清华大学数据库技术与应用

数据可视化 I

授课教师：计算机系王健楠

授课学期：2026年（春季）

课程议程

可视化目标

数据可视化的目标

目标一：帮助自己理解数据和结果

探索性分析的关键部分

在深入分析前**快速把握数据**趋势

灵活轻便，**可反复迭代**

第一部分

目标二：向他人展示结果和结论

需要**精心**组织和筛选

深思熟虑地呈现

围绕沟通目标**精准打磨**

第二部分

分布可视化

什么是分布 (Distribution)

形式化定义

> **分布 (Distribution) 描述的是**单个变量在其所有可能取值上的分配情况，包括：
>
> (1) 变量可能取的**值集合** (2) **每个**值出现的**频率或概率**

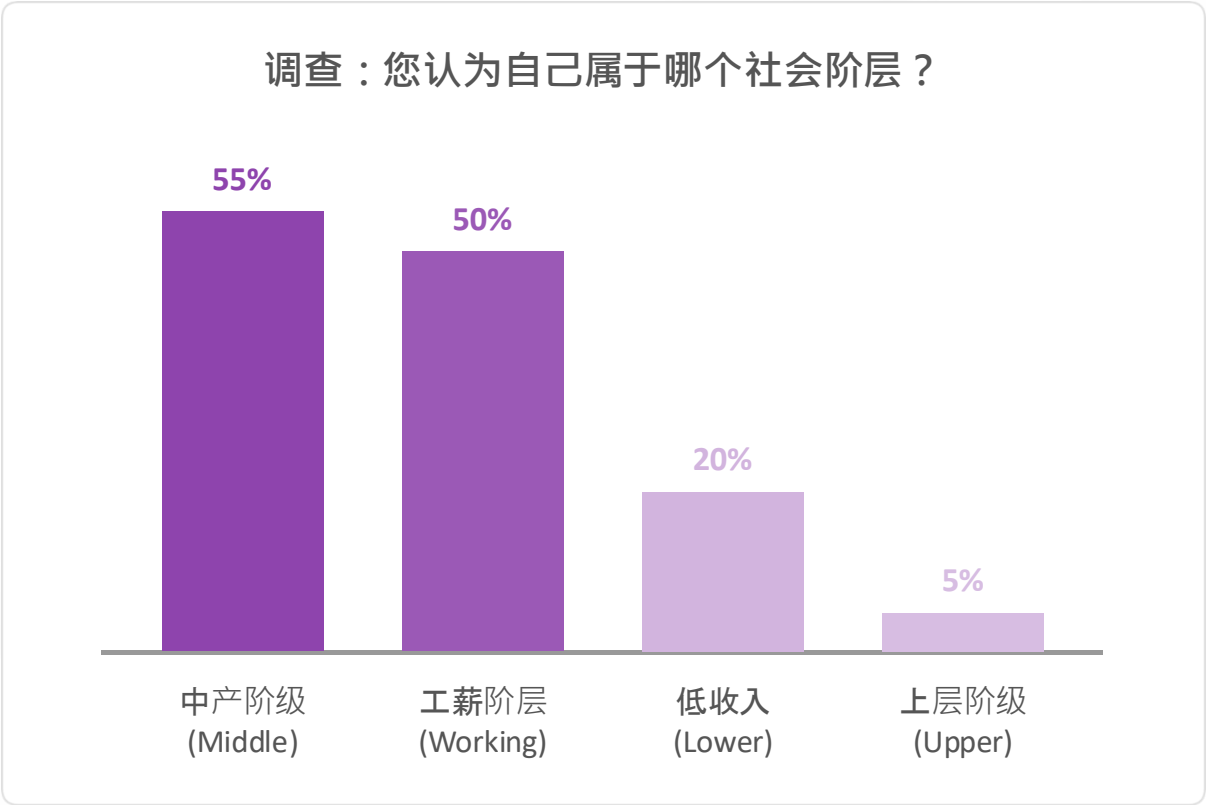
示例说明

以清华大学教师分布为例，我们统计各院系的教师人数：
院系列表（计算机系、数学系、物理系等）和各院系对应的教师人数，构成了**“所属院系”**这个变量的分布。

注意事项

- **所有百分比之和必**须为 **100%**（**若使用比例**）
- **所有**计数之和必须为 **数据总数**（**若使用**频数）

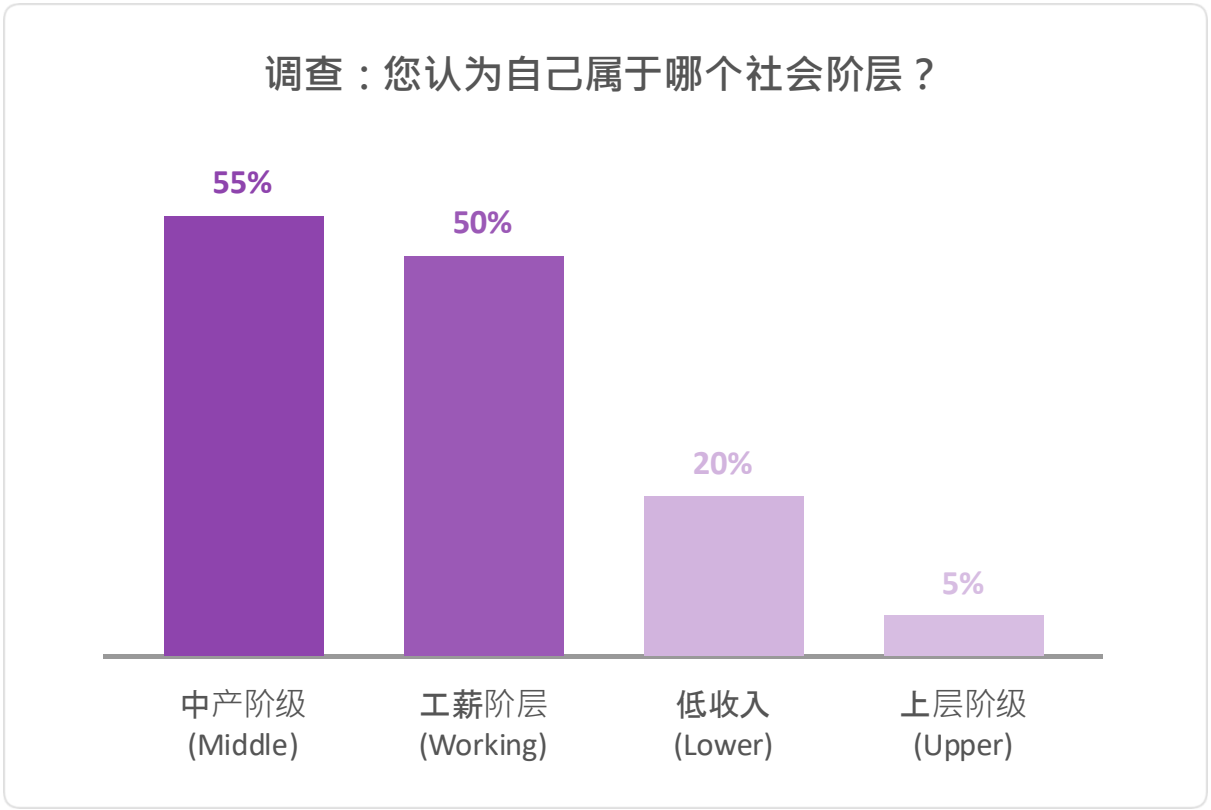
这个图表展示的是分布吗？



** 注：受访者可以选择多个认同的类别。*

课堂练习 - 答案

问题：这个图表展示的是分布吗？



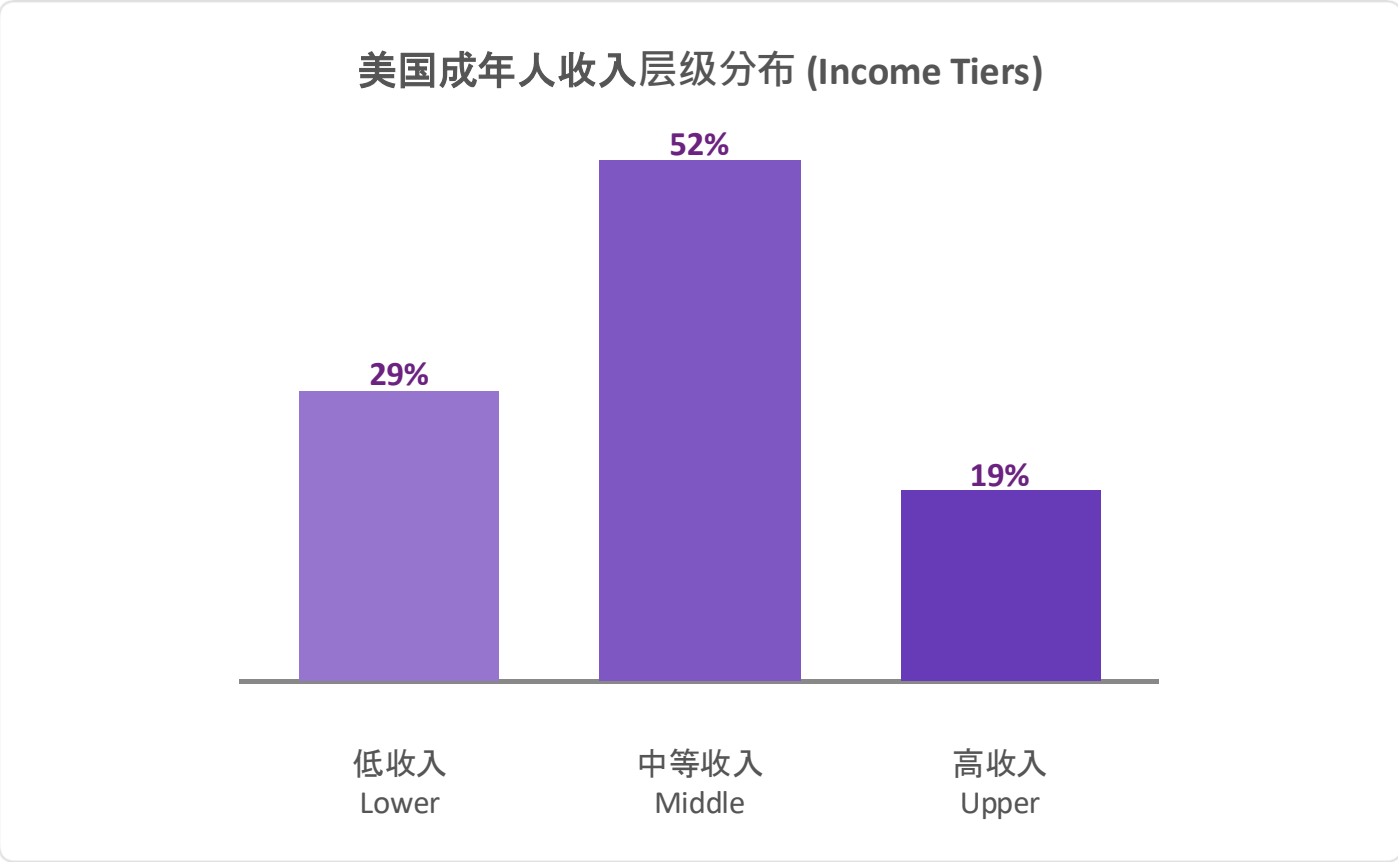
否 (No)

