[](http://www.cug.edu.cn/new/image/001/xinxiaoming.jpg)

**课程报告**

**课程名称： Python语言课程设计B**

**课程题目：2010-2014北京市PM2.5数据分析**

**学生姓名： 胡佳琪**

**学生班级： 22经管8班**

**学生学号： 08J22809**

**2023 年 6 月**

**目录**

[课程名称：Python语言课程设计B 1](#_Toc13947)

[一、题目要求：2010-2014北京市PM2.5数据分析 3](#_Toc3577)

[1.1根据以上数据处理任务 4](#_Toc31625)

[1.2根据以上统计结果，书写不少于 300 字的结果分析 4](#_Toc5679)

[二、需求分析 4](#_Toc7663)

[三、概要设计 4](#_Toc14984)

[四、详细设计 5](#_Toc22622)

[4.1主函数模块详细设计 5](#_Toc13928)

[4.2功能选择模块详细设计 5](#_Toc32348)

[4.3程序健壮性和异常控制模块详细设计 7](#_Toc9789)

[4.4数据读取模块详细设计 7](#_Toc32145)

[4.5数据预处理及导出模块详细设计 7](#_Toc24608)

[4.6数据选择及导出模块详细设计 8](#_Toc15999)

[4.7数据转存及导出模块详细设计 8](#_Toc4655)

[4.8数据统计及可视化模块详细设计 9](#_Toc2024)

[五、 完整代码及运行结果 10](#_Toc23766)

[六、结果分析 21](#_Toc4212)

[七、致谢 21](#_Toc9776)

## **一、题目要求：2010-2014北京市PM2.5数据分析**

已知2010-2014北京市PM2.5数据PRSA\_data\_2010. 1. 1-2014. 12.31.csv，包含 No（行序号）、year（行数据记录年份）、month（行数据记录月份）、day（行数据记录日）、hour（行数据记录时间）、pm2.5（PM2.5 浓度）、DEWP（Dew Point）、TEMP（温度）、PRES（压力）、cbwd（风向）、Iws（风力）、Is（累计下雪时长）、Ir（累计下雨时长）13个字段，如表1所示。要求根据所学知识完成以下任务。

