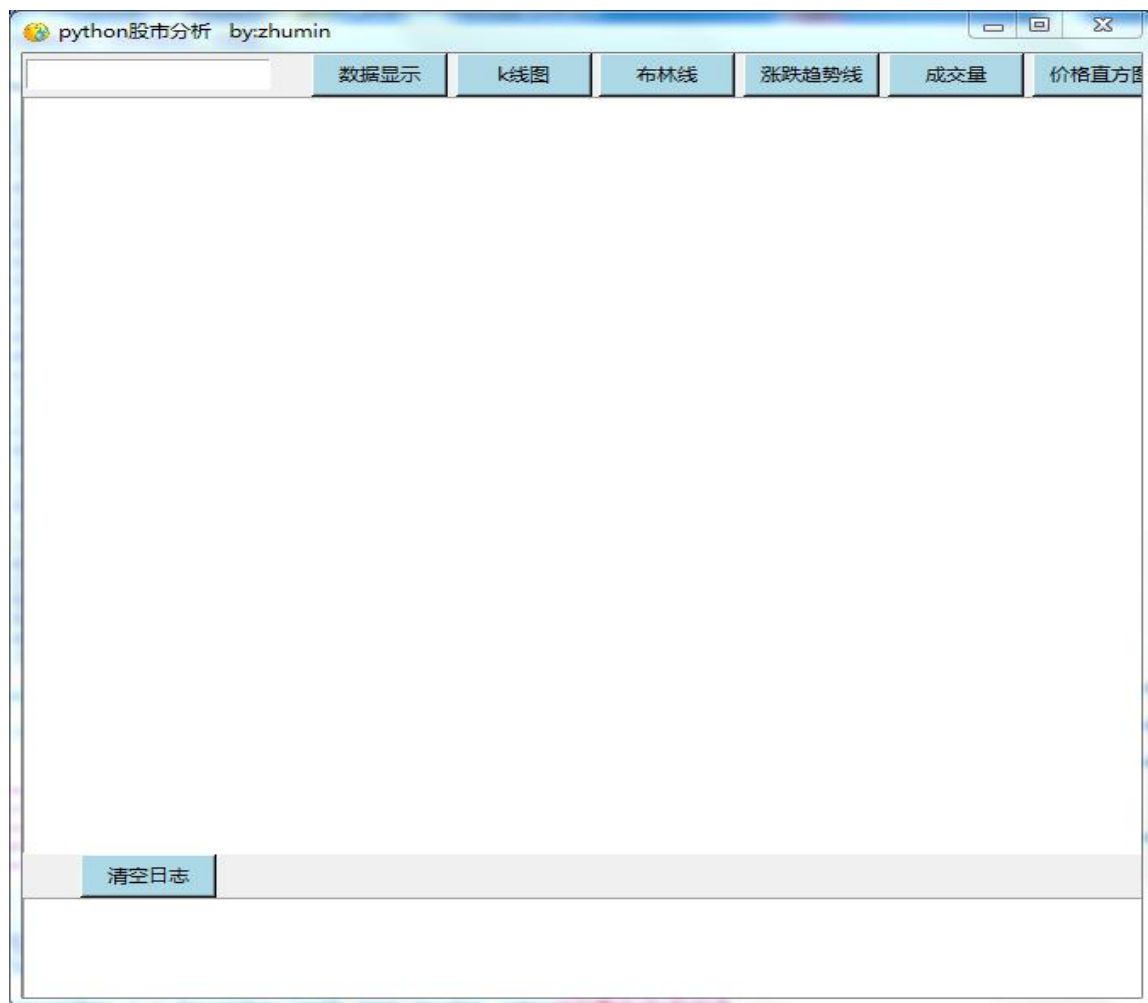


图 4.5 可视化主界面

定义可视化主界面的关键代码在下面给出：

```
class MY_GUI():  
    def __init__(self, init_window_name):  
        self.init_window_name = init_window_name  
  
def gui_start():  
    init_window = Tk()  
    #实例化出一个父窗口  
    ZMJ_PORTAL = MY_GUI(init_window)  
    # 设置根窗口默认属性  
    ZMJ_PORTAL.set_init_window()  
    init_window.mainloop()  
    #父窗口进入事件循环，可以理解为保持窗口运行，否则界面不展示  
gui_start()
```

主界面设计中,用户能够通过单行文本框中手动输入股票代码来让系统去自动爬取相应的股票数据,单行文本框的实现的主要代码如下:

```
self.stock_id_control = Entry(self.init_window_name)
self.stock_id_control.grid(row=0, column=1)
```

主界面设计中,用户能够通过点击主界面中的按钮来激活事件驱动,按钮的实现的主要代码如下:

```
Button(self.init_window_name, text="数据显示", bg="lightblue",
        width=10, command=self.show_content).grid(row=0, column=6)
Button(self.init_window_name, text="k线图", bg="lightblue",
        width=10, command=self.show_k_line).grid(row=0, column=7)
Button(self.init_window_name, text="布林线", bg="lightblue",
        width=10, command=self.show_bulin).grid(row=0, column=8)
Button(self.init_window_name, text="涨跌趋势线", bg="lightblue",
        width=10, command=self.show_zhangdie).grid(row=0, column=9)
Button(self.init_window_name, text="成交量", bg="lightblue",
        width=10, command=self.show_chengjiao).grid(row=0, column=10)
Button(self.init_window_name, text="价格直方图", bg="lightblue",
        width=10, command=self.show_zhifangtu).grid(row=0, column=11)
```

主界面设计中,多行文本框的设计是为了接收数据并下载,显示给用户。即使显示的是单只股票的数据,数量也是非常庞大的,所以加入了纵向滚动条的设计,使得多行文本框绑定滚动条,使得用户能够拉动滚动条来查看下载的数据。多行文本框和滚动条的设计的主要代码如下:

```

#文本框
self.init_data_Text = Text(self.init_window_name,
                            width=100, height=40)
self.init_data_Text.grid(row=1, column=0,
                          rowspan=12, columnspan=14)
#创建纵向滚动条
self.init_data_scrollbar_y =
Scrollbar(self.init_window_name)
#将创建的滚动条通过command参数绑定到需要拖动的Text上
self.init_data_scrollbar_y.config(command=
                                   self.init_data_Text.yview)
#Text反向绑定滚动条
self.init_data_Text.config(yscrollcommand=
                           self.init_data_scrollbar_y.set)
self.init_data_scrollbar_y.grid(row=1, column=13,
                                rowspan=12, sticky='NS')

