

# 28–matplotlib(条形图)

## 一、简单垂直条形图

## 二、简单水平条形图

## 三、水平交错条形图

## 四、垂直堆叠条形图

条形图实际上是用来表示分组（或离散）变量的可视化，可以使用matplotlib模块中的**bar函数**完成条形图的绘制。

## 一、简单垂直条形图

### 案例一：直辖市GDP水平

中国的四个直辖市分别为北京市、上海市、天津市和重庆市，其2017年上半年的GDP分别为12406.8亿、13908.57亿、9386.87亿、9143.64亿。对于这样一组数据，我们该如何使用条形图来展示各自的GDP水平呢？

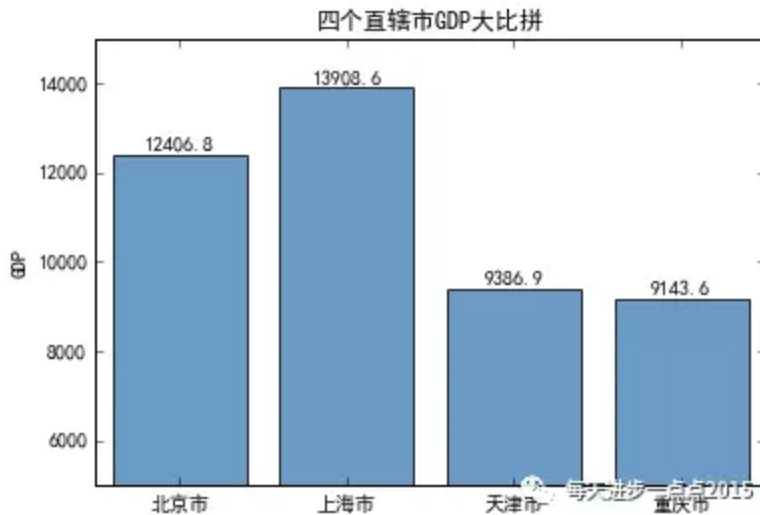
```
# 导入绘图模块
import matplotlib.pyplot as plt

# 构建数据
GDP = [12406.8, 13908.57, 9386.87, 9143.64]

# 中文乱码的处理
plt.rcParams['font.sans-serif'] = ['Microsoft YaHei']
plt.rcParams['axes.unicode_minus'] = False

# 绘图
plt.bar(range(4), GDP, align = 'center', color='steelblue', alpha = 0.8)
# 添加轴标签
plt.ylabel('GDP')
# 添加标题
plt.title('四个直辖市GDP大比拼')
# 添加刻度标签
plt.xticks(range(4), ['北京市', '上海市', '天津市', '重庆市'])
# 设置Y轴的刻度范围
plt.ylim([5000, 15000])
# 为每个条形图添加数值标签
for x, y in enumerate(GDP):
```

```
plt.text(x, y+100, '%s' %round(y, 1), ha='center')
# 显示图形
plt.show()
```



### 代码解读

- 由于matplotlib对中文的支持并不是很友好，所以需要提前对绘图进行字体的设置，即通过`rcParams`来设置字体，这里将字体设置为微软雅黑，同时为了避免坐标轴不能正常的显示负号，也需要进行设置；
- `bar`函数指定了条形图的x轴、y轴值，设置x轴刻度标签为水平居中，条形图的填充色`color`为铁蓝色，同时设置透明度`alpha`为0.8；
- 添加y轴标签、标题、x轴刻度标签值，为了让条形图显示各柱体之间的差异，将y轴范围设置在5000~15000；
- 通过循环的方式，添加条形图的数值标签；

## 二、简单水平条形图

### 案例二：同一本书不同平台最低价比较

很多人在买一本书的时候，都比较喜欢货比三家，例如《python数据分析实战》在亚马逊、当当网、中国图书网、京东和天猫的最低价格分别为39.5、39.9、45.4、38.9、33.34。针对这个数据，我们也可以通过条形图来完成，这里使用水平条形图来显示：

```
# 导入绘图模块
import matplotlib.pyplot as plt

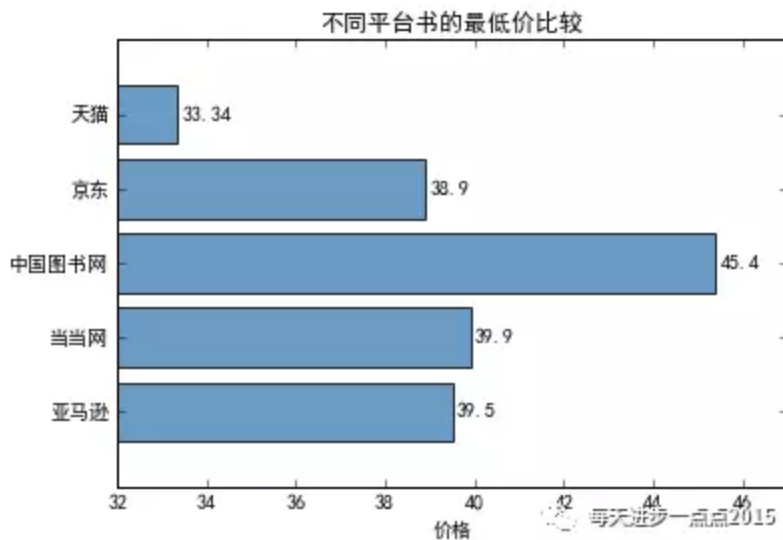
# 构建数据
price = [39.5, 39.9, 45.4, 38.9, 33.34]
```

```

# 中文乱码的处理
plt.rcParams['font.sans-serif'] = ['Microsoft YaHei']
plt.rcParams['axes.unicode_minus'] = False

# 绘图
plt.barh(range(5), price, align = 'center', color='steelblue', alpha = 0.8)
# 添加轴标签
plt.xlabel('价格')
# 添加标题
plt.title('不同平台书的最低价比较')
# 添加刻度标签
plt.yticks(range(5), ['亚马逊', '当当网', '中国图书网', '京东', '天猫'])
# 设置Y轴的刻度范围
plt.xlim([32, 47])
# 为每个条形图添加数值标签
for x, y in enumerate(price):
    plt.text(y+0.1, x, '%s' %y, va='center')
# 显示图形
plt.show()

```



### 代码解读

- 水平条形图的绘制与垂直条形图的绘制步骤一致，只是调用了**barh函数**来完成。需要注意的是，条形图的数值标签设置有一些不一样，需要将标签垂直居中显示，使用**va参数**即可。

## 三、水平交错条形图

以上讲的简单垂直和水平条形图是基于一种离散变量的情况，针对两种离散变量的条形图我们可以使用水平交错条形图和堆叠条形图，下面我们就来看看这两种条形图是如何绘制的。

### 案例三：胡润财富榜：亿万资产超高净值家庭数

利用水平交错条形图对比2016年和2017年亿万资产超高净值家庭数（top5），其数据如下：

```
# 导入绘图模块
import matplotlib.pyplot as plt
import numpy as np

# 构建数据
Y2016 = [15600, 12700, 11300, 4270, 3620]
Y2017 = [17400, 14800, 12000, 5200, 4020]
labels = ['北京', '上海', '香港', '深圳', '广州']
bar_width = 0.45

# 中文乱码的处理
plt.rcParams['font.sans-serif'] = ['Microsoft YaHei']
plt.rcParams['axes.unicode_minus'] = False

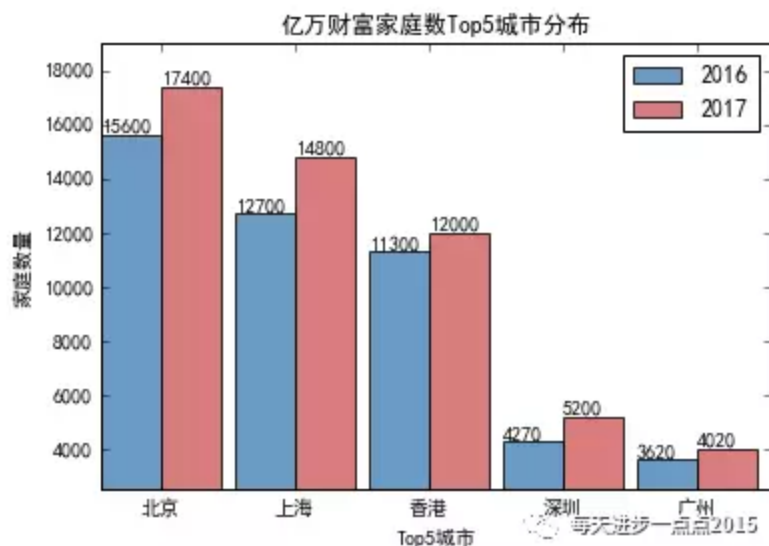
# 绘图
plt.bar(np.arange(5), Y2016, label = '2016', color = 'steelblue', alpha = 0.8, width = bar_width)
plt.bar(np.arange(5)+bar_width, Y2017, label = '2017', color = 'indianred', alpha = 0.8, width = bar_width)

# 添加轴标签
plt.xlabel('Top5城市')
plt.ylabel('家庭数量')
# 添加标题
plt.title('亿万财富家庭数Top5城市分布')
# 添加刻度标签
plt.xticks(np.arange(5)+bar_width, labels)
# 设置Y轴的刻度范围
plt.ylim([2500, 19000])
# 为每个条形图添加数值标签
for x2016, y2016 in enumerate(Y2016):
    plt.text(x2016, y2016+100, '%s' %y2016)

for x2017, y2017 in enumerate(Y2017):
    plt.text(x2017+bar_width, y2017+100, '%s' %y2017)

# 显示图例
```

```
plt.legend()
# 显示图形
plt.show()
```



## 代码解读

- 水平交错条形图绘制的思想很简单，就是在第一个条形图绘制好的基础上，往左移一定的距离，再去绘制第二个条形图，所以在代码中会出现两个bar函数；
- 图例的绘制需要在bar函数中添加label参数；color和alpha参数分别代表条形图的填充色和透明度；
- 给条形图添加数值标签，同样需要使用两次for循环的方式实现；

## 四、垂直堆叠条形图

垂直堆叠条形图的绘制思想与水平交错条形图一样，只不过一个是向上偏移，一个是往左偏移，具体我们以案例说明。

### 案例四：2017年物流运输量情况分布

```
# 导入模块
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# 导入数据
data = pd.read_excel('C:\\Users\\Administrator\\Desktop\\货运.xls')

# 绘图
plt.bar(np.arange(8), data.loc[0, :][1:], color = 'red', alpha = 0.8, label =
```

```

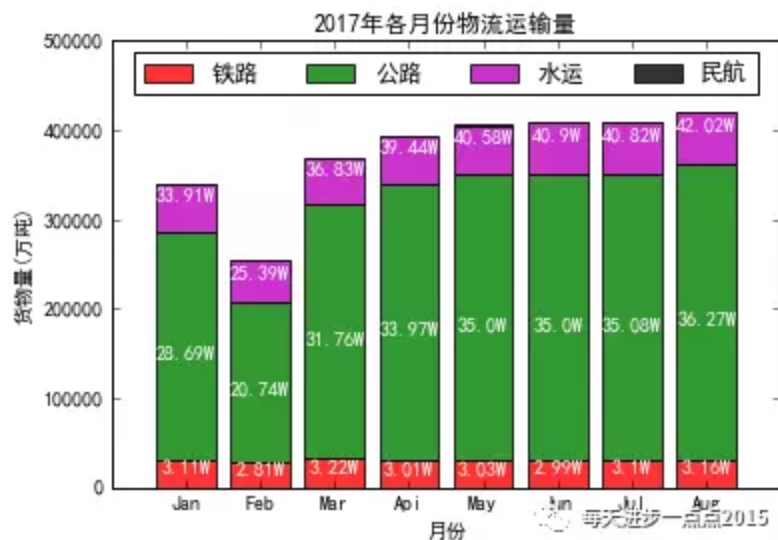
    '铁路', align = 'center')
plt.bar(np.arange(8), data.loc[1, :][1:], bottom = data.loc[0, :][1:], color
= 'green', alpha = 0.8, label = '公路', align = 'center')
plt.bar(np.arange(8), data.loc[2, :][1:], bottom = data.loc[0, :][1:] + data.loc[1, :][1:], color = 'm', alpha = 0.8, label = '水运', align = 'center')
plt.bar(np.arange(8), data.loc[3, :][1:], bottom = data.loc[0, :][1:] + data.loc[1, :][1:] + data.loc[2, :][1:], color = 'black', alpha = 0.8, label = '民航',
align = 'center')

# 添加轴标签
plt.xlabel('月份')
plt.ylabel('货物量(万吨)')
# 添加标题
plt.title('2017年各月份物流运输量')
# 添加刻度标签
plt.xticks(np.arange(8), data.columns[1:])
# 设置Y轴的刻度范围
plt.ylim([0, 500000])
# 为每个条形图添加数值标签
for x_t, y_t in enumerate(data.loc[0, :][1:]):
    plt.text(x_t, y_t/2, '%sW' % (round(y_t/10000, 2)), ha='center', color = 'white')

for x_g, y_g in enumerate(data.loc[0, :][1:] + data.loc[1, :][1:]):
    plt.text(x_g, y_g/2, '%sW' % (round(y_g/10000, 2)), ha='center', color = 'white')

for x_s, y_s in enumerate(data.loc[0, :][1:] + data.loc[1, :][1:] + data.loc[2, :][1:]):
    plt.text(x_s, y_s-20000, '%sW' % (round(y_s/10000, 2)), ha='center', color = 'white')
# 显示图例
plt.legend(loc='upper center', ncol=4)
# 显示图形
plt.show()

```



## 代码解读

- 垂直条形图的绘制不仅仅需要提供x,y轴的数值，还需要提供**bottom参数**，其目的就是在某个条形图顶端的基础上，绘制其他条形图，以此类推可以绘制多个堆叠条形图；
- **图例的位置**选择在了正上方，且设置列数为4，表面图例以一排的形式展现；
- 堆叠条形图的数值标签，任然是按照y轴方向堆叠的思想，贴上数值标签值；
- 由于航空运输的货物量非常少，导致图中几乎无法显示；