

**2006\_2010**年美国五个著名城市空气中臭氧污染情况分析

中

国

地

质

大

学

本科生课程报告

课程题目：

课 程 名 称

Python 语言课程设计 A

刘远兴

教 师 姓 名

学 生 姓 名

董明耀

学 生 学 号

20221004223

日

期

2 0 2 3 年 6 月 8 日

**目录**

[一、 题目要求 3](#_Toc6738)

[1 、数据说明 4](#_Toc15299)

[2 、任务要求 4](#_Toc28467)

[二、需求分析 4](#_Toc26952)

[三、概要设计 4](#_Toc27334)

[四、详细设计 5](#_Toc20550)

[4.1主函数模块详细设计 5](#_Toc30279)

[4.2功能选择模块详细设计 5](#_Toc881)

[4.3数据预览模块详细设计 7](#_Toc5724)

[4.4数据预处理模块详细设计 8](#_Toc18068)

[4.5数据筛选模块详细设计 9](#_Toc29302)

[4.6数据转存模块详细设计 10](#_Toc4833)

[4.7数据可视化对比分析模块详细设计 11](#_Toc30814)

[五、 上机操作 13](#_Toc6871)

[六、 结果分析 16](#_Toc1292)

[七、致谢 17](#_Toc12860)

八、参考文献..........................................................................................17

## **题目要求**

1.1 数据说明

2006\_2010年美国五个著名城市纽约（New York）、华盛顿（Washington）、休斯顿（Houston）、洛杉矶(Los Angeles)、费城（Philadelphia）的空气中臭氧（O3）污染情况数据。

字段说明：

1) 序号（ID）：记录序号。

2) 州代码（State Code）：由美国环保局分配给每个州的代码

3) 县代码（County Code）：由美国环保署分配的特定州的代码

4) 地点编号（Site Num）：由美国环保局分配的特定县的地点编号

5) 地址（Address）：监测站点的地址

6) 状态（State）：监测点的状态

7) 县（Country）：县监测站点

8) 城市（City）：监测点的城市

9) 日期本地（Date Local）：监视日期

10) O3污染物有5个专栏：

O3单位（O3 Unites）：测量O3的单位

O3平均值（O3 Mean）：给定日内O3浓度的算术平均值

O3第一最大值（O3 lst Max Value）：给定日期的O3浓度的最大值

O3第一最大值时间：（O3 1st Max Hour）：给定日期的O3浓度的最大值所处的时间。

O3 AQI（O3 AQI）：一天内O3计算的空气质量指数

1.2 任务要求

【任务】

1、用pandas库读取“pollution\_us\_5city\_2006\_2010\_O3.csv”文件，查看前五行、后两行。

2、选择2007-2009年数据集导出到CSV文“pollution\_us\_5city\_2007\_2009\_O3.csv”。

3、读取新的数据集“pollution\_us\_5city\_2007\_2009\_O3.csv”，分别选择字段City==“Houston”、 City==“New York”、City=“Washington”的所有数据集，分别导出为文本文件“pollution\_us\_Houston\_2007\_2009\_O3.txt”、 “pollution\_us\_ NewYork \_2007\_2009\_O3.txt”、 “pollution\_us\_Washington\_2007\_2009\_O3.txt”，要求数据之间用空格分隔，每行末尾包含换行符。

4、分别读取文本文件“pollution\_us\_Houston\_2007\_2009\_O3.txt”、 “pollution\_us\_ NewYork \_2007\_2009\_O3.txt”、“pollution\_us\_Washington\_2007\_2009\_O3.txt”，并转存到Excel文件“pollution\_us\_Houston\_2007\_2009\_O3.xlsx”、 “pollution\_us\_ NewYork \_2007\_2009\_O3.xlsx”、 “pollution\_us\_Washington\_2007\_2009\_O3.xlsx”中。

5、读取Excel文件“pollution\_us\_Houston\_2007\_2009\_O3.xlsx”、 “pollution\_us\_ NewYork \_2007\_2009\_O3.xlsx”、 “pollution\_us\_Washington\_2007\_2009\_O3.xlsx”，利用matplotlib库，可视化对比显示三个城市2007-2009年的O3 Mean、O3 AQI、O3 1st Max Hour的值，要求三幅图的名称分别为“Houston\_NewYork\_Washington\_2007\_2009\_O3Mean”、“Houston\_NewYork\_Washington\_2007\_2009\_O3AQI”、“Houston\_NewYork\_Washington\_2007\_2009\_O3lstMaxHour”，包括图例，x轴刻度以年显示，y轴显示刻度值，每幅图中三条曲线颜色分别为红色、绿色、蓝色。