```

主界面设计中，日志动态的设计是为了能够展示用户所触发的事件驱动，同时也为用户提供了清空日志的选项。设计的主要代码如下：

```

self.log_label = Label(self.init_window_name,
                        text="日志").grid(row=15, column=1)
Button(self.init_window_name, text="清空日志", bg="lightblue",
       width=10, command=self.delete_log_Text).
grid(row=15, column=1)
self.log_data_Text = Text(self.init_window_name,
                           width=100, height=9)
self.log_data_Text.grid(row=16, column=1, columnspan=13)

```

4.3 数据显示模块

4.3.1 数据显示模块设计

数据显示模块的设计的目的是为了能够让用户在输入想要知道信息的股票代码后，点击主界面上相应的按钮就能够显示获取到的股票数据。数据显示模块实现流程图如图 4.6 所示：

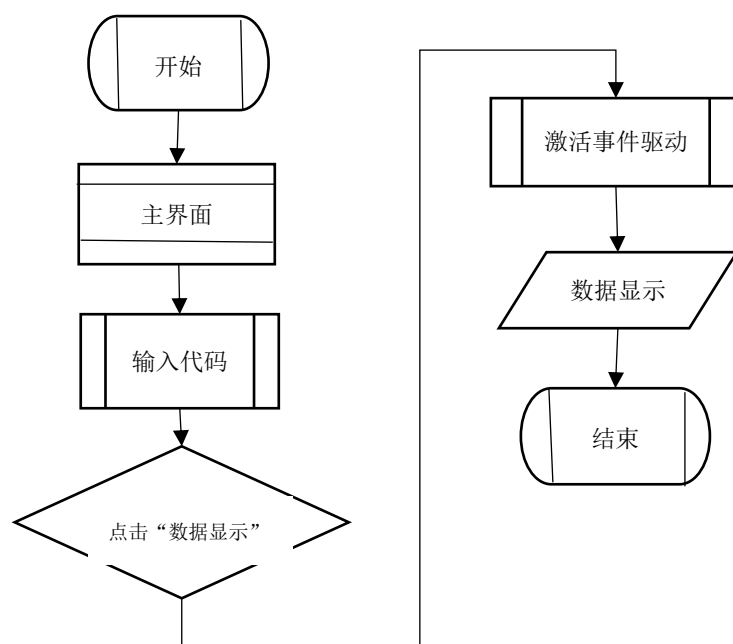


图 4.6 数据显示模块流程图

4.3.2 数据显示模块实现

数据下载模块绑定了单行手动输入文本框、数据显示按钮、多行数据显示文本框和日志框。

用户在文本框里输入股票代码之后，相当于给了系统一个去获取相应股票数据的指令，在用户点击数据显示按钮之后，触发事件驱动，多行文本框内会显示系统获取到的数据，日志框里会打印出用户动态轨迹。

因为股票数据可能存在时间跨度大的现象，老旧的数据对于新的信息传递可能会产生一定的影响，所以我们只截取了近两个月来的股票数据作为有效的分析数据。程序正常运行的效果图如图 4.7 所示：

