

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

\*\*\*

学 生 学 号

\*\*\*

日

期

2 0 \*\* 年 1 2 月 3 0 日

**目录**

[一、 题目要求 3](#_Toc27)

[1 、数据说明 4](#_Toc14174)

[2 、任务要求 4](#_Toc29215)

[二、需求分析 4](#_Toc2419)

[三、概要设计 4](#_Toc28964)

[四、详细设计 5](#_Toc25529)

[4.1主函数模块详细设计 5](#_Toc14063)

[4.2功能选择模块详细设计 5](#_Toc1184)

[4.3数据预览模块详细设计 7](#_Toc11524)

[4.4数据预处理模块详细设计 8](#_Toc2338)

[4.5数据筛选模块详细设计 9](#_Toc6099)

[4.6数据转存模块详细设计 10](#_Toc22988)

[4.7数据统计及可视化模块详细设计 10](#_Toc1070)

[五、 上机操作 12](#_Toc17620)

[六、结果分析 15](#_Toc20850)

[七、致谢](#_Toc22040) [15](#_Toc22040)

八、参考文献................................................................15

## **题目要求**

1.1 数据说明

2010-2014年北京市每小时天气预报数据。

字段说明：

1) No: 行序号

2) year:行数据记录年份

3) month: 行数据记录月份

4) day: 行数据记录日期

5) hour: 行数据记录时间

6) pm2.5: PM2.5浓度 (ug/m^3)

7) DEWP: Dew Point (单位：â.ƒ

8) TEMP: 温度(单位：â.ƒ

9) PRES: 压力(单位：hPa)

10) cbwd: 风向

11) Iws: 风力(单位：m/s)

12) Is: 累计下雪时长

13) Ir: 累计下雨时长

1.2 任务要求

【任务】

1、用pandas库读取“PRSA\_data\_2010.1.1-2014.12.31.csv”文件，查看前三行、后两行。

2、用pandas数据预处理模块将缺失值丢弃处理，删除列DEWP、TEMP、PRES、cbwd、Iws、Is、Ir，并将剩余列导出到新的csv文件“pm25\_data\_2010.1.1-2014.12.31.csv”。

3、利用pandas库重新读取新的数据集“pm25\_data\_2010.1.1-2014.12.31.csv”，并选择字段pm2.5 大于300的所有数据集，导出为文本文件“pm25\_hazardous\_data\_2010.1.1-2014.12.31.txt”，要求数据之间用逗号分隔，每行末尾包含换行符。

4、读取文本文件“pm25\_hazardous\_data\_2010.1.1-2014.12.31.txt”并转存到Excel文件pm25\_hazardous\_data\_2010.1.1-2014.12.31.xlsx中。

5、重新读取文本文件读取文本文件“pm25\_hazardous\_data\_2010.1.1-2014.12.31.txt”，分别统计出现最多的month、day、hour，并将month、day、hour 的出现频次用直方图显示。要求包括图例、图标题，x y轴均显示刻度值，直方图填充颜色分别为红色、绿色、蓝色；并将结果保存为png图片保存，分辨率为400dpi，png图片命名分别为“pm25\_hazardous\_month\_day\_hour.png”。

【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。

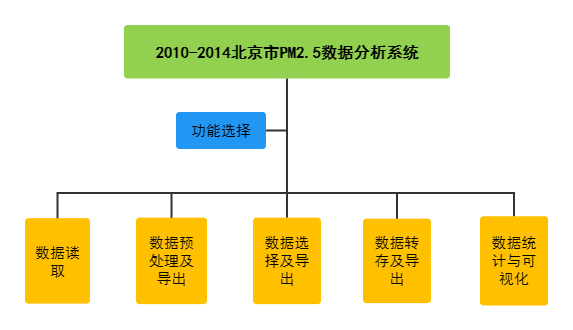
2 、根据以上统计结果，书写不少于 300 字的结果分析。

## 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读取、数据预处理及导出、数据选择及导出、数据转存及导出、数据统计与可视化展示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。