原因解释：

虽然展示了百分比，但个体可能属于**多个类别**。

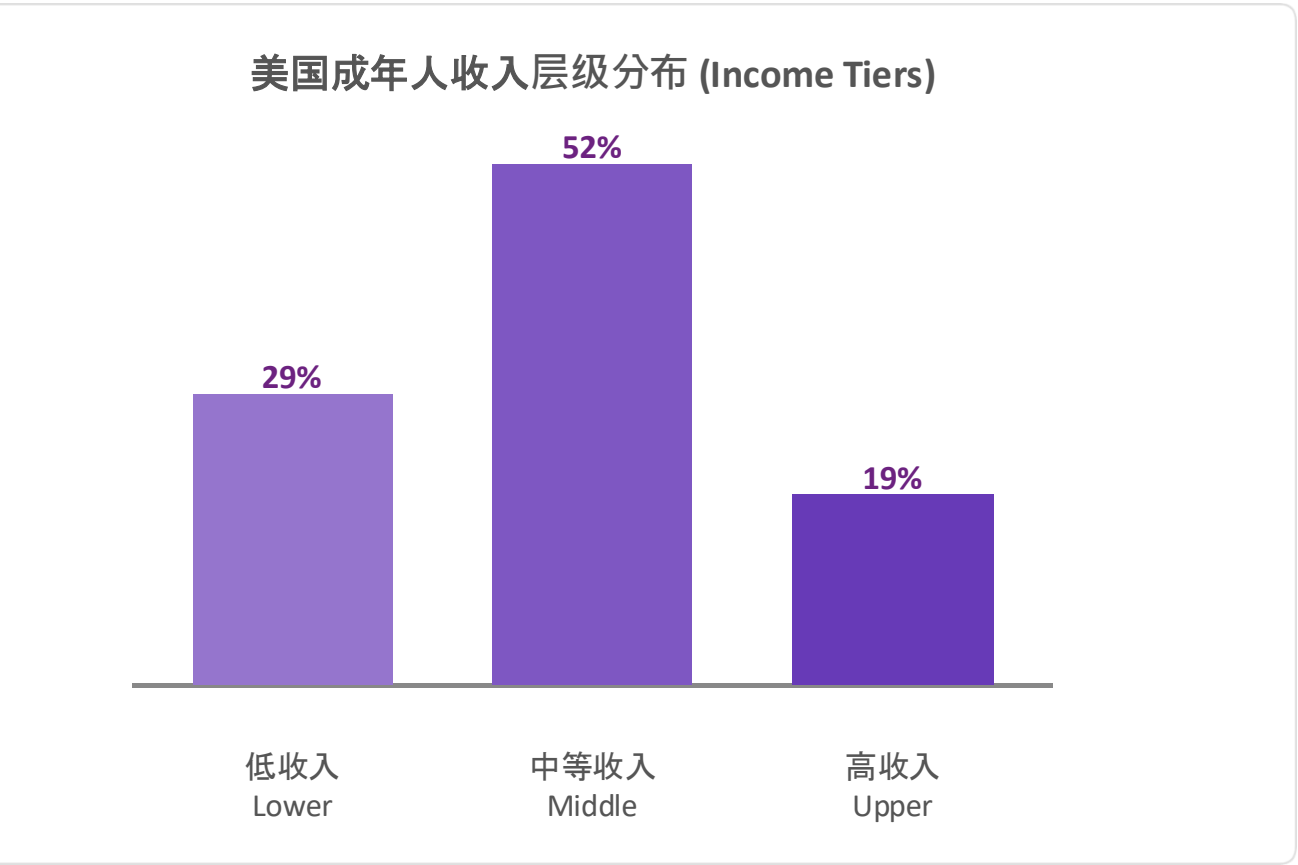
所有类别的总和可能超过100%。

这个图表展示的是分布吗？



课堂练习 - 答案

问题：这个图表展示的是分布吗？



是 (Yes)