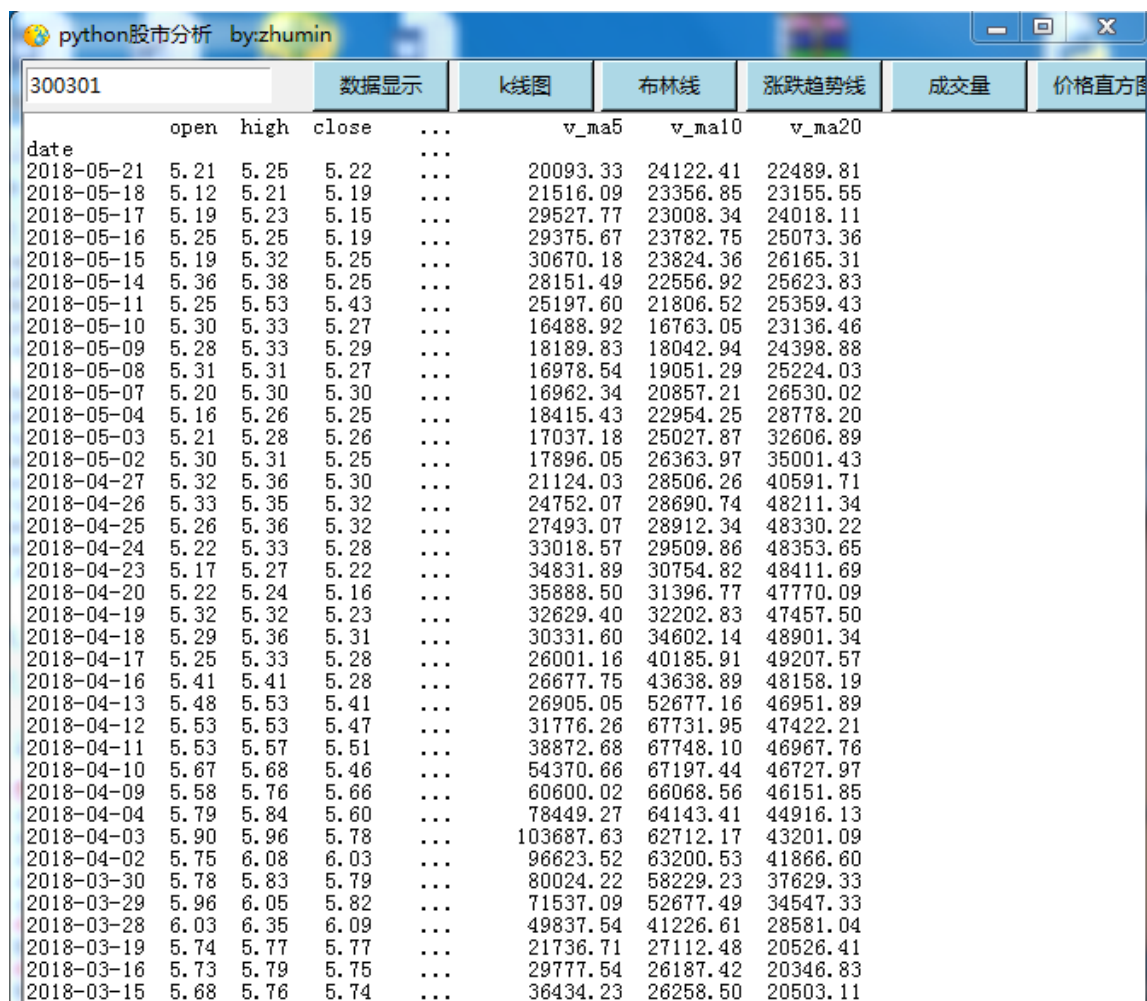


图 4.7 数据下载模块运行效果图

数据下载模块的具体实现代码如下：

```
def show_content(self):
    stock_id = self.stock_id_control.get()
    self.write_log_to_Text("INFO:显示%s 股票原始内容"
                           % stock_id)
    start = datetime.datetime(2018, 3, 1)
    end = datetime.datetime(2018, 5, 21)
    self.init_data_Text.delete(1.0, END)
    #选取一支股票
    self.stock = ts.get_hist_data(stock_id, str(start),
                                   str(end))
    print(self.stock)
    self.init_data_Text.insert(END, "%s" % self.stock)
```

4.4 数据分析模块

股市数据庞大并且多维度，在进行股市数据的分析的时候，Python 编程语言自带的开源的 NumPy 数值计算包提供了很大的便利。使用 NumPy，就能够很自然地使用数组和矩阵，其中自带的一些方法能够对数组或者矩阵进行切片分析，对于股市数据这一类的多维度数据的分析十分适合。分析之后，将运用 matplotlib 图形绘制包进行图形绘制。本小结就对于股票成交量、趋势线、股票价格直方图、K 线图、布林带进行分析。

4.4.1 数据分析模块设计

数据分析模块的设计的目的是为了能够让用户通过点击相应数据分析的按钮，以此来查看调用各种图形分析结果。数据分析包括以下五类内容：K 线图分析、布林线分析、成交量分析、趋势线分析和股价分布直方图分析。数据分析模块实现流程图如图 4.8 所示：

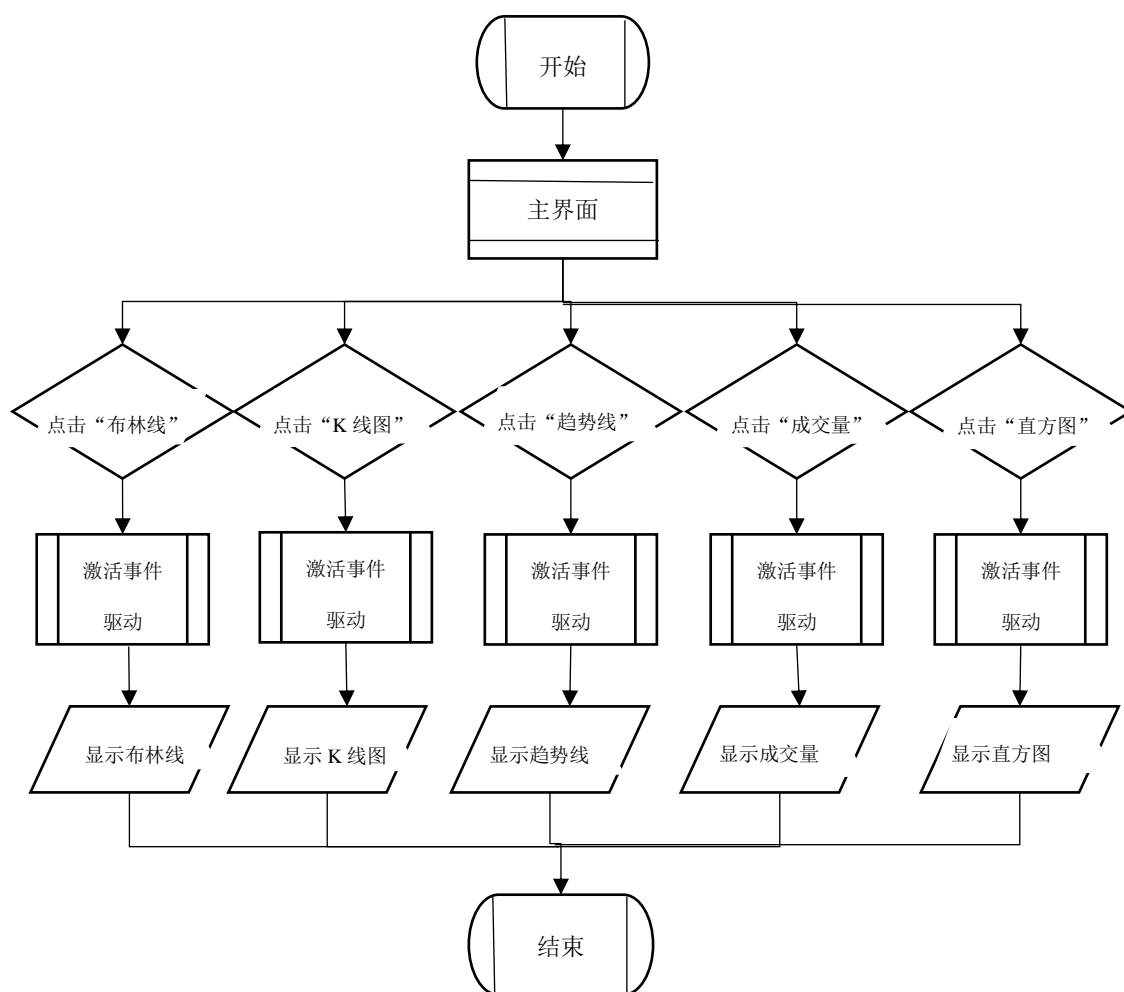


图 4.8 数据分析模块流程图

4.4.2 数据分析模块实现

(1) 成交量分析:

成交量是一种供需的表现,指一个时间单位内对某项交易成交的数量。成交量是股市数据中一个比较突出的指标。编写代码时需要对“volume”这一列的数据进行切片分析。

成交量分析的设计绑定了“成交量”按钮和多行日志文本框,当用户点击“成交量”按钮的时候激活事件驱动,系统会给出一个单独窗口来展示成交量分析的图形,并且会获取用户动态轨迹。程序正常运行的效果图如图 4.9 所示:

