# **目录**

[目录 1](#_Toc966599153)

[1.普遍优化方法 2](#_Toc2065658563)

[1. 隐藏不必要的图形元素 2](#_Toc1302830939)

[2. 调整网格 2](#_Toc936326961)

[3. 设置图例和标题 2](#_Toc87068311)

[4. 优化标签 2](#_Toc920739370)

[5. 调整坐标轴刻度和字体 3](#_Toc99431308)

[6. 自定义颜色和样式 3](#_Toc399716190)

[7. 设置背景颜色 4](#_Toc701157514)

[8. 在数据点上添加标签 4](#_Toc1111566709)

[9. 添加误差条 4](#_Toc1141432910)

[10. 设置子图布局 4](#_Toc591499719)

[将多个图在一张图中显示（多图共存） 4](#_Toc633975270)

[11. 调整坐标轴范围 4](#_Toc1555990123)

[12. 添加色条（Colorbar） 4](#_Toc1617627742)

[13. 绘制填充区域 5](#_Toc326488774)

[14. 添加注释（Annotations） 5](#_Toc476106533)

[15.关于文字无法呈现 5](#_Toc398431409)

[2.特定图优化方法 6](#_Toc582679717)

[条形图\直方图 6](#_Toc572573299)

[热图 12](#_Toc365214086)

[散点图 14](#_Toc644880276)

[箱线图 15](#_Toc152832323)

[饼状图 17](#_Toc262410849)

[六边形分箱图（数据量大） 18](#_Toc1555211852)

[带回归的散点图 18](#_Toc1422128927)

[主图内插入子图 19](#_Toc227884979)

[上下堆叠 19](#_Toc1099499452)

[散点热力图 20](#_Toc190507329)

[3.需要注意的代表图： 20](#_Toc2106044473)

[4.总结 22](#_Toc1463987857)

[matplotlib 22](#_Toc1523768920)

[Seaborn 24](#_Toc1241747965)

[5.配色网站： 26](#_Toc811966209)

# 1.普遍优化方法

## 隐藏不必要的图形元素

plt.gca().spines['right'].set\_visible(False)

plt.gca().spines['top'].set\_visible(False)

隐藏四个边框

## 调整网格

可以为主刻度和次刻度设置网格，并自定义其样式。

可调整参数：

• True/False：控制是否显示网格。

• which：'major'（主刻度）或'minor'（次刻度）。

• linestyle：'-'（实线）、'--'（虚线）、'-.'（点划线）、':'（点线）等。

• linewidth：线宽，如0.5、0.7、1.0等。

• color：网格颜色，如'#c7c7c7'，也可用标准颜色名（如'black'、'red'等）。

## 设置图例和标题

图例位置：可以指定图例的位置以避免与图形内容重叠

plt.legend(loc='best')

标题样式：为图像添加标题并设置其字体大小和样式。

plt.title('Sample Title', fontsize=14, fontweight='bold')

可调整参数：

• loc：图例位置，如'best'、'upper right'、'lower left'等。

• fontsize：标题的字体大小，如12、14、16等。

• fontweight：字体粗细，如'normal'、'bold'等。

## 优化标签

**设置坐标轴标签**：自定义坐标轴的标签名称和字体大小。

**plt.xlabel('X Label', fontsize=12)**

**plt.ylabel('Y Label', fontsize=12)**

## 调整坐标轴刻度和字体

自定义刻度标签：可以自定义坐标轴的刻度标签和间隔。

plt.xticks(ticks=[0, 1, 2], labels=['A', 'B', 'C'], fontsize=10)

plt.yticks(fontsize=10)

可调整参数：

• ticks：刻度位置的列表。

• labels：刻度对应的标签列表。

• fontsize：刻度字体大小。

## 自定义颜色和样式

:• ’.’：点标记

• ’,’：像素点

• ‘o’：圆圈

• ‘v’：下三角形

• ’^’：上三角形

• ’<’：左三角形

• ’>’：右三角形

• ‘1’：下部花瓣

• ‘2’：上部花瓣

• ‘3’：左部花瓣

• ‘4’：右部花瓣

• ‘s’：正方形

• ‘p’：五边形（实心五角星）

• ’\*’：星形

• ‘h’：六边形1

• ‘H’：六边形2

• ’+’：加号

• ‘x’：x号

• ‘D’：菱形

• ‘d’：窄菱形

• ’|’：垂直线

• ’\_’：水平线

linestyle 控制网格线的样式。以下是常见的样式选项：

• ’-’ 或 ‘solid’：实线。

• ’–’ 或 ‘dashed’：虚线。

• ’-.’ 或 ‘dashdot’：点划线。

• ’:’ 或 ‘dotted’：点线。

## 7. 设置背景颜色

调整图形背景颜色：可以为图形或特定部分设置背景颜色。

plt.gca().set\_facecolor('#f0f0f0')

## 在数据点上添加标签

为数据点添加文本标签：在数据点上添加文字标签，以显示数值或其他相关信息。

for i in range(len(x)):

plt.text(x[i], y1[i], f'({x[i]}, {y1[i]:.2f})', fontsize=10, ha='right', va='bottom', color='#1f77b4')

## 添加误差条

显示数据点的误差条：可以在数据点上添加误差条，以表示数据的不确定性范围。

plt.errorbar(x, y, yerr=0.1, fmt='o', ecolor='black', capsize=5, elinewidth=2)

## 设置子图布局

**将多个图在一张图中显示（多图共存）**

fig, ax = plt.subplots(2, 2, figsize=(10, 8))

ax[0, 0].plot(x, y1)

ax[0, 1].plot(x, y2)

ax[1, 0].plot(x, y3)

ax[1, 1].plot(x, y4)

fig.tight\_layout()

## 调整坐标轴范围

设置坐标轴的范围：可以手动设置坐标轴的最小值和最大值，以聚焦图表的特定部分。

plt.xlim(0, 10)

plt.ylim(-1, 1)

## 添加色条（Colorbar）

为热图或其他图形添加色条：显示图像中颜色对应的数值范围。

plt.imshow(data, cmap='viridis')

plt.colorbar()

可调整参数：

• cmap：颜色映射，如'viridis'、'plasma'、'inferno'等。

• colorbar：显示色条，并可以通过shrink参数调整其大小。

## 绘制填充区域

在曲线下方绘制填充区域：为曲线图添加阴影填充区域，以强调数据变化区域。

plt.fill\_between(x, y1, y2, color='lightblue', alpha=0.5)

可调整参数：

• color：填充颜色，如'lightblue'。

• alpha：填充透明度，取值范围为0（完全透明）到1（不透明）。

## 14. 添加注释（Annotations）

在图表中特定位置添加注释：可以在图表中添加注释来解释数据点或突出显示特定信息

plt.annotate('Important Point', xy=(2, 1), xytext=(3, 1.5),

arrowprops=dict(facecolor='black', shrink=0.05),

fontsize=12, color='red')

