zhuanlan.zhihu.com

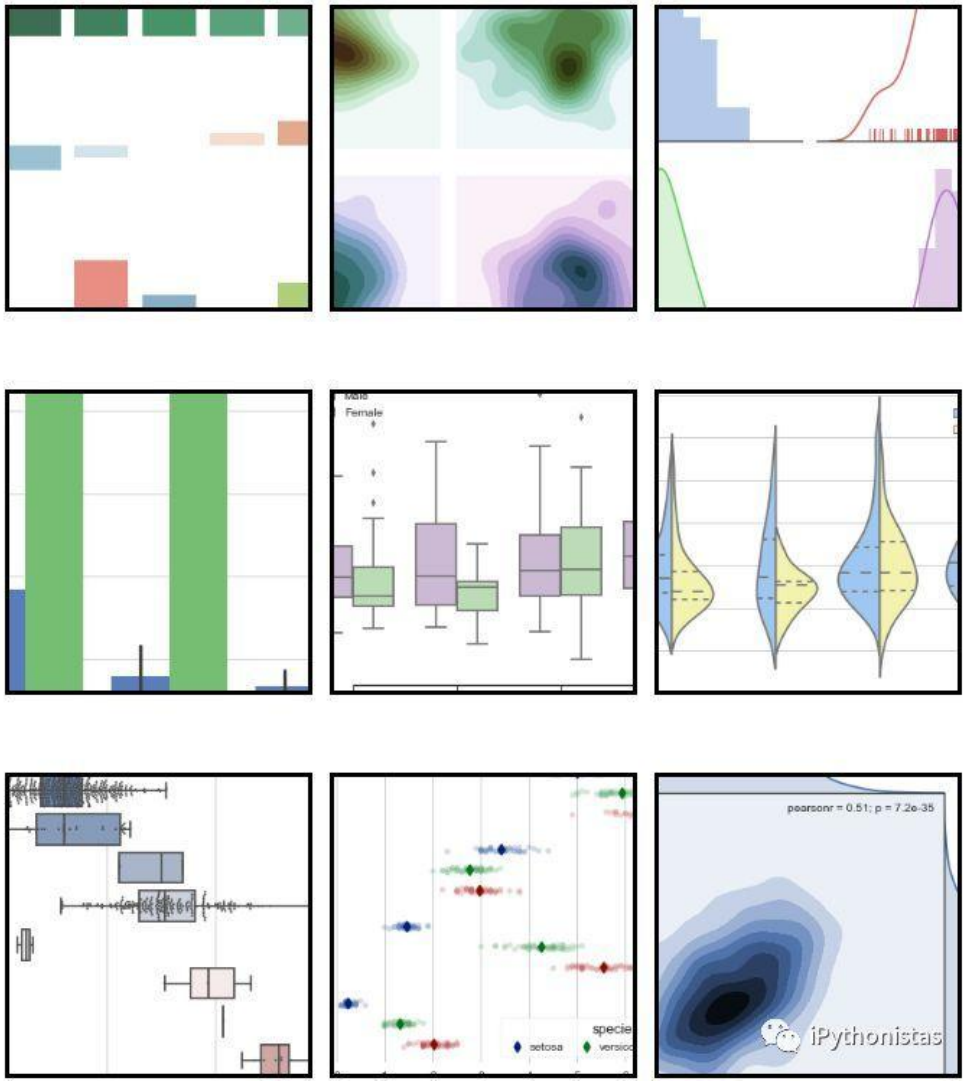
Python可视化 | Seaborn5分钟入门(三)——boxplot和violinplot

18-22 minutes

微信公众号：「Python读财」

如有问题或建议，请公众号留言

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。**但应强调的是，应该把Seaborn视为matplotlib的补充，而不是替代物。**



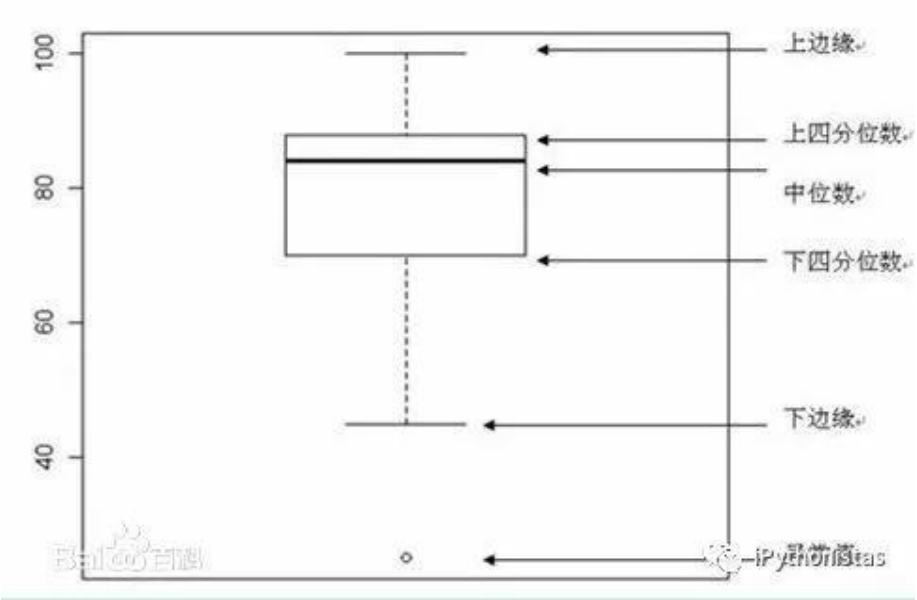
注：所有代码均在IPython notebook中实现

---

boxplot

箱形图（Box-plot）又称为盒须图、盒式图或箱线图，是一种用作显示一组数据分散情况资料的统计图。它能显示出一组数据的**最大值**、**最小值**、**中位数**及**上下四分位数**。因形状如箱子而得名。在各种领域也经

常被使用，常见于品质管理。图解如下：



接下来我们介绍Seaborn中的箱型图的具体实现方法，这是boxplot的API：

> *seaborn.boxplot(x=None, y=None, hue=None, data=None, order=None, hue_order=None, orient=None, color=None, palette=None, saturation=0.75, width=0.8, dodge=True, fliersize=5, linewidth=None, whis=1.5, notch=False, ax=None, **kwargs)*