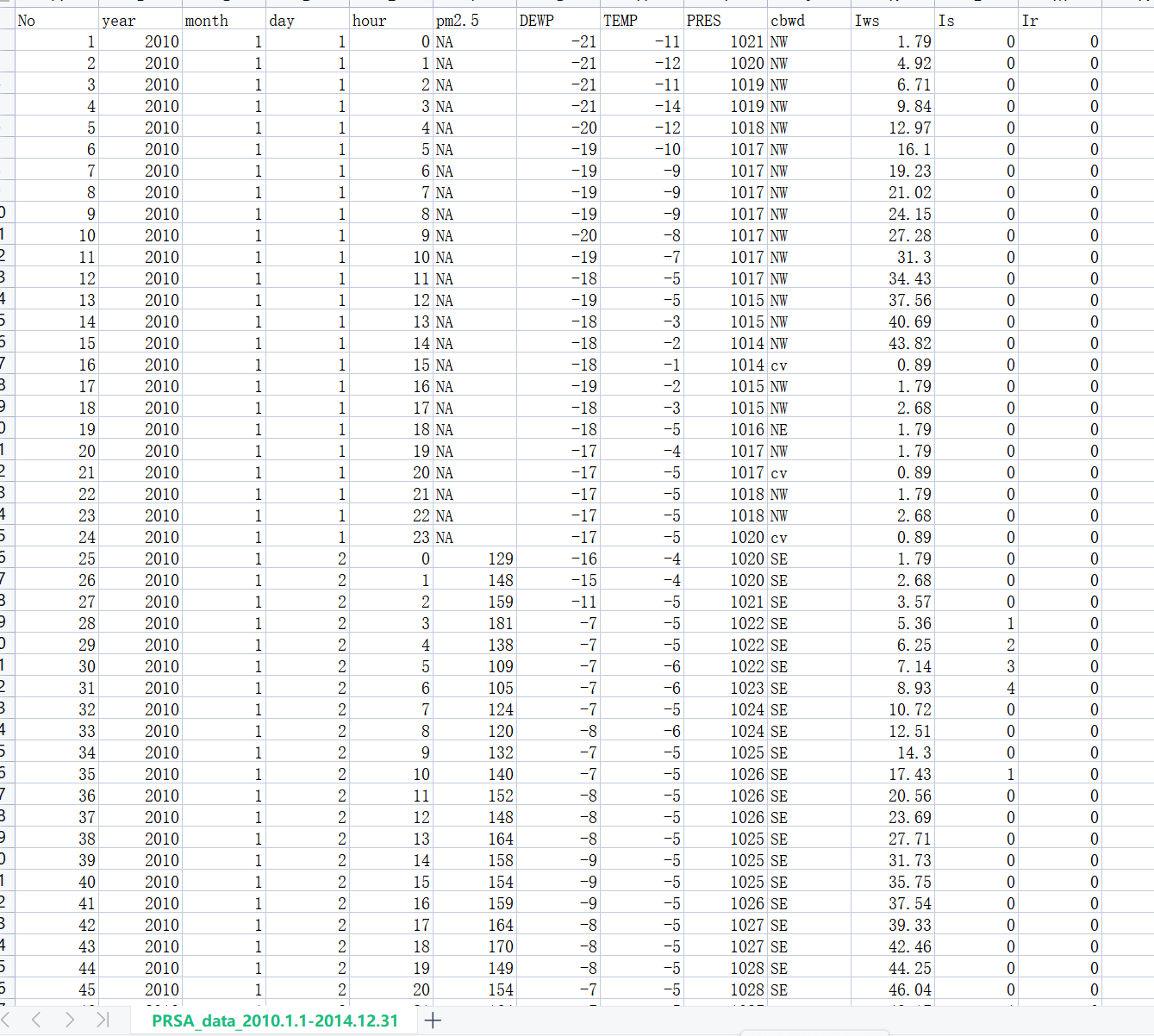


表 1：2010-2014北京市PM2.5数据（节选）

【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1所示。

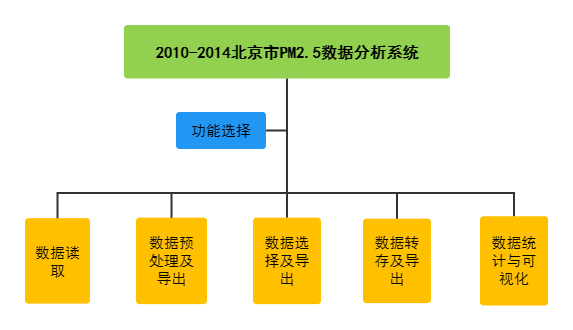


图 1：2010-2014北京市PM2.5数据分析系统概要设计

## 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 】



### 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。

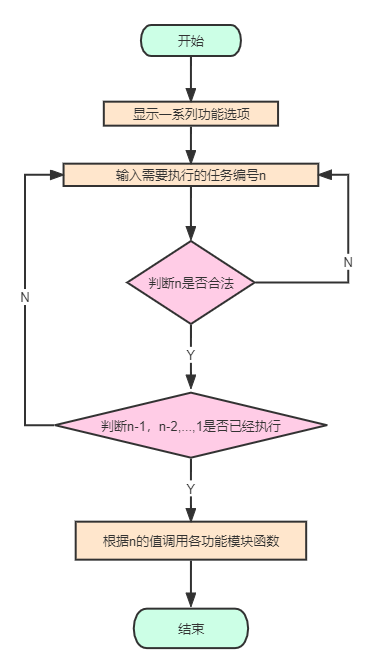


图 2：功能选择模块程序流程图

【关键代码】

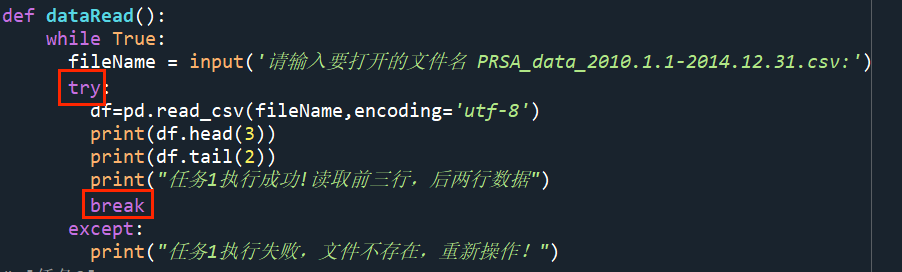
4.3程序健壮性和异常控制模块详细设计

【分析】

为保证程序的合理运行，程序中应提供必要的异常控制代码，保证程序的健壮性。

在本题中，在各个任务执行过程中进行必要的判断(如文件是否存在、输入)来保持程序的健壮性；用try语句进行程序异常控制{要注意break子句的使用来断开循环}。

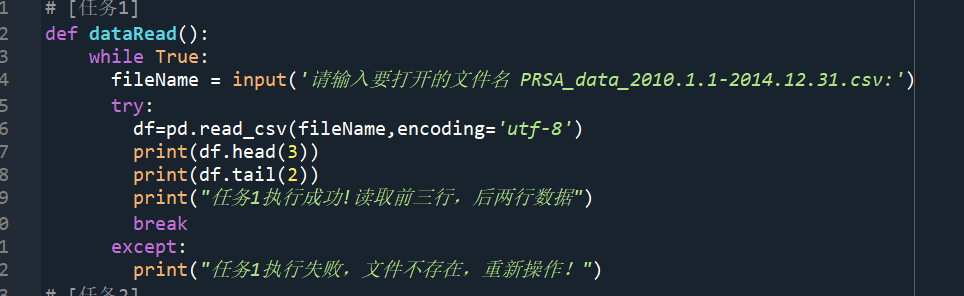
【关键代码】



4.4数据读取模块详细设计

【分析】

数据读取除了文件操作open函数外，更为便捷的是Pandas提供的read\_csv和read\_excel两个功能，因此本示例选择read\_csv来进行数据读取并且用对DataFrame对象的head（）和tail（）来查看前三行、后两行。

