江西财经大學
JIANGXI UNIVERSITY OF FINANCE AND ECONOMICS

课程名称：Python语言与数据分析

课 程 报 告

项目名称 沪深股票数据分析

班　　级 金融201

学　　号 0204787

姓　　名 朱梓钰

任课教师 肖　泉

开课学期： 20 至 21 学年 第 1 学期

完成时间： 21 年 7 月 4 日

《沪深股票》数据分析报告

目录

1 概述

中国股票市场是由三部分组成：A 股市场、B 股市场和 H 股市场。A 股、B 股和 H 股的主要区别在于计价和发行方式的不同。相比较于 A 股，B 股同样是以人民币定价且在境内证券交易所上市交易，所不同的是，B 股需以外币认购和买卖，且主要针对国外及港澳台地区的组织机构和个人。H 股指的是在香港证券市场上市、使用港币交易的股票。

1990 年底创建的 A 股市场无论是上市公司的数量，还是市场的总市值，都是中国股票市场当之无愧的代表。A 股的正式名称是人民币普通股票。它是由我国境内的公司发行，供境内机构、组织、或个人（不含台、港、澳投资者）以人民币认购和交易的普通股票。A 股没有实物股票，采用无纸化电子记账，实行“T+1”的交割制度，有涨跌幅 10%的限制。

中国的 A 股市场经过多年的发展，在各行各业内均出现了一些业绩优良、红利优厚、交易活跃的大公司股票，他们在各自的细分行业内占有支配性地位，被称为“蓝筹股”(Blue Chips)。这些大公司股票由于管理水平高、收益稳定，即使在行业不景气的时候，也有能力维持稳定的收益，因此风险相对较小。

其中沪深股票，本报告旨在分析沪深股票的基础数据如股票代码、名称、上市日期、退市日期以及更高级的收益、风险、峰度、偏度等，为各位读者日后的股票选择提供一个新的思路启发。

2 数据描述

（1）数据准备

本报告数据来源于 Tushare pro 财经数据借口 https://waditu.com/document/2

输入参数

名称	类型	必选	描述
is_hs	str	N	是否沪深港通标的，N否 H沪股通 S深股通
list_status	str	N	上市状态 L上市 D退市 P暂停上市，默认是L
exchange	str	N	交易所 SSE上交所 SZSE深交所
ts_code	str	N	TS股票代码
market	str	N	市场类别
limit	int	N	
offset	int	N	
name	str	N	名称

输出参数

名称	类型	默认显示	描述
ts_code	str	Y	TS代码
symbol	str	Y	股票代码
name	str	Y	股票名称
area	str	Y	地域
industry	str	Y	所属行业
fullname	str	N	股票全称
enname	str	N	英文全称

（2）数据缺失情况

导入数据后，判断数据值是否有所缺失

```
In [6]: runfile('C:/Users/Administrator
Administrator/.spyder-py3')
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4369 entries, 0 to 4368
Data columns (total 6 columns):
ts_code      4369 non-null object
symbol       4369 non-null object
name         4369 non-null object
area         4367 non-null object
industry     4368 non-null object
list_date    4369 non-null object
dtypes: object(6)
memory usage: 204.9+ KB
None
```

如图所示，显示没有缺失，对数据分析不会产生影响

（3）数据清洗

数据清洗是数据分析的基础，也是非常重要的一步，因为数据清洗在提高数据质量的同时也可以避免脏数据影响分析结果。

所谓数据清洗，实际上就是对缺失值、异常值的删除处理或填充处理，以及为了方便数据的获取和分析，对列名的重命名、列数据的类型转换或者是排序等操作。

查看异常值，Pandas 的 describe( )可以用来统计数据集的集中趋势，结果如下：

```
In [7]: runfile('C:/Users/Administrator/.spyder-py3/temp.py', wdir='C:/Us
Administrator/.spyder-py3')
          ts_code  symbol  name  area industry list_date
count        4369    4369  4369  4367     4368      4369
unique       4369    4369  4369    32      110      2238
top     600475.SH  002692  银河电子    浙江     软件服务  20091030
freq            1       1     1   558      240        27
```