## 15.关于文字无法呈现

macos：

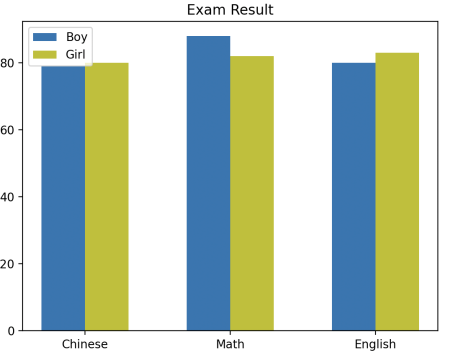
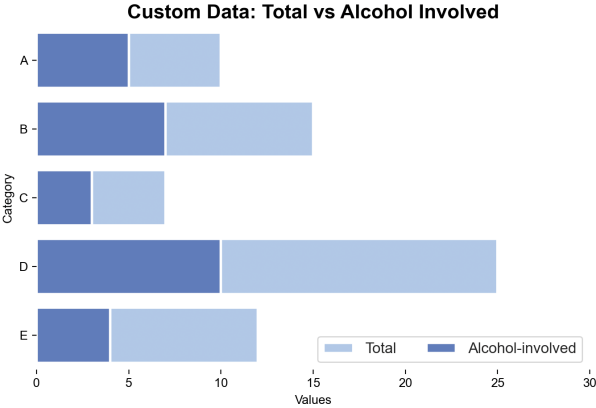
plt.rcParams['font.sans-serif'] = ['PingFang HK'] # 使用 macOS 默认的中文字体  
plt.rcParams['axes.unicode\_minus'] = False # 用来正常显示负号

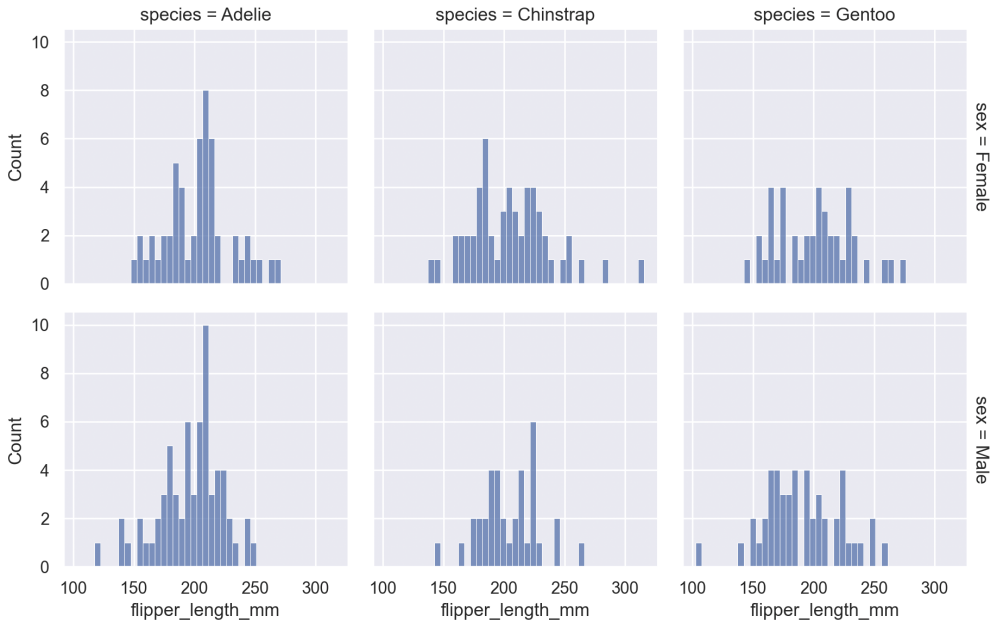
win：plt.rcParams['font.sans-serif'] = ['SimHei'] # 设置字体为黑体  
plt.rcParams['axes.unicode\_minus'] = False # 用来正常显示负号

# 2.特定图优化方法

## 条形图\直方图

分组条形图、单条形图、水平条形图、多面板等





sns.barplot() 函数

sns.displot（）：多面板直方图

barplot() 用于绘制条形图，以下是可以配置的参数：

x 和 y 参数

• 功能：指定用于 x 轴和 y 轴的数据列。

• 可选值：

• 字符串：数据框中的列名，如 x="total"，y="abbrev"

hue 参数

• 功能：根据类别变量为不同的条形图指定颜色。

• 可选值：

• 字符串：数据框中的列名

order 参数

• 功能：指定类别变量的顺序。

• 可选值：

• 列表：如 ["A", "B", "C"]

orient 参数

• 功能：指定条形图的方向。

• 可选值：

• 'v'：垂直方向

• 'h'：水平方向（默认）

其他美化配置

sns.set\_color\_codes() 函数

• 功能：设置颜色代码的风格。

• 可选值：

• "pastel"：浅色调

• "muted"：暗色调

sns.set\_theme() 函数

• 功能：设置图表的主题风格。

• 可选值：

• "whitegrid"：白色网格背景

• "darkgrid"：深色网格背景

• "white"：白色背景，无网格

• "dark"：深色背景，无网格

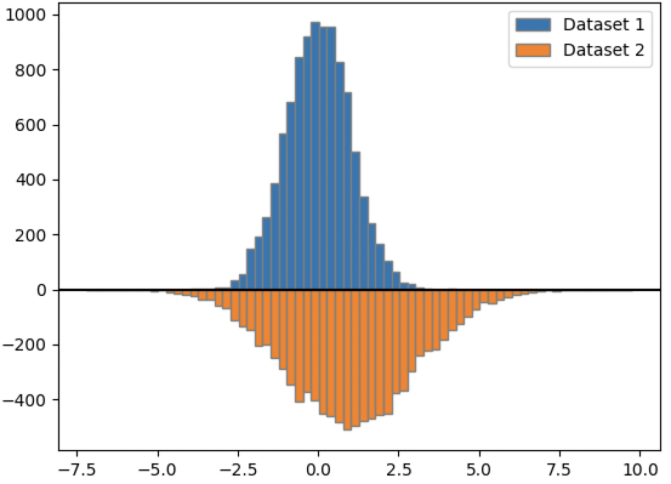
sns.despine() 函数

• 功能：移除图表的顶部和右侧边框，以简化图表的外观。

• 可选值：

• 布尔值：如 left=True 表示移除左侧边框，bottom=True 表示移除底部边框

双直方图



bins 参数

• 功能：定义直方图的边界，决定每个柱状条的宽度和数量。

• 可选值：

• 整数：如 10，表示划分为 10 个等宽区间

• 序列：如 np.arange(min\_val, max\_val, step)，表示自定义的边界值序列

• 'auto'：自动选择最优的划分方式

weights 参数

• 功能：设置每个数据点的权重，用于控制直方图的高度。

• 可选值：

• 数组：如 np.ones\_like(dataset)，表示所有数据点权重相等

• 负值数组：如 -np.ones\_like(dataset)，用于绘制反向直方图（负值表示向下的柱状条）

color 参数

• 功能：设置直方图柱状条的颜色。

• 可选值：

• 颜色名称：如 'blue'、'green'