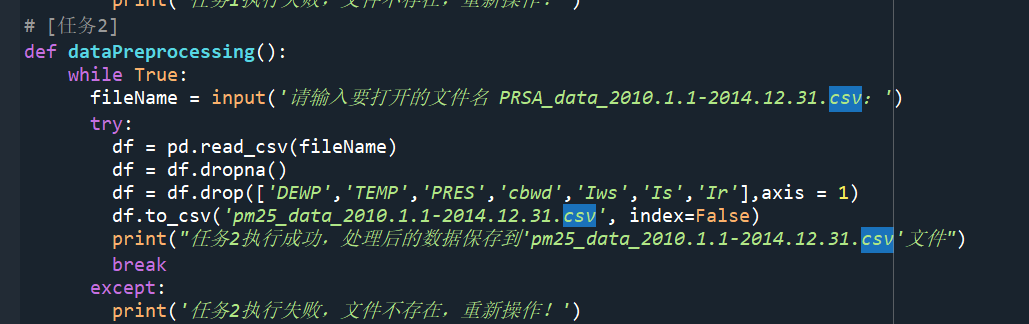


4.5数据预处理及导出模块详细设计

【分析】

数据预处理主要包含缺失值丢弃处理、缺失值补充、重复值丢弃、异常值丢弃或规约等处理方式。本示例要求将缺失值丢弃处理，直接利用Pandas中的dropna()功能即可。同时运用drop函数删除行、删除列，由于drop函数默认删除行，列需要加axis = 1{df = drop(columns = ['A']) #此方法同样可以删除列,这里没有采用该法方法。}最后利用pandas中的to\_csv导出文件。

【关键代码】

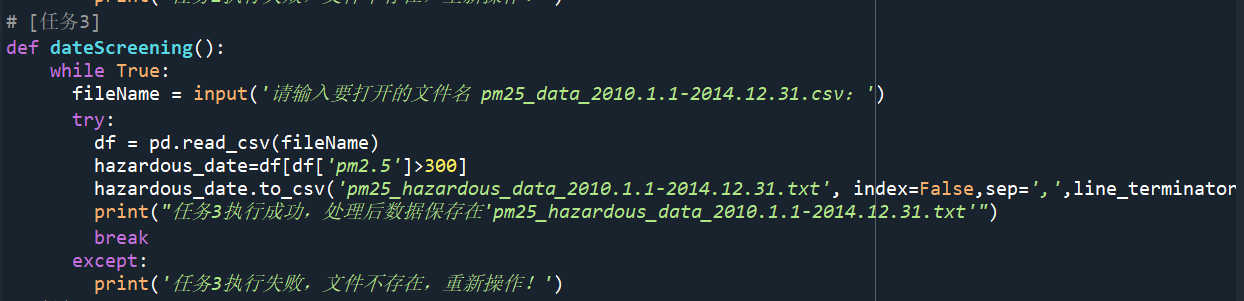


### 4.6数据选择及导出模块详细设计

【分析】

在进行数据处理和分析时，经常需要对数据进行处理和清洗。通过这两个环节可以对数据进行规范化、去重、筛选、转换等操作，从而使得数据更加适合后续的分析、建模与可视化。因此本题中选取用[>]提取列指定列满足条件数据，用sep参数控制分割方式,以","隔开。

【关键代码】



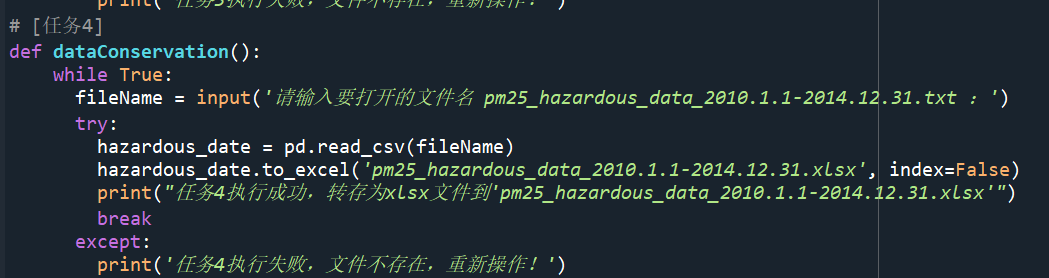
### 4.7数据转存及导出模块详细设计

【分析】

在进行数据处理和分析时，经常会需要对数据格式按照所需进行转换，使得不同格式之间的数据交换更加便捷和高效，也可以帮助我们将数据转化为可视化所需的格式和维度。这可以通过python中的pandas库、xlrd/openpyxl库、csv库、json库等来实现。而数据导出除了文件操作的writer(s)和writelines(s)之外，更为便捷的是Pandas提供的to\_csv和to\_excel用法。