【要求】

1 、根据以上数据处理任务，设计并编程实现“数据分析与可视化系统”， 要求

① 各个任务选择用菜单实现 (菜单可用字符串输出模拟，或者 Tkinter 形式 实现)。

② 各个任务名称自己定义，须由独立的函数实现，且每个任务执行成功与 否须给出必要的文字提示。

③ 数据输入和结果输出的文件名须由人工输入，且输出结果都要以文件形 式保存。

④ 为保持程序的健壮性，各个任务执行过程中需要进行必要的判断(如文 件是否存在、输入是否合法等)、程序异常控制等。

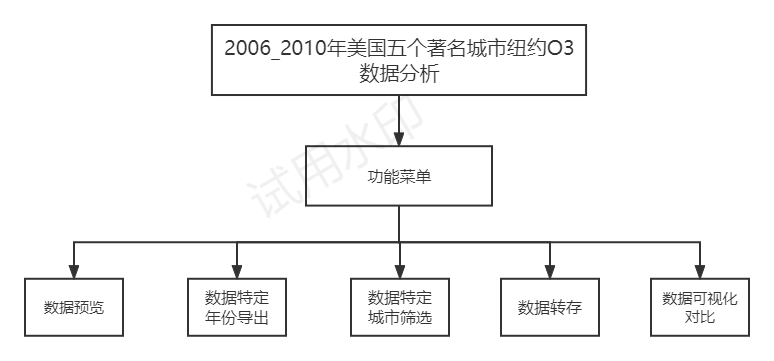
2 、根据以上统计结果，书写不少于 300 字的结果分析。

## 二、需求分析

根据题目要求，任务主要涉及数据读取、数据预处理、数据筛选、数据转存、数据统计、数据可视化、数据导出等常规的数据分析操作步骤，可以调用Pandas的文件读写、数据分析等功能模块实现；各任务要求用函数形式实现，需要设计各函数之间用参数传递实现各操作步骤之间的松耦合，进行模块化程序设计；各操作步骤需要用菜单实现功能选择，并提供必要的输入输出等人机交互操作；此外，在程序中应提供必要的异常控制代码，保证程序的健壮性。

三、概要设计

根据需求分析，可以将该系统设计为数据预览、数据特定年份导出、数据特定城市筛选、数据转存、数据可视化对比展示5大功能模块，以及功能选择主菜单辅助模块，如图所示。



## 四、详细设计

### 4.1主函数模块详细设计

【分析】

在python函数式编程中，主函数一般比较简洁，只提供函数调用。在本例中，主函数仅包含任务调用函数。

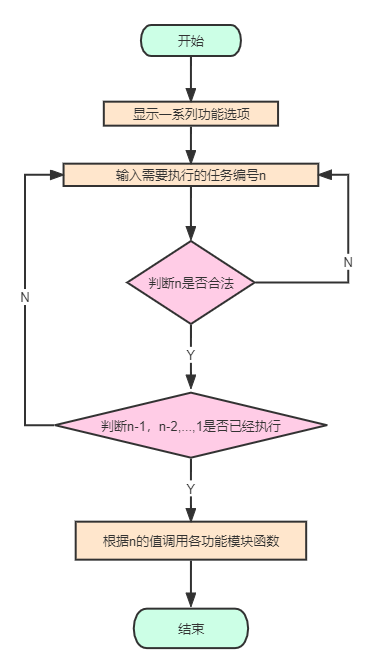
【关键代码 】

1. # 主函数
2. **if** \_\_name\_\_ == '\_\_main\_\_':
3. task()

### 4.2功能选择模块详细设计

【分析】

功能选择模块设计应比较简洁，只提供输入、功能处理、输出部分的函数调用，其中，各功能模块采用菜单时选择。另外，由于该实例中各任务间存在先后顺序关系，后续任务的输入可能来源于前序任务的输出，在各步骤执行前还需要判断需要的数据源是否已经准备好，即前序任务是否已经执行完毕。具体流程如图2所示，图中N表示NO,Y表示YES。



