

## 目录

一、柱状图.....	1
1、普通柱状图.....	1
2、两对柱子竖直方向堆积.....	2
3、两对柱子并排排列.....	3
4、给每根柱体加标注.....	3
二、饼状图.....	4

## 一、柱状图

### 1、普通柱状图

使用 `bar` 函数绘制柱状图，其常用参数如下：

参数	说明
<code>x</code>	柱形图的横坐标序列，可以是数字，也可以是字符串的列表，如 <code>x=['class1', 'class2', 'class3', 'class4', 'class5']</code> 此时起到了 <code>tick_label</code> 的作用
<code>y</code>	柱形图的高度，通常就是我们需要展示的数据
<code>align</code>	柱体对齐方式
<code>color</code>	<code>ylabel()</code> 类似
<code>edgecolor</code>	柱体边缘线条的颜色
<code>tick_label</code>	x轴的文字标签
<code>alpha</code>	透明度，数值越小越透明
<code>ls</code>	<code>linestyle</code> 的缩写，柱体边缘线条的样式
<code>lw</code>	<code>linewidth</code> 的缩写，边缘线条的宽度
<code>width</code>	柱体的宽度
<code>label</code>	标签，在展示图例时用到
<code>hatch</code>	柱体填充，取值有： <code>/, \,  , -, +, x, o, O, ., *</code> 等，且符号数量越多，越密集

其中非常常用的是 `width`、`label`、`alpha`、`edgecolor`、`hatch`。

此外，常用 `axe.set_xticks` 来设置刻度，使用 `axe.set_xtick labels` 来设置每个刻度上的标签。

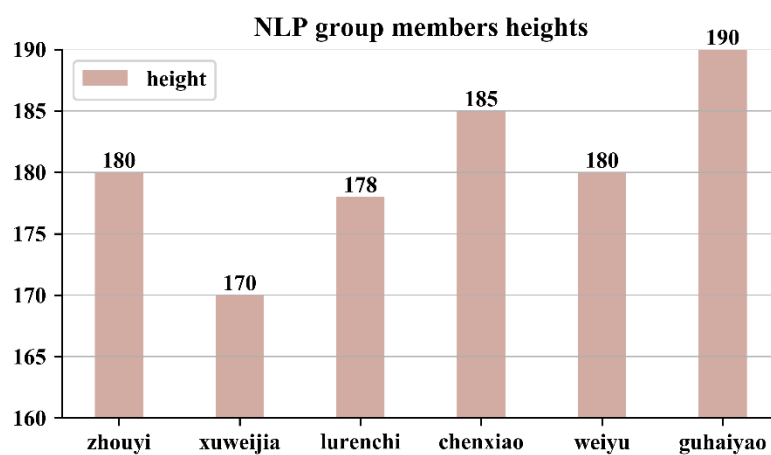
`axes.spines['top'].set_visible(False)`可以用来设置边框不可见，可选的有 `top`、`right`、`bottom`、`left` 四条边。

示例：

代码及注释：

```
fig, axes = plt.subplots(1, 1, figsize=(5, 3))
# 画柱状图
axes.bar(x, a1, width=0.4, label='height', color="#D2ACA3")
# 图例
axes.legend(loc='best')
# 设置坐标轴刻度、标签
axes.set_xticks([0, 1, 2, 3, 4, 5])
axes.set_yticks([160, 165, 170, 175, 180, 185, 190])
axes.set_ylim((160, 190))
axes.set_xticklabels(['zhouyi', 'xuweijia', 'lurenchi', 'chenxiao', 'weiyu', 'guhaiyao'])
# 设置title
axes.set_title('NLP group members heights')
# 网格线
axes.grid(linewidth=0.5, which="major", axis='y')
# 隐藏上、右边框
axes.spines['top'].set_visible(False)
axes.spines['right'].set_visible(False)
```

效果：



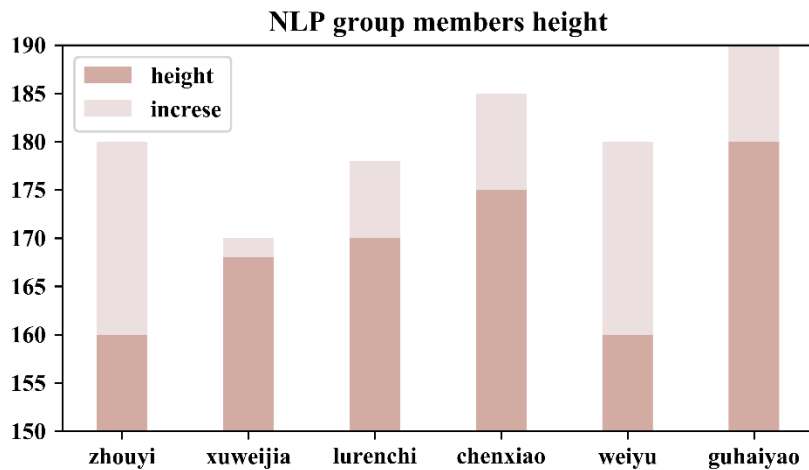
## 2、两对柱子竖直方向堆积

底下的柱体和普通柱状图画法一样。

画上面的柱子时，只需要再调用一个 bar 函数，这个 bar 函数要添加 bottom 参数，将 bottom 的值设置为底下柱子的数值即可。

示例：

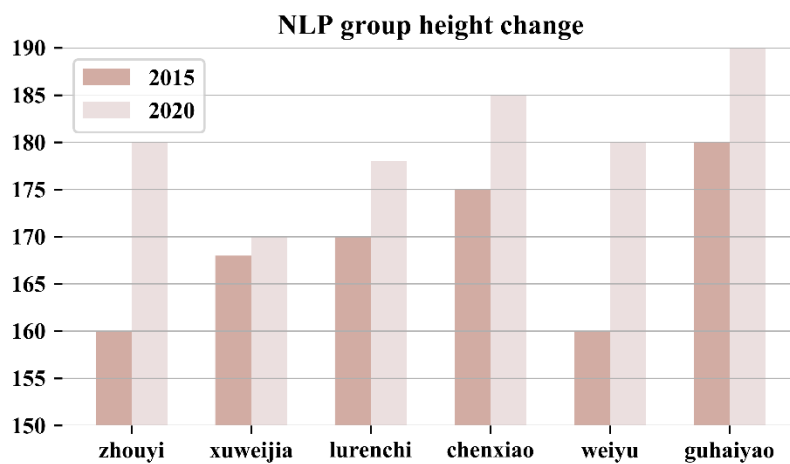
```
axes.bar(x, y1, width=0.4, label='height', color='#D2ACA3')
axes.bar(x, y2, width=0.4, bottom=y1, color='#EBDFDF', label='increse')
```



### 3、两对柱子并排排列

调用两次 bar 函数绘制两组柱体。让一组柱子的 x 值为  $x - \text{width} / 2$ ，另一组柱子的 x 值为  $x + \text{width} / 2$ ，即可实现两条柱子并排排列。示例：

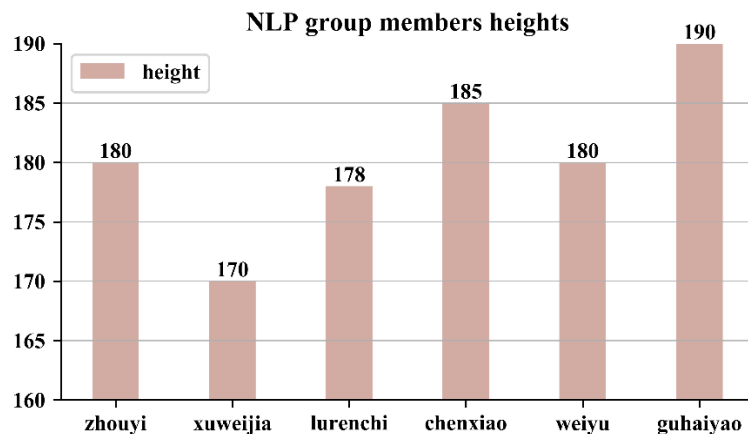
```
x = np.arange(6)
width = 0.3
axes.bar(x - width / 2, a1, width=width, label='2015', color="#ffe699")
axes.bar(x + width / 2, a2, width=width, label='2020', color="#bdd7ee")
```



### 4、给每根柱体加标注

一般使用循环的形式加 text，并将参数 ha 设置为 center、va 设置为 bottom。示例：

```
for i in range(6):
    axes.text(x[i], a1[i], a1[i], ha='center', va='bottom')
```



## 二、饼状图

饼图使用 `plt.pie` 函数绘制，其各参数如下：

参数	说明
<code>values</code>	需要展示的百分比数据
<code>explode</code>	饼片边缘偏离半径的百分比，如果对应的饼片为0，则不偏离
<code>labels</code>	饼片显示的标签
<code>autopct</code>	显示的百分比样式
<code>startangle</code>	第一个饼片逆时针旋转的角度，以x轴为起始位置
<code>shadow</code>	是否绘制阴影
<code>pctdistance</code>	百分比的text离圆心的距离，相对于半径
<code>labeldistance</code>	饼片标签离圆心的距离，相对于半径
<code>colors</code>	饼片的颜色

其中 `labels`、`value`、`colors`、`explode` 是维数相同的列表。还有一些常用的参数：

`textprops` 可以设置所有文字（`label` 和 `value`）的属性、`wedgeprops` 可以将饼状图设置为环状图。

`pie()` 函数的返回值为饼片列表、文字标签列表、数值标签列表。可以先获得返回值，再通过 `setp()` 函数对相应的文字进行操作、调整。

此外，一般会使用 `plt.axis('equal')` 让饼状图看起来更圆。

示例：

代码及注释：

```

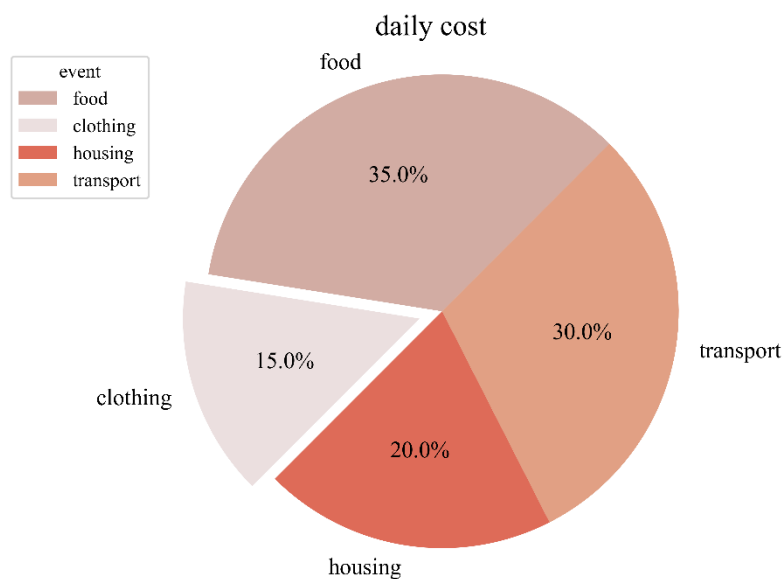
fig, axe = plt.subplots(1, 1, figsize=(8, 5))

font1 = {'family': 'Times New Roman', 'size': '15'}
# 饼状图各个部分的标签、值、颜色
labels = ['food', 'clothing', 'housing', 'transport']
values = [0.35, 0.15, 0.2, 0.3]
colors = ['#D2ACA3', '#EBDFDF', '#DE6B58', '#E1A084']
# 突出显示
explode = [0, 0.1, 0, 0]
# 标题
axe.set_title("daily cost", fontdict=font1)
# 画饼状图
wedge, texts, pcts = axe.pie(values, labels=labels, colors=colors, startangle=45, autopct='%3.1f%%',
                             , explode=explode)
axe.axis('equal')
# 图例
axe.legend(wedge, labels, fontsize=10, title='event', loc=2)
# 设置文本的属性

plt.setp(texts, size=12)
plt.setp(pcts, size=12)
plt.show()

```

效果：



如果给 pie 函数加上了 wedgeprops 参数，则可画出环形饼图，示例如下：

将上面的代码中的 pie 函数的参数增加 wedgeprops=dict(width=0.4)：