因此，本题运用pandas中的to\_excel来转存文件。

【关键代码】

****

### 4.8数据统计及可视化模块详细设计

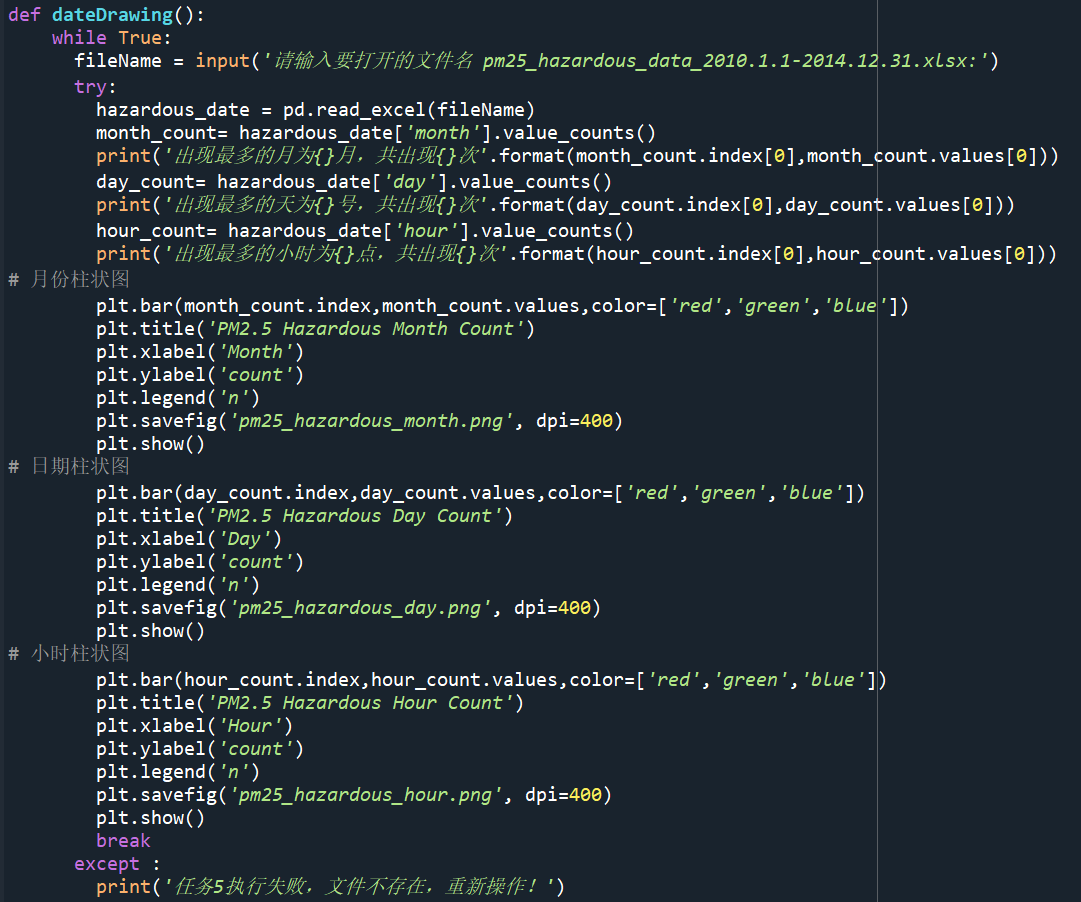
【分析】

数据统计，是数据分析的重要组成部分。

数据可视化，也是数据分析的重要组成部分。在 Python生态系统中,包括诸如matplotlib、Scaborm、HoloViews. Altair、PyQtGraph、ggplot、Bokch、pygal、VisPy、NetworkX、Plotly、geoplotlib、folium、Gleam、vincent、mpld3、python-igraph、missingno、Mayavi2、Leather等等众多的扩展库实现数据可视化功能。不同的可视化库在不同方面具有不同的偏重，如 Seaborn、Altair更偏重于专业统计图表绘制，PyQ1Graph、VisPy、Mayavi2更适合于数学、工程等领域制图，NctworkX.python-igraph 更适合于网络研究和分析制图geoplotlib、folium适合于绘制地图，等等。然而，在诸多的第三方可视化库中，matplotlib是Python中最为著名的绘图系统，很多其他的绘图系统如 seaborn 也是由其封装而来。此外，Pandas也提供了自己的绘图接口,并结合matlab实现更为丰富的功能。

结合所学及题目所需，本题用value\_counts()函数对出现次数最多的month、day、hour进行计数按照指定列统计出现个数{value\_counts()所得数据自动按出现个数降序排列，无需再排序,直接取第一个数据即可；format方法字符串格式化。index和values按索引或值取数}最后基于plotlib库中的bar()函数绘制条形图来实现柱状图的绘制，{也可以用直方图函数hist直接生成直方图}

【关键代码】



## 完整代码及运行结果

【完整代码】

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import os

import pandas as pd

import matplotlib.pyplot as plt

# [任务1]

def dataRead():

while True:

fileName = input('请输入要打开的文件名 PRSA\_data\_2010.1.1-2014.12.31.csv:')

try:

df=pd.read\_csv(fileName,encoding='utf-8')

print(df.head(3))

print(df.tail(2))

print("任务1执行成功!读取前三行，后两行数据")

break

except:

print("任务1执行失败，文件不存在，重新操作！")

# [任务2]

def dataPreprocessing():

while True:

fileName = input('请输入要打开的文件名 PRSA\_data\_2010.1.1-2014.12.31.csv：')

try:

df = pd.read\_csv(fileName)

df = df.dropna()

df = df.drop(['DEWP','TEMP','PRES','cbwd','Iws','Is','Ir'],axis = 1)

df.to\_csv('pm25\_data\_2010.1.1-2014.12.31.csv', index=False)

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")

break

except:

print('任务2执行失败，文件不存在，重新操作！')

# [任务3]

def dateScreening():

while True:

fileName = input('请输入要打开的文件名 pm25\_data\_2010.1.1-2014.12.31.csv：')

try:

df = pd.read\_csv(fileName)

hazardous\_date=df[df['pm2.5']>300]

hazardous\_date.to\_csv('pm25\_hazardous\_data\_2010.1.1-2014.12.31.txt', index=False,sep=',',line\_terminator='\n',)