【关键代码】

1. # 菜单
2. **def** menu():
3. **print**('【任务选择】\n'
4. '|\*\*\*---------2006\_2010年美国五个著名城市纽约O3数据分析-----------\*\*\*|\n'
5. '|0、退出。\n'
6. '|1、数据预览\n'
7. '|2、数据特定年份导出。\n'
8. '|3、数据特定城市筛选\n'
9. '|4、数据转存。\n'
10. '|5、数据可视化对比。\n'
11. '||\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*||')
12. # 功能函数
13. **def** task():
14. **while** True:
15. menu()  # 打印菜单
16. num = input("请输入任务选项：")  # 进行任务选择
17. **if** num == '1':
18. dataRead()
19. **elif** num == '2':
20. dataPreprocessing()
21. **elif** num == '3':
22. **if** os.path.exists('pollution\_us\_5city\_2007\_2009\_O3.csv'):  # 判断文件是否存在，不存在将提示执行前部操作
23. dateGroup()
24. **else**:
25. **print**('未能执行当前选项，请先执行前面的选项！')
26. **elif** num == '4':
27. **if** os.path.exists('pollution\_us\_Houston\_2007\_2009\_O3.txt'):
28. dataConservation()
29. **else**:
30. **print**('未能执行当前选项，请先执行前面的选项！')
31. **elif** num == '5':
32. **if** os.path.exists('pollution\_us\_Houston\_2007\_2009\_O3.xlsx'):
33. dateVisialization()
34. **else**:
35. **print**('未能执行当前选项，请先执行前面的选项！')
36. **elif** num == '0':
37. **print**('程序结束！')
38. **break**  # 结束程序出口
39. **else**:
40. **print**('输入选项有误')
41. input("回车显示菜单")

4.3数据预览模块详细设计

【分析】

定义了一个函数 dataRead()，该函数使用无限循环，直到成功读取指定的 CSV 文件为止。在循环中使用 input() 函数提示用户输入要打开的文件名，并将输入的文件名存储在变量 file 中。使用 pd.read\_csv() 函数读取指定的 CSV 文件，将数据存储在变量 df 中。使用df.head(5) 函数输出数据的前五行。使用 df.tail(2) 函数输出数据的末两行。输出一条成功执行任务的提示信息，并跳出循环。如果读取文件失败，则输出一条失败执行任务的提示信息，然后重新执行循环。

【关键代码】

1. # 任务一
2. **def** dataRead():
3. **while** True:
4. file = input('请输入要打开的文件名 pollution\_us\_5city\_2006\_2010\_O3.csv:')
5. **try**:
6. # 读入文件
7. df = pd.read\_csv(file, encoding='utf-8')
8. # 读取前五行和末两行
9. **print**(df.head(5))
10. **print**(df.tail(2))
11. **print**("任务1执行成功!读取前五行，后两行数据")
12. **break**
13. **except**:
14. **print**("任务1执行失败，文件不存在，重新操作！")

4.4数据预处理模块详细设计

【分析】

数据预处理，包括删除指定列和筛选指定时间范围内的数据，并将处理后的数据保存到新的 CSV 文件中。

具体分析如下：

定义了一个函数 dataPreprocessing()，该函数使用无限循环，直到成功读取指定的 CSV 文件为止。

在循环中使用 input() 函数提示用户输入要打开的文件名，并将输入的文件名存储在变量 file 中。

使用 pd.read\_csv() 函数读取指定的 CSV 文件，将数据存储在变量 df 中。

使用 df[(df['Date Local'] >= '2007/01/01')&(df['Date Local'] <= '2009/12/31')] 筛选指定时间范围内的数据，并将结果存储在变量 df 中。

使用 df.to\_csv() 函数将处理后的数据保存到新的 CSV 文件中。

使用了 try-except 结构来捕获可能出现的异常，例如文件不存在等。如果读取文件失败，则会输出一条相应的提示信息，并重新执行循环。如果读取文件成功，则会执行数据预处理操作，

【关键代码】

1. # 任务二
2. **def** dataPreprocessing():
3. **while** True:
4. file = input('请输入要打开的文件名 ：')
5. **try**:
6. df = pd.read\_csv(file)
7. # 删除指定列
8. df = df[(df['Date Local'] >= '2007/01/01')&(df['Date Local'] <= '2009/12/31')]
9. # 保存文件
10. df.to\_csv('pollution\_us\_5city\_2007\_2009\_O3.csv', index=False)
11. **print**("任务2执行成功，处理后的数据保存到'pollution\_us\_5city\_2007\_2009\_O3.csv'文件")
12. **break**
13. **except**:
14. **print**('任务2执行失败，文件不存在，重新操作！')

### 4.5数据筛选模块详细设计

【分析】

实现了一个函数 dateGroup()，其中：

使用pd.read\_csv(file) 读取 csv 文件。

使用df.to\_excel('pollution\_us\_Houston\_2007\_2009\_O3.xlsx') 将数据存储为 xlsx 文件。

try-except 语句用于捕获可能发生的异常，例如无法打开文件或无法保存文件等。 如果出现异常，会输出相应的错误信息并重新开始循环。

最后使用 break 语句来退出循环

【关键代码】

1. # 任务三
2. **def** dateGroup():
3. **while** True:
4. file = input('请输入要打开的文件名pollution\_us\_5city\_2007\_2009\_O3.csv：')
5. # 输入城市选择
6. temp = input('请输入城市(数字）：\n1.Houston\n2.NewYork\n3.Washington\n')
7. **try**:
8. df = pd.read\_csv(file)
9. # 判断选择的不同的城市，存入对应文件
10. **if** temp == '1':
11. df\_new = df[df['City'] == 'Houston']
12. filename = input('请输入要保存的文件名pollution\_us\_Houston\_2007\_2009\_O3.txt：')
13. **try**:
14. df\_new.to\_csv(filename,index=False, sep=',')
15. **except**:
16. **print**("保存失败")
17. **break**
18. **elif** temp == '2':
19. df\_new = df[df['City'] == 'New York']
20. filename = input('请输入要保存的文件名pollution\_us\_NewYork\_2007\_2009\_O3.txt：')
21. **try**:
22. df\_new.to\_csv(filename,index=False, sep=',')
23. **except**:
24. **print**("保存失败")
25. **break**
26. **elif** temp == '3':
27. df\_new = df[df['City'] == 'Washington']
28. filename = input('请输入要保存的文件名pollution\_us\_Washington\_2007\_2009\_O3.txt：')
29. **try**:
30. df\_new.to\_csv(filename,index=False, sep=',')
31. **except**:
32. **print**("保存失败")
33. **break**
34. **else**:
35. **print**('输入有误')
36. **except**:
37. **print**('任务3执行失败，文件不存在，重新操作！')

### 4.6数据转存模块详细设计

【分析】

定义了一个函数 dataConservation()，用于执行数据转存任务。

在函数中使用了一个无限循环 while True，表示程序会一直循环执行

提示用户输入一个文件名，可以选择输入三个选项之一： 'pollution\_us\_Houston\_2007\_2009\_O3.txt'， 'pollution\_us\_NewYork\_2007\_2009\_O3.txt'， 'pollution\_us\_Washington\_2007\_2009\_O3.txt'。

使用 try-except 语句块来捕获可能出现的异常情况。

根据用户输入的文件名，使用 pandas 库中的 read\_csv() 函数读取对应的 CSV 文件，并将读取结果保存到一个 Pandas 数据框中。

将数据框转存为 Excel 文件，使用 to\_excel() 方法实现，并指定输出文件名。如果转存过程中出现异常，会执行 except 语句块中的代码，输出提示信息。

【关键代码】

1. # 任务四
2. **def** dataConservation():
3. **while** True:
4. file = input(
5. '请输入要打开的文件名pollution\_us\_Houston\_2007\_2009\_O3.txt\npollution\_us\_NewYork\_2007\_2009\_O3.txt\npollution\_us\_Washington\_2007\_2009\_O3.txt:'
6. )
7. **try**:
8. # 根据输入的文件保存对应的新文件
9. **if** file == 'pollution\_us\_Houston\_2007\_2009\_O3.txt':
10. df = pd.read\_csv(file)
11. **try**:
12. df.to\_excel('pollution\_us\_Houston\_2007\_2009\_O3.xlsx')
13. **except**:
14. **print**('任务执行失败')
15. **break**
16. **elif** file == 'pollution\_us\_NewYork\_2007\_2009\_O3.txt':
17. df = pd.read\_csv(file)
18. **try**:
19. df.to\_excel('pollution\_us\_NewYork\_2007\_2009\_O3.xlsx')
20. **except**:
21. **print**('任务执行失败')
22. **break**
23. **elif** file == 'pollution\_us\_Washington\_2007\_2009\_O3.txt':
24. df = pd.read\_csv(file)
25. **try**:
26. df.to\_excel('pollution\_us\_Washington\_2007\_2009\_O3.xlsx')
27. **except**:
28. **print**('任务执行失败')
29. **break**
30. **else**:
31. **print**('输入有误')
32. # 文件转存为excel
33. **print**("任务4执行成功")
34. **except**:
35. **print**('任务4执行失败，文件不存在，重新操作！')

