# seaborn可视化(2) ——分类

| 类型 | 函数 | 说明 |
| --- | --- | --- |
| 散点图 | **catplot**(x,y,data) 默认kind = 'scatter' | catplot()默认为散点图 |
| | catplot(x,y,data,**kind = 'strip'**) | 画一个散点图，其中一个变量是分类的 |
| | catplot(x,y,data,**kind = 'swarm'**) | 画一个没有重叠点的分类散点图 |
| 分布图 | catplot(x,y,data,**kind = 'box'**) 箱型图 | 画一个箱型图来显示与类别相关的分布 |
| | catplot(x,y,data,**kind = 'boxen'**) 改进箱型图 | 为较大的数据集绘制增强的箱型图 |
| | catplot(x,y,data,**kind = 'violin'**) 小提琴图 | 绘制箱线图和核密度估计的组合 |
| 其他 | catplot(x,y,data,**kind = 'bar'**) 条形图 | 将点估计和置信区间显示为矩形条 |
| | catplot(x,y,data,**kind = 'count'**) 条形计数图 | 使用条形图显示每个分类库中的观察计数 |
| | catplot(x,y,data,**kind = 'point'**) 点线图 | 使用散点图符号显示点估计和置信区间 |

全文： http://seaborn.pydata.org/tutorial/categorical.html#

stripplot() : http://seaborn.pydata.org/generated/seaborn.stripplot.html#seaborn.stripplot

swarmplot(): http://seaborn.pydata.org/generated/seaborn.swarmplot.html#seaborn.swarmplot

boxplot(): http://seaborn.pydata.org/generated/seaborn.boxplot.html#seaborn.boxplot

violinplot(): http://seaborn.pydata.org/generated/seaborn.violinplot.html#seaborn.violinplot

boxenplot()： http://seaborn.pydata.org/generated/seaborn.boxenplot.html#seaborn.boxenplot

pointplot(): http://seaborn.pydata.org/generated/seaborn.pointplot.html#seaborn.pointplot

barplot(): http://seaborn.pydata.org/generated/seaborn.barplot.html#seaborn.barplot

countplot(): http://seaborn.pydata.org/generated/seaborn.countplot.html#seaborn.countplot

---

## 散点图分类

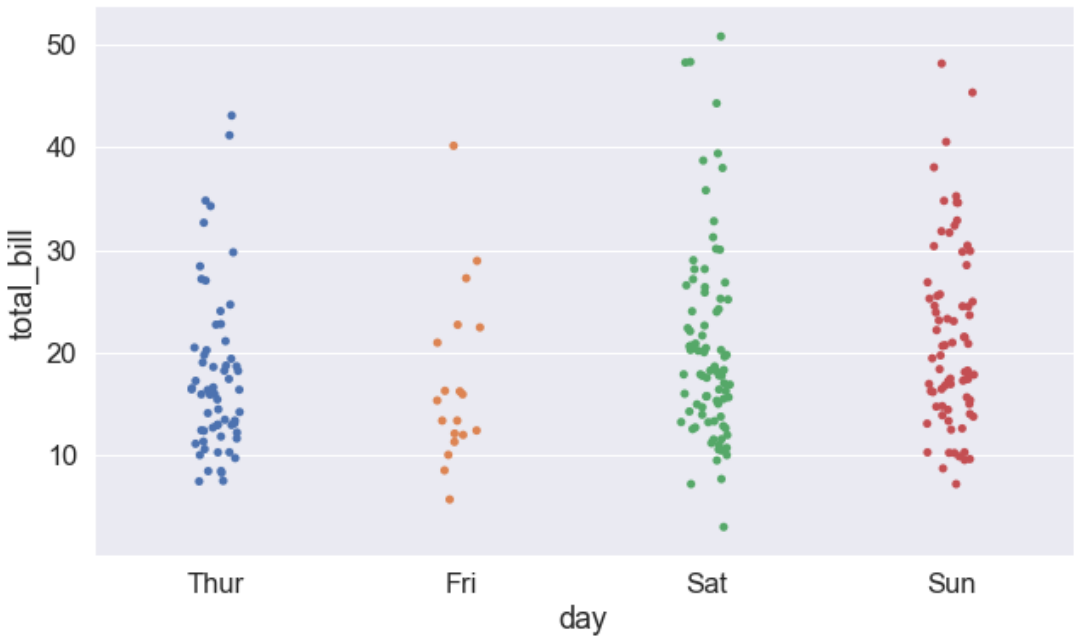
- **catplot() or warmplot() : 散点图可视化分类变量**

```
sns.catplot(
    ['x=None', 'y=None', 'hue=None', 'data=None', 'row=None', 'col=None',
    'col_wrap=None', 'estimator=<function mean at 0x000001A2D75D2950>',
```

```
    'ci=95', 'n_boot=1000', 'units=None', 'order=None', 'hue_order=None',
    'row_order=None', 'col_order=None', "kind='strip'", 'height=5',
    'aspect=1', 'orient=None', 'color=None', 'palette=None', 'legend=True',
    'legend_out=True', 'sharex=True', 'sharey=True', 'margin_titles=False',
    'facet_kws=None', '**kwargs'],)

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="ticks", color_codes=True)

#使用seaborn自带的小费数据集 tips
tips = sns.load_dataset('tips')
tips.head()
Out[42]:
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female     No  Sun  Dinner     4
```