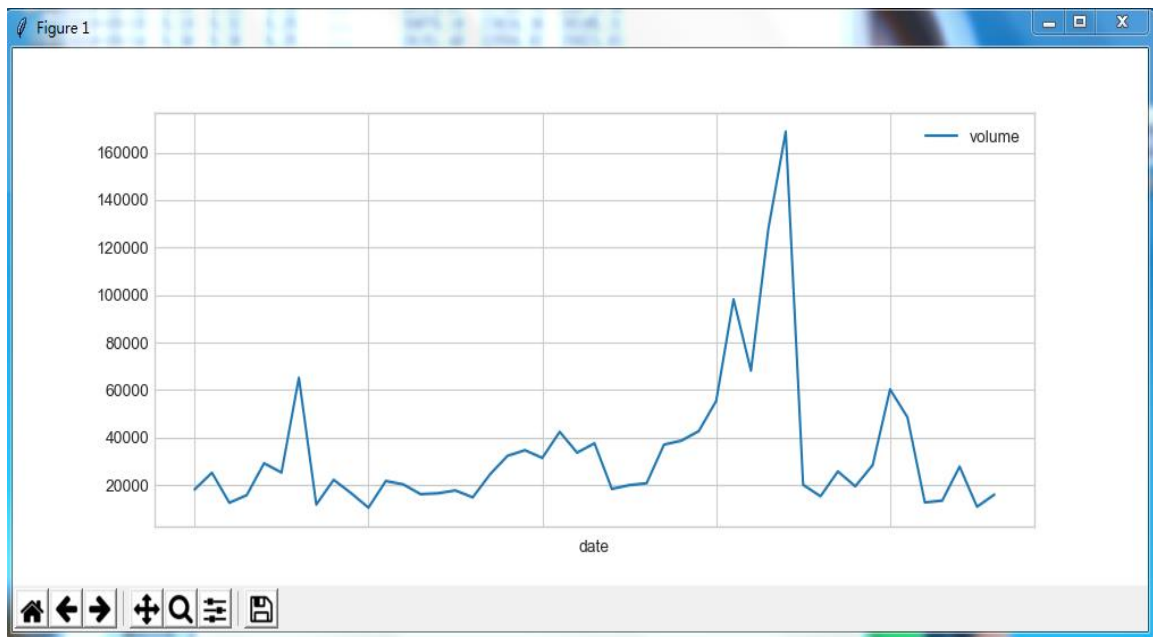


图 4.9 成交量分析

成交量分析的具体实现代码如下:

```
#成交量分析
def datestr2date(s):
    s = s.decode('utf-8')
    return datetime.datetime.strptime(s, "%Y-%m-%d").date()

d, v = np.loadtxt(filepath, delimiter=',', skiprows=1,
                  usecols=(0, 5), unpack=True,
                  dtype={"names": ('date', 'volume'),
                        "formats": (np.object_, np.float32,
                                    np.float64, np.float64)},
                  converters={0: datestr2date})

#绘制图形
plt.plot(d, v, 'b-', label = "trading volume")
plt.title('stock ' + filepath[:-4] + ' trading volume')
plt.legend()
plt.show()
```

(2) 趋势线分析:

趋势线，就是根据股价上下变动的趋势所画出的线路，画趋势线的目的，即依其脉络寻找出恰当的卖点与买点。趋势线是能够反映股票数据的一个突出指标。编写代码时将对“close”这一列的数据进行切片分析。

趋势线的分析设计绑定了“趋势线”按钮和多行日志文本框。程序正常运行的效果图如图 4.10 所示：

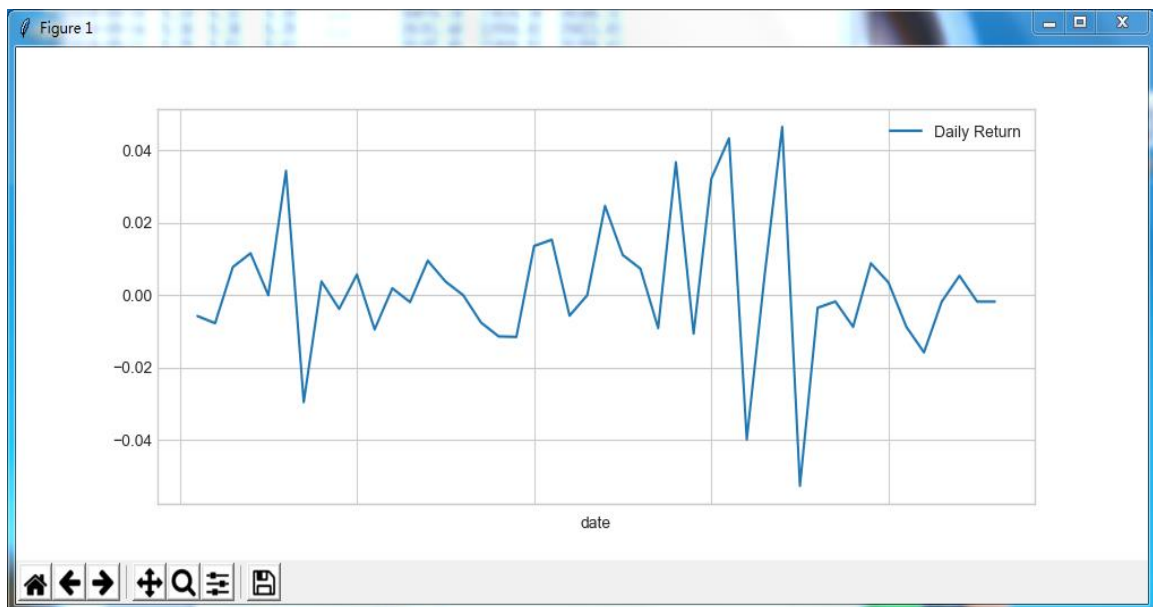


图 4.10 趋势线分析

趋势线分析的具体实现代码如下：

```
#趋势线分析
def datestr2date(s):
    s = s.decode('utf-8')
    return datetime.datetime.strptime(s, "%Y-%m-%d").date()

d, c, h, l = np.loadtxt(filepath, delimiter=',', skiprows=1,
                        usecols=(0, 3, 2, 4), unpack=True,
                        dtype={"names": ('date', 'close', 'high', 'low'),
                              "formats": (np.object_, np.float32,
                                           np.float64, np.float64)},
                        converters={0: datestr2date})

#绘制图形
plt.plot(d, c, 'y', label = "close")
plt.title('stock ' + filepath[:-4] + ' trend lines')
plt.legend()
plt.show()
```