我们从具体的实例出发

```
%matplotlib inline
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
plt.rc("font",family="SimHei",size="15")  #解决中文乱码问题
```

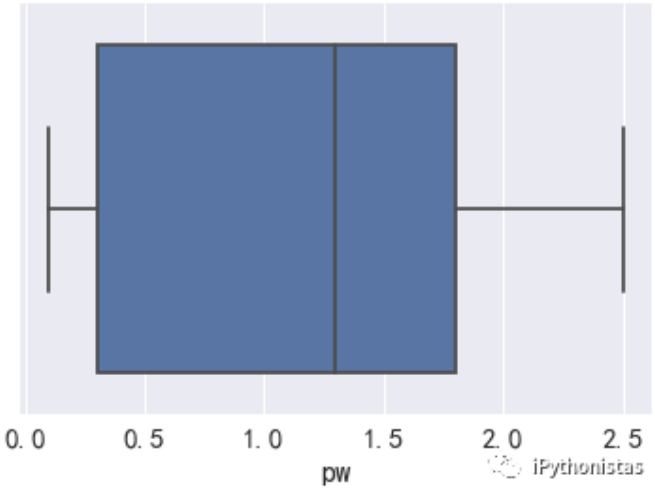
本文所使用的数据集是鸢尾花卉数据集

	sl	sw	pl	pw	catagory	color
0	5.1	3.5	1.4	0.2	0	red
1	4.9	3.0	1.4	0.2	0	yellow
2	4.7	3.2	1.3	0.2	0	green
3	4.6	3.1	1.5	0.2	0	green
4	5.0	3.6	1.4	0.2	0	blue
5	5.4	3.9	1.7	0.4		

x，y：dataframe中的**列名（str）**或者**矢量数据**

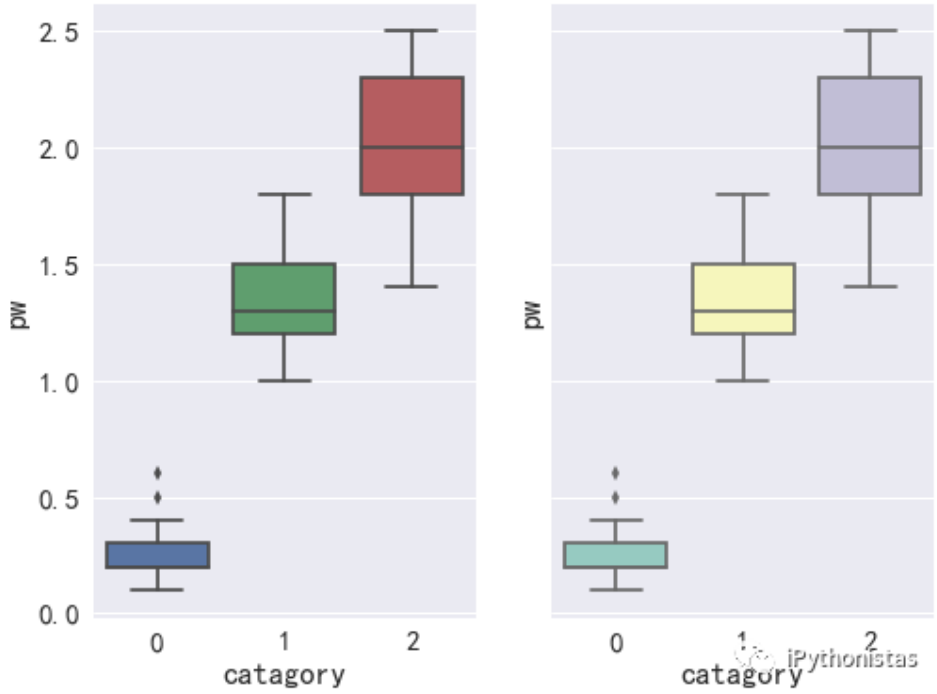
data：dataframe或者数组

```
sns.boxplot(x=data["pw"],data=data)
```



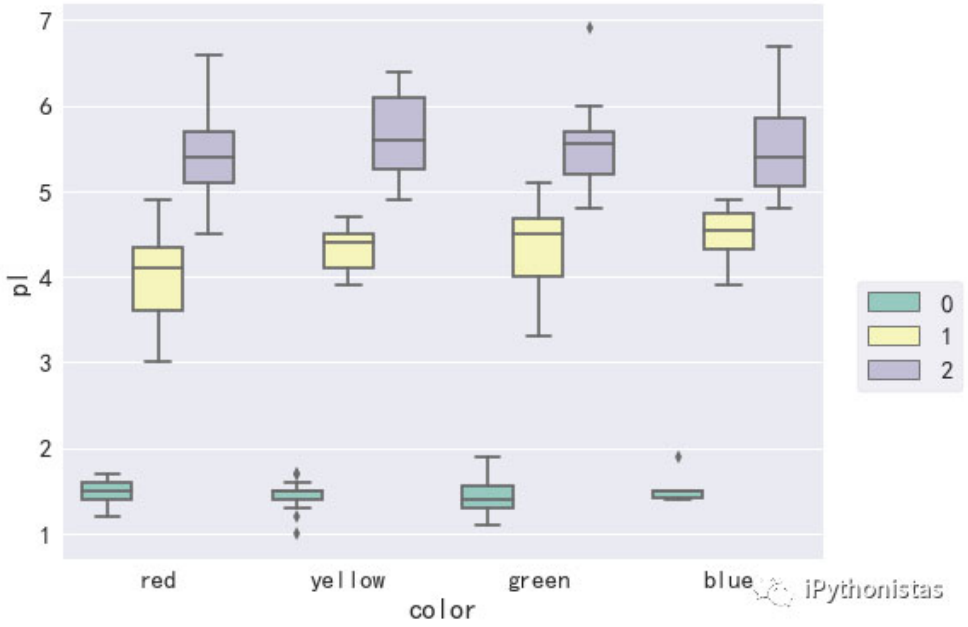
palette：调色板，控制图像的色调

```
fig,axes=plt.subplots(1,2,sharey=True)
sns.boxplot(x="catagory",y="pw",data=data,ax=axes[0]) #左图
sns.boxplot(x="catagory",y="pw",data=data,palette="Set3",ax=axes[1]) #右图
```