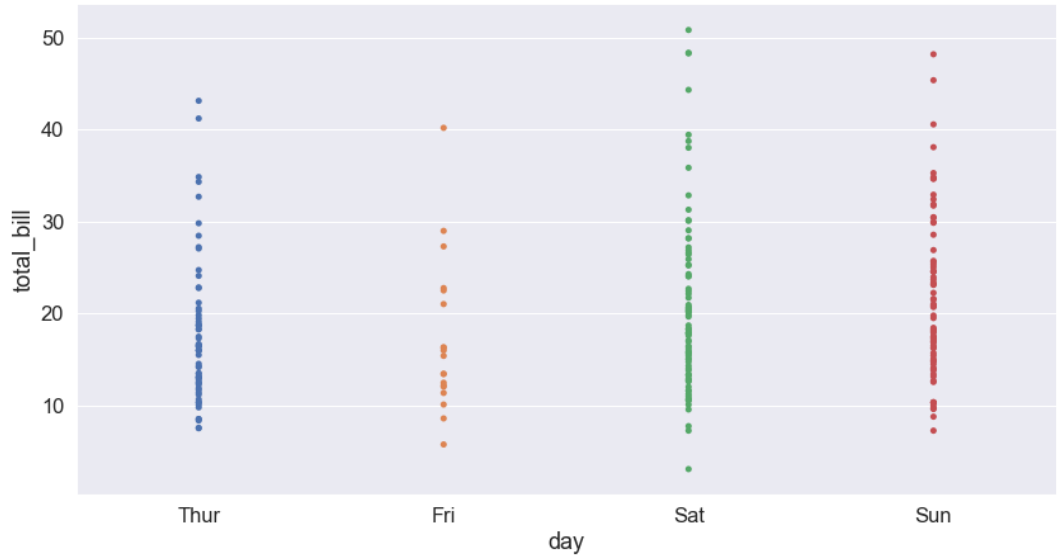
**1:最基本的情况:**

```
sns.catplot(x="day", y="total_bill", data=tips)
```



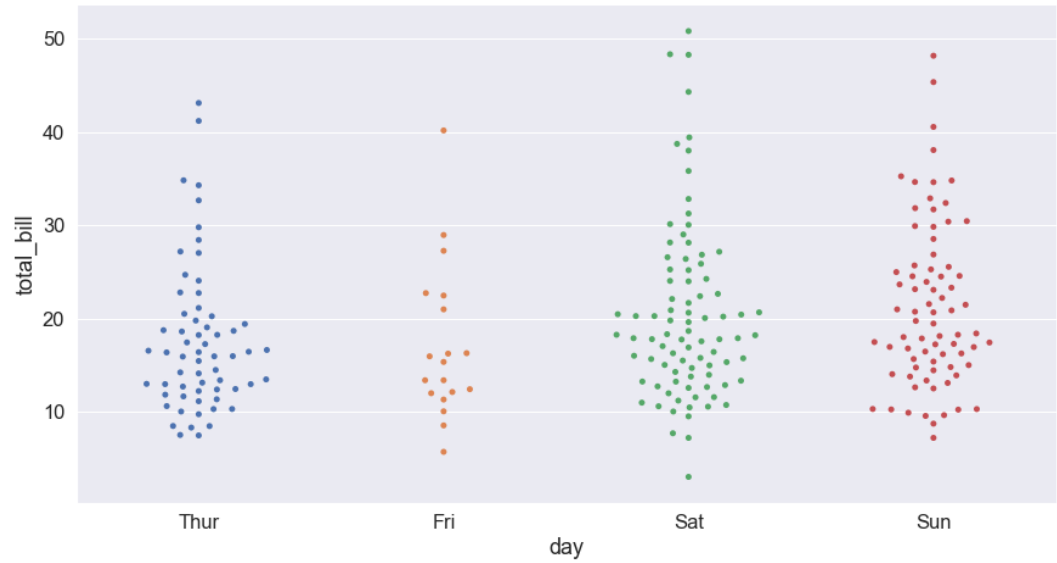

**2. jitter：控制数据抖动**

```
#jitter : 控制数据抖动
sns.catplot(x="day", y="total_bill", jitter=False, data=tips)
```



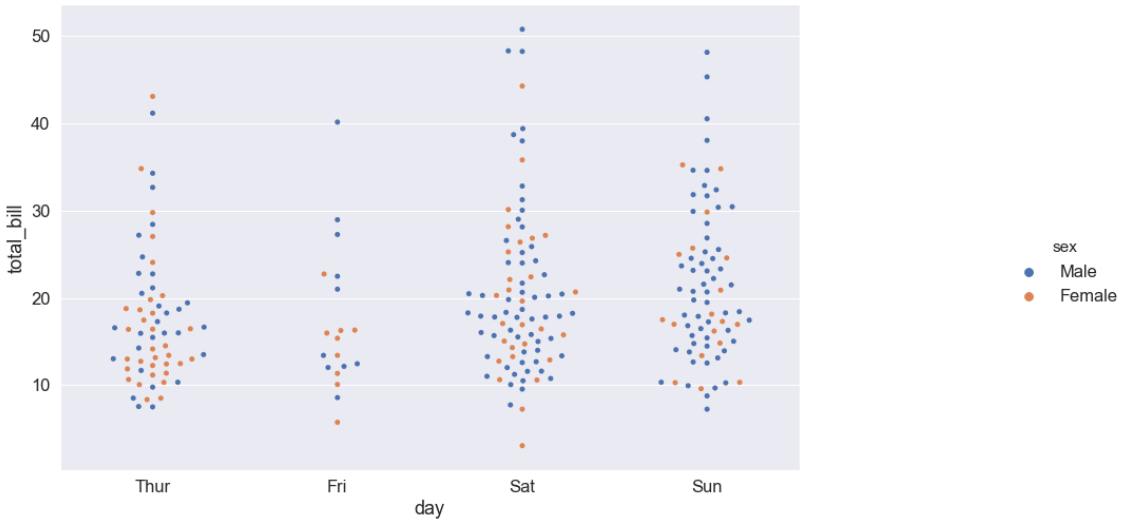

### 3. swarmplot()

```
"""
第二种方法使用防止重叠的算法沿分类轴调整点。虽然它只适用于相对较小的数据集，
但它可以更好地表示观测值的分布。这种情节有时被称为“beeswarm”，
由swarmplot()在seaborn中绘制，在catplot()中设置kind=“swarm”激活swarmplot()
"""
sns.catplot(x="day", y="total_bill", kind="swarm", data=tips)
```