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'")

break

except:

print('任务3执行失败，文件不存在，重新操作！')

# [任务4]

def dataConservation():

while True:

fileName = input('请输入要打开的文件名 pm25\_hazardous\_data\_2010.1.1-2014.12.31.txt ：')

try:

hazardous\_date = pd.read\_csv(fileName)

hazardous\_date.to\_excel('pm25\_hazardous\_data\_2010.1.1-2014.12.31.xlsx', index=False)

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'")

break

except:

print('任务4执行失败，文件不存在，重新操作！')

# [任务5]

def dateDrawing():

while True:

fileName = input('请输入要打开的文件名 pm25\_hazardous\_data\_2010.1.1-2014.12.31.xlsx:')

try:

hazardous\_date = pd.read\_excel(fileName)

month\_count= hazardous\_date['month'].value\_counts()

print('出现最多的月为{}月，共出现{}次'.format(month\_count.index[0],month\_count.values[0]))

day\_count= hazardous\_date['day'].value\_counts()

print('出现最多的天为{}号，共出现{}次'.format(day\_count.index[0],day\_count.values[0]))

hour\_count= hazardous\_date['hour'].value\_counts()

print('出现最多的小时为{}点，共出现{}次'.format(hour\_count.index[0],hour\_count.values[0]))

# 月份柱状图

plt.bar(month\_count.index,month\_count.values,color=['red','green','blue'])

plt.title('PM2.5 Hazardous Month Count')

plt.xlabel('Month')

plt.ylabel('count')

plt.legend('n')

plt.savefig('pm25\_hazardous\_month.png', dpi=400)

plt.show()

# 日期柱状图

plt.bar(day\_count.index,day\_count.values,color=['red','green','blue'])

plt.title('PM2.5 Hazardous Day Count')

plt.xlabel('Day')

plt.ylabel('count')

plt.legend('n')

plt.savefig('pm25\_hazardous\_day.png', dpi=400)

plt.show()

# 小时柱状图

plt.bar(hour\_count.index,hour\_count.values,color=['red','green','blue'])

plt.title('PM2.5 Hazardous Hour Count')

plt.xlabel('Hour')

plt.ylabel('count')

plt.legend('n')

plt.savefig('pm25\_hazardous\_hour.png', dpi=400)

plt.show()

break

except :

print('任务5执行失败，文件不存在，重新操作！')

def menu():

print('【任务选择】\n'

'|\*\*\*---------2010-2014 年北京市 PM2.5 数据分析-----------\*\*\*|\n'

'|0、退出。\n'

'|1、数据前三行后两行读取\n'

'|2、数据缺失值丢弃，删除特定列处理。\n'

'|3、数据按pm2.5进行筛选\n'

'|4、数据转存。\n'

'|5、数据汇总处理与可视化。\n'

'||\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*||')

#打印菜单

def task():

while True:

menu() #打印菜单

num = input("请输入任务选项：") #进行任务选择

if num == '1':

dataRead()

elif num == '2':

dataPreprocessing()

elif num == '3':

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

dateScreening()

else:

print('未能执行当前选项，请先执行前面的选项！')

elif num == '4':

if os.path.exists('pm25\_hazardous\_data\_2010.1.1-2014.12.31.txt'):

dataConservation()

else:

print('未能执行当前选项，请先执行前面的选项！')

elif num == '5':

if os.path.exists('pm25\_hazardous\_data\_2010.1.1-2014.12.31.xlsx'):

dateDrawing()

else:

print('未能执行当前选项，请先执行前面的选项！')

elif num == '0':

print('程序结束！')

break #结束程序出口

else:

print('输入选项有误')

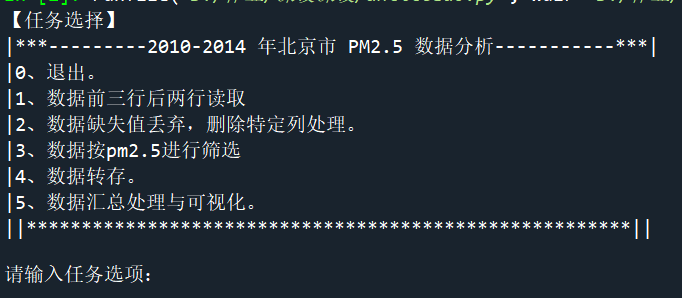
input("回车显示菜单")

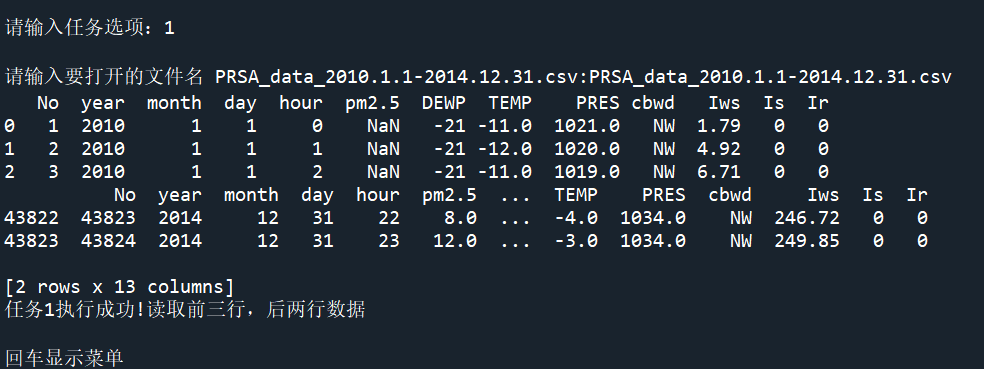
if \_\_name\_\_ == '\_\_main\_\_':