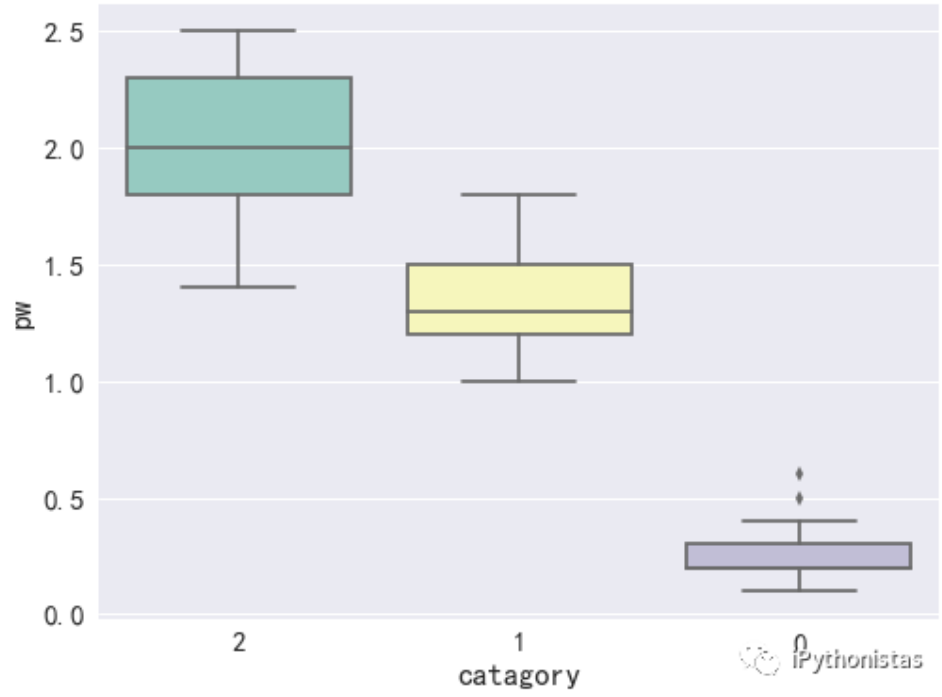
hue（str）：dataframe的列名，按照列名中的值分类形成分类的条形图

```
sns.boxplot(x="color",y="pl",data=data,hue="catagory",palette="Set3")
```



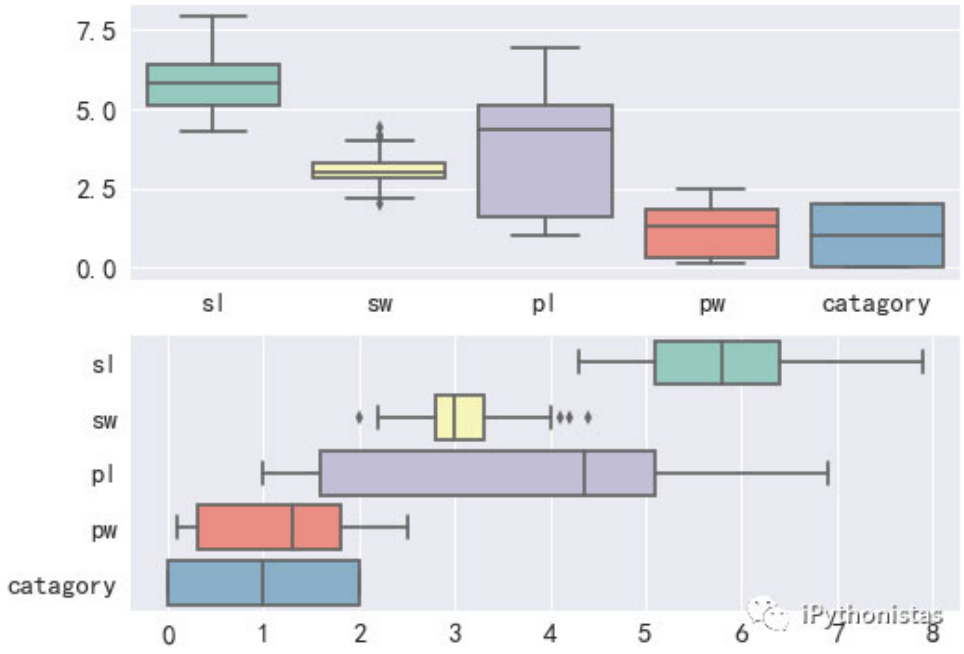
order, hue_order (lists of strings)：用于控制条形图的顺序

```
sns.boxplot(x="catagory",y="pw",data=data,palette="Set3",order=[2,1,0])
```



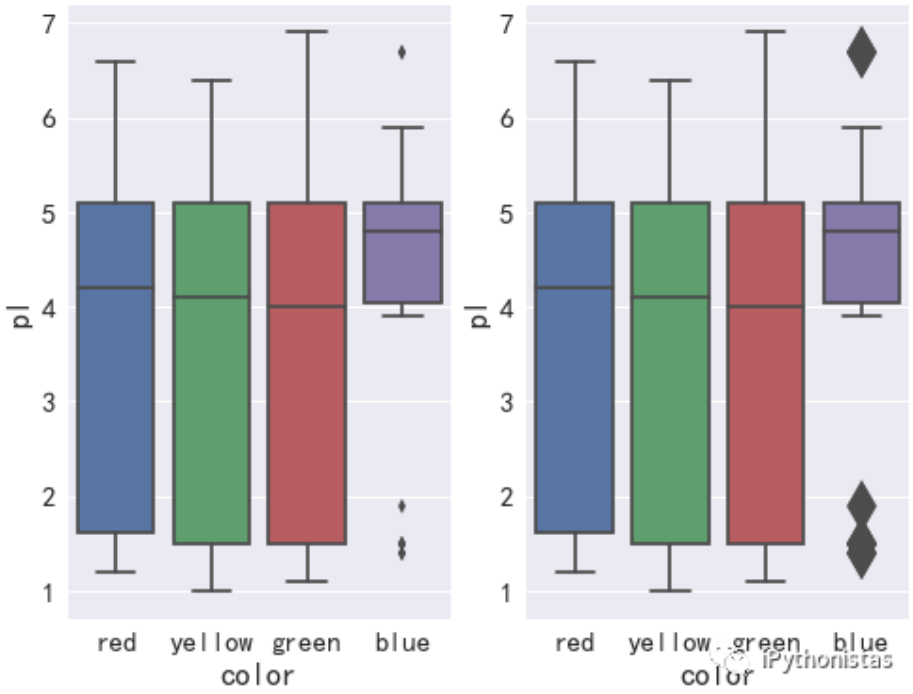
orient："v"|"h" 用于控制图像使水平还是竖直显示（这通常是从输入变量的dtype推断出来的，此参数一般当**不传入x、y，只传入data的时候**使用）

```
fig,axes=plt.subplots(2,1)
sns.boxplot(data=data,orient="v",palette="Set3",ax=axes[0])  #竖直显示
sns.boxplot(data=data,orient="h",palette="Set3",ax=axes[1])  #水平显示
```