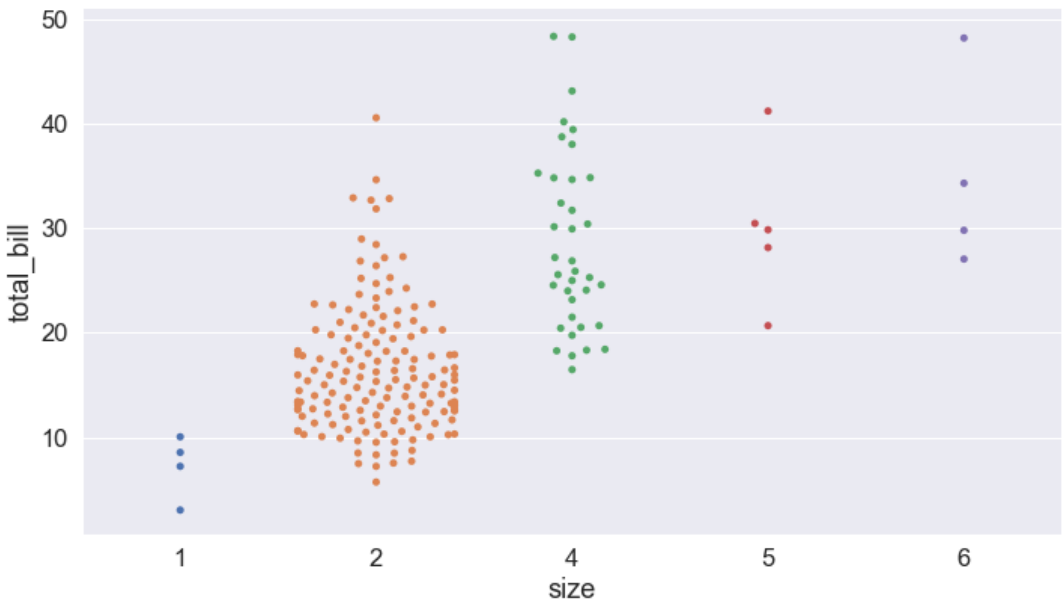

### 4. hue：类内分组

```
#hue ： 类内分组
sns.catplot(x="day", y="total_bill", hue="sex", kind="swarm", data=tips)
```



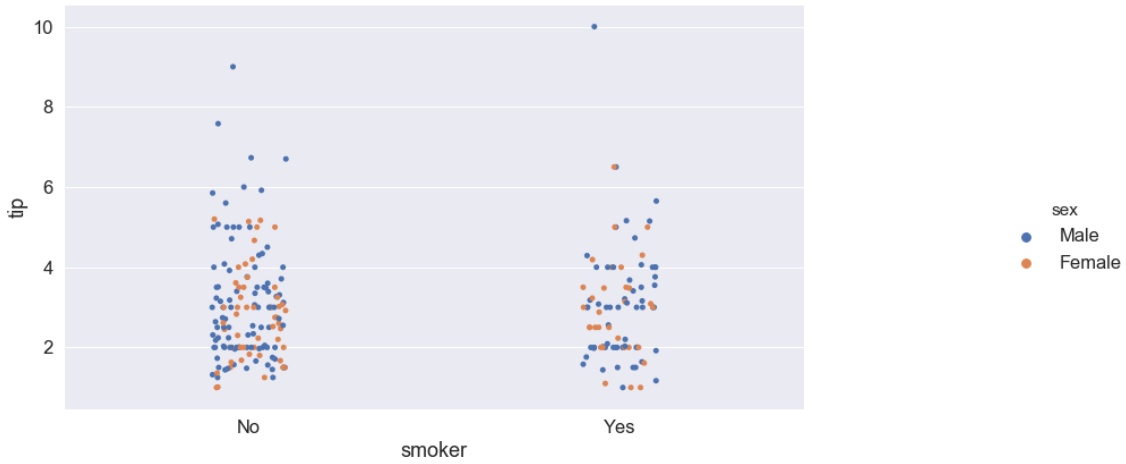

**5.筛选数据**

```
#筛选数据
sns.catplot(x="size", y="total_bill", kind="swarm",
            data=tips.query("size != 3"))
```



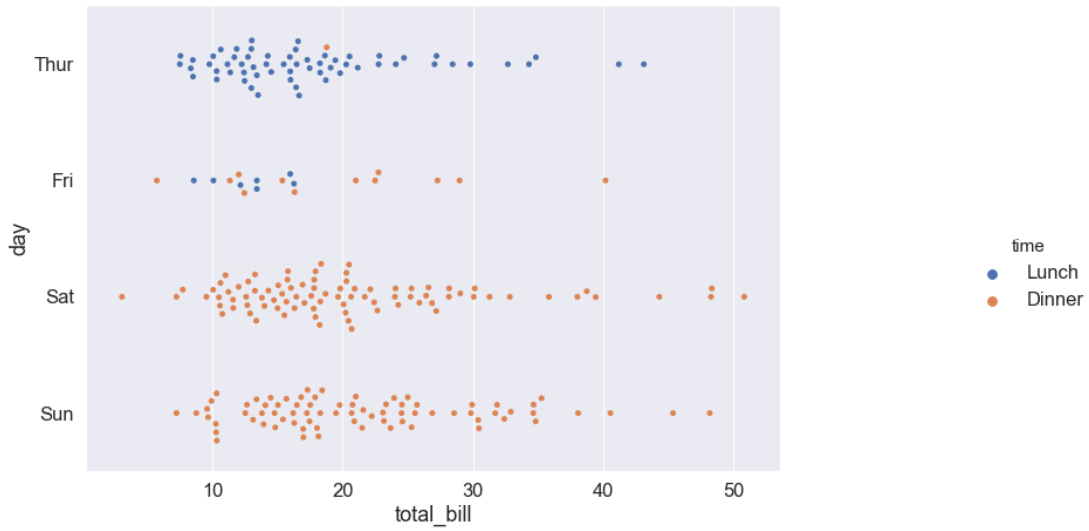

**6.order: 指定顺序**

```
#指定顺序
sns.catplot(x="smoker", y="tip", order=["No", "Yes"], data=tips)sns.relplot(x="total_bill", y="tip", size="size", sizes=(15, 200), data=tips)
```




