  **matplotlib可视化库入门** 

—————

06/17 - Mindlake - 创建

06/20 - Mindlake - 第一次修改

**简介**

Matplotlib是一个Python的可视化库，主要用来绘制2D图表，目前的版本是2.2.2。按照官网上的定义来说，“Matplotlib让简单的事情简单做，困难的事情能够做”（Matplotlib tries to make easy things easy and hard things possible. ）

那么在这里我们就一起看一看Matplotlib到底能做哪些简单的事情。



**第0件事：准备**

[ 在我们让matplotlib真正提供可视化服务前，我们需要了解和准备的事情 ]

**- 需要了解的事情：**这篇入门教程，主要是为一些刚刚了解python，但又不甘于只简单绘制项目中提供的箱线图和柱状图，还想尝试更多可视化图形的同学准备的，或者是那些想要了解不同参数带来了可视化图形绘制差异的同学准备的。我会Step by Step的教大家怎么去写代码和调整代码。

**- 注意：**本文的所有代码需要在UDA课程中的项目工作区，或者Jupyter Notebook中运行。

**- 代码1：**import matplotlib.pyplot as plt

**- 代码说明：**这是首先且必备的代码，用来告诉python引入matplotlib包。

**- 代码2：**%matplotlib inline

**- 代码说明：**这是Jupyter Notebook的专用代码，用来告诉Jupyter Notebook把可视化图形绘制到工作区内。（没有这行代码，所有的输出都只是一串字符，不会有图像。）

**- 数据来源：**所有数据均来源于我们的项目《5个城市PM2.5数据分析》，在使用下面的代码前，我假定了你已经知道如何获得**df\_all\_cities**这个变量的值。

**第一件事：直方图**

[ 直方图用来表示不同分组的频率分布情况 ]

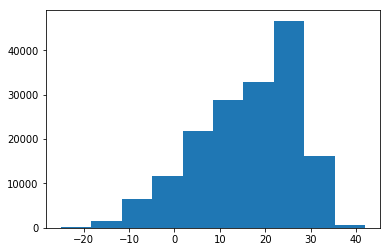
**- 函数：**plt.hist(x,bins=None)

**- 函数说明：**x表示一个数组，一般是整形数组或者浮点数数组。bins代表分组标准，默认自动分组，但也可以指定分组。

**- 代码1：**array\_test=np.array(df\_all\_cities['TEMP'])

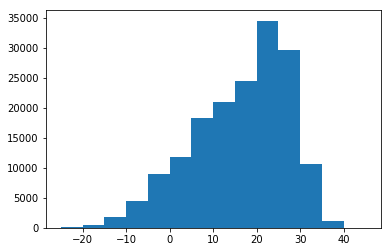
**- 代码说明：**我们选择**df\_all\_cities**中的**‘TEMP’**数据列进行绘图，但首先需要用上面的代码将其转变成绘图可用的数据类型，并储存在**array\_test**变量中。

**- 代码2：**plt.hist(array\_test)



绘图如上，可以看出hist()函数，自动对函数进行了分组，但是这个分组并不是按照整数（按照我们习惯的分类方式）进行的。我们想让它按照（-25，-20）、（-20，-15）这样，每5个单位分一组。如果你想这样，那么就必须指定bins的值。

**- 代码3：**plt.hist(array\_test，[-25,-20,-15,-10,-5,0,5,10,15,20,25,30,35,40,45])



这下整齐好看了，但是这一长串的bins值，太长，而且非常容易计算错误。怎么办，我们使用range(-25,50,5)来生成这样一组数据。-25表示计算起点，50表示计算终点（但不包括50），5表示从计算起点到计算终点每次增加的值。于是我们刚才复杂的代码就简化成下面这个：

**- 代码4：**plt.hist(array\_test，range(-25,50,5))

此时的绘图效果和上面代码3的一样。

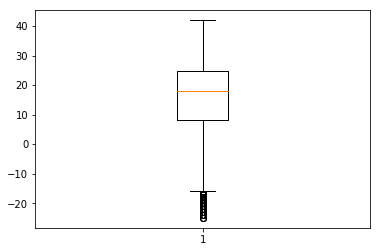
**- 上述直方图的解释：**从上面的图可以看出，20摄氏度到25摄氏度相对于其他的分组来说，占的比例最高。超过30摄氏度的分组，占的比例比较小，说明从整体上来说。高温天气比较少。

**第二件事：箱线图**

[ 箱线图用来，展示数据中值的分布情况，可以用箱线图来展示数据的最大值、最小值、四分位值（Q1，Q3）和中位数 ]

我们继续使用上面生成的array\_test变量进行演示。

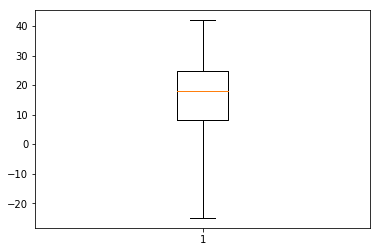
**-代码1：**plt.boxplot(array\_test)



需要解读下上面的图形。最中间的那个矩形（中间有一条橘色线的）就是箱线图中的“箱”。箱子的下边线，指Q1（数据中的25%的值都小于这个线所代表的值--Y轴），箱子的上边线，指Q3（数据中的75%的值都小于这个线所代表的值）。中间的橘色线是中位数。箱子最上方的横线和最下方的横线，代表数据集的最大值和最小值。

下面的黑色线是什么？因为函数会默认将大于某个值的数据定为异常值，并在图中呈现出点的状态，所以你就看了一条黑线。如果你不想这样，想把所有的点都纳入箱线图内，不自动设定异常值，那么可以使用下面的函数参数：

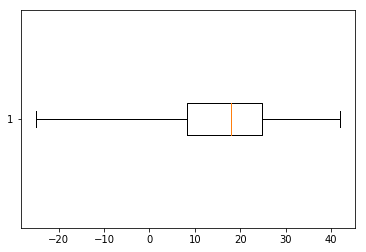
**-代码2：**plt.boxplot(array\_test，whis='range')



可以看到下方的黑线消失了，同时表示最小值的那条线也发生了变化，向底部进行了移动。

如果你不喜欢竖着的箱线图，喜欢横着的箱线图。可以继续增加参数：

**-代码3：**plt.boxplot(array\_test,vert=False,whis='range')



喏~这应该就是你喜欢的样子。

**第三件事：柱状图**

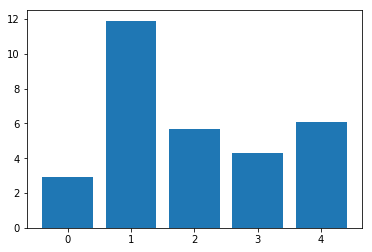
[ 柱状图用来，展示数据之间的差异性 ，一般适用于少量数据的情形]

**-函数：**plt.bar(range(N),data)

**-函数说明：**函数中的两个参数，range(N)是指你要绘制多少个柱。一般来说，多少个数据就应当绘制多少个柱，假如你有6个数据，要绘制6个柱，那么参数就应该是 - range(6)。第2个参数data就是包含数据值的变量或者直接可以输入数据。这个数据值的个数应该和刚刚rang(N)中的‘N’一致，如果不一致，将会报错。

我们看看下面这个代码将要绘制的图。

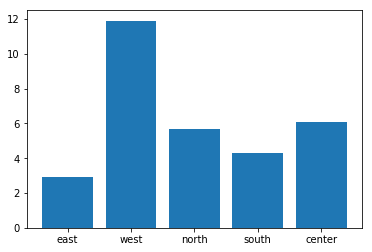
**-代码1：**plt.bar(range(5),[2.9,11.9,5.7,4.3,6.1])



可以看出来，很简单就绘制了柱状图，但是如果你打算把每个柱下面的“0,1,2,3,4”的名称更换成指定的列名，继续增加一组参数。

**-代码2：**plt.bar(range(5),[2.9,11.9,5.7,4.3,6.1]，

tick\_label=['east','west','north','south','center'])



2个参数，基本上就解决大多数的绘制问题。

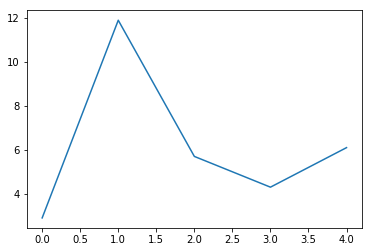
**第四件事：折线图**

[ 折线图用来，展示数据之间的差异性，一般适用于较多数据的情形 ]

**-函数：**plt.plot(range(N),data)

上述代码的两个参数和条状图很像，range(N)只要绘制多少个点，后面就是数据值。每个点之间用直线进行连接。我们看看下面这个代码将要绘制的图。我们依旧用上面的数据。

**-代码1：**plt.plot(range(5),[2.9,11.9,5.7,4.3,6.1])



再重复一下柱状图和折线图的区别：**数据量大的情况下，折线图会比条状图更容易看出变化的趋势。**

**第五件事：散点图**

[ 散点图用来，展示2组数据之间的关系，一般用于大量数据的情形 ]

**-函数：**plt.scatter(x,y)

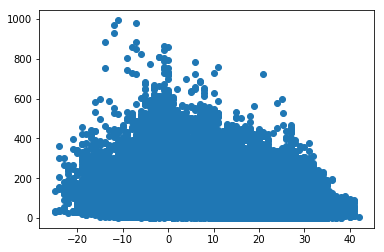
这里需要2组值，并且这2组值的数据量应该是完全一样。绘图的时候，函数会依次取出x数组和y数组的第一个值，绘制一个坐标点，然后是第二个值，绘制一个坐标点，直到最后一个值。为了提供x和y两个数组，我们先增加一个变量。

**-代码1：**array0=np.array(df['PM\_US\_Post']))

因为这里需要2组数据，因此，我们需要再初始化并赋值一个数据变量**array0**。方法和**array\_test**一样。

**-代码2：**plt.scatter(array\_test,array0)

因为我想展示温度和PM2.5的变化情况，所以**array\_test**变量取代x参数的位置，而**array0**变量取代y参数的位置。绘图如下：



**-上述散点图的解释：**从图中可以看到PM2.5的变化和温度高低关系不大，因为无论高温还是低温，都会有PM2.5在0-600区间变化。

**好了，以上就是给大家简单介绍几个matplotlib函数的绘图方法，其实每个函数都有很多参数，有兴趣的同学可以访问官网或者相关教程进行进一步的学习。同时，如果有同学觉得matplotlib绘制的可视化图形，不够炫酷，那么建议大家移步到《seanborn可视化库入门》这个介绍seanborn库的入门教程，相信大家可以在这里找到你想要的东西。**

- Matplotlib(2.2.2) -