(3) K线图分析：

股市中的 K 线图的画法包含四个数据，即开盘价、最高价、最低价、收盘价，所有的 k 线都是围绕这四个数据展开，反映大势的状况和价格信息。编写代

码时将对“open”、“high”、“low”、“close”这四列的数据进行切片分析，并进行图形绘制。

K线图的分析设计绑定了“K线图”按钮和多行日志文本框。程序正常运行的效果图如图 4.11 所示：

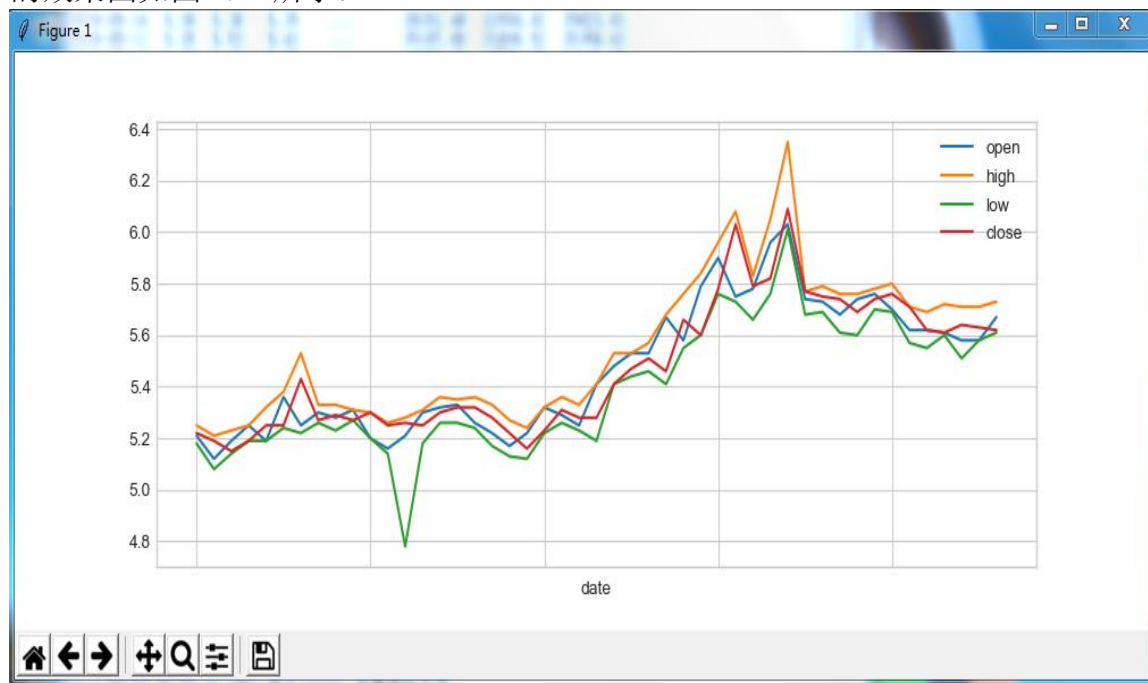


图 4.11 K 线图分析

K 线图分析的具体实现代码如下：

```
#k线图分析
def plot_K_line(filepath):
    data_list = []
    hist_data = ts.get_hist_data(filepath[:-4])
    for dates, row in hist_data.iterrows():
        date_time = datetime.datetime.strptime(dates, '%Y-%m-%d')
        t = date2num(date_time)
        open, high, low, close = row[:4]
        datas = (t, open, high, low, close)
        data_list.append(datas)
```

(4) 布林线分析：

布林线指标是股市技术分析的常用工具之一，通过计算股价的“标准差”，再求股价的“信赖区间”。布林线指标也是股票市场最实用的技术分析参考指标。

布林线的分析设计绑定了“布林线”按钮和多行日志文本框。程序正常运行的效果图如图 4.12 所示：

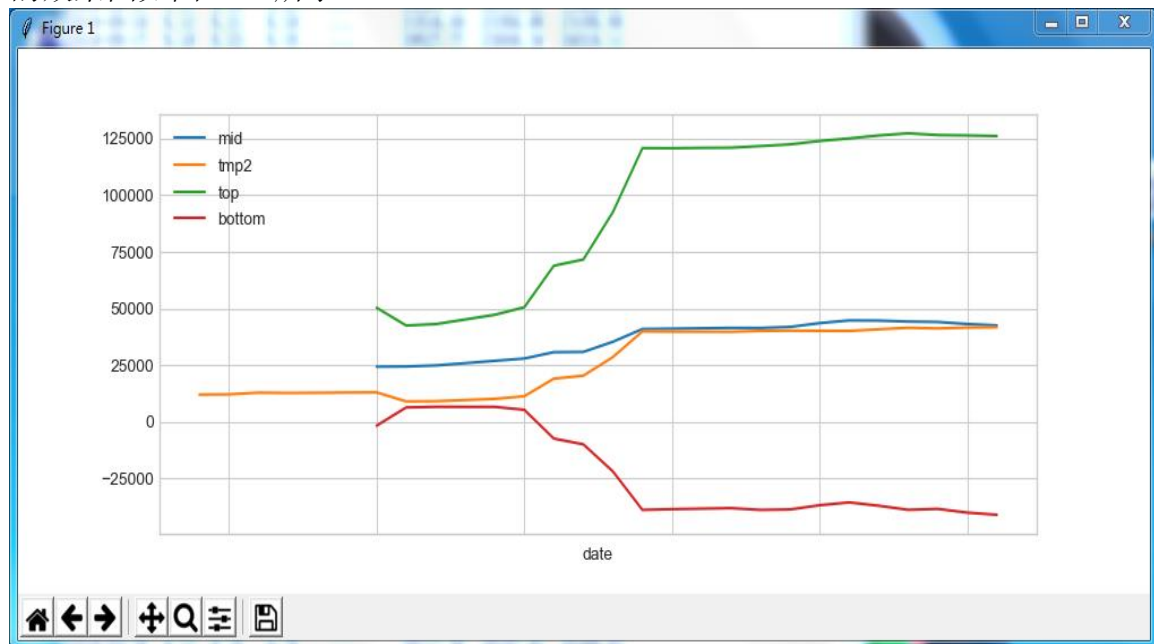


图 4.12 布林线分析

布林线分析的具体实现代码如下：

```
#计算布林线
N = 20 #20日移动平均计算的布林线
#平均权重
weights = np.ones(N)/N
#实现移动平均
sma = np.convolve(weights,c)[N-1:-N+1]
deviation = []
lenc = len(c)
for i in range(N-1,lenc):
    dev = c[i-N+1:i+1]
    deviation.append(np.std(dev))
deviation = 2 * np.array(deviation) #两倍标准差
upperBB = sma + deviation#压力线
lowerBB = sma - deviation#支撑线
```