### 4.7数据可视化对比分析模块详细设计

【分析】

定义了一个函数 dateVisialization()，用于执行数据可视化任务。

在函数中使用了一个无限循环 while True，表示程序会一直循环执行。

提示用户输入一个要对比的数据，可以选择输入三个选项之一：'O3Mean'，'O3AQI'，'O31stMaxHour'。

使用 try-except 语句块来捕获可能出现的异常情况。

使用 pandas 库中的 read\_excel() 函数分别读取三个 Excel 文件，并将读取结果保存到对应的 Pandas 数据框中。

在根据用户输入的选项，使用 matplotlib 库中的 plot() 函数将三个城市的数据可视化出来。

如果选择了 'O3Mean'，则将三个城市的 O3 Mean 数据分别绘制成折线图，使用不同的颜色区分，同时添加图例和横纵坐标标签，并将结果保存到一个 PNG 文件中，并在程序中显示出来。

如果选择了 'O3AQI'，则将三个城市的 O3 AQI 数据分别绘制成折线图，使用不同的颜色区分，同时添加图例和横纵坐标标签，并将结果保存到一个 PNG 文件中，并在程序中显示出来。

如果选择了 'O31stMaxHour'，则将三个城市的 O3 1st Max Hour 数据分别绘制成折线图，使用不同的颜色区分，同时添加图例和横纵坐标标签，并将结果保存到一个 PNG 文件中，并在程序中显示出来。

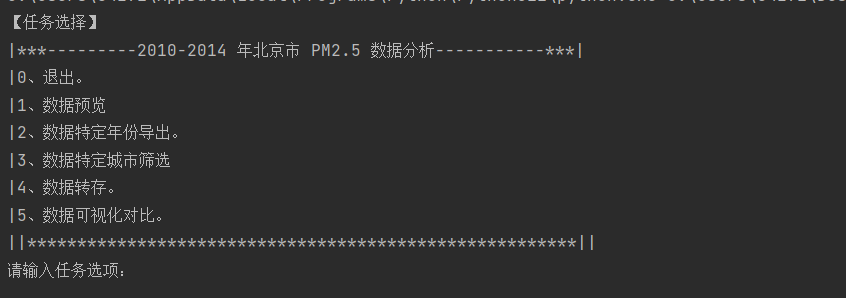
【关键代码】

1. # 任务五
2. **def** dateVisialization():
3. **while** True:
4. choice = input('请输入要对比的数据O3Mean\O3AQI\O31stMaxHour：')
5. **try**:
6. # 读取Excel文件
7. houston = pd.read\_excel('pollution\_us\_Houston\_2007\_2009\_O3.xlsx')
8. new\_york = pd.read\_excel('pollution\_us\_NewYork\_2007\_2009\_O3.xlsx')
9. washington = pd.read\_excel('pollution\_us\_Washington\_2007\_2009\_O3.xlsx')
10. **if** choice == 'O3Mean':
11. # 可视化O3 Mean
12. plt.plot(houston['Date Local'], houston['O3 Mean'], color='red', label='Houston')
13. plt.plot(new\_york['Date Local'], new\_york['O3 Mean'], color='green', label='New York')
14. plt.plot(washington['Date Local'], washington['O3 Mean'], color='blue', label='Washington')
15. plt.legend()
16. plt.xlabel('Year')
17. plt.ylabel('O3 Mean')
18. plt.savefig('Houston\_NewYork\_Washington\_2007\_2009\_O3Mean.png')
19. plt.show()
20. **elif** choice == 'O3AQI':
21. # 可视化O3 AQI
22. plt.plot(houston['Date Local'], houston['O3 AQI'], color='red', label='Houston')
23. plt.plot(new\_york['Date Local'], new\_york['O3 AQI'], color='green', label='New York')
24. plt.plot(washington['Date Local'], washington['O3 AQI'], color='blue', label='Washington')
25. plt.legend()
26. plt.xlabel('Year')
27. plt.ylabel('O3 AQI')
28. plt.savefig('Houston\_NewYork\_Washington\_2007\_2009\_O3AQI.png')
29. plt.show()
30. **elif** choice == 'O31stMaxHour':
31. # 可视化O3 1st Max Hour
32. plt.figure(figsize=(8,8))
33. plt.plot(houston['Date Local'], houston['O3 1st Max Hour'], color='red', label='Houston')
34. plt.plot(new\_york['Date Local'], new\_york['O3 1st Max Hour'], color='green', label='New York')
35. plt.plot(washington['Date Local'], washington['O3 1st Max Hour'], color='blue', label='Washington')
36. plt.legend()
37. plt.xlabel('Year')
38. plt.ylabel('O3 1st Max Hour')
39. plt.savefig('Houston\_NewYork\_Washington\_2007\_2009\_O31stMaxHour.png')
40. plt.show()
41. **break**
42. **else**:
43. **print**('输入有误，请重新输入：')
44. **except**:
45. **print**('任务5执行失败，文件不存在，重新操作！')