表明存在银河电子这一异常值的存在，所以，我们在分析的时候，应该舍弃这一异常值，避免带来干扰。

（4）数据整理

由于许多分析的维度都是建立在时间序列的基础上的，所以，将字符串类型的数据改为时间戳类型的数据可以便于我们分析。

Index		industry	list date	list dare
0		银行	19910403	1991-04-03 00:00:00
1		全国地产	19910129	1991-01-29 00:00:00
2		软件服务	19910114	1991-01-14 00:00:00
3		环境保护	19901210	1990-12-10 00:00:00
4		区域地产	19920427	1992-04-27 00:00:00
5		酒店餐饮	19920413	1992-04-13 00:00:00
6		运输设备	19920507	1992-05-07 00:00:00
7		综合类	19910625	1991-06-25 00:00:00
8		建筑工程	19951027	1995-10-27 00:00:00
9		区域地产	19920330	1992-03-30 00:00:00
				1992-02-28

如图所示，所有的时间已经转为了时间戳格式，便于数据的截取。

3 数据分析内容

（1）股票收益率

收益率的计算公式：R=(P2-P1)/P1,其中 P2 为后一天价格，P1 为前一天价格。股票收益是股票投资者以买卖股票的方式所取得的收益，也称为股利。股票收益率是反映股票收益水平的指标。因此这项分析很有意义，我们可以利用这项数据的分析结果，在预算有限的情况下，做出最优选择，得到最高收入。

采用 Tushare 内置函数 get_k_data()来获取股票的日线数据，在计算收益率时，把 close 作为收盘价格 P。

（2）单支股票和市场平均收益率比较

股票市场的平均收益率通常是指股票价格指数的收益率，目前 A 股指数有上证指数、深圳成分指数,沪深 300 指数、创业板指数等十几种,一般将股票与所属市场指数进行对比，以 6000000 浦发银行与上证指数在近 3 年的月收益率和累计收益率的比较为例，程序主要分为 4 个步骤，第一步，分别下载股票和指数的月线数据；第二步，合并上证指数和浦发银行的股票数据，因为单只股票存在停牌的情况，所以，以上证指数的交易日为基准来合并两个数据列，第三步，分别计算股票和指数的月收益率及其累计收益率；第四步，绘制图形，进行比较。

