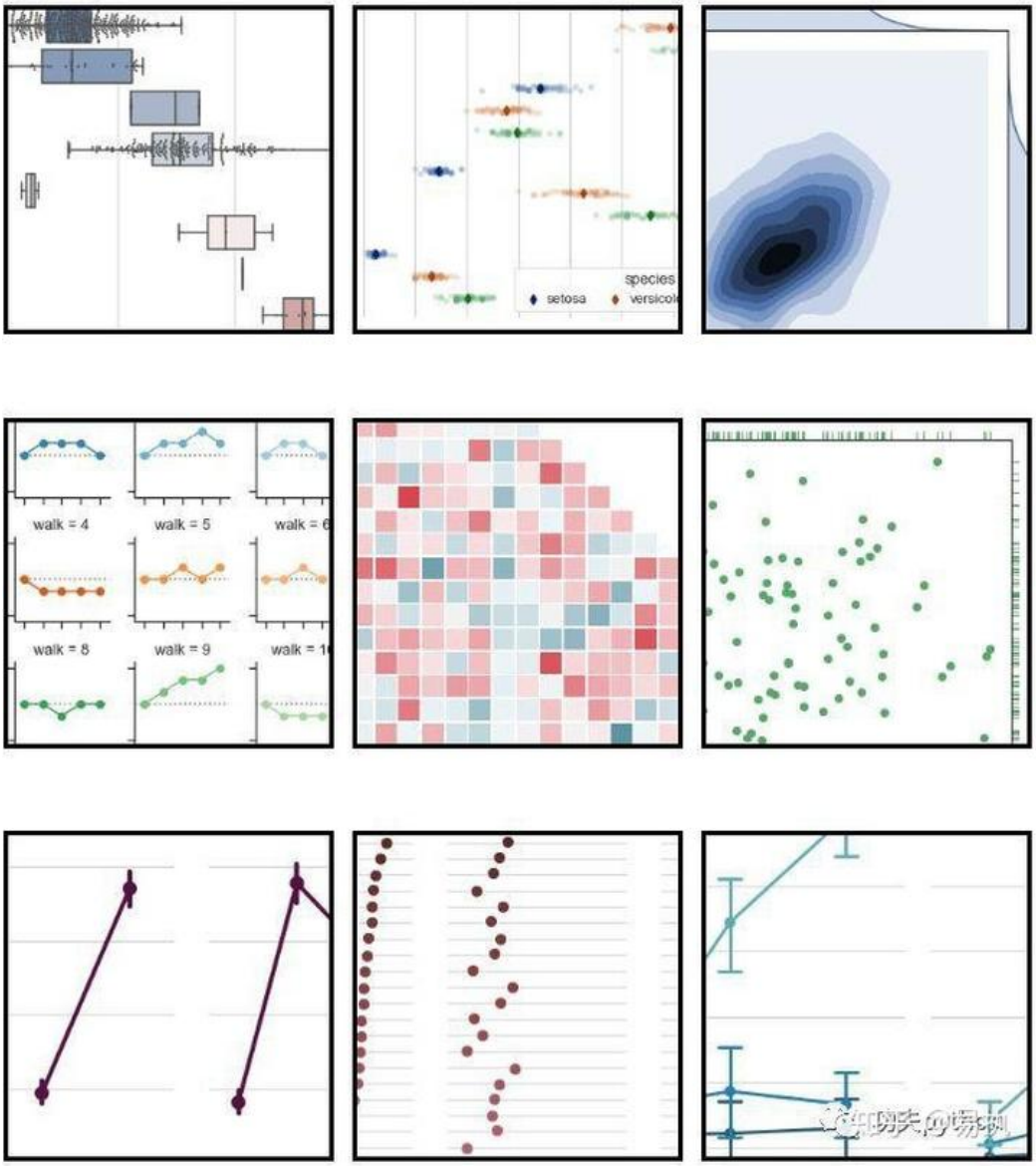
zhuanlan.zhihu.com

Python可视化 | Seaborn5分钟入门(六)——heatmap热力图

8-10 minutes

Seaborn是基于matplotlib的Python可视化库。它提供了一个高级界面来绘制有吸引力的统计图形。Seaborn其实是在matplotlib的基础上进行了更高级的API封装，从而使得作图更加容易，不需要经过大量的调整就能使你的图变得精致。



注：所有代码均在IPython notebook中实现

heatmap 热力图

热力图在实际中常用于展示一组变量的相关系数矩阵，在展示列联表的数据分布上也有较大的用途，通过热力图我们可以非常直观地感受到数值大小的差异状况。heatmap的API如下所示：

```
seaborn.heatmap(data,vmin=None,vmax=None,cmap=None,center=None,
robust=False,annot=None,fmt='.2g',annot_kws=None,linewidths=0,linecolor='white',
cbar=True,cbar_kws=None,cbar_ax=None,square=False,xticklabels='auto',
yticklabels='auto', mask=None, ax=None, **kwargs)
```


下面将演示这些主要参数的用法，第一件事还是先导入相关的packages。

```
import seaborn as sns
%matplotlib inline
sns.set(font_scale=1.5)
```

