2014-2020 年长沙地区空气质量与气象因素的相关性分析

唐亚周 519021910804
电子信息与电气工程学院 计算机科学与技术专业
指导老师：皮玲

摘 要 一直以来，空气质量问题都是我国城市化发展过程中亟待解决的难题。长沙市作为一座快速发展的新一线城市，空气质量问题自然十分关键。本文以天为单位，通过分析以 $PM_{2.5}$ 为代表的空气质量指数和当天的天气状况、气温、风速等指标的关系，得出一系列结论。

关键词 空气质量，长沙，天气，量化分析

目录

1 研究背景

近期召开的十九届五中全会审议通过的《中共中央关于制定国民经济和社会发展第十四个五年规划和二〇三五年远景目标的建议》（以下简称《建议》），从多个方面对生态文明建设和生态环境保护作出重要部署、提出明确要求，为做好“十四五”生态环境保护工作指明了前进方向、提供了根本遵循。2021 年是我国现代化建设进程中具有特殊重要性的一年，也是“十四五”时期的开局之年，要立足新发展阶段，坚持新发展理念，构建新发展格局，把生态文明建设和生态环境保护作为推动高质量发展的题中应有之义，进一步凸显其重要地位和关键作用，抓紧谋划“十四五”生态环境保护工作，坚持稳中求进工作总基调，对标对表 2035 年远景目标，牢固树立落实绿水青山就是金山银山的理念，促进经济社会发展全面绿色转型，加快推动绿色低碳发展，持续改善环境质量，提升生态系统质量和稳定性，全面提高资源利用效率，更加注重系统观念在生态环境保护工作中的科学运用和实践深化，突出精准治污、科学治污、依法治污，更好发挥生态环境保护对高质量发展、构建新发展格局的支撑服务保障作用，为开启全面建设社会主义现代化国家新征程、向第二个百年奋斗目标进军奠定坚实基础。[1]

在这样的大背景下，借这次概率统计课程大作业的机会，本人决定对家乡长沙的空气质量进行一次研究。

2 前期准备

2.1 文献调研

经过查阅资料[2] 得知，空气质量的主要影响因素有局地污染和西北地区沙尘传输造成的自然降尘、气象要素（降水量、风速、逆温等）、地形的空间差异和人类活动等多个方面。考虑到长沙市的地理特征、产业结构以及数据收集的难易程度，我选择对气象因素和空气质量的关系进行研究。

2.2 数据搜集及处理

本人在网络[3] 上搜集到 2014 年到 2020 年长沙市的空气质量指数，再利用 python 爬虫在网络[4] 上获取 2014 年到 2020 年长沙市的气象要素数据。然后去除数据集中的无效数据，得到 2531 个有效数据。

3 数据分析

3.1 空气质量指数与温度的关系

收集到的数据中，有当日最高温度和最低温度两个数据，将这两个数据求平均值处理，得到的值作为一天的温度。以年份为单位，将温度及 $PM_{2.5}$ 和日期的关系进行可视化，如图 1所示

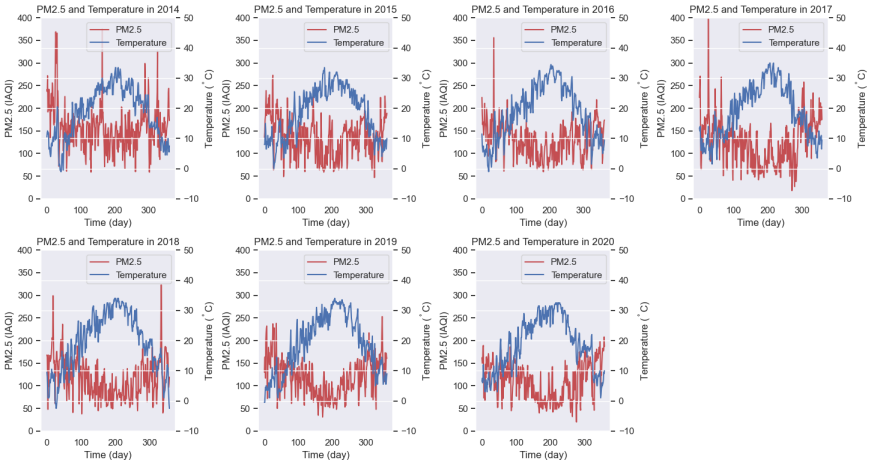

图 1: 2014-2020 年长沙地区各年 $PM_{2.5}$ 指数及气温与日期的关系图

可以看出，总体上 $PM_{2.5}$ 指数与气温呈负相关。然后对于 $PM_{2.5}$ 指数和气温绘制二维直方图，如图 2所示，并使用 Python 内置函数计算相关系数。

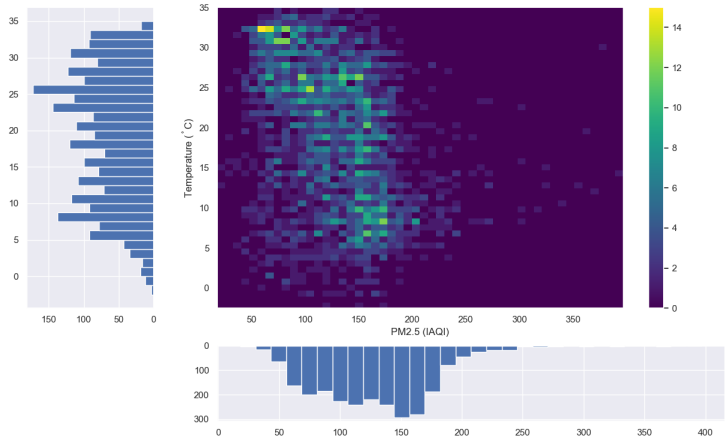

图 2: 2014-2020 年长沙地区 $PM_{2.5}$ 指数与气温的分布图

$$\rho_{XY} = \frac{\operatorname{cov}(X,Y)}{\sqrt{\mathrm{D}(X)\cdot\mathrm{D}(Y)}} = -0.35 \tag{1}$$

由此看出，$PM_{2.5}$ 指数与温度呈一定的负相关，但相关性并不明显。

3.2 空气质量指数与天气状况的关系

通过遍历数据，我们得到了 7 年来长沙地区出现过的所有天气情况，并按照降雨量/降雪量的大小将其分级，如表 1所示。

将数据中每天的两个天气信息及其对应的级别均记录下来，也就是说一个空气质量的信息对应着两个天气状况信息。得到 1 级天气 2568 次、2 级天气 544 次、3 级天气 1592 次、四级天气 358 次。

表 1: 天气状况及人为分级

天气情况	级别	分级依据
晴	1	一般为干燥
多云	1	一般为干燥天气
阴	2	一般湿度比较大，但没有降水
小雨	3	一般降水量较少
小到中雨	3	一般降水量较少
阵雨	3	一般降水量较少
雷阵雨	3	一般降水量较少
雨	3	一般降水量较少
中雨	4	一般降水量较大
中到大雨	4	一般降水量较大
大雨	4	一般降水量较大
大到暴雨	4	一般降水量较大
暴雨	4	一般降水量较大
冻雨	3	一般降水量较少
雨夹雪	3	一般降水量较少
小雪	3	一般降水量较少
小雪-中雪	3	一般降水量较少
中雪	4	一般降水量较大
中到大雪	4	一般降水量较大
大雪	4	一般降水量较大

对四种天气下的 $PM_{2.5}$ 指数做统计，得到直方图如图 3所示。

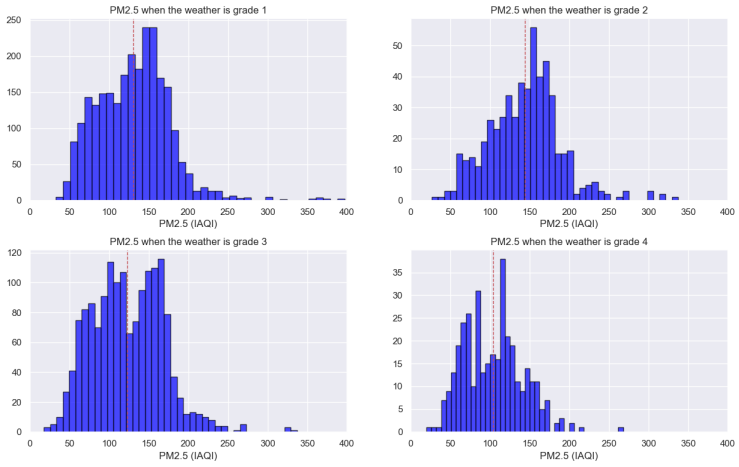

图 3: 2014-2020 年间长沙地区 4 钟天气情况下的 $PM_{2.5}$ 指数分布直方图

由 $E(X) = \sum_{i=1}^{n} x_i p_i$ 和 $D(X) = \sum_{i=1}^{n} [x_i - E(X)]^2 p_i$ 可计算出四种天气下 $PM_{2.5}$ 指数的期望和方差，如表 2所示。

表 2: 各级别天气下 $PM_{2.5}$ 指数的样本期望和方差

天气级别	期望	方差
级别 1	130.87	2032.00
级别 2	143.81	2051.42
级别 3	123.16	1993.46
级别 4	104.01	1392.35

表 3: 各级别天气下中度污染及以上天数的样本期望和数量

天气级别	期望	出现次数	占总数比例
级别 1	177.79	867	33.67%
级别 2	180.49	253	46.51%
级别 3	175.75	469	29.46%
级别 4	169.62	43	12.01%

可以看到，总体上来说，当天气的级别越高，也就是降水量越大、空气越湿润时，$PM_{2.5}$ 指数总体上越小。另外，我们可以按照国家标准[5] 里所规定的”中度污染“以上的标准，也就是 $PM_{2.5}$ 的 IAQI 值大于等于 151 的情况下，对四种级别的天气进行统计，如图 4和表 3所示。可以看到，总体上来说，当天气的级别越高，出现”中度污染“及以上的比例要减小。

但是在级别 2 的天气（阴天）时，$PM_{2.5}$ 指数的期望要比级别 1（晴天和多云）要大，出现”中度污染“及以上的次数比例也要高于级别 1。综上所述，$PM_{2.5}$ 指数在不同天气下从高到低排序应该是 $2 > 1 > 3 > 4$。

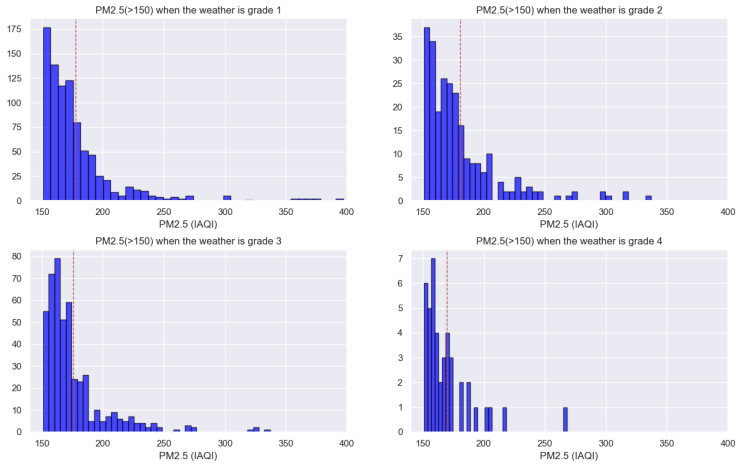

图 4: 2014-2020 年间长沙地区 4 钟天气情况下的 $PM_{2.5}$ 指数分布直方图（中度污染及以上）

3.3 空气质量指数与风速的关系

在收集到的数据中，一天当中可能有 2-4 个风力等级的数据。考虑到风力等级与风速并非线性关系，直接将一天中的所有风力等级数据取平均值作为一天的风力等级并不合理。故同样选择将数据中每天的所有风力等级均记录下来，也就是说一个空气质量的信息对应着 2-4 个风力等级信息。得到 1 级风 1558 次，2 级风 1558 次，3 级风 3450 次，4 级风 680 次，5

表 4: 各级别风力下 $PM_{2.5}$ 指数的样本期望以及“中度污染”及以上天数比例

风力等级	期望	“中度污染”及以上空气出现次数	“中度污染”及以上天数占总数比例
级别 1（软风）	115.96	336	21.57%
级别 2（轻风）	115.96	336	21.57%
级别 3（微风）	133.78	1287	37.30%
级别 4（和风）	112.60	159	23.38%
级别 5（清风）	99.98	9	16.67%
级别 6（强风）	82.80	0	0%

级风 54 次，6 级风 5 次。然后按照 3.2 节中相同的处理方式，如表 4和图 5所示。[1]

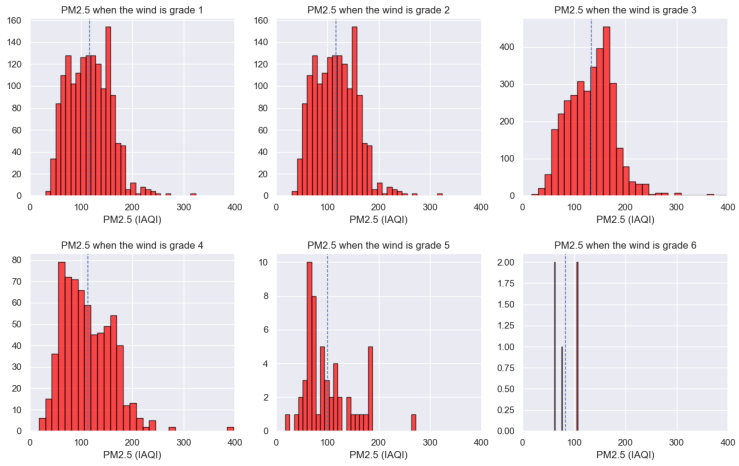

图 5: 2014-2020 年间长沙地区不同风力等级下的 $PM_{2.5}$ 指数分布直方图

由此可见，当风力等级为 3 级时，$PM_{2.5}$ 指数最高，风较小或较大时，$PM_{2.5}$ 指数都较低。

4 结论与展望

4.1 结论

经过对于气温、天气状况（降水量）和风力等级这 3 个影响因素的分析，可以得到以下关于长沙地区 $PM_{2.5}$ 指数的结论：

1. 与气温呈负相关，夏季 $PM_{2.5}$ 指数要低于冬季。但我认为气温并非直接影响 $PM_{2.5}$ 指数的原因。冬季气候寒冷，人们燃煤取暖带来的污染增加，使 $PM_{2.5}$ 指数上升；再加上冬季气候干燥，$PM_{2.5}$ 可以长时间存在于空气中。因此气温只能算作间接影响了空气质量。

2. 阴天时最高，其次是晴天/多云，然后是较小的雨雪天气，最后是较大的雨雪天气。雨雪天气导致空气质量变好，主要是因为其对空气中的悬浮物有着洗涤的作用。同时许多资料[6][7] 显示，阴天并不意味着空气质量差。因此统计得出的阴天 $PM_{2.5}$ 指数最高并没有合理的解释。

[1]其中 1 级风和 2 级风的数据完全相同，原因是在原始数据中，所有的 1、2 级风均被描述为“1-2 级风”。

3. 微风时最高，其次是和风、软风和轻风，在清风和强风时最低。资料[9][?] 显示，风确实可以影响 $PM_{2.5}$ 指数，但是需要较大的风力，而且和风向有关，但本次研究并没有考虑到风向的影响。因此得出结论，风速达到一定程度（4 级及以上）时，风速越大，$PM_{2.5}$ 指数越小，空气质量越好。

4.2 展望

本课题以一定数据对长沙地区的 PM2.5 指数的影响因素进行了分析探究，从气象特征方面给出了结论。但对空气质量影响更大的因素并非气象特征，而是人类活动，比如燃煤、工业废气、汽车尾气等。但由于时间和数据来源的限制，不能在本课题中进行探究。随着气候变化的加剧和空气质量的下降，人类的生存环境面临着较大的危险。因此，我们每个人都要养成节能减排的生活方式，为气候环境的保护贡献自己的力量。

参考文献

[1] 中华人民共和国生态环境部. 生态环境部举行党的十九届五中全会精神辅导报告会 [EB/OL].https://www.mee.gov.cn/xxgk2018/xxgk/xxgk15/202012/t20201212812690.html,2020-12-12.

[2] 李小飞, 张明军, 王圣杰, 赵爱芳, 马潜. 中国空气污染指数变化特征及影响因素分析 [J]. 环境科学,2012,33(06):1936-1943.

[3] The World Air Quality Project.Air Quality Historical Data Platform[DB/OL].https://aqicn.org/data-platform/register/,2020-12-26.

[4] 天气网. 历史天气查询 [DB/OL].http://lishi.tianqi.com/,2020-12-26.

[5] HJ 633—2012, 环境空气质量指数（AQI）技术规定（试行）[S]. 中国: 中华人民共和国生态环境部,2012.

[6] 新京报. 北京空气质量降为二级良专家：阴天不意味着污染 [EB/OL].http://www.chinanews.com/gn/2015/09-05/7505775.shtml,2015-9-5.

[7] 人民网. 成都上空灰蒙蒙？官方解释：阴天不一定是污染天 [EB/OL].http://sc.sina.com.cn/news/b/2020-01-12/detail-iihnzhha1856304.shtml,2020-1-12.

[8] 江晨阳. 韶山风环境及城市植被对 $PM_{2.5}$ 浓度影响的研究 [D]. 湖南大学,2019.

[9] 徐迎春, 谌伟, 刘佳颖, 刘佩廷. 小时气象要素对 $PM_{2.5}$ 浓度的影响 [J]. 环境与发展,2019,31(04):173+221.

A Python 爬虫代码

```python
#coding=utf-8
# 代码参考：https://blog.csdn.net/qq_31903733/article/details/85269367

import io
import sys
import requests
from bs4 import BeautifulSoup
import numpy as np
import pandas as pd
sys.stdout = io.TextIOWrapper(sys.stdout.buffer,
                              encoding='gb18030')  #改变标准输出的默认编码，防止控制台打印乱码


def get_soup(url):
    try:
        r = requests.get(url, timeout=30)
        r.raise_for_status()  #若请求不成功,抛出HTTPError 异常
        #r.encoding = 'gbk'  #与该网站编码匹配
        soup = BeautifulSoup(r.text, 'lxml')
        return soup
    except HTTPError:
        return "Request Error"


def get_data():
    soup = get_soup(url)
    all_weather = soup.find('div',
                            class_="wdetail").find('table').find_all("tr")
    data = list()
    for tr in all_weather[1:]:
        td_li = tr.find_all("td")
        for td in td_li:
            s = td.get_text()
            data.append("".join(s.split()))
    res = np.array(data).reshape(-1, 4)
    return res


def saveTocsv(data, fileName):
    '''
    将天气数据保存至csv文件
    '''
    result_weather = pd.DataFrame(data, columns=['date', 'tq', 'temp', 'wind'])
    result_weather.to_csv(fileName, index=False, encoding='utf-8')
    print('Save all weather success!')


if __name__ == '__main__':
    for i in range(14, 21, 1):
        for j in range(1, 13, 1):
            if j < 10:
                url = "http://www.tianqihoubao.com/lishi/changsha/month/20" + str(
                    i) + '0' + str(j) + ".html"
            else:
                url = "http://www.tianqihoubao.com/lishi/changsha/month/20" + str(
                    i) + str(j) + ".html"
            print(url)
            data = get_data()
            if j < 10:
                saveTocsv(data, str(i) + '0' + str(j) + ".csv")
            else:
                saveTocsv(data, str(i) + str(j) + ".csv")
```

Listing 1: **dataCollect.py**

B Python 数据分析代码

```
import csv
import matplotlib
import matplotlib.pyplot as plt
from scipy.stats import norm
from matplotlib import rc
import re
from numpy import *
import numpy as np
import pandas as pd
import seaborn as sns

sns.set()  # 声明使用 Seaborn 样式

weatherScore = {
    '晴': 1,
    '多云': 1,
    '阴': 2,
    '小雨': 3,
    '小到中雨': 3,
    '中雨': 4,
    '阵雨': 3,
    '雷阵雨': 3,
    '雨': 3,
    '中到大雨': 4,
    '大雨': 4,
    '大到暴雨': 4,
    '暴雨': 4,
    '冻雨': 3,
    '雨夹雪': 3,
    '小雪': 3,
    '小雪-中雪': 3,
    '中雪': 4,
    '中到大雪': 4,
    '大雪': 4
}

weatherList = [[] for i in range(7)]
tempList = [[] for i in range(7)]
windList = [[] for i in range(2531)]
pm25List = [[] for i in range(7)]
pm25DayList = []


def date_transder(my_str):
    pattern = re.compile('(\d+)\D(\d+)\D(\d+)')
    dd = pattern.findall(my_str)
    date_strr = dd[0][0] + '/' + str(int(dd[0][1])) + '/' + str(int(dd[0][2]))
    return date_strr


def getWeather():

    # PM2.5
    with open('changsha-air-quality.csv', newline='',
              encoding='utf-8') as csvfile1:
        spamreader1 = csv.reader(csvfile1, delimiter=' ', quotechar='|')
        index = 0
        for row1 in spamreader1:
            if row1[0].split(',')[0] == 'date':
```

```
                continue
            elif row1[0].split(',')[0] == 'year':
                index += 1
                continue
            else:
                pm25 = int(row1[0].split(',')[1])
                pm25List[index].append(pm25)
                pm25DayList.append(row1[0].split(',')[0])
        count = 0
        for i in range(14, 21, 1):
            for j in range(1, 13, 1):
                time = i * 100 + j
                with open(str(time) + '.csv', newline='',
                          encoding='utf-8') as csvfile2:
                    spamreader2 = csv.reader(csvfile2,
                                             delimiter=' ',
                                             quotechar='|')
                    for row2 in spamreader2:
                        if row2[0].split(',')[0] == 'date':
                            continue

                        # 清除空白日期
                        if date_transder(
                                row2[0].split(',')[0]) != pm25DayList[count]:
                            continue

                        # weather
                        w1 = row2[0].split(',')[1].split('/')[0]
                        w2 = row2[0].split(',')[1].split('/')[1]
                        weatherList[i - 14].append(weatherScore[w1])
                        weatherList[i - 14].append(weatherScore[w2])

                        # temperature
                        temp = list(
                            map(
                                int,
                                re.findall("-?[0-9]\d*",
                                           row2[0].split(',')[2])))
                        tempDay = mean(temp)
                        tempList[i - 14].append(tempDay)

                        # wind
                        wind = list(
                            map(
                                int,
                                re.findall("\d+\.?\d*",
                                           row2[0].split(',')[3])))
                        windList[count] = wind[:]

                        count += 1


def display1():
    rc('mathtext', default='regular')
    fig1 = plt.figure()
    for i in range(2):
        for j in range(4):
            if i == 1 and j == 3:
                break
            time = np.arange(len(tempList[i * 4 + j]))
            PM = pm25List[i * 4 + j]
            T = tempList[i * 4 + j]

            ax1 = fig1.add_subplot(2, 4, i * 4 + j + 1)
            lns1 = ax1.plot(time, PM, '-r', label='PM2.5')
            ax2 = ax1.twinx()
            lns2 = ax2.plot(time, T, '-', label='Temperature')
```

```
127
128             # added these three lines
129             lns = lns1 + lns2
130             labs = [l.get_label() for l in lns]
131             ax1.legend(lns, labs, loc=0)
132
133             ax1.grid()
134             ax1.set_title("PM2.5 and Temperature in 20" + str(14 + i * 4 + j))
135             ax1.set_xlabel("Time (day)")
136             ax1.set_ylabel(r"PM2.5 (IAQI)")
137             ax2.set_ylabel(r"Temperature ($^\circ$C)")
138             ax1.set_ylim(0, 400)
139             ax2.set_ylim(-10, 50)
140
141     PM = [i for item in pm25List for i in item]
142     T = [i for item in tempList for i in item]
143
144     fig2 = plt.figure(0)
145     grids = plt.GridSpec(4, 4, wspace=0.5, hspace=0.5)
146
147     # 主图