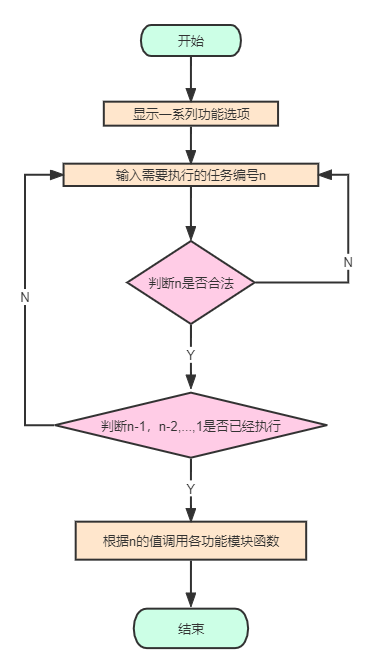
【关键代码 】

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. **def** menu():
2. **print**('【任务选择】\n'
3. '|\*\*\*---------2010-2014 年北京市 PM2.5 数据分析-----------\*\*\*|\n'
4. '|0、退出。\n'
5. '|1、数据前三行后两行读取\n'
6. '|2、数据缺失值丢弃，删除特定列处理。\n'
7. '|3、数据按pm2.5进行筛选\n'
8. '|4、数据转存。\n'
9. '|5、数据汇总处理与可视化。\n'
10. '||\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*||')

13. # 打印菜单
14. **def** task():
15. **while** True:
16. menu()  # 打印菜单
17. num = input("请输入任务选项：")  # 进行任务选择
18. **if** num == '1':
19. dataRead()
20. **elif** num == '2':
21. dataPreprocessing()
22. **elif** num == '3':
23. **if** os.path.exists('pm25\_data\_2010.1.1-2014.12.31.csv'):  # 判断文件是否存在，不存在将提示执行前部操作
24. dateScreening()
25. **else**:
26. **print**('未能执行当前选项，请先执行前面的选项！')
27. **elif** num == '4':
28. **if** os.path.exists('pm25\_hazardous\_data\_2010.1.1-2014.12.31.txt'):
29. dataConservation()
30. **else**:
31. **print**('未能执行当前选项，请先执行前面的选项！')
32. **elif** num == '5':
33. **if** os.path.exists('pm25\_hazardous\_data\_2010.1.1-2014.12.31.xlsx'):
34. dateDrawing()
35. **else**:
36. **print**('未能执行当前选项，请先执行前面的选项！')
37. **elif** num == '0':
38. **print**('程序结束！')
39. **break**  # 结束程序出口
40. **else**:
41. **print**('输入选项有误')
42. input("回车显示菜单")

4.3数据预览模块详细设计

【分析】

1. 通过 input() 函数获取用户输入的文件名。
2. 使用 pandas 库中的 read\_csv() 函数读取该文件，并指定编码为 utf-8。
3. 如果读取成功，则使用 head() 和 tail() 函数分别读取该文件的前三行和后两行数据，并打印输出。
4. 最后输出一条提示信息，表示任务1执行成功。

【关键代码】

1. # 数据预览
2. **def** dataRead():
3. **while** True:
4. fileName = input('请输入要打开的文件名 PRSA\_data\_2010.1.1-2014.12.31.csv:')
5. **try**:
6. # 读入文件
7. df = pd.read\_csv(fileName, encoding='utf-8')
8. # 读取前三行和末两行
9. **print**(df.head(3))
10. **print**(df.tail(2))
11. **print**("任务1执行成功!读取前三行，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】

1. 通过 input() 函数获取用户输入的文件名。
2. 使用 pandas 库中的 read\_csv() 函数读取该文件。
3. 使用 dropna() 函数删除该文件中的缺失值（NaN）。
4. 使用 drop() 函数删除该文件中的指定列（DEWP、TEMP、PRES、cbwd、Iws、Is、Ir）。
5. 使用 to\_csv() 函数将处理后的数据保存为新的 CSV 文件（文件名为 pm25\_data\_2010.1.1-2014.12.31.csv），并指定 index=False，表示不保存索引列。
6. 最后输出一条提示信息，表示任务2执行成功，并提示保存的文件名。