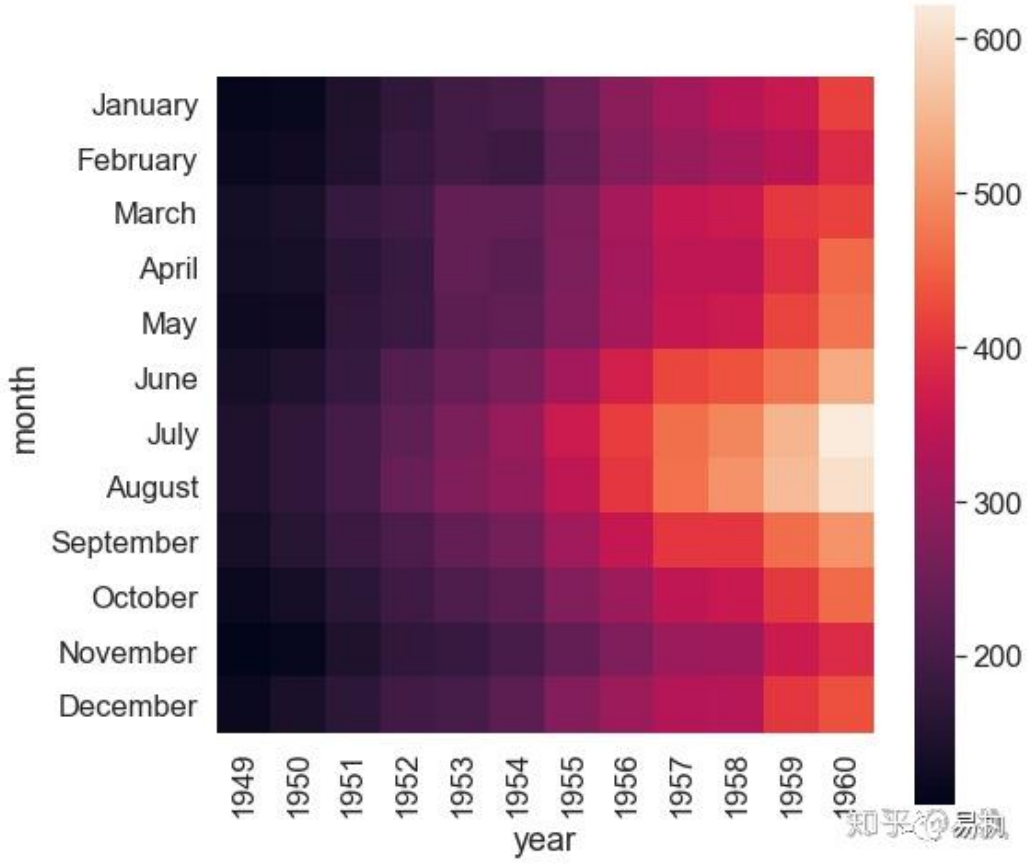
本次演示采用的数据集是Seaborn中内置的**flights航班数据集**:

```
#导入数据集后按年月两个维度进行数据透视
data=sns.load_dataset("flights")\
        .pivot("month","year","passengers")
data.head()
```

year	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960
month												
January	112	115	145	171	196	204	242	284	315	340	360	417
February	118	126	150	180	196	188	233	277	301	318	342	391
March	132	141	178	193	236	235	267	317	356	362	406	419
April	129	135	163	181	235	227	269	313	348	348	396	461
May	121	125	172	183	229	234	270	318	355	363	420	472

如上图所示，dataframe中的数据代表了1949年-1960年每个月的航班乘客数量，接下来热力图就隆重登场啦！

```
sns.set_context({"figure.figsize":(8,8)})
sns.heatmap(data=data,square=True)
#可以看到热力图主要展示的是二维数据的数据关系
#不同大小的值对应不同的颜色深浅
```

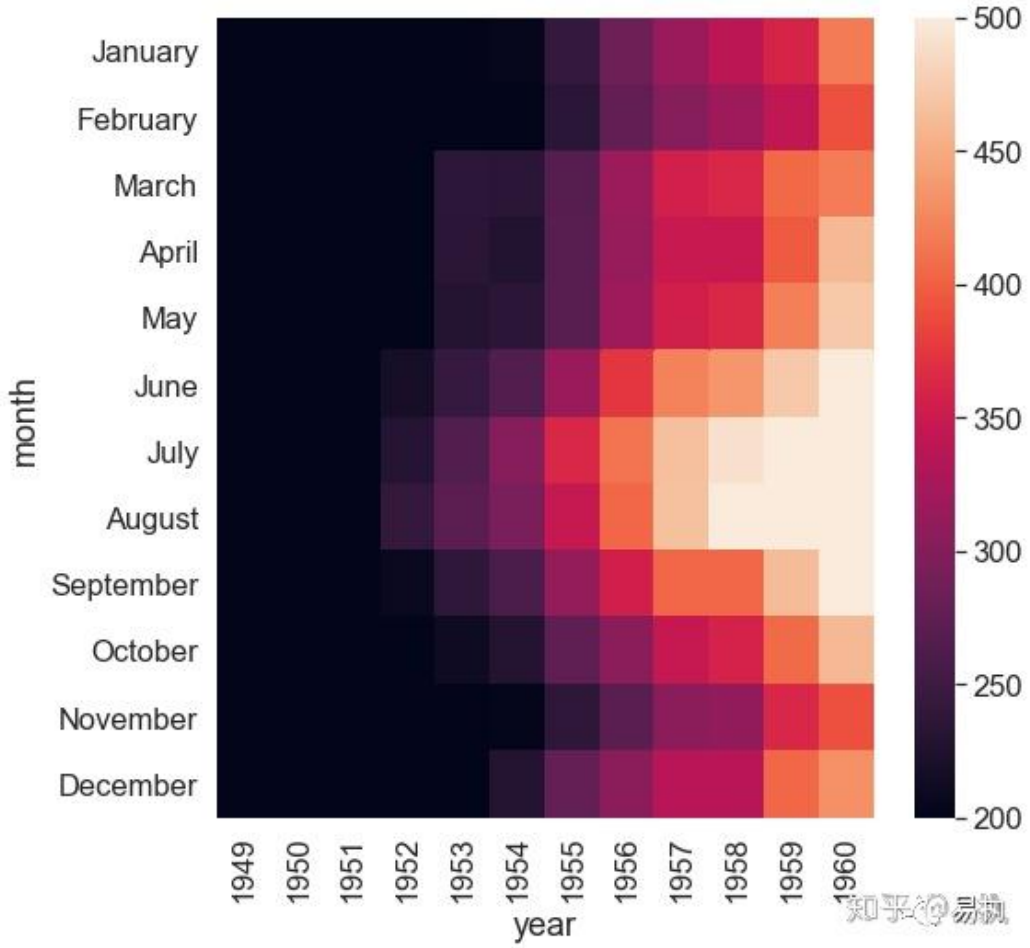


热力图的右侧是颜色带，上面代表了数值到颜色的映射，数值由小到大对应色彩由暗到亮。从上面的heatmap中我们可以得到两层信息，一是随着时间的推移，飞机的乘客数量是在逐步增多的，二是航班的乘坐旺季在七月和八月份。下面就具体的参数进行演示。

`vmax`：设置颜色带的最大值

`vmin`：设置颜色带的最小值

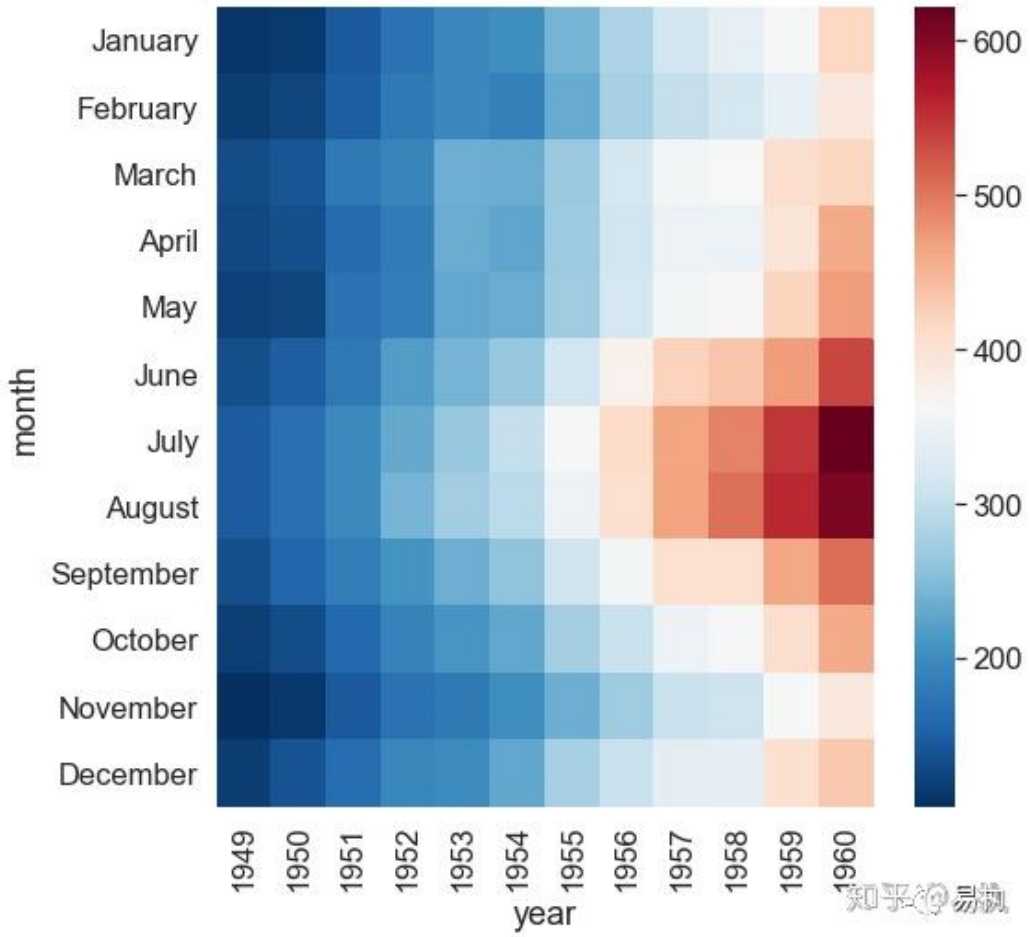
```
sns.heatmap(data=data,vmin=200,vmax=500)
```



可以看到右侧的颜色带最大最小值变了，而heatmap中颜色映射关系也会随之调整，将本图和上面的图进行对比便一目了然。

`cmap`：设置颜色带的色系

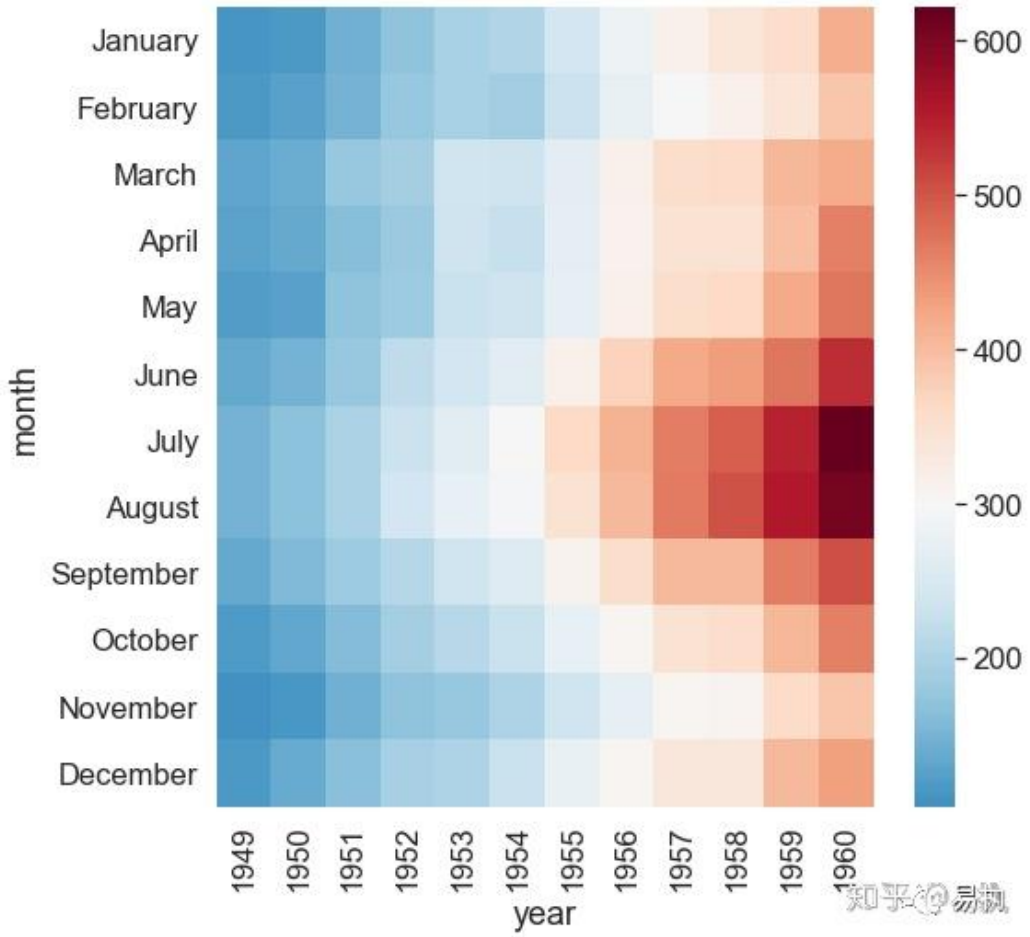
```
sns.heatmap(data=data,cmap="RdBu_r")
```



好像变好看了?

