BeiTown教你科学过冬

长沙已经连续2个月没怎么正式见过太阳了。





刚好最近又一直在用python做数据分析方向的研究，本着科(chi)学(bao)过(cheng)冬(de)的态度，决定还是要分析一下长沙历年的气象情况。

本文简单记录了整个工作流程，以此方便其他地区的朋友参照本范例科学分析。



【数据篇】

首先当然是去找数据源，历史气象数据不是什么机密数据，在国家气象信息中心就可以找到：

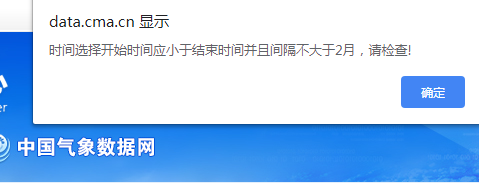


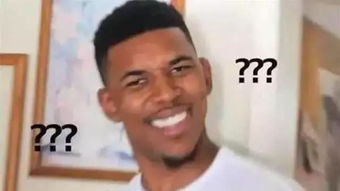
中国地面国际交换站气候资料日值数据集(V3.0)

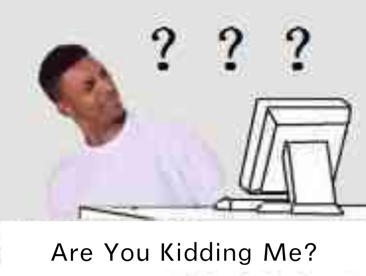
数据还是比较全的，从建国后开始至今接近70年的数据。

看来我国的科学数据信息还是比较开放的。

但是，当你要批量下载数据的时候…







也就是说，一次检索不能大于2个月，那么下载70年的数据需要设置这个页面420次，再加上生成和下载，手速快不吃不喝连续作业也大概需要花掉20个小时的时间，所以。。。