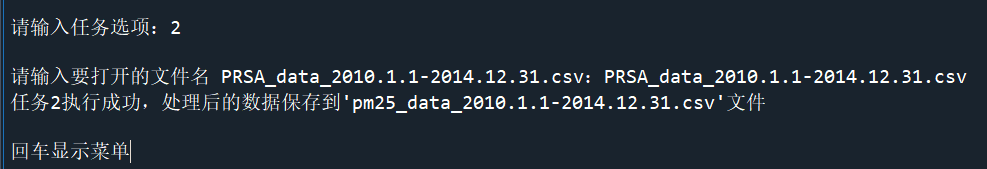
task() # 调用功能选择函数

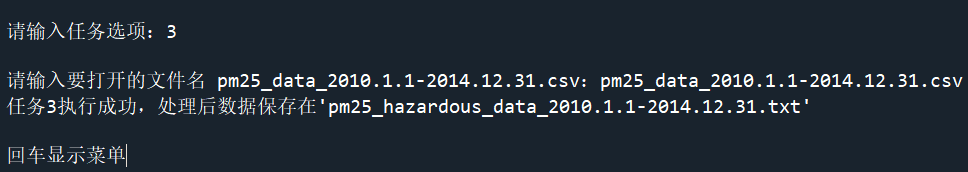
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