**7.轴序**

```
#轴序
sns.catplot(x="total_bill", y="day", hue="time", kind="swarm", data=tips)
```




---

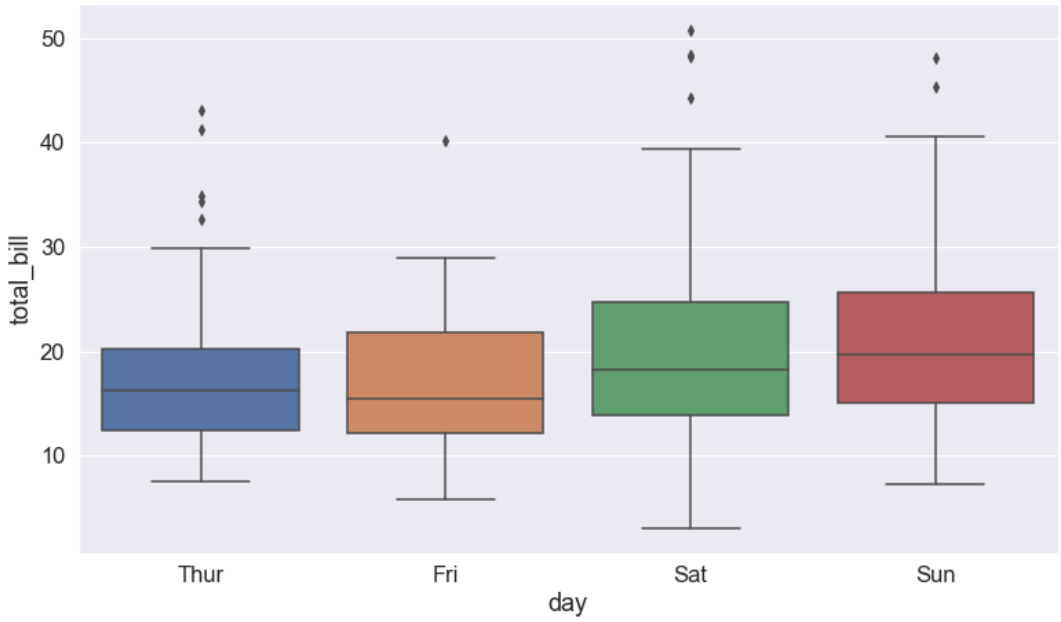
## 分类数据的分布

- **箱型图：boxplot（）画一个方框图来显示与类别相关的分布**

```
sns.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'],)
or
sns.catplot(kind = 'box')
```

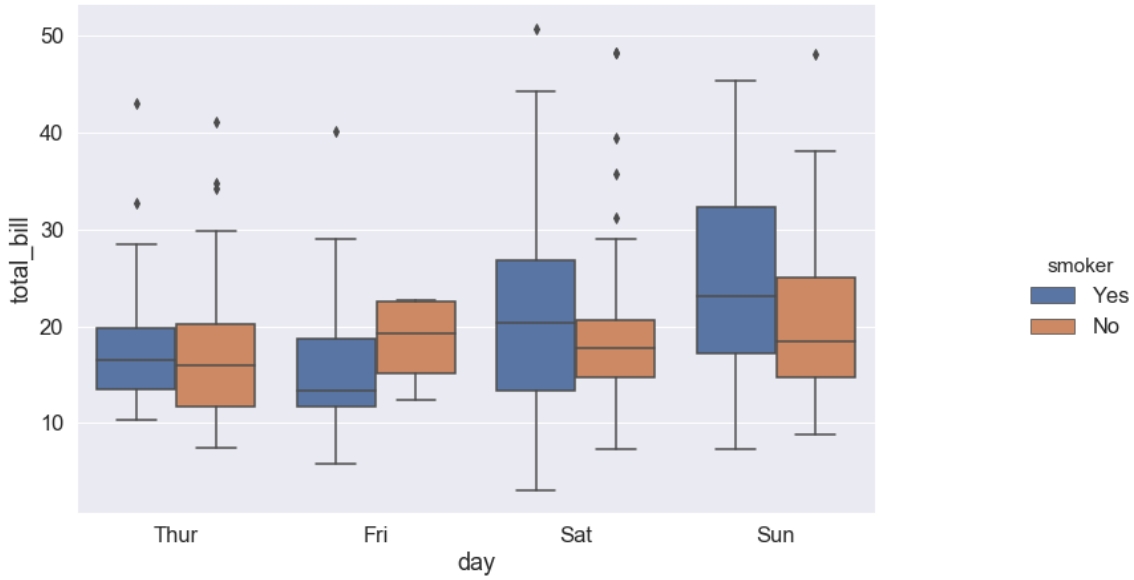
**1.默认情况**

```
#箱型图
sns.catplot(x="day", y="total_bill", kind="box", data=tips)
or
sns.boxplot(x="day", y="total_bill", data=tips)
```