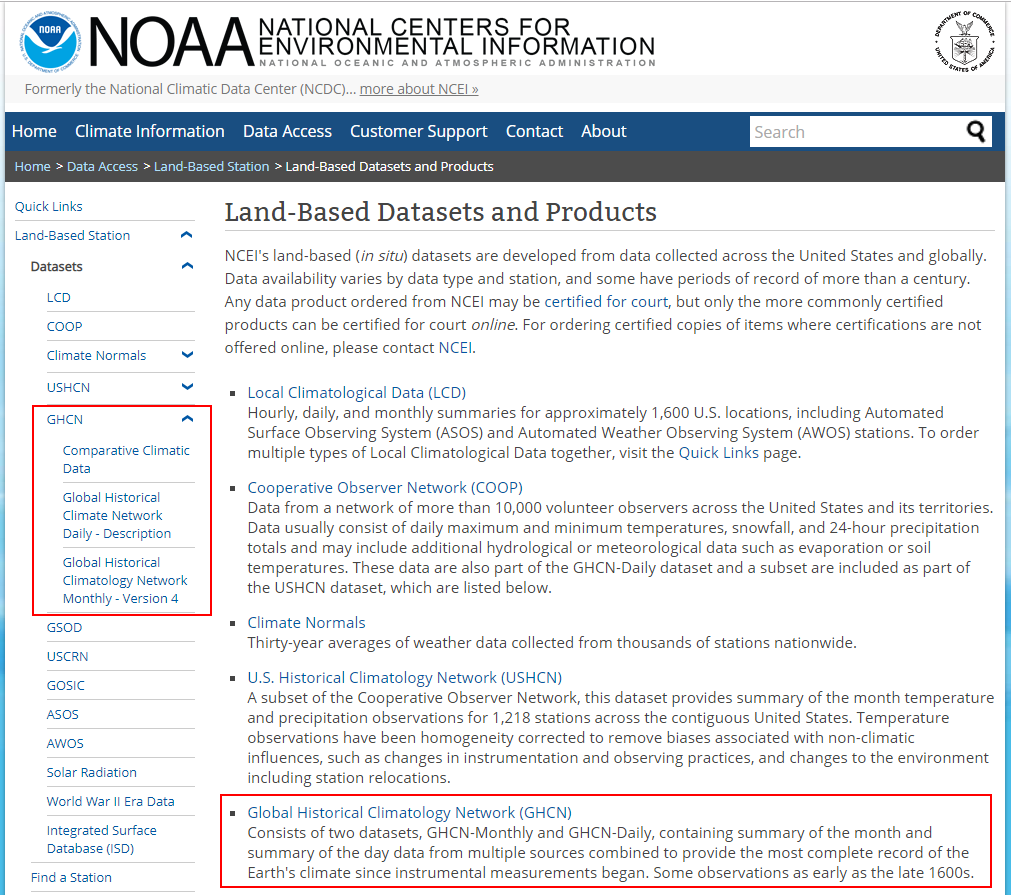


理论上我们可以用python写一个爬虫把数据爬下来，但是不到最后需要自己造轮子的时候，尽量还是寻找更简单的方法。

地面国际交换站之所以叫国际交换站就是因为这个数据是和其他国家共享的，我国下载受限的话，理论上在数据更公开的其他国家（此处手动@USA）可以尝试搜索一下。

NOAA（美国国家海洋和大气管理局）应该有我们需要的数据，首先官网进去找到数据中心的地址，为了节约大家的搜索时间这里直接贴出数据库的入口地址

（<https://www.ncdc.noaa.gov/data-access/land-based-station-data/land-based-datasets>）



我们要找的数据在GHCN（全球历史气象学网络），看描述感觉很硬核，记录了自测量器诞生以来地球的所有历史气象数据，最早可追溯到了17世纪。



我们当然不需要这么多，能获取到目标地区气象站的几十年完整数据即可。

之后通过给出的链接进入FTP，这里可以下载全部的数据，没有任何地区和权限限制，只要连接了网络就可以下载。

不得不说一个国家的科技进步水平和其科技资源的开放程度是成正比的。

点击进入FTP，然后：

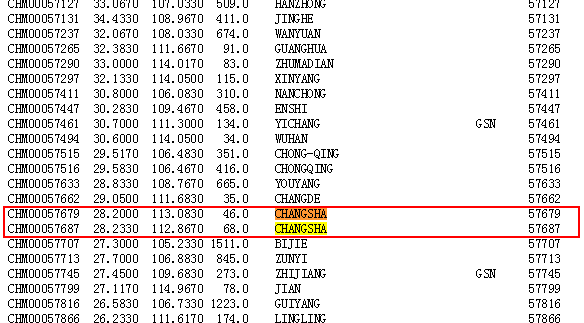


GHCN-FTP根目录



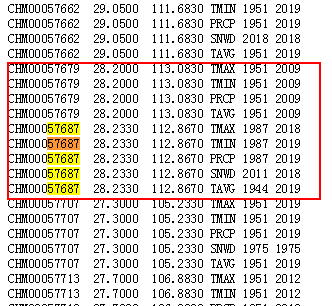
通常进一个FTP数据库，第一时间有点不知道从何入手是很正常的，基本上我会把root下的readme.txt看一遍，一般心里就大概有数了。

ghcnd-stations.txt记录了所有气象站的编码信息，在之后的查询中我们需要用编码来检索数据。



ghcnd-stations.txt

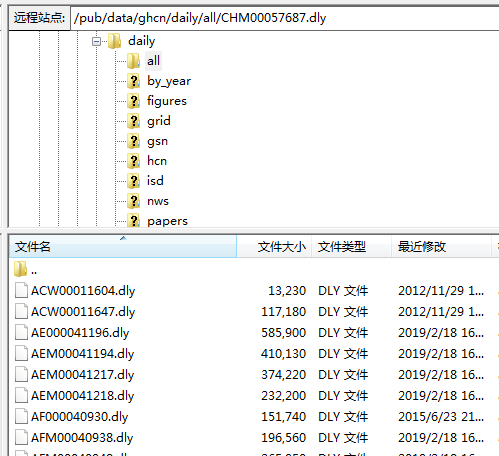
搜索目标地点CHANGSHA（长沙），看到有两个气象站，编号57687的那个之前在中国国家气象信息中心查到过，通过经纬度和地图可以发现它架设在岳麓区。编号57687这个气象站从1970年1月开始监测，但是不存在编号57679的气象站，我们先不管它，可以把这两个站点的数据都下载之后再看。



ghcnd-inventory.txt

ghcnd-inventory.txt记录了数据详细清单，通过搜索之前查询到的长沙气象站的编码，可以看到气象站57687的数据清单， 平均气温（TAVG）记录从1944年就开始了，降雨量（PRCP）从1987年开始。而气象站57679的数据从1951年开始就非常统一和完整，只是在十年前停止了记录，可能是关闭了。所以之后把这两个气象站的数据合并一下，基本就可以完整反映出长沙地区1951-2019的气象情况。

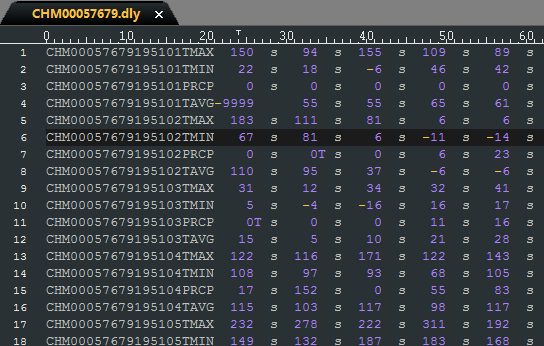
我们需要的数据本体在all目录下：



all目录

分别下载CHM00057679.dly和CHM00057687.dly这两个气象站的全部数据。

最后我们用UE打开文件，可以看到，记录从1951年1月开始，其中包含了最高气温，最低气温，平均气温（0.1摄氏度）及降雨量（0.1mm）。这里的s是列分隔符，每一列表示当月中某一天的数据值。



CHM00057679.dly

其实还是想再多拿到一些数据，比如相对湿度，云覆盖率，气压等。不过有了气温和降水数据基本也可以统计出长沙历史的天气概况了。如果有朋友能够拿到更丰富的气象数据，也欢迎提供一下。