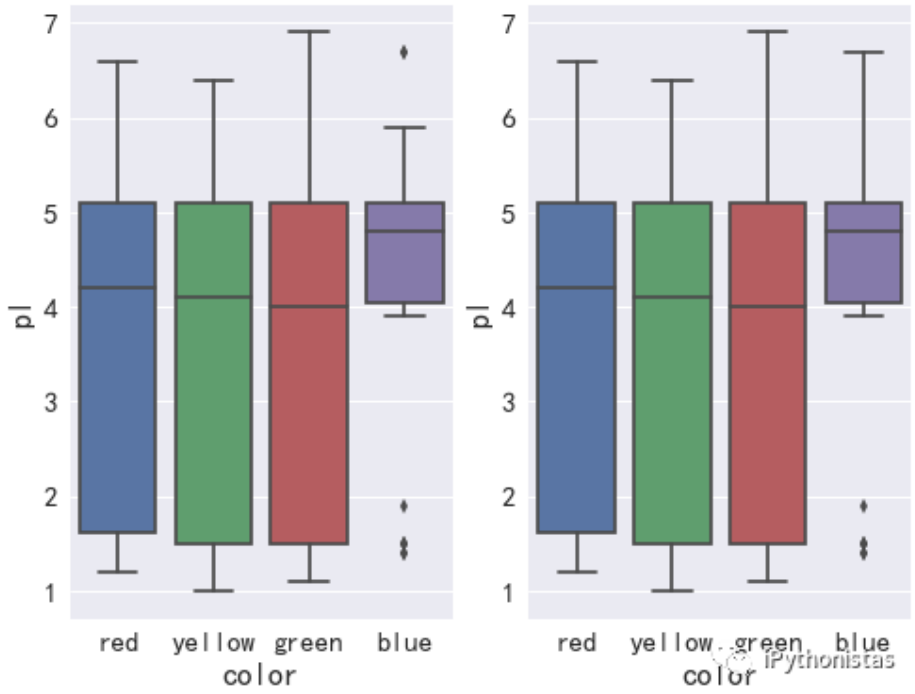
fliersize：float，用于指示离群值观察的**标记大小**

```
fig,axes=plt.subplots(1,2)
sns.boxplot(x="color",y="pl",data=data,ax=axes[0]) #fliersize默认为5
sns.boxplot(x="color",y="pl",data=data,fliersize=20,ax=axes[1])
```



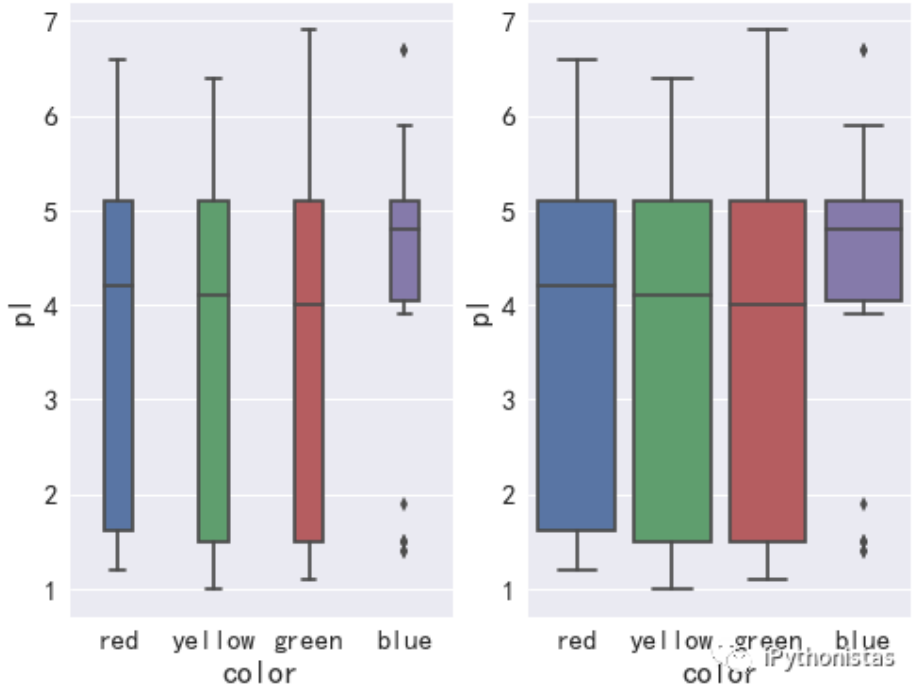
whis：确定离群值的上下界（IQR超过低和高四分位数的比例），此范围之外的点将被识别为异常值。IQR指的是上下四分位的差值。

```
fig,axes=plt.subplots(1,2)
sns.boxplot(x="color",y="pl",data=data,whis=1,ax=axes[0])  #左图
sns.boxplot(x="color",y="pl",data=data,whis=2,ax=axes[1])  #右图
```



width：float，控制箱型图的宽度

```
fig,axes=plt.subplots(1,2)
sns.boxplot(x="color",y="pl",data=data,width=0.3,ax=axes[0])  #左图
sns.boxplot(x="color",y="pl",data=data,width=0.8,ax=axes[1])  #右图
```