【关键代码】

1. # 数据预处理
2. **def** dataPreprocessing():
3. **while** True:
4. fileName = input('请输入要打开的文件名 PRSA\_data\_2010.1.1-2014.12.31.csv：')
5. **try**:
6. df = pd.read\_csv(fileName)
7. # 丢弃废弃值
8. df = df.dropna()
9. # 删除指定列
10. df = df.drop(['DEWP', 'TEMP', 'PRES', 'cbwd',
11. 'Iws', 'Is', 'Ir'], axis=1)
12. # 保存文件
13. df.to\_csv('pm25\_data\_2010.1.1-2014.12.31.csv', index=False)
14. **print**("任务2执行成功，处理后的数据保存到'pm25\_data\_2010.1.1-2014.12.31.csv'文件")
15. **break**
16. **except**:
17. **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】

1. 使用 while True 循环等待用户输入要打开的文件名。
2. 使用 pandas 库中的 read\_csv() 函数读取该文件。
3. 使用查询条件 df['pm2.5'] > 300，筛选出 pm2.5 值大于 300 的行数据，即危险的空气质量数据。
4. 使用 to\_csv() 函数将筛选出的数据保存为新的文本文件（文件名为 pm25\_hazardous\_data\_2010.1.1-2014.12.31.txt），并指定 index=False 和 sep=','，表示不保存索引列，以逗号作为分隔符。
5. 最后输出一条提示信息，表示任务3执行成功，并提示保存的文件名。
6. 如果读取文件失败，则输出一条错误信息提示用户重新输入文件名。

【关键代码】

1. # 数据筛选

4. **def** dateScreening():
5. **while** True:
6. fileName = input('请输入要打开的文件名 pm25\_data\_2010.1.1-2014.12.31.csv：')
7. **try**:
8. df = pd.read\_csv(fileName)
9. # 选择PM2.5>300的列
10. hazardous\_date = df[df['pm2.5'] > 300]
11. # 存至新文件
12. hazardous\_date.to\_csv('pm25\_hazardous\_data\_2010.1.1-2014.12.31.txt',
13. index=False, sep=',')
14. **print**("任务3执行成功，处理后数据保存在'pm25\_hazardous\_data\_2010.1.1-2014.12.31.txt'")
15. **break**
16. **except**:
17. **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】

1. 使用 while True 循环等待用户输入要打开的文件名。
2. 尝试使用 pandas 库中的 read\_csv() 函数读取该文件。
3. 使用 to\_excel() 函数将读取到的数据保存为新的 Excel 文件（文件名为 pm25\_hazardous\_data\_2010.1.1-2014.12.31.xlsx），并指定 index=False，表示不保存索引列。
4. 最后输出一条提示信息，表示任务4执行成功，并提示保存的文件名。
5. 如果读取文件失败，则会提示用户重新输入文件名，直到读取成功为止。

【关键代码】

1. # 数据转存
2. **def** dataConservation():
3. **while** True:
4. fileName = input(
5. '请输入要打开的文件名 pm25\_hazardous\_data\_2010.1.1-2014.12.31.txt ：')
6. **try**:
7. # 文件转存为excel
8. hazardous\_date = pd.read\_csv(fileName)
9. hazardous\_date.to\_excel(
10. 'pm25\_hazardous\_data\_2010.1.1-2014.12.31.xlsx', index=False)
11. **print**("任务4执行成功，转存为xlsx文件到'pm25\_hazardous\_data\_2010.1.1-2014.12.31.xlsx'")
12. **break**
13. **except**:
14. **print**('任务4执行失败，文件不存在，重新操作！')

### 4.7数据统计及可视化模块详细设计

【分析】

1. while True 循环等待用户输入要打开的 Excel 文件名。
2. 尝试使用 pandas 库中的 read\_excel() 函数读取该文件。
3. 使用 value\_counts() 函数对该文件中的 month、day、hour 列进行计数，分别得到每个月、每天和每小时出现次数的统计结果。
4. 分别输出最多出现的月份、日期和小时，并打印输出统计结果。
5. 使用 matplotlib 库绘制三个柱状图，分别表示危险空气质量数据中每个月、每天和每小时出现次数的分布情况，保存为 PNG 格式的图片文件（文件名分别为 pm25\_hazardous\_month.png、pm25\_hazardous\_day.png、pm25\_hazardous\_hour.png）。