148     mean_plot = fig2.add_subplot(grids[0:3, 1:])
149     plt.hist2d(PM, T, bins=50, cmap='viridis')
150     plt.colorbar()
151
152     # x轴上的图
153     xhist = fig2.add_subplot(grids[-1, 1:])
154     plt.hist(PM, bins=30, orientation='vertical')
155     xhist.invert_yaxis()
156
157     # y轴上的图
158     yhist = fig2.add_subplot(grids[:-1, 0])
159     plt.hist(T, bins=30, orientation='horizontal')
160     yhist.invert_xaxis()
161
162     mean_plot.set_xlabel(r"PM2.5 (IAQI)")
163     mean_plot.set_ylabel(r"Temperature ($^\circ$C)")
164
165     pccs = np.corrcoef(PM, T)
166     print(pccs)
167
168     plt.show()
169
170
171 def display2():
172     fig1 = plt.figure()
173
174     pm25ForWeather = [[] for i in range(4)]
175
176     PM = [i for item in pm25List for i in item]
177     W = [i for item in weatherList for i in item]
178
179     for index in range(len(W)):
180         if PM[int(index / 4)] <= 150:
181             continue
182         if W[index] == 1:
183             pm25ForWeather[0].append(PM[int(index / 4)])
184         elif W[index] == 2:
185             pm25ForWeather[1].append(PM[int(index / 4)])
186         elif W[index] == 3:
187             pm25ForWeather[2].append(PM[int(index / 4)])
188         elif W[index] == 4:
189             pm25ForWeather[3].append(PM[int(index / 4)])
190         else:
191             print("WTF?")
192     print(len(pm25ForWeather))
193     print(len(pm25ForWeather[0]))
```

```
194
195     for i in range(2):
196         for j in range(2):
197             time = np.arange(len(tempList[i * 2 + j]))
198             pm25PerWeather = pm25ForWeather[i * 2 + j]
199             print('\n')
200             ax = fig1.add_subplot(2, 2, i * 2 + j + 1)
201             # the histogram of the data
202             ax.hist(pm25PerWeather,
203                     bins=40,
204                     facecolor="blue",
205                     edgecolor="black",
206                     alpha=0.7)
207             print(np.mean(pm25PerWeather))
208             print(np.var(pm25PerWeather))
209             print(len(pm25PerWeather))
210             ax.axvline(np.mean(pm25PerWeather),
211                        color='r',
212                        linestyle='dashed',
213                        linewidth=1)
214             ax.set_title("PM2.5(>150) when the weather is grade " +
215                          str(i * 2 + j + 1))
216             ax.set_xlabel(r"PM2.5 (IAQI)")
217             ax.set_xlim(140, 400)
218
219     plt.show()
220
221
222 def display3():
223     print(pd.value_counts([i for item in windList for i in item]))
224     fig1 = plt.figure()
225
226     pm25ForWind = [[] for i in range(6)]
227
228     PM = [i for item in pm25List for i in item]
229     #W = [i for item in windList for i in item]
230
231     for index in range(len(windList)):
232         if PM[index] <= 150:
233             continue
234         for j in windList[index]:
235             if j == 1:
236                 pm25ForWind[0].append(PM[index])
237             elif j == 2:
238                 pm25ForWind[1].append(PM[index])
239             elif j == 3:
240                 pm25ForWind[2].append(PM[index])
241             elif j == 4:
242                 pm25ForWind[3].append(PM[index])
243             elif j == 5:
244                 pm25ForWind[4].append(PM[index])
245             elif j == 6:
246                 pm25ForWind[5].append(PM[index])
247             else:
248                 print("WTF?")
249
250     print(len(pm25ForWind))
251     print(len(pm25ForWind[0]))
252
253     for i in range(2):
254         for j in range(3):
255             time = np.arange(len(tempList[i * 3 + j]))
256             pm25PerWind = pm25ForWind[i * 3 + j]
257             print('\n')
258             ax = fig1.add_subplot(2, 3, i * 3 + j + 1)
259             # the histogram of the data
260             ax.hist(pm25PerWind,
```

```
                    bins=30,
                    facecolor="red",
                    edgecolor="black",
                    alpha=0.7)
            print(np.mean(pm25PerWind))
            print(np.var(pm25PerWind))
            print(len(pm25PerWind))
            ax.axvline(np.mean(pm25PerWind),
                       color='b',
                       linestyle='dashed',
                       linewidth=1)

            ax.set_title("PM2.5(>150) when the wind is grade " +
                         str(i * 3 + j + 1))
            ax.set_xlabel(r"PM2.5 (IAQI)")
            ax.set_xlim(140, 400)

    plt.show()


if __name__ == '__main__':
    getWeather()
    display1()
    display2()
    display3()
```

Listing 2: **dataAnalyse.py**