(5) 股价分布直方图分析：

直方图可以将数据的分布可视化。我们要分析绘制的是股票价格直方图，需要对包含数据的数组以及柱形的数量进行设置。

股价分布直方图的分析设计绑定了“价格直方图”按钮和多行日志文本框。程序正常运行的效果图如图 4.13 所示：

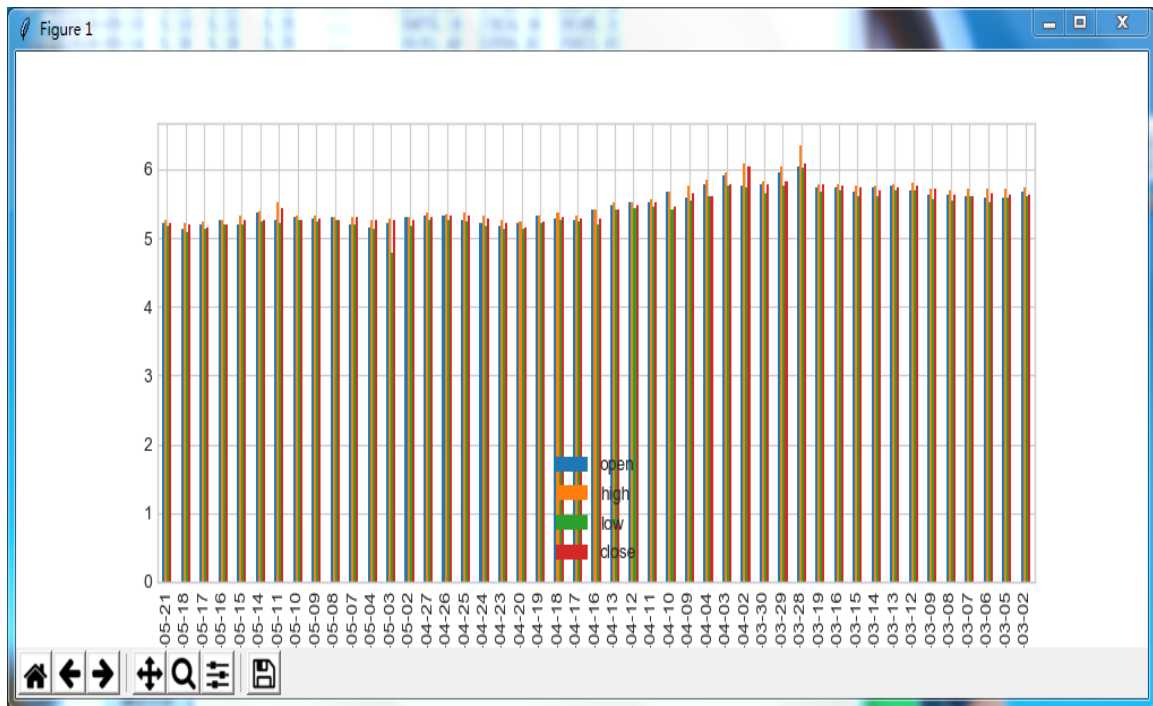


图 4.13 股价分布直方图分析

股价分布直方图分析的具体实现代码如下：

```
#股价分布直方图分析
def show_zhifangtu(self):
    self.stock[['open', 'high', 'low', 'close']].
    plot(legend=True, kind='bar', figsize=(10, 4))
    plt.show()
```

4.5 动态日志模块

4.5.1 动态日志模块设计

动态日志模块设计的目的是为了在用户激活事件驱动之后，捕捉动态日志打印呈现出来，提供给用户参考，并且能够让用户进行清空操作。动态日志模块的实现流程图如图 4.14 所示：

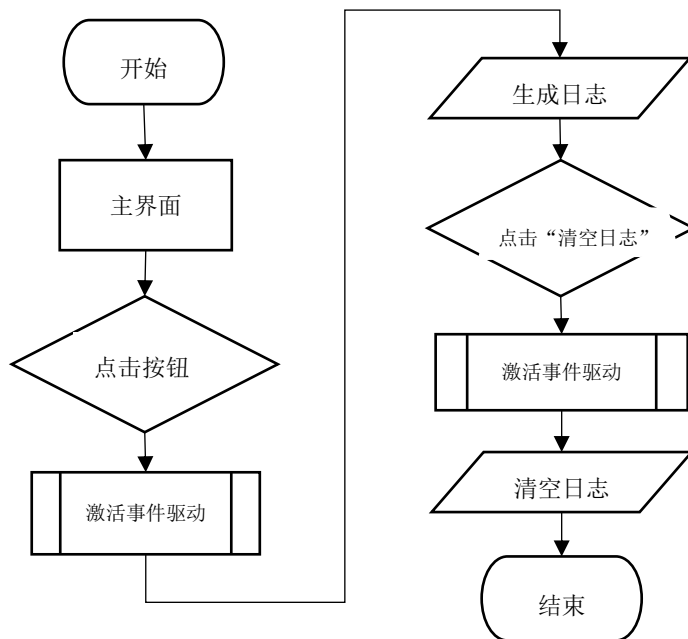


图 4.14 动态日志模块流程图

4.5.2 动态日志模块实现

用户每次激活事件驱动，系统都会响应并发送事件给用户。与此同时，系统也会将这一过程形成动态日志打印出来，仅供用户参考。用户也可以对动态日志进行清空操作。

动态日志模块包括多行日志显示文本框和“清空日志”按钮。用户点击数据下载模块或者数据分析模块中的任意一个按钮，激活相应的事件驱动后，系统会形成日志显示在多行文本框中；如果用户点击“清空日志”按钮，会激活日志清空的事件驱动，使得打印出来的日志动态全部清空。