`center`：设置颜色带的分界线

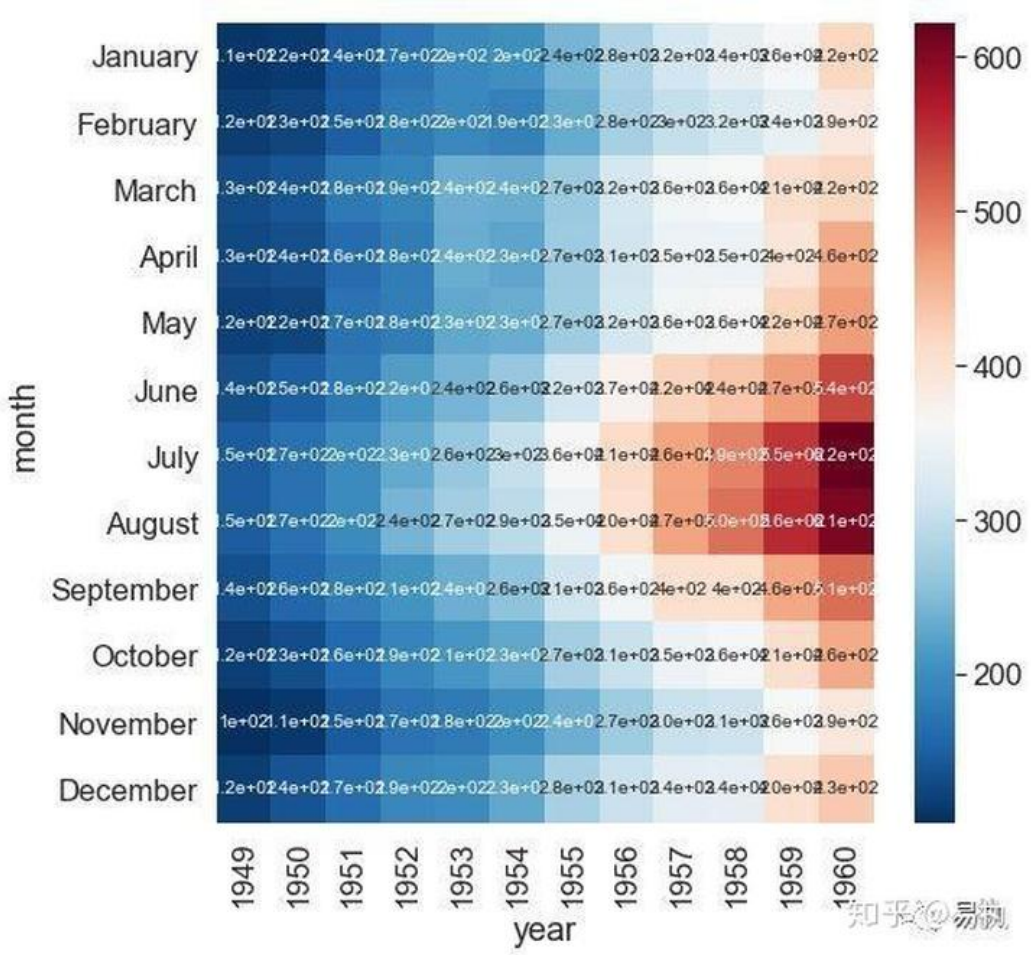
```
sns.heatmap(data=data,cmap="RdBu_r",center=300)
```



细心的朋友可以察觉到颜色带上色彩两级的分界线变成了300

annot：是否显示数值注释

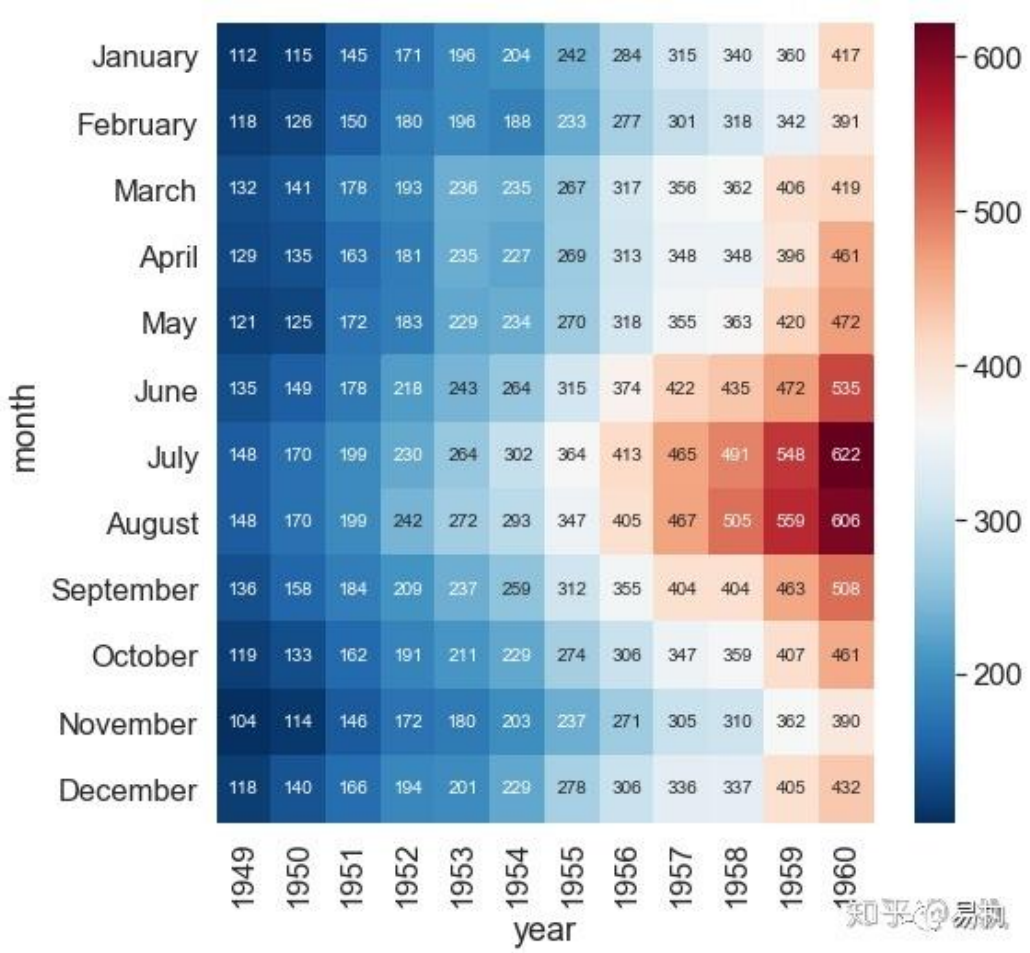
```
sns.heatmap(data=data,annot=True,cmap="RdBu_r")
```



怎么回事？乱码了吗？其实数值注释默认显示的是**科学记数法**的数值，我们得把数值进行格式化，这就用到了下面的参数。

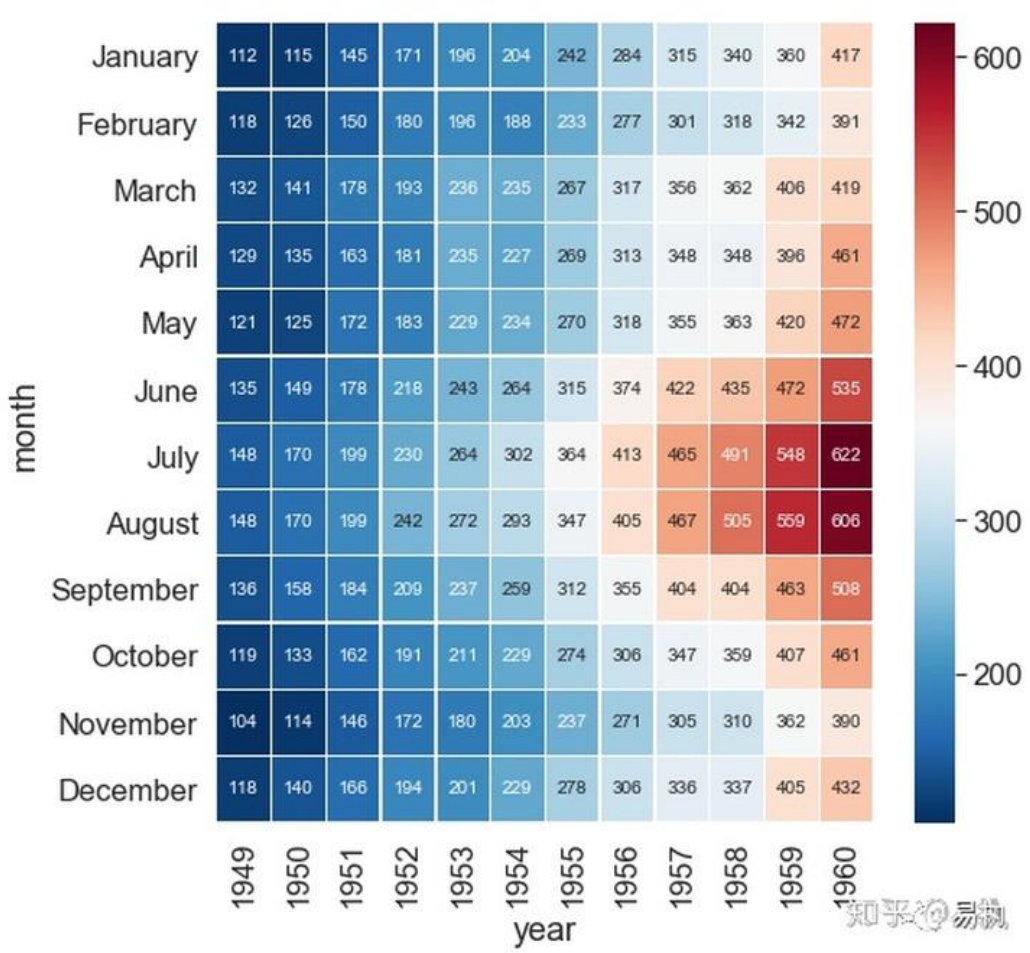
`fmt`：format的缩写，设置数值的格式化形式

```
sns.heatmap(data=data,annot=True,fmt="d",cmap="RdBu_r")
#foramt为int类型
```



`linewidths`：控制每个小方格之间的间距

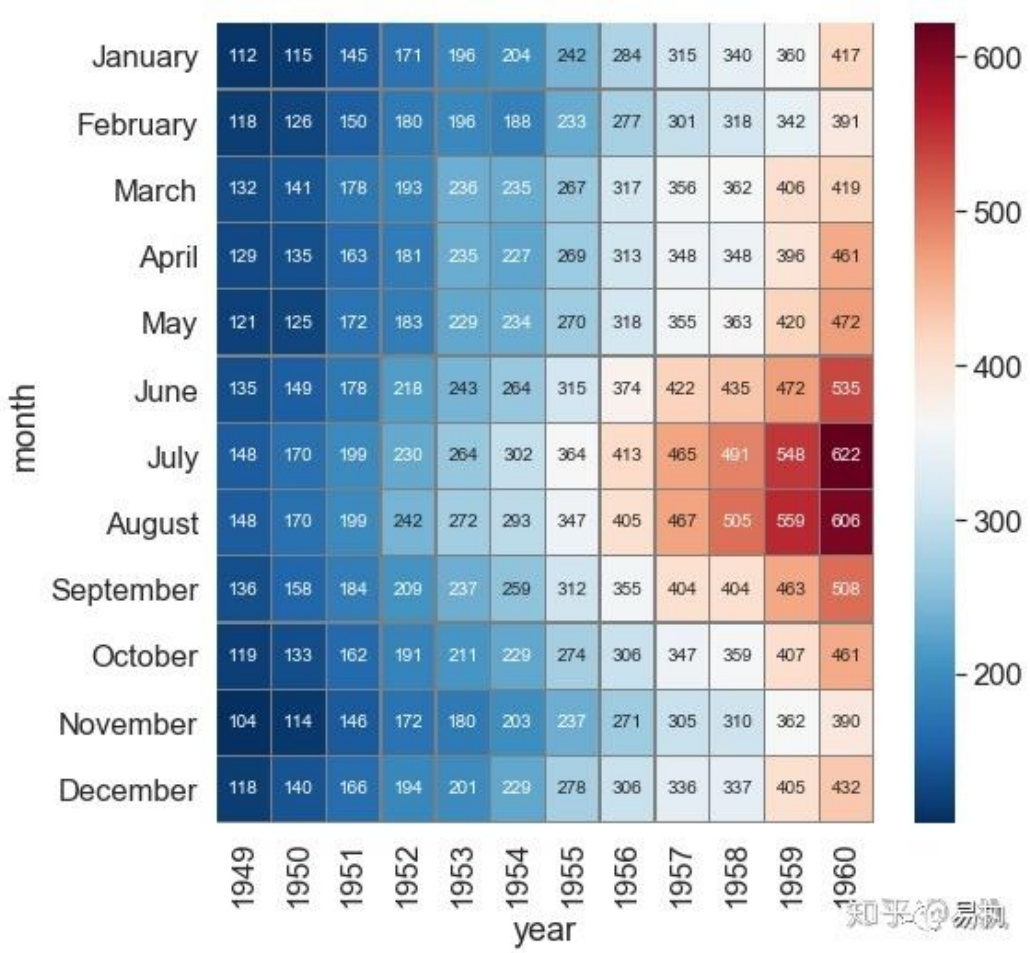
```
sns.heatmap(data=data,annot=True,fmt="d",linewidths=0.3,cmap="RdBu_r")
#可以看到每个小方格之产生了间隙
```



`linecolor`：控制分割线的颜色

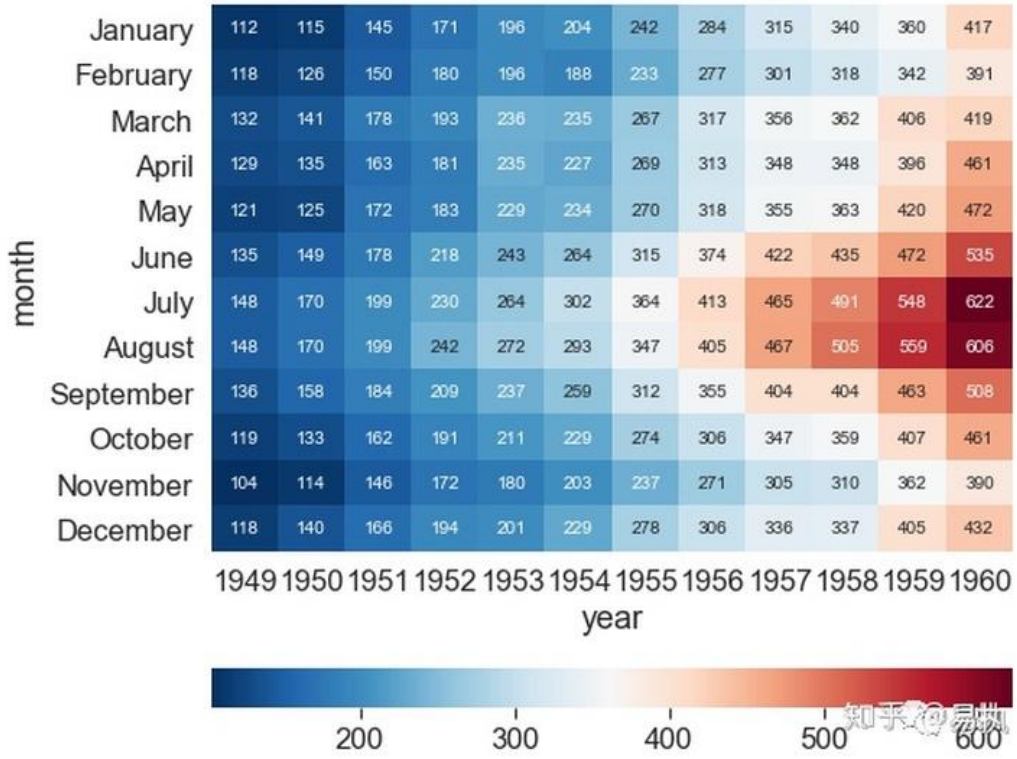
```
sns.heatmap(data=data,annot=True,fmt="d",linewidths=0.3,linecolor="grey",cmap="RdBu_r")

#原来的白色间隙变成了灰色间隙
```



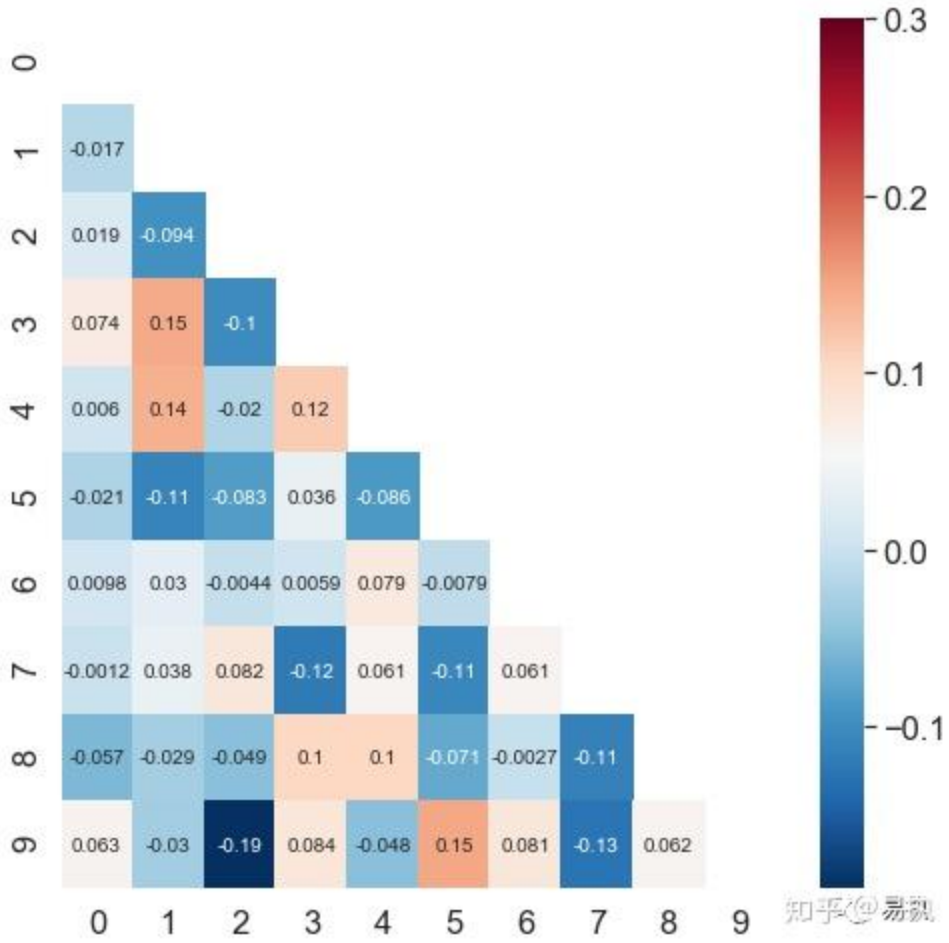
cbar_kws：关于颜色带的设置

```
sns.heatmap(data=data,annot=True,fmt="d",cmap="RdBu_r",
                        cbar_kws={"orientation":"horizontal"})
#横向显示颜色帮
```



mask：传入布尔型矩阵，若为矩阵内为True，则热力图相应的位置的数据将会被屏蔽掉（常用在绘制相关系数矩阵图）

```
import numpy as np
#随机生成一个200行10列的数据集
data_new = np.random.randn(200,10)
#求出这个数据集的相关系数矩阵 corr = np.corrcoef(data_new,rowvar=False)
#以corr的形状生成一个全为0的矩阵
mask = np.zeros_like(corr)
#将mask的对角线及以上设置为True
#这部分就是对应要被遮掉的部分mask[np.triu_indices_from(mask)] = True
with sns.axes_style("white"):
    sns.heatmap(corr, mask=mask, vmax=0.3, annot=True,cmap="RdBu_r")
```



如果大家对上面的代码流程不大了解，可以把mask打印出来看看

```
1    mask
```

```
array([[1., 1., 1., 1., 1., 1., 1., 1., 1., 1.],
       [0., 1., 1., 1., 1., 1., 1., 1., 1., 1.],
       [0., 0., 1., 1., 1., 1., 1., 1., 1., 1.],
       [0., 0., 0., 1., 1., 1., 1., 1., 1., 1.],
       [0., 0., 0., 0., 1., 1., 1., 1., 1., 1.],
       [0., 0., 0., 0., 0., 1., 1., 1., 1., 1.],
       [0., 0., 0., 0., 0., 0., 1., 1., 1., 1.],
       [0., 0., 0., 0., 0., 0., 0., 1., 1., 1.],
       [0., 0., 0., 0., 0., 0., 0., 0., 1., 1.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 1.]])
```


参照mask和上面绘制的图，应该就很容易理解了，mask中为1的部分，就是要被盖掉的部分。演示到此为止，想更深入的学习可以自行查阅官方文档

原创不易，如果觉得有点用，希望可以随手点个赞，拜谢各位老铁。