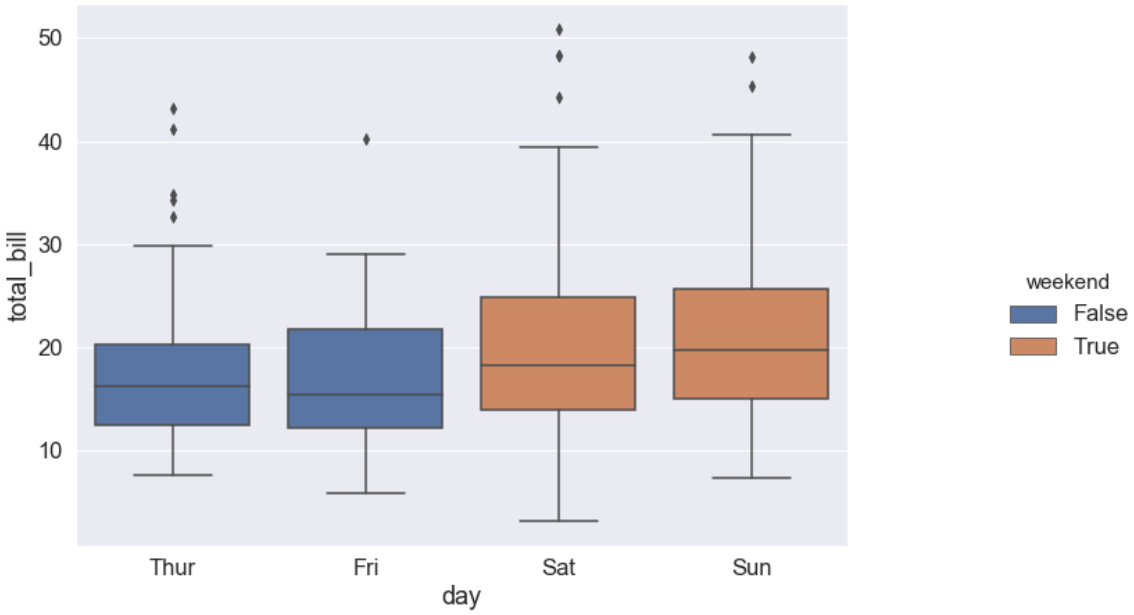

**2. 类内分组**

```
# hue : 类内分组
sns.catplot(x="day", y="total_bill", hue="smoker", kind="box", data=tips)
```



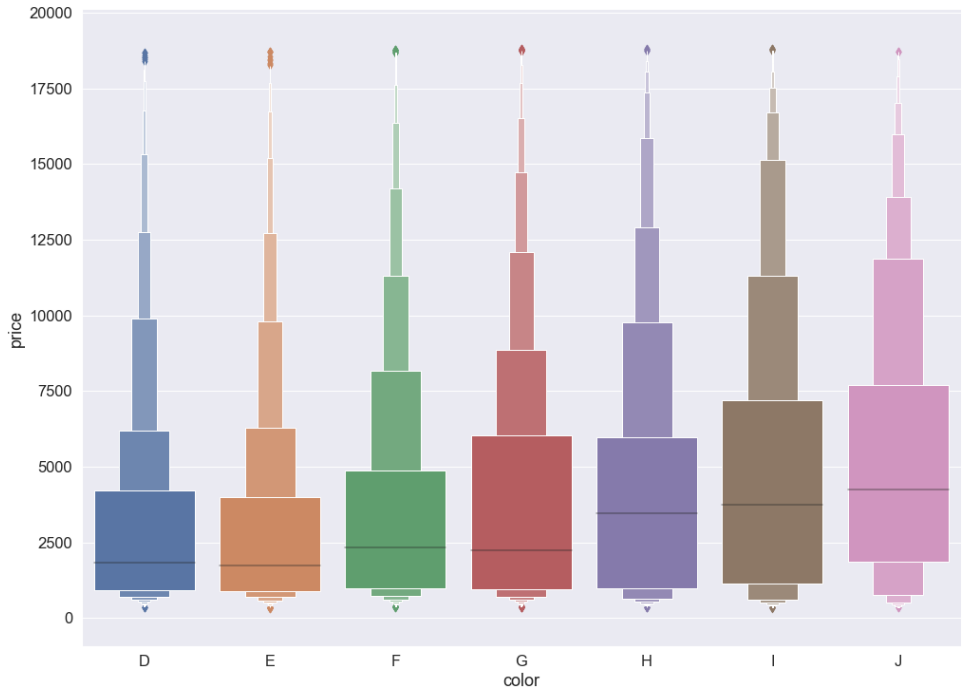

### 3.自定义分组

```
# 指定分组
tips["weekend"] = tips["day"].isin(["Sat", "Sun"])
sns.catplot(x="day", y="total_bill", hue="weekend",
            kind="box", dodge=False, data=tips)
```



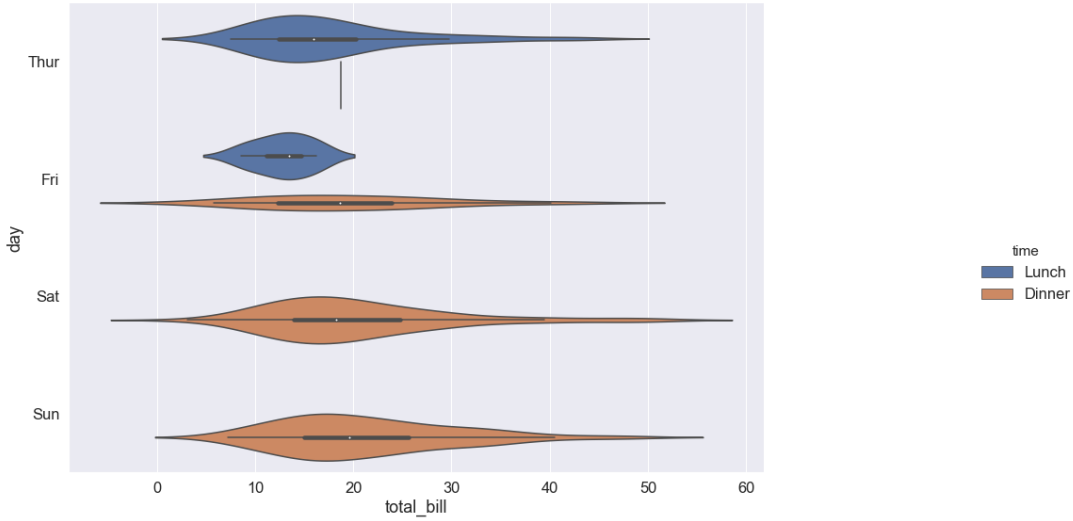

### 4.boxenplot()

```
"""
一个相关的函数boxenplot()绘制了一个类似于箱形图的图，但是经过了优化，
可以显示关于分布形状的更多信息。它最适合较大的数据集：
"""
diamonds = sns.load_dataset("diamonds")
sns.catplot(x="color", y="price", kind="boxen",
            data=diamonds.sort_values("color"))
```