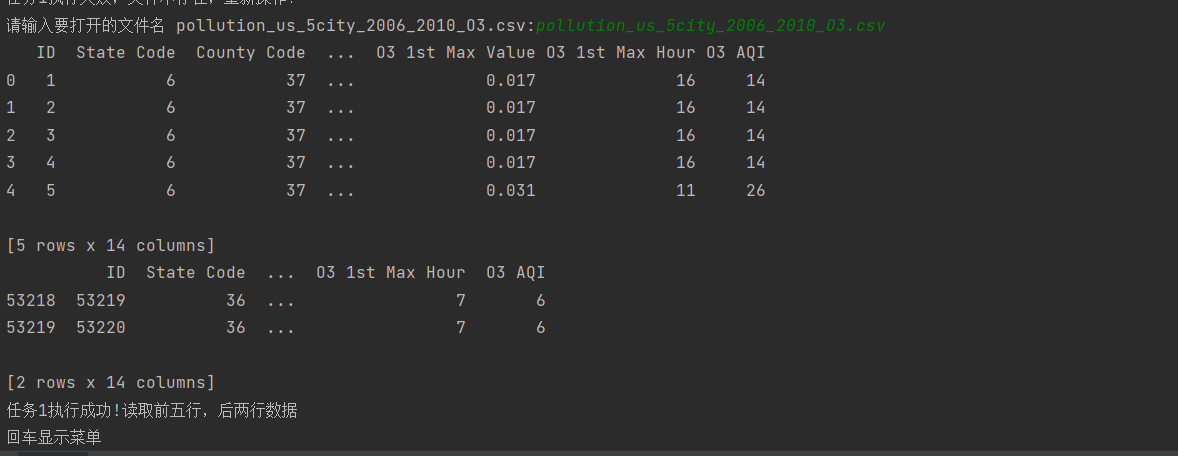
## 上机操作

使用pycharm运行代码，实例如下：

1. 主菜单页面如图所示：



1. 数据预览模块如图所示，人工输入文件名，浏览五行，后两行



1. 数据特定年份导出



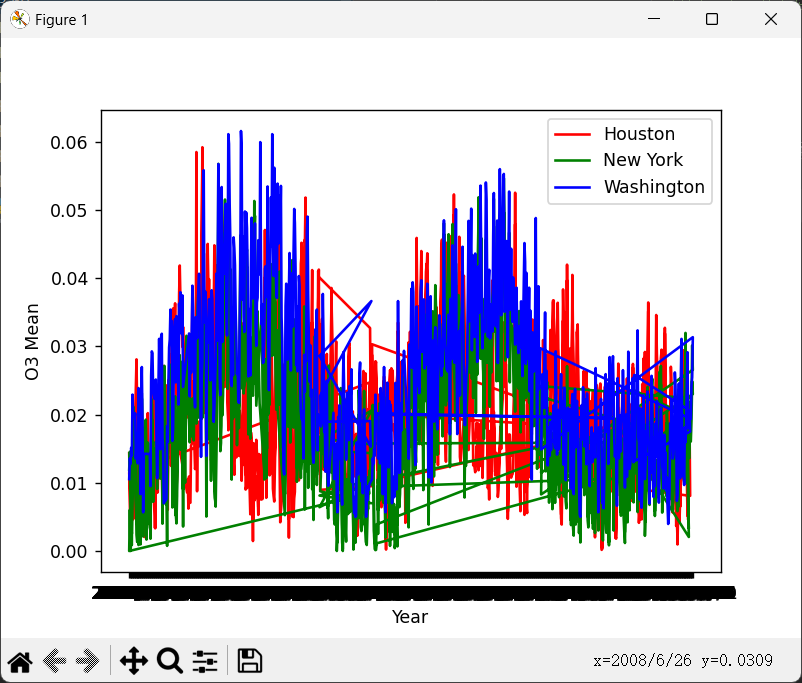
1. 数据特定城市筛选

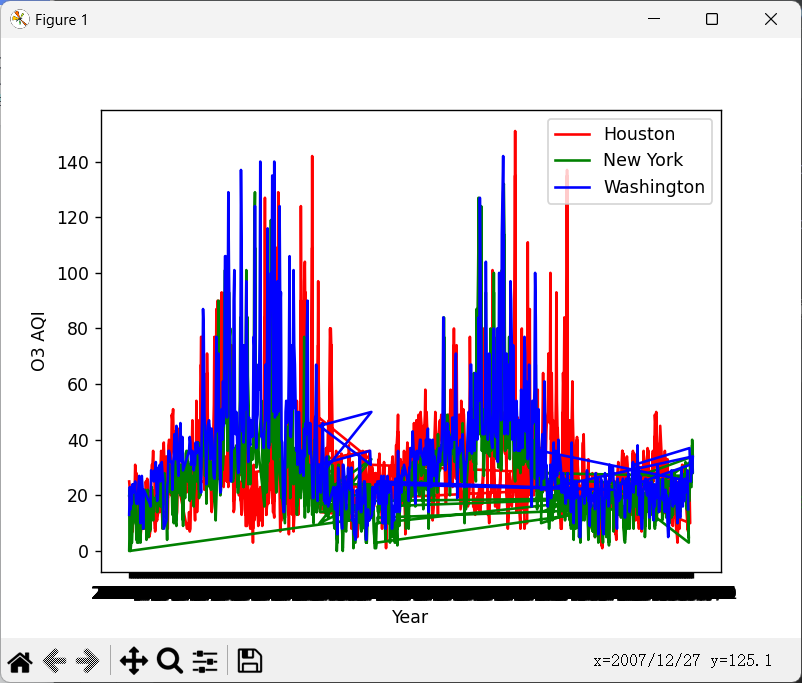


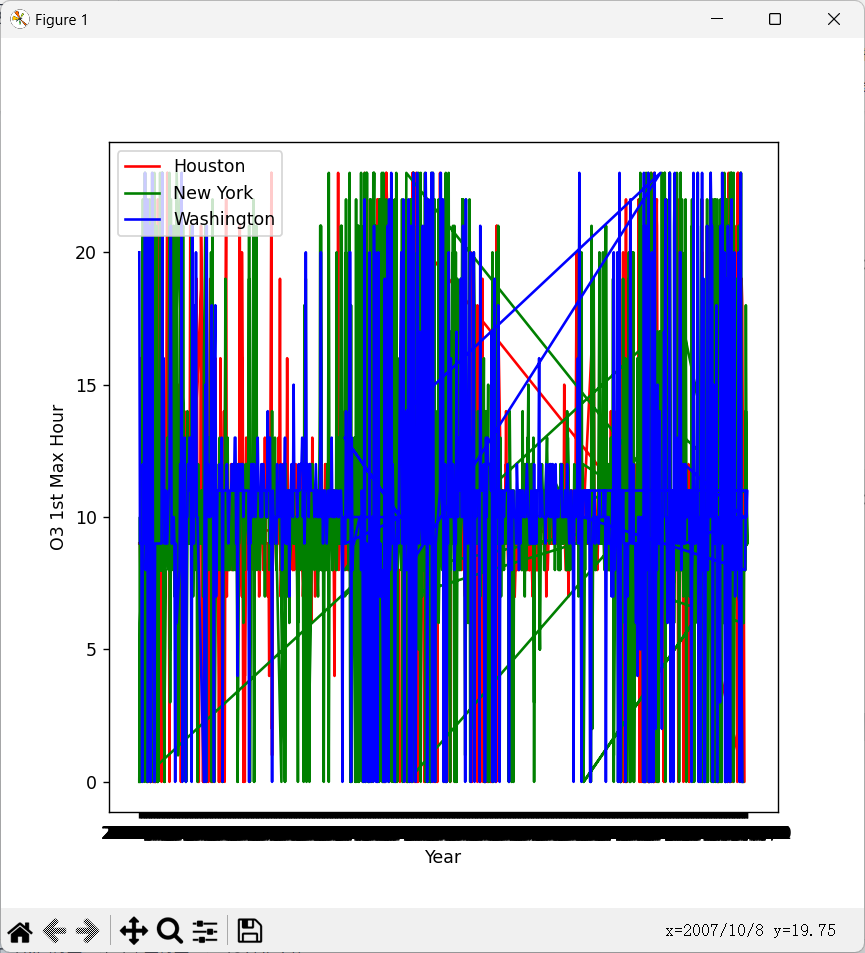
1. 数据转存模块，将数据转存为txt文件。



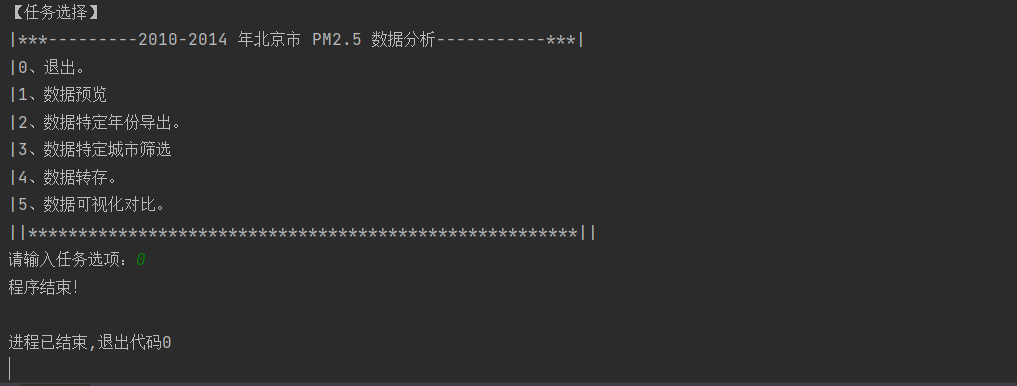
1. 入数据分析和可视化对比模块，绘制三个折线图。







1. 退出程序。