原因解释：

该图展示了“**收入**层级”这个**定性有序变量**的分布。

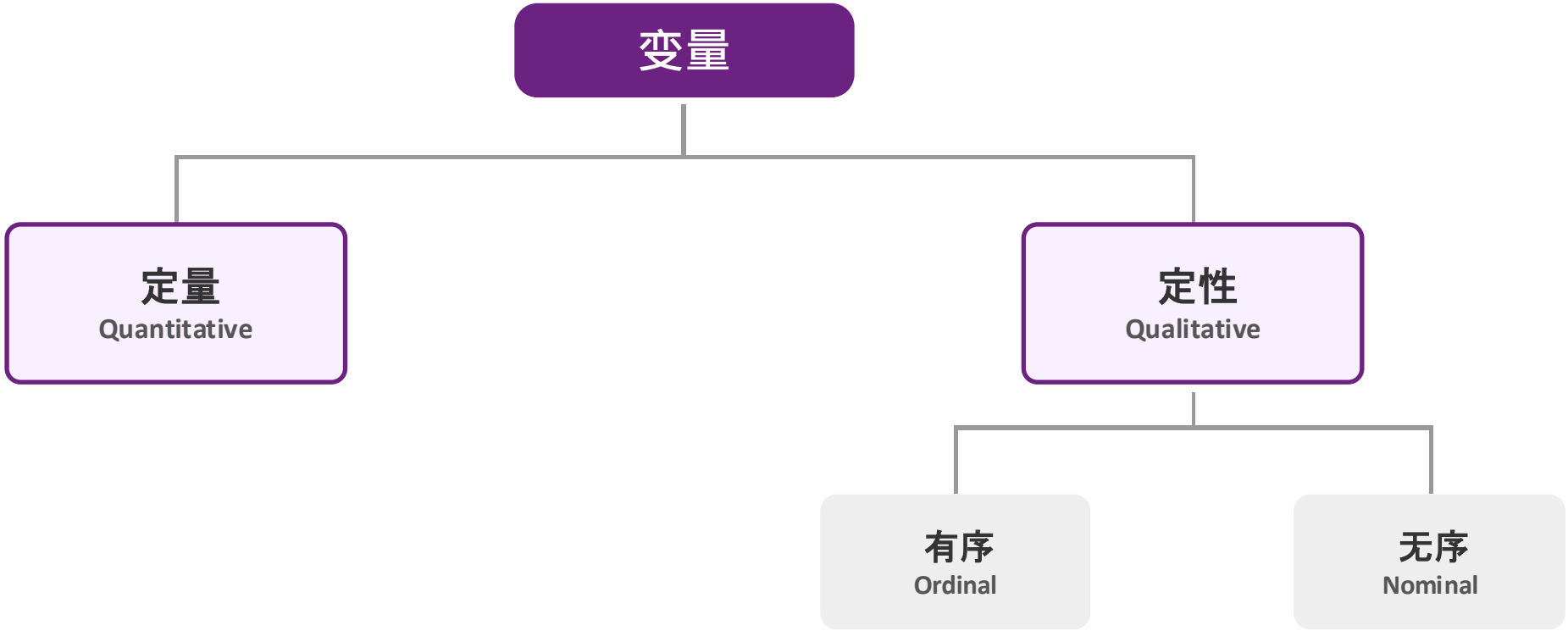
每个个体只属于一个收入层级类别（互斥）。

显示的值是各类别中个体的比例。

所有个体都被涵盖，总百分比为 **100%** (29+52+19=100)。

变量类型与图表选择

不同的图表适合展示不同类型的变量。可视化工作的第一步始终是：
识别要可视化的变量类型，然后据此选择合适的图表。



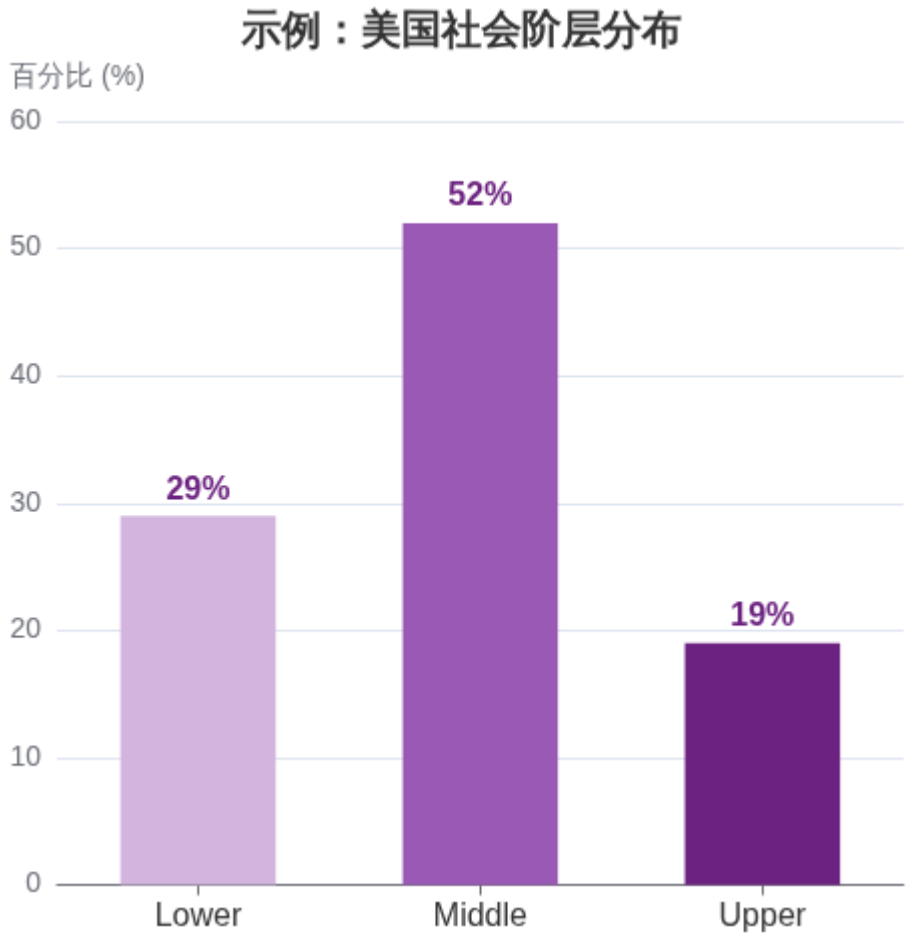
条形图：定性变量的分布

条形图 (Bar Plot) 是展示定性变量分布最常用的方法。

适用范围：
适用于定性变量（有序和无序）。
* **也可用于取**值较少的离散定量变量。

视觉编码：
✓ **长度** 编码数值（频率或计数）。
✗ **宽度 无任何含义！**

颜色使用：
颜色可用于区分子类别，但并非必需。



示例数据集：世界银行数据

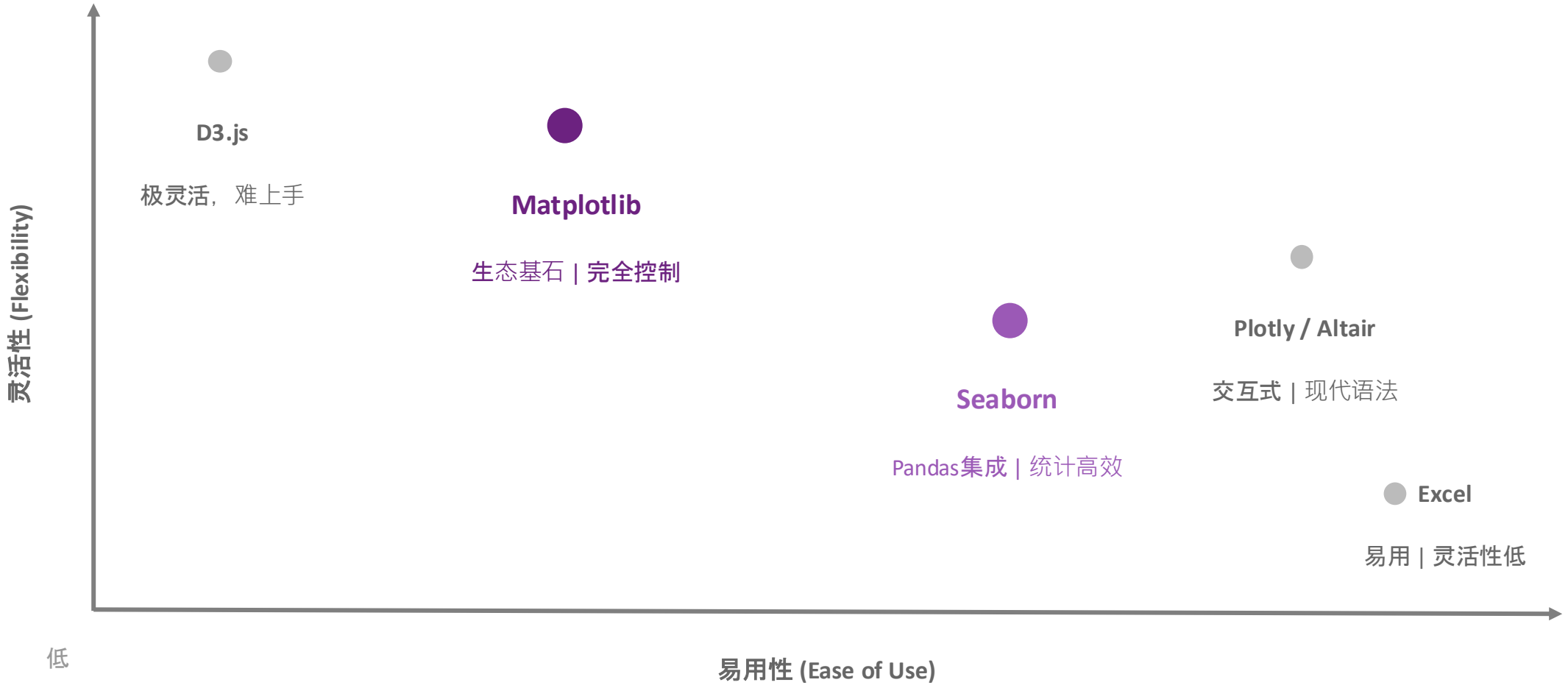
本节课我们将使用世界银行 (World Bank) 关于世界各国的数据集 wb 进行大部分演示。 该数据集包含了国家、大洲以及各项经济指标。

数据集预览 (前5行):

Country	Continent	Year	GDP growth %	GNI per capita	Population
United States	North America	2021	5.9%	$70,430	331,893,745
China	Asia	2021	8.1%	$11,880	1,412,360,000
India	Asia	2021	8.7%	$2,170	1,407,563,842
Germany	Europe	2021	2.6%	$51,660	83,129,285
Brazil	South America	2021	4.6%	$7,720	214,326,223

Python数据可视化工具生态

选择工具时的核心权衡 (Trade-off)：**灵活性 (Flexibility) 与 易用性 (Ease of Use)**



生成条形图：Matplotlib

```
import matplotlib.pyplot as plt # plt 是 matplotlib.pyplot 的常用别名
```

基本绘图结构

```
plt.plotting_function(x_values, y_values)
```

传入两个序列（列表、数组或Series），分别对应 x 轴和 y 轴的数据。

添加标签和标题

```
plt.xlabel("x轴标签")
plt.ylabel("y轴标签")
plt.title("图表标题")
```

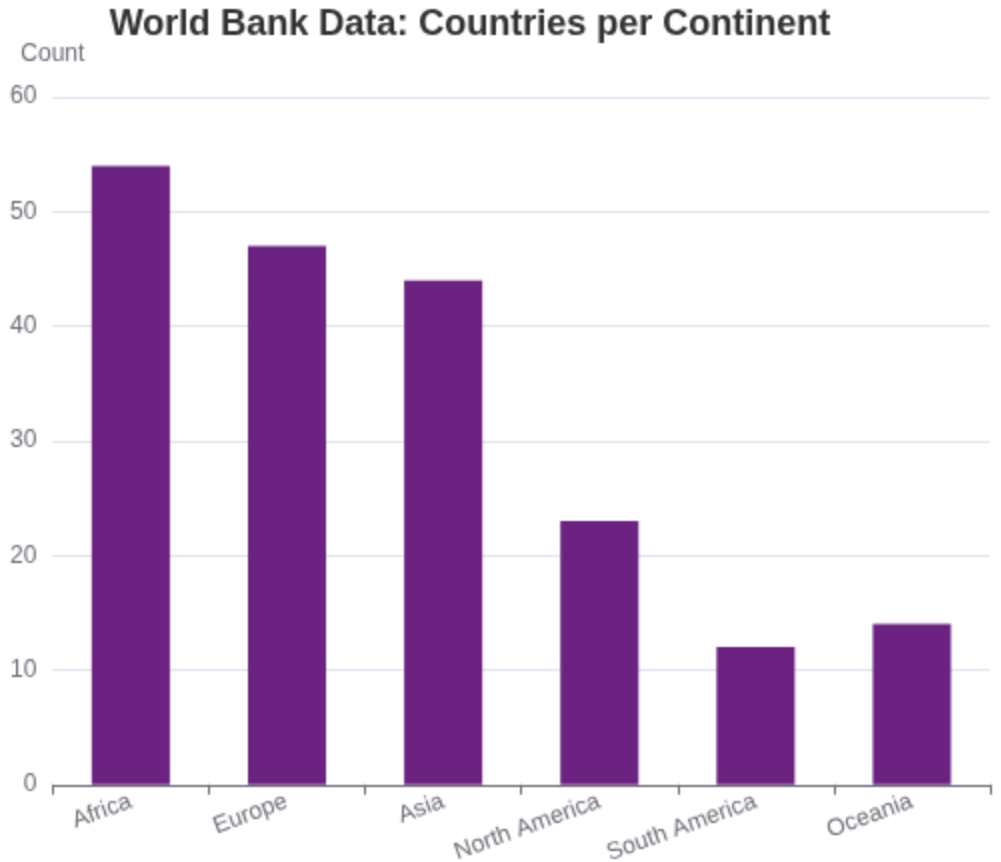
Matplotlib 条形图示例

使用 `plt.bar()` 函数创建条形图：

```
# 计算各大洲的国家数量
continents = wb["Continent"].value_counts()
# 创建条形图
plt.bar( continents.index,continents.values )
```

第一个参数传递 **x轴数据**（各大洲名称）

第二个参数传递 **y轴数据**（对应的计数值 ）



生成条形图：pandas 原生绘图

Pandas 提供了便捷的原生绘图方法，可以直接在 Series 或 DataFrame 对象上调用：

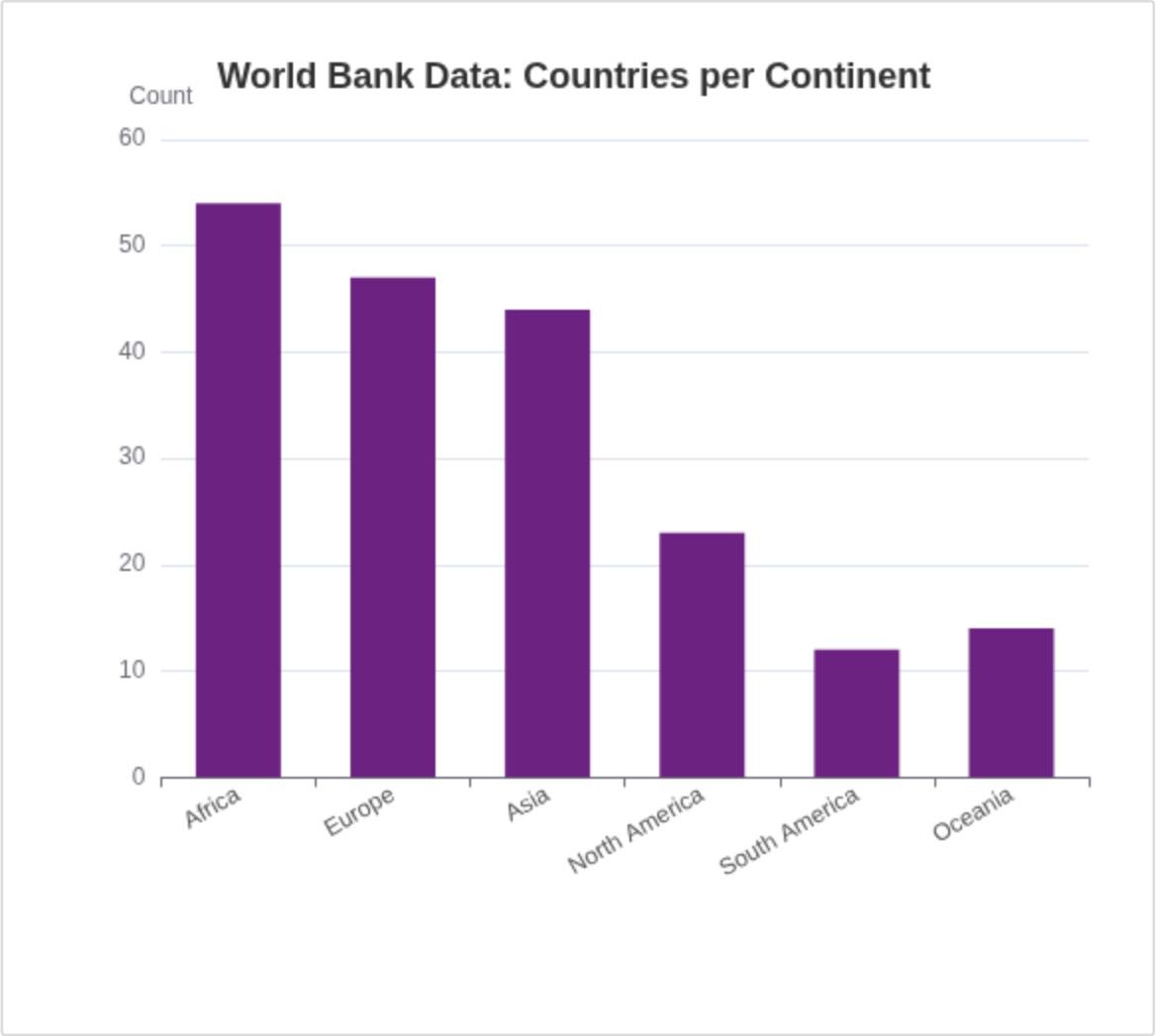
```
wb["Continent"].value_counts().plot(kind='bar')
```

`.plot()` 是 Pandas 对象的一个方法，底层依赖 Matplotlib。

参数 `kind='bar'` 指定生成条形图。

它会自动使用 Series 的索引作为 x 轴，值作为 y 轴。

这种方式语法极其简洁，非常适合快速探索数据。



生成条形图：seaborn

1. 导入库

通常使用别名 sns

```
import seaborn as sns
```

2. 函数结构

传入整个 **DataFrame**，指定列名字符串

```
sns.plotting_func(data=df, x="x_col", y="y_col")
```

3. 添加标签和标题

使用与 Matplotlib 相同的语法：

```
plt.xlabel("x axis label")

plt.ylabel("y axis label")

plt.title("Title of the plot")
```

关键说明：Seaborn 与 Matplotlib 的关系

Seaborn **构建在 Matplotlib 之上**！

底层使用 Matplotlib 进行绘制。

提供了更便捷的 DataFrame 接口和高级统计图表功能。

seaborn 条形图示例

`sns.countplot()` **在更高的抽象**层次上操作，自动**完成**计数和绘图：

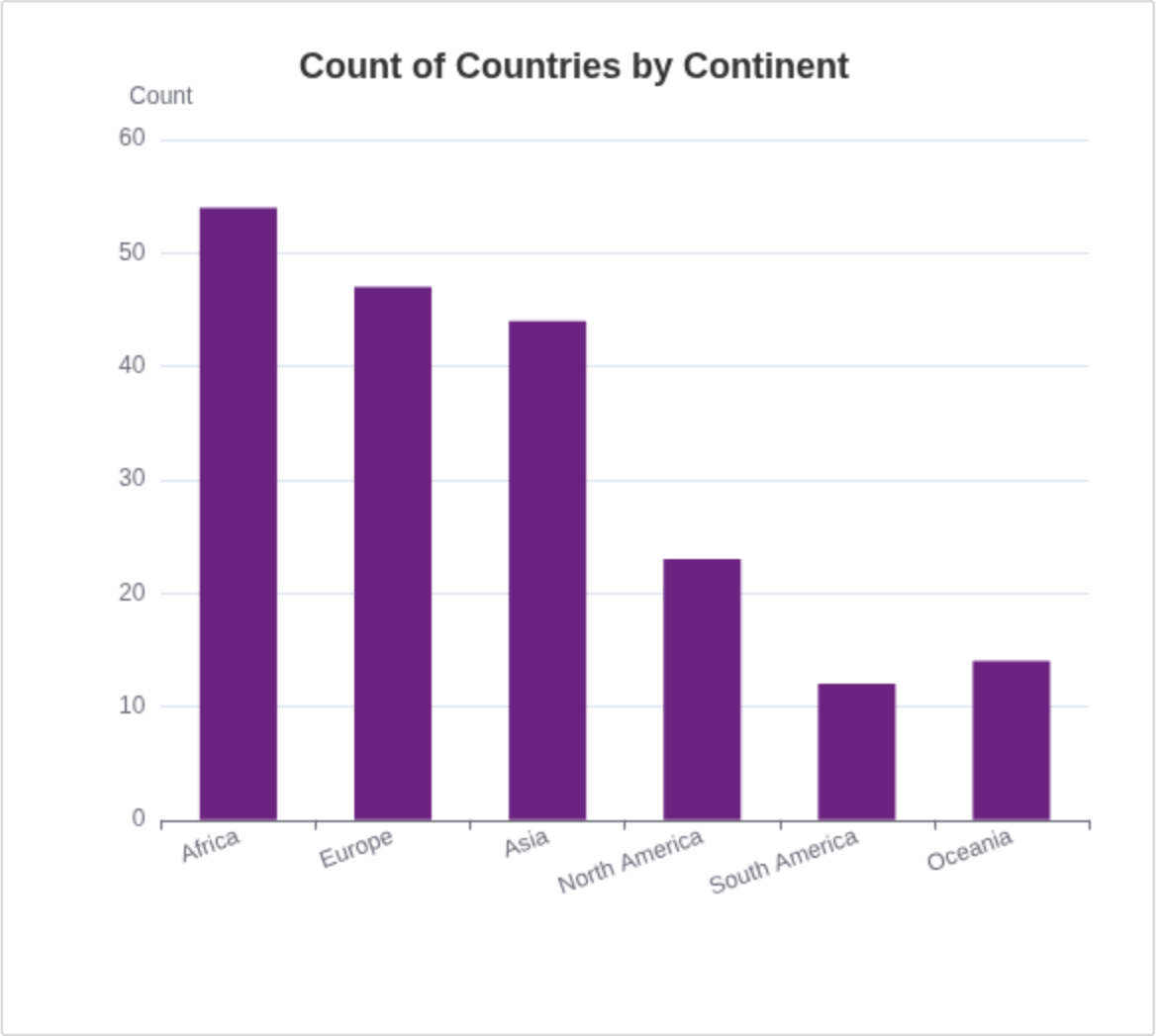
```
import seaborn as sns


sns.countplot(data=wb, x="Continent")
```

只需传入整个 **DataFrame** (`data=wb`)

指定要统计的 **列名** (`x="Continent"`)

优势：不需要先手动调用 `.value_counts()`，**函数内部会自**动处理**聚合**。



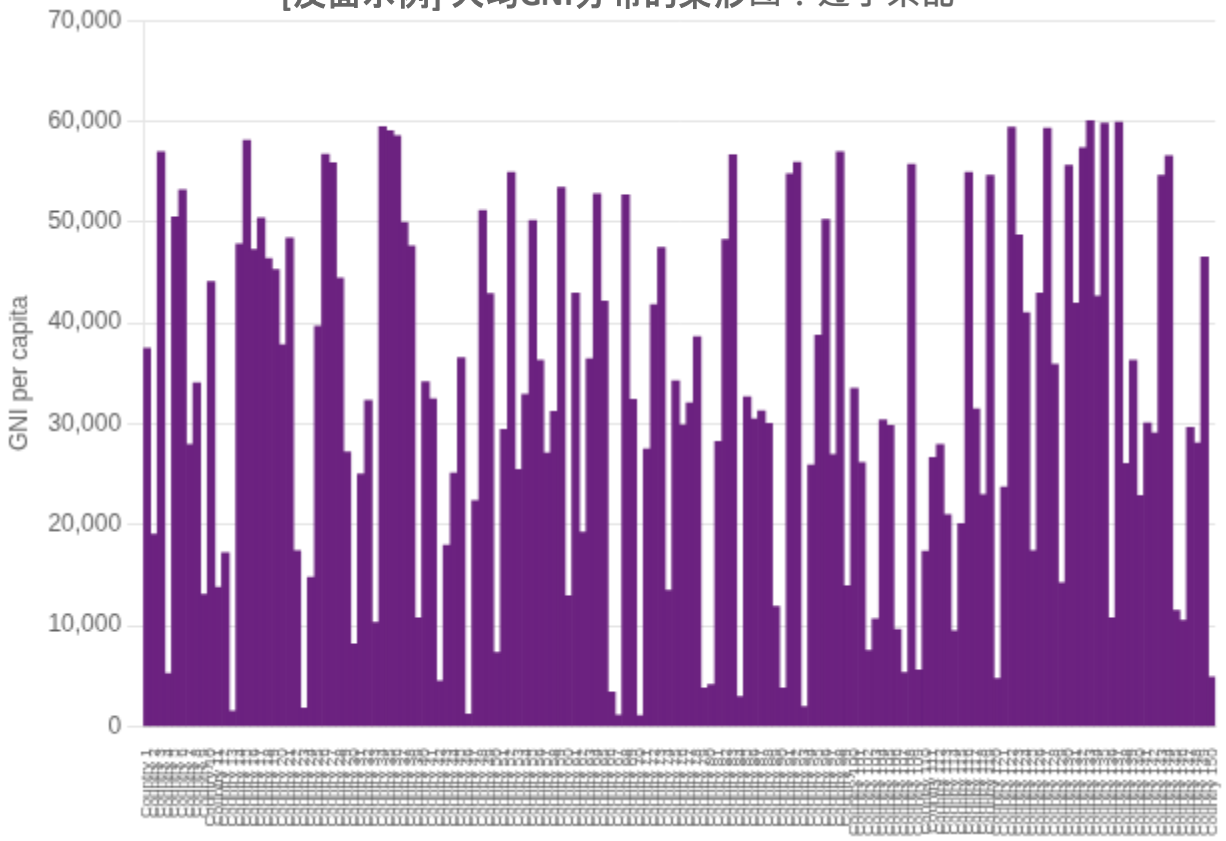
定量变量的分布

为什么条形图不适合定量变量？

条形图会为每个**唯一数值**创建一个单独的条形。
连续数据通常有成百上千个唯一值。

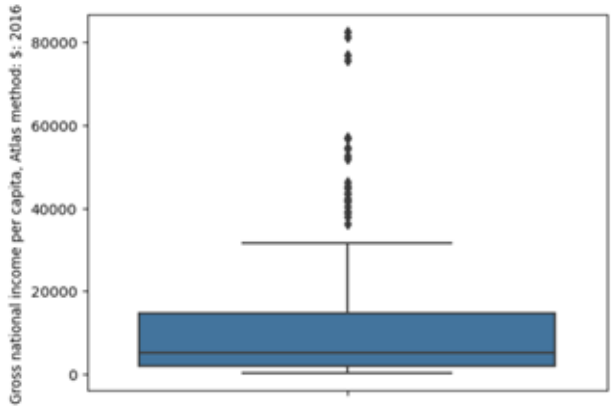
生成数百个拥挤的细条。
X轴标签重叠，无法阅读。
无法看出数据的分布形态（如偏态、峰值）。

[反面示例] 人均GNI分布的条形图：过于杂乱

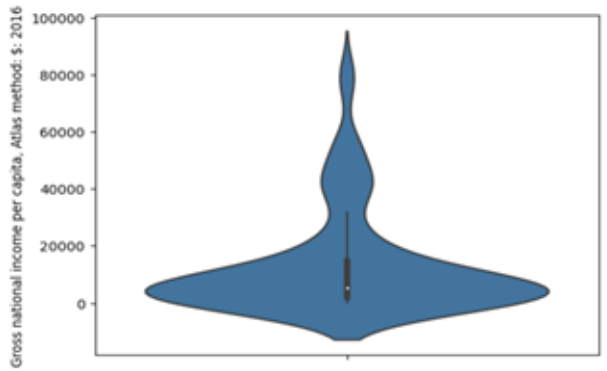


定量变量的分布

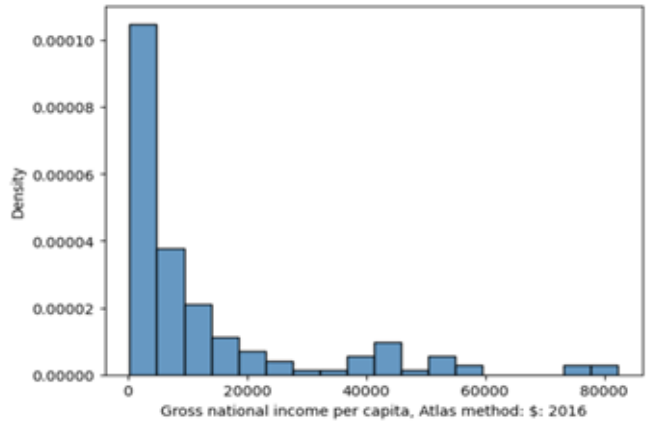
前面提到条形图适合定性变量。对于**连续定量变量**，为了避免为每个唯一值生成条形，我们通常使用以下三种**主要方法**：



1. 箱线图 (Box Plot)



2. 小提琴图 (Violin Plot)



3. 直方图 (Histogram)

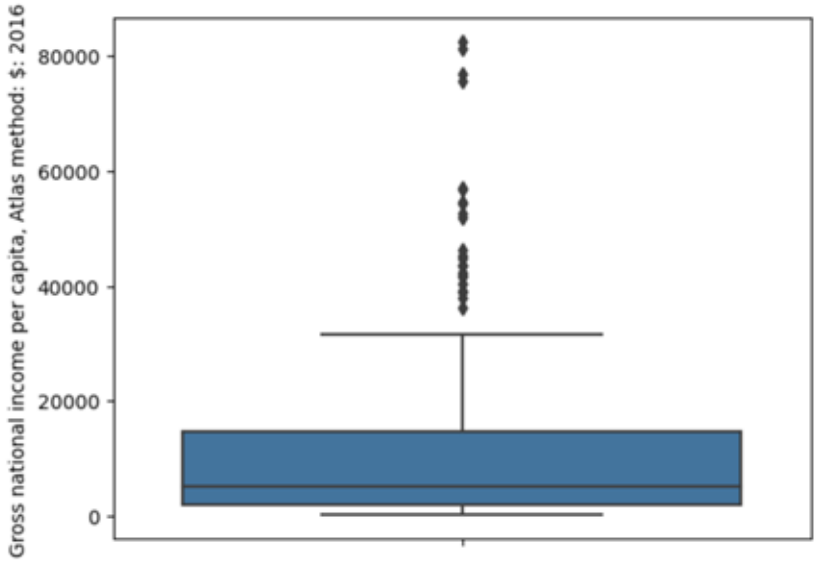
定量变量的分布可视化

为了可视化连续定量变量的分布，通常使用**箱线图和小提琴图（都利用四分位数信息来概括数据的分布情况）**

1. 箱线图 (Box Plot)