（3）股票 k 线图

不同种类财经数据的结构与变化特征是有差别的，适合描述不同种类数据特征的图自然也会不同。线图和点图是金融分析者最常用的二维图，因为三维线图

与点图比较容易展示金融数据的变化特征，并且绘制方法也较简单。本部分绘制K线图（也称蜡烛图）来分析600000浦发银行的股票。

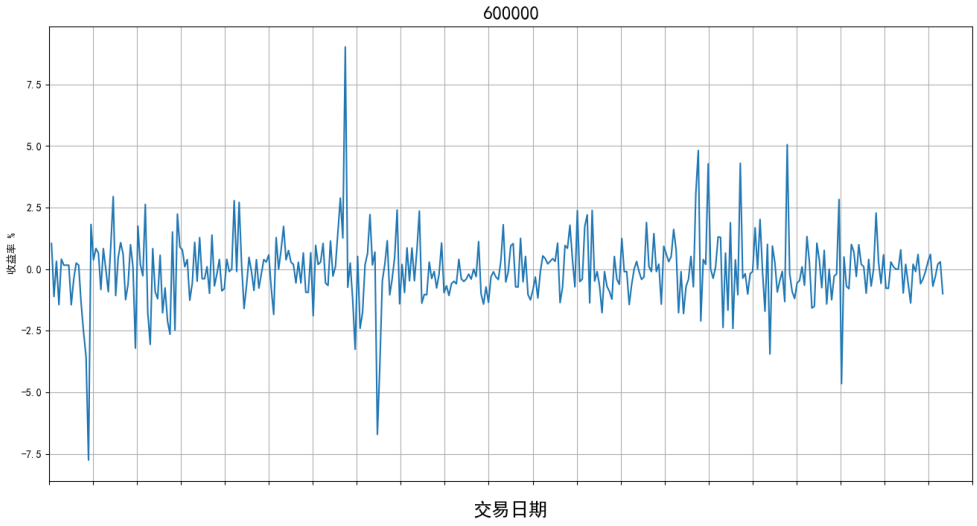
（4）上市公司占比分析

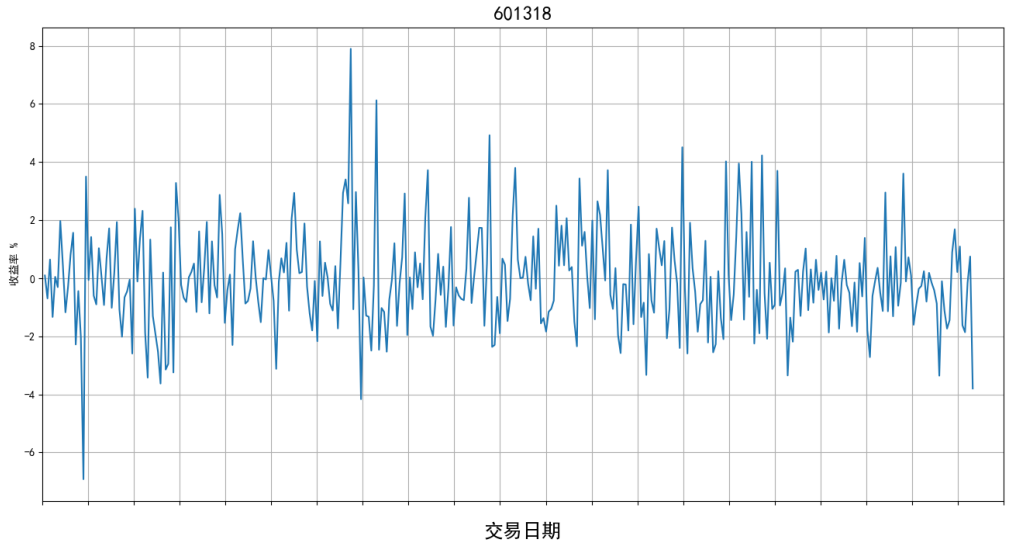
输出参数

名称	类型	默认显示	描述
ts_code	str	Y	股票代码
exchange	str	Y	交易所代码，SSE上交所 SZSE深交所
chairman	str	Y	法人代表
manager	str	Y	总经理
secretary	str	Y	董秘
reg_capital	float	Y	注册资本
setup_date	str	Y	注册日期
province	str	Y	所在省份
city	str	Y	所在城市
introduction	str	N	公司介绍
website	str	Y	公司主页
email	str	Y	电子邮件
office	str	N	办公室
employees	int	Y	员工人数
main_business	str	N	主要业务及产品
business_scope	str	N	经营范围

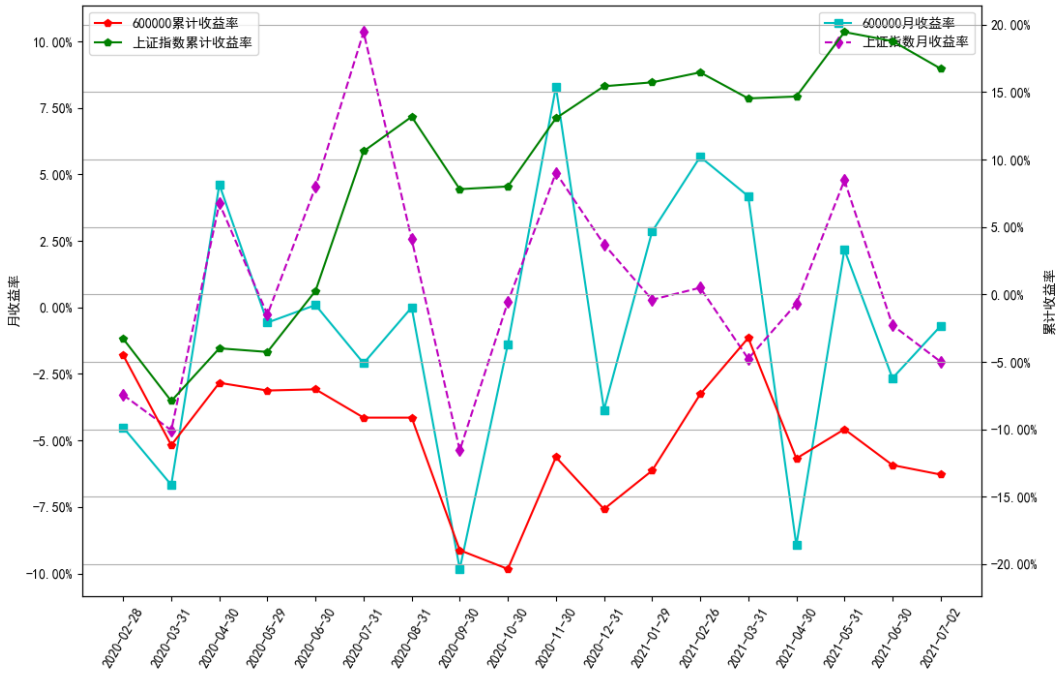
4 数据分析图表

（1）股票收益率



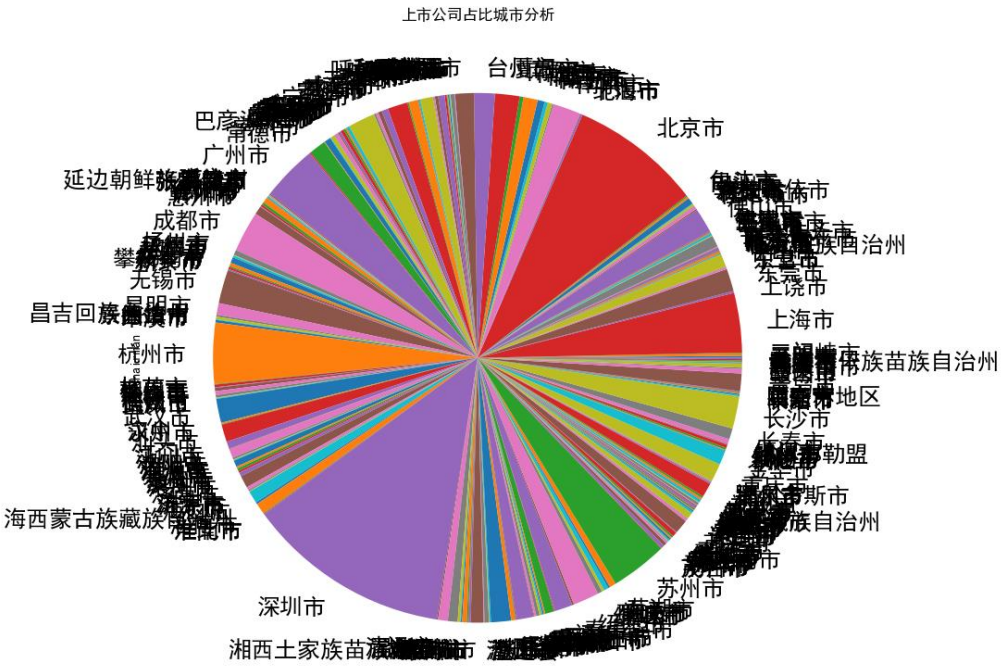


（2）单支股票和平均收益率分析



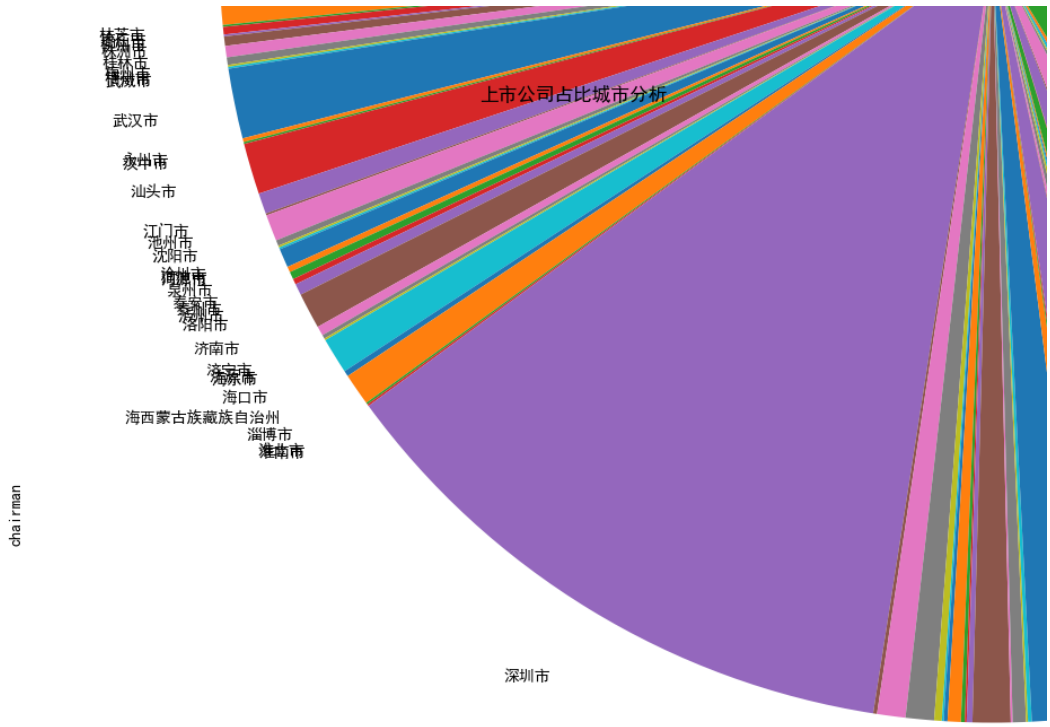
（3）股票 k 线图（以 600000 浦发银行为例）

（4）上市公司占比分析



（因为城市非常多，故图片中的城市名字有很多重叠，不太清晰）

下图为一张放大的图片



5 数据分析结果

（1）股票收益率

以 600000 浦发银行为例，从图上可以看出，浦发银行的股票收益率基本保持在 1%左右，且发展较稳，非常友好，适合股票新手买入。

而此时对比另一支，以 601318 中国平安为例，从图中可以看出中国平安的股票波动更大，且密集，但收益率更高，同时风险也更高，适合有一定股票基础的买者。

因此，对于购买股票的建议是，一定要根据自己的实际能力情况购入，不要过高也不要过低估计自己，这样才能做出最有利的选择。

（2）单支股票和平均收益率分析

通过以上图表可以看出，600000 浦发银行的月收益率波动较大，近期亏损较多，其中在 2020 年 9 月 30 日的月收益率达到最低，为-10%，通过红线可以看出，浦发银行的累计收益率在近 1 年中也是负数，低于 0%，情况不太乐观；观察上证指数的累计收益率发现，虽然不是一直为正，但是，它在持续走高，情况令人欣慰，但其月收益率波动也较大，其中，月收益率最高点在 2020 年 7 月 31 日，达到了 10%，截止到昨天，其月收益率在不断攀升。

因此，购买股票有风险，收益率也是一直在不停波动的。

（3）股票 k 线图

如图所示，可以看到浦发银行 2020-01-01 至今的一个 k 线图，绿色 k 线图代表收盘价低于开盘价，红色 k 线图代表收盘价高于开盘价，股价上涨。通过红色与绿色的对比分析，我们可以很好地看到浦发银行的股价变化，基本来看，红绿分布较均匀，特别是在今年的 5

月上旬，股价持续上涨，取得收益效果较好。

（3）上市公司城市占比分析（以沪深股票为例）

从饼状图中可以看出，沪深股票的深交所的上市公司大多集中在深圳、北京等发达城市，其中深圳占比约为 12.5%，北京占比约为 7%，其次，杭州、上海、广州、苏州等地的占比也非常大，上市公司也很多，由此可以看出，股票公司的上市大多集中在沿海、较发达的城市，经济繁荣，利于后续发展。

当我们想买入一支我们不太熟悉的股票的时候，我们可以参考它的上市公司分布状况来得出一些决定性策略。如果它的上市公司基本分布在较发达的城市，那么我们可以说，它的未来发展前景应该是广阔的，可以考虑购入。

6 总结

在整个数据分析过程中，我自我认为，我的分析报告完成过程还是非常曲折的，耗时长、工作量大，但是，如果用一句话来描述我最终的感受，我认为是值得的！

我首先从数据的收集整理开始做起，我先是在企图在阿里云天池平台获取有关于数据，可是，我发现我不太会运用阿里云天池获取数据，于是，我改成了使用 Tushare 来分析沪深股票的相关数据。我先选好了几个分析指标，我选择的是股票收益率分析、单支股票和市场平均收益率比较，股票 k 线图分析以及股票上市公司城市占比分析，选好分析指标之后，我进行了基本的数据清洗、数据缺失情况处理等前期操作，这些操作我运用的还可以。

在接下来的数据指标分析过程中，的确是有些困难，但是，我并没有轻易放弃，通过网上搜寻报错原因，查找书籍资料，修改自己的代码，在一步步报错中修正代码，比如说，在分析单支股票和市场平均收益率时，在重命名之后，我试图用 merge 来将两张 DataFrame 二维表做一个连接，可是，我发现，无论我怎么尝试都不行，于是，我查找书籍后，发现了另一种连接方法，我还可以使用 insert 将我要连接的表的数据直接插入到另一张表中，最后成功了。画出图形的那一刻，我还是非常的激动的，看着自己画出来的图，我体会到了 python 这门学科的魅力，虽然分析不出来的时候很头疼，但是一旦做出来了就会令人很欣喜。

在这次运用 python 来分析沪深股票的过程中，我对于股票的了解也更多了，作为一名金融专业的学生，我感到非常荣幸能够用我的所学来分析金融相关股票的数据。同时，在 python 的运用中，我不仅巩固了我已经学习的知识，比如在运用 groupby 进行分组的时候，我分完组发现又无法得到分组后的结果时，我猛然想起，我在之前也犯过这样的错误，我又重新温习了一遍，groupby 所运用的分组，只是一个形式而已，它还并未完全分组，当你取它的相关属性的时候，它就真正的分组了，相信我以后应该对于这个知识点印象非常深刻了。同时，我还通过网络的查询，学习到了非常多的其他的代码知识。我感叹于编程人员的聪慧，同时，也深感自己所学习到的还是非常皮毛的一部分基础知识而已。但是，我在 python 学习的过程中感到非常的快乐，对于自己每一点的成长进步都欣喜不已。

因此，在接下来的暑假当中，我不会荒废我的 python ，我会继续运用 python 做一些与专业相关的金融数据分析，不断练习基础代码，为今年 9 月份的计算机二级做准备！希望在未来的人生当中，我也能够运用 python 做出与金融专业相关的数据分析！

附录-数据分析代码

```
import tushare as ts
import pandas as pd
pro = ts.pro_api()
data=pro.stock_basic(exchange='',list_status='L',fields='ts_code,symbol,name,area,industry,list_da
```

```
te')#获取数据
df = pro.trade_cal(exchange='', start_date='20200101', end_date='20210702')
df1=pro.namechange(ts_code='600848.SH',fields='ts_code,name,start_date,end_date,change_reas
on')#获取股票曾用名
df2 = pro.hs_const(hs_type='SH')
df3 = pro.hs_const(hs_type='SZ')#获取沪深股票成分股
data['list_dare']=pd.to_datetime(data['list_date'])#修改时间格式，将其转为时间戳形式
print(data.dtypes)
```

（1）股票收益率部分代码

```
import tushare as ts
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import ticker
plt.rcParams['font.sans-serif']=['SimHei']#设置黑体
plt.rcParams['axes.unicode_minus']=False#正常显示正负号
startday='2020-01-01';endday='2021-07-02';tscode='601318'
data1=ts.get_k_data(tscode,start=startday,end=endday)#取得日线数据
data1.to_excel('601318.xlsx',index=False)
df4=pd.read_excel('601318.xlsx',dtype={'code':'str'})
df4['ret']=np.round((df4['close']-df4['close'].shift(1))/df4['close'].shift(1),6)
df4=df4.dropna()
date_tickers=df4['date'].values
def format_date(x,pos):
    if x<0 or x>len(date_tickers)-1:
        return' '
        return date_tickers[int(x)]
fig,ax=plt.subplots(figsize=(16,9))
fig.subplots_adjust(bottom=0.2)
ax.set_xlim([0,len(date_tickers)+12])#设置 X 轴范围
ax.plot(df4['ret']*100,label='收益率')
ax.xaxis.set_major_locator(ticker.LinearLocator(22))
ax.xaxis.set_major_formatter(ticker.FuncFormatter(format_date))
plt.title(tscode,fontsize=18)
plt.xlabel('交易日期',fontsize=18)
plt.ylabel('收益率 %')
ax.grid(True)
plt.show()
```

（2）单支股票和市场平均收益率比较

```
import tushare as ts
import numpy as np
```

```
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import ticker
plt.rcParams['font.sans-serif']= ['SimHei'] #  正常显示中文标签
plt.rcParams['axes.unicode_minus']= False #  正常显示负号
startday = '2020-01-01';endday='2021-07-03';tscode ='600000'; tsindx ='sh'
df1=ts.get_k_data(tscode,start=startday,end =endday,ktype='M')
df2=ts.get_k_data(tsindx,start=startday,end =endday,ktype ='M')
df1.to_excel('600000.xlsx',index=False)
df2.to_excel('sh000001.xlsx', index=False)
df1 =pd.read_excel('600000.xlsx',dtype={'code':'str'}) #  从文件读取
df2=pd.read_excel('sh000001.xlsx',dtype={'code':'str'})
df = df2[['date', 'close']].copy() #  指数数据
df.rename(columns={'close':'indclose'}, inplace = True) #  重命名为 indclose
m=df1['close']
df.insert(2,'close',m)
df.fillna(method='ffill',inplace=True)
df.fillna(method='bfill',inplace=True)
df['stk_log_ret']=np.round(np.log(df['close']/df['close'].shift(1)),4)#月收益率
df['ind_log_ret']=np.round(np.log(df['indclose']/df['indclose'].shift(1)),4)
df['stk_log_ret_cum']=df['stk_log_ret'].cumsum()
df['ind_log_ret_cum']=df['ind_log_ret'].cumsum()#累计收益率
#绘制收益率曲线
fig,ax=plt.subplots(figsize=(12,9))
fig.subplots_adjust(bottom=0.2)
ax.plot(df.date,df['stk_log_ret']*100,'-cs',lw=1.5,label=tscode+'月收益率')
ax.plot(df.date,df['ind_log_ret']*100,'--md',lw=1.5,label='上证指数月收益率')
plt.legend()
ymajorFormatter=ticker.FormatStrFormatter('%.2f%%')#设置 y 轴标签文本的格式
ax.yaxis.set_major_formatter(ymajorFormatter)#y 轴数据以百分比格式
plt.xticks(rotation=60)
plt.ylabel('月收益率')
ax2=ax.twinx()
ax2.yaxis.set_major_formatter(ymajorFormatter)#显示百分比
ax2.plot(df.date,df['stk_log_ret_cum']*100,'-rp',lw=1.5,label=tscode+'累计收益率')
ax2.plot(df.date,df['ind_log_ret_cum']*100,'-gp',lw=1.5,label='上证指数累计收益率')
plt.legend()
plt.grid(True)
plt.ylabel('累计收益率')
```

（3）浦发银行股票 k 线图

```
import tushare as ts
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import ticker
from mpl_finance import candlestick_ochl
plt.rcParams['font.sans-serif']=['SimHei']#设置黑体
pro=ts.pro_api()
code='600000.SH'
df= pro.daily(ts_code=code,start_date='20200101')
df=df.sort_values(by='trade_date', ascending=True) #  原始数据按照日期降序排列
df['trade_date2']=df['trade_date'].copy()
df['dates']=np.arange(0,len(df))
def format_date(x, pos):
    if (x< 0) or (x > len(date_tickers)-1):
        return''
    return date_tickers[int(x)]
date_tickers= df['trade_date2'].values
df2=df.query('trade_date2 >="20171001"').reset_index()
df2['dates']=np.arange(0, len(df2))
date_tickers = df2['trade_date2'].values
fig,ax=plt.subplots(figsize=(12,9))
fig.subplots_adjust(bottom=0.2)
candlestick_ochl(ax,quotes=df2[['dates','open','close','high','low']].values,width=0.55,colorup='r',c
olordown='g',alpha=0.95)
ax.xaxis.set_major_formatter(ticker.FuncFormatter(format_date))
ax.set_ylabel('交易价格')
plt.title(code)
plt.grid(True)
plt.xticks(rotation=30)
plt.xlabel('交易日期')
plt.show()
```

（4）上市公司城市占比分析

```
df=pro.stock_company(exchange='SZSE',
fields='ts_code,chairman,reg_capital,setup_date,city,main_business,introduction,business_scope')
m=df.groupby('city').count()
a=m['chairman']
a.plot(kind='pie',title='上市公司占比城市分析',fontsize=10)
```