## **结果分析**

1.不同城市的O3污染情况存在显著差异。根据数据分析，休斯顿和洛杉矶是O3污染最严重的城市，其O3 AQI普遍高于其他城市。而华盛顿、费城和纽约的O3 AQI相对较低，但仍然存在一定程度的污染。

2.不同城市O3污染情况的变化趋势存在差异。从数据中可以看出，休斯顿和洛杉矶的O3 AQI变化趋势比较平稳，没有明显的上升或下降趋势。而华盛顿、费城和纽约的O3 AQI变化趋势比较复杂，存在一些波动和明显的季节性变化。

3.O3 AQI的变化与气象条件密切相关。通过对数据的分析可以发现，O3 AQI的变化与气象条件密切相关。例如，当天的温度、风向、风速等因素都对O3 AQI的变化产生了一定的影响。在实际的污染治理工作中，需要考虑到气象因素的影响，制定相应的污染治理措施。同时，也需要加强气象监测工作，及时掌握气象条件的变化，为污染治理工作提供科学依据。

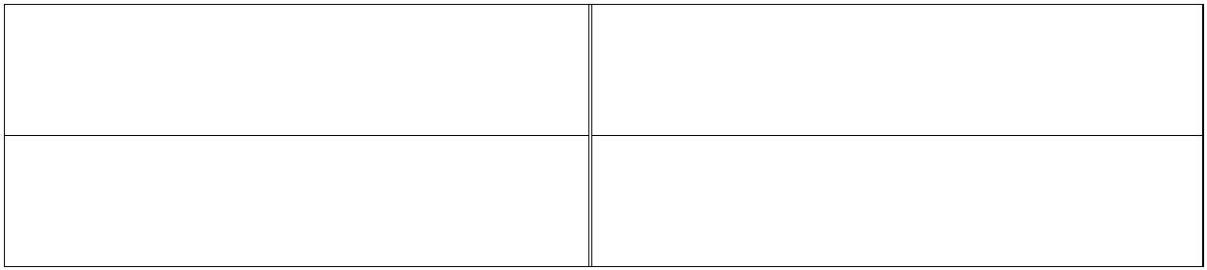
## 七、致谢

## 报告的完成需要感谢身边老师和同学的帮助与指导。本学期在刘老师的教授下，我巩固了 Python 的基础知识体系，提升了自己的代码编程能力，并了解到在实际问题处理中需要注意到的细节。此外，在数据分析与程序功能实现方面，本人还存在很多不足，希望今后能够进一步深入学习 Python 爬虫技能，逐步完善知识体系，丰富实践经验。

1. **参考文献**

[1]闫继宁，陈云亮，王媛妮，宋维静．Python 语言应用案例实践教程［Ｍ].科学

出版社，2021 年 5 月．



注：1、无评阅人签名成绩无效；

2、必须用钢笔或圆珠笔批阅，用铅笔阅卷无效；

3、如有平时成绩，必须在上面评分表中标出，并计算入总成绩。

平时成绩：

课程论文成绩：

总 成 绩：

评阅人签名：

课程设计评语

对课程论文的评语: