全过程的数据分析（股票数据）说明文档

姓名：吴义豪　　学号：41824334
北京科技大学计通学院 通信 1804 班

1. 作业过程

1.1 数据分析

作业提供数据为上证 A 股的数据，即上海机场的股票数据。数据为 CSV 文件，包含了 2003-4 至 2016-6 股票的基本信息，文件中数据存在空值，每一年的数据不一定覆盖全年 12 个月。

1.2 作业要求

（1）读取数据，在列上仅保留：代码、简称，日期，开盘价(元)，收盘价(元)，成交金额(元)。在行上，删除包含空值的行。

（2）对数据进行汇总，获得每个月（按自然月）的平均开盘价（元）和平均收盘价（元），总成交金额（元）。此时获得数据：代码、简称，月份，平均开盘价，平均收盘价，总成交金额。

（3）绘制图形，横坐标是月份，纵坐标是股价，绘制平均开盘价（元）、平均收盘价（元）随月份的变化（两条曲线）。

（4）取所有月份的总成交金额构成的一组数值，判定这组数值是否符合正态分布，并简述回顾正态分布检验的原理。

2. 代码说明

2.1　文件读取与数据处理

列上仅保留：代码、简称，日期，开盘价(元)，收盘价(元)，成交金额(元)，在行上，把包含空值的行删除，完成作业要求 1：

```
1.  data = pd.read_csv('./600004.SH.CSV',encoding='gbk')
2.  save_column = ['代码','简称','日期','开盘价(元)','收盘价(元)','成交金额(元)']
3.  save_data = data.loc[:,save_column]
4.  save_data.dropna(axis=0,how='any',inplace=True)   #删除含空值的行
5.  # save_data.to_csv('./Result1.csv',encoding='utf_8_sig')   #just for test
```

```
save_data.index = pd.to_datetime(save_data.日期)    #将日期格式转化为 datetime 类型并作为标签
```

2.2 数据汇总与保存

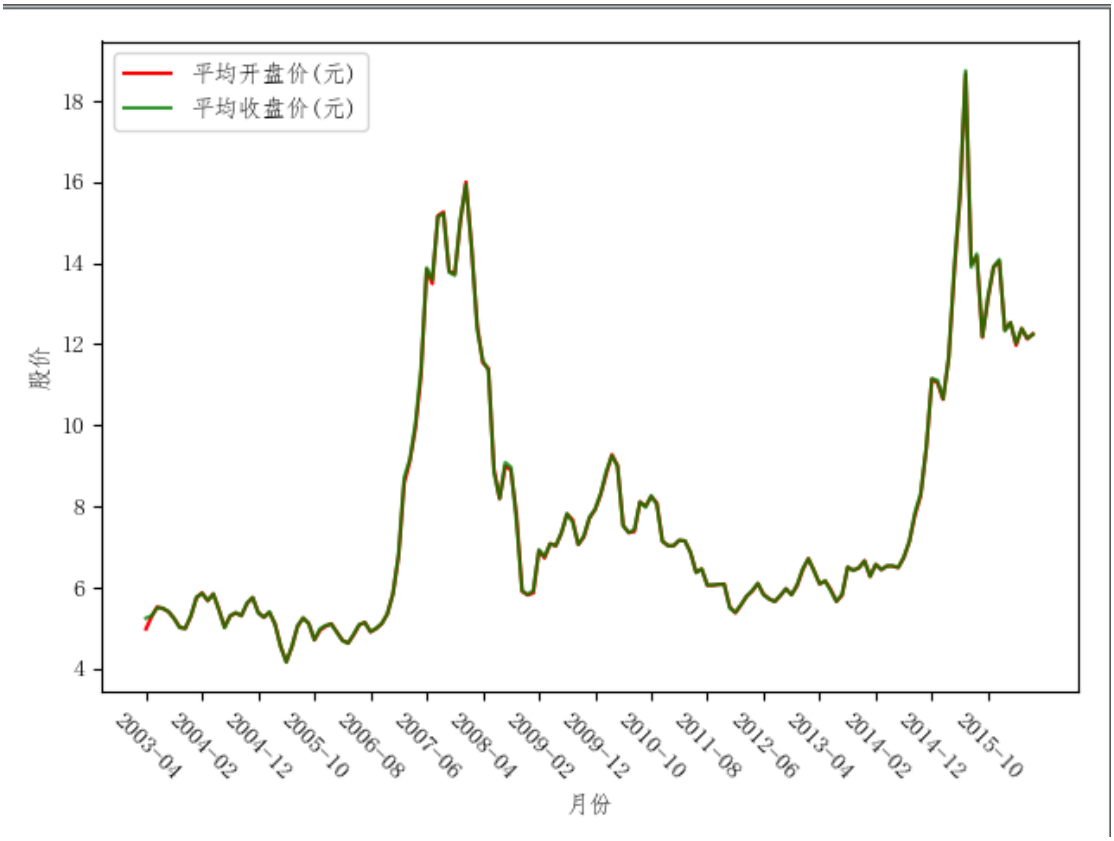
按代码、简称，月份，平均开盘价，平均收盘价，总成交金额存储数据并保存到 Result.csv 文件中：

```
new_stock = save_data.loc[:,['开盘价(元)','收盘价(元)']].resample('M').apply(np.mean)  #按月对开盘价和收盘价求平均值
new_stock['总成交金额(元)'] = save_data.loc[:,'成交金额(元)'].resample('M').apply(np.sum) #按月对成交金额求和
new_stock.rename(columns={'开盘价(元)':'平均开盘价(元)','收盘价(元)':'平均收盘价(元)'},inplace=True)
index=list(new_stock.index)
date=[str(i).split('-')[0]+'-'+str(i).split('-')[1] for i in index]  #只取日期的年和月
new_stock.insert(0,'代码','600004.SH')
new_stock.insert(1,'简称','白云机场')
new_stock.insert(2,'日期',date)
new_stock.index = range(1,len(new_stock) + 1)  #将 index 的 datetime 变为序号
new_stock.to_csv('./Result.csv',encoding='utf_8_sig')
```

2.3 图形绘制

绘制平均开盘价和平均收盘价随月份的变化：

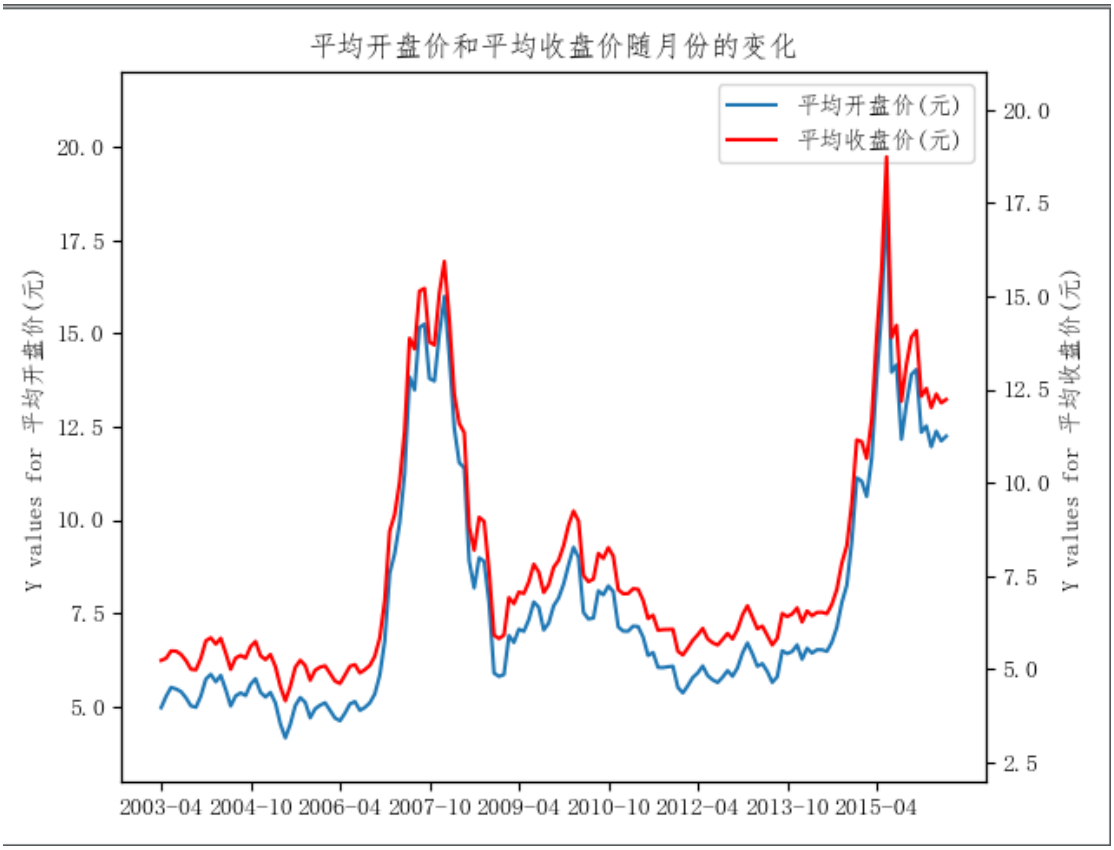
```
plt.rcParams['font.sans-serif']=['FangSong']
plt.plot(new_stock.index,new_stock['平均开盘价(元)'], color='red',label='平均开盘价(元)')
plt.plot(new_stock.index,new_stock['平均收盘价(元)'], color='green',label='平均收盘价(元)',alpha=0.8)
plt.xlabel('月份')
plt.ylabel('股价')
plt.xticks(new_stock.index[::10],new_stock['日期'][::10])
plt.xticks(rotation=-45)
plt.legend()
plt.show()
```



同一张图上绘制两条曲线

由于数值差距不大，发现两曲线已经近似重合，为了使效果更加明显，可以采用双 Y 轴绘制曲线图，将左 Y 轴设置为[3,22]，右 Y 轴设置为[2,21]：

```
1.  fig = plt.figure()
2.  ax1 = fig.add_subplot(111)
3.  lns1 = ax1.plot(new_stock.index,new_stock['平均开盘价(元)'])
4.  ax1.set_ylabel('Y values for 平均开盘价(元)')
5.  ax1.set_title("平均开盘价和平均收盘价随月份的变化")
6.  ax1.set_ylim([3,22])
7.  ax2 = ax1.twinx()  # this is the important function
8.  lns2 = ax2.plot(new_stock.index,new_stock['平均收盘价(元)'], 'r')
9.  ax2.set_ylim([2,21])
10. ax2.set_ylabel('Y values for 平均收盘价(元)')
11. ax2.set_xlabel('月份 ')
12. plt.xticks(new_stock.index[::18],new_stock['日期'][::18])
13. lns = lns1 +lns2
14. labs = [l.get_label() for l in lns]
15. ax1.legend(lns,labs,loc=0)
16. plt.xticks(rotation=-45)
17. plt.show()
```



双 Y 轴绘制曲线图

2.4 判定数值是否符合正态分布

2.4.1 简单回顾正态分布检验的原理

正态分布检验就是判断样本所代表的背景总体与理论正态分布是否没有显著差异的检验，python 中常用的检验正态分布的方法有：shapiro 方法，normaltest 方法，kstest 方法，anderson 方法。这里检验时采用 normaltest 方法，原理是基于数据的 skewness 和 kurtosis。利用 normaltest 函数得到该组数据的 pvalue 值，与显著性水平进行比较，对于是否否定原假设做出比较，然后得出结论。

2.4.2 代码实现

```
1.  [statics,pvalue]=stats.normaltest(new_stock['总成交金额(元)'])
2.  if pvalue>0.001:
3.    print( 'p 值为{}，高于显著性水平(0.001),总成交金额（元）服从正态分布'.format(pvalue))
4.  else:
5.    print('p 值为{}，低于显著性水平(0.001),总成交金额（元）不服从正态分布'.format(pvalue))
```

```
6.  new_stock['总成交金额(元)'].plot(kind="hist",bins=50)
7.  plt.savefig('假设检验.png', dpi=300)
8.  plt.show()
```

3. 结论与可视化结果

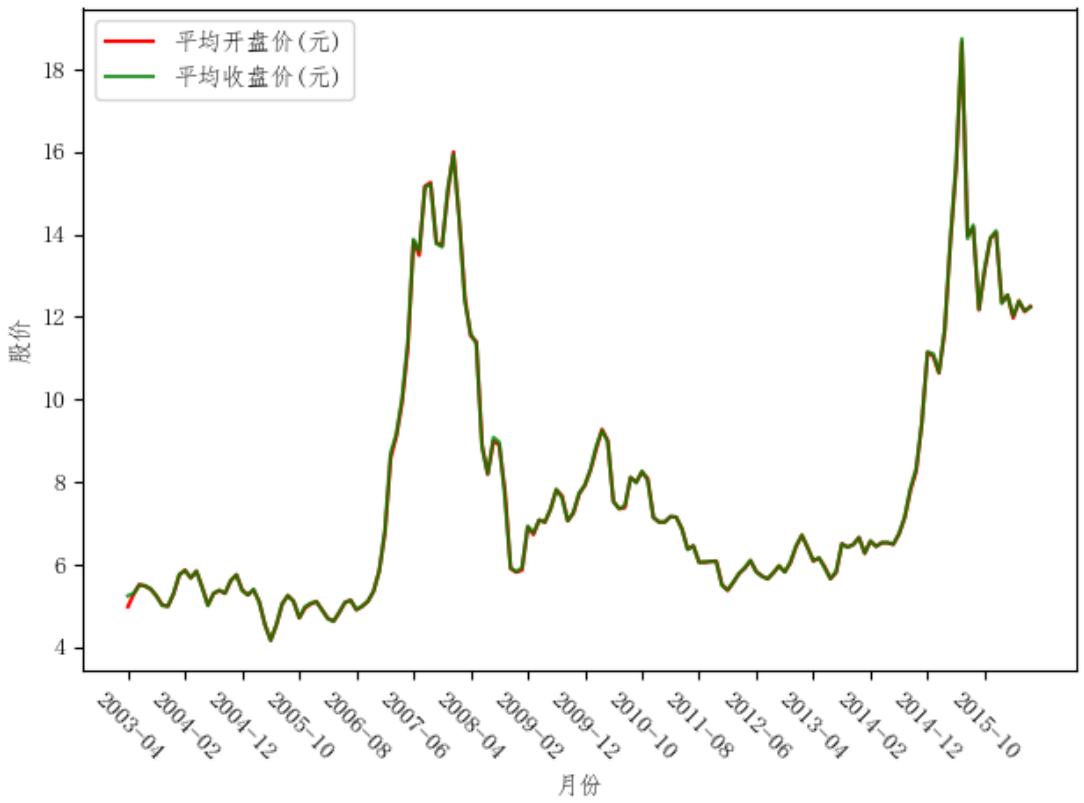
3.1 结论

通过用 normaltest 方法对所有月份的总成交金额构成的一组数值进行正态分布判断，发现总成交金额（元）不服从正态分布。

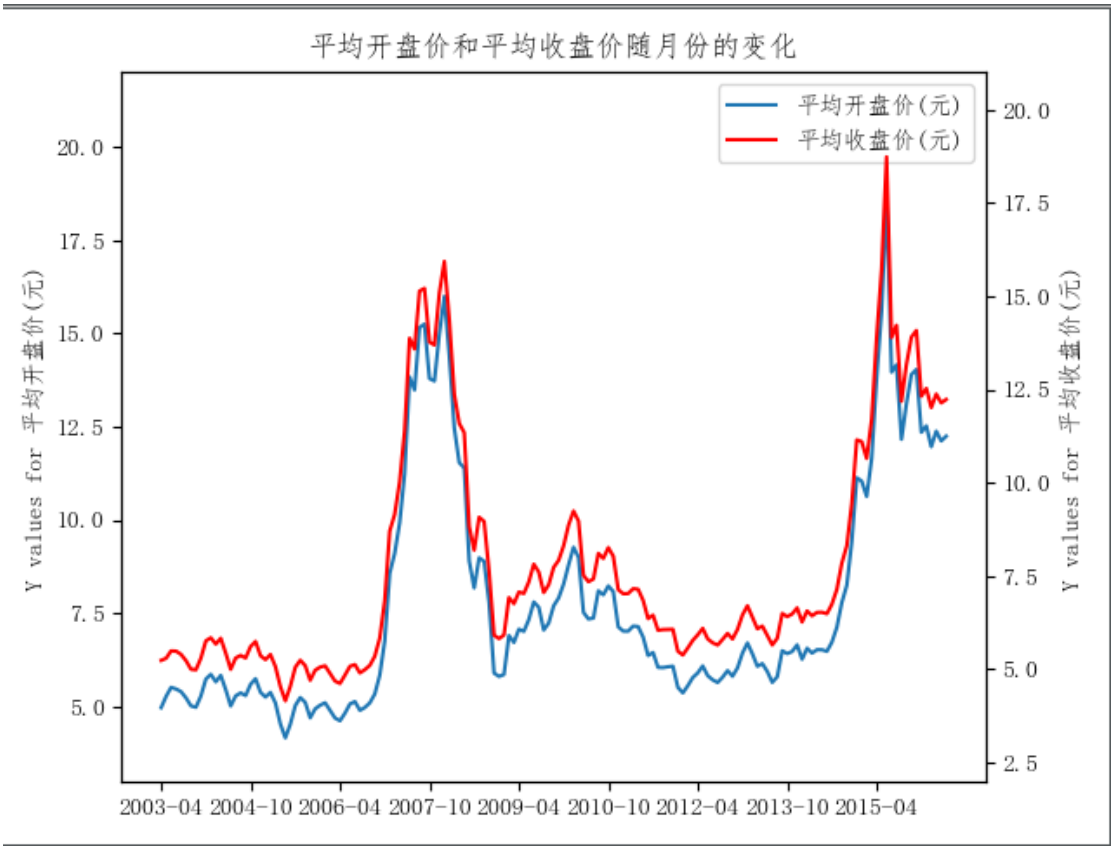
```
D:\anaconda3\python.exe D:/pycharmcode/data_sic/data_pro.py
p值为3.898889638238027e-21，低于显著性水平(0.001),总成交金额（元）不服从正态分布

Process finished with exit code 0
```

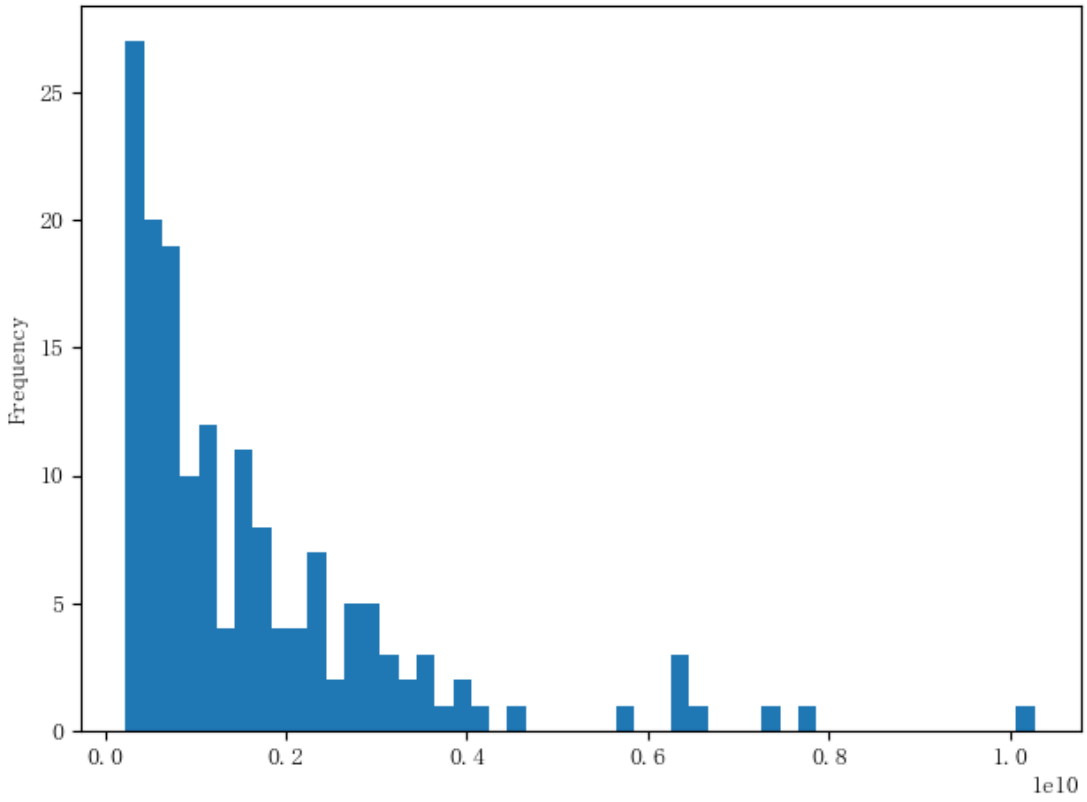
3.2 可视化结果



平均开盘价（元）、平均收盘价（元）随月份的变化



双 Y 轴绘制股份随月份的变化



假设检验图

4. 数据处理中的问题与解决

4.1 保存处理后的 csv 文件出现乱码

4.1.1 问题

希望随时保存中间处理得到的 csv 文件确保过程按预期进行，使用以下命令直接保存文件出现乱码，尝试加入 encoding="utf_8"后仍乱码：

```
1. new_stock.to_csv('./Resultluanma.csv')
```

	浠 g 燼	绠€绉?鏃ニ	骞冲潎寮€	骞冲潎鏀朵	鎬绘垚浜ら	噹棰?鍏?
1	600004.SH	鐗戒箱鏈哄	Apr-03	4.972067	5.239067	2.86E+09
2	600004.SH	鐗戒箱鏈哄	May-03	5.276007	5.296487	4.23E+09
3	600004.SH	鐗戒箱鏈哄	Jun-03	5.515475	5.493185	1.81E+09
4	600004.SH	鐗戒箱鏈哄	Jul-03	5.472352	5.487543	1.91E+09
5	600004.SH	鐗戒箱鏈哄	Aug-03	5.401648	5.4008	4.97E+08
6	600004.SH	鐗戒箱鏈哄	Sep-03	5.234691	5.232768	5.48E+08
7	600004.SH	鐗戒箱鏈哄	Oct-03	5.019939	5.002872	6.52E+08
8	600004.SH	鐗戒箱鏈哄	Nov-03	4.98224	4.98645	7.7E+08
9	600004.SH	鐗戒箱鏈哄	Dec-03	5.286461	5.299813	1.46E+09
10	600004.SH	鐗戒箱鏈哄	Jan-04	5.738885	5.762508	1.78E+09
11	600004.SH	鐗戒箱鏈哄	Feb-04	5.86475	5.846075	2.34E+09
12	600004.SH	鐗戒箱鏈哄	Mar-04	5.665835	5.675787	1.24E+09
13	600004.SH	鐗戒箱鏈哄	Apr-04	5.837436	5.833064	1.68E+09
14	600004.SH	鐗戒箱鏈哄	May-04	5.453212	5.434025	3.89E+08
15	600004.SH	鐗戒箱鏈哄	Jun-04	5.0174	5.001057	5.56E+08
16	600004.SH	鐗戒箱鏈哄	Jul-04	5.287536	5.304164	8.63E+08

Csv 文件出现乱码

4.1.2 解决方案

保存时编码方式变为 encoding="utf_8_sig"即可：

```
1. new_sock.to_csv(file_name3,encoding="utf_8_sig")
```

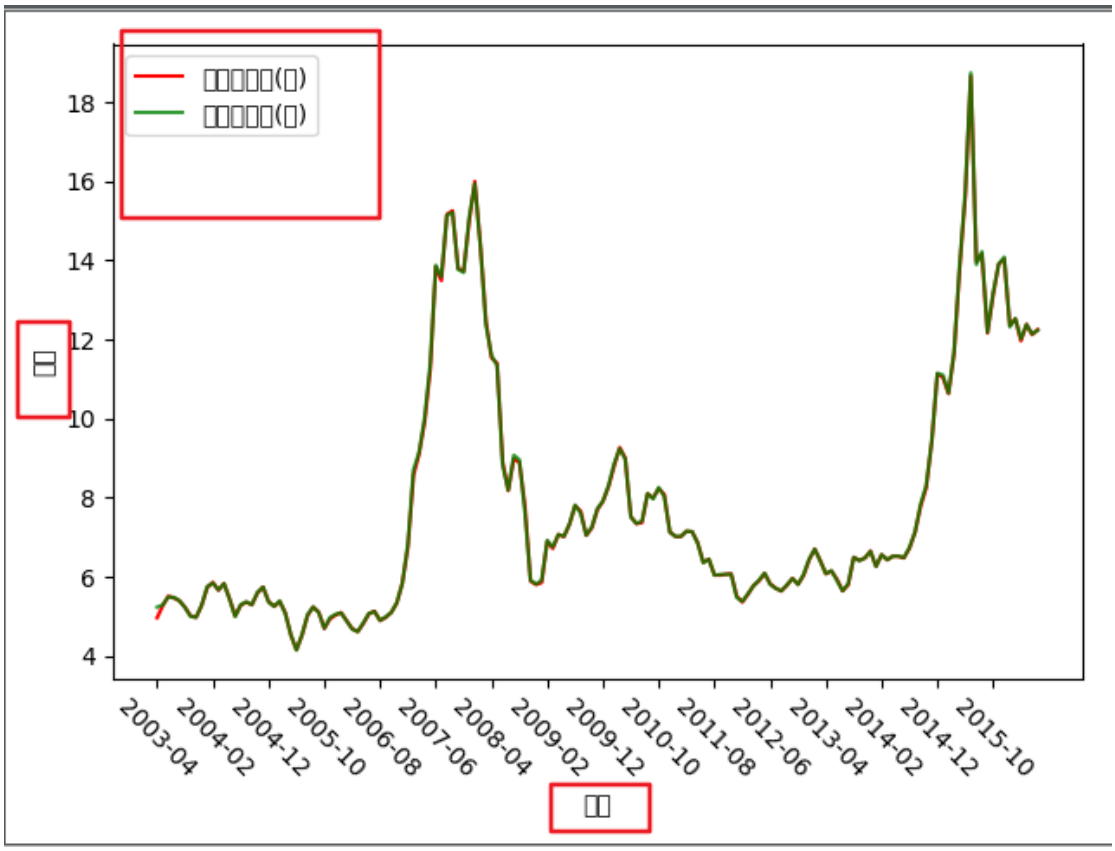
A	B	C	D	E	F	G	H
	代码	简称	日期	平均开盘价	平均收盘价	总成交金额(元)	
1	600004.SH	白云机场	Apr-03	4.972067	5.239067	2.86E+09	
2	600004.SH	白云机场	May-03	5.276007	5.296487	4.23E+09	
3	600004.SH	白云机场	Jun-03	5.515475	5.493185	1.81E+09	
4	600004.SH	白云机场	Jul-03	5.472352	5.487543	1.91E+09	
5	600004.SH	白云机场	Aug-03	5.401648	5.4008	4.97E+08	
6	600004.SH	白云机场	Sep-03	5.234691	5.232768	5.48E+08	
7	600004.SH	白云机场	Oct-03	5.019939	5.002872	6.52E+08	
8	600004.SH	白云机场	Nov-03	4.98224	4.98645	7.7E+08	
9	600004.SH	白云机场	Dec-03	5.286461	5.299813	1.46E+09	
10	600004.SH	白云机场	Jan-04	5.738885	5.762508	1.78E+09	
11	600004.SH	白云机场	Feb-04	5.86475	5.846075	2.34E+09	
12	600004.SH	白云机场	Mar-04	5.665835	5.675787	1.24E+09	

正确保存 csv 文件效果

4.2 Pandas matplotlib 画图无法显示中文字体的问题

4.2.1 问题

直接绘图时显示的图片，会显示中文为方块，查找解决方案时发现 Pandas 在绘图时,会显示中文为方块,主要原因有二：matplotlib 字体问题；seaborn 字体问题。



中文显示为方块例图

4.2.2 解决方案（一）

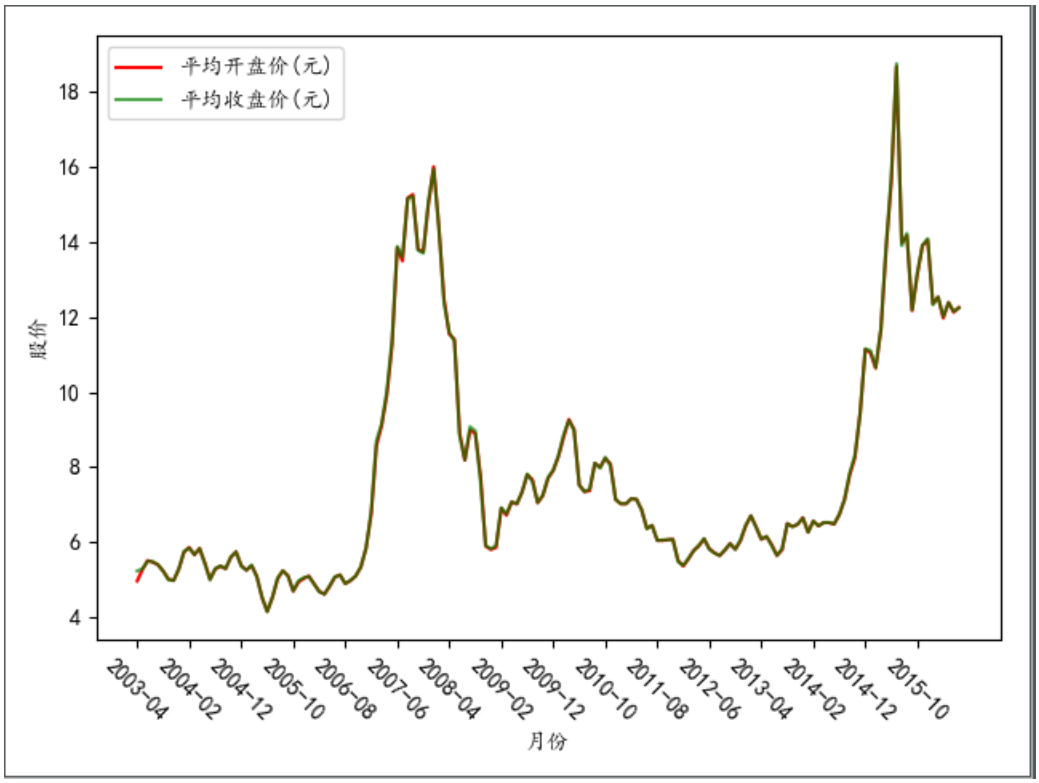
matplotlib 动态修改配置

```
1. import matplotlib as mpl
2. mpl.rcParams['font.sans-serif'] = ['KaiTi']
3. mpl.rcParams['font.serif'] = ['KaiTi']
4. # mpl.rcParams['axes.unicode_minus'] = False # 解决保存图像是负号'-'显示为方块的问题,或者转换负号为字符串
```

4.2.3 解决方案（二）

matplotlib 设置自定义字体

```
1. import numpy as np
2. import pylab as pl
3. import matplotlib.font_manager as fm
4.
5. myfont = fm.FontProperties(fname=r'D:\Fonts\simkai.ttf') # 设置字体
6. plt.xlabel('月份',fontproperties=myfont,fontsize=24)
7. plt.ylabel('股价',fontproperties=myfont,fontsize=24)
8. plt.show()
```



汉字正确显示的效果图

4.3 plt.savefig 保存图片时一片空白

4.3.1 问题

采用如下命令时保存图片，结果图片是一片空白：

```
plt.xticks(new_stock.index[::10],new_stock['日期'][::10])
plt.xticks(rotation=-45)
plt.legend()
plt.show()
plt.savefig('平均开盘价和平均收盘价随月份的变化.png', dpi=300)
```

原因：其实产生这个现象的原因很简单：在 plt.show() 后调用了 plt.savefig() ，在 plt.show() 后实际上已经创建了一个新的空白的图片（坐标轴），这时候你再 plt.savefig() 就会保存这个新生成的空白图片。

4.3.2 解决方案（一）

在 plt.show() 之前调用 plt.savefig()：

```
import matplotlib.pyplot as plt

""" 一些画图代码 """

plt.savefig("filename.png")
plt.show()
```

4.3.3 解决方案（二）

画图的时候获取当前图像（这一点非常类似于 Matlab 的句柄的概念）：

```
# gcf: Get Current Figure
fig = plt.gcf()
plt.show()
fig.savefig('tessstttyyy.png', dpi=100)
```

5. 参考与致谢

1. https://blog.csdn.net/lvshu_yuan/article/details/80413005
2. https://blog.csdn.net/zhuzuwei/article/details/80890007
3. https://www.jb51.net/article/154380.htm
4. https://www.cnblogs.com/shona/p/12364216.html
5. https://www.cnblogs.com/zgq25302111/p/11334044.html