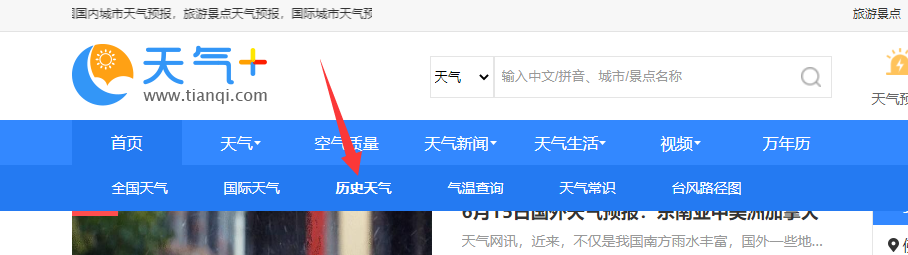
# 1 选题背景介绍

每年的春夏之交时节广东都会汛期。本次报告通过爬取天气网(https://www.tianqi.com/)的历史天气，搜集了广东佛山近10年6月份的天气信息（最高与最低气温、天气状况、风向），并进行数据分析，最后，使用近十年的数据作为输入，运用单变量线性回归和逻辑回归等方式对佛山未来天气的关联分析与预测

# 2 数据获取与处理

## 2.1 获取网页链接

进入天气网https://www.tianqi.com里，打开历史天气信息。



找到佛山的城市历史天气并点击打开。



以2022年6月为例，得到了一个指定的网页地址链接：<http://lishi.tianqi.com/foshan/202206.html>。

通过观察这个链接我们可以发现：链接的前缀是天气网的二级域名，而后面则是城市名与日期的结合。那我们可以推理得到历史天气的链接为:

http://lishi.tianqi.com/ [城市名]/[年份与月].html

如何论证这个猜测呢？我们把时间跳转到到2021年6月和广州2022年6月网页，拿到的链接分别为：

https://lishi.tianqi.com/foshan/202205.html



https://lishi.tianqi.com/guangzhou/202206.html



从上我们可以确定我们的网页链接获取方式是正确的，历史天气网页的链接地址模板为：http://lishi.tianqi.com/ [城市名]/[年份与月].html

## 2.2 分析页面信息

本次报告爬取的目标是佛山自2011年-2022年来的天气信息，在分析网页时，我们以2022年6月的网页界面举例，分析网页结构与获取信息的方法。

### 2.2.1 获取指定的网页信息位置

进入页面以后，我们看到了有效的信息区域在页面的主要区域中。



数据区域中，我们可以清楚的看到我们需要的气温、天气、风向信息的内容，由此我们可以确定爬取的区域在这块区域内。

### 2.2.2 分析网页源代码

我们使用Google浏览器的开发者模式。在页面内，点击键盘上的F12或空白区域内右键菜单栏内点击“检查”。



我们点击网页元素检查按钮或CTRL+SHIFT+C



查找网页表格内的页面元素内容并得到网页链接的源代码



我们发现，网页会用一个名字为“thrui”的网页ui标签将所有信息标签存储进来。这个时候我们在查看下面的信息的一个各个信息表格块的源代码。











通过观察可以看到，我们存储的每个信息都存储在指定的标签内：

日期：th200标签

气温、天气、风向：th140标签

同时，它的每个信息都很有序的排序在li表格内。

由此，我们可以确定我们的爬取思路

1. 进入网页界面内，获取网页的源代码；
2. 分析网页源代码，获取存储着天气信息的“thrui”标签的内容；
3. 通过分析“thrui”标签内的li表格信息，并从中提取指定信息标签；
4. 从信息标签内提取“th200”（日期）、“th140”（气温、天气、风向）的标签文本信息并存储指定信息。

### 2.2.3 编写Python爬虫代码

代码已作为附件且写好完整注释存储在压缩包内，本处以伪代码方式介绍代码编写思路。

1. 程序由主函数进入，程序会要求用户输入需要爬取的城市名称，可以是中文名也可是拼音。程序再让用户输入开始爬取的开始和结束的年份、月份。因为年份和月份可能会存在一定的跨度，所以采用数字序列循环生成方式，将年份和月份生成在一个一维序列内，并将这个序列送入爬虫体内。例如：需要爬取2011-2022的年份，循环会生成2011-2022的所有数字存储到数组内，月份同理。
2. 程序在主函数由用户输入完整城市名、年份、月份信息后，将这些信息都送入爬虫函数Spider内，函数会首先处理送入进来的爬取信息，将他们送入链接生成方法内，生成 http://lishi.tianqi.com/ [城市名]/[年份与月].html的链接以列表的方式返回并存储在url列表内。此时，爬虫函数spider开始以返回的链接列表开始进入内容循环，将每个链接提取出来并通过模拟请求获取网页源代码返回至爬虫函数内。
3. 拿到网页源代码后，使用BeautifulSoup模块构建网页源代码的代码树，通过提取源代码里的“thrui”标签以提取的天气信息的代码块。再构建两个正则匹配器来匹配标签“th200”和“th140”用于匹配日期和天气信息。
4. 首先通过正则匹配器提取出该代码块里的所有时间，因为时间和天气数量是对等的。再接着，通过提取出来的时间信息的列表来划定一个滑窗，用于提取指定框体内的天气信息，这个天气信息是对应这个天数的天气信息。
5. 提取完链接列表url里的所有信息以后，使用pandas模块的dataFrame方法将所有数据构造成一个pandas数据框架，并将其保存为csv文档。至此得到一个包含[日期、最高气温、最低气温、天气、风向]的数据集。

附爬虫体代码截图：



# 3 数据分析与挖掘

## 3.1数据预处理与清洗

因为在后续的分析过程当中，需要用到天气信息中是否下雨的因素，所以在数据清洗过程需要提取出有下雨的天气信息。

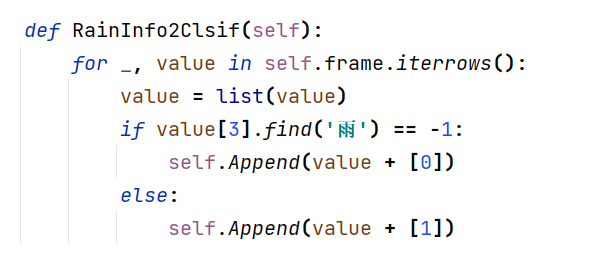
设计思路：

（下雨的标记成1，不下雨为0，结果存储在result 列中）

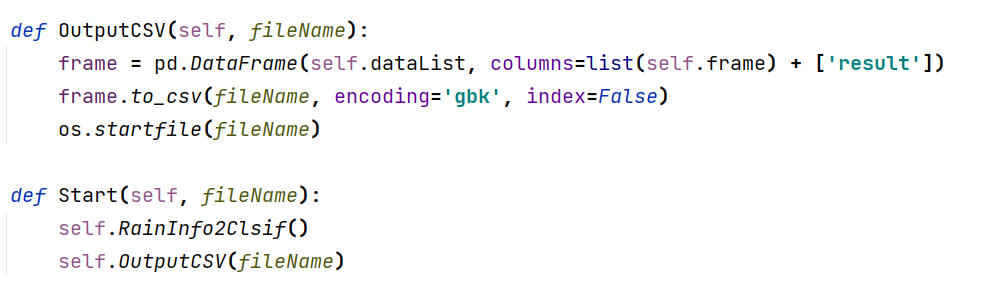
1. 筛选空数据与数据预处理



1. 因为信息中必定带雨，所以我们在设计时只需要遍历“天气信息”列的内容，在遍历的同时判断是否存在“雨”这一关键词，存在则存储标记；



1. 将存储标记的信息编辑成新的“result”列与导出数据



## 3.2逻辑回归分析

在本例中，通过逻辑回归分析与绘制可视化图表的方式。将分析最低气温与日期是否与最高气温有相关。

因为代码长度过长，本处以伪代码的方式进行代码设计讲解：

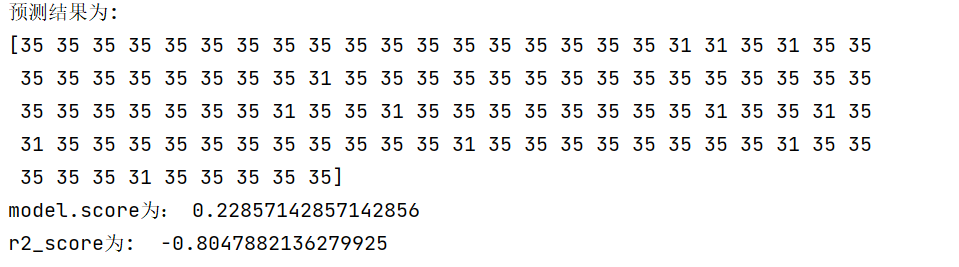
1. 导入相应数据分析库与机器学习库，本处引入了Python的 数据分析库pandas、matplotlib.pyplot、数值计算库numpy和机器学习库sklearn
2. 从数据集中读取数据读入pandas-dataframe框架中，接着对数据进行预处理，包括对温度内的“°C”内容进行提出，并将数值更新为整型；接着再对日期进行处理，将2011年6月1日到2022年6月17日的日期全部转为1-347的序列排序并导入在numpy数组内。
3. 引入机器学习库sklearn准备开始训练，使用lbfgs算法利用损失函数二阶导数矩阵即海森矩阵来迭代优化损失函数并进行模型训练。
4. 训练完成后，将数据值导出并对模型进行评估和绘制可视化图表，根据计算结果和图表判断其是否相关。

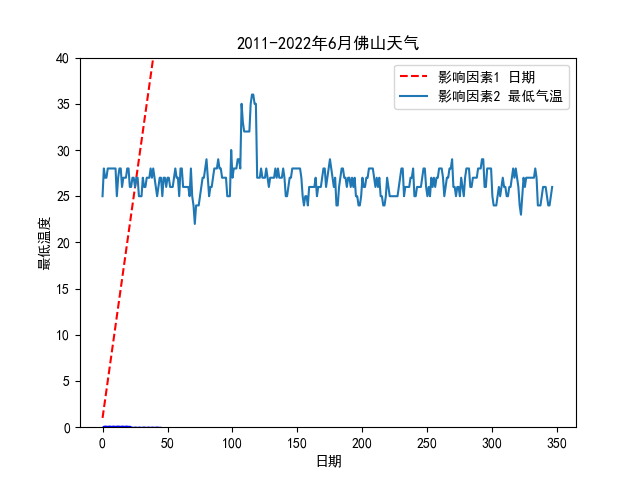
附训练模型处完整代码：





得到绘制完成的数据结构和可视化图表





## 3.3 单线性回归分析

在本例中，将通过单线性回归分析对未来7天的最高气温进行预测。

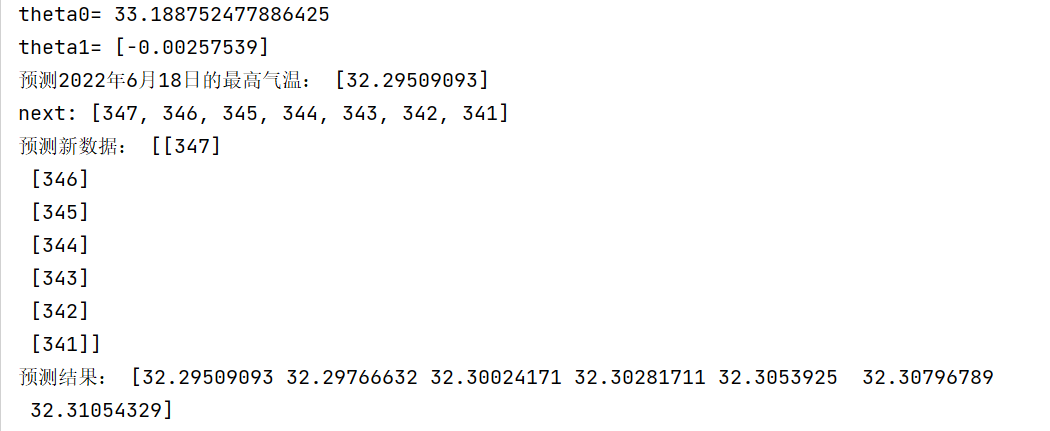
因为代码长度过长，本处以伪代码的方式进行代码设计讲解：

1. 导入相应数据分析库与机器学习库，本处引入了Python的 数据分析库pandas、matplotlib.pyplot、数值计算库numpy和机器学习库sklearn
2. 从数据集中读取数据读入pandas-dataframe框架中，接着对数据进行预处理，包括对温度内的“°C”内容进行提出，并将数值更新为整型；接着再对日期进行处理，将2011年6月1日到2022年6月17日的日期全部转为1-347的序列排序并导入在numpy数组内。
3. 引入机器学习库sklearn准备开始训练，并将数据拟合直线与预测概率。
4. 训练完成后，将数据值导出并对模型内容进行输出，获取未来七天内的区间值概率的数值即为最终结果。

附完整训练模型代码：



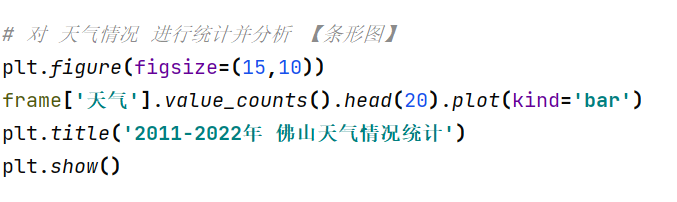
输出结果：

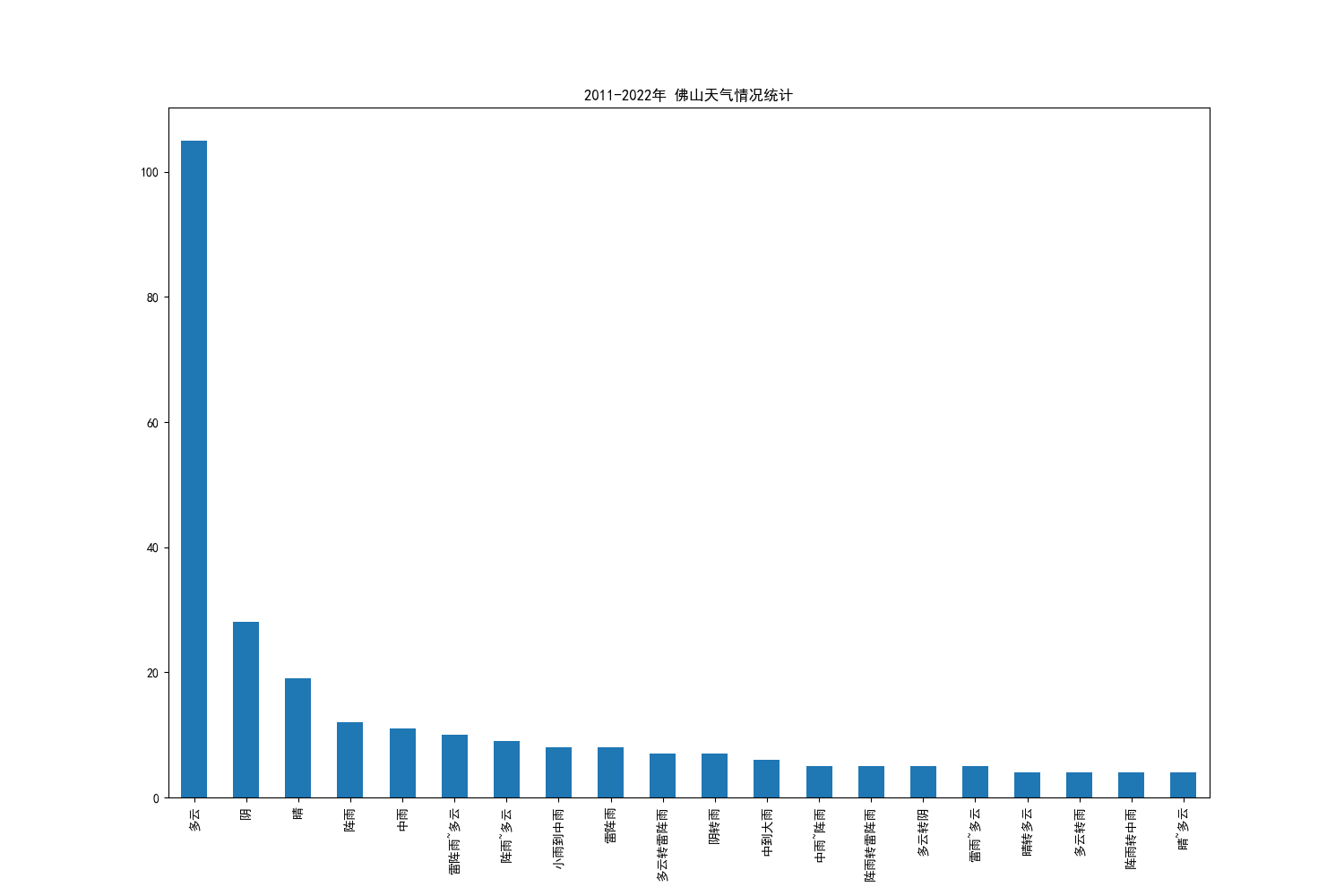


# 4 数据可视化

通过python - matplotlib.pyplot 模块，对数据进行统计，内容包括对2011-2022佛山天气的：天气状况、风向、最高气温、最高气温热力分布进行可视化分析；

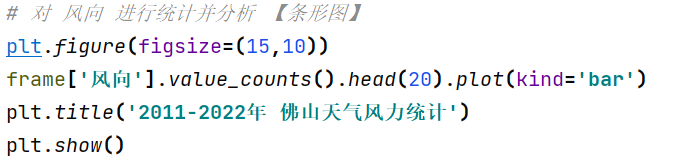
## 4.1 对佛山天气状况的统计

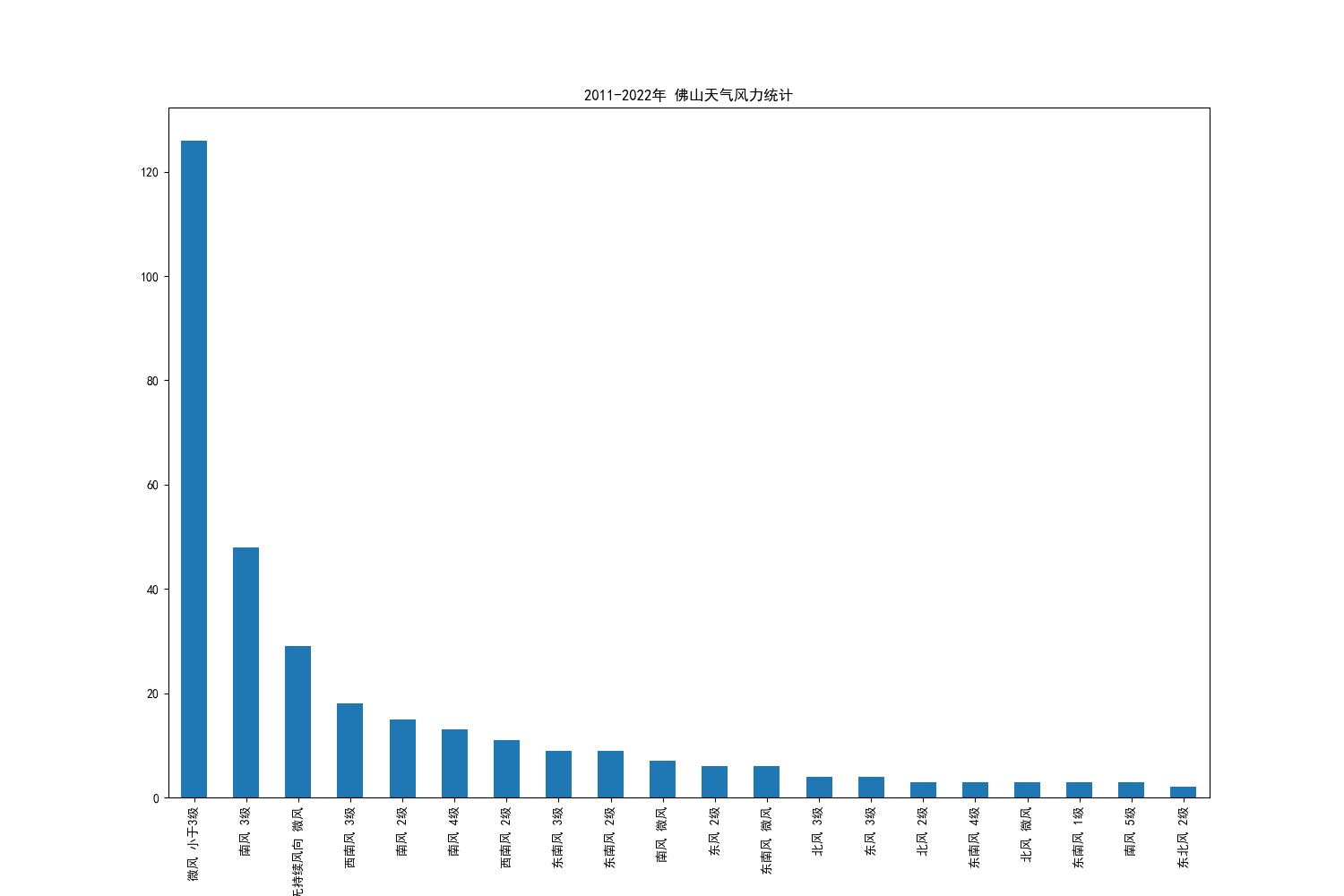




可以观察到的是，佛山6月份的天气普遍为“多云”，而且全天晴天天气情况较少（虽然晴天排序第三，但是综合其他与晴天相对的天气而言，晴天的可能性会比非晴天的天气状况要少）；

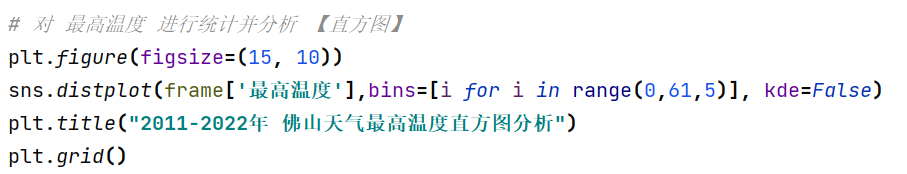
## 4.2 对佛山风向状况的分析

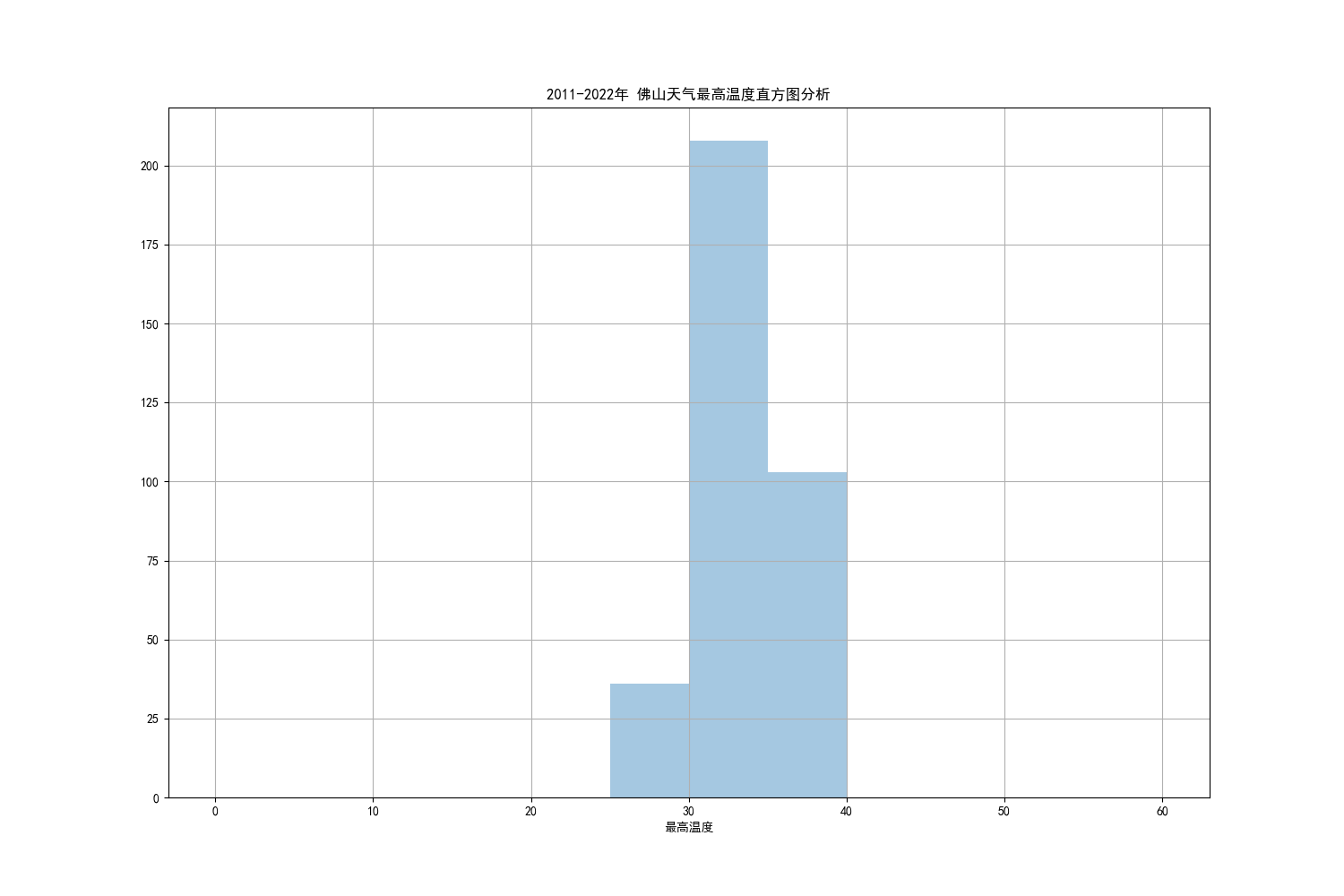




可以观察到的是，佛山的风力多为3级风，且多为南风。

## 4.3 佛山最高气温直方图分析

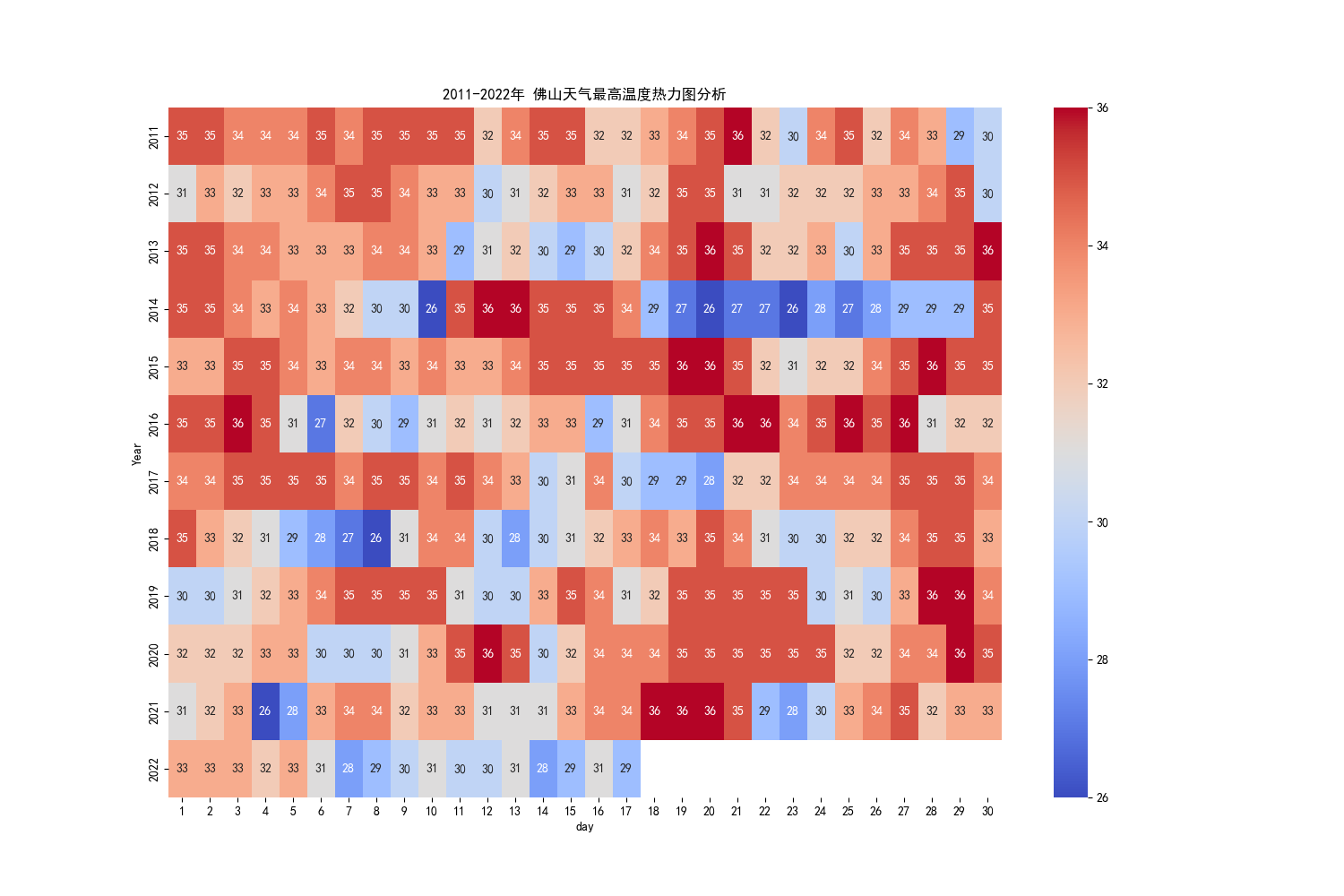




可以观察到的是，佛山6月份的气温大多数维持在30°C左右，不超过40°C。

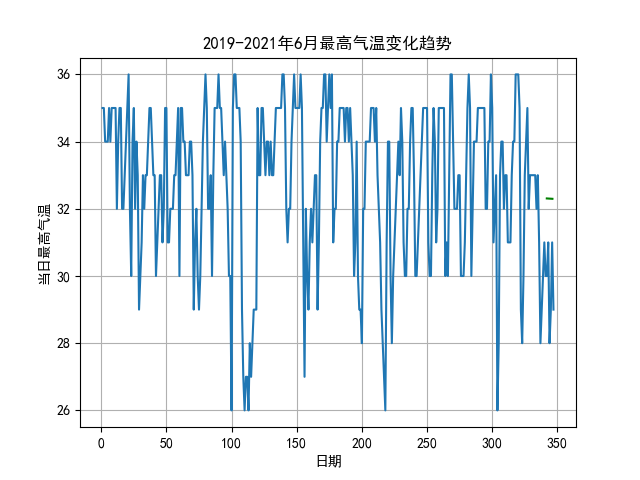
## 4.4 佛山最高气温均值热力图





由热力图我们可以直观发现到：佛山6月份最高气温为36°，以30°以上天气为主，部分时期最高温度最低为26°，在2021、2018、2014年出现过6月份的“低温”情况。

## 4.5 佛山最高气温变化趋势折线图



我们将2011-2022的6月分转化为1-347天的序列长度，并对其变化情况做了这些那图，由图可见，6月份气温集中在30°之内

# 5 结论与对策

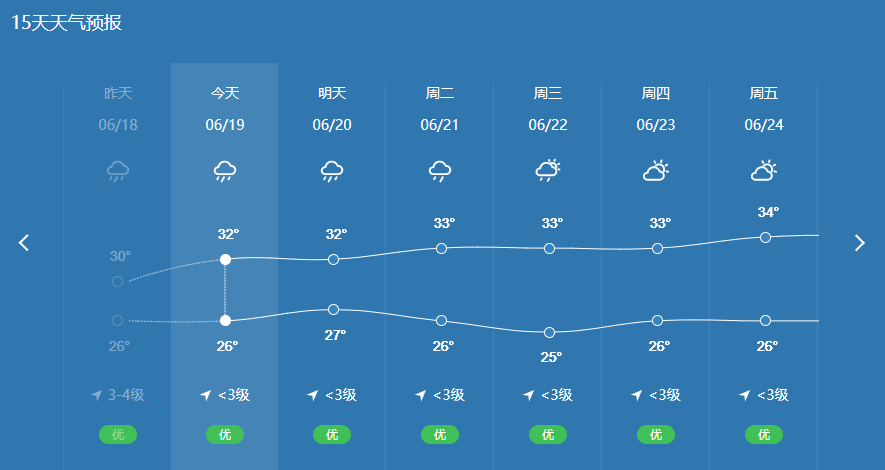
## 5.1 逻辑回归分析

结论：

1. 综上可以观察到，所有数据点并不明显分成两个类别。说明最低温度和日期并不能对最高温度进行分类
2. 线性回归主要都是针对训练数据和计算结果均为数值的情形。
3. 而在本例中，结果不是数值而是某种分类：这里分成日期和最低气温两类。
4. 而且发现，两类并不显示有明显的分界线。这进一步说明最高温度的影响因素不是日期和最低气温。
5. 我们通过数据爬取预测模型最终得出结论：最高温度的影响因素与日期和最低气温毫无关联

## 5.2 单线性回归分析

现实天气报告：



结论：

根据现实中的天气报告结果而看，预测准确性是符合预期的，误差不超过3°的天气。

## 爬虫结论与方法：

1. 首先确定爬取网址与目标数据；
2. 通过使用网站的开发者工具对于网站源码内容进行分析，提取关键标签与元素的信息，并根据网页代码的呈现，判断爬取的方式与思路。
3. 使用python编码的方式，模拟人类的请求获取网页源代码，并使用爬虫代码框架将网页源码形成相应的语法树。再使用正则表达式进行数据提取。
4. 完成内容提取后，将内容保存。