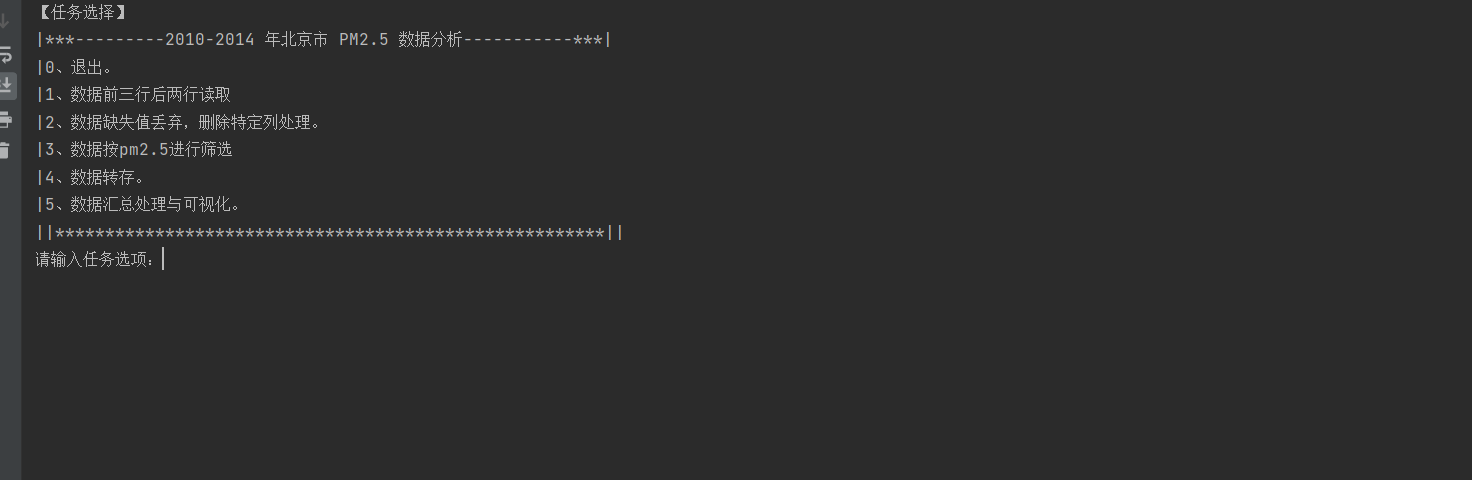
【关键代码】

1. # 数据汇总和可视化
2. **def** dateDrawing():
3. **while** True:
4. fileName = input(
5. '请输入要打开的文件名 pm25\_hazardous\_data\_2010.1.1-2014.12.31.xlsx:')
6. **try**:
7. hazardous\_date = pd.read\_excel(fileName)
8. # 选择month列计算出现的次数
9. month\_count = hazardous\_date['month'].value\_counts()
10. **print**('出现最多的月为{}月，共出现{}次'.format(
11. month\_count.index[0], month\_count.values[0]))
12. # 选择day列计算出现的次数
13. day\_count = hazardous\_date['day'].value\_counts()
14. **print**('出现最多的天为{}号，共出现{}次'.format(
15. day\_count.index[0], day\_count.values[0]))
16. # 选择hour列计算出现的次数
17. hour\_count = hazardous\_date['hour'].value\_counts()
18. **print**('出现最多的小时为{}点，共出现{}次'.format(
19. hour\_count.index[0], hour\_count.values[0]))
20. # 月份柱状图
21. plt.bar(month\_count.index, month\_count.values,
22. color=['red', 'green', 'blue'])
23. plt.title('PM2.5 Hazardous Month Count')
24. plt.xlabel('Month')
25. plt.ylabel('count')
26. plt.legend('n')
27. plt.savefig('pm25\_hazardous\_month.png', dpi=400)
28. plt.show()
29. # 日期柱状图
30. plt.bar(day\_count.index, day\_count.values,
31. color=['red', 'green', 'blue'])
32. plt.title('PM2.5 Hazardous Day Count')
33. plt.xlabel('Day')
34. plt.ylabel('count')
35. plt.legend('n')
36. plt.savefig('pm25\_hazardous\_day.png', dpi=400)
37. plt.show()
38. # 小时柱状图
39. plt.bar(hour\_count.index, hour\_count.values,
40. color=['red', 'green', 'blue'])
41. plt.title('PM2.5 Hazardous Hour Count')
42. plt.xlabel('Hour')
43. plt.ylabel('count')
44. plt.legend('n')
45. plt.savefig('pm25\_hazardous\_hour.png', dpi=400)
46. plt.show()
47. **break**
48. **except**:
49. **print**('任务5执行失败，文件不存在，重新操作！')

