****

**Python数据处理**

**实验报告**

|  |  |
| --- | --- |
| 学生姓名 | 李栩雅 |
| 学 号 | 8208201116 |
| 专业班级 | 计算机科学与技术2002 |
| 指导教师 | 王 斌 |
| 学 院 | 计算机学院 |
| 完成时间 | 2022.5.13 |

计算机学院

2022年6月

实验一 爬取股票数据

1. **实验要求**
   1. **实验目标**

开发网络爬虫在东方财富、新浪财经或者纳斯达克等财经网站上爬取一只股票的每天的开盘价，收盘价，最高价，最低价等信息，并存储在数据库中，并开发GUI应用可视化。

**1.2.基本要求**

（1） 掌握网络爬虫的开发方法；

（2） 掌握Python开发数据库的GUI界面；

（3） 掌握Matplotlib绘制股票的K线图；

1. **实验过程设计**

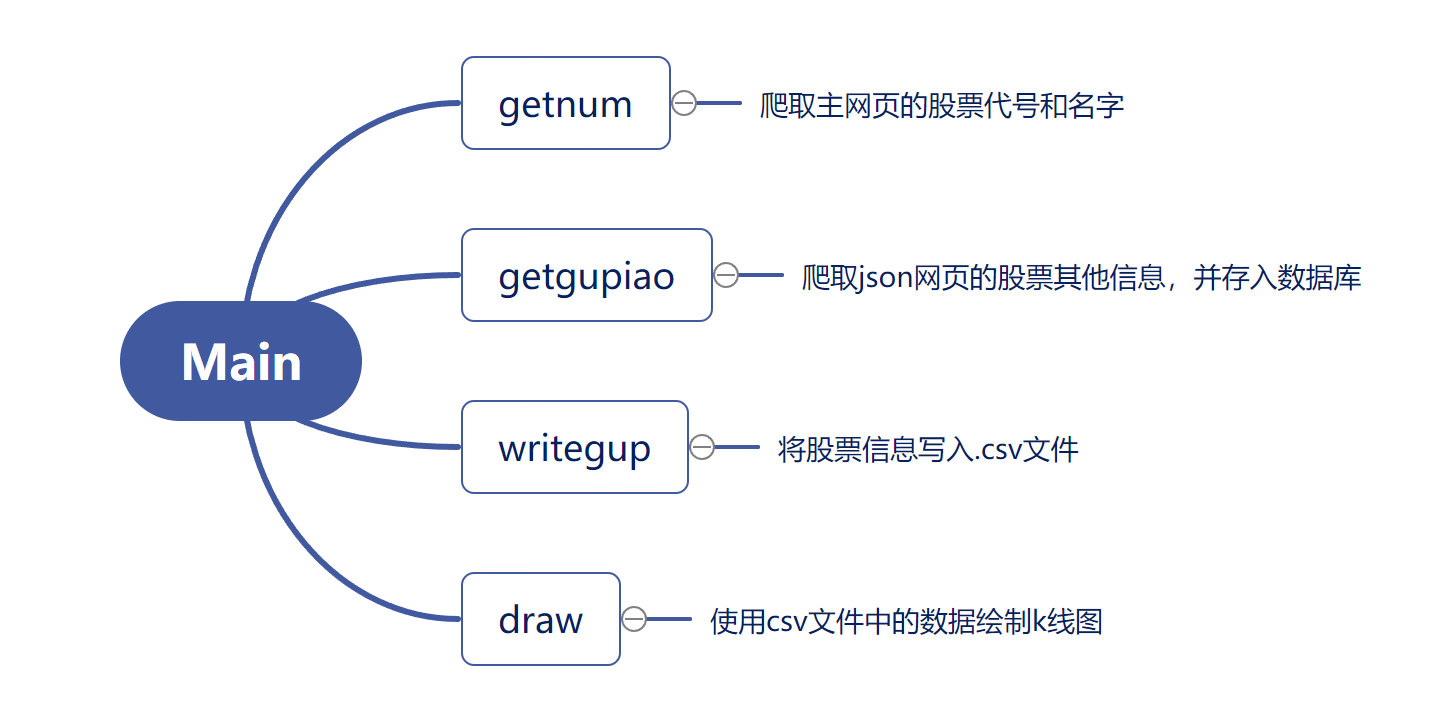
（1） 在互联网中寻找可以提取数据的网站；