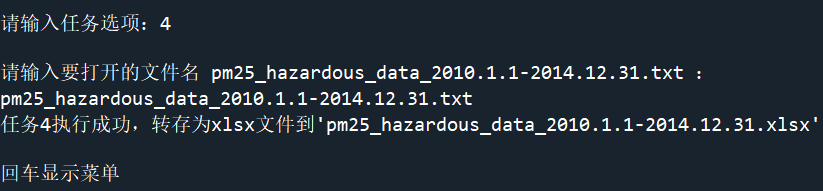
【运行结果】



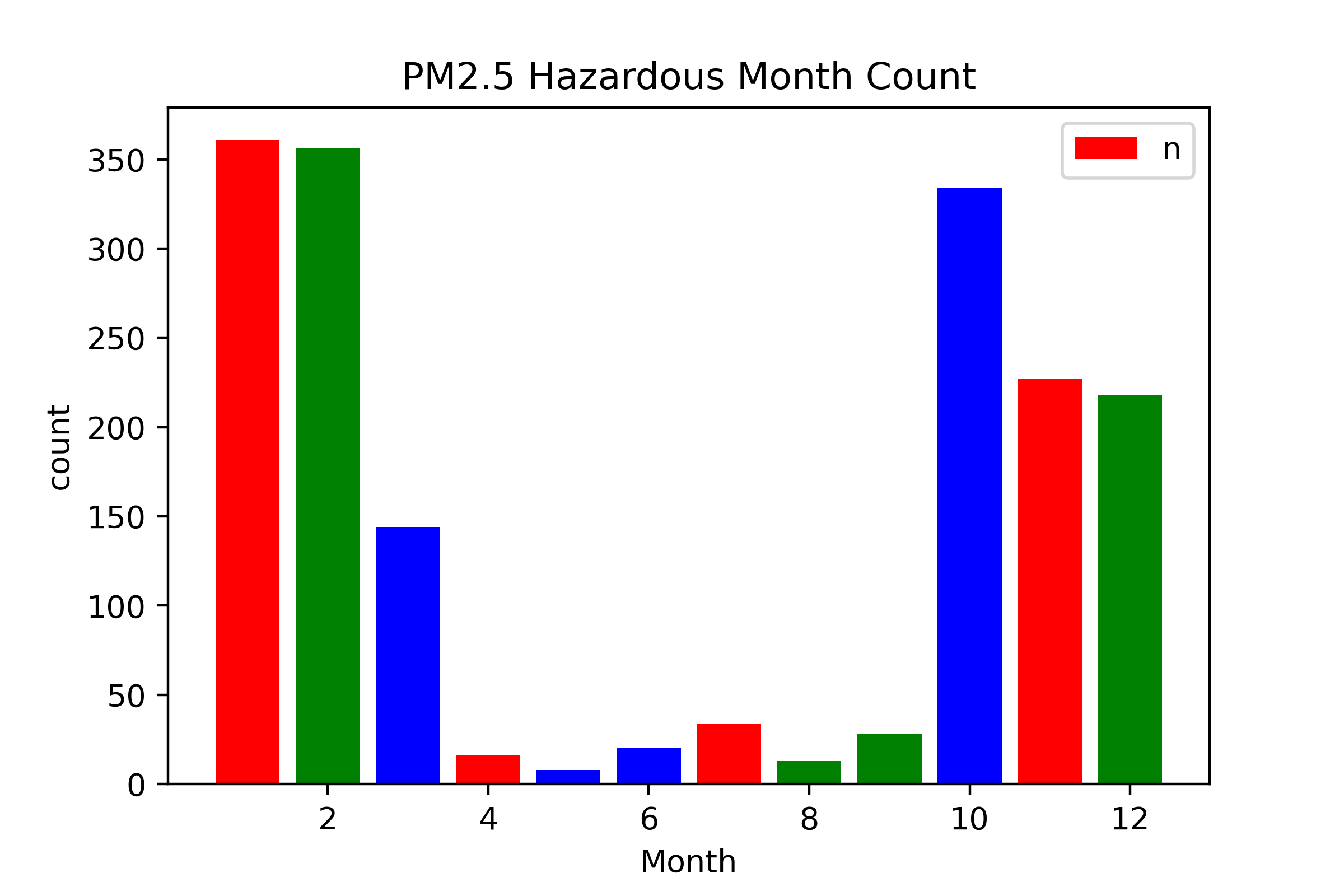
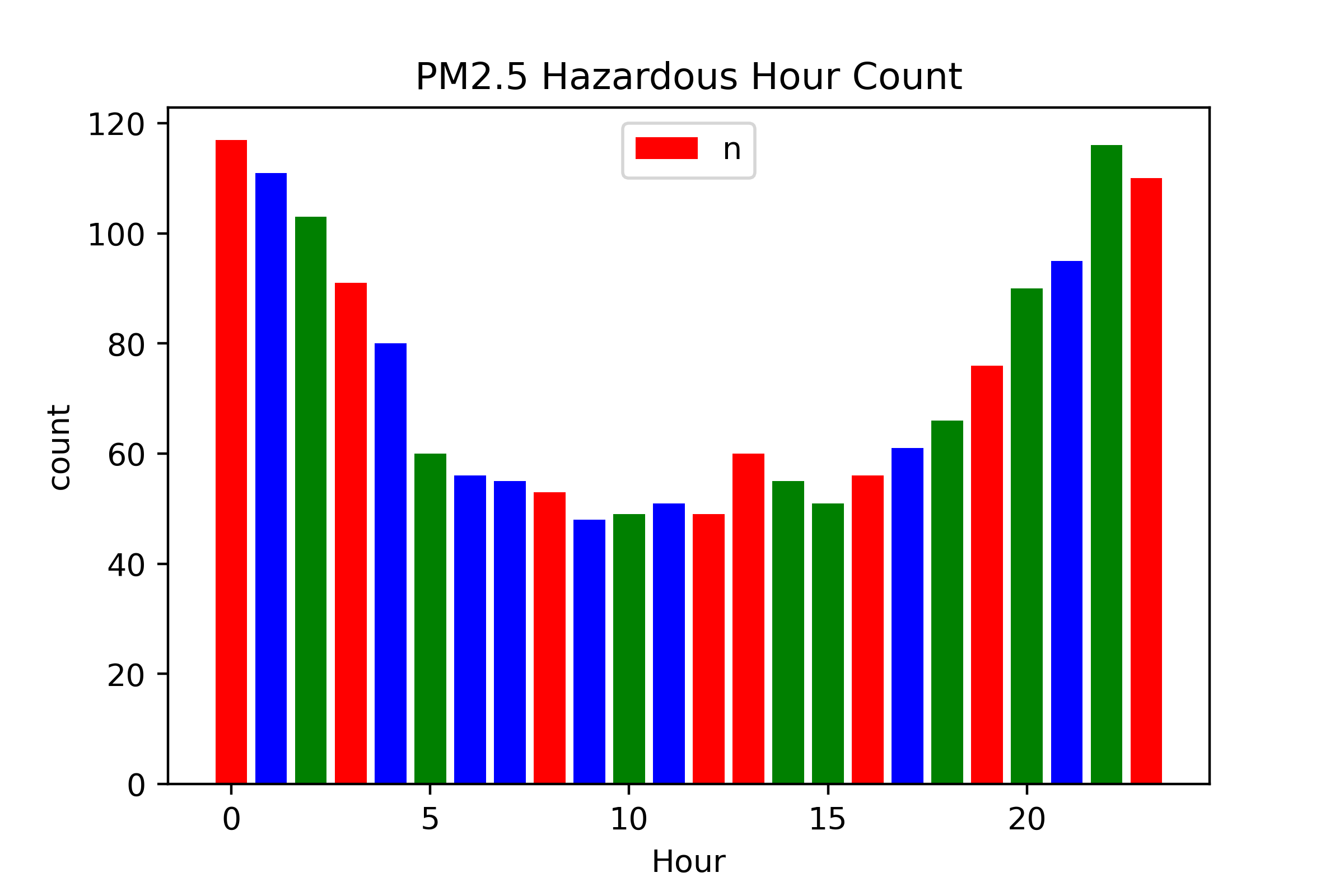
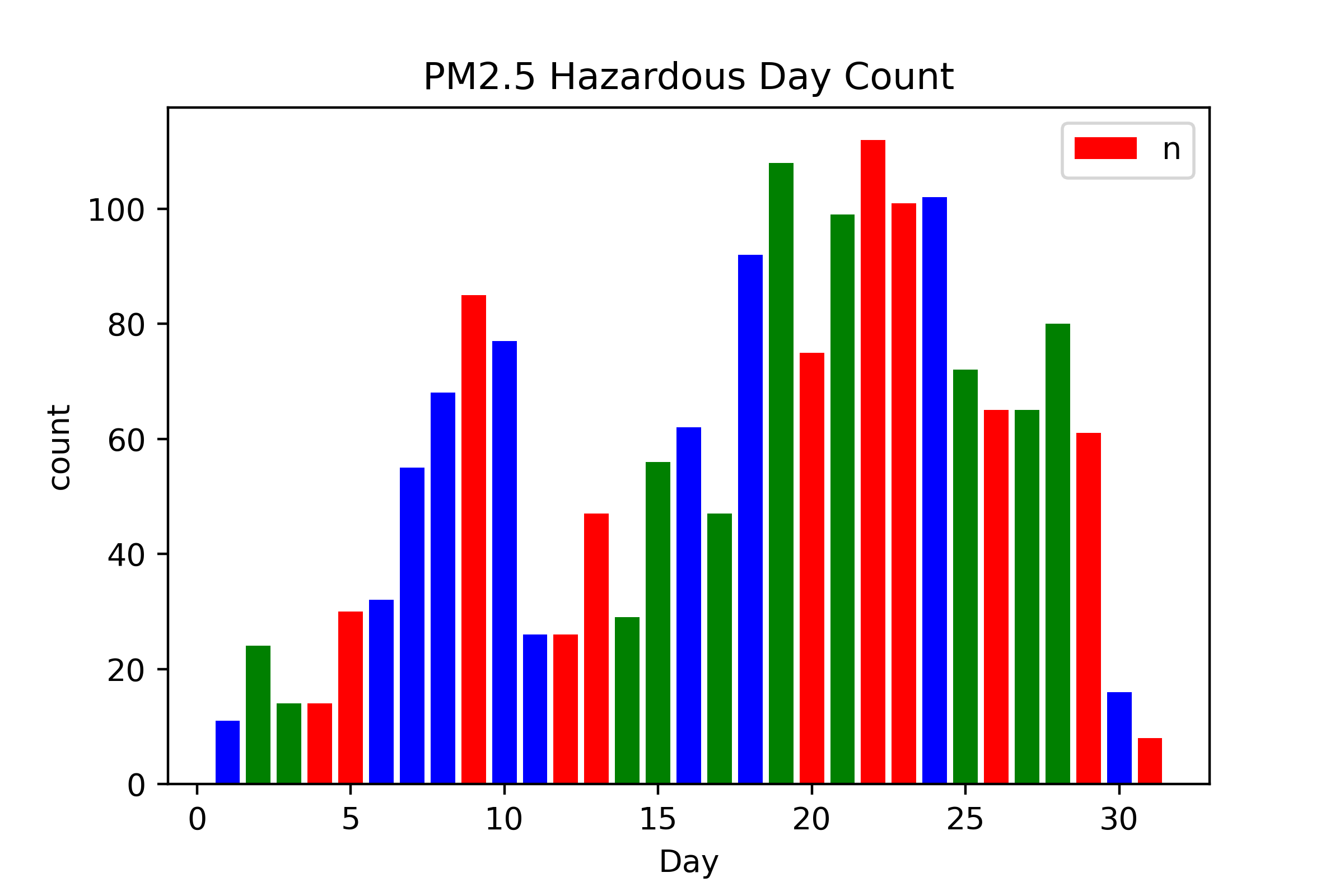