• HEX颜色代码：如 '#1f77b4'

label 参数

• 功能：为直方图添加标签，用于图例显示。

• 可选值：

• 字符串：如 "Dataset 1"、"Dataset 2"

alpha 参数

• 功能：设置直方图的透明度。

• 可选值：

• 介于 0 到 1 之间的浮点数

histtype 参数

• 功能：控制直方图的类型。

• 可选值：

• 'bar'：标准条形直方图

• 'step'：仅绘制轮廓

• 'stepfilled'：填充的轮廓直方图

edgecolor 参数

• 功能：设置直方图柱状条的边框颜色。

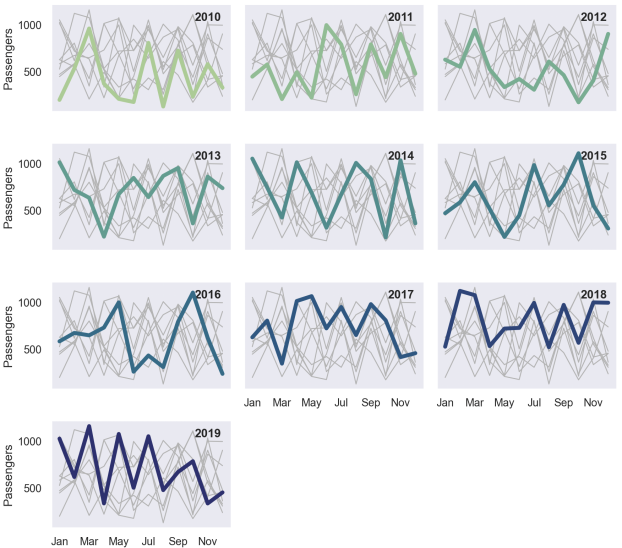
• 可选值：

• 颜色名称：如 'black'

• HEX颜色代码：如 '#000000'

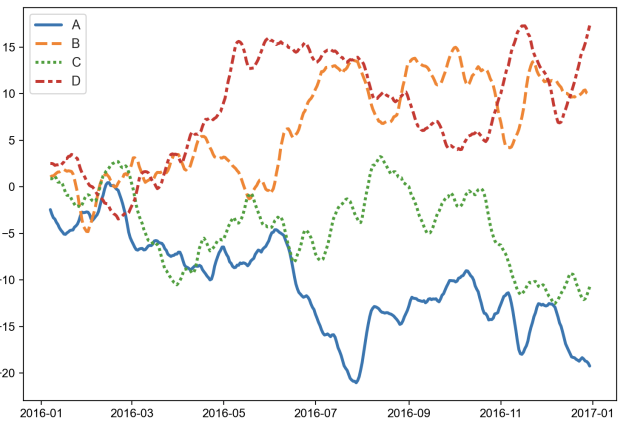
线图：

时间序列图



折线图

**sns.lineplot() 函数**



palette 参数

• 功能：设置用于颜色映射的调色板。

• 可选值：

• Seaborn 提供的调色板名称，如 "tab10"（一个经典的 10 色调色板）

• 其他内置调色板：如 "viridis"、"coolwarm"、"Blues"

• 自定义颜色列表：如 ["#ff9999", "#66b3ff", "#99ff99", "#ffcc99"]

linewidth 参数

• 功能：设置线条的宽度。

• 可选值：

• 浮点数值：如 linewidth=2.5 表示线宽为 2.5 像素。

hue 参数（未在此示例中使用，但常见）

• 功能：根据类别变量为不同的线条指定颜色。

• 可选值：

• 字符串：数据框中的列名

style 参数（未在此示例中使用，但常见）

• 功能：根据类别变量为不同的线条指定样式（如虚线、点线）。

• 可选值：

• 字符串：数据框中的列名

markers 参数（未在此示例中使用，但常见）

• 功能：设置线条上的标记点样式。

• 可选值：

• 布尔值或字符：如 True 表示显示标记点，或者使用 'o' 显示圆形标记点。

其他美化配置

sns.set\_theme() 函数

• 功能：设置图表的主题风格。

• 可选值：

• "whitegrid"：白色网格背景，适合数据对比分析

• "darkgrid"：深色网格背景

• "white"：白色背景，无网格

• "dark"：深色背景，无网格

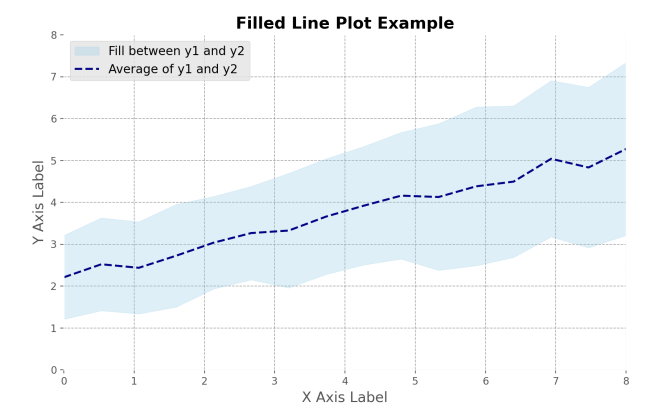
f, ax = plt.subplots(figsize=(9, 6)) 函数

• 功能：创建一个图形和轴对象，并设置图形的大小。

• 可选值：

• figsize=(9, 6)：设置图形大小为 9x6 英寸，调整图表的纵横比和占据的空间

填充折线图

****

ax.fill\_between() 函数

ax.fill\_between() 用于在两条线之间填充颜色区域。以下是可以配置的参数：

x 参数

• 功能：定义 x 轴上的坐标点。

• 可选值：

• 一维数组或序列：如 np.linspace(0, 8, 16)

y1 和 y2 参数

• 功能：定义在 y 轴上的两个数据集，填充区域将在这两条线之间生成。

• 可选值：

• 一维数组或序列：如 y1 = 3 + 4\*x/8 + np.random.uniform(0.0, 0.5, len(x))

alpha 参数

• 功能：设置填充区域的透明度。

• 可选值：

• 介于 0 到 1 之间的浮点数，例如 alpha=0.5 表示 50% 透明度

color 参数

• 功能：设置填充区域的颜色。

• 可选值：

• 颜色名称：如 'blue'、'green'