（2） 构造网络爬虫框架以及正则表达式；

（3） 设计数据库表，建立数据库，将获取的数据存储到数据库中；

（4） 设计数据库访问语句，可以以表格的形式展现数据；（5） 可以将一只股票30天的数据用K线图展现。

1. **程序概要设计**



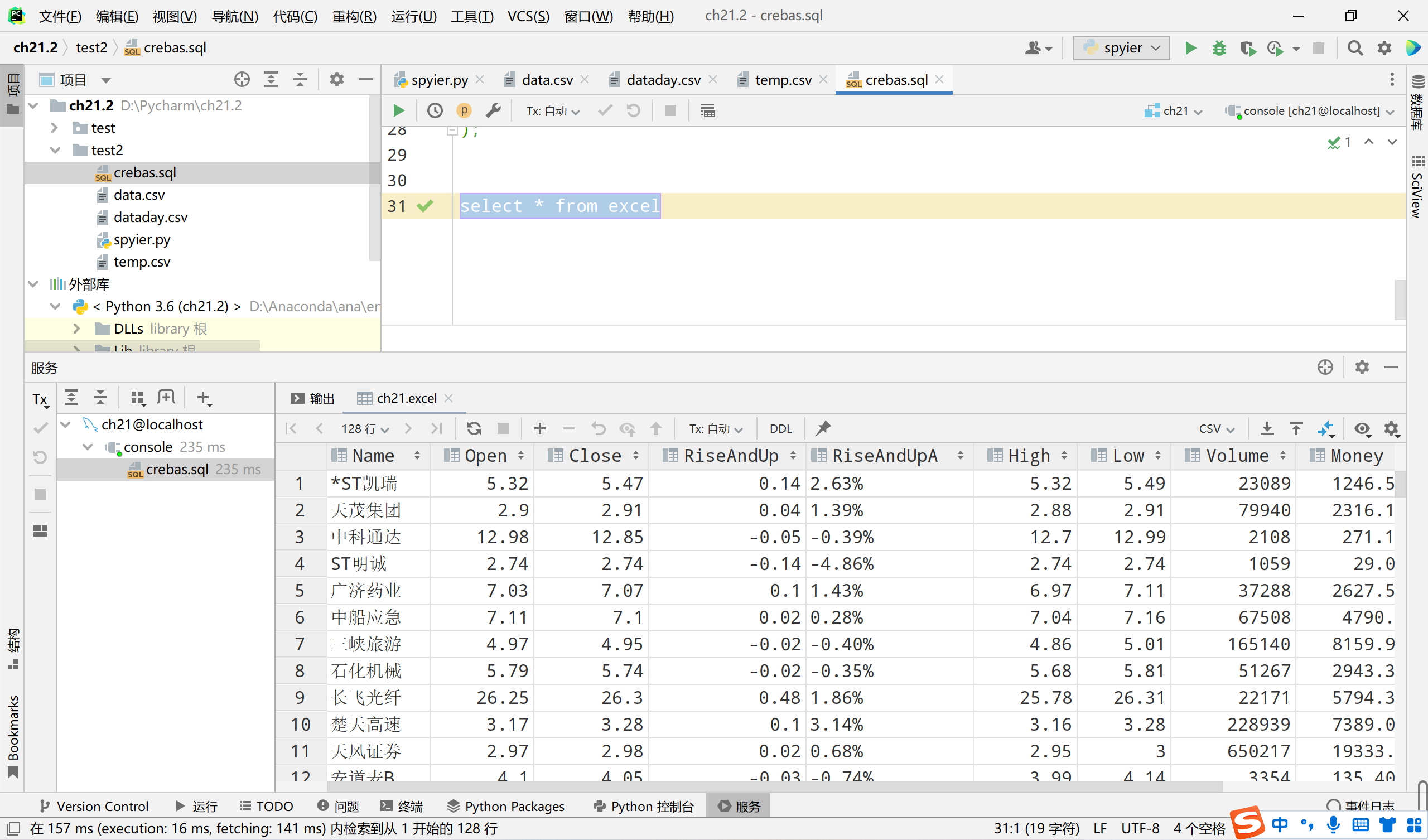
1. **程序设计**

import csv  
import time  
import requests  
from bs4 import BeautifulSoup  
import json  
import pymysql  
import matplotlib.dates as mdates  
import matplotlib.pyplot as plt  
import mplfinance  
import mpl\_finance  
import tkinter as tk  
import pandas as pd  
import numpy as np  
  