## 六**、结果分析**

（1）通过筛选并查看2010-2014年北京市PM2.5>300的数据而转存得到的excel表可知：2010-2014年北京市PM2.5>300出现最少的月为5月，共出现8次；出现最少的天为31号，共出现8次；出现最少的小时为9点，共出现48次。

（2）通过统计2010-2014年北京市PM2.5>300的数据而转存得到的excel表可知：2010-2014年北京市PM2.5>300出现最多的月为1月，共出现361次；出现最多的天为22号，共出现112次；出现最多的小时为0点，共出现117次。

（3）通过数据可视化展示出的三个柱状图图可知：2010-2014年北京市无论是哪个月哪天哪个时间段，都有可能存在PM2.5>300的情况，因此无论什么时候出行都应该做好防护；也侧面反映出北京市空气质量有待提高。

（4）结合上述分析可知，在2010-2014年间北京市1月22号零时PM2.5>300出现的可能性最大，因此在这个时间段最好减少外出，若要出行一定要做好防护措施；在2010-2014年间北京市5月31号九时PM2.5>300出现的可能性最小，在一定程度上利于出行，但是也应该做好相应的防护准备，健康安全出行。

## 七、致谢

学习python短短几个月，感触颇多。于我而言，这是一门全新的未知的领域。在选课之初，我便心里先打了退堂鼓。然而经过短短几个月的学习，从最开始照着例题敲都频繁报错，到现在我能够自己敲下整个代码，并独立能够写完这份报告，感觉还是有点小骄傲的。文艺点可以说：“轻舟已过万重山。”

感谢任老师的悉心教导，虽然每次上课抽人提问总是让我提心吊胆，但是我也确实因此记住了很多知识点；感谢我的朋友们，在我每次问他们相对幼稚问题的时候，他们总是耐心地解答，让我一点点地进步.......

行文自此，也许python课程的学习到这里就告一段落了，但是我学习的道路，对于新事物探索的脚步，从未停止。加油吧！