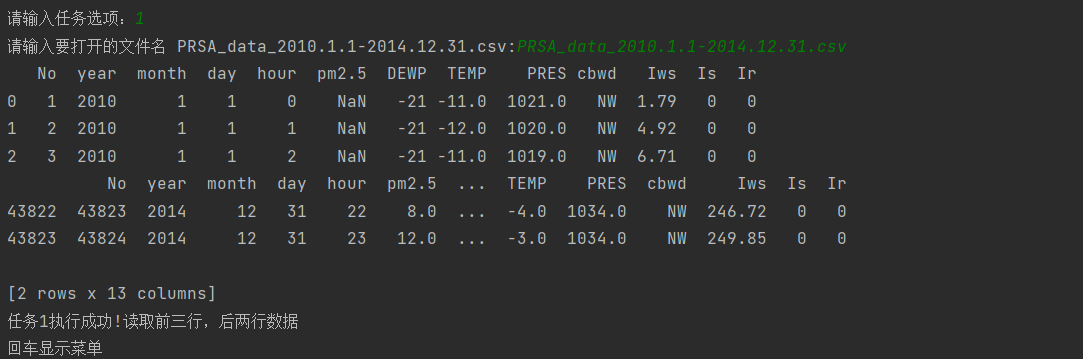
## 上机操作

使用pycharm运行代码，实例如下：

1. 进入主菜单页面



1. 输入‘1’，进入数据预览模块，人工输入文件名，浏览三行，后两行



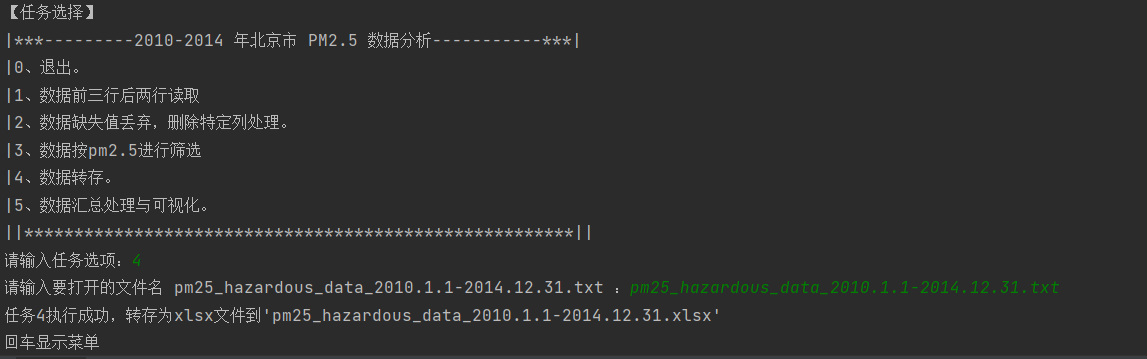
1. 输入‘2’，回车进入数据预处理模块，成功处理文件



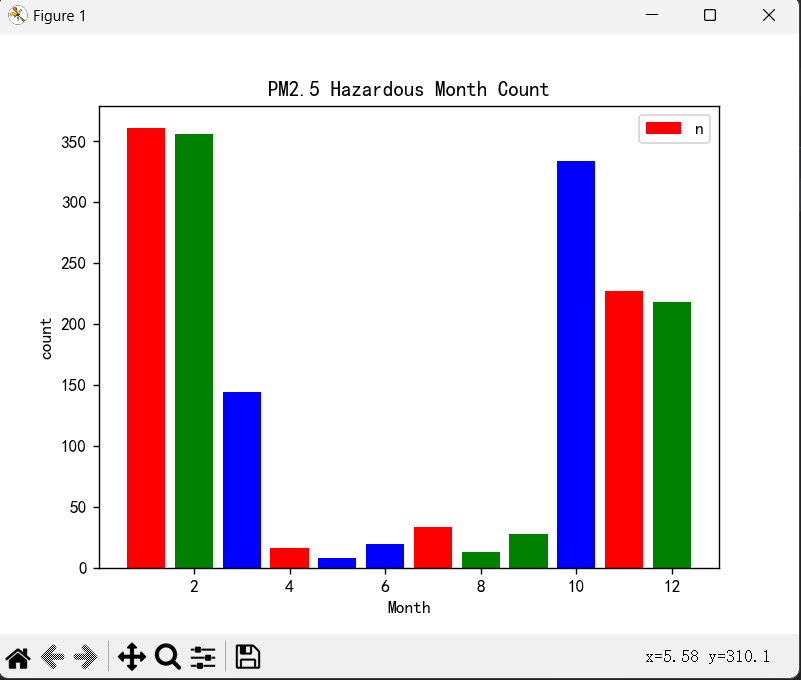
1. 输入‘3’，进入数据筛选模块，并将筛选后的数据保存至新文件

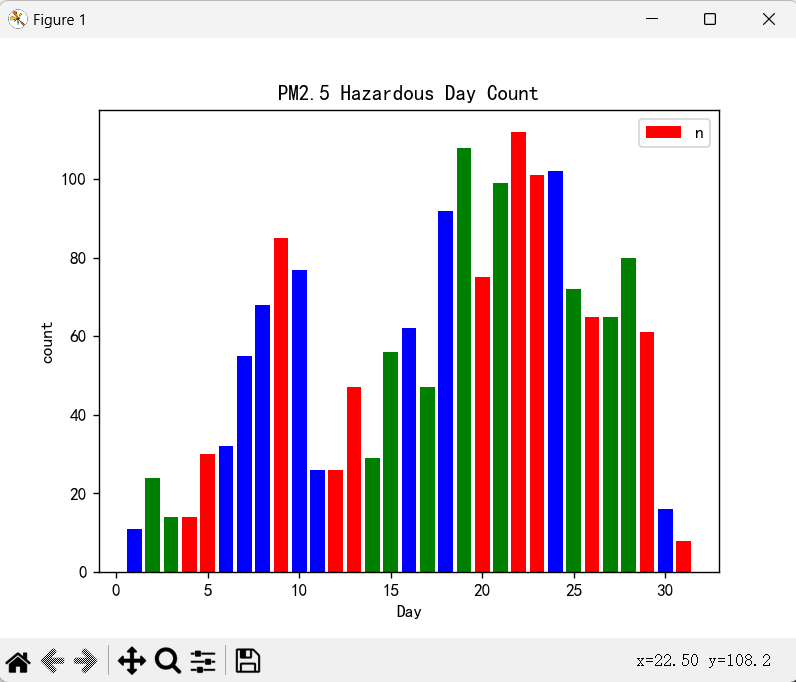


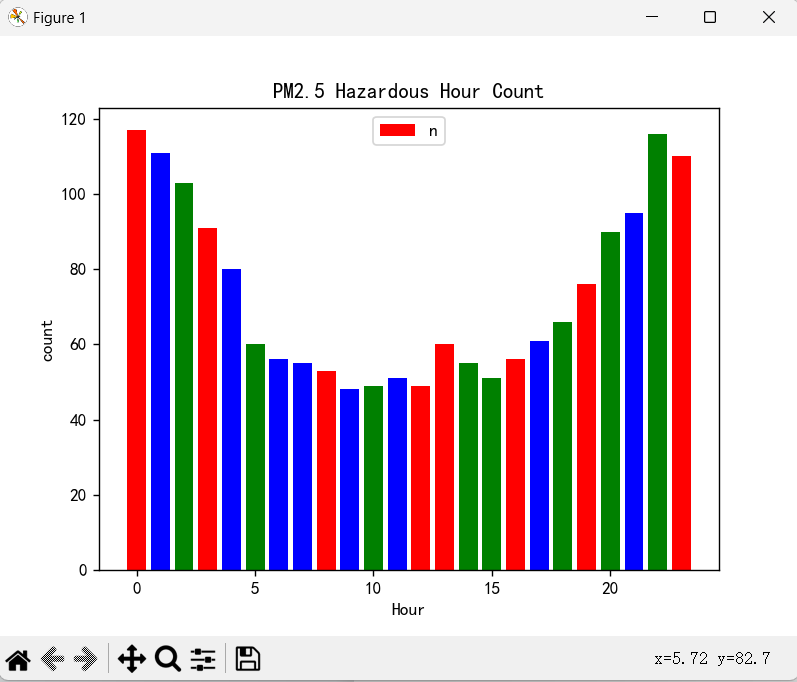
1. 输入‘4’，进入数据转存模块，将数据转存为txt文件。