---

violinplot

violinplot与boxplot扮演类似的角色，它显示了定量数据在一个（或多个）分类变量的多个层次上的分布，这些分布可以进行比较。不像箱形图中所有绘图组件都对应于**实际数据点**，小提琴绘图以基础分布的**核密度估计**为特征。具体用法如下：

*seaborn.violinplot(x=None, y=None, hue=None, data=None, order=None, hue_order=None, bw='scott', cut=2, scale='area', scale_hue=True, gridsize=100, width=0.8, inner='box', split=False, dodge=True, orient=None, linewidth=None, color=None, palette=None, saturation=0.75, ax=None, **kwargs)*

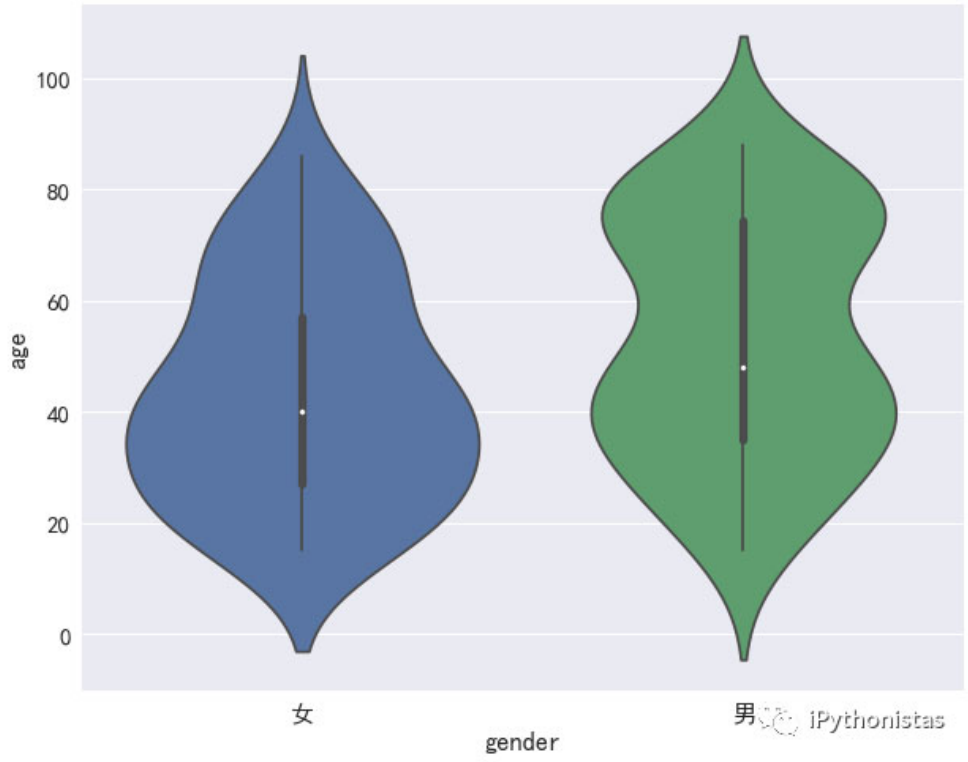
实例所用的数据集如下：

	age	color	gender	height	smoker	weight
0	43	red	女	183	False	63
1	76	red	男	163	False	78
2	42	yellow	女	153	True	55
3	40	white	女	159	False	65
4	25	red	男	175	False	56
5	15	green	女	170	False	69

在这里就不再介绍**x，y，hue，data，order，hue_order，palette**参数的用法，这些参数的用法和之前介绍的图形的用法是一样的，如有需要可以查看之前的内容。

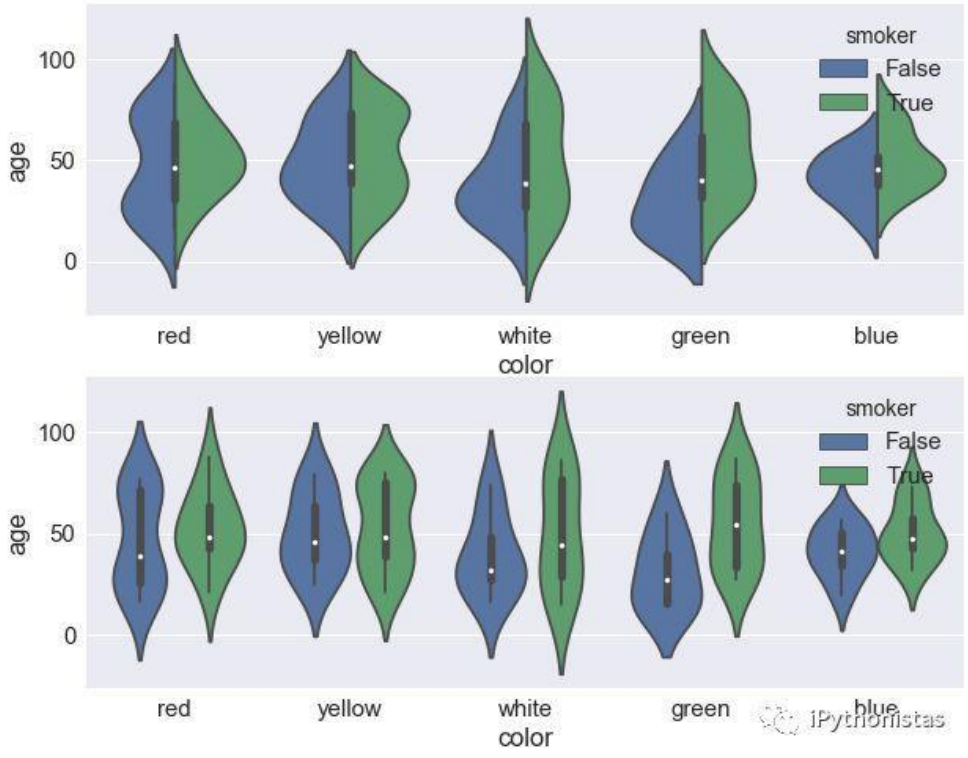
先来画一个小提琴图：

```
sns.violinplot(x="gender",y="age",data=data)
```



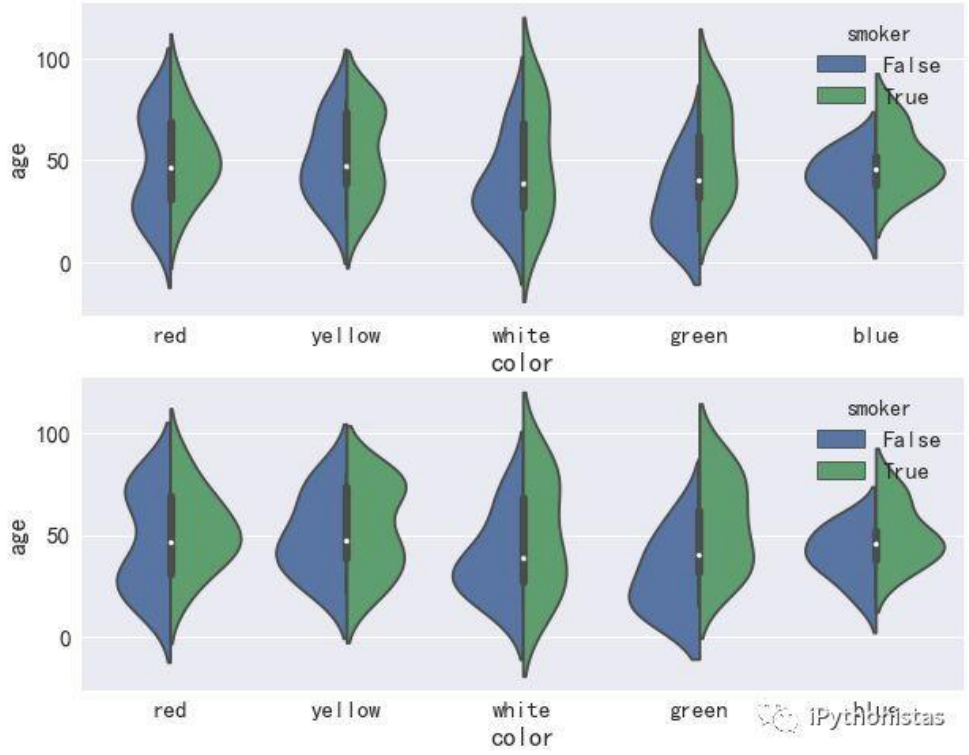
split：将split设置为true则绘制分拆的violinplot以比较经过hue拆分后的两个量：

```
fig,axes=plt.subplots(2,1)
ax=sns.violinplot(x="color",y="age",data=data,hue="smoker",split=True,ax=axes[0])
#上图，拆分后的图
ax=sns.violinplot(x="color",y="age",data=data,hue="smoker",ax=axes[1])  #下图
```



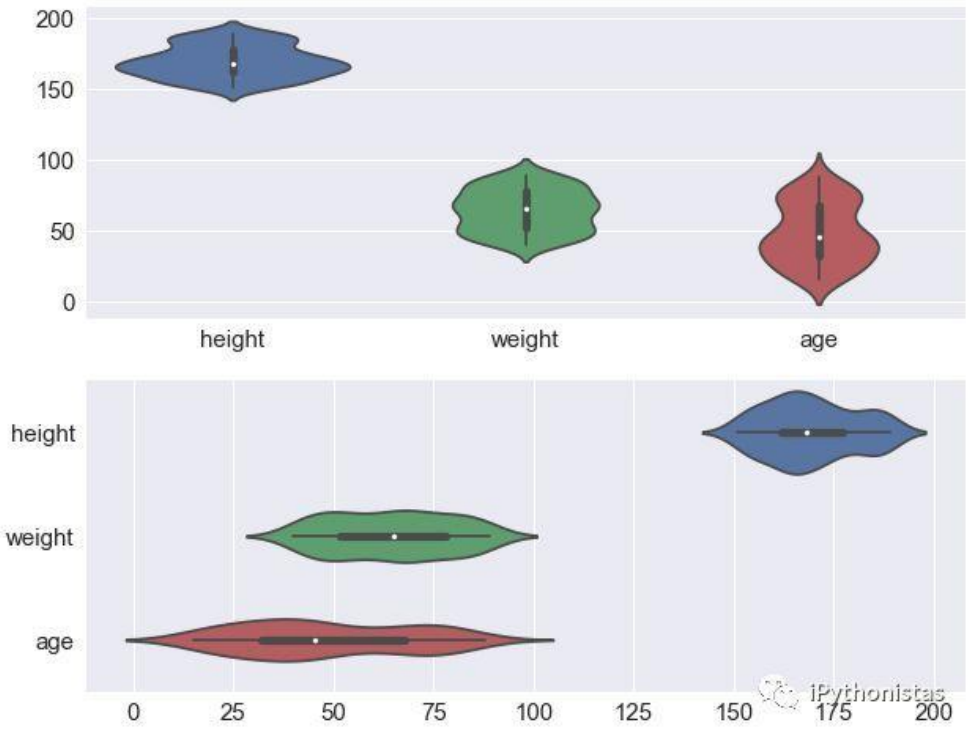
scale_hue：bool，当使用色调变量（hue参数）嵌套小提琴时，此参数确定缩放是在主要**分组变量**（scale_hue = true）的每个级别内还是在图上的**所有小提琴**（scale_hue = false）内计算出来的。

```
fig,axes=plt.subplots(2,1)
ax=sns.violinplot(x="color",y="age",data=data,hue="smoker",split=True,scale_hue=False,ax=axes[
 #上图
ax=sns.violinplot(x="color",y="age",data=data,hue="smoker",split=True,scale_hue=True,ax=axes[1
  #下图
```



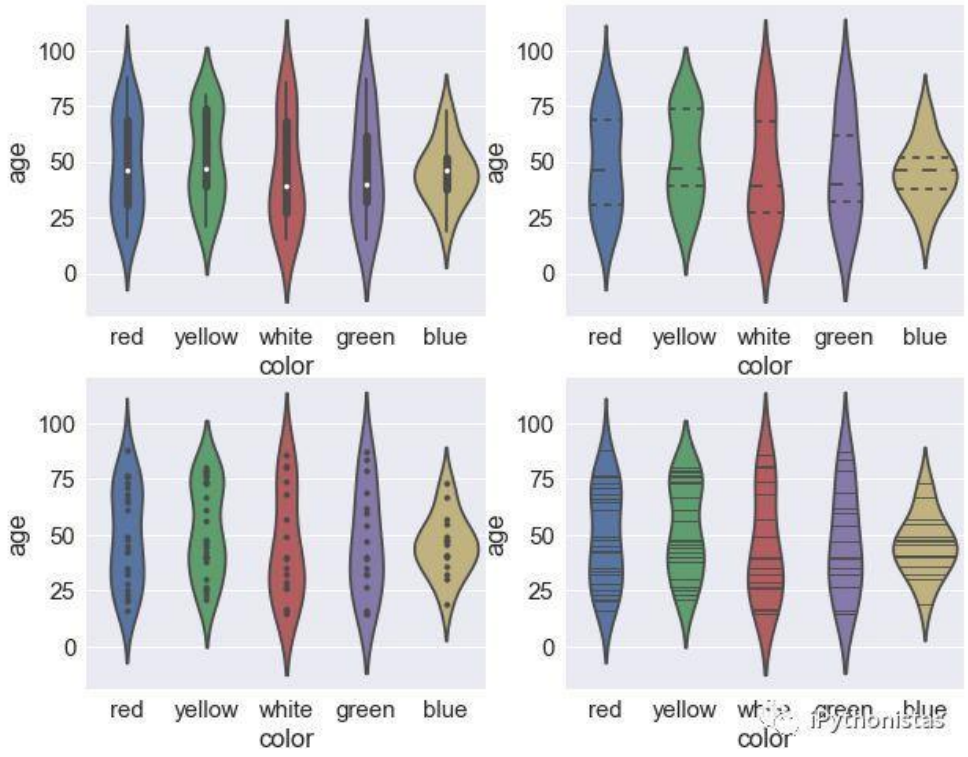
orient："v"|"h" 用于控制图像使水平还是竖直显示（这通常是从输入变量的dtype推断出来的，此参数一般当**不传入x、y，只传入data的时候**使用）

```
fig,axes=plt.subplots(2,1)
sns.violinplot(data=data[["height","weight","age"]],orient="v",ax=axes[0]) #上图
sns.violinplot(data=data[["height","weight","age"]],orient="h",ax=axes[1]) #下图
```



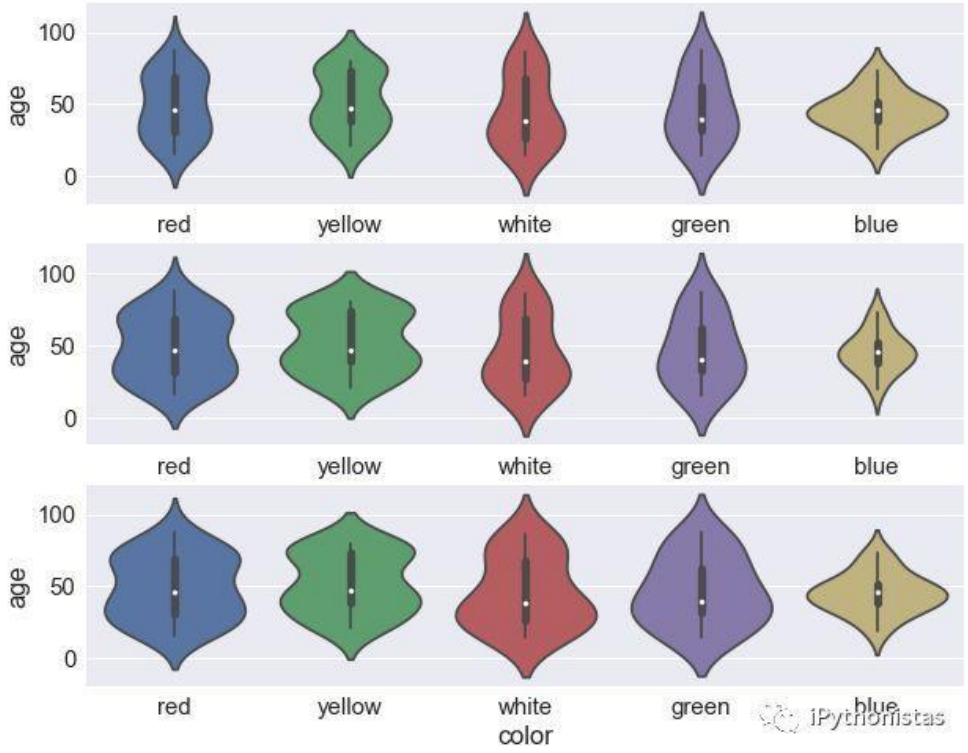
inner：控制violinplot**内部数据点**的表示，有**“box”**, **“quartile”**, **“point”**, **“stick”**四种方式。

```
fig,axes=plt.subplots(2,2)
sns.violinplot(x="color",y="age",data=data,inner="box",ax=axes[0,0])  #钢琴图内显示箱型图（左上）
sns.violinplot(x="color",y="age",data=data,inner="quartile",ax=axes[0,1])  #钢琴图内显示四分位数线（右上）
sns.violinplot(x="color",y="age",data=data,inner="point",ax=axes[1,0])  #钢琴图内显示具体数据点（左下）
sns.violinplot(x="color",y="age",data=data,inner="stick",ax=axes[1,1])  #钢琴图内显示具体数据棒（右下）
```