  
class LoginPage:  
 def \_\_init\_\_(self, master: tk.Tk):  
 self.root = master  
 self.root.geometry('400x240')  
 self.page = tk.Frame(root)  
 self.page.pack()  
 self.root.title('爬取搜狐财经股票数据')  
 tk.Label(self.page).grid(row=0, column=0)  
 tk.Button(self.page, text='点击爬取数据', command=self.getnum).grid(row=2, column=1, pady=10)  
 tk.Button(self.page, text='点击实现数据可视化', command=self.writegup).grid(row=3, column=1, pady=10)  
  
  
 def getnum(self):  
 html = requests.get("https://q.stock.sohu.com/cn/bk\_3137.shtml") # 获取想要的股票号码  
 html.raise\_for\_status  
 text = html.text  
 # print(text)  
 soup = BeautifulSoup(text, 'html.parser')  
 tdL1 = soup.find\_all('td', attrs={"class": "e1"})  
 tdL2 = soup.find\_all('td', attrs={"class": "e2"})  
 numL = []  
 for td1, td2 in zip(tdL1, tdL2):  
 try:  
 numL.append([td1.text, td2.text])  
 except:  
 continue  
 self.getgupiao(numL) # 返回所有股票号码  
  
 def getgupiao(self, numL):  
 connection = pymysql.connect(host='localhost',  
 user='root',  
 password='123456',  
 database='ch21',  
 charset='gbk',  
 autocommit=True)  
 print("数据库连接成功")  
 count = 0  
 for num in numL:  
 try:  
 count = count + 1  
 url = 'https://q.stock.sohu.com/hisHq?code=cn\_' + num[  
 0] + '&stat=1&order=D&period=d&callback=historySearchHandler&rt=jsonp&0.13888967033291877'  
 r = requests.get(url)  
 r.raise\_for\_status()  
 r.encoding = "gbk"  
 html = r.text[21:-2] # 去BOM头  
 data = json.loads(html)  
 datalist = data[0]['hq']  
 # print(num[1])  
 # print(datalist[0])  
 if count == 2:  
 f = open('dataday.csv', 'w', encoding='gbk')  
 csv\_write = csv.writer(f)  
 csv\_write.writerow(['日期', '开盘', '收盘', '涨跌额', '涨跌幅 ', '最低', '最高', '成交量(手)', '成交金额(万)', '换手率'])  
 for data in datalist:  
 csv\_write.writerow(data)  
 f.close()  
 try:  
 with connection.cursor() as cursor:  
 sql = 'insert into excel (Name, Open, Close, RiseAndUp, RiseAndUpA, High, Low, Volume, Money, TurnoverRate) values (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)'  
 value = (num[1], datalist[0][1], datalist[0][2], datalist[0][3], datalist[0][4], datalist[0][5],  
 datalist[0][6], datalist[0][7], datalist[0][8], datalist[0][9])  
 print(value)  
 affectedcount = cursor.execute(sql, value)  
 # sql = 'insert into user (userid, name) values (%s,%s)'  
 print('影响的数据行数：{0}'.format(affectedcount))  
 connection.commit()  
 # print(num[1]+'成功')  
 except pymysql.DatabaseError as error:  
 connection.rollback()  
 print('插入数据失败' + error)  
  
 '''datalist = data[0]['hq']  
 with open(num[1]+'.csv', "w", newline='') as csvFile: # 写入股票信息  
 csvWriter = csv.writer(csvFile)  
 csvWriter.writerow(['日期', '开盘', '收盘', '涨跌额', '涨跌幅 ', '最低', '最高', '成交量(手)', '成交金额(万)', '换手率'])  
 for data in datalist:  
 csvWriter.writerow(data)  
 break  
 csvFile.close  
 print(num[1], '爬取成功')'''  
  
 except:  
 continue  
  
 connection.close()  
  
 def pot\_candlestick\_ohlc(self, datafile):  
 """绘制K线图"""  
  
 # 从CSV文件中读入数据DataFrame数据结构中  
 quotes = pd.read\_csv(datafile,  
 index\_col=0,  
 parse\_dates=True,  
 infer\_datetime\_format=True)  
  
 # 绘制一个子图，并设置子图大小  
 fig, ax = plt.subplots(figsize=(10, 5))  
 # 调整子图参数SubplotParams  
 fig.subplots\_adjust(bottom=0.2)  
  
 mpl\_finance.candlestick\_ohlc(ax, zip(mdates.date2num(quotes.index.to\_pydatetime()),  
 quotes['Open'], quotes['High'],  
 quotes['Low'], quotes['Close']),  
 width=1, colorup='r', colordown='g')  
  
 ax.xaxis\_date()  
 ax.autoscale\_view()  
 plt.setp(plt.gca().get\_xticklabels(), rotation=45, horizontalalignment='right')  
  
 plt.show()  
  
 def writegup(self):  
 conn = pymysql.connect(host='localhost',  
 user='root',  
 password='123456',  
 database='ch21',  
 charset='gbk',  
 autocommit=True)  
 cursor = conn.cursor()  
 sql = " select \* into outfile '{0}' CHARACTER SET GBK FIELDS TERMINATED BY ',' LINES TERMINATED BY '\\r\\n' from(select '日期', '开盘', '收盘', '涨跌额', '涨跌幅 ', '最低', '最高', '成交量(手)', '成交金额(万)', '换手率' union select \* from excel) as b "  
 try:  
 cursor.execute(sql.format("D:/Pycharm/ch21.2/test2/data.csv", ))  
 conn.commit()  
 except Exception as e:  
 print(e)  
 conn.rollback()  
 conn.close()  
 self.draw()  
  
 def draw(self):  
 result\_set = pd.read\_csv("dataday.csv", error\_bad\_lines=False, encoding='gbk')  
 data = []  
 col\_1 = result\_set["日期"]  
 data\_1 = np.array(col\_1)  
 col\_2 = result\_set["开盘"]  
 data\_2 = np.array(col\_2)  
 col\_3 = result\_set["收盘"]  
 data\_3 = np.array(col\_3)  
 col\_4 = result\_set["最低"]  
 data\_4 = np.array(col\_4)  
 col\_5 = result\_set["最高"]  
 data\_5 = np.array(col\_5)  
 col\_6 = result\_set["成交金额(万)"]  
 data\_6 = np.array(col\_6)  
 data.append(data\_1)  
 data.append(data\_2)  
 data.append(data\_3)  
 data.append(data\_4)  
 data.append(data\_5)  
 data.append(data\_6)  
 # print(data[4][0])  
 colsname = ['Date', 'Open', 'High', 'Low', 'Close', 'Volume']  
 # 临时数据文件名  
 datafile = 'temp.csv'  
 with open(datafile, 'w', newline='', encoding='utf-8') as wf:  
 writer = csv.writer(wf)  
 writer.writerow(colsname)  
 rows = []  
 for col in range(0, 80):  
 for row in range(0, 6):  
 rows.append(data[row][col])  
 # print(data[row][col])  
 row += 1  
 col += 1  
 print(rows)  
 writer.writerow(rows)  
 rows = []  
 self.pot\_candlestick\_ohlc(datafile)  
  
 '''datafile = 'temp.csv'  
 # 写如数据到临时数据文件  
 with open(datafile, 'w', newline='', encoding='utf-8') as wf:  
 writer = csv.writer(wf)  
 writer.writerow(colsname)  
 '''  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 root = tk.Tk()  
 LoginPage(master=root)  
 root.mainloop()  
 # numL = getnum()  
 # getgupiao(numL)  
 # writegup()  
 # draw()  
 #print("爬取完成！")