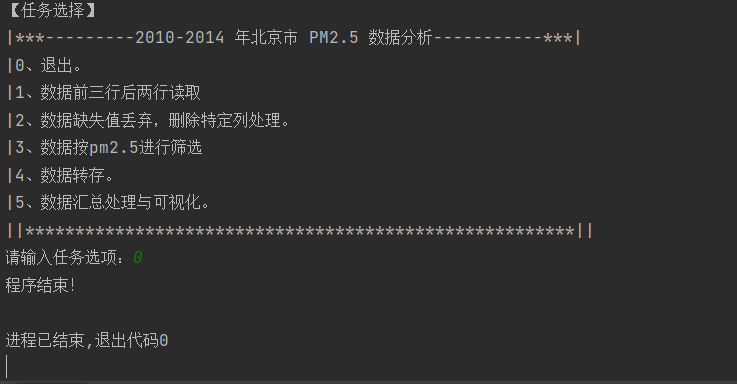
1. 输入‘5’，回车进入数据分析和可视化模块，输入文件名后执行，成功绘三种直方图。







1. 输入‘0’，退出程序。



## **结果分析**

（1）

分析2010-2014年北京市PM2.5>300数据得知，出现PM2.5超过300的次数最少的月份为5月，共出现了8次；出现次数最少的日期为31号，也是8次；出现次数最少的小时为9点，共出现了48次。

（2）

统计2010-2014年北京市PM2.5>300数据得知，出现PM2.5超过300的次数最多的月份为1月，共出现了361次；出现次数最多的日期为22号，共出现了112次；出现次数最多的小时为0点，共出现了117次。

（3）