程序正常运行的界面效果图如图 4.15 所示：

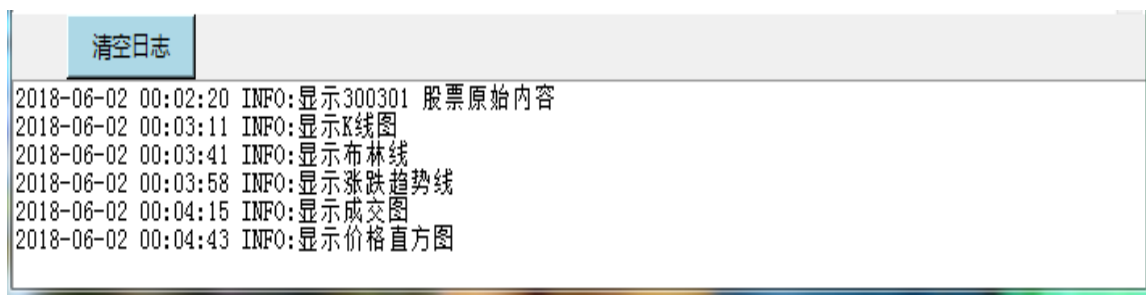


图 4.15 动态日志模块效果图

动态日志模块的具体实现代码如下：

```
# 获取当前时间
def get_current_time(self):

    current_time = time.strftime('%Y-%m-%d %H:%M:%S',
                                time.localtime(time.time()))

    return current_time

# 日志动态打印
def write_log_to_Text(self, logmsg):

    global LOG_LINE_NUM

    current_time = self.get_current_time()

    logmsg_in = str(current_time) + " " + str(logmsg) + "\n"

    if LOG_LINE_NUM <= 7:

        self.log_data_Text.insert(END, logmsg_in)

        LOG_LINE_NUM = LOG_LINE_NUM + 1
    else:
        self.log_data_Text.delete(1.0, 2.0)

        self.log_data_Text.insert(END, logmsg_in)

# 清空文本内容
def delete_Text(self):

    self.init_data_Text.delete(1.0, END)

# 清空日志
def delete_log_Text(self):

    self.log_data_Text.delete(1.0, END)
```

第五章 测试

5.1 测试目的与测试方法

5.1.1 测试目的

表面上来看，软件测试的目的与其他阶段的目的都相反，是竭力证明程序中有错误，不能按照预定要求正确工作的过程。但是暴露问题并不是软件测试的最终目的，发现问题是为了解决问题，测试阶段的根本目的是尽可能多地发现并排除隐藏的错误的，最终把一个高质量的软件系统交给用户使用^[2]。

G.Myers 给出了关于测试的一些规则，这些规则也可以看作是测试的目标或者定义：

- 1.测试是为了发现程序中的错误而执行程序的过程。
- 2.好的测试方案是极可能发现迄今为止尚未发现的错误的测试方案。
- 3.成功的测试则是找出了至今位置尚未发现的错误的测试。

测试是软件开发过程中非常重要的一个步骤，是使用人工操作或者软件自动运行的方式来检验它是否满足规定的需求或弄清预期结果与实际结果之间的差别的过程，目的是为了测试其运行结果是否满足预期的功能需求^[3]。

在应用软件设计完成之后都要经过多重的测试，以此来发现软件在整个设计过程中可能存在的问题同时及时解决问题。

5.1.2 测试方法

整个测试过程主要涵盖系统测试、单元测试、组装测试。

采用的测试方法主要是白盒测试和黑盒测试两种：

白盒测试：可以指导系统内部工作流程的测试过程，因此被称为结构测试或者逻辑驱动测，测试者参照源程序分析程序的内部结构而不考虑程序功能，只按照内部结构测试程序。

黑盒测试：可以把程序当作是一个被蒙住的黑盒子，因此又称为功能测试，测试过程中完全不需要考虑程序内部结构和内部特性，仅需要在程序接口中进行测试，故而只能检查出系统功能是否能够正常运行。

5.2 系统测试设计

5.2.1 系统测试对象

本章系统测试的对象是“股市数据分析工具”，该系统包含 EventManager 模块、数据下载模块、数据分析模块、动态日志模块。

5.2.2 测试执行

本次测试是严格地按照项目的计划和测试的计划执行的系统测试，该系统测试采用的是单元测试，集成测试，完善性测试等多种方式进行测试，结果显示所有功能都能得以实现，并且准时完成了测试的任务。本次测试的内容主要包括：功能测试、性能测试、安全测试和可用性测试四项。

5.3 功能测试

5.3.1 功能测试

功能测试对系统的适用性、准确性、可操作性、安全性这四项内容进行测试，测试之后的结果将汇总展示。

本系统功能测试结果如表 6.1 所示：

表 6.1 功能测试结果

测试内容	测试结果
安全性	好
适用性	好
准确性	好
可操作性	好

5.3.2 测试用例

本次测试主要分为主界面 EventManager 模块、数据下载模块、数据分析模块、动态日志模块这四个模块进行用例测试。

股市数据分析工具各模块测试用例表						
编号	功能模块	功能点	预期成果	是否与预期一致	测试结论	测试人
1	主界面 EventManager 模块	主界面可视化	呈现主界面，各个控件完好显示	是	通过	朱敏
2	数据下载模块	输入股票代码	文本框内能够显示所输入的代码	是	通过	朱敏
		点击“数据显示”按钮	获取到的数据能够显示在文本框里	是	通过	朱敏
		拉动滚动条	能够通过拉动滚动条查看到所有数据	是	通过	朱敏
3	数据分析模块	点击“成交量”按钮	弹出窗口显示成交量分析	是	通过	朱敏
		点击“趋势线”按钮	弹出窗口显示趋势线分析	是	通过	朱敏
		点击“K线图”按钮	弹出窗口显示K线图分析	是	通过	朱敏
		点击“布林线”按钮	弹出窗口显示布林线分析	是	通过	朱敏
		点击“价格直方图”按钮	弹出窗口显示股价分布直方图分析	是	通过	朱敏
4	动态日志模块	动态日志生成	在文本框中能显示日志动态	是	通过	朱敏
		点击“清空日志”按钮	清空日志	是	通过	朱敏