注意：**箱子的宽度没有任何含义。**

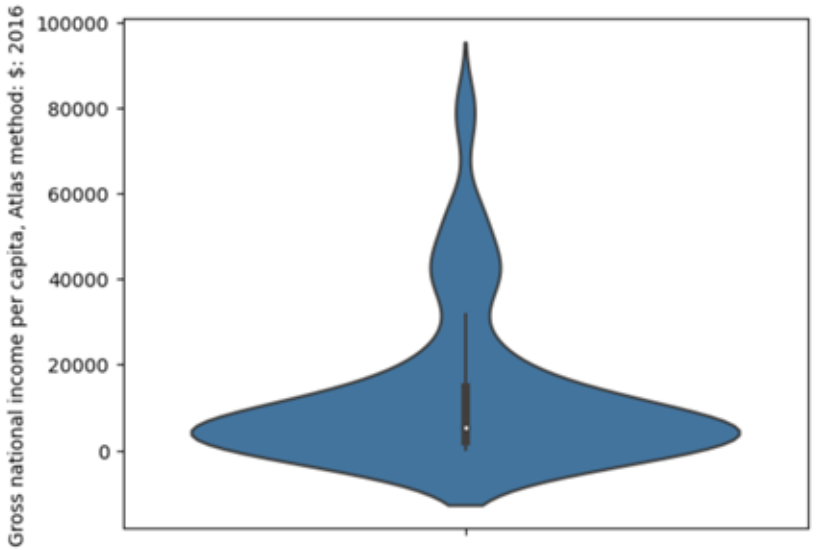
```
sns.boxplot(data=df, y="y_variable")
```



2. 小提琴图 (Violin Plot)

注意：**"小提琴"的宽度表示数据密度。**

```
sns.violinplot(data=df, y="y_variable")
```



四分位数 (Quartiles)

对于定量变量：

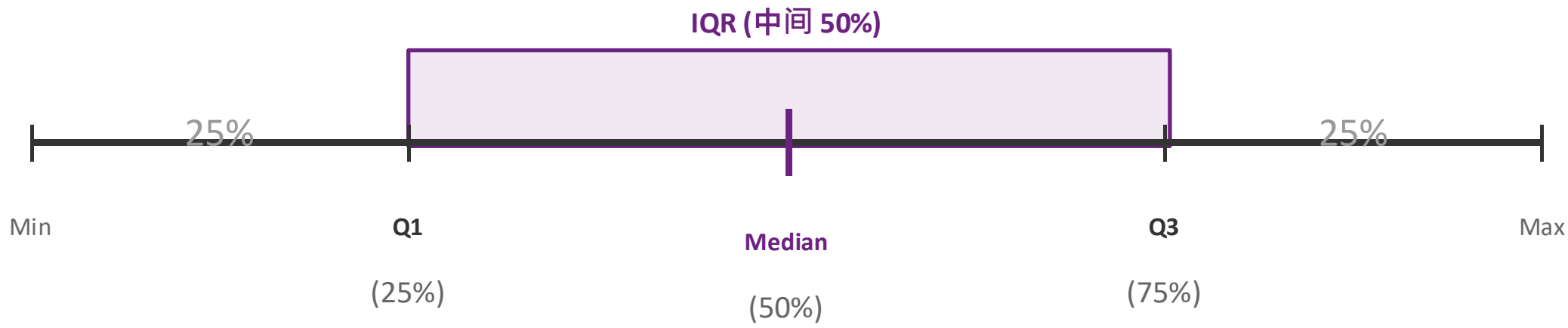
第一四分位数 (Lower Quartile, Q1)：第 25 百分位数

第二四分位数 (Median)：第 50 百分位数，即中位数。

第三四分位数 (Upper Quartile, Q3)：第 75 百分位数

四分位距 (Interquartile Range, IQR)：衡量数据的分散程度。 **IQR = Q3 - Q1**

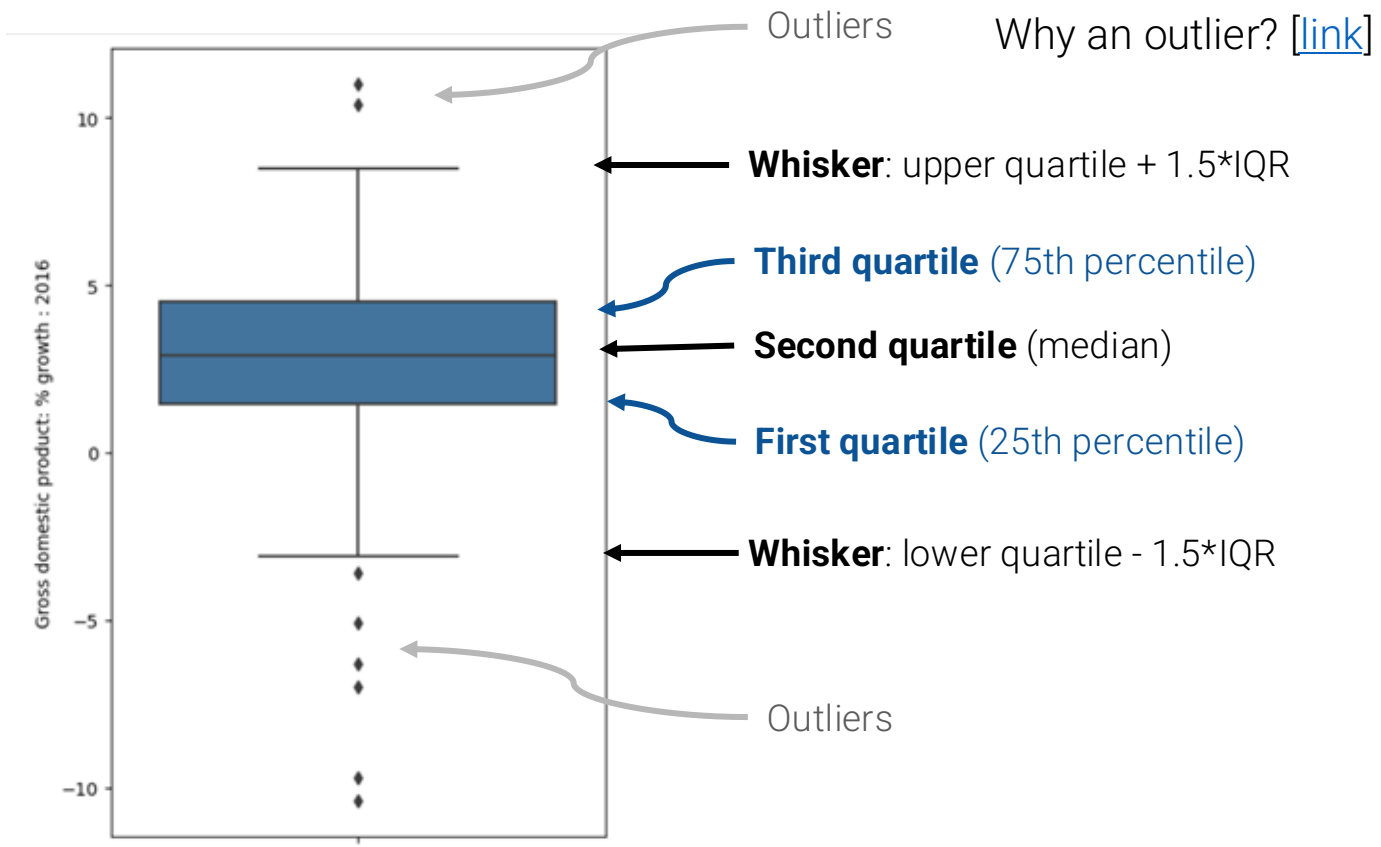
区间 **[Q1, Q3]** 包含了数据的 **“中间 50%”**。



箱线图 (Box Plot)

Seaborn 代码：

```
sns.boxplot(data=wb, y="Gross domestic product: % growth : 2016")
```



小提琴图 (Violin Plot)

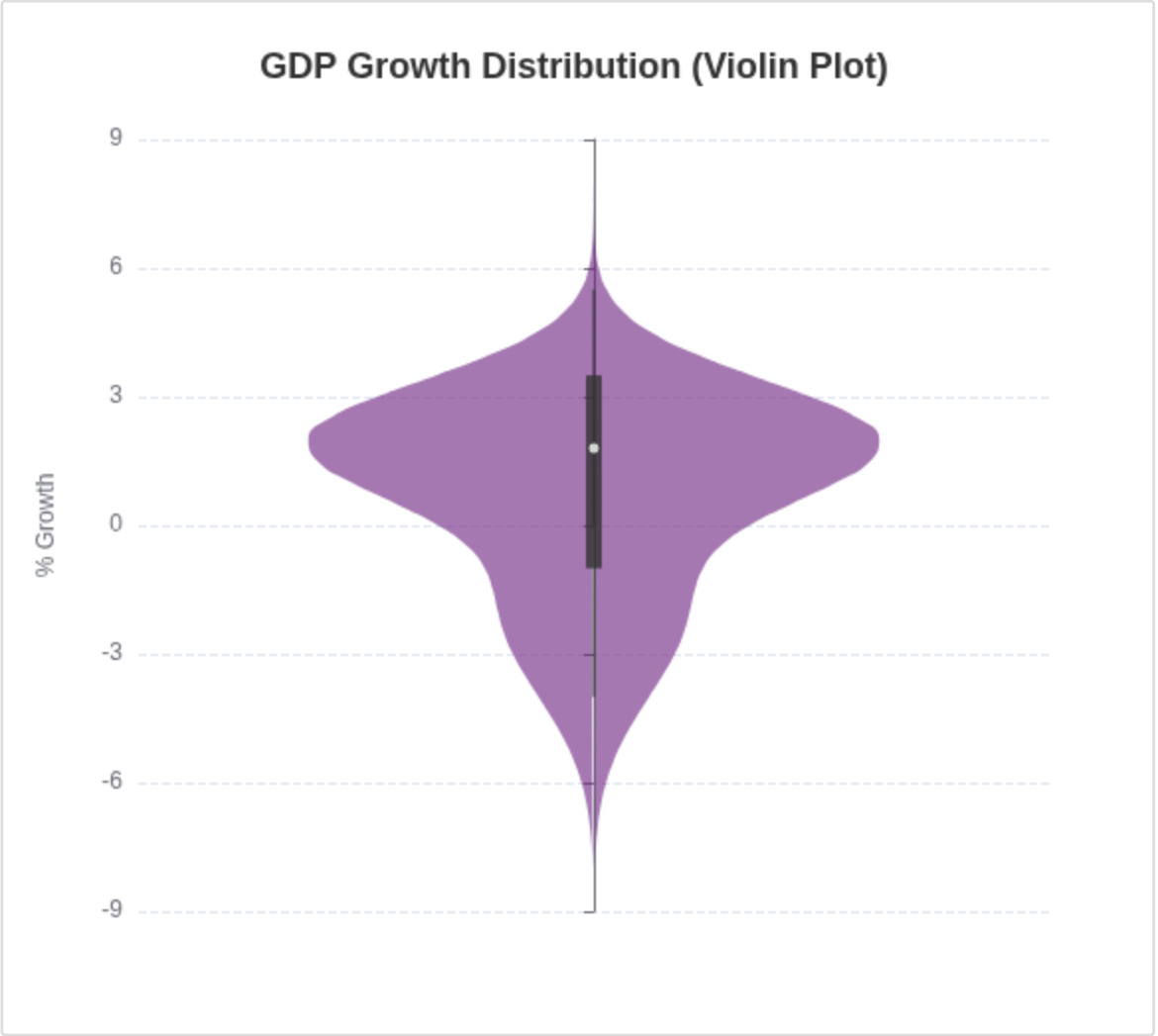
小提琴图类似箱线图，但增加了一层信息：
平滑的密度曲线。

```
import seaborn as sns


sns.violinplot(
      data=wb,
      y="Gross domestic product..."
 )
```

宽度的含义：
“小提琴”的宽度现在表示该处数据点的**密度**（频率）。越宽的地方，**数据点越密集**。

内部结构：
仔细观察，内部通常包含一个微型的箱线图结构，展示了中位数、四分位数和须。



按类别对比分布

如何同时展示两个变量？

如果想对比**定量连续变量**在不同**定性类别**中的分布，可以使用**并列**图表。

并列箱线图或**并列小提琴图**是最佳选择。

示例：对比不同大洲（定性）的 GDP 增长率（定量）分布。

```
# x轴放定性变量，y轴放定量变量


sns.boxplot(
    data=wb,
    x="Continent",
    y="GDP Growth %"
 )
```

直方图 (Histogram)

工作原理

将数值相近的数据点归入同一个**“箱” (Bin)**。

调整每个箱的高度，使得**箱的面积等于其包含数据点的百分比（即密度）**。

示例解读

假设第一个箱的数据如下：

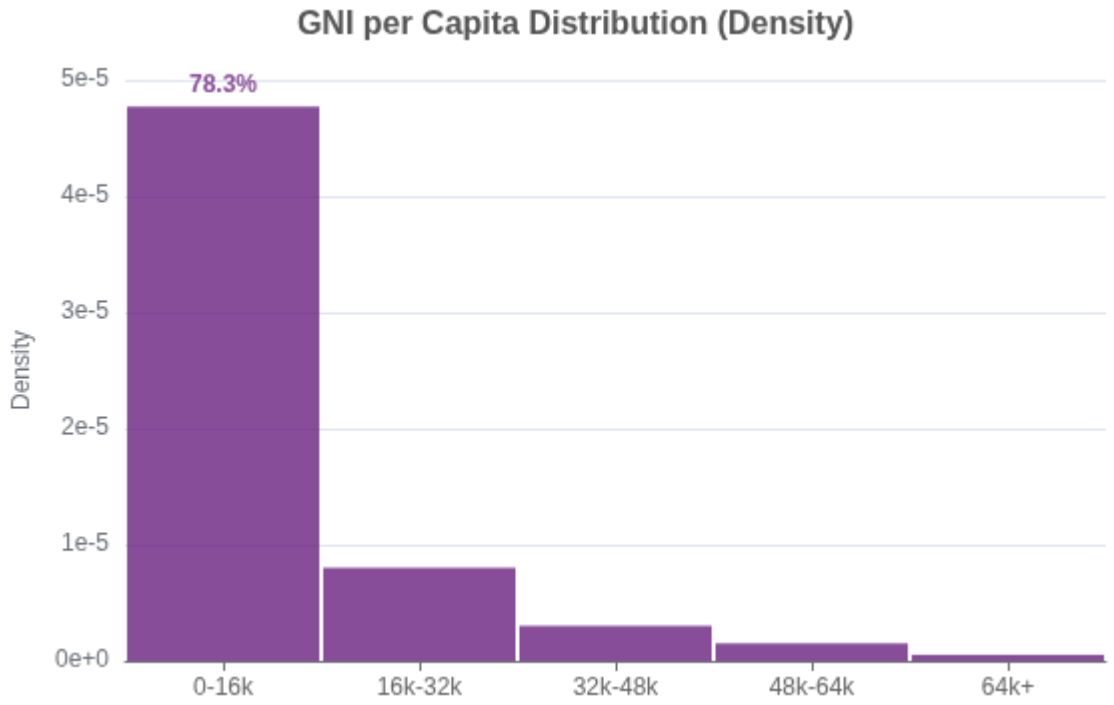
宽度：$16,410

高度：4.77×10^{-5}

计算包含的数据比例：

$$16{,}410 \times (4.77 \times 10^{-5}) \approx 0.783$$

➔ 该箱包含 78.3% 的数据点



代码实现

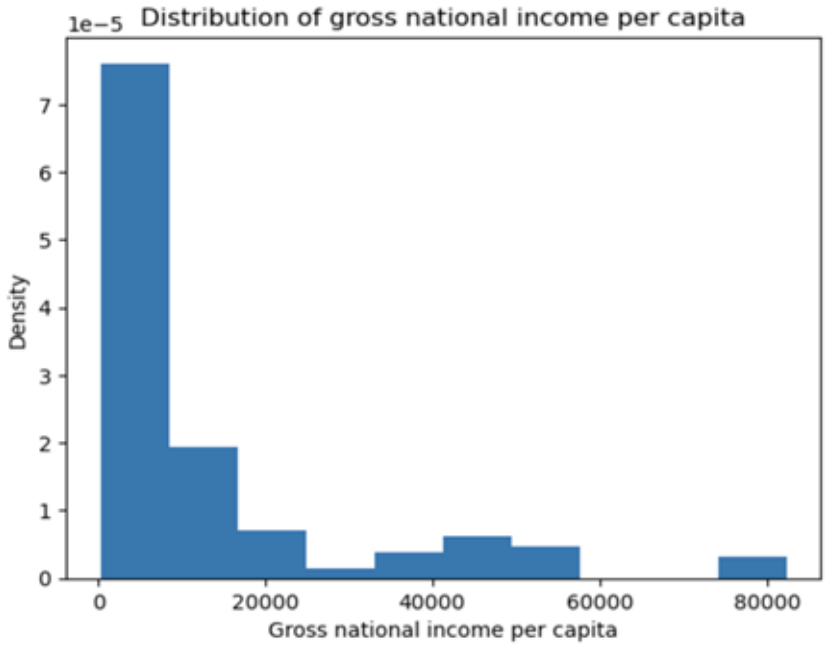
```
plt.hist(values, density=True)
# 或者使用 Seaborn
sns.histplot(data=df, x="col", stat="density")
```