通过数据的可视化展示，我们可以看到三个柱状图，显示了2010-2014年北京市PM2.5超过300的次数在不同的月份、日期和小时上的分布情况。这些图表表明，无论是哪个月份、日期还是小时，都有可能出现PM2.5超过300的情况。因此，无论何时外出，我们都应该做好相应的防护措施。此外，这些图表也间接反映了北京市空气质量有待提高。

（4）

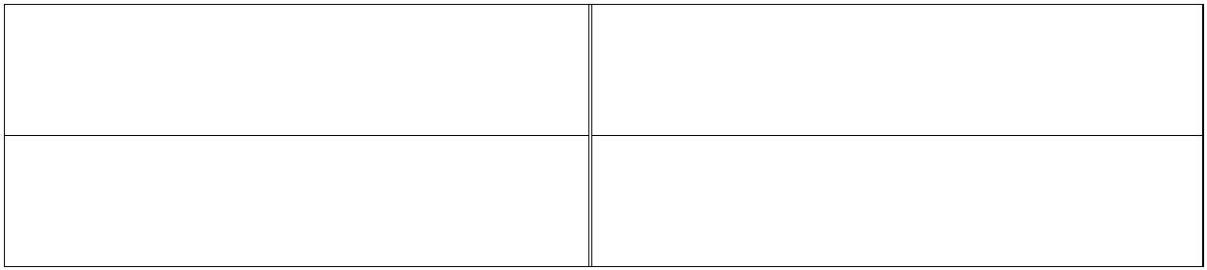
综合以上分析，我们可以得出结论：在2010-2014年期间，北京市1月22号的0点PM2.5超过300的可能性最大。因此，在这个时间段，最好减少外出，如果需要外出，务必做好防护措施。另一方面，在2010-2014年的5月31号的9点，PM2.5超过300的可能性最小，相对较利于外出。然而，即使在这

## 七、致谢

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: