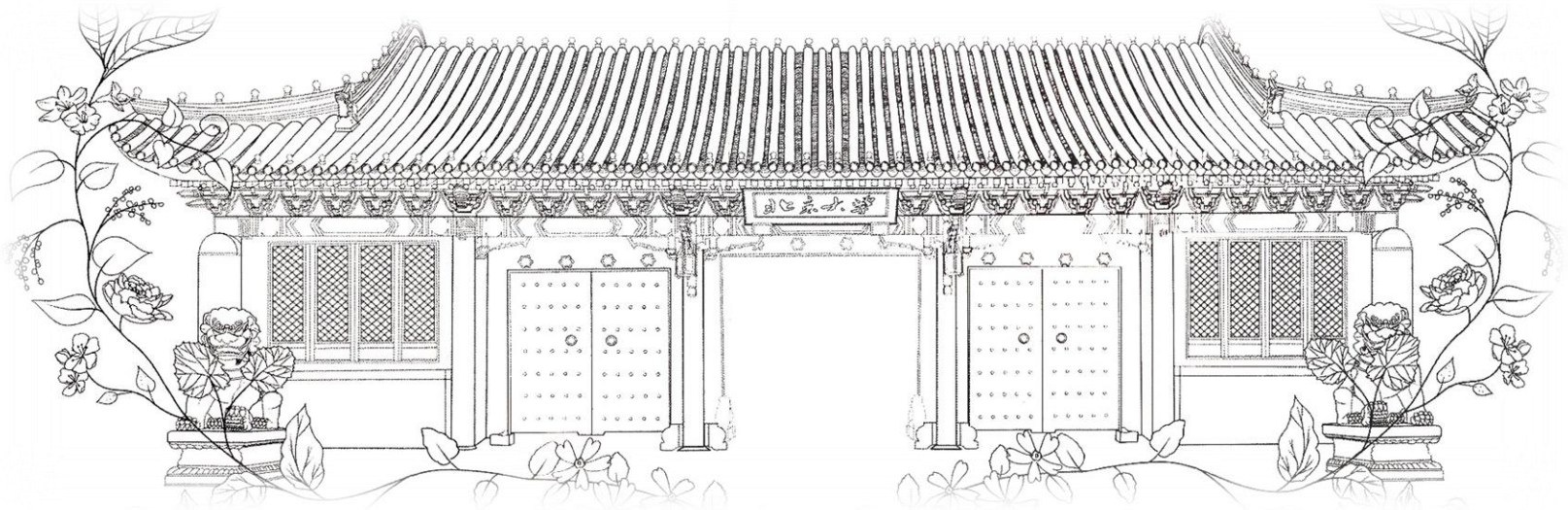
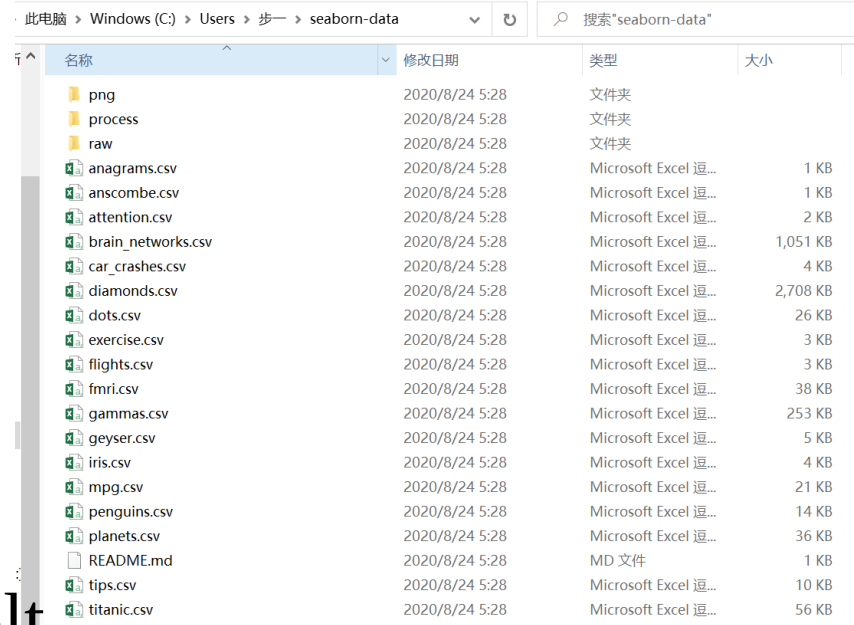
第2章 数据可视化 II



seaborn包



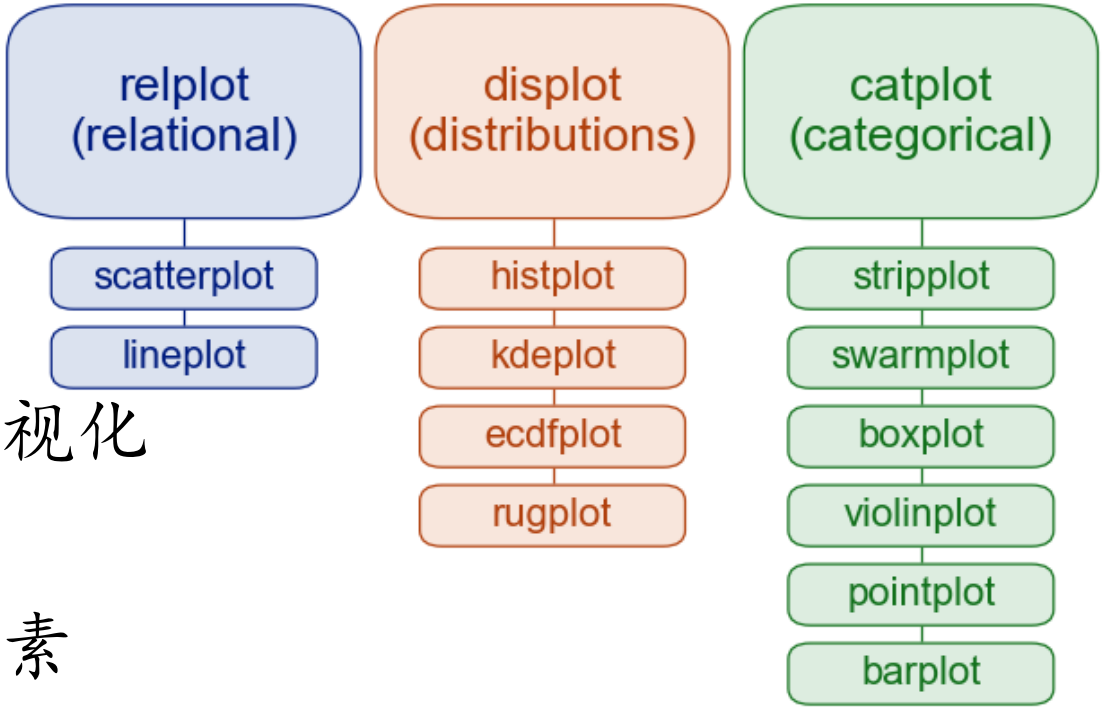
- import numpy as np
- import pandas as pd
- import matplotlib.pyplot as plt
- import seaborn as sns
- sns.set_theme(style="darkgrid")
- 数据集https://github.com/mwaskom/seaborn-data
  需要解压后放到本地的seaborn-data文件夹内
- Gallery:
  - http://seaborn.pydata.org/examples/index.html

注意

- 以下代码基于seaborn的0.11版本…

目录

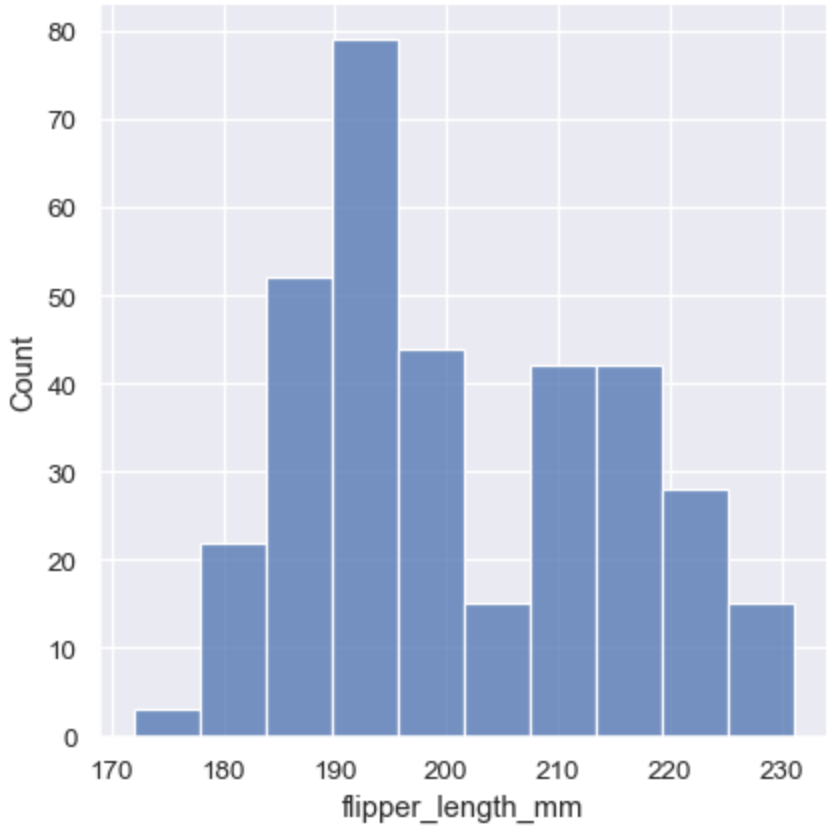
- 单变量分布可视化
- 多变量间关系的可视化
- 定类变量的可视化
- 可视化中的美学因素



单变量分布的可视化

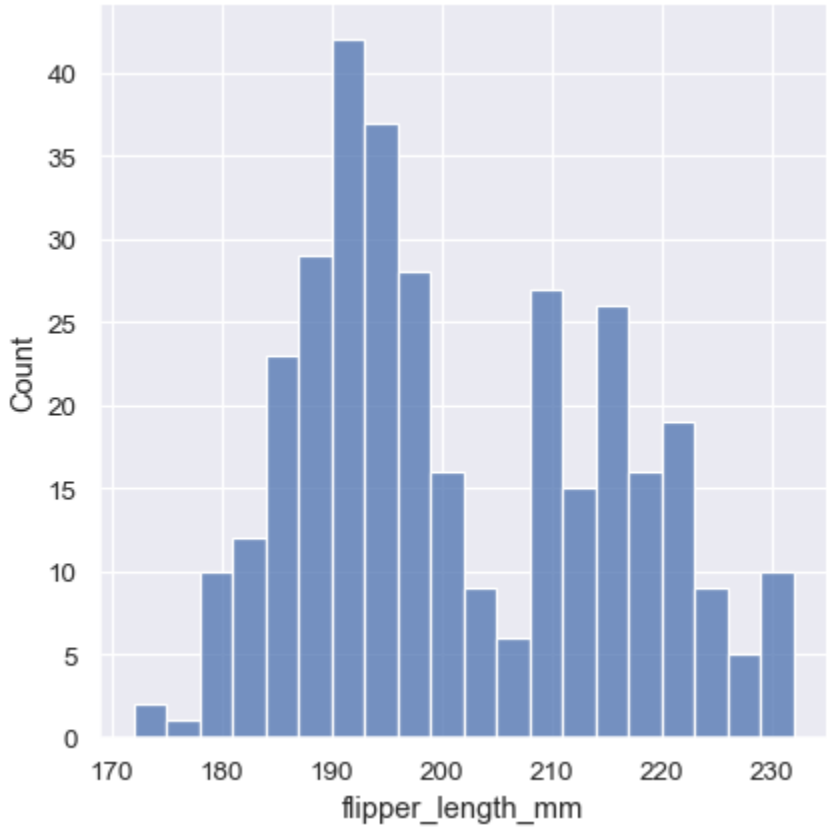
单变量直方图

- penguins = sns.load_dataset("penguins")
- sns.displot(penguins, x="flipper_length_mm")



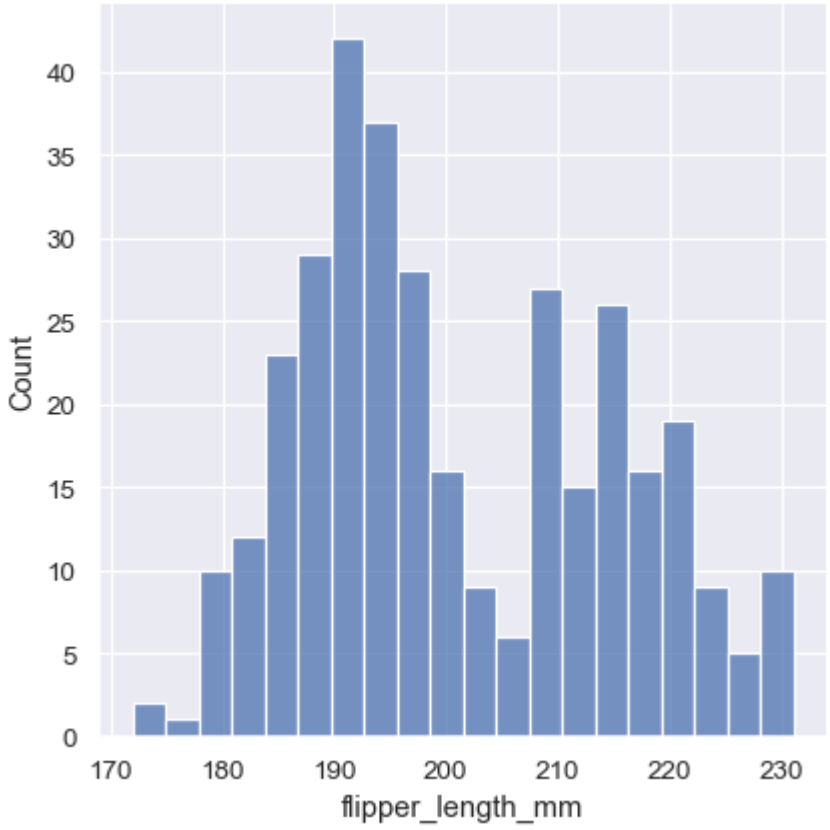
单变量直方图

- sns.displot(penguins, x="flipper_length_mm", binwidth=3)



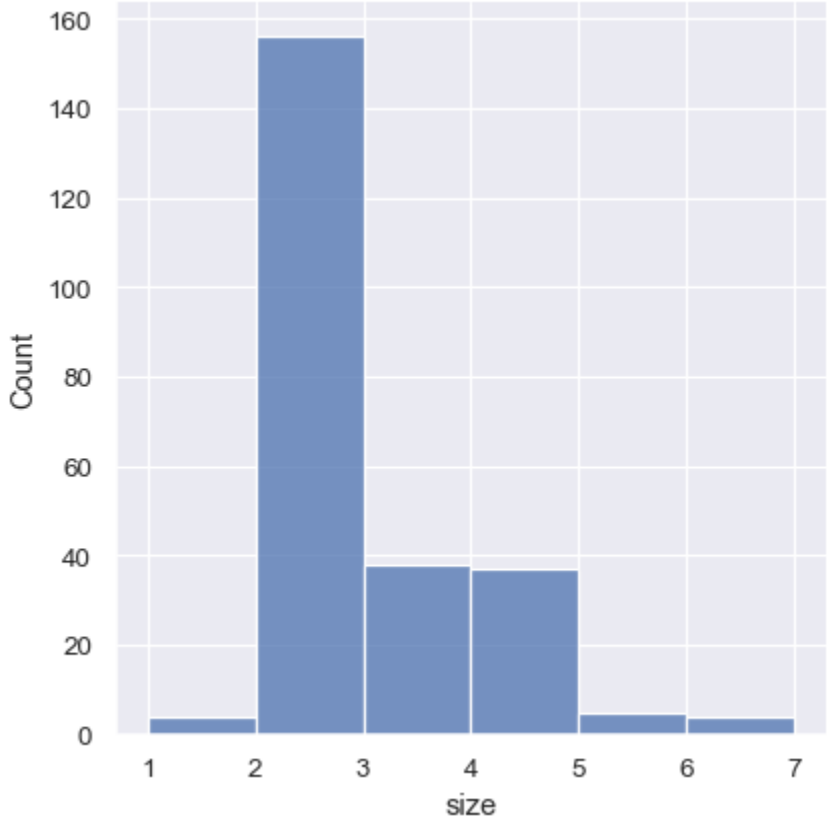
单变量直方图

- sns.displot(penguins, x="flipper_length_mm", bins=20)



单变量直方图

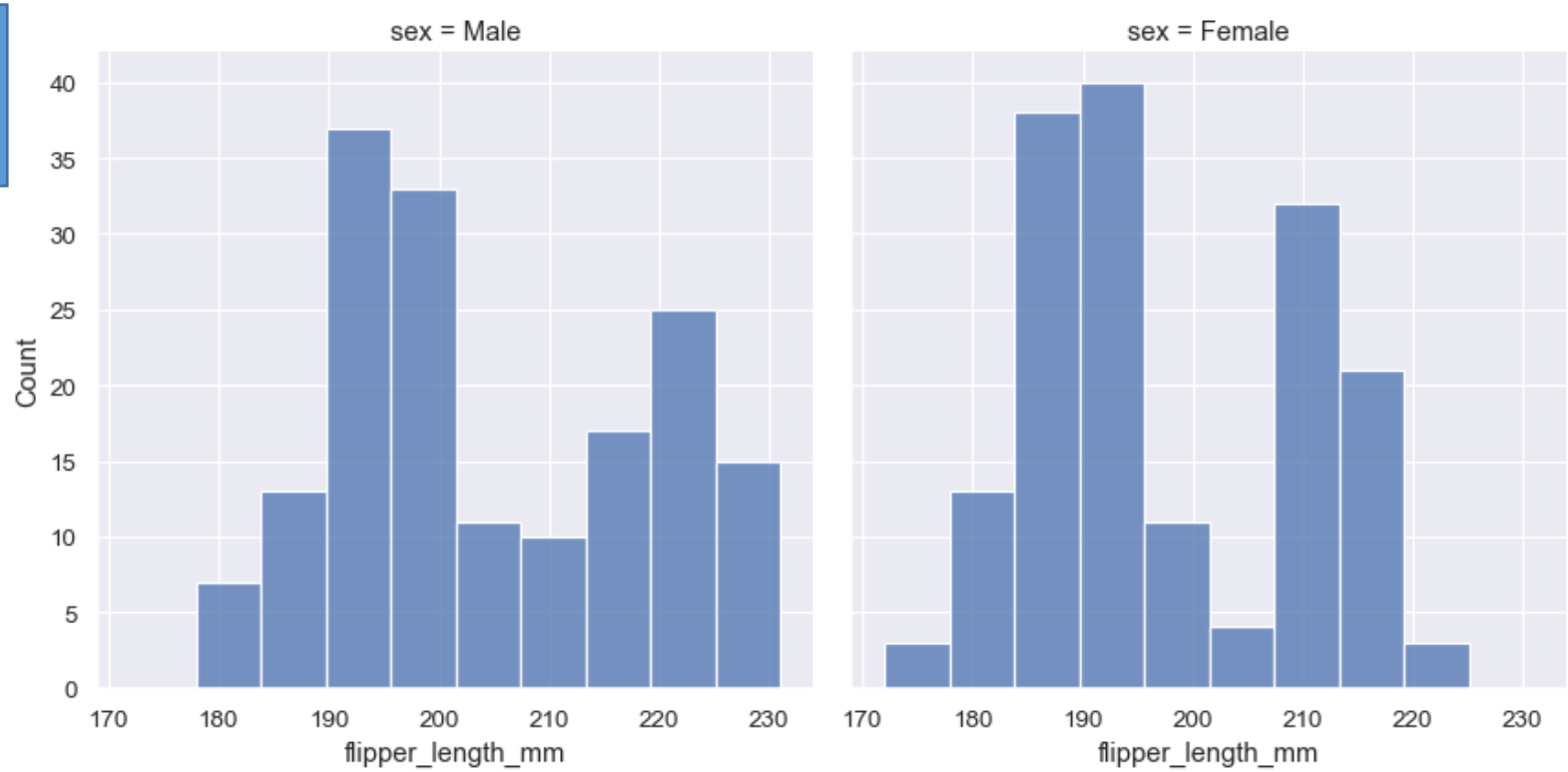
- tips = sns.load_dataset("tips")
- sns.displot(tips, x="size", bins=[1, 2, 3, 4, 5, 6, 7])



单变量直方图

- sns.displot(penguins, x="flipper_length_mm", col="sex")

col: 根据给定列中的不同值分不同子图

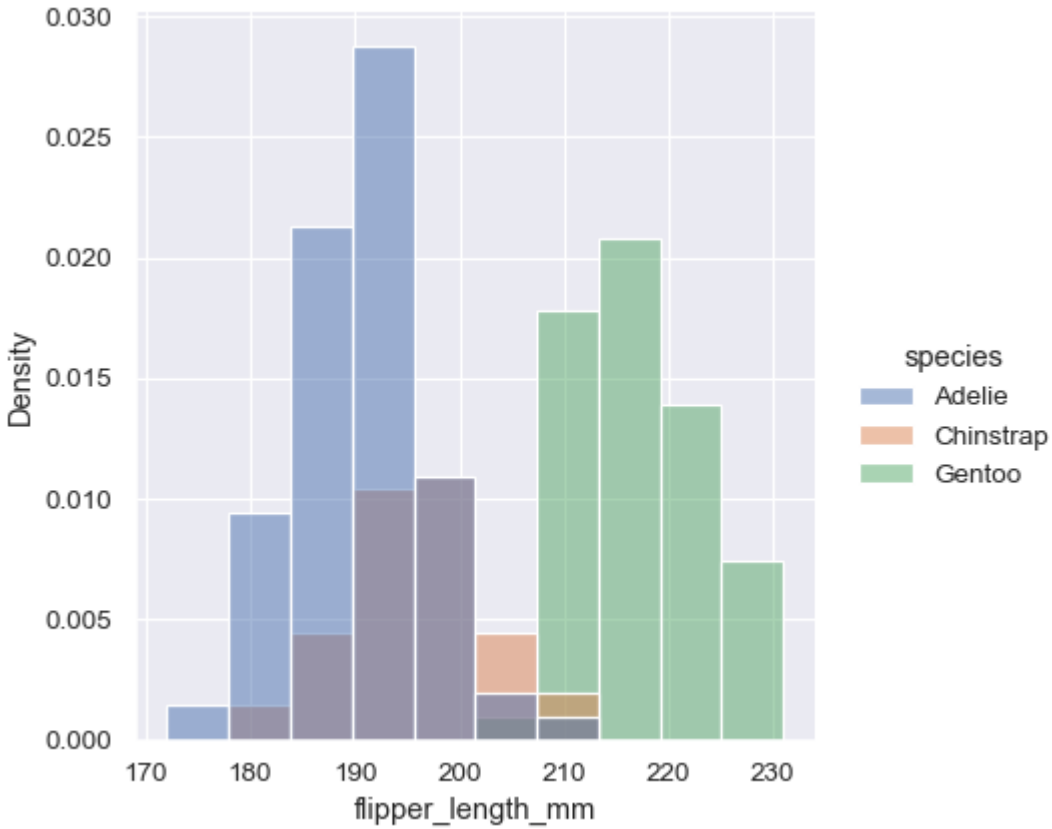


单变量直方图：标准化

Stat='density': 箱面积之和为1
Stat='probability': 箱高之和为1

- sns.displot(penguins, x="flipper_length_mm", hue="species", stat="density")

hue: 根据给定列中的不同值分隔行



密度分布 (核密度, kernel density estimation)

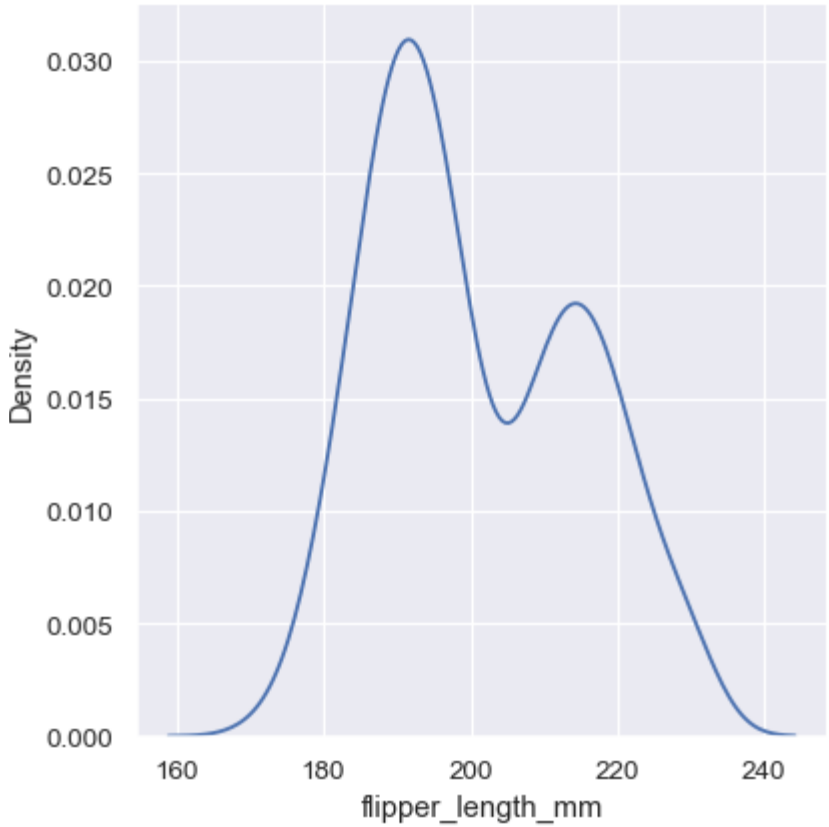
- sns.displot(penguins, x="flipper_length_mm", kind="kde")

直方图

核密度图

- `histplot()` (with `kind="hist"`; the default)
- `kdeplot()` (with `kind="kde"`)
- `ecdfplot()` (with `kind="ecdf"`; univariate-only)

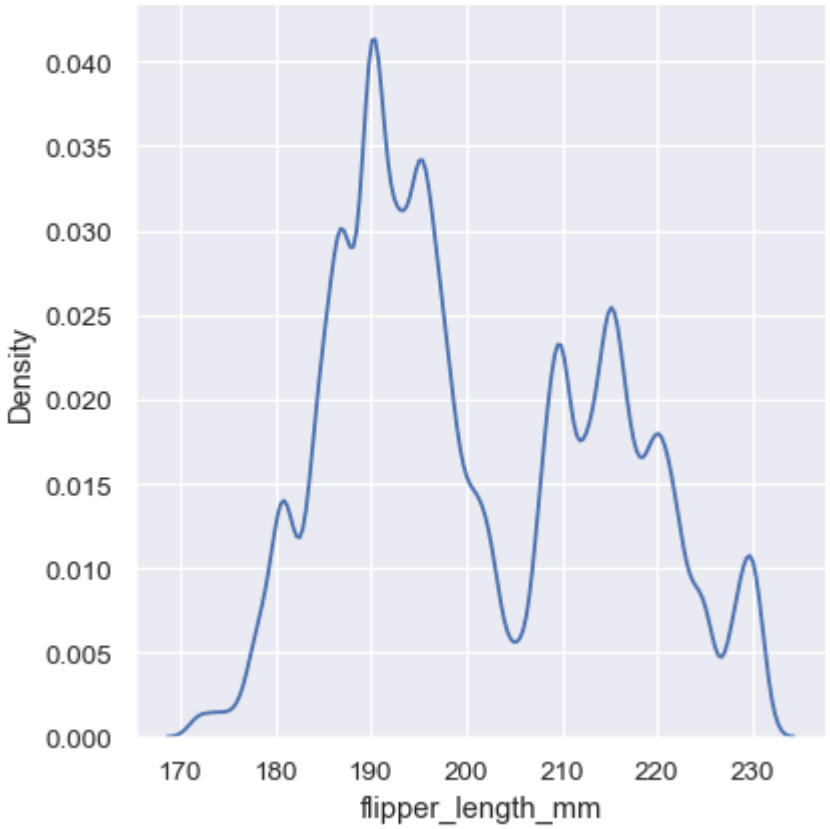
累积分布图，empirical cumulative distribution function



密度分布 (核密度, kernel density estimation)

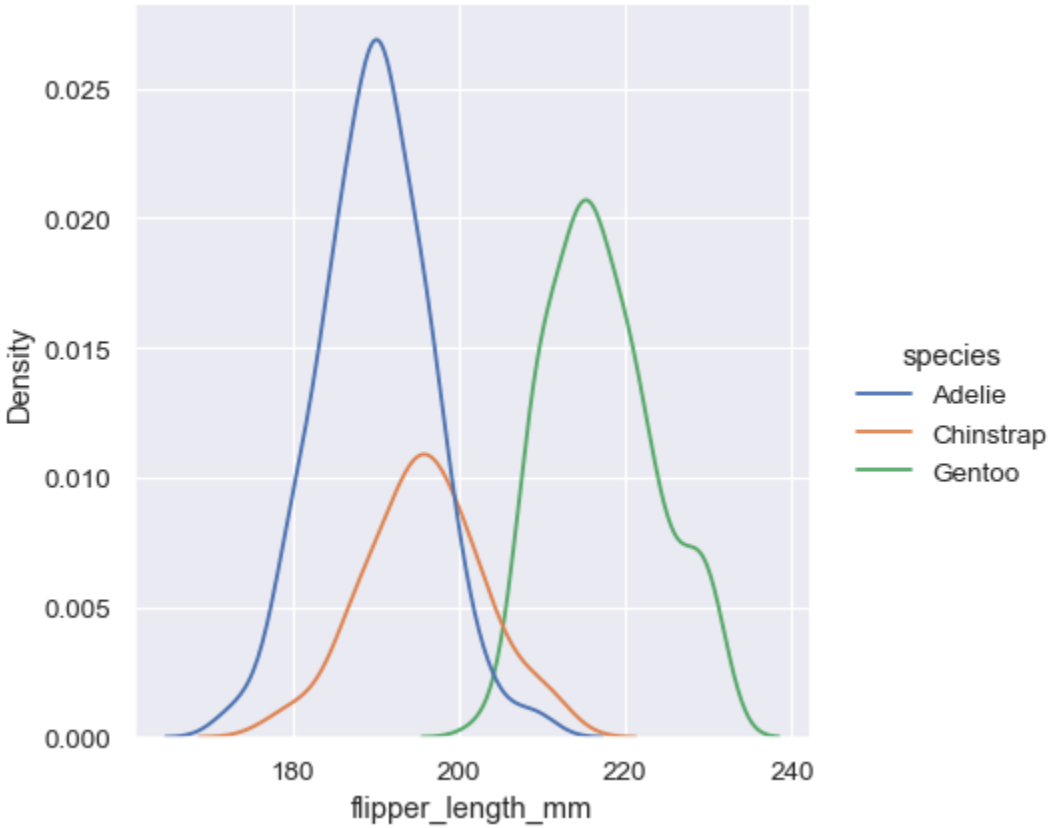
- sns.displot(penguins, x="flipper_length_mm", kind="kde", bw_adjust=.25)

分箱数量可能会影响结论



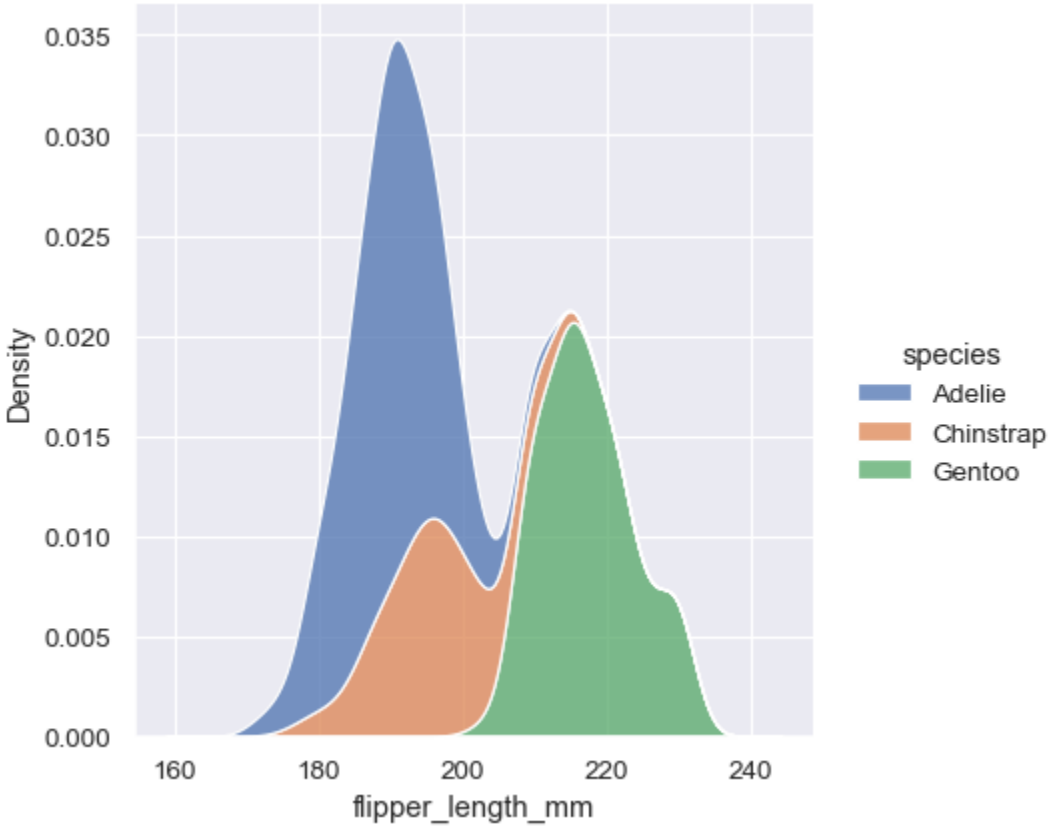
密度分布 (核密度, kernel density estimation)

- sns.displot(penguins, x="flipper_length_mm", hue="species", kind="kde")



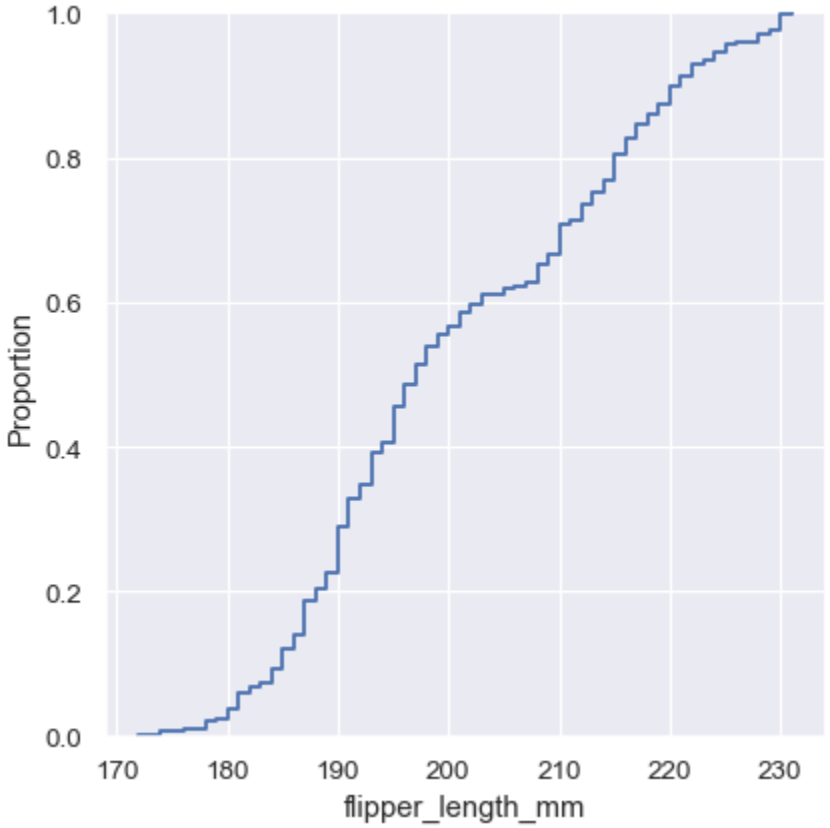
密度分布 (核密度, kernel density estimation)

- sns.displot(penguins, x="flipper_length_mm", hue="species", kind="kde", multiple="stack")



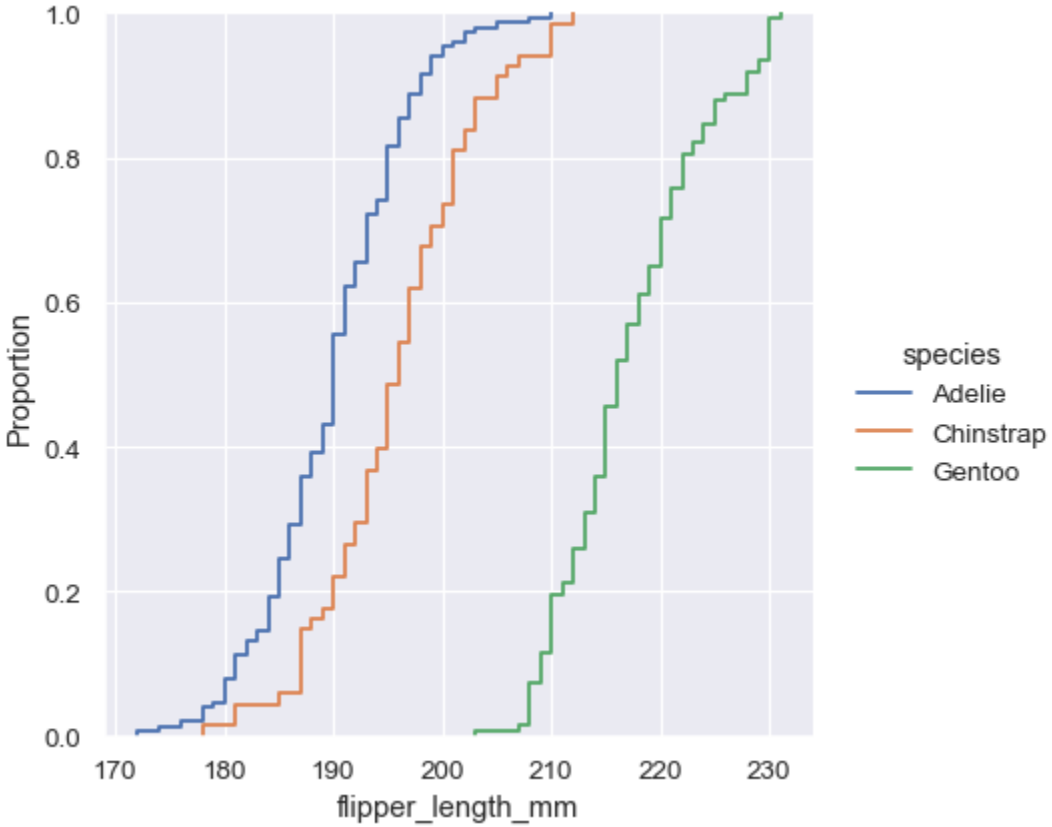
累积分布 (cumulative distribution function，CDF)

- sns.displot(penguins, x="flipper_length_mm", kind="ecdf")



累积分布 (cumulative distribution function，CDF)

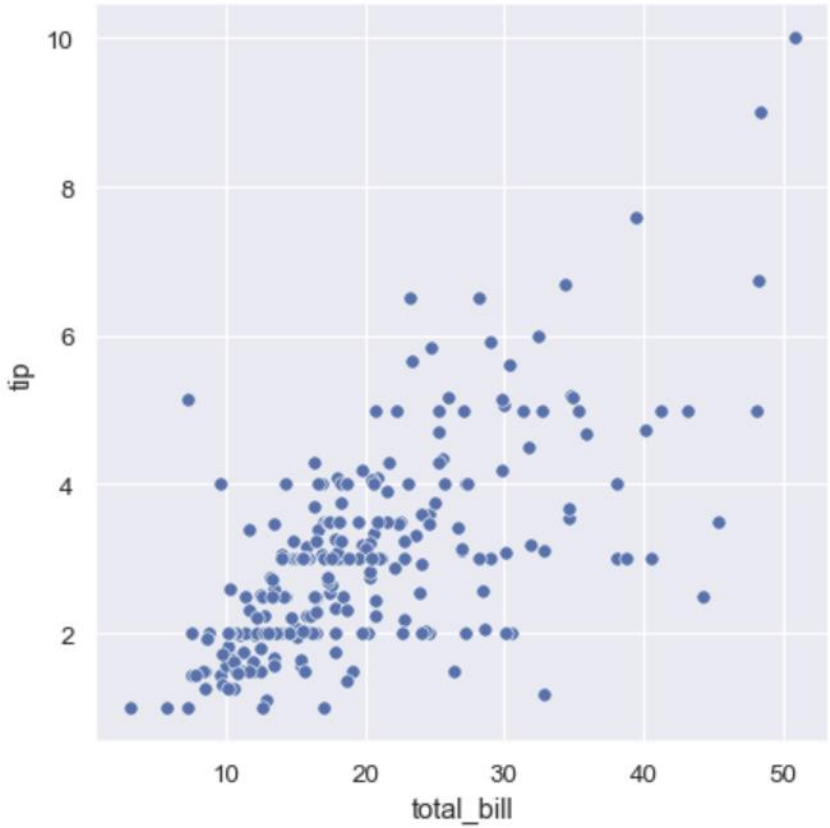
- sns.displot(penguins, x="flipper_length_mm", hue="species", kind="ecdf")



多变量间关系的可视化

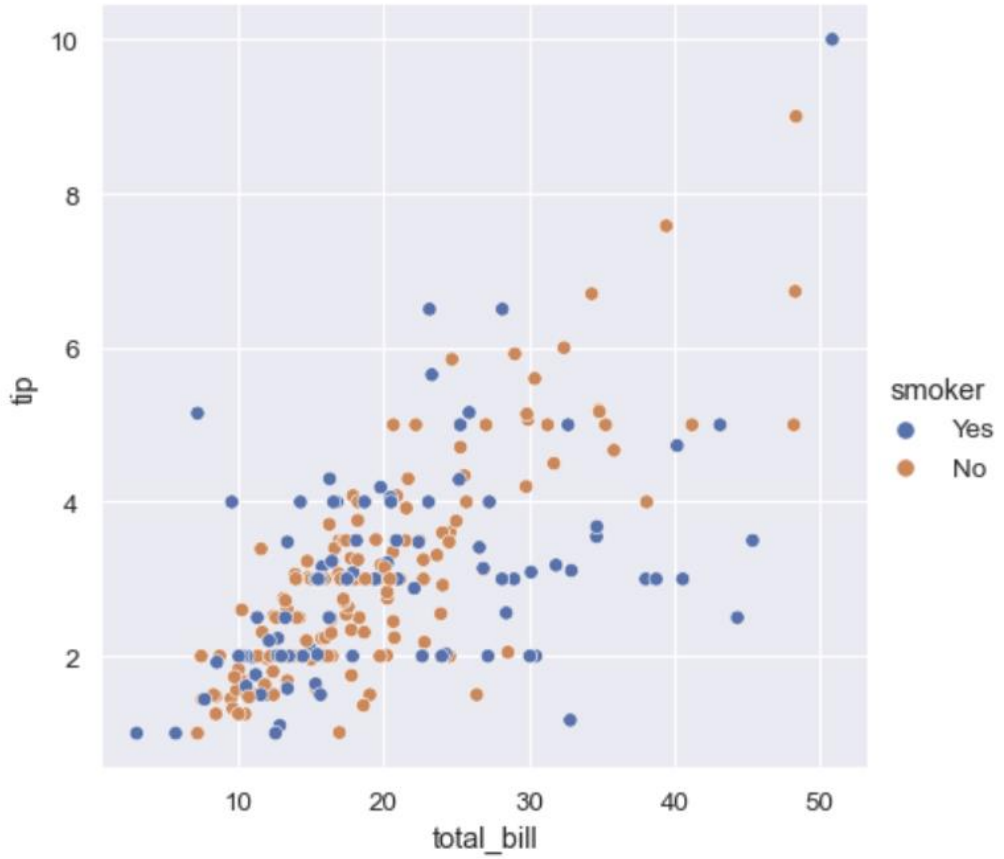
散点图

- tips = sns.load_dataset("tips")
- sns.relplot(x="total_bill", y="tip", data=tips);



散点图

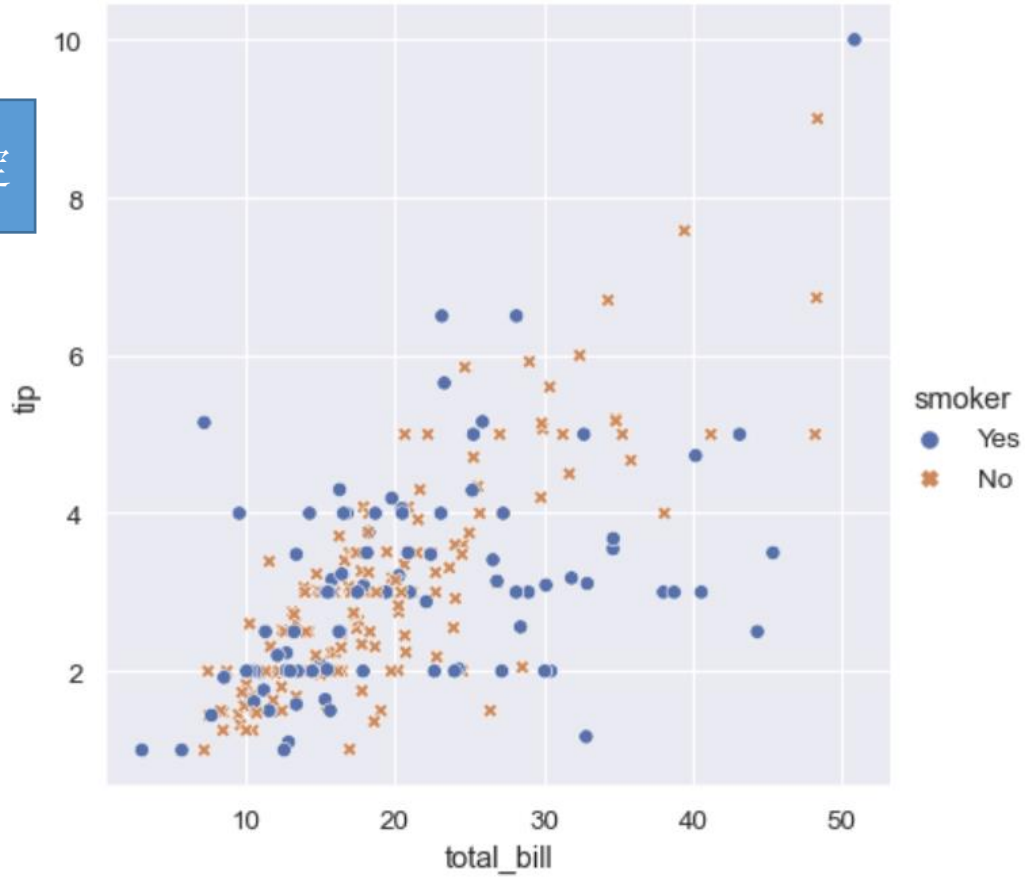
- sns.relplot(x="total_bill", y="tip", hue="smoker", data=tips)



散点图

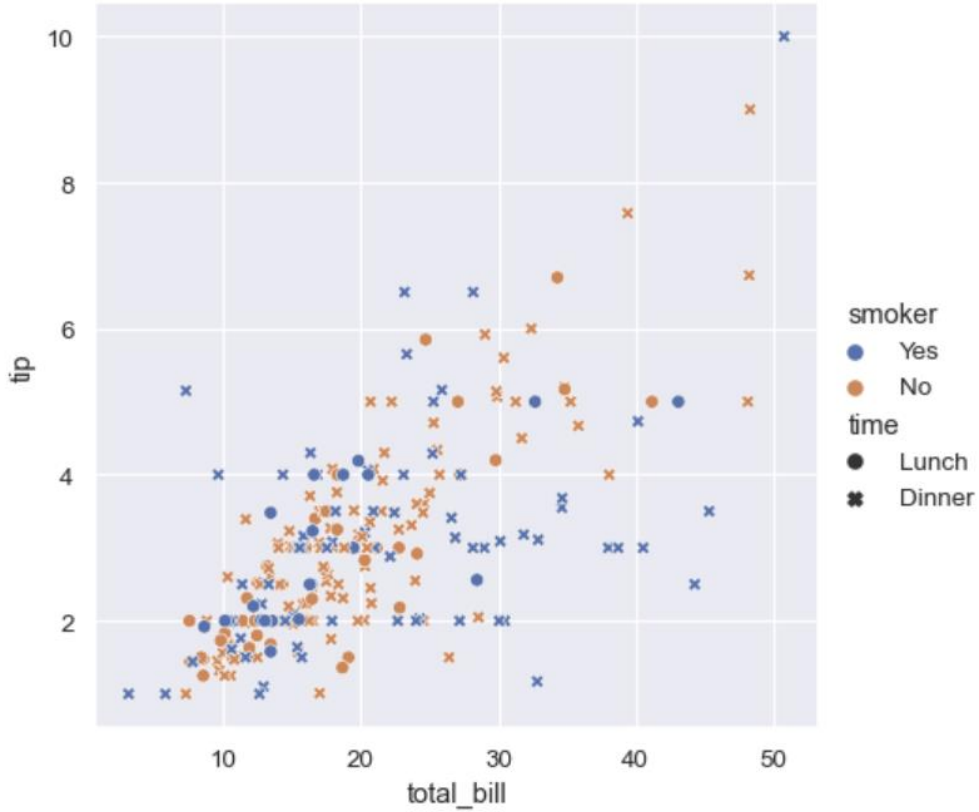
- sns.relplot(x="total_bill", y="tip", hue="smoker", style="smoker", data=tips)

点的类型也由smoker决定



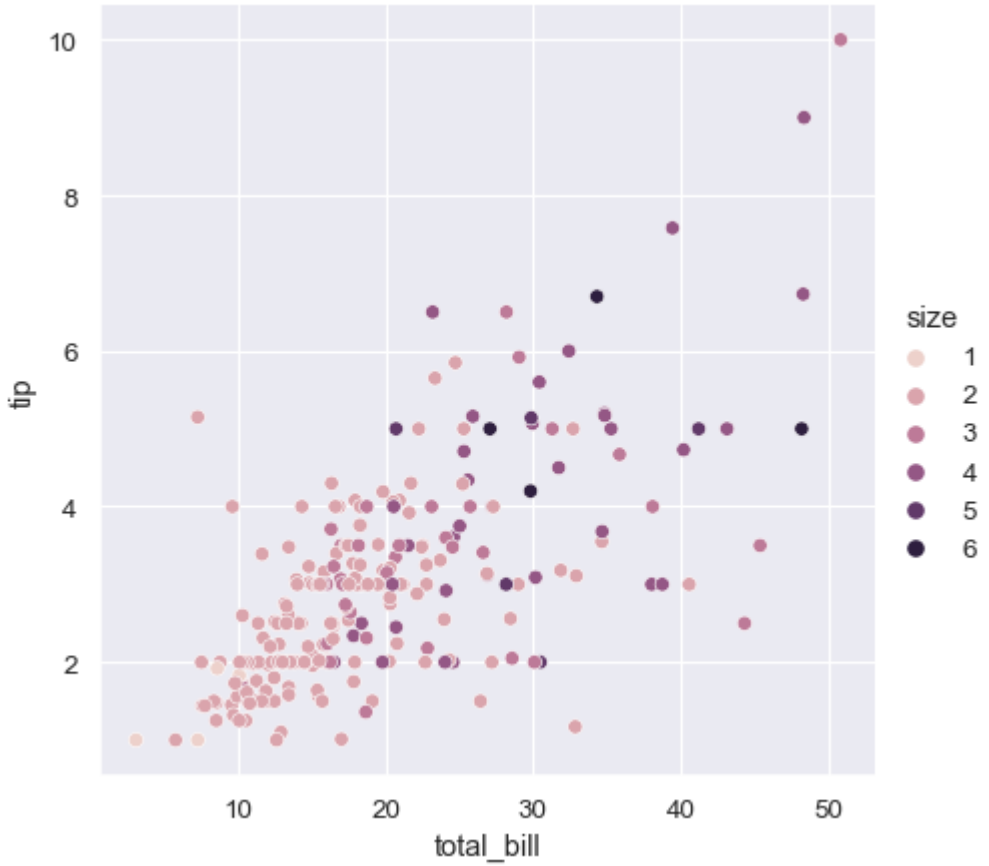
散点图

- sns.relplot(x="total_bill", y="tip", hue="smoker", style="time", data=tips)



散点图

- sns.relplot(x="total_bill", y="tip", hue="size", data=tips)



散点图

立方螺旋的色盘
生成算法

l 亮度 lightness
s 饱和度 saturation
h 色调 first hue
r 围绕调色板范围内的色相控制盘旋转 rotation

- sns.relplot(x="total_bill", y="tip", hue="size", palette="ch:r=-.5,l=.75", data=tips)

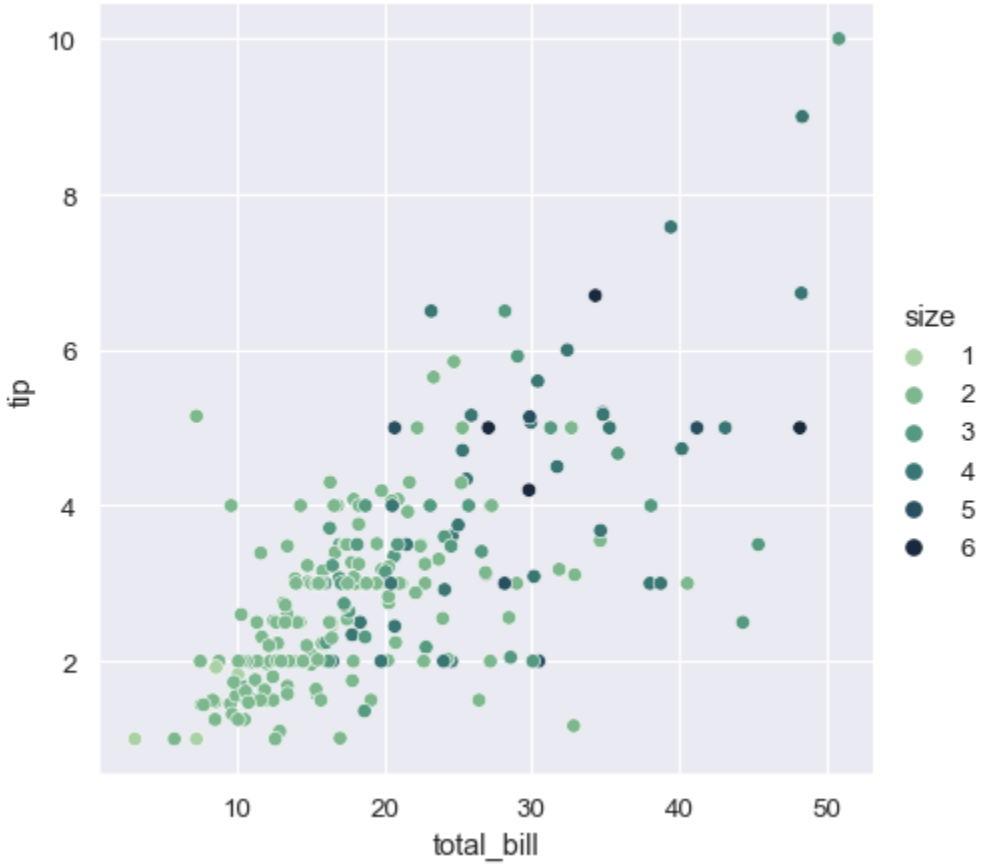
seaborn.color_palette

seaborn.**color_palette** (*palette=None, n_colors=None, desat=None,*

Return a list of colors or continuous colormap defining a palette.

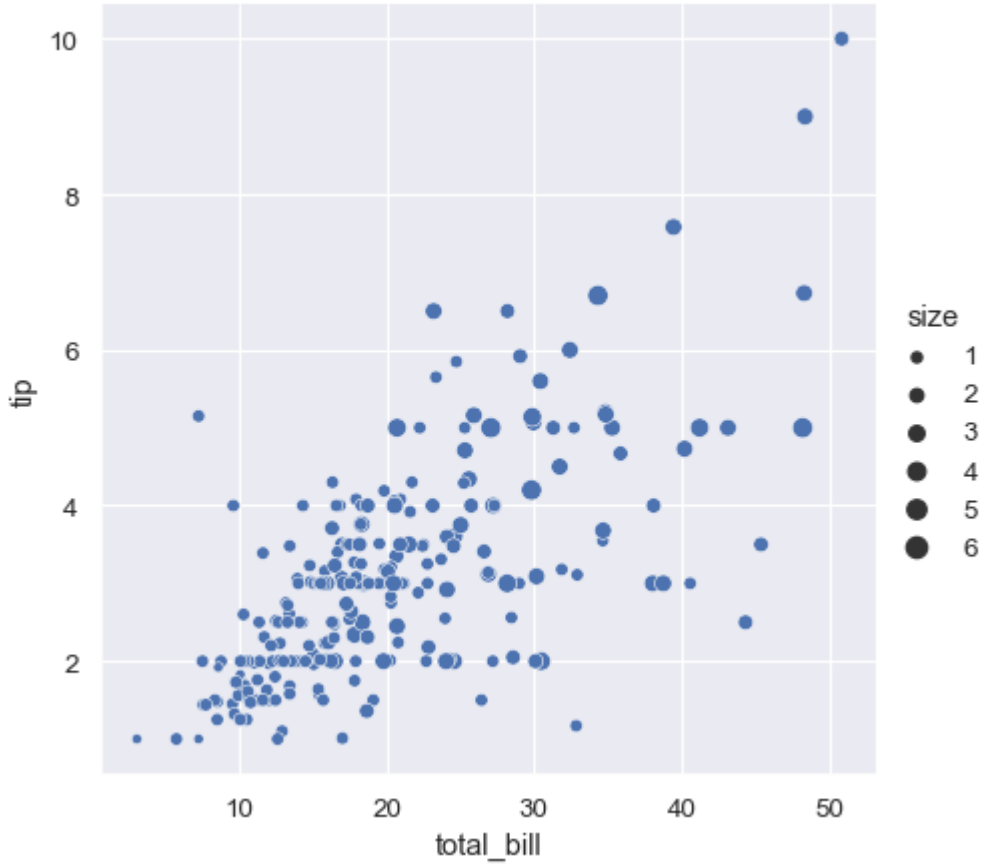
Possible `palette` **values include:**

- Name of a seaborn palette (deep, muted, bright, pastel, dark, colorblind)
- Name of matplotlib colormap
- 'husl' or 'hls'
- 'ch:<cubehelix arguments>'
- 'light:<color>', 'dark:<color>', 'blend:<color>,<color>',
- A sequence of colors in any format matplotlib accepts



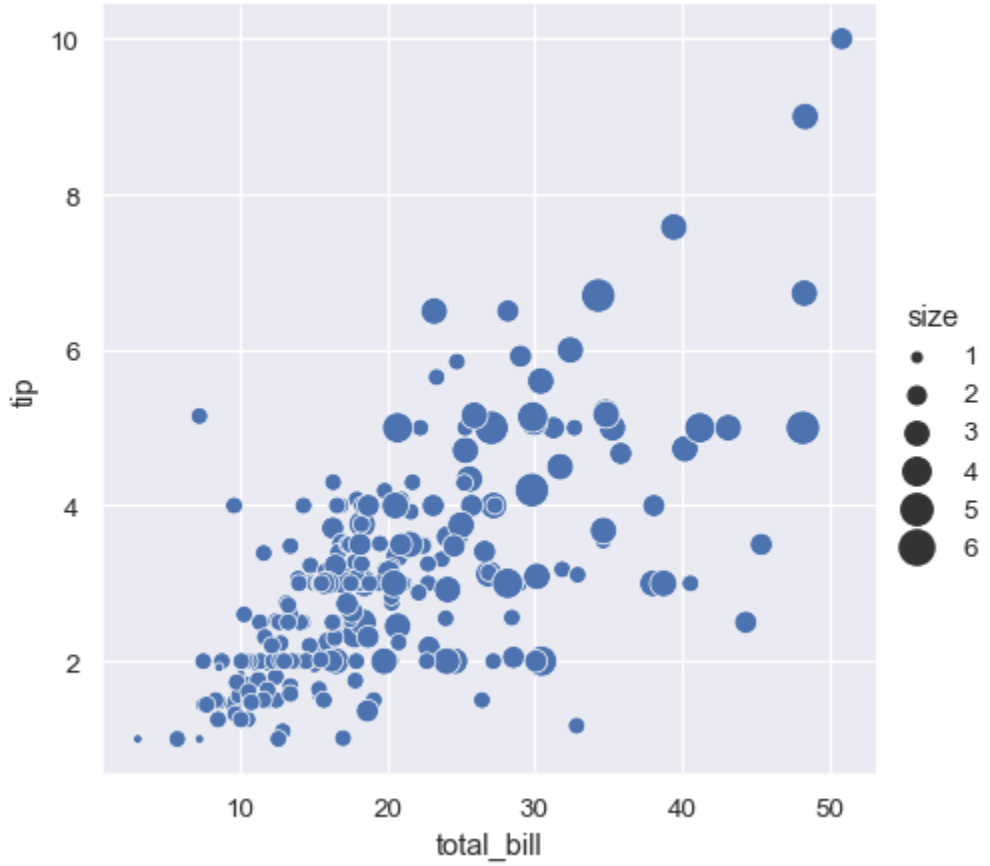
散点图

- sns.relplot(x="total_bill", y="tip", size="size", data=tips)



散点图

- sns.relplot(x="total_bill", y="tip", size="size", sizes=(15, 200), data=tips)

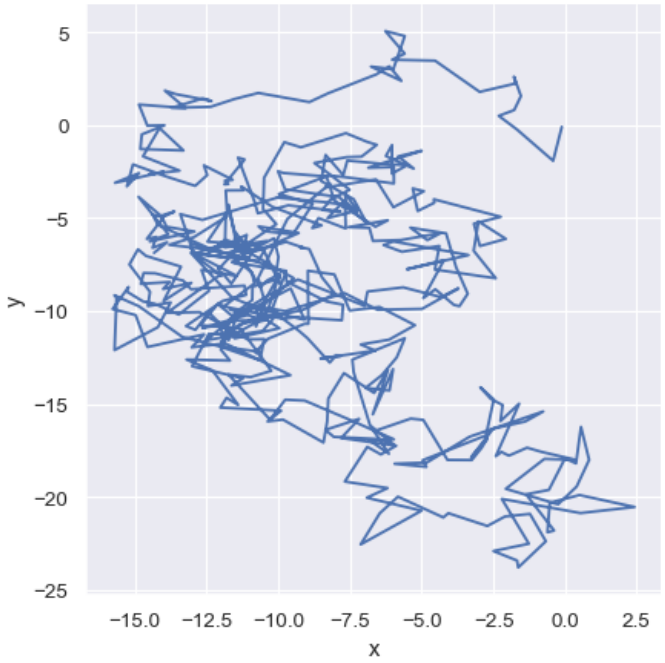


折线图

- df = pd.DataFrame(dict(time=np.arange(500), value=np.random.randn(500).cumsum()))
- g = sns.relplot(x="time", y="value", kind="line", data=df)
- g.fig.autofmt_xdate()

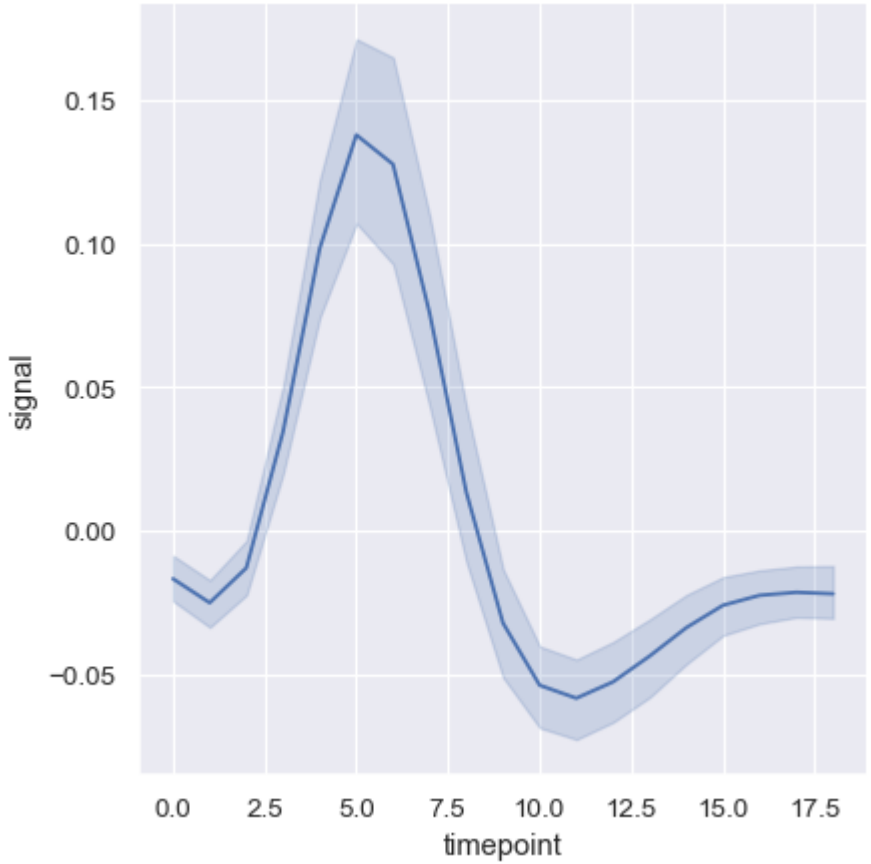
折线图

- df = pd.DataFrame(np.random.randn(500, 2).cumsum(axis=0), columns=["x", "y"])
- sns.relplot(x="x", y="y", sort=False, kind="line", data=df)



折线图

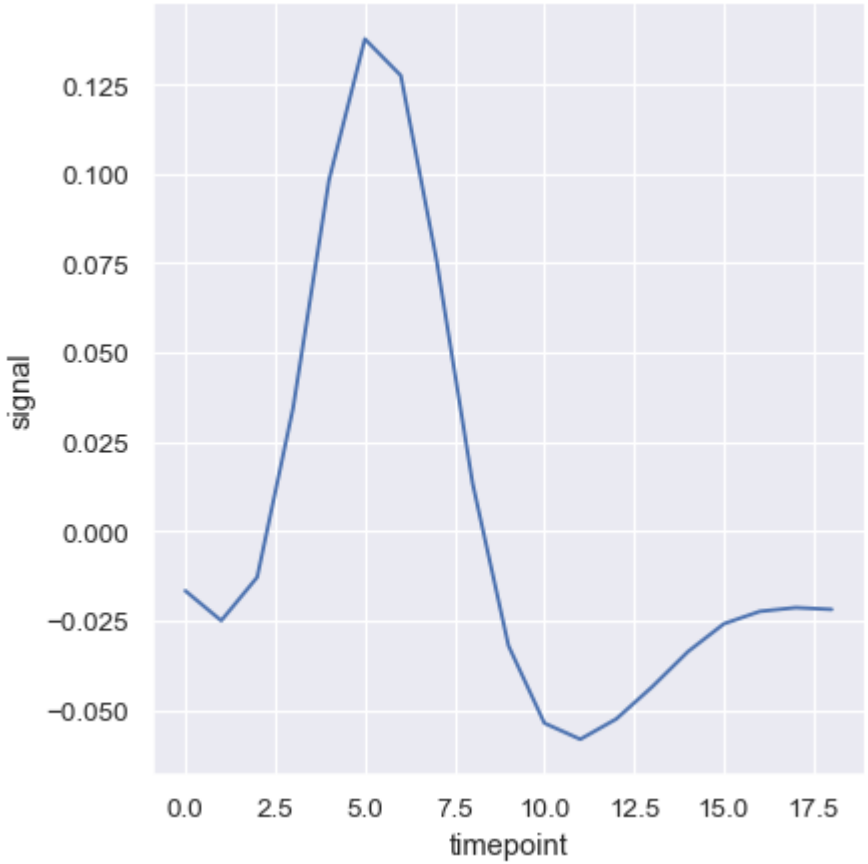
- fmri = sns.load_dataset("fmri")
- sns.relplot(x="timepoint", y="signal", kind="line", data=fmri)



折线图

置信区间confidence interval

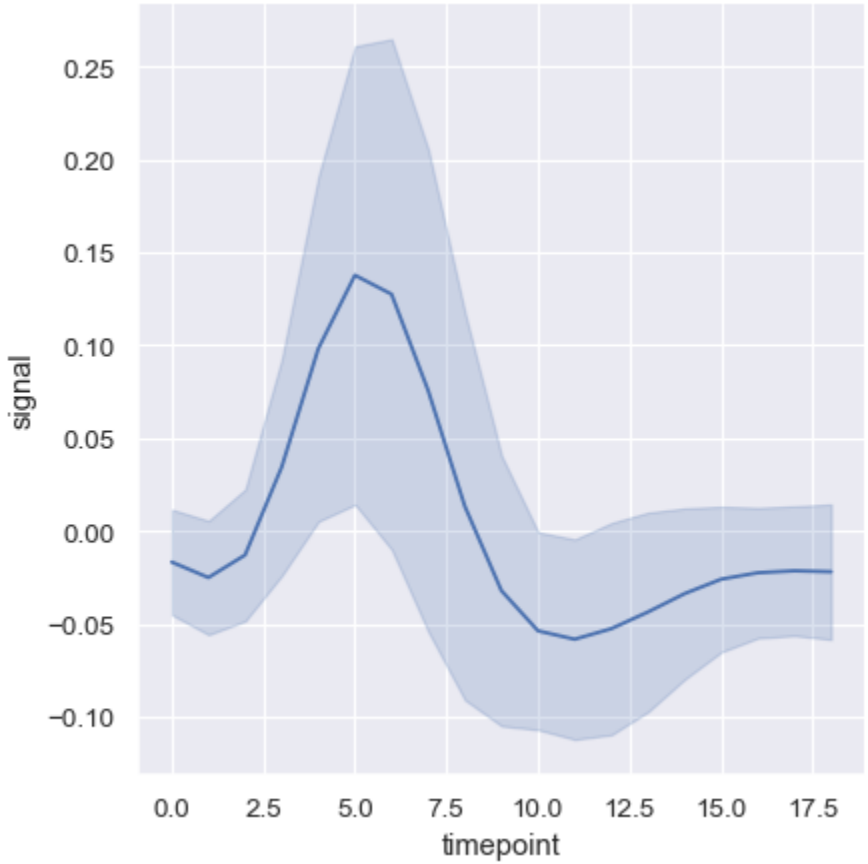
- sns.relplot(x="timepoint", y="signal", ci=None, kind="line", data=fmri)



折线图

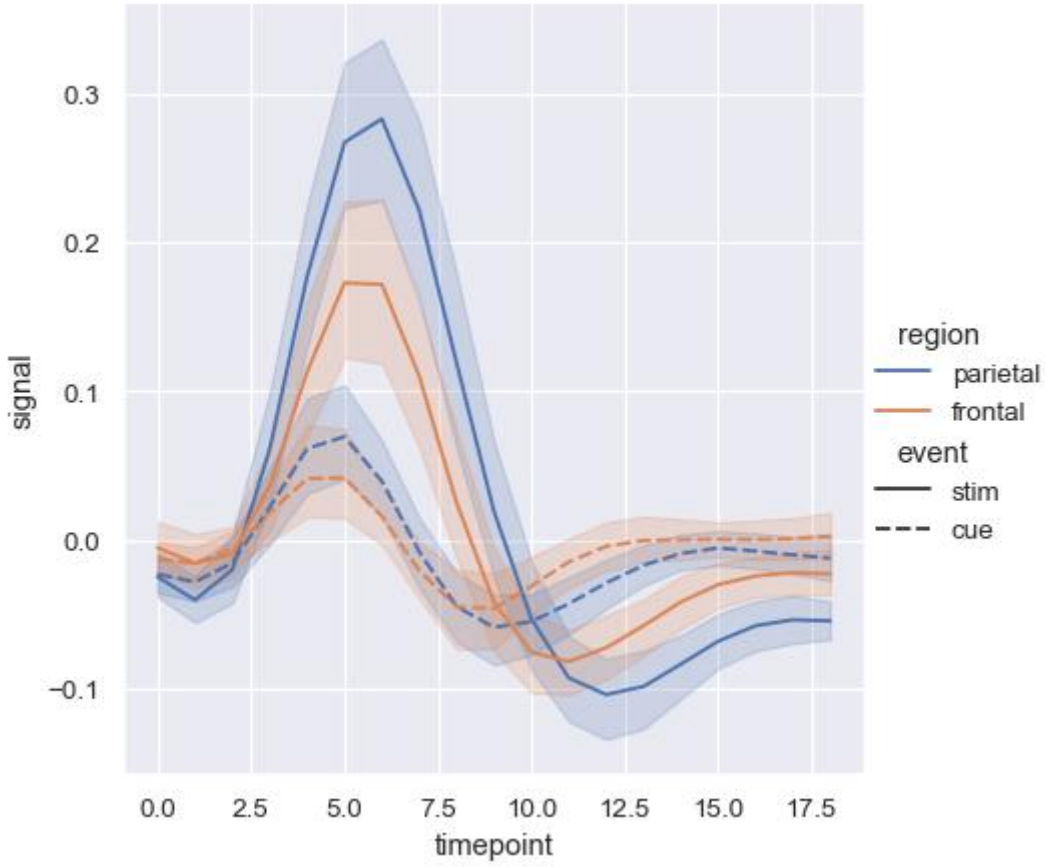
使用标准差绘制阴影

- sns.relplot(x="timepoint", y="signal", kind="line", ci="sd", data=fmri)



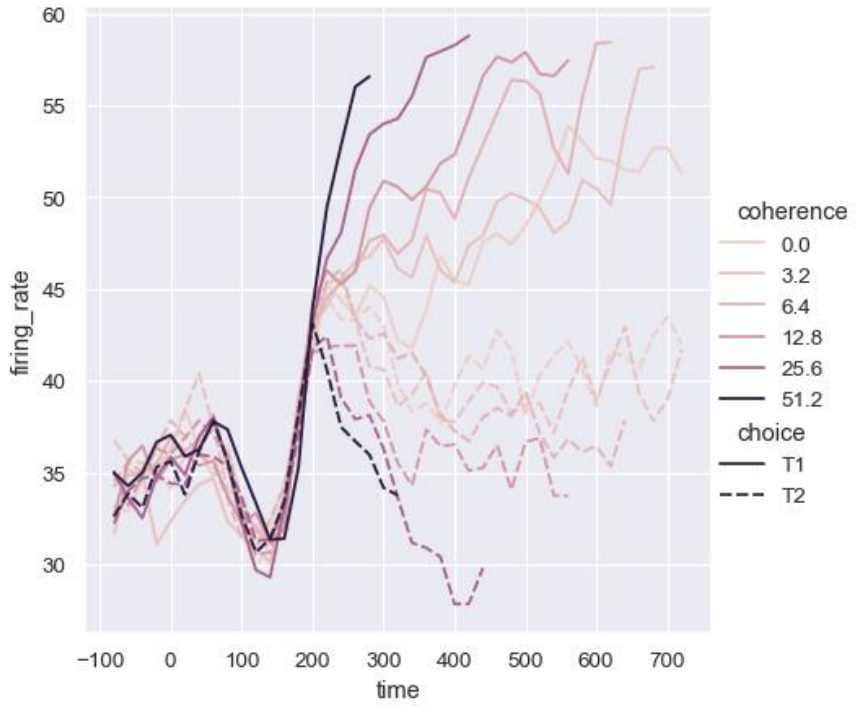
折线图

- sns.relplot(x="timepoint", y="signal", hue="region", style="event", kind="line", data=fmri)



折线图

- dots = sns.load_dataset("dots").query("align == 'dots'")
- sns.relplot(x="time", y="firing_rate", hue="coherence", style="choice", kind="line", data=dots)



分面(facet)的方法表示多个关系

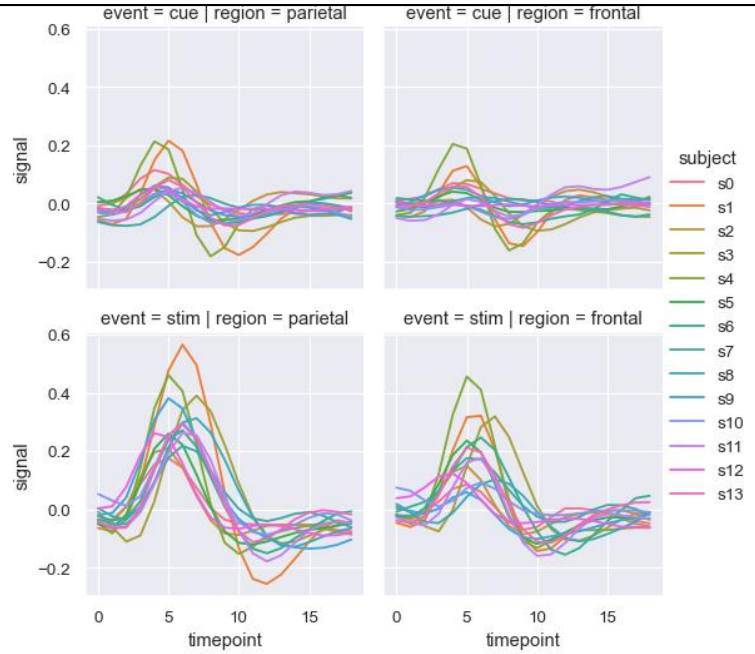
- sns.relplot(x="total_bill", y="tip", hue="smoker", col="time", data=tips)

分面(facet)的方法表示多个关系

- sns.relplot(x="timepoint", y="signal", hue="subject", col="region", row="event", height=3, kind="line", estimator=None, data=fmri)

estimator : ***name of pandas method or callable or None***

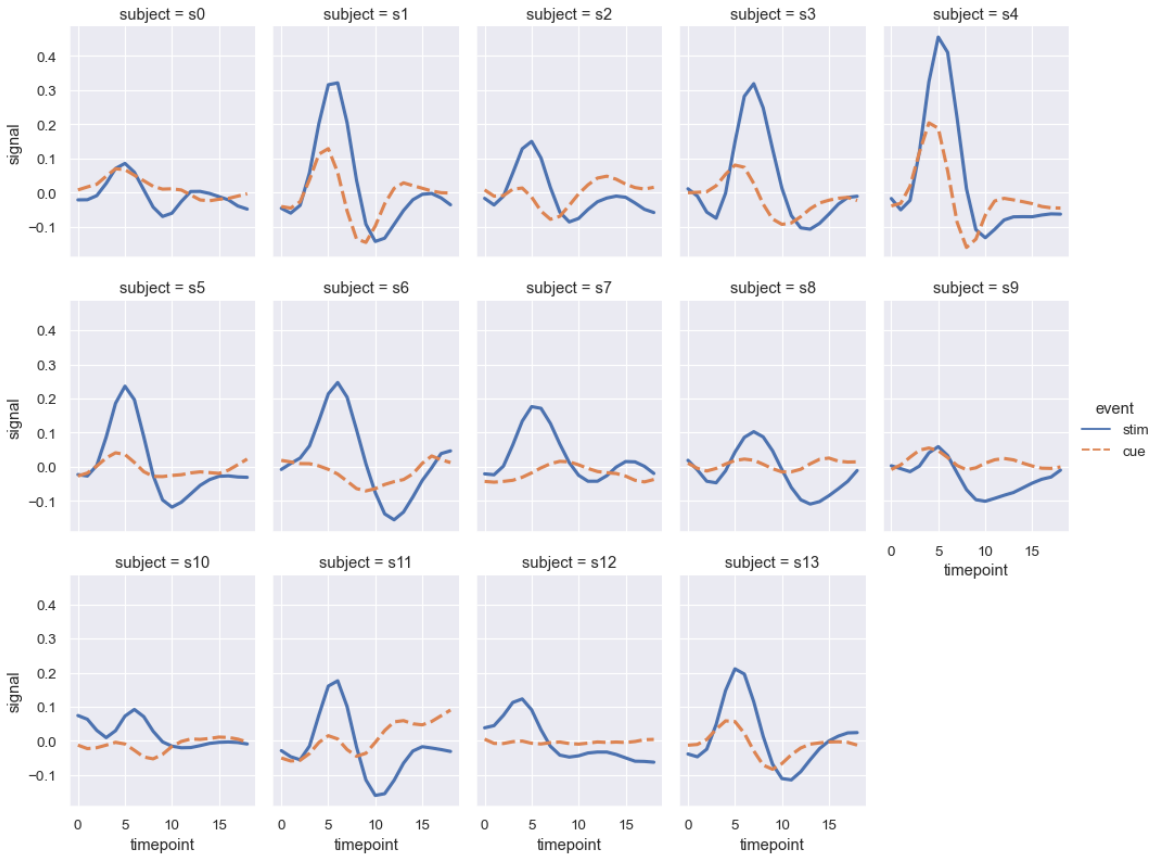
Method for aggregating across multiple observations of the `y` variable at the same `x` level. If `None`, all observations will be drawn. *Currently non-functional.*



相对高度

分面(facet)的方法表示多个关系

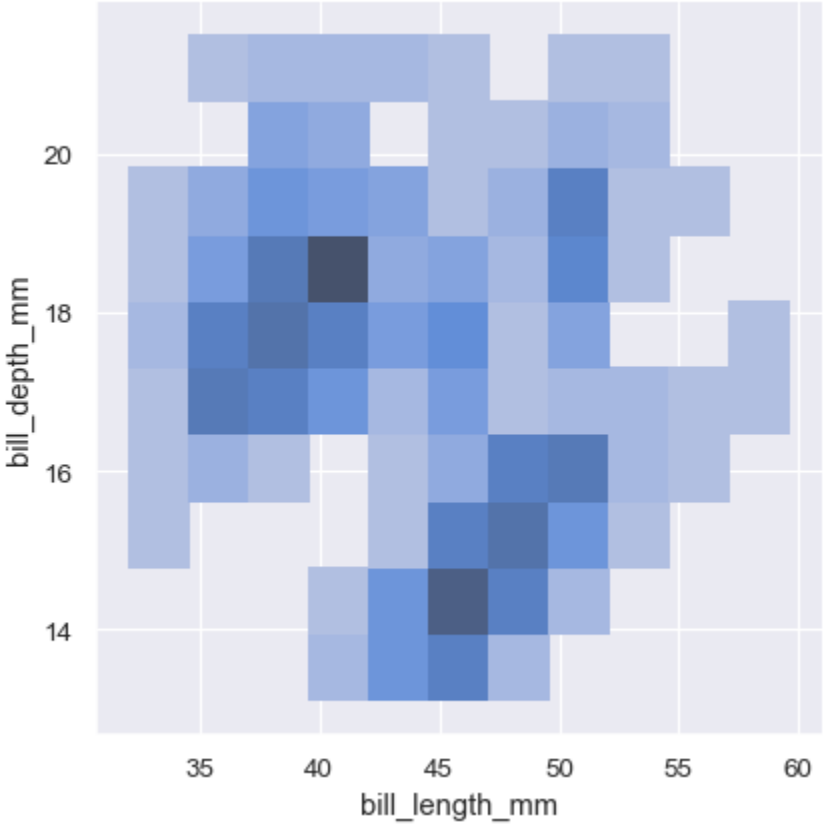
- sns.relplot(x="timepoint", y="signal", hue="event", style="event", col="subject", col_wrap=5, height=3, aspect=.75, linewidth=2.5, kind="line", data=fmri.query("region == 'frontal'"))



热力图

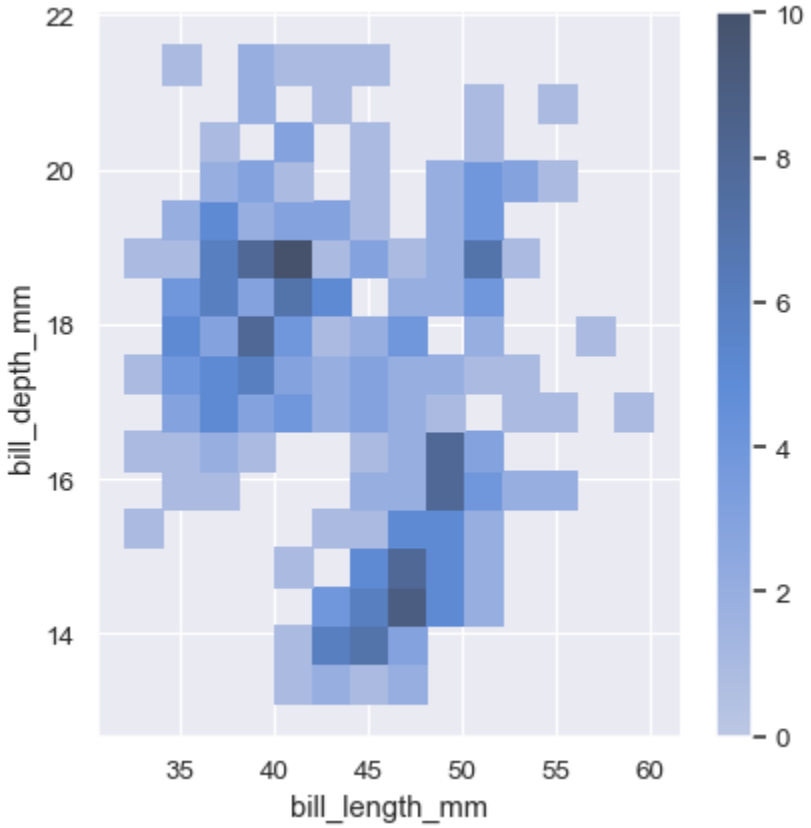
热力图本质上是多个变量分布的可视化！

- sns.displot(penguins, x="bill_length_mm", y="bill_depth_mm")



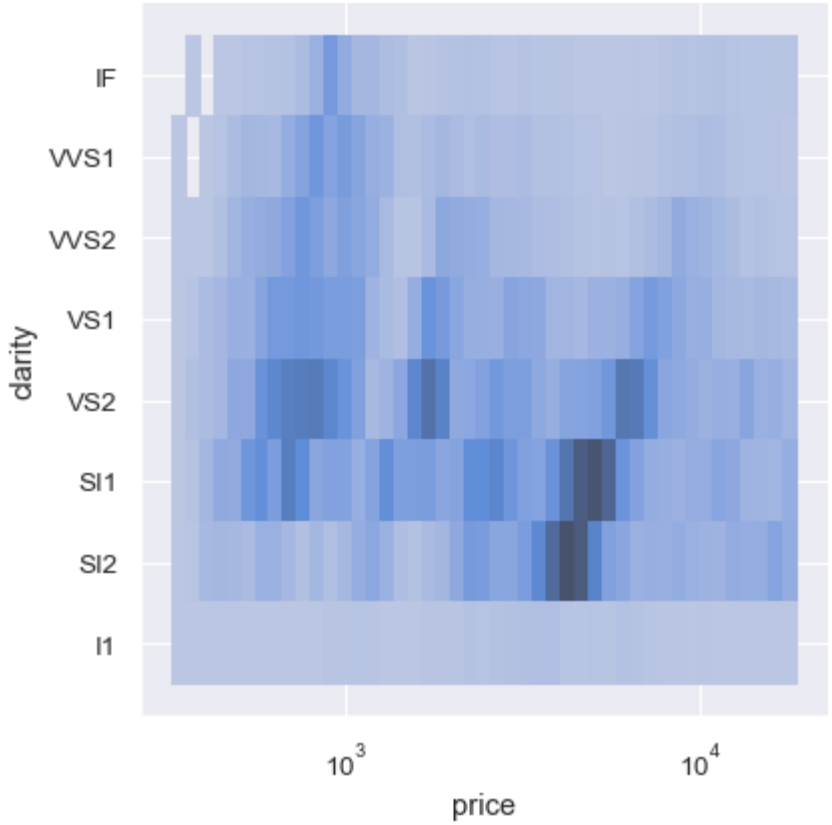
热力图

- sns.displot(penguins, x="bill_length_mm", y="bill_depth_mm", binwidth=(2, .5), cbar=True)



热力图

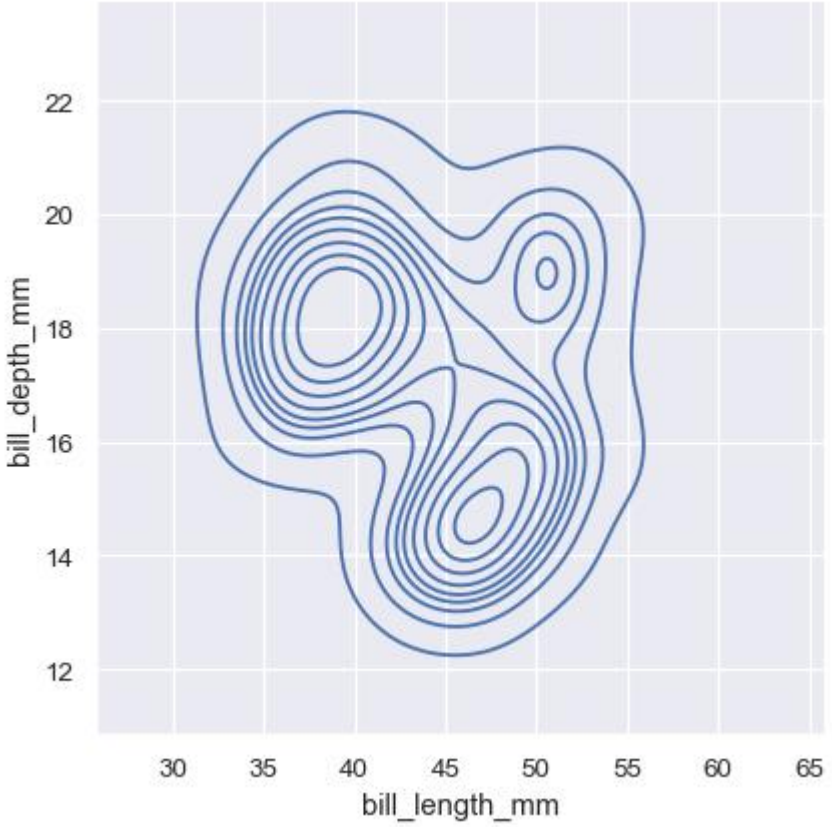
- diamonds = sns.load_dataset("diamonds")
- sns.displot(diamonds, x="price", y="clarity", log_scale=(True, False))



等高线

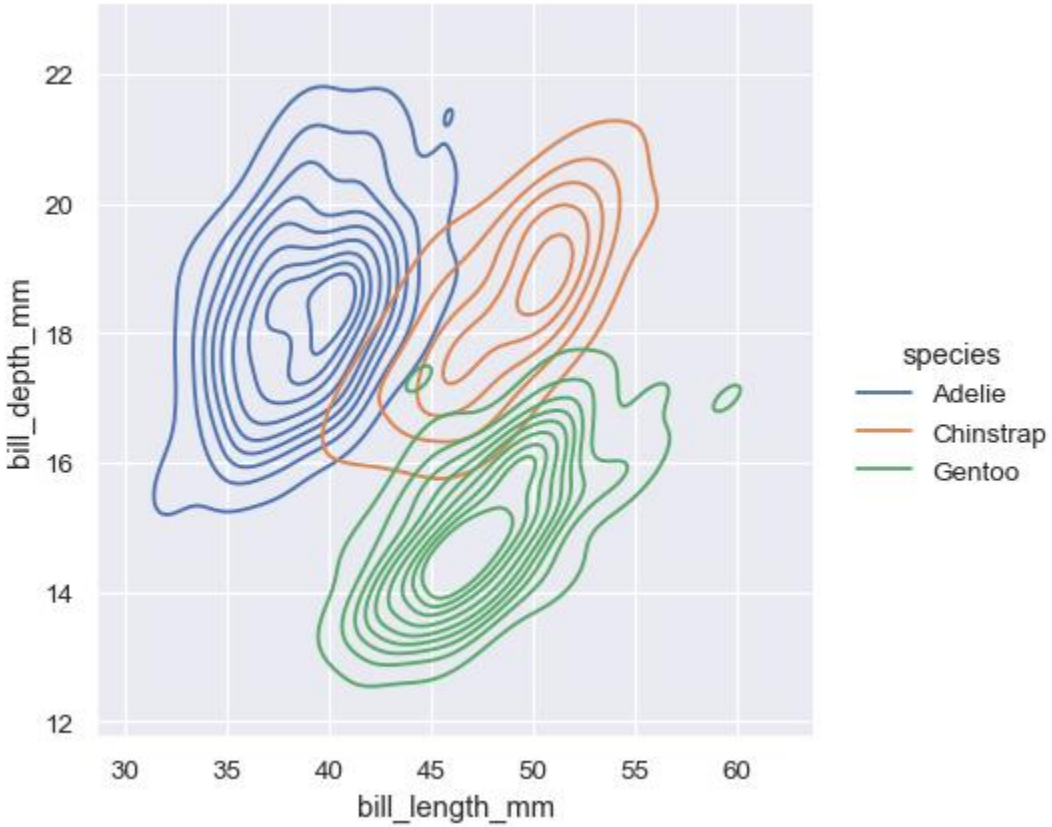
等高线本质上也是多个变量分布的可视化！

- sns.displot(penguins, x="bill_length_mm", y="bill_depth_mm", kind="kde")



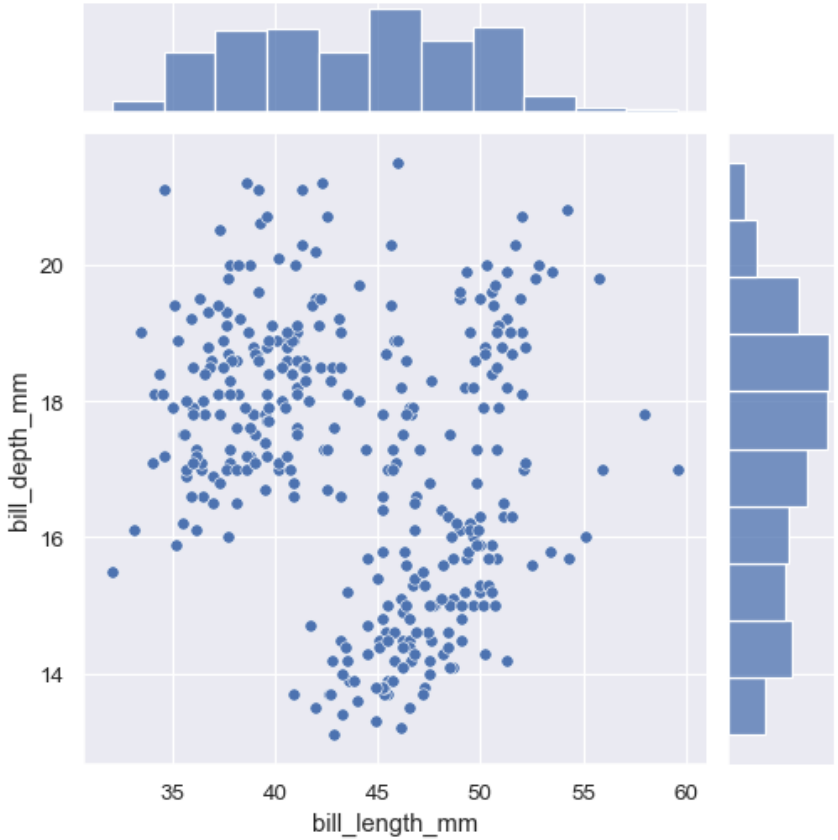
等高线

- sns.displot(penguins, x="bill_length_mm", y="bill_depth_mm", hue="species", kind="kde")



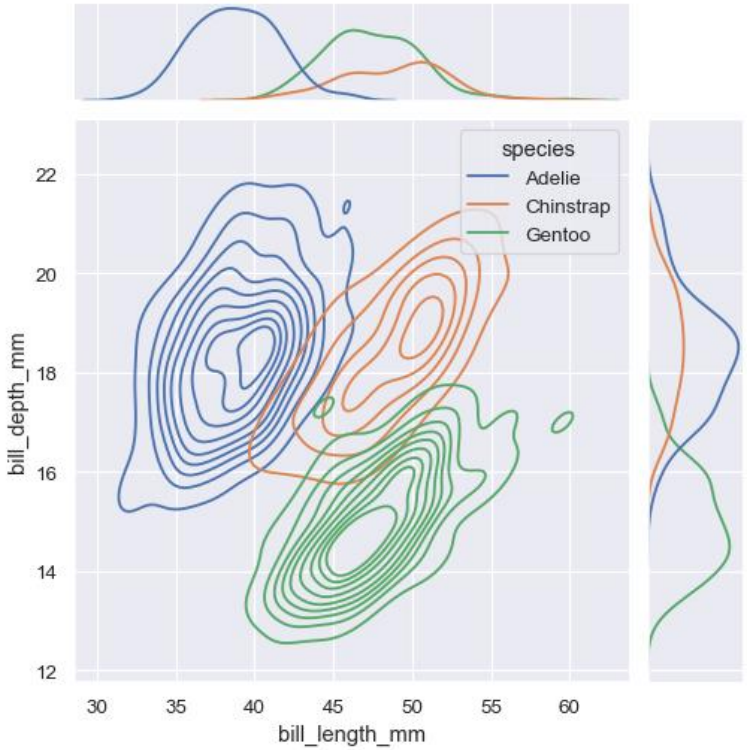
jointplot()

- sns.jointplot(data=penguins, x="bill_length_mm", y="bill_depth_mm")



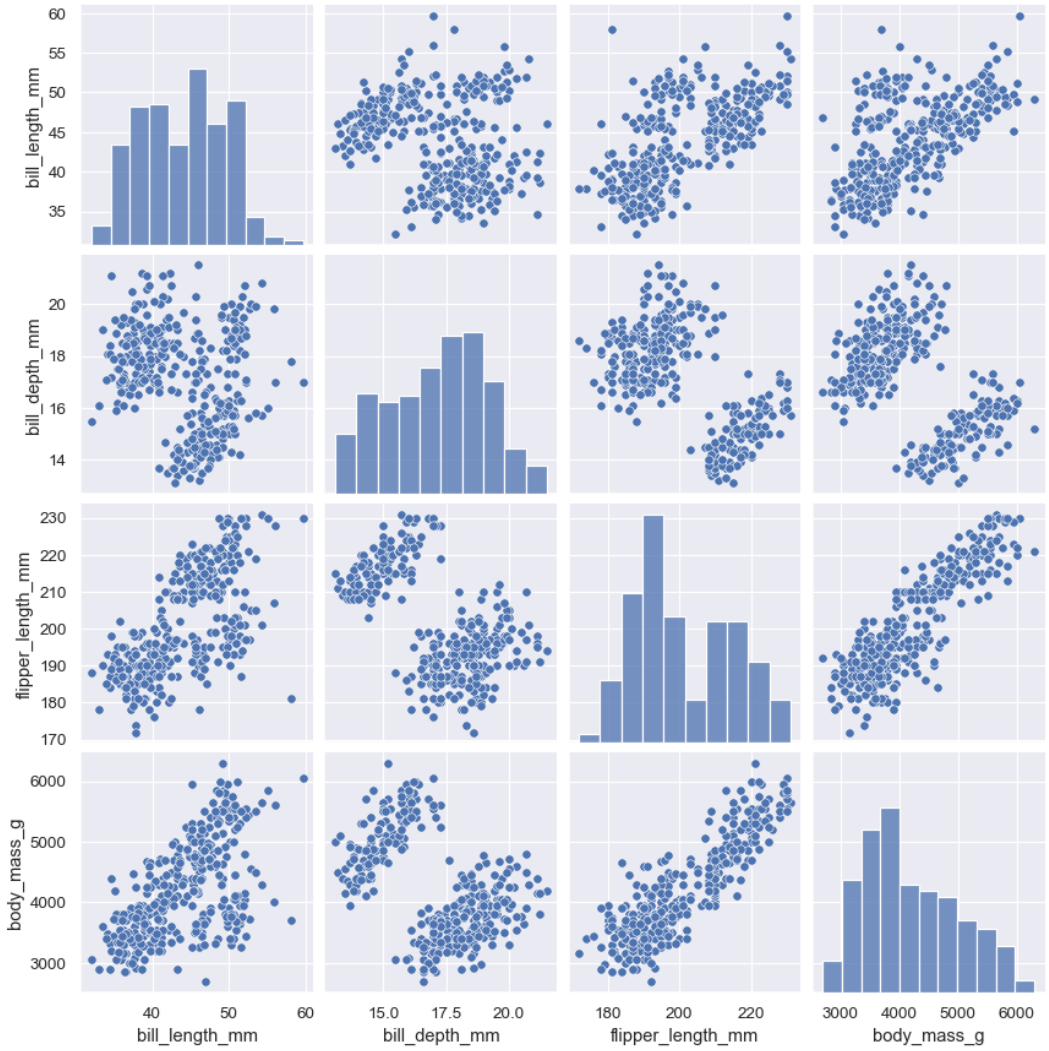
jointplot()

- sns.jointplot(
- data=penguins,
- x="bill_length_mm", y="bill_depth_mm", hue="species",
- kind="kde"
-)



pairplot()

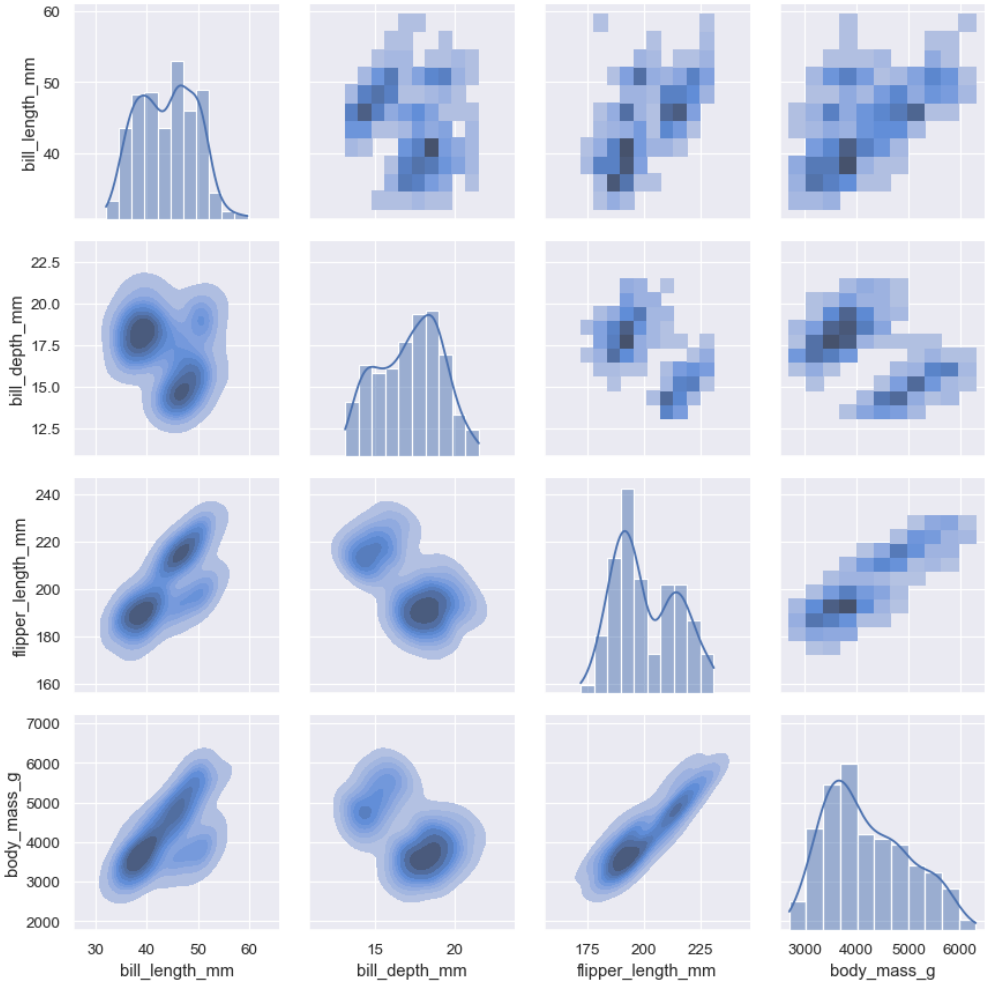
- sns.pairplot(penguins)



PairGrid()

- g = sns.PairGrid(penguins)
- g.map_upper(sns.histplot)
- g.map_lower(sns.kdeplot, fill=True)
- g.map_diag(sns.histplot, kde=True)

上三角、下三角和对角线



定类变量的可视化

定类变量的可视化：方法

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_theme(style="ticks",
color_codes=True)
```

Categorical scatterplots:

- `stripplot()` (with `kind="strip"`; the default)
- `swarmplot()` (with `kind="swarm"`)

Categorical distribution plots:

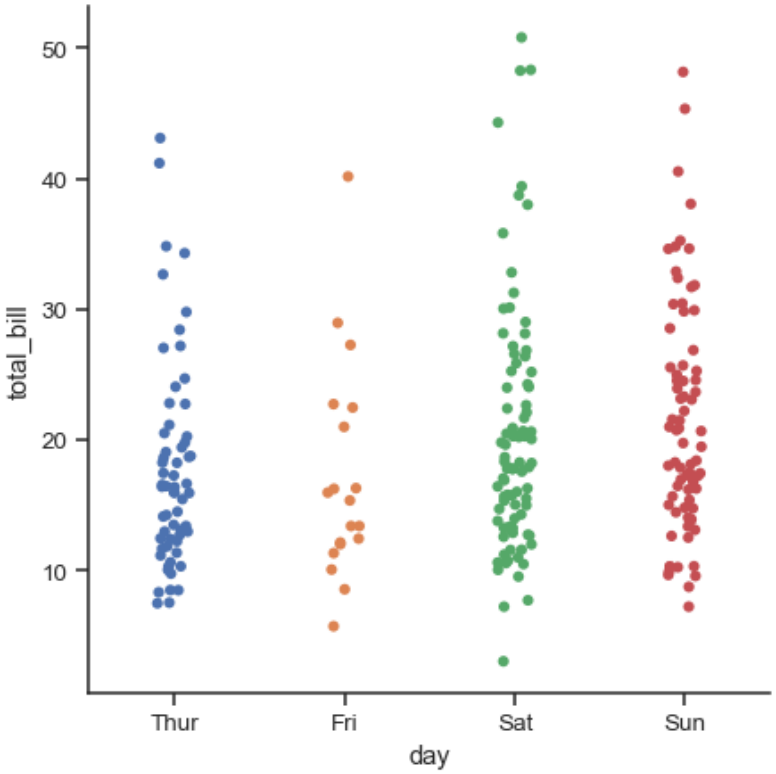
- `boxplot()` (with `kind="box"`)
- `violinplot()` (with `kind="violin"`)
- `boxenplot()` (with `kind="boxen"`)

Categorical estimate plots:

- `pointplot()` (with `kind="point"`)
- `barplot()` (with `kind="bar"`)
- `countplot()` (with `kind="count"`)

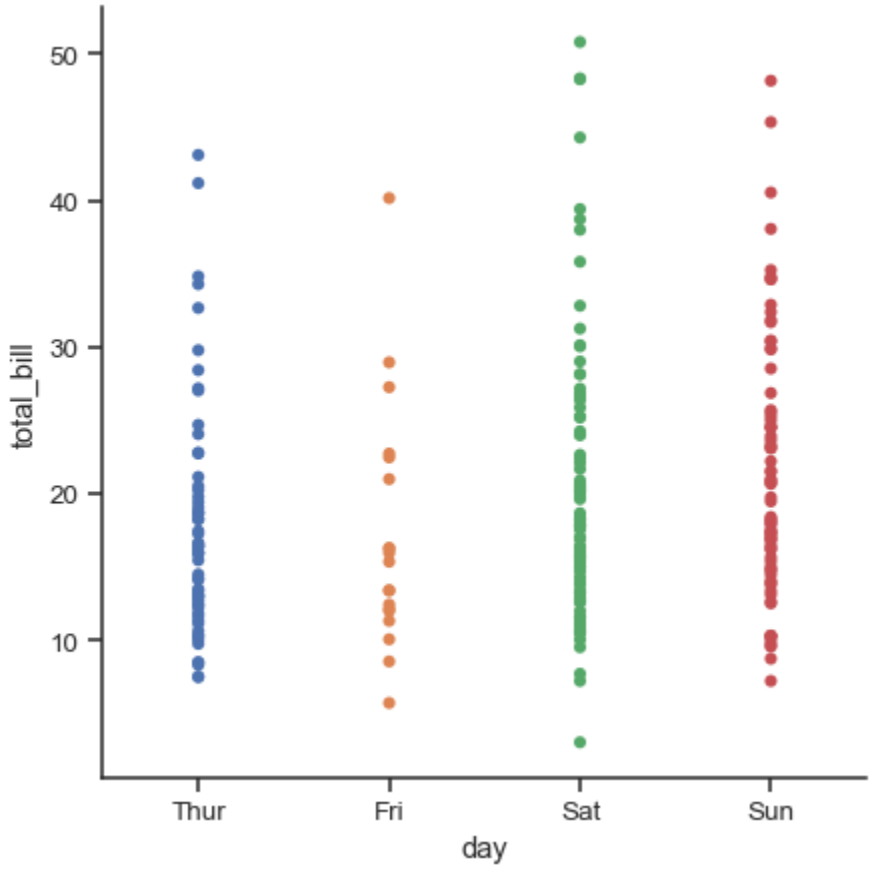
定类散点图

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



定类散点图

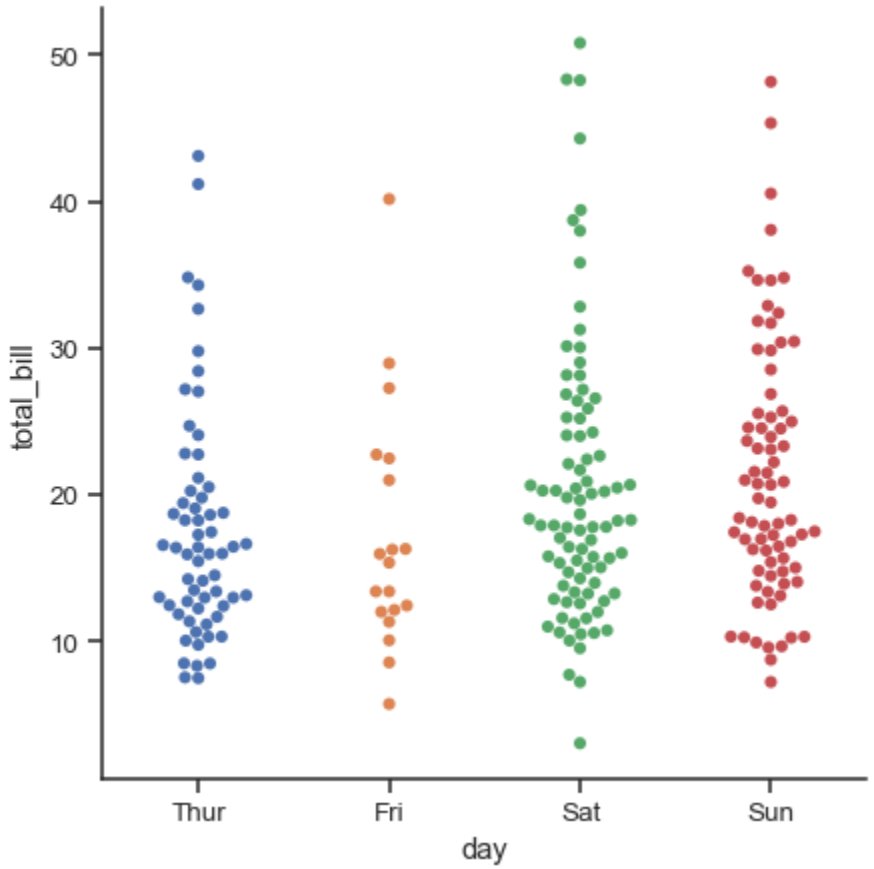
- sns.catplot(x="day", y="total_bill", jitter=False, data=tips)



定类散点图

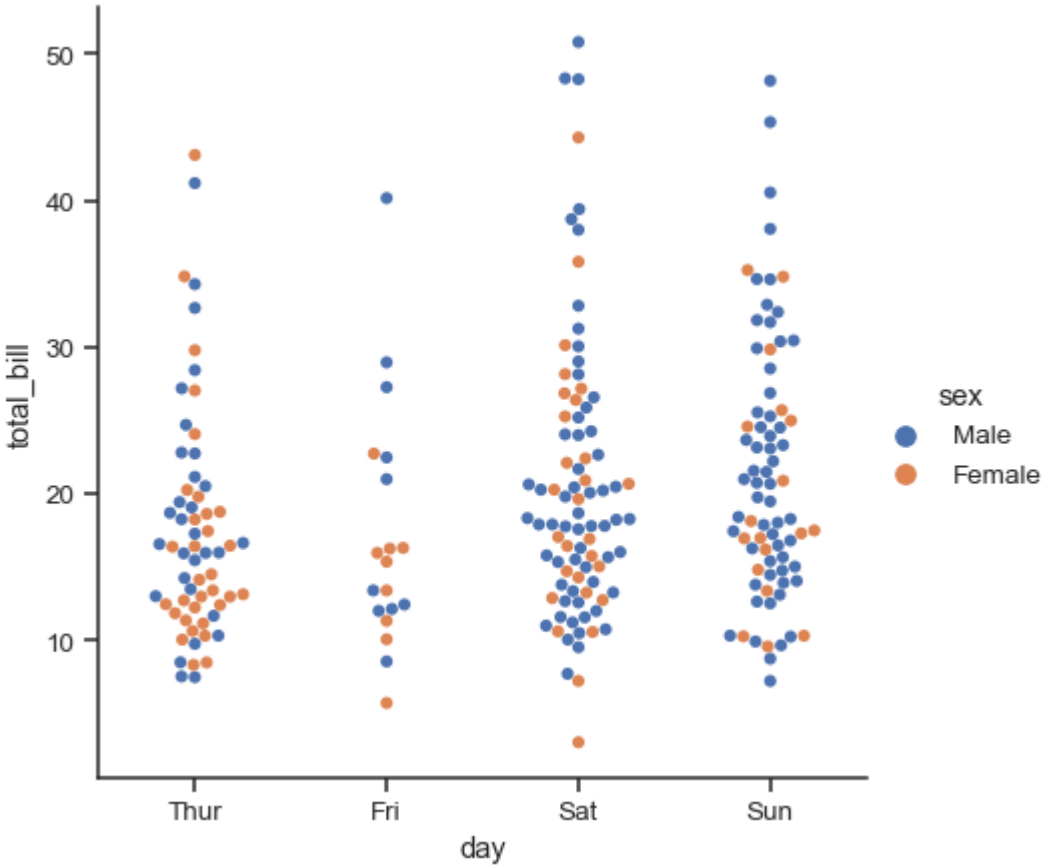
A categorical scatterplot with non-overlapping points

- sns.catplot(x="day", y="total_bill", kind="swarm", data=tips)



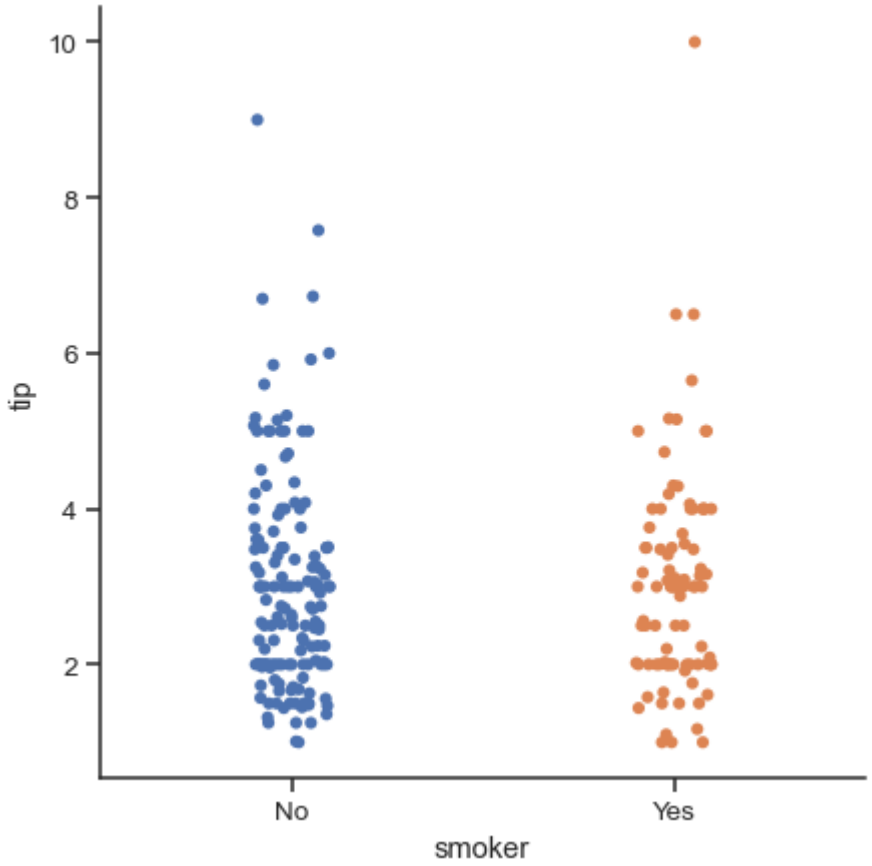
定类散点图

- sns.catplot(x="day", y="total_bill", hue="sex", kind="swarm", data=tips)



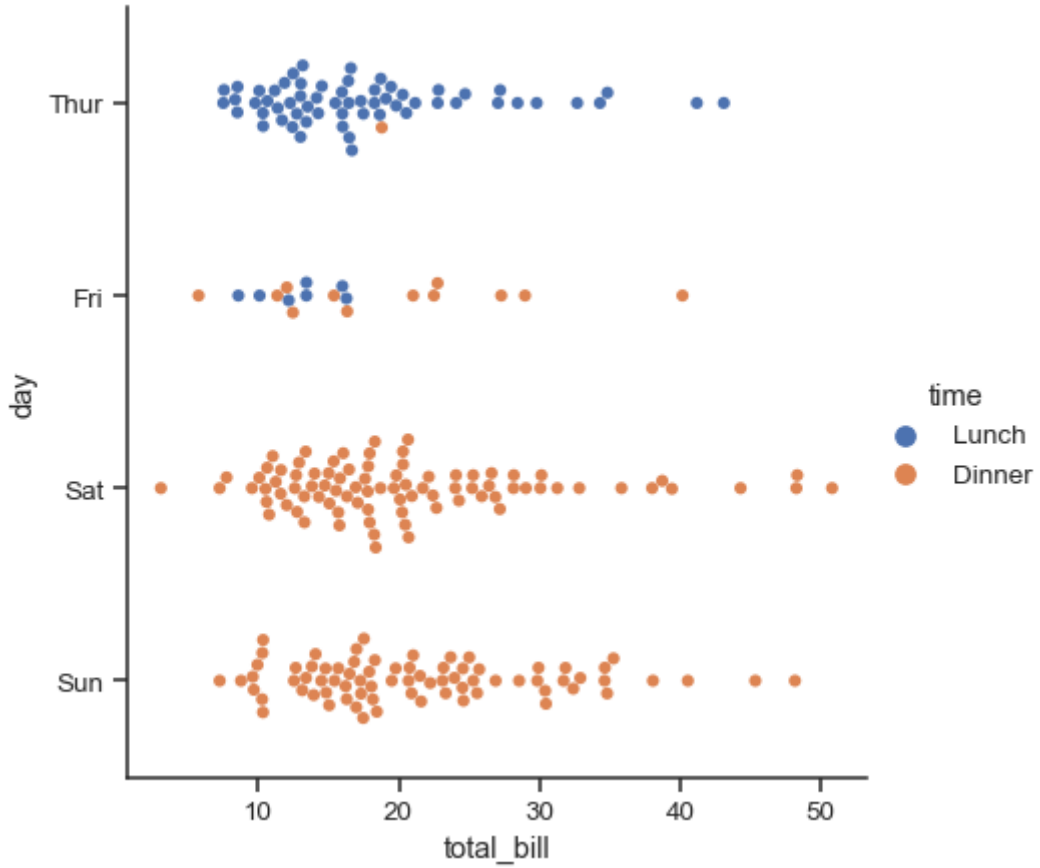
定类散点图

- sns.catplot(x="smoker", y="tip", order=["No", "Yes"], data=tips)



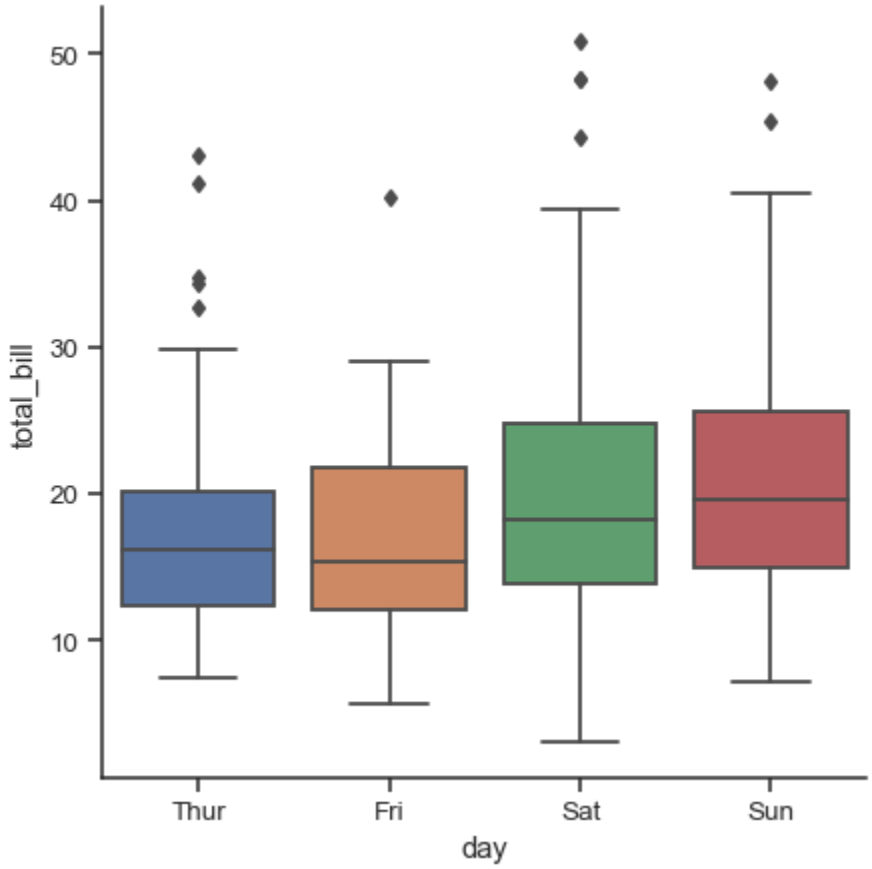
定类散点图

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



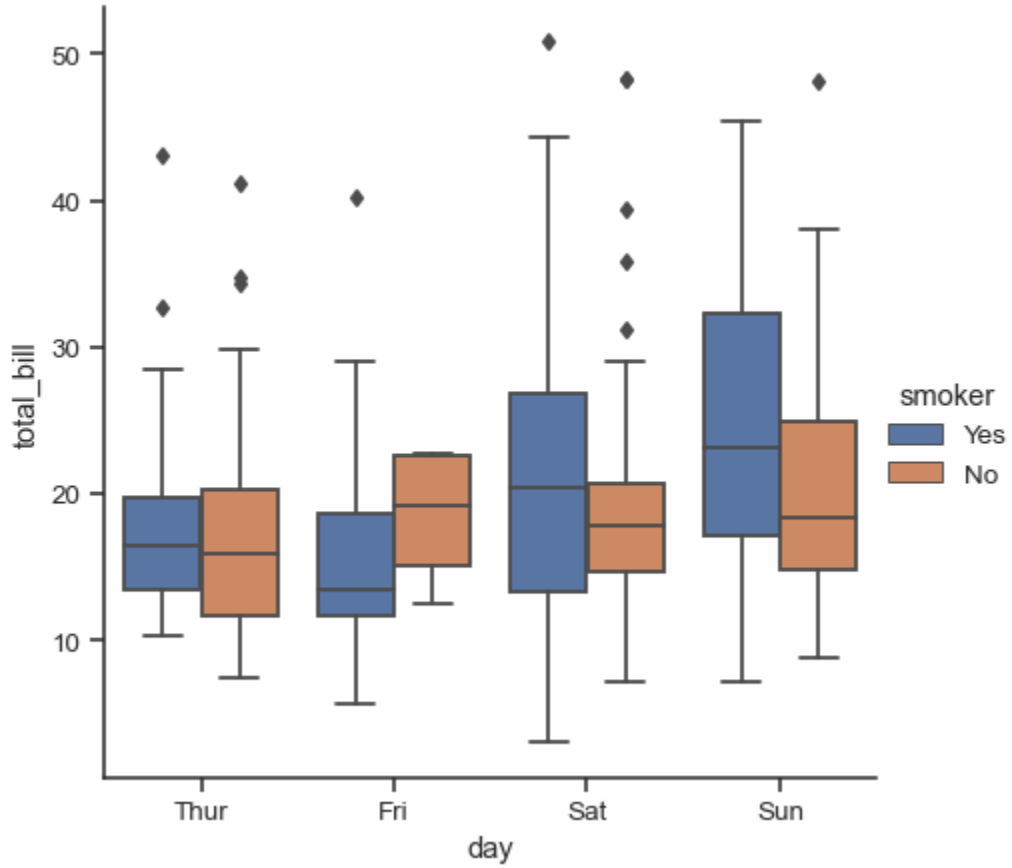
定类箱线图

- sns.catplot(x="day", y="total_bill", kind="box", data=tips)



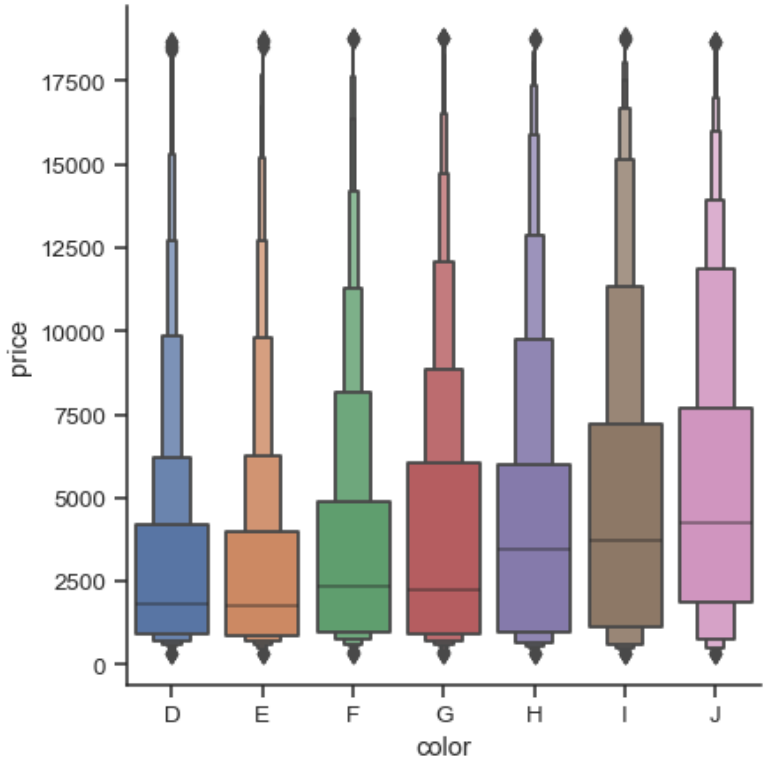
定类箱线图

- sns.catplot(x="day", y="total_bill", hue="smoker", kind="box", data=tips)



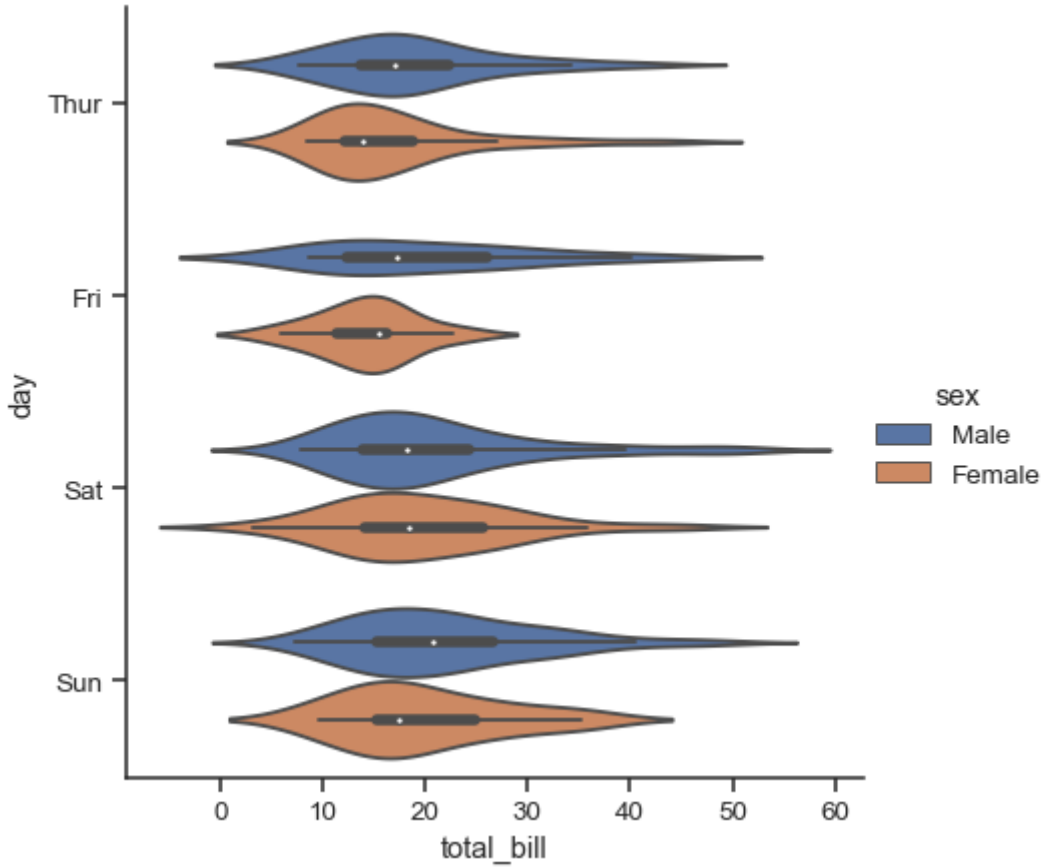
定类箱线图

- diamonds = sns.load_dataset("diamonds")
- sns.catplot(x="color", y="price", kind="boxen", data=diamonds.sort_values("color"))



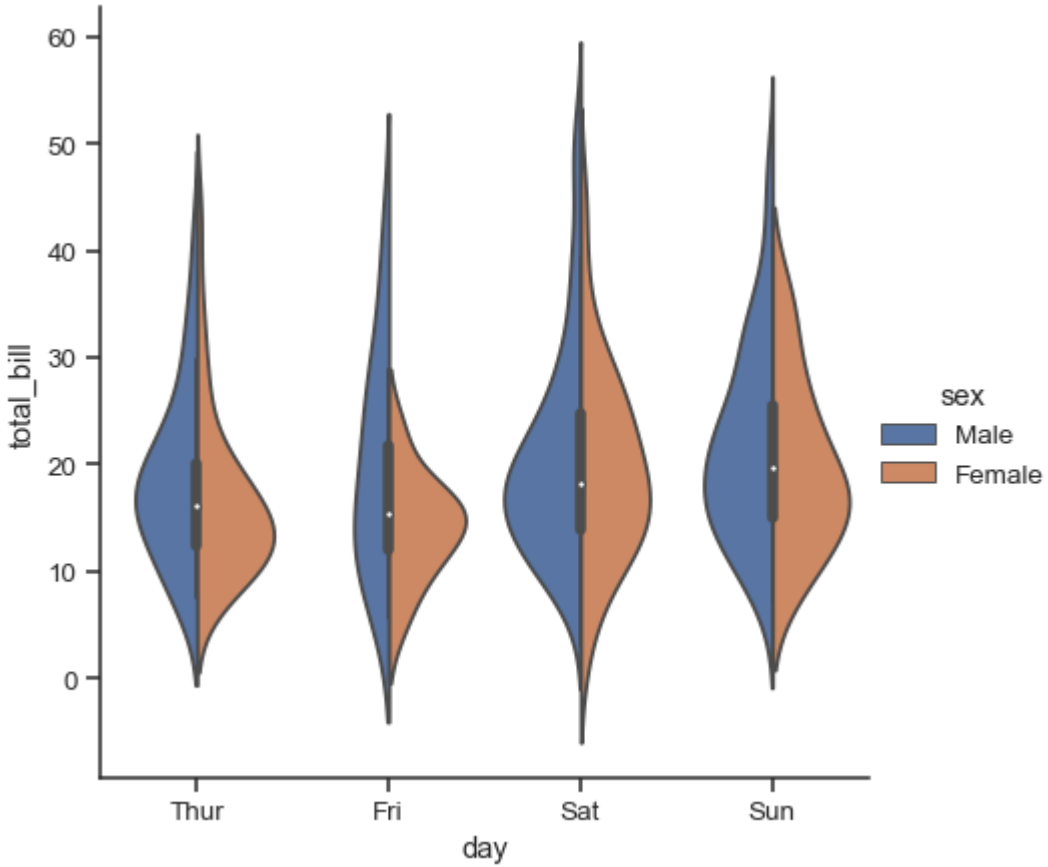
定类小提琴图

- sns.catplot(x="total_bill", y="day", hue="sex", kind="violin", data=tips)



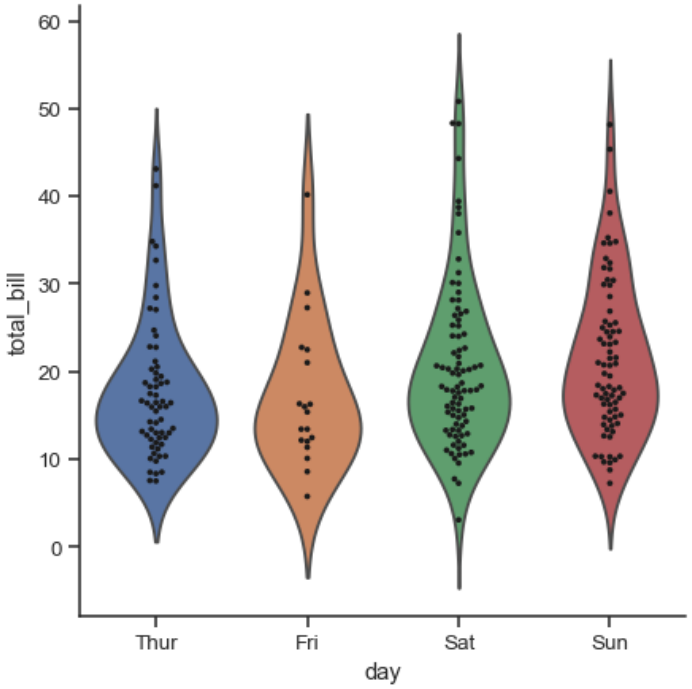
定类小提琴图

- sns.catplot(x="day", y="total_bill", hue="sex", kind="violin", split=True, data=tips)



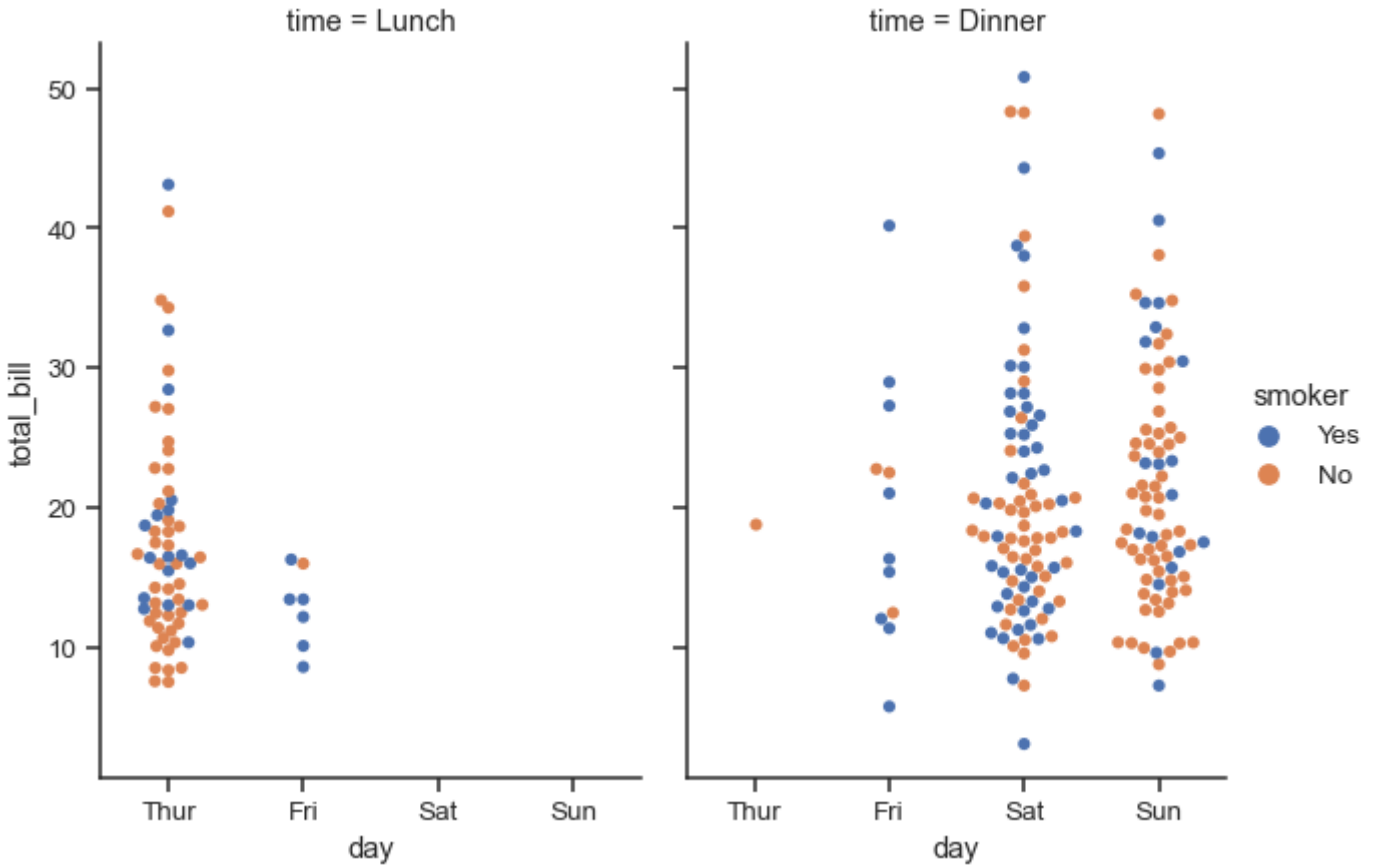
定类小提琴图

- 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)



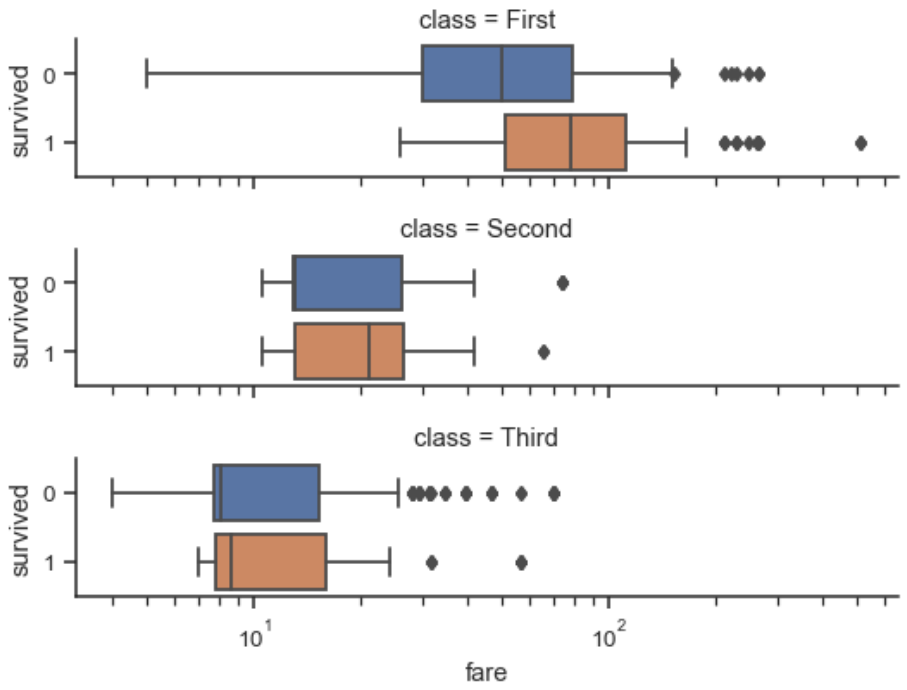
多个定类变量间的关系

- sns.catplot(x="day", y="total_bill", hue="smoker", col="time", aspect=.7, kind="swarm", data=tips)



多个定类变量间的关系

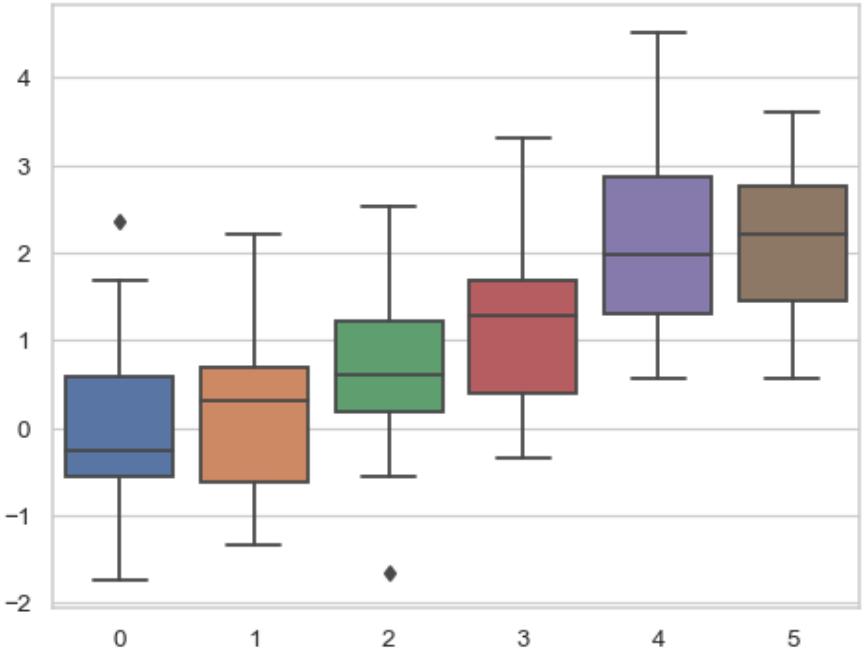
- g = sns.catplot(x="fare", y="survived", row="class", kind="box", orient="h", height=1.5, aspect=4, data=titanic.query("fare > 0"))
- g.set(xscale="log")



可视化中的美学因素

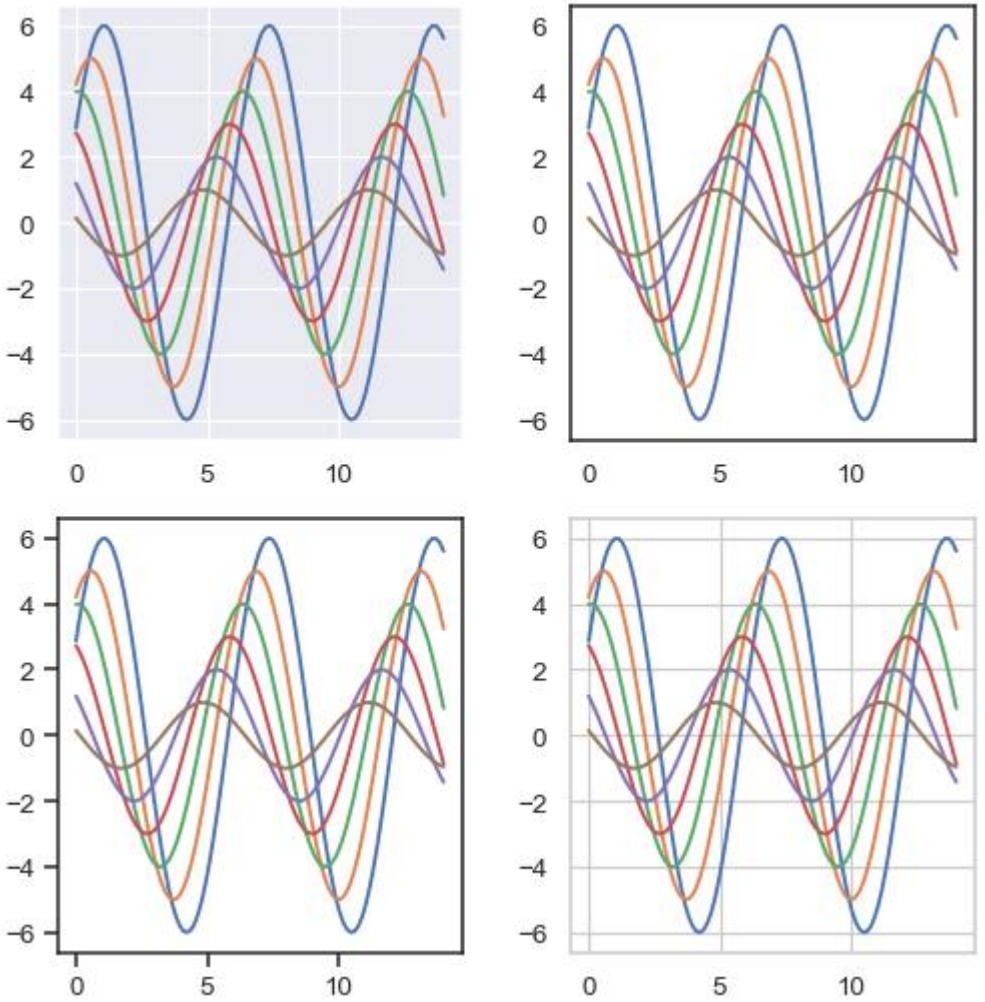
图的样式 (style)

- darkgrid, whitegrid, dark, white, ticks
  - sns.set_style("whitegrid")
  - data = np.random.normal(size=(20, 6)) + np.arange(6) / 2
  - sns.boxplot(data=data)



图的样式

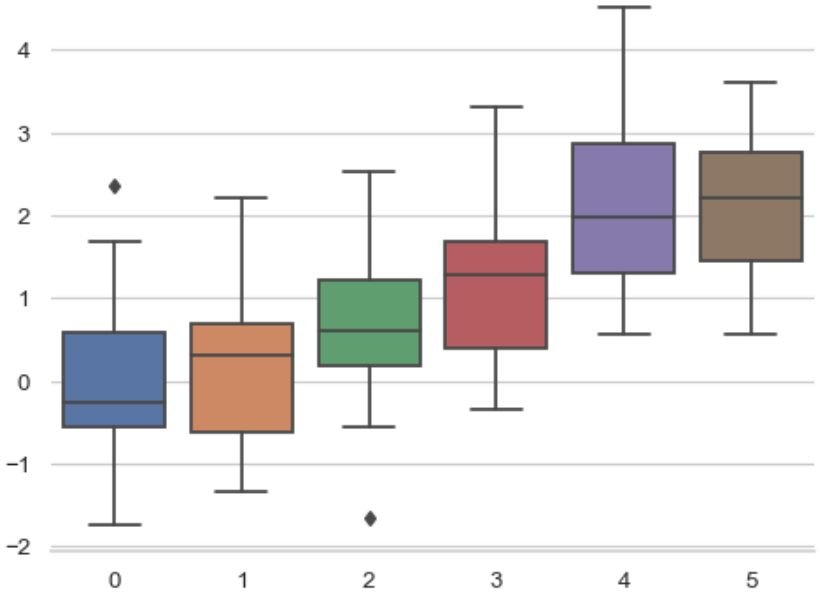
- f = plt.figure(figsize=(6, 6))
- gs = f.add_gridspec(2, 2)
- with sns.axes_style("darkgrid"):
- ax = f.add_subplot(gs[0, 0])
- sinplot()
- with sns.axes_style("white"):
- ax = f.add_subplot(gs[0, 1])
- sinplot()
- with sns.axes_style("ticks"):
- ax = f.add_subplot(gs[1, 0])
- sinplot()
- with sns.axes_style("whitegrid"):
- ax = f.add_subplot(gs[1, 1])
- sinplot()
- f.tight_layout()