5.4 测试结果

经过对上述的测试结果分析，本课题所研发的股市数据分析工具符合预期的需求。测试结果为合格。

第六章 总结与展望

至此，基于 `numpy` 的股市数据分析工具的研发课题已经完成。毋庸置疑的，代码编写的整个过程中不可避免地遇到了各种各样的困难，但这些困难并没有使我打退堂鼓。在股市数据分析工具的研发课题中，对于以前从未接触过的 `Python` 编程语言进行了认真地学习。这一次课题研究让我完整地体验了系统分析设计到实现的过程，在这个过程中，我查阅了大量的资料，将所学到的所积累的理论知识运用到了真实的实践操作当中，对于系统开发有了更深刻的认识。首先参考了相关的基于 `Python` 和 `numpy` 实现数据分析的论文资料，经过一定的积累和调查，结合学习过的相关软件工程知识，对本课题进行了可行性分析和需求分析，在确定这个课题值得解决并且确定系统必须完成的工作之后，再深层次地进入到系统设计的环节，通过课题辅导老师的指导并结合自身不断地学习积累，最终实现了股市数据分析工具的开发。在这个过程中，我自身所存在的很多不足之处也逐渐显露出来。对于之前学到过的理论知识，自我感觉掌握得不错，有些相对基础的应用技能对于我而言也并不存在很大的难度，但是这些在实际的操作面前相形见绌，如果要着手解决实际问题的话，在具备一定理论知识储备的条件下，还是需要学会结合实际情况考虑的。经过为期四个月的努力学习和不解积累，完成了股市数据分析工具的开发。当然，该系统还存在有不成熟的地方，比如说一些技术上的缺陷，但是也在查阅中国知网上的相关文献和登录 `DSCN` 博客查找相关文章之后最终解决了问题。

毕业设计是为了验收我们作为学生思念在校学习的成果，考验的是我们的学习能力和对所学知识的运用的能力，因为需要通过不断的学习相关知识来解决接连出现的新问题，实实在在的让我感觉到在这个“问题出现—>查找相关资料—>解决问题”的循环过程中，自己各方面的能力都得到了锻炼。这次锻炼大大地提高了我学习新知识的能力和结合实际运用所学知识的能力，在不断解决出现的新问题的过程里我享受到了探索知识的乐趣和亲手解决困难后的满足感。

本次课题的开发，让我将四年里在学校课堂上所学相关计算机的理论知识与实际要解决的问题结合了起来，融会贯通，得到了很大的提升。既对计算机的相关知识进行了巩固加强，也从中学到很多在课堂上学不到的知识，同时还加强了 my 动手能力，培养了我独立思考的能力。通过本次课题的研究，我深刻地意识到软件开发人员的辛苦，但也体验到了问题得到解决后的激动喜悦的心情，使我面对问题的独立思考能力大大的增强，同时也培养了我结合实际将所学知识学以致用能力，为今后的去做专业相关的工作打下了基础。

致谢

在经过为期四个月的学习和积累之后，尽管过程中遇到了许多困难和不顺利的因素，但是在查阅了大量的相关资料以解决问题后，我的毕业设计还是如预期那样完成了。这次毕业设计不仅仅是校方领导及老师对于我们作为学生这四年来在学校学习的成果验收，更是我们学生对于自我所学知识的掌握水平和知识应用水平的评定。

在我的毕业设计即将结束的此刻，我要向徐远纯老师表达我的感谢，在他的悉心指导之下，我才能够有方向有针对性的完成课题所要求研发的股市数据分析工具的设计开发工作和相对应的毕业论文的书写工作。老师即使是在教学和科研任务很紧张繁重的情况之下，依然抽出时间对我进行毕业设计的辅导工作，指导我解决在毕业设计中出现的问题，

还推荐了相关的书籍与我参考，书籍是针对毕业设计课题的，对我而言很有参考价值，读了以后受益良多。除此之外，老师还会定期地让我们汇报毕业设计工作的进展程度，发现问题及时反馈给我们并指导我们修改的方向，在这课题的一整个设计与开发的过程当中，老师总是能够一针见血地提出宝贵的建议，他的建议和指导让我觉得受益匪浅，正是这种督促、指导和建议使得我的毕业设计工作和毕业论文的书写工作能够按时完成。

大学四年在学校里学习知识，我要感谢每一位传授我知识的老师。老师们认真负责，在课堂上对我们要求严格，兢兢业业勤勤恳恳地教导我们专业知识。与此同时，还在人格升华和品行修养上深深地影响着我们，教导我们无论身处何处都应该端正自己的态度以积极正确的心态去面对眼前的境况。谢谢各位老师！

我还要向我的同学们表达我的感谢，在我因为毕业设计遇到问题而情绪低落的时候，在我身边不断的鼓励我，让我能够在困难里坚持下来。

最后，我在此祝愿所有的老师和同学事业有成！也祝愿我的学校越来越好！

参考文献

- [1] Ivan Idris, 张驭宇. Python 数据分析基础教程 NumPy 学习指南 (第二版) [M]. 人民邮电出版社, 2014.
- [2] Micheal Dawson, 王金兰. Python 编程初学者指南[M]. 人民邮电出版社, 2014.
- [3] 张藩, 牟用敏. 软件工程导论 (第六版) [M]. 清华大学出版社, 2013
- [4] 任永功, 于戈. 数据可视化技术的研究与进展[J]. 计算机科学, 2004(12): 92-96.
- [5] 陈明. 大数据可视化分析[J]. 计算机教育, 2015(05): 94-97.
- [6] 黎爽. 基于 Python 科学计算包的金融应用实现[D]. 江西财经大学, 2017.
- [7] 肖旻, 陈行. 基于 Python 语言编程特点及应用之探讨[J]. 电脑知识与技术, 2014, 10(34): 8177-8178.
- [8] 嵩天, 黄天羽, 礼欣. Python 语言: 程序设计课程教学改革理想选择[J]. 中国大学教学, 2016(02): 42-47.
- [9] 高蓓. 基于 Python 的图表自动生成系统[D]. 山西大学, 2017.
- [10] 姜杉彪, 黄凯林, 卢昱江等. 基于 Python 的专业网络爬虫的设计与实现[J]. 企业科技与发展, 2016(08): 17-19.
- [11] 张茗芳. 动态语言 Python 探讨与比较[J]. 企业科技与发展, 2012(13): 57-60.
- [12] 严婷, 文欣秀, 赵嘉豪等. 基于 Python 的可视化数据分析平台设计与实现[J]. 计算机时代, 2017(12): 54-56.
- [13] 沈殊璇, 薄亚明. 适合于科学计算的脚本语言 Python[J]. 微计算机应用, 2002(05): 289-291.
- [14] 郭丽蓉. 基于 Python 的网络爬虫程序设计[J]. 电子技术与软件工程, 2017(23): 248-249.
- [15] 管华. 对当今 Python 快速发展的研究与展望[J]. 信息系统工程, 2015(12): 114+116.