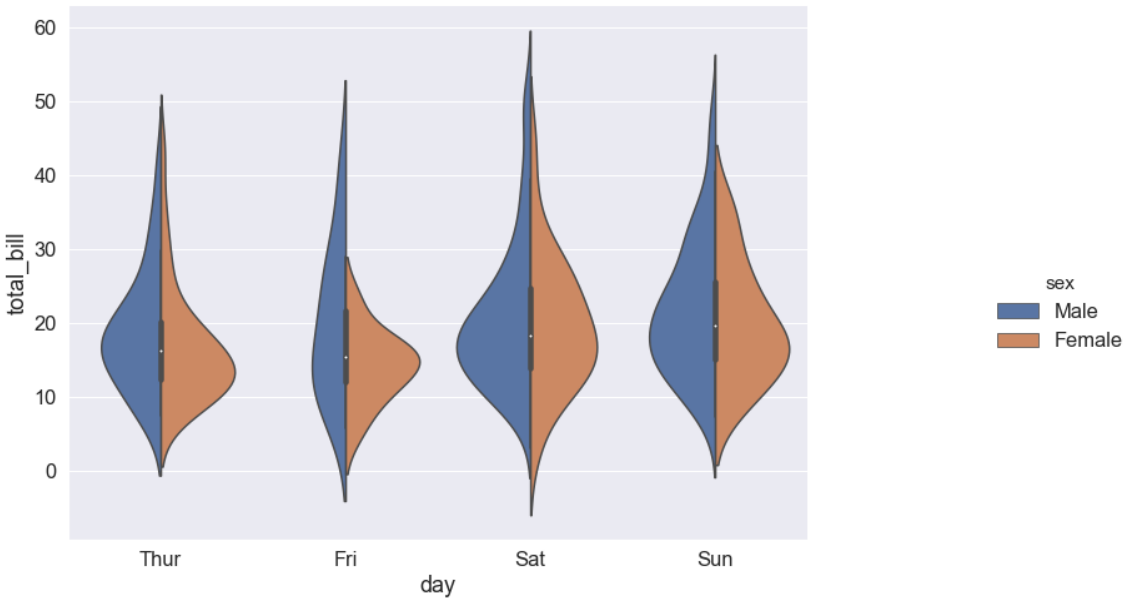

**5.Violinplot()**

```
#小提琴图：结合了箱线图和分布教程中描述的内核密度估计过程
sns.catplot(x="total_bill", y="day", hue="time",
            kind="violin", data=tips)
```



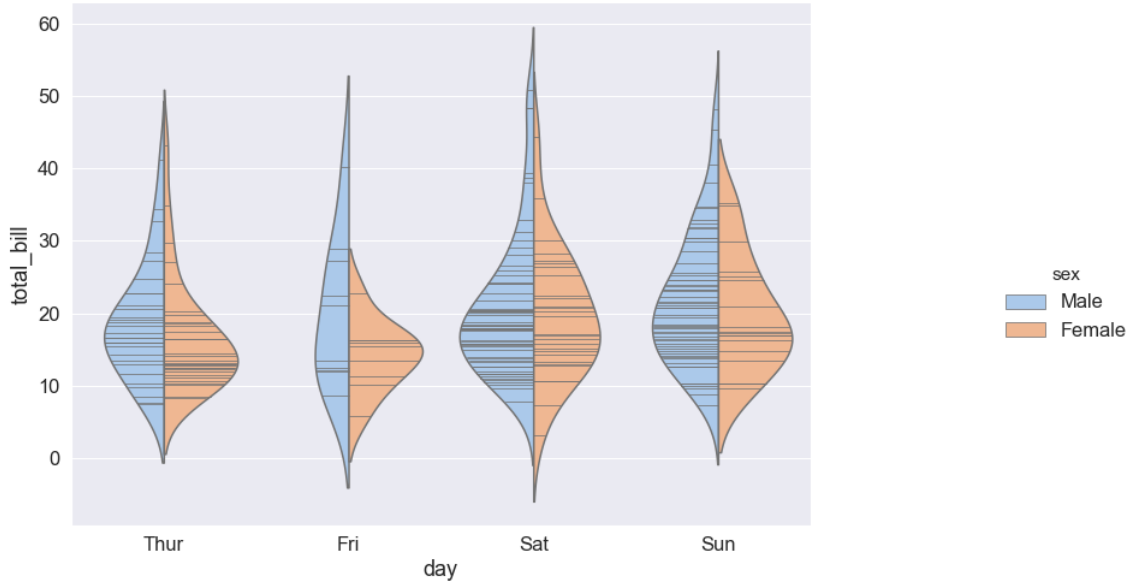

### 6. 分割小提琴图

```
#分割小提琴图
sns.catplot(x="day", y="total_bill", hue="sex",
            kind="violin", split=True, data=tips)
```