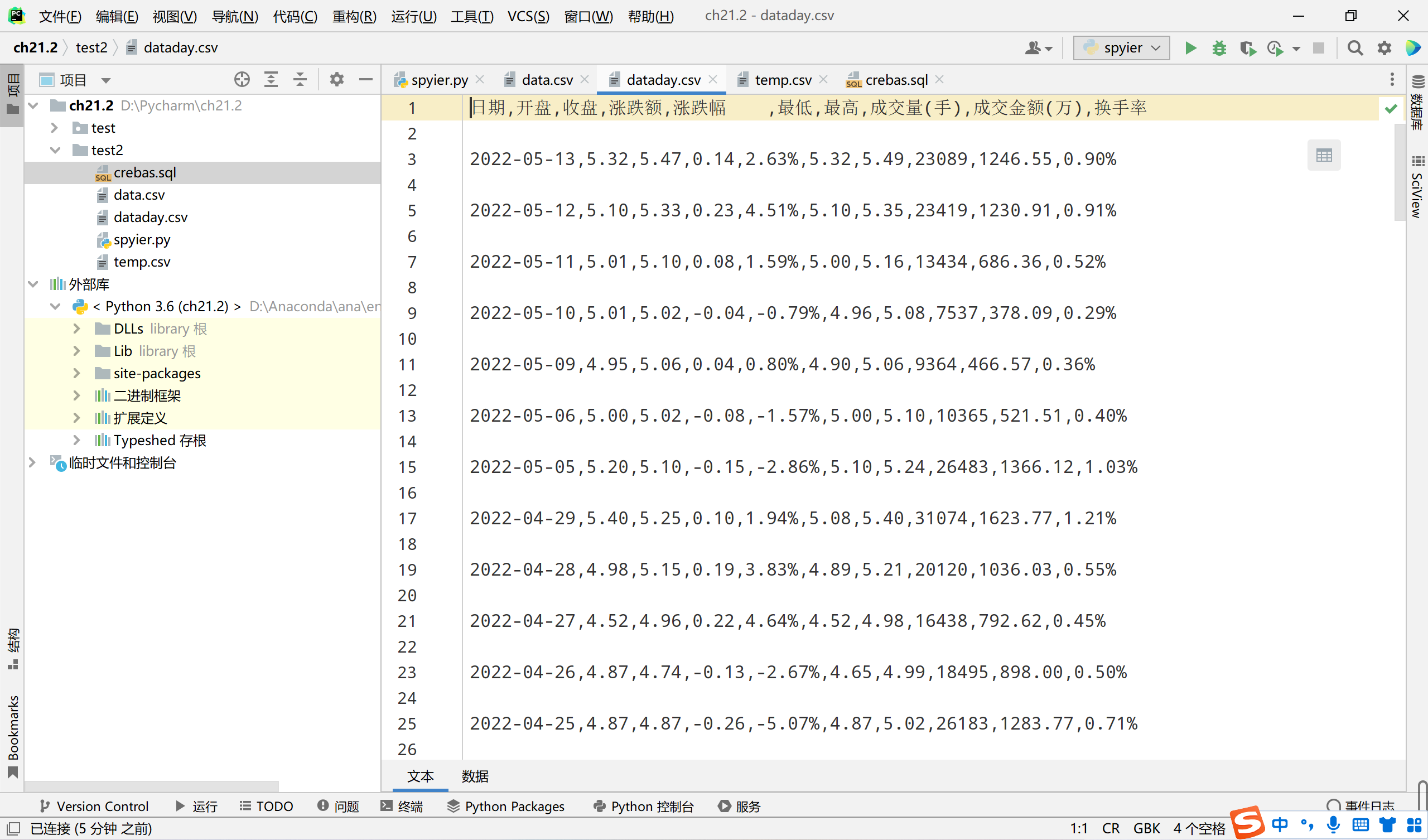
1. **运行结果展示**



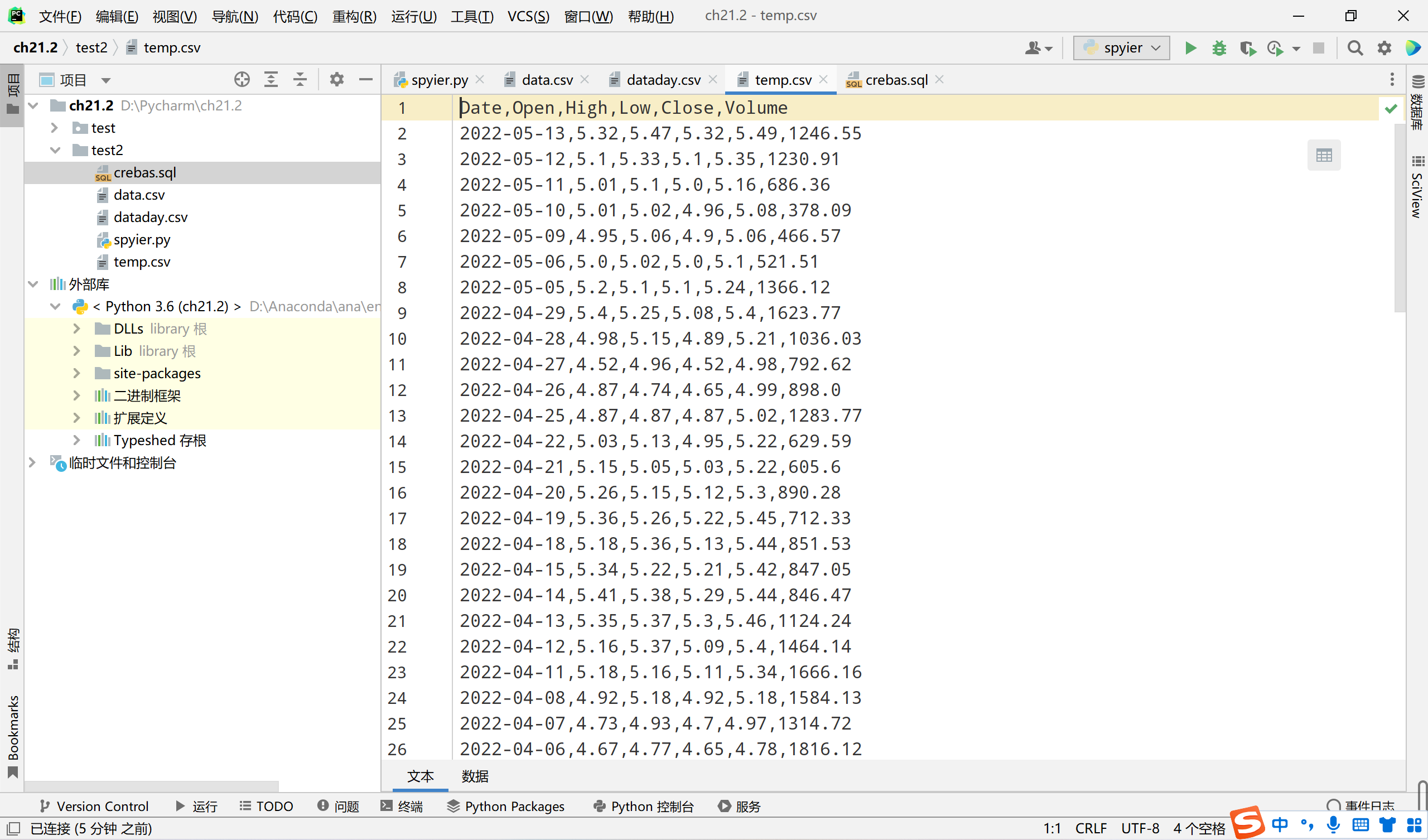
**可视化交互界面**



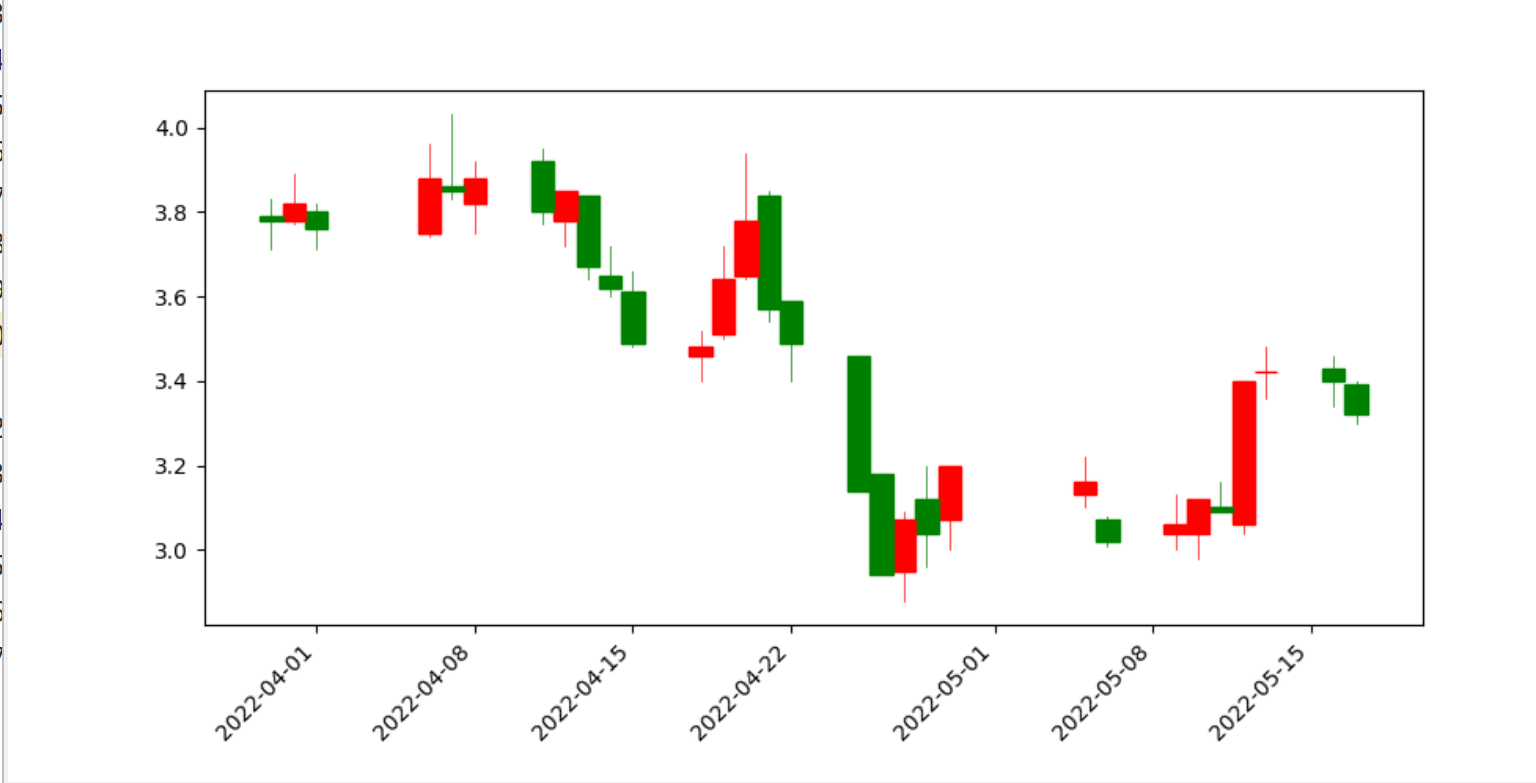
**数据库查询结果**



**随机选取的一支股票近80天的数据**



**用于绘制可视化图表的数据**



**绘制的k线图**

1. **心得体会**
   1. 遇到的问题

1）之前没接触过python，发现语法比其他高级语言简练很多，一开始很是陌生，重新打开教程学了一遍，然后才开始做实验。

2）一开始代码的思路不是很清晰，后面查阅了相关的博客了解到了，画出流程图然后写了代码。

3）对于爬虫还比较陌生，发现需要自学一部分html的框架知识

* 1. 收获

通过这次实验，我掌握了简单的python爬虫方法，也感受到了各种爬虫的妙用，以后可以利用这个技术干更多有趣的事，同时也让自己对python的各种数据类型更加熟练，运用的更加得心应手，还学会了不少新技能，比如使用mysql来存储数据。我也通过这次实验捡回了一些以前前端html的知识，发现了python这门高度封装的语言的便利性。以前有学过数据可视化的语言，不过是js，这次使用python也有了更多不一样的体验。