• HEX 颜色代码：如 '#1f77b4'

linewidth 参数

• 功能：设置填充区域边界线的宽度。

• 可选值：

• 浮点数值：如 0 表示没有边界线

where 参数

• 功能：定义在哪些区域进行填充。

• 可选值：

• 布尔数组：如 where=(y1 > y2) 表示仅在 y1 > y2 的区域填充

interpolate 参数

• 功能：如果为 True，则在填充之前将 x 坐标内插到多个点，以保证 y1 和 y2 交叉时填充区域的正确性。

• 可选值：

• True 或 False

ax.plot() 函数

ax.plot() 用于绘制线条，以下是可以配置的参数：

x 参数

• 功能：定义 x 轴上的坐标点。

• 可选值：

• 一维数组或序列：如 np.linspace(0, 8, 16)

y 参数

• 功能：定义 y 轴上的数据集。

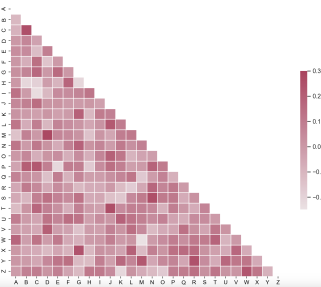
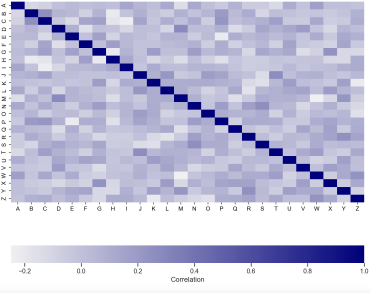
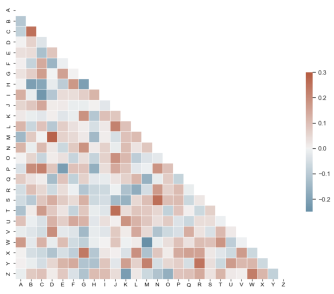
• 可选值：

• 一维数组或序列：如 (y1 + y2)/2

面积折线图

## 热图

可视化矩阵图



sns.heatmap() 函数用于绘制热力图

mask 参数

• 功能：设置遮罩区域，遮罩部分将不显示在热力图中。

• 可选值：

• 布尔矩阵：如 np.triu(np.ones\_like(corr, dtype=bool))，用于遮罩矩阵的上三角部分

cmap 参数

• 功能：设置热力图的颜色映射。

• 可选值：

• sns.diverging\_palette() 或 sns.light\_palette() 生成的颜色映射

• 其他常用颜色映射：如 plt.cm.coolwarm、plt.cm.viridis

vmax 和 vmin 参数

• 功能：设置颜色映射的最大值和最小值，以控制颜色范围。

• 可选值：

• 浮点数值：如 vmax=0.3，vmin=-0.3

center 参数

• 功能：设置颜色映射的中心点，即颜色渐变的中点值。

• 可选值：

• 浮点数值：如 center=0

square 参数

• 功能：控制每个单元格的形状是否为正方形。

• 可选值：

• True：使单元格为正方形

• False：使用默认的矩形单元格

linecolor 参数

• 功能：设置单元格之间的线条颜色。

• 可选值：

• 颜色名称或 HEX 颜色代码：如 linecolor='black'

annot 参数

• 功能：在单元格中显示数据值。

• 可选值：

• True：显示数据值

• False：不显示数据值（默认）

fmt 参数

• 功能：设置单元格中显示数据的格式。

• 可选值：

• 字符串格式：如 fmt=".2f" 表示保留两位小数

cbar\_kws 参数

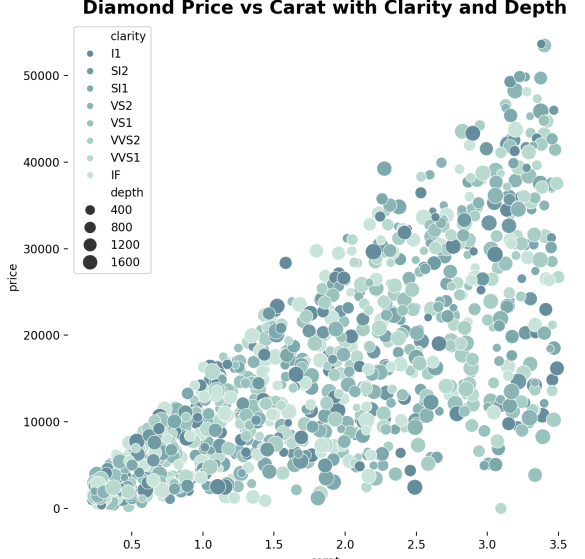
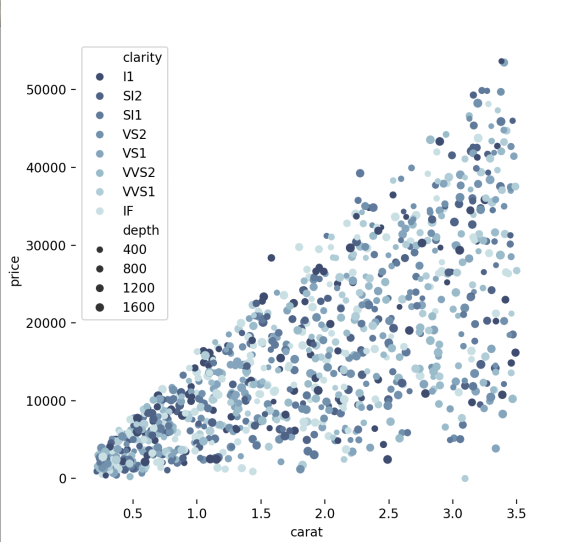
• 功能：设置颜色条的属性。

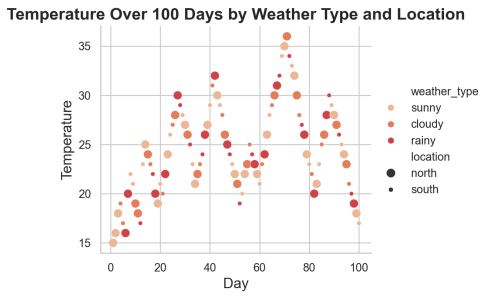
• 可选值：

• 字典：如 {"orientation": "horizontal", "label": "Correlation"}，用于控制颜色条的方向和标签

sns.light\_palette("#95e1d3", as\_cmap=True)#配色自己找喜欢的配色

## 散点图





sns.scatterplot() 函数

hue 参数

• 功能：根据类别变量为不同的点指定颜色。

• 可选值：

• 字符串：数据框中的列名，如 hue="clarity"

size 参数

• 功能：根据数值变量为不同的点指定大小。

• 可选值：

• 字符串：数据框中的列名，如 size="depth"

palette 参数

• 功能：设置用于颜色映射的调色板。

• 可选值：

• Seaborn 提供的调色板名称，如 "ch:r=-.2,d=.3\_r"（一个渐变的颜色调色板）

• 其他内置调色板：如 "viridis"、"coolwarm"、"Blues" 等

• 自定义颜色列表：如 ["#ff9999", "#66b3ff", "#99ff99", "#ffcc99"]

hue\_order 参数

• 功能：指定类别变量的顺序。

• 可选值：

• 列表：如 ["I1", "SI2", "SI1", "VS2", "VS1", "VVS2", "VVS1", "IF"]，用于对 hue 变量进行排序

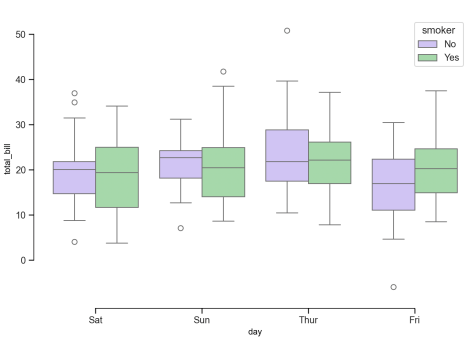
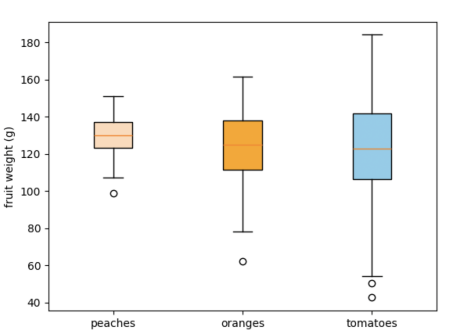
sizes 参数

• 功能：设置点的最小和最大大小。

• 可选值：

• 元组：如 sizes=(20, 50) 表示点的大小范围从 20 到 50

## 箱线图



ax.boxplot() 函数

ax.boxplot() 用于绘制箱线图，以下是可以配置的参数：

x 参数

• 功能：定义要绘制箱线图的数据集，可以是一个列表或数组。

• 可选值：

• 多个数组或列表：如 [array1, array2, array3]

patch\_artist 参数

• 功能：控制是否填充箱线图的内部颜色。

• 可选值：

• True：填充颜色

• False：不填充颜色（默认）

vert 参数

• 功能：控制箱线图的方向。

• 可选值：

• True：垂直方向（默认）

• False：水平方向

widths 参数

• 功能：设置箱线图的宽度。

• 可选值：

• 浮点数值或数组：如 0.5，或者 [0.4, 0.5, 0.6] 表示每个箱线图的宽度

notch 参数

• 功能：是否在箱线图中显示凹口，以便更好地比较中位数。

• 可选值：

• True：显示凹口

• False：不显示凹口（默认）

meanline 参数

• 功能：是否用线表示均值。

• 可选值：

• True：使用线表示均值

• False：使用点表示均值（默认）

showmeans 参数

• 功能：是否在箱线图中显示均值标记。

• 可选值：

• True：显示均值

• False：不显示均值（默认）

showfliers 参数

• 功能：是否显示异常值（离群点）。

• 可选值：

• True：显示异常值（默认）

• False：不显示异常值

showbox 参数

• 功能：是否显示箱体。

• 可选值：

• True：显示箱体（默认）

• False：不显示箱体

showcaps 参数

• 功能：是否显示箱线图顶端和底端的线帽。

• 可选值：

• True：显示线帽（默认）

• False：不显示线帽

定制箱线图的颜色

在代码中使用了 patch.set\_facecolor(color) 来设置每个箱线图的填充颜色。以下是一些相关配置：

set\_facecolor() 方法

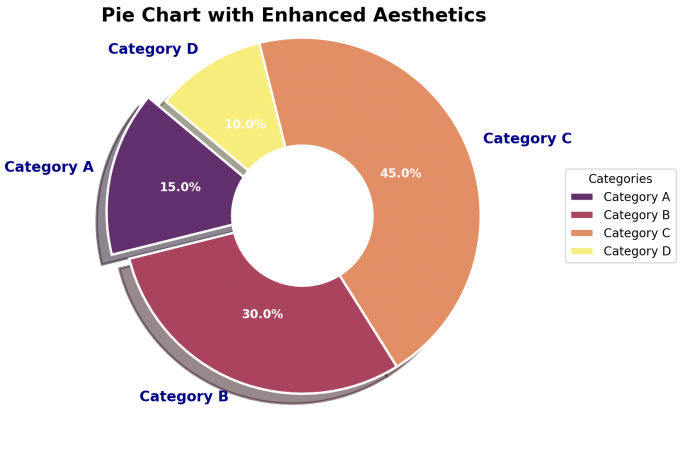
• 功能：设置箱线图内部的填充颜色。

• 可选值：

• 颜色名称：如 'peachpuff'、'orange'、'skyblue'

• HEX 颜色代码：如 '#FF6347'、'#4682B4'

## 饼状图

****

ax.pie() 函数

explode 参数：列表或数组：如 (0.1, 0, 0, 0) 表示第一个切片分离，其他切片不分离

autopct 参数

• 功能：设置切片上显示的百分比格式。

• 可选值：

• 字符串格式：如 '%1.1f%%' 表示保留小数点后一位

shadow 参数

• 功能：是否为饼图添加阴影效果。

• 可选值：

• True：显示阴影

• False：不显示阴影（默认）

startangle 参数

• 功能：设置饼图的起始角度（从x轴正方向逆时针旋转）。

• 可选值：

• 浮点数值：如 150 表示起始角度为150度

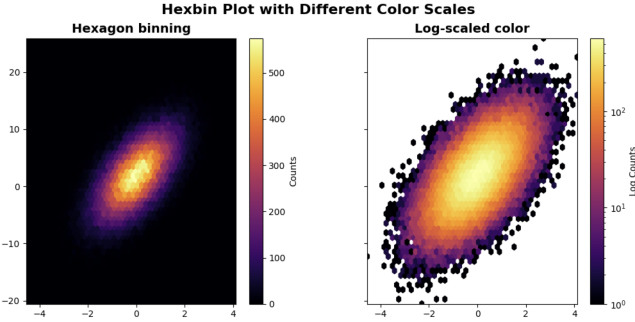
wedgeprops 参数

• 功能：设置每个切片的属性，如边框颜色。

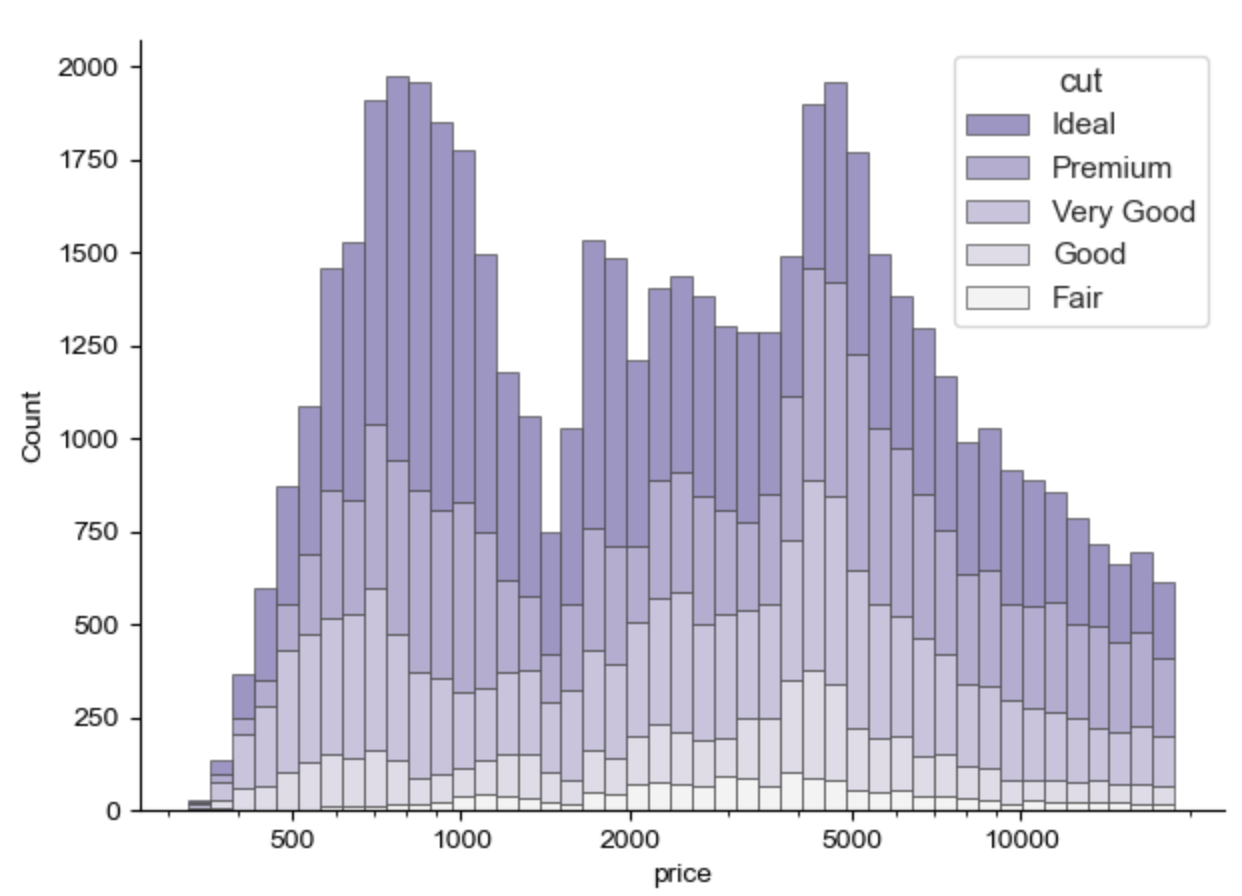
• 可选值：

• 字典：如 dict(edgecolor='w') 表示设置白色边框

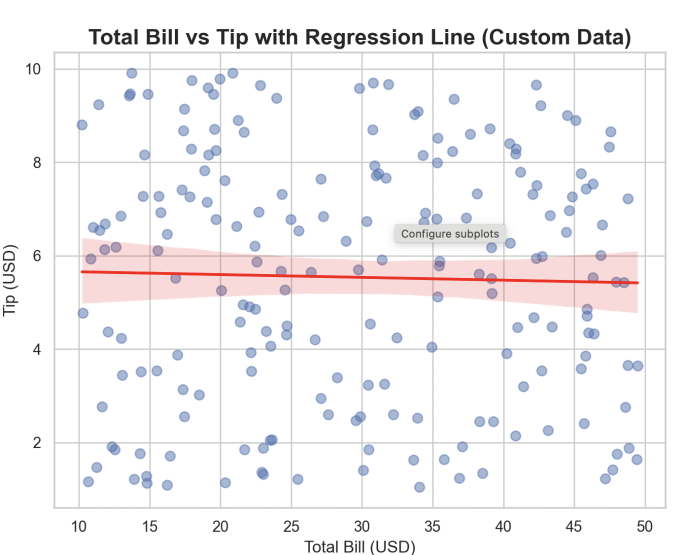
## 六边形分箱图（数据量大）



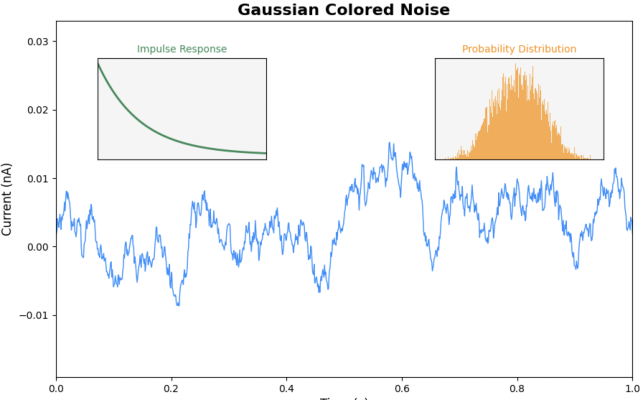
**对数堆叠直方图（数据量大）**



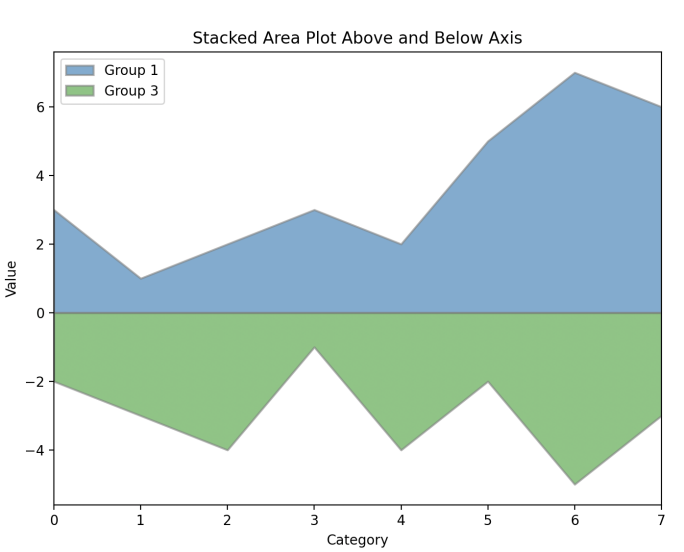
带回归的散点图：

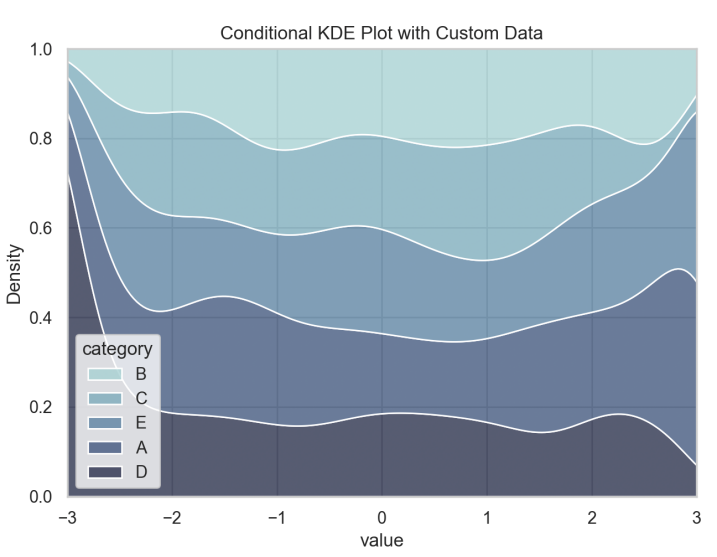
****

## 主图内插入子图

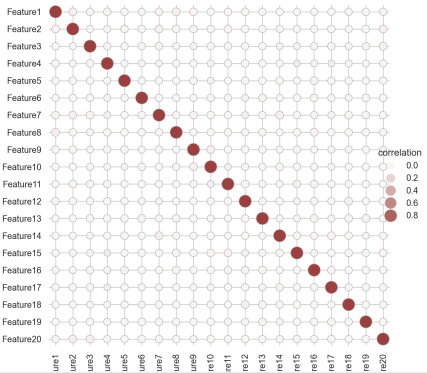
****

上下堆叠**：**

****

****

散点热力图**：**

****

# 3.需要注意的代表图：

**90%的图分析均为折线\曲线+条形图+热力图，其他图大多数也是散点、曲线、折线面积图的组合，改起来大同小异，配色美观是展示的关键**

**聚类分析**：Kmeans++、Q，R：聚类图、散点、肘部法则图

Spearman ，Pearson等：热力图

**Hex bin：数据量大**

**多元线性回归**：

残差图 (Residual Plot)： 纵轴为残差，横轴为预测值，展示模型拟合的好坏。

• 散点图 (Scatter Plot)： 在2D或3D空间中显示独立变量与因变量之间的关系。

• 回归系数图 (Coefficient Plot)： 展示每个独立变量的回归系数，通常以条形图的形式

**支持向量机 (SVM)**：分类边界图 (Decision Boundary Plot)： 显示SVM分类器的决策边界及支持向量的位置，常与散点图结合使用。

• 学习曲线图 (Learning Curve Plot)： 展示训练集和验证集上的误差随训练集大小的变化情况。

• 核函数影响图 (Kernel Influence Plot)： 通过可视化展示不同核函数对分类结果的影响。

时间序列预测

时间序列图 、自相关图 (ACF Plot) 和 偏自相关图 (PACF Plot)、残差诊断图 (Residual Diagnostic Plot)

**DBSCAN**

聚类结果图 (Cluster Result Plot)：显示数据点如何根据密度分布聚类，不同的颜色表示不同的簇，噪声点通常以不同的标记或颜色表示。

• k-距离图 (k-distance Plot)：用于帮助选择最合适的eps参数，通过绘制每个点到其第k个最近邻点的距离，来识别拐点。

散点图+折线

**动态规划**

手画

**PCA**

差解释率图 (Scree Plot)：展示每个主成分解释的方差比例，帮助确定保留多少个主成分。

• 主成分散点图 (PCA Scatter Plot)：在二维或三维空间中显示数据在前两个或三个主成分上的投影，帮助观察数据的分布情况和分组模式。

• 累计解释方差图 (Cumulative Explained Variance Plot)：展示累积的解释方差比例，通常用于决定保留的主成分数量。

没啥说的，就几个散点图

**Fisher判别**

• 判别边界图 (Discriminant Boundary Plot)：展示不同类别的决策边界和数据点的分布情况。--这个比较关键 类似于SVM

• 散点图 (Scatter Plot)：展示数据在投影到Fisher判别方向上的分布，通常在二维空间中显示不同类别的数据点及其分类结果。

• 投影直方图 (Projection Histogram)：显示不同类别的数据点在Fisher判别方向上的投影值分布，帮助判断分类效果。

**LSTM 长短期记忆网络**：

损失曲线图 (Loss Curve Plot)： 展示训练和验证的损失随时间的变化。

• 预测 vs 真实值图 (Prediction vs Actual Plot)： 比较模型预测的值与实际的值。

• 时间步长注意力图 (Attention Weight Plot, 如果有使用注意力机制)： 可视化LSTM对不同时间步长的关注度

**0-1 规划 (Binary Integer Programming)**：

优化过程可视化 (Optimization Process Visualization)： 展示求解过程中的目标函数值变化情况。

• 可行解空间图 (Feasible Region Plot)： 显示不同约束条件下的可行解空间，通常在简单的2D情况下使用。

**贝叶斯判别**

决策边界图 (Decision Boundary Plot)：展示贝叶斯分类器的决策边界，通常与散点图结合使用，显示不同类别的数据点及其分类结果。

• 后验概率图 (Posterior Probability Plot)：展示数据点属于每个类别的后验概率分布，可以帮助理解模型的分类置信度。

• 混淆矩阵图 (Confusion Matrix Plot)：展示分类结果的混淆矩阵，显示真实类别与预测类别的对应关系。---没啥说的，热力图

抽象图

物理图、流程图、思维导图等

# 4.总结

## matplotlib

**基础绘图函数**

1. plot(): 绘制线图或散点图。

2. scatter(): 绘制散点图。

3. bar(): 绘制柱状图。

4. barh(): 绘制水平柱状图。

5. hist(): 绘制直方图。

6. pie(): 绘制饼图。

7. boxplot(): 绘制箱线图。

8. violinplot(): 绘制小提琴图。

9. stem(): 绘制茎叶图。

10. step(): 绘制阶梯图。

**图像与热图**

11. imshow(): 显示图像或矩阵数据。

12. matshow(): 专门用于显示矩阵数据的图像。

13. pcolormesh(): 创建伪彩色图像，用于二维数组的数据。

14. pcolor(): 绘制二维数组的伪彩色图（较 pcolormesh 更慢但更灵活）。

15. contour(): 绘制等高线图。

16. contourf(): 绘制填充的等高线图。

17. hexbin(): 绘制六边形箱图（Hexbin plot），用于显示点密度。

**高级绘图函数**

18. errorbar(): 绘制带有误差条的线图。

19. fill(): 绘制填充的多边形。

20. fill\_between(): 在两条水平线之间填充颜色。

21. fill\_betweenx(): 在两条垂直线之间填充颜色。

22. stackplot(): 绘制堆叠面积图。

23. eventplot(): 绘制事件图，用于表示时间线上的事件。

**统计函数**

24. boxplot(): 绘制箱线图。

25. violinplot(): 绘制小提琴图。

26. hist2d(): 绘制二维直方图。

27. density(): 绘制密度图（通常与 seaborn 结合使用）。

**三维绘图函数（需要 mpl\_toolkits.mplot3d）**

28. plot\_surface(): 绘制三维表面图。

29. plot\_wireframe(): 绘制三维线框图。

30. plot\_trisurf(): 绘制三角表面图。

31. scatter3D(): 绘制三维散点图。

32. contour3D(): 绘制三维等高线图。

**注释与标记**

33. annotate(): 添加注释文本。

34. text(): 添加文本标签。

35. arrow(): 绘制箭头。

**36. quiver(): 绘制二维箭头场。**

**极坐标和极轴图**

37. polar(): 绘制极坐标图。

38. polarplot(): 在极坐标系中绘制线图。

39. rose(): 绘制玫瑰图（直方图的极坐标版本）。

**条形图与误差条**

40. broken\_barh(): 绘制带有破折线的水平条形图。

41. errorbar(): 绘制带误差条的线图或条形图。

**时间序列**

42. acorr(): 绘制自相关图。

43. xcorr(): 绘制互相关图。

**其他特定用途的函数**

44. spy(): 可视化稀疏矩阵。

45. specgram(): 绘制频谱图。

46. csd(): 绘制相对功率谱密度图。

47. cohere(): 绘制相干图。

**自定义与绘图辅助**

48. subplot(): 创建子图。

49. subplots(): 创建多个子图并返回Figure和Axes对象。

50. subplot2grid(): 使用网格布局创建子图。

51. add\_axes(): 在指定位置添加自定义坐标轴。

52. add\_subplot(): 在图形中添加子图。

## Seaborn

**分类数据绘图函数**

1. barplot(): 绘制分类数据的条形图。

• 用途：显示分类数据的平均值或其他聚合统计。

• 示例：sns.barplot(x="category", y="value", data=data)

2. countplot(): 绘制分类数据的计数条形图。

• 用途：显示每个分类的频率。

• 示例：sns.countplot(x="category", data=data)

3. boxplot(): 绘制分类数据的箱线图。

• 用途：显示分类数据的分布和异常值。

• 示例：sns.boxplot(x="category", y="value", data=data)

4. violinplot(): 绘制分类数据的小提琴图。

• 用途：显示分类数据的分布，结合了箱线图和核密度图。

• 示例：sns.violinplot(x="category", y="value", data=data)

5. stripplot(): 绘制分类数据的散点图。

• 用途：显示每个分类的所有数据点，通常与 boxplot 或 violinplot 结合使用。

• 示例：sns.stripplot(x="category", y="value", data=data)

6. swarmplot(): 绘制分类数据的分散散点图。

• 用途：类似 stripplot，但数据点不会重叠。

• 示例：sns.swarmplot(x="category", y="value", data=data)

7. pointplot(): 绘制分类数据的点图。

• 用途：显示分类数据的均值和置信区间。

• 示例：sns.pointplot(x="category", y="value", data=data)

8. catplot(): 绘制分类数据的通用图表。

• 用途：结合不同类型的分类图表，如 barplot、boxplot、violinplot 等。

• 示例：sns.catplot(x="category", y="value", data=data, kind="bar")

**关系图与回归分析**

9. scatterplot(): 绘制散点图。

• 用途：显示两个连续变量之间的关系。

• 示例：sns.scatterplot(x="x", y="y", data=data)

10. lineplot(): 绘制折线图。

• 用途：显示两个变量之间的关系，通常用于时间序列数据。

• 示例：sns.lineplot(x="x", y="y", data=data)

11. regplot(): 绘制带回归线的散点图。

• 用途：显示两个变量之间的线性关系。

• 示例：sns.regplot(x="x", y="y", data=data)

12. lmplot(): 绘制带有回归模型的多图。

• 用途：与 regplot 类似，但支持更多的灵活性，如 hue 和 col。

• 示例：sns.lmplot(x="x", y="y", hue="category", data=data)

13. relplot(): 绘制关系图的通用接口。

• 用途：结合 scatterplot 和 lineplot，可以通过参数 kind 指定图表类型。

• 示例：sns.relplot(x="x", y="y", hue="category", data=data, kind="scatter")

**分布图与密度图**

14. histplot(): 绘制直方图。

• 用途：显示单变量的分布。

• 示例：sns.histplot(data=data, x="value")

15. kdeplot(): 绘制核密度估计图。

• 用途：显示单变量或双变量的密度分布。

• 示例：sns.kdeplot(data=data, x="value")

16. distplot(): 绘制直方图与密度图的组合图（即将被废弃，推荐使用 histplot 和 kdeplot）。

• 用途：显示数据的分布及其密度。

• 示例：sns.distplot(data["value"])

17. ecdfplot(): 绘制经验累积分布函数图。

• 用途：显示数据的累积分布。

• 示例：sns.ecdfplot(data=data, x="value")

18. jointplot(): 绘制两个变量的联合分布图。

• 用途：显示两个变量的关系及其边际分布。

• 示例：sns.jointplot(x="x", y="y", data=data)

19. pairplot(): 绘制成对变量关系的矩阵图。

• 用途：显示数据集中所有变量对之间的关系。

• 示例：sns.pairplot(data)

20. heatmap(): 绘制热力图。

• 用途：显示矩阵数据的热力图，通常用于相关性矩阵。

• 示例：sns.heatmap(data.corr())

21. clustermap(): 绘制带有聚类的热力图。

• 用途：显示带有层次聚类的矩阵数据。

• 示例：sns.clustermap(data)

**时间序列分析**

22. tsplot(): 绘制时间序列数据的图表（已被废弃）。

• 用途：分析时间序列数据（推荐使用 lineplot）。

• 示例：sns.lineplot(data=data, x="time", y="value")

**其他高级功能**

23. pairgrid(): 创建成对变量的网格图。

• 用途：自定义变量对之间的关系图表。

• 示例：g = sns.PairGrid(data)

24. facetgrid(): 创建分面网格图。

• 用途：在多个子图上绘制图表，支持分类变量的不同分组。

• 示例：g = sns.FacetGrid(data, col="category")

25. jointgrid(): 创建联合分布的网格图。

• 用途：更灵活地自定义 jointplot 图表。

• 示例：g = sns.JointGrid(x="x", y="y", data=data)

26. residplot(): 绘制回归残差图。

• 用途：分析模型的残差分布。

• 示例：sns.residplot(x="x", y="y", data=data)

**调色板与样式设置**

27. color\_palette(): 设置调色板。

• 用途：自定义图表的颜色主题。

• 示例：sns.color\_palette("husl")

28. set\_style(): 设置图表的背景样式。

• 用途：选择图表的样式主题，如白色、暗色等。

• 示例：sns.set\_style("whitegrid")

29. set\_context(): 设置图表的上下文大小。

• 用途：根据不同的使用场景调整图表元素的大小。

• 示例：sns.set\_context("talk")

30. set(): 全局设置样式和调色板。

• 用途：简便地设置 seaborn 的所有样式和调色板参数。

• 示例：sns.set(style="whitegrid", palette="pastel")

# 5.配色网站：

**色卡搭配**

**<http://www.peiseka.cn>**

**配图官网**

**https://matplotlib.org/stable/users/explain/toolkits/mplot3d.html**

**<https://gallery.pyecharts.org/#/BMap/air_quality_baidu_map?id=pyecharts-代码-效果>**

**https://echarts.apache.org/examples/en/index.html**