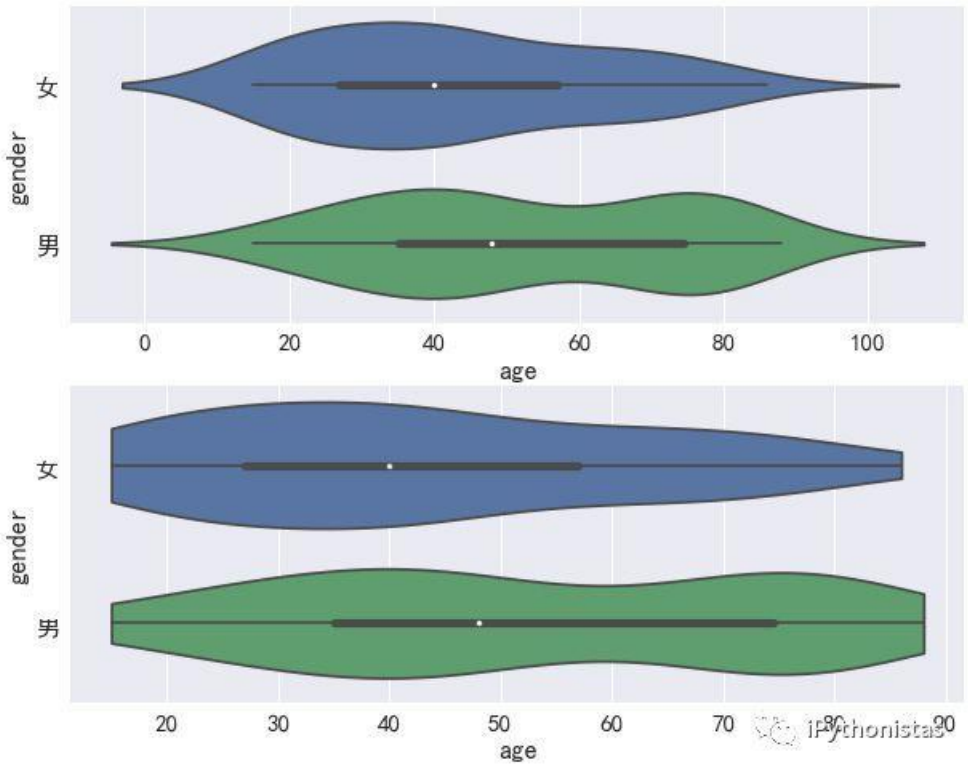
scale：该参数用于缩放每把小提琴的宽度，有**“area”**, **“count”**, **“width”**三种方式

```
fig,axes=plt.subplots(3,1)
sns.violinplot(x="color",y="age",data=data,scale="area",ax=axes[0]) #如果为"area"，每把小提琴将有相同的面积(上图)
sns.violinplot(x="color",y="age",data=data,scale="count",ax=axes[1])  #如果为"count"，小提琴的宽度将根据该小组中观察的数量来缩放（中图）
sns.violinplot(x="color",y="age",data=data,scale="width",ax=axes[2])  #如果为"age"，每把小提琴将有相同的宽度（下图）
```



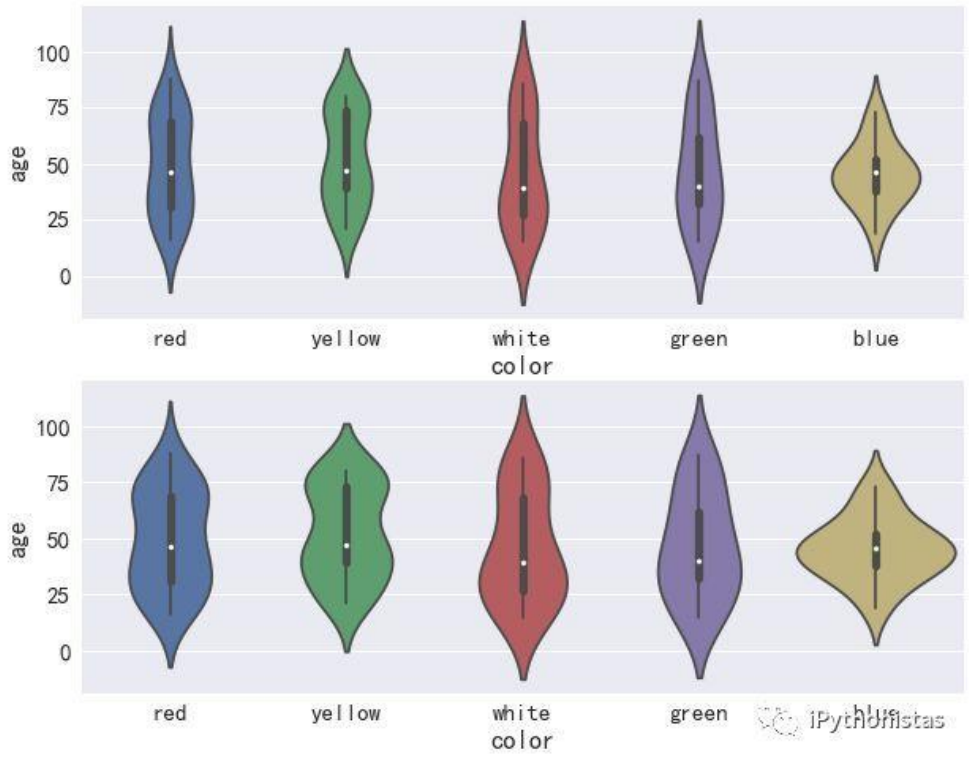
cut：float，距离，以带宽大小为单位，以控制小提琴图外壳延伸超过内部极端数据点的密度。设置为0以将小提琴范围限制在观察数据的范围内（即，在ggplot中具有与trim = true相同的效果）

```
fig,axes=plt.subplots(2,1)
sns.violinplot(x="age",y="gender",data=data,ax=axes[0]) #上图
sns.violinplot(x="age",y="gender",data=data,cut=0,ax=axes[1])  #下图
```



width：float，控制钢琴图的宽度（比例）

```
fig,axes=plt.subplots(2,1)
sns.violinplot(x="color",y="age",data=data,ax=axes[0],width=0.5)  #上图
sns.violinplot(x="color",y="age",data=data,ax=axes[1],width=0.9)  #下图
```



这已经是Seaborn入门系列的第三篇文章了，相信大家已经大概了解Seaborn的作图过程，也可以体会到用Seaborn作图相比于matplotlib更加简单。以上内容是我结合官方文档和自己的一点理解写成的，有什么错误大家可以**指出来并提提意见**，**共同交流、进步**，也希望我写的这些能够给阅读完本文的你或或少的帮助！