关于数据获取的内容就先说到这里，肯定还有能获取更全面数据的方法，以后有新的发现会另外再开篇拓展。



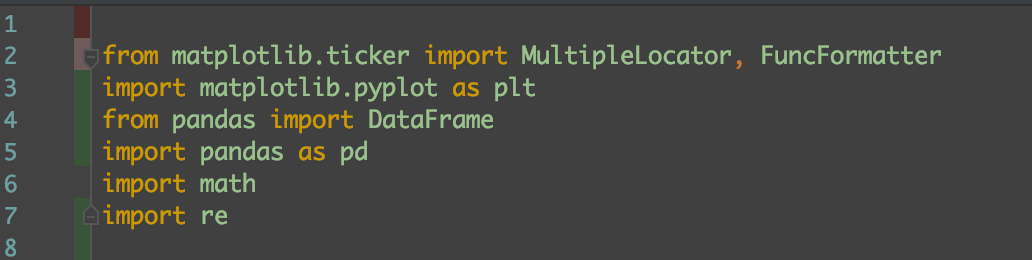
【编码篇】

拿到数据以后，就可以开始用python编写代码开始分析了：



对代码不感兴趣的同学可以直接跳过本节直接看分析结果。

首先这个demo用到的库如下



Pandas用来做数据整理，Matplotlib用来绘图，Math是数学基本库，以及Re正则库，用来对原始格式的数据进行分割筛选。

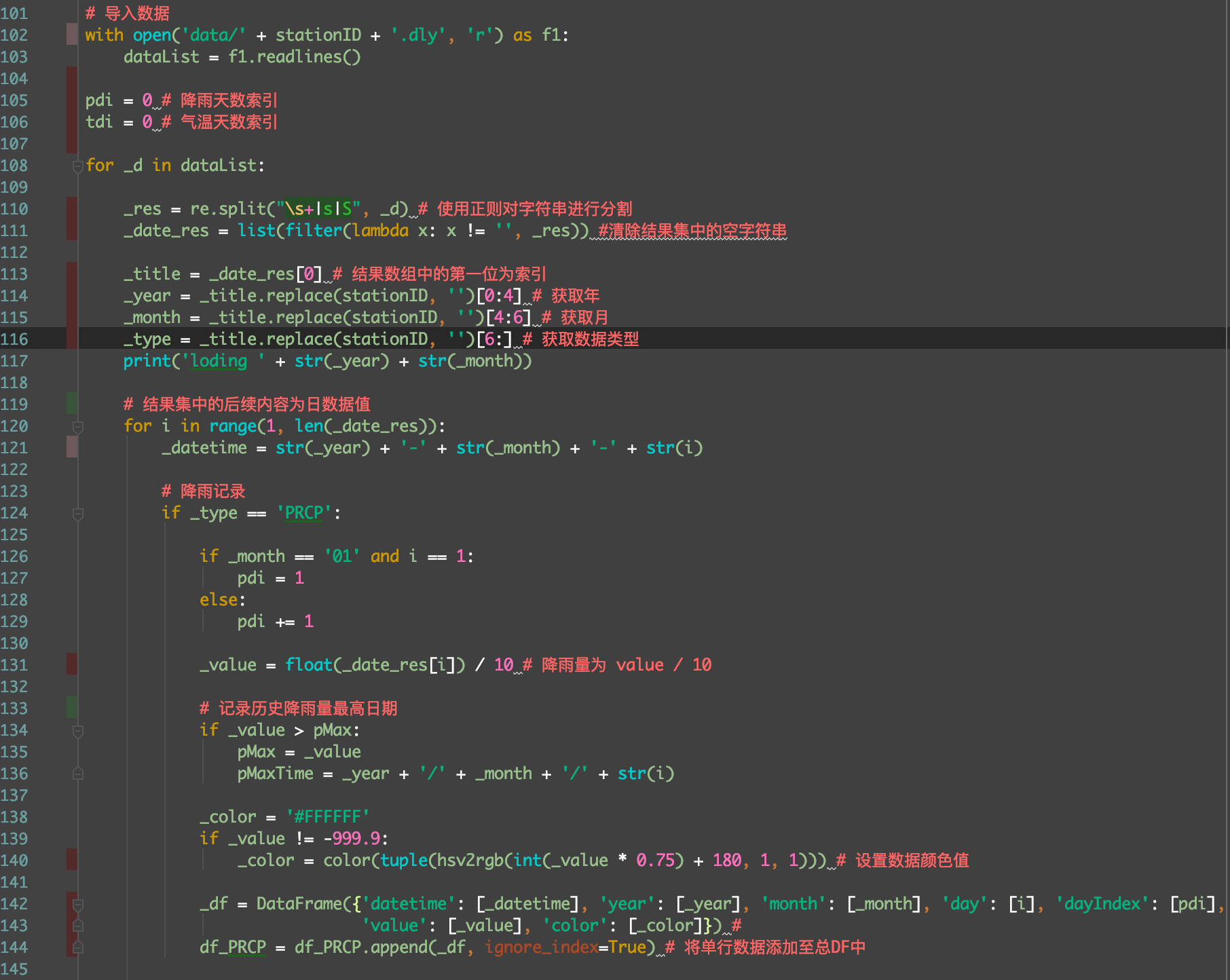
Step1:数据读取

从老美那拿到的数据结构虽然已经比较清晰了，但还是不能拿来直接用，我们需要把逐月数据转换成逐天数据，然后保存到DataFrame里。

首先定义四个空的DataFrame，注释已标。



之后对dly文件进行逐行（逐月）读取，将单行进行字符串的分割，取出时间以及每个单日数据。



这里有两个循环，一个循环是dly文本文件的逐行读取，另一个循环是每一行中每一天的列数值读取和创建一行日数据df。获取并组合成新的日数据df之后，就可以添加进之前创建的指定类型的空DataFrame中。这里只截图了PRCP（降雨统计）的部分，其他部分同理。

关于数据显示时的颜色值，这里用了HSV颜色模型的思路，通过改变H（色相）值，来实现通过颜色表示数据大小的效果。实际也可以不需要，只是为了显示时更好看一些而已。这一块不是重点，可以略过。

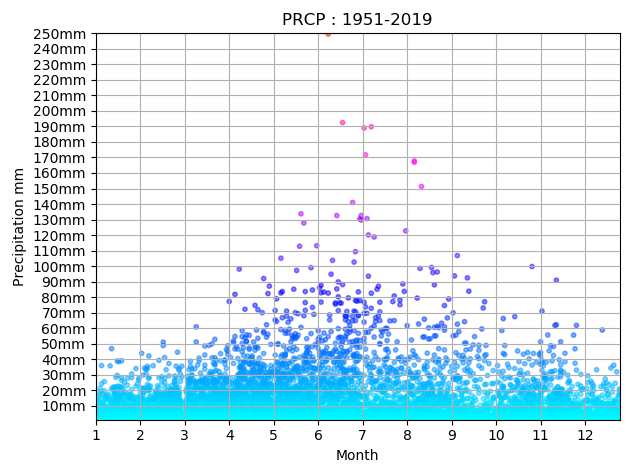
Step2:散点图绘制

依旧以PRCP为例：



绘制图表可以调用Maltplotlib.pyplot库下的各个方法。其中绘制散点图的方法为scatter()，参数中x轴填入日索引（当年的第几天）集合，y轴填入当日降雨量集合，s表示点的大小，这里设置为40，c表示颜色，可以使用固定字符串或者之前设置匹配数值的颜色集合。marker表示点的形状，alpha表示点的透明度。

运行之后的效果是这样的：



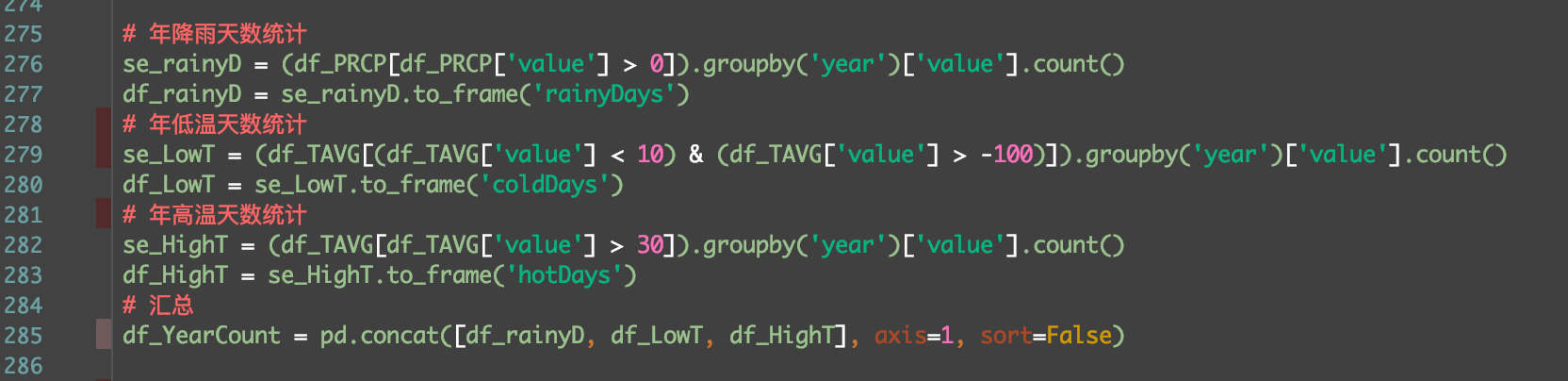
每一个点表示70年中某一天的值，通过这张图就可以大概看出历年降雨量和日期的关系，以及分布情况。

平均气温TAVG也有一张散点图，用来展示历年气温分布和日期的关系，和PRCP的显示方法相同，因此这里就不再重复说明。

Step3:线性图绘制

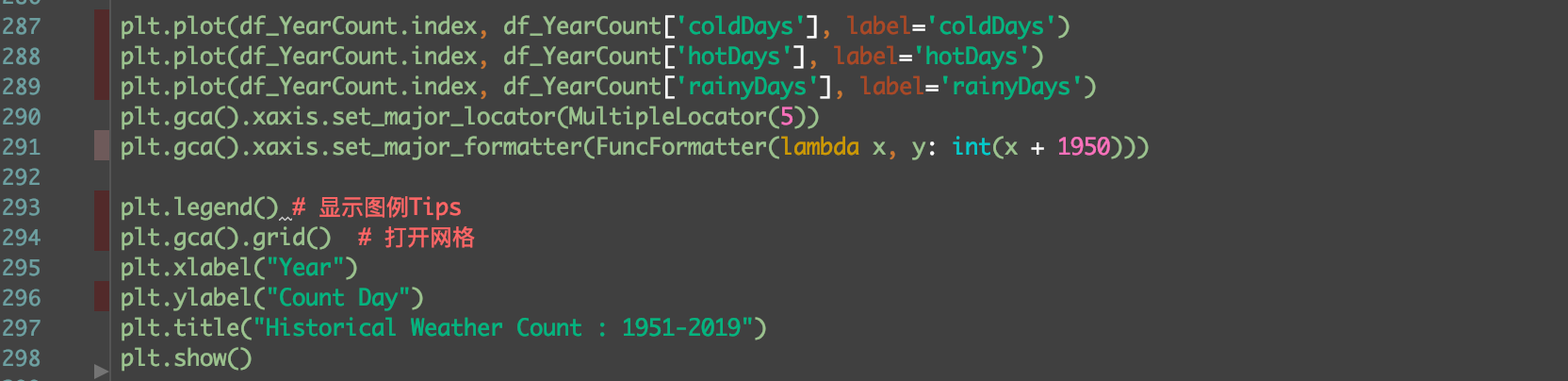
在本步骤中我们将用线性图来展示数据在逐年中的变化情况。需要统计的数据有：单个年份中有降雨的天数，炎热（TAVG > 30°C）天数以及寒冷（TAVG < 10°C）

统计数据的代码如下：

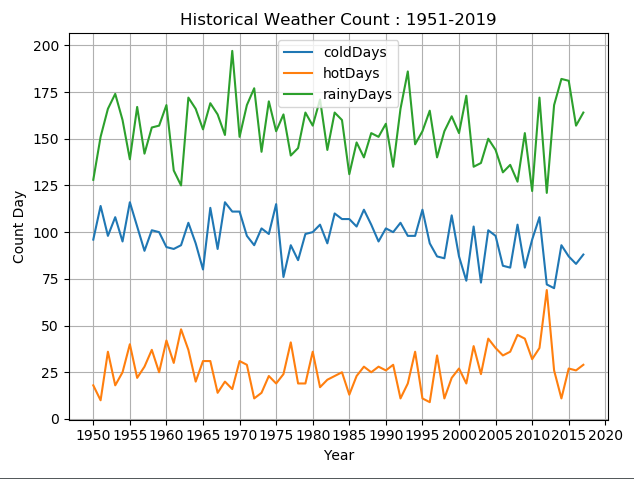


通过DataFrame的groupby()方法来将逐日数据按照年份进行分组，之后再使用count()方法对分组进行计数统计，产生的数据集即为我们需要的年统计数据。

有了数据之后，绘制线性图的方法也很简单：



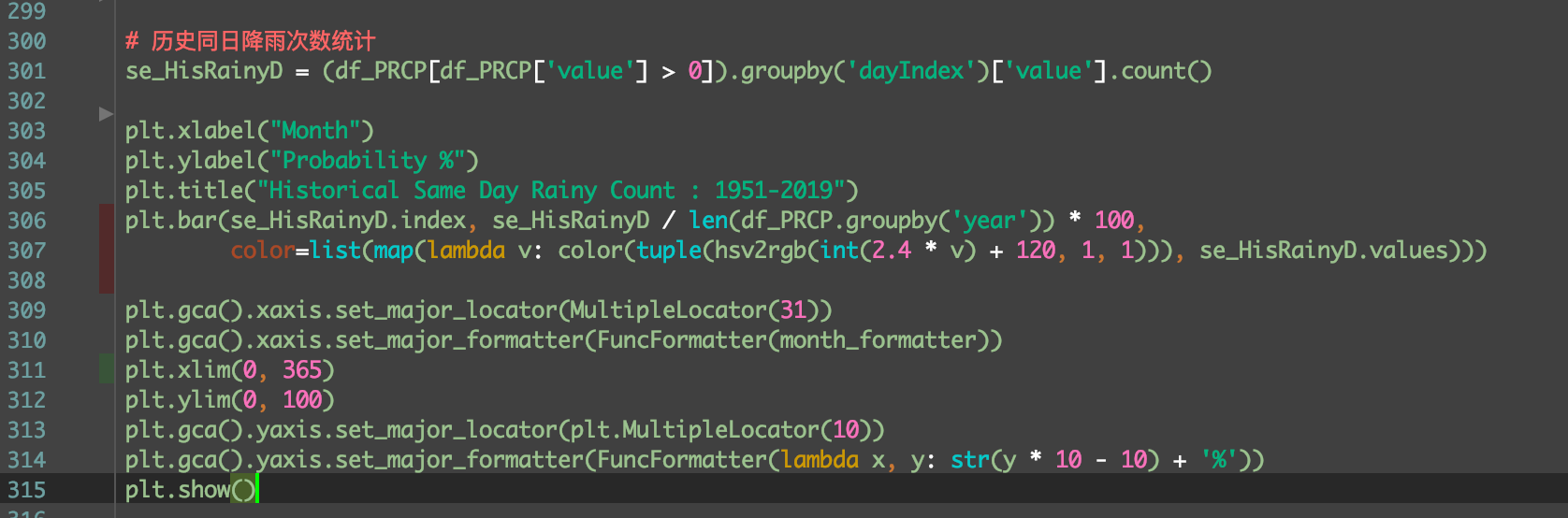
在同一张画布上调用三次plot()方法，分别绘制年降雨天数、年高温天数、年低温天数这三个数据集的逐年线性变化曲线：



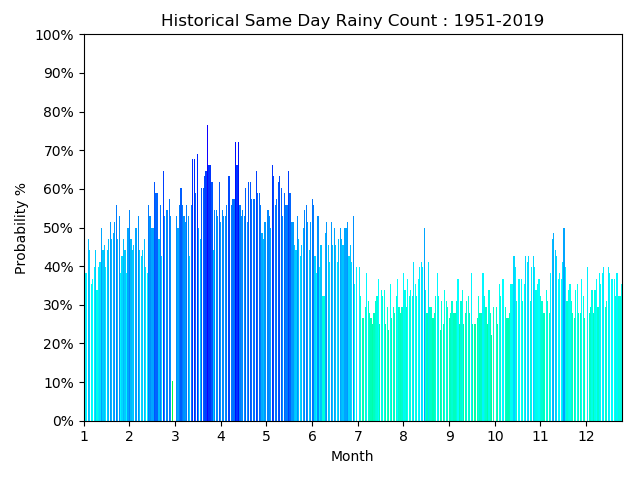
Step4:柱状图绘制

之后我们还需要再统计两类直观数据，即历史上同一天的事件发生频率（如某一天的历史降雨频率）以及所有年份中事件连续发生的最值和时间段（如每一年连续降雨最长天数以及发生的日期）。

首先是事件发生频率图，以降雨为例：



运行结果如下：



事件发生频率还可用于统计历史年份某一天严寒或者酷暑的概率，数据集的筛选方式不同但原理相同。

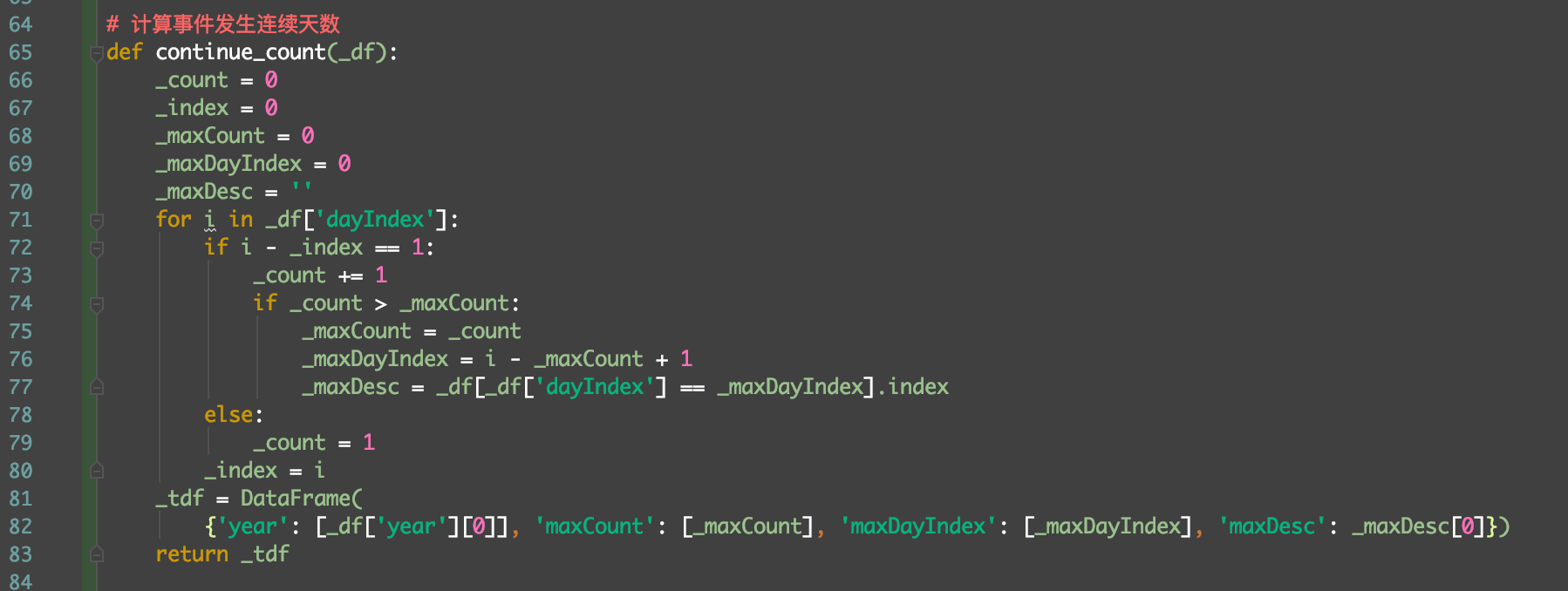
接下来就是我们最后的一类数据分析了，即事件连续发生天数逐年最值以及发生日期。这里仍旧分析降雨数据。我们需要计算出每一年连续下雨最长的天数以及从什么时候开始到什么时候结束。

代码如下：



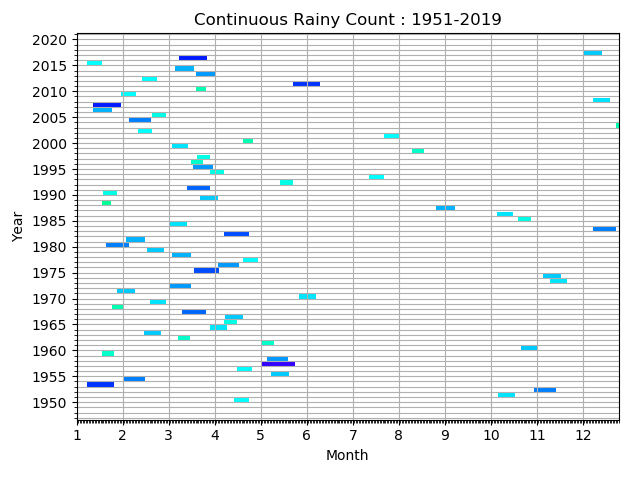
注意到这里用的是plt.barh()方法，即横向柱状图绘制，同时添加了left=？标签，用于对柱状图块进行位移（位移用来表示从哪一天开始，不设置的话默认都是从x=0开始）。

另外自定义了一个计算函数，稍微复杂一些所以不适合用lambda表达式来直接写：



基本思路就是循环结果集中的所有数据，遇到连续索引的日期就进行累加，当累加超过之前保存过的最大值时就覆盖掉之前的记录，所有结果循环完毕之后，最后留下的记录就是最大值记录。

最后的效果如图：



每一个柱状图块的长度表示了每一年的降雨最大天数，图块的位置表示了连续降雨的起始和结束时间。

编码写到这里，最核心的几个步骤就结束了，剩下的就是一些同类型的拓展统计需求，使用上述的数据集和方法基本上都能够实现，这里就不再拓展编码说明了。

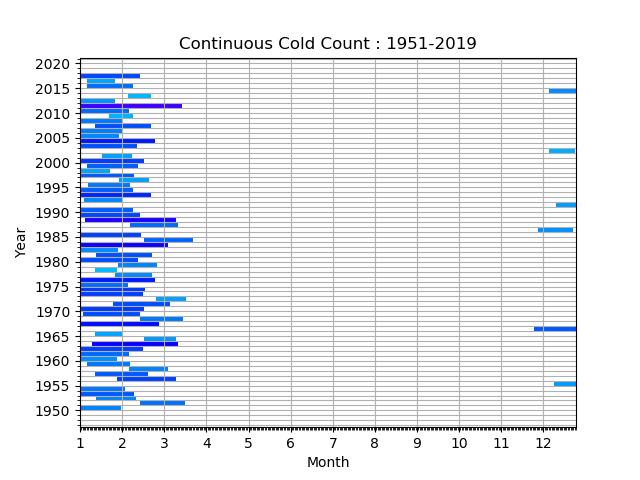
最后，我们还需要再做一件事情，那就是——图表分析。



【分析篇】

首先要感谢能够看到这里的同学，也其实大家比较感兴趣的可能也就是这里了。

直接上图先：



不说了，我再去看一眼上个月晒的衣服干了没有。