```
def sinplot(flip=1):
    x = np.linspace(0, 14, 100)
    for i in range(1, 7):
        plt.plot(x, np.sin(x + i * .5) * (7 - i) * flip)
```

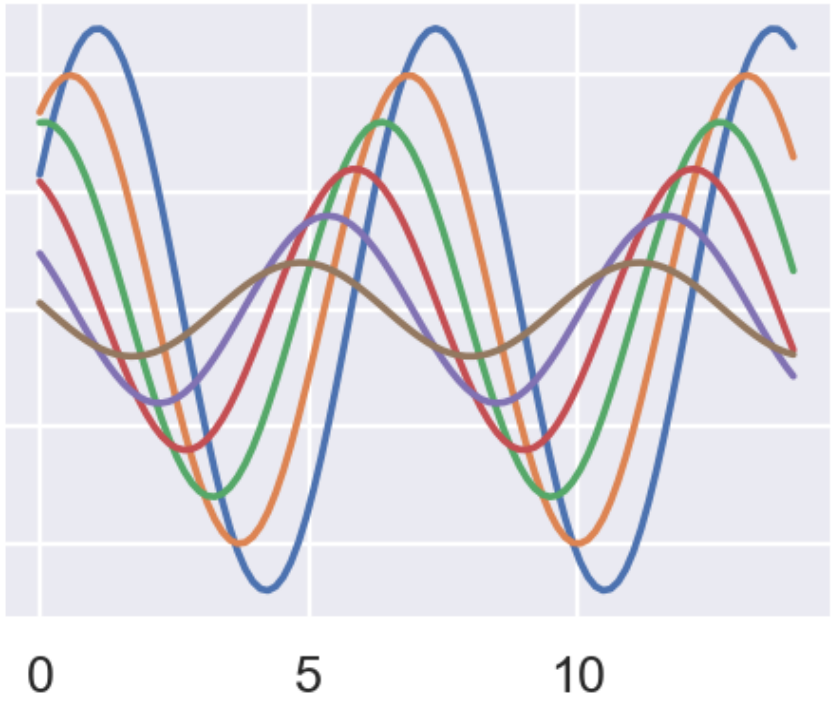
轴脊 (axes spines)

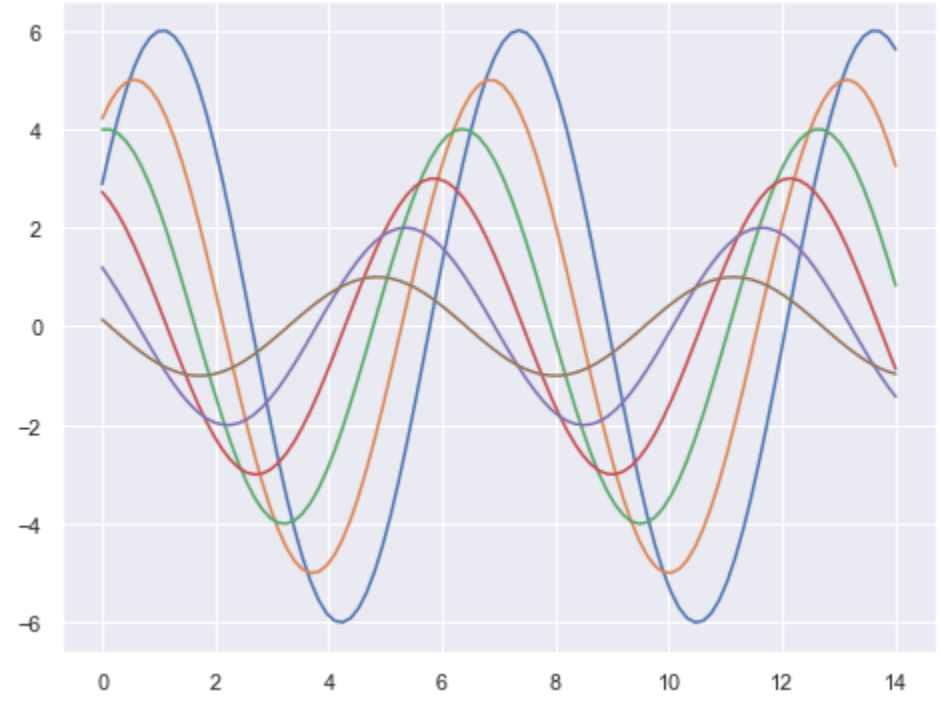
- sns.set_style("whitegrid")
- sns.boxplot(data=data, palette="deep")
- sns.despine(left=True)

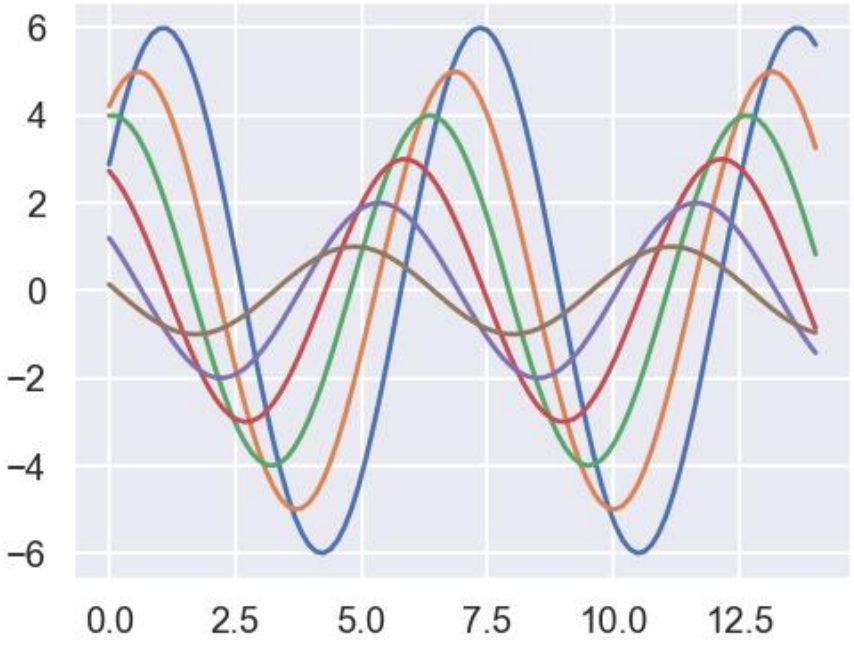


图的场景

- sns.set_context("paper")
- sns.set_context("talk")
- sns.set_context("poster")







布置个人作业2（数据可视化）

谢谢！