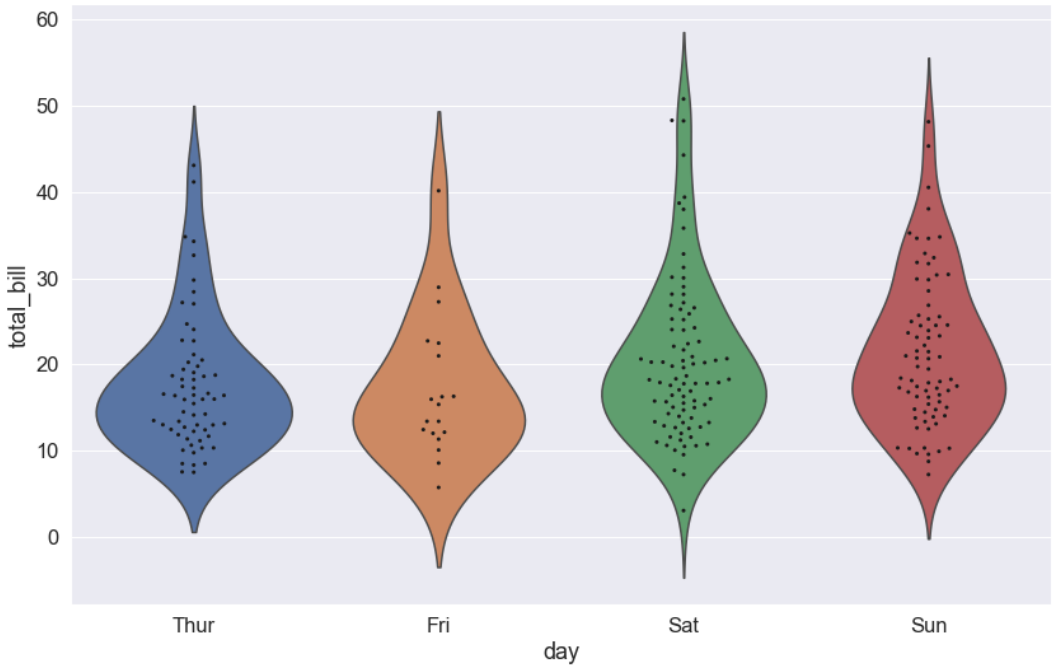

### 7. 显示每个单独数据的观察值

```
#显示每个单独数据的观察值
sns.catplot(x="day", y="total_bill", hue="sex",
            kind="violin", inner="stick", split=True,
            palette="pastel", data=tips)
```




### 8. 结合散点图和小提琴图

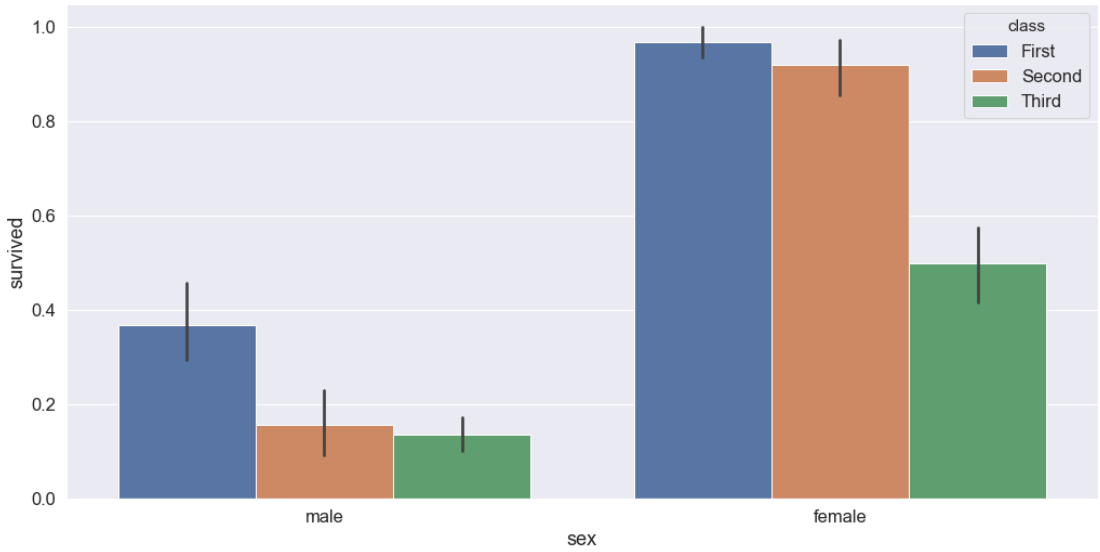
```
#结合散点图和小提琴图
g = sns.catplot(x="day", y="total_bill", kind="violin", inner=None, data=tips)
sns.swarmplot(x="day", y="total_bill", color="k", size=3, data=tips, ax=g.ax)
```




## 类别内统计估计

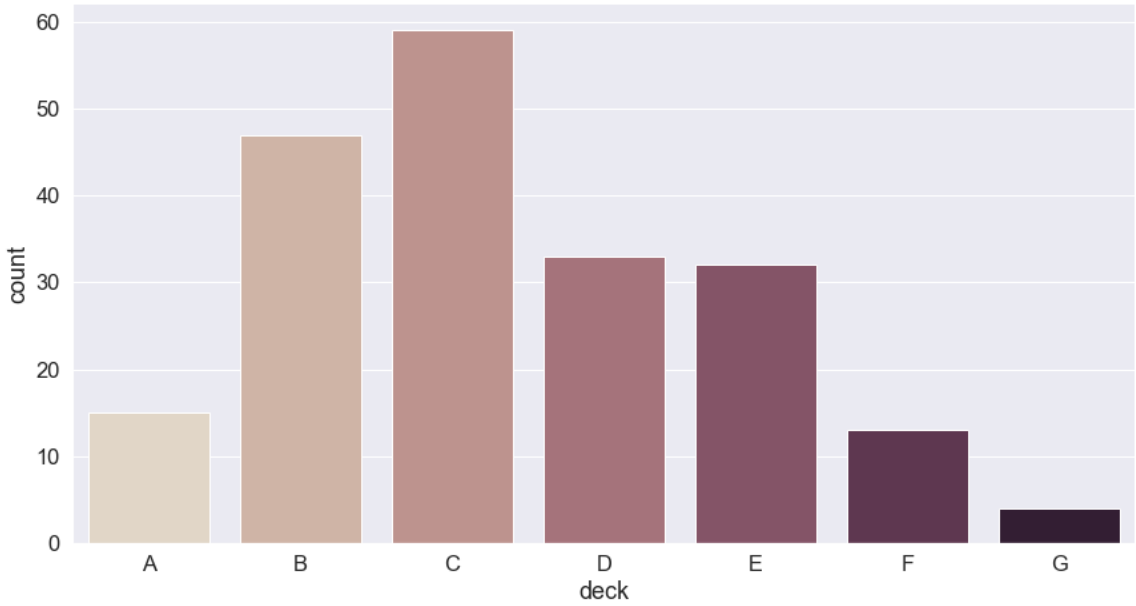
### 1.barplot()

```
titanic = sns.load_dataset("titanic")
titanic.head(5)
Out[140]:
   survived  pclass     sex   age  sibsp  parch     fare embarked  class  \
0         0       3    male  22.0      1      0   7.2500        S  Third
1         1       1  female  38.0      1      0  71.2833        C  First
2         1       3  female  26.0      0      0   7.9250        S  Third
3         1       1  female  35.0      1      0  53.1000        S  First
4         0       3    male  35.0      0      0   8.0500        S  Third

     who  adult_male deck  embark_town alive  alone
0    man        True  NaN  Southampton    no  False
1  woman       False    C    Cherbourg   yes  False
2  woman       False  NaN  Southampton   yes   True
3  woman       False    C  Southampton   yes  False
4    man        True  NaN  Southampton    no   True

sns.catplot(x="sex", y="survived", hue="class", kind="bar", data=titanic)
or
sns.barplot(x = 'sex',y = 'survived',hue = 'class',data = titanic)
```



