

28–matplotlib(树地图)

前言

函数语法及参数

前言

用[可视化](#)的方法来表达[离散变量](#)的数值情况，不仅仅[可以使用条形图、饼图、热力图](#)，我们还可以借助于[树地图](#)来完成。树地图的思想就是通过方块的面积来表示，面积越大，其代表的值就越大，反之亦然。今天要跟大家分享的就是如何通过Python这个工具，完成树地图的绘制。

函数语法及参数

在Python中，可以借助于squarify包来绘制，即squarify.plot函数。首先，我们来看一下这个函数的[语法及参数含义](#)：

```
squarify.plot(sizes,
               norm_x=100,
               norm_y=100,
               color=None,
               label=None,
               value=None,
               alpha,
               **kwargs)
```

- **sizes**: 指定离散变量各水平对应的数值，即[反映树地图子块的面积大小](#)；
- **norm_x**: 默认将x轴的范围限定在0–100之内；
- **norm_y**: 默认将y轴的范围限定在0–100之内；
- **color**: 自定义设置树地图子块的[填充色](#)；
- **label**: 为每个子块指定[标签](#)；
- **value**: 为每个子块添加[数值大小的标签](#)；
- **alpha**: 设置填充色的[透明度](#)；
- ****kwargs**: 关键字参数，与条形图的关键字参数类似，如设置边框色、边框粗细等；

介绍完了绘图所需要的函数，先来抛一个树地图的效果图，该图反映的是**2017年8月中央财政收入**的主要来源情况：



树地图的绘制

图中的数据来自《中华人民共和国财政部》官网，具体可以从**2017年8月财政收支情况** (http://gks.mof.gov.cn/zhengfuxinxi/tongjishuju/201709/t20170911_2695830.html) 查看。

```
# 导入第三方包
import matplotlib.pyplot as plt
import squarify

#中文及负号处理办法
plt.rcParams['font.sans-serif'] = 'Microsoft YaHei'
plt.rcParams['axes.unicode_minus'] = False

# 创建数据
name = ['国内增值税', '国内消费税', '企业所得税', '个人所得税',
        '进口增值税、消费税', '出口退税', '城市维护建设税',
        '车辆购置税', '印花稅', '资源稅', '土地和房稅', '车船稅烟叶稅等']
income = [3908, 856, 801, 868, 1361, 1042, 320, 291, 175, 111, 414, 63]

# 绘图
colors = ['steelblue', '#9999ff', 'red', 'indianred',
```

```

    'green', 'yellow', 'orange']
    plot = squarify.plot(sizes = income, # 指定绘图数据
                        label = name, # 指定标签
                        color = colors, # 指定自定义颜色
                        alpha = 0.6, # 指定透明度
                        value = income, # 添加数值标签
                        edgecolor = 'white', # 设置边界框为白色
                        linewidth = 3 # 设置边框宽度为3
                    )
    # 设置标签大小
plt.rc('font', size=8)
# 设置标题大小
plot.set_title('2017年8月中央财政收支情况', fontdict = {'fontsize':15})

# 去除坐标轴
plt.axis('off')
# 去除上边框和右边框刻度
plt.tick_params(top = 'off', right = 'off')
# 显示图形
plt.show()

```

通过上面的这些绘图语句和美化语句，就可以得到一个形象不错的树地图了。是不是很简单呢，如果你的工作中需要绘制离散变量的可视化，也可以考虑考虑这张图呢~简单明了形象佳！