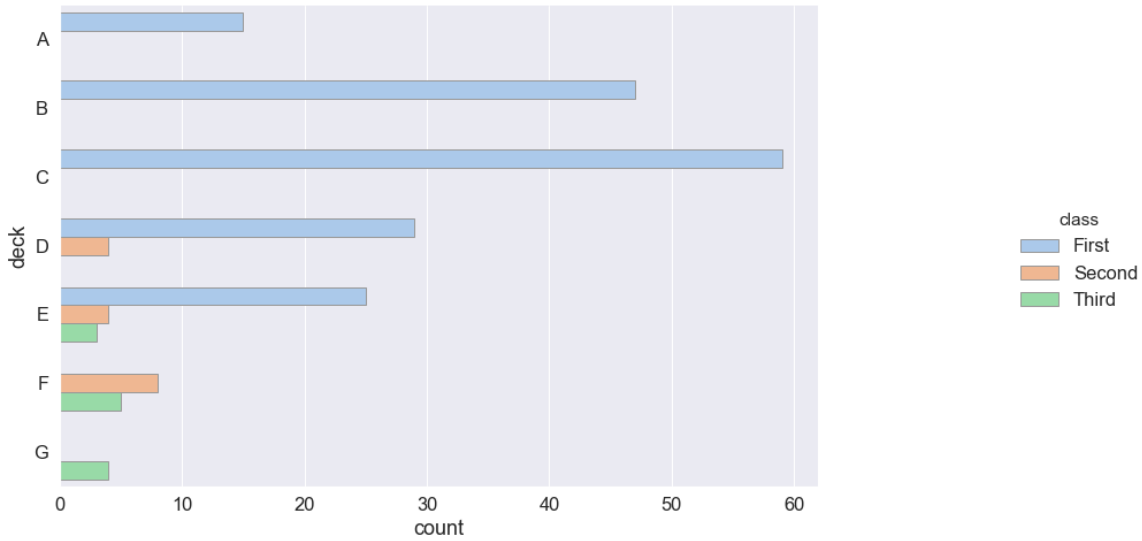

### 2. countplot()

```
#countplot() :使用条形图显示每个分类库中的观察计数
sns.catplot(x="deck", kind="count", palette="ch:.25", data=titanic)
```



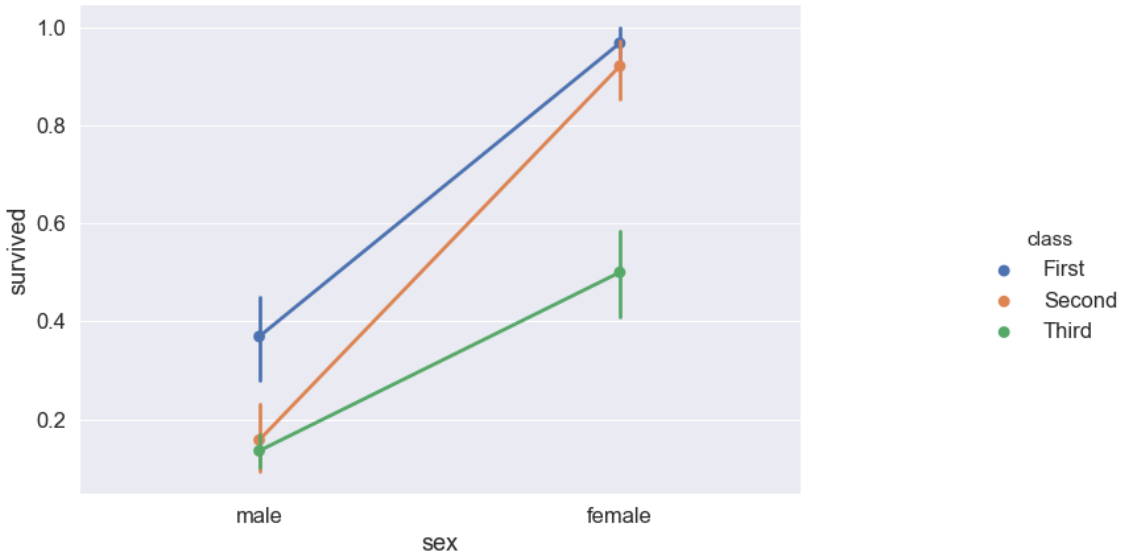

### 3. 条形分组计数

```
#计数分组
sns.catplot(y="deck", hue="class", kind="count",
            palette="pastel", edgecolor=".6",
            data=titanic)
```




### 4.pointplot()

```
#pointplot :使用散点图符号显示点估计和置信区间
sns.catplot(x="sex", y="survived", hue="class", kind="point", data=titanic)
```


1.0
0.8
0.6
0.4
0.2
survived
male
female
sex
class
First
Second
Third