直方图的代码实现

Matplotlib 实现

```
plt.hist(x_values, density=True)
```

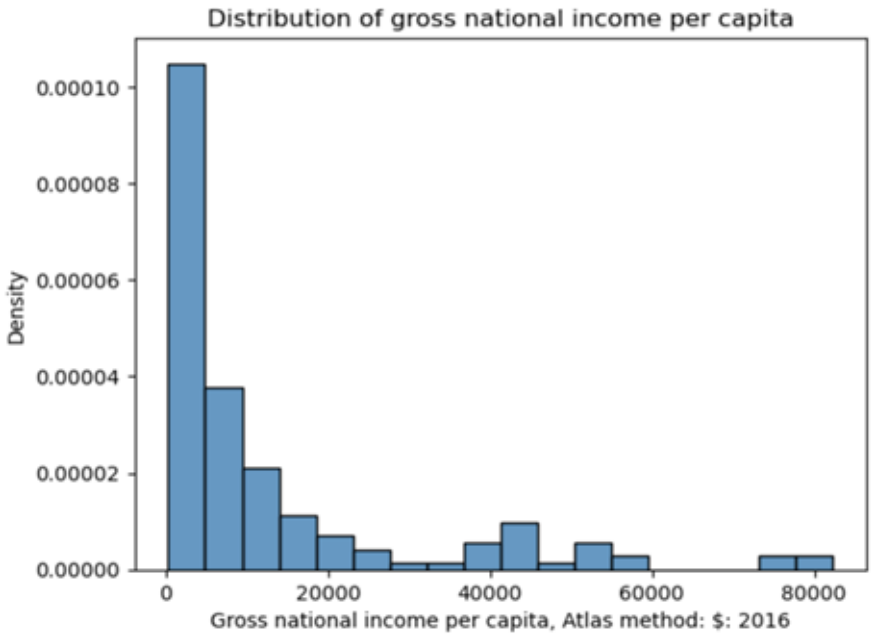
density=True 使得直方图的总面积归一化为1（即表示概率密度）。



Seaborn 实现

```
sns.histplot(data=df, x="col", stat="density")
```

stat="density" 参数指定统计量为密度。Seaborn 提供更高级级的接口。



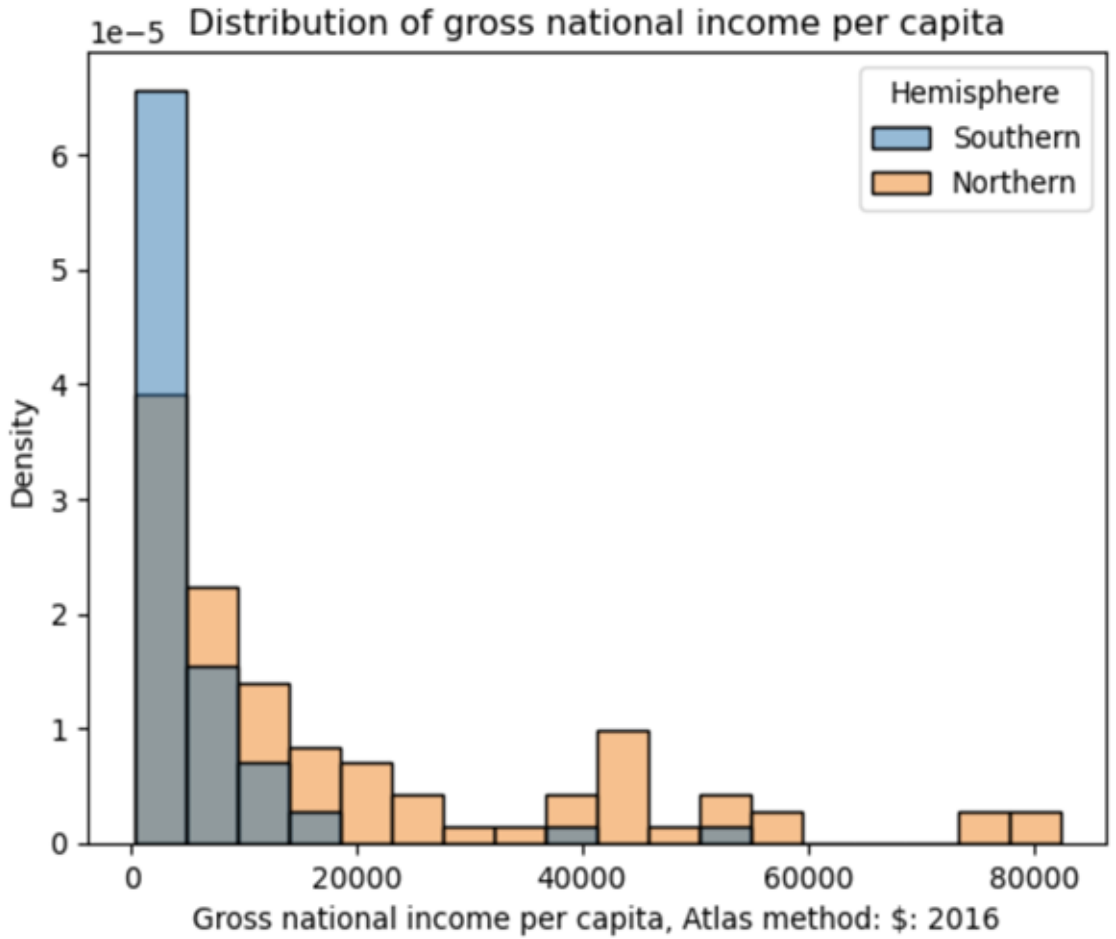
叠加直方图：对比不同类别

要对比定量变量在不同定性类别中的分布，可以将**直方图叠加 (Overlay) 在一起**。
这比并排展示更能直观比较分布的重叠和差异。

```
sns.histplot(

   data=wb,
   hue="Hemisphere",
   x="Gross national income..."

)
```

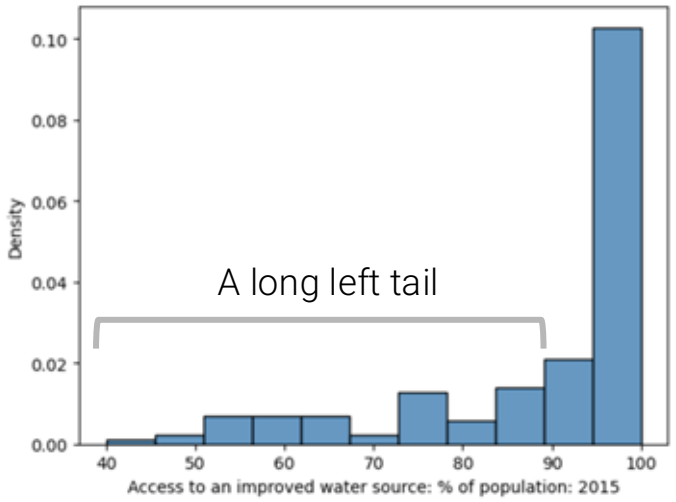


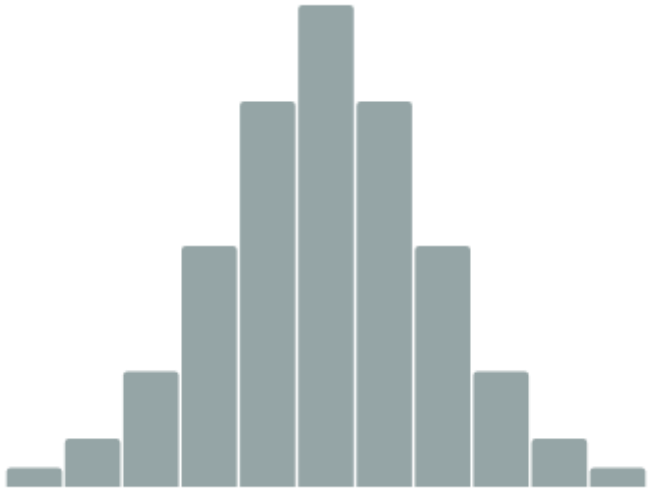
hue 参数：
指定DataFrame中的某一列，Seaborn会自动为该列中的不同类别分配不同的颜色。

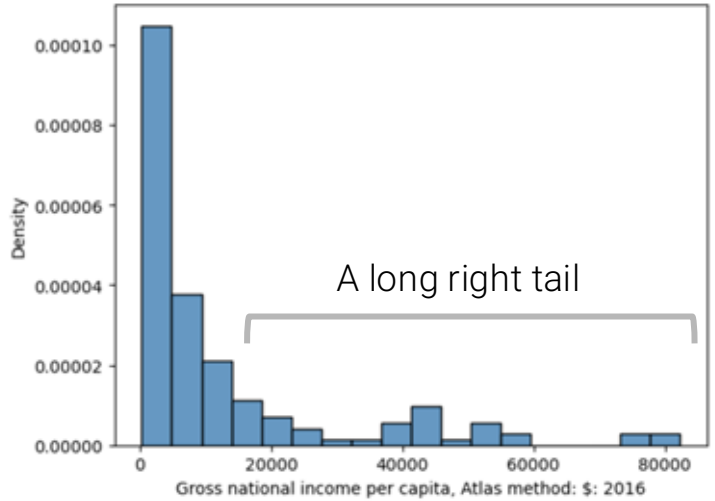
重要提示：
使用颜色编码信息时，务必添加**图例 (legend)**，否则读者无法知道颜色代表的含义！

解读直方图：偏态 (Skewness)

直方图的 **偏态 (Skewness)** 描述了分布的“尾巴”延伸的方向。







左偏 (Left-skewed)

长尾向左侧延伸

对称 (Symmetric)

左右大致镜像

右偏 (Right-skewed)

长尾向右侧延伸

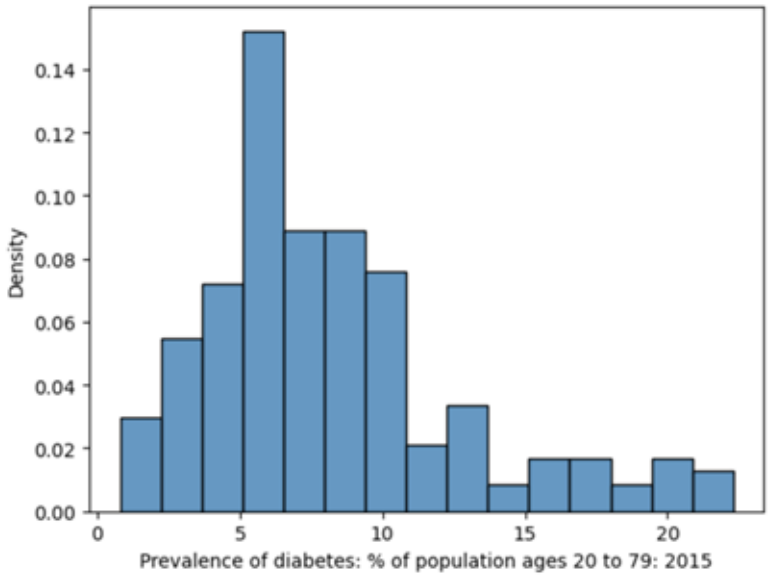
解读直方图：众数 (Modes)

直方图的 **众数 (Mode) 指的是分布中的峰值点。**

单峰 (Unimodal)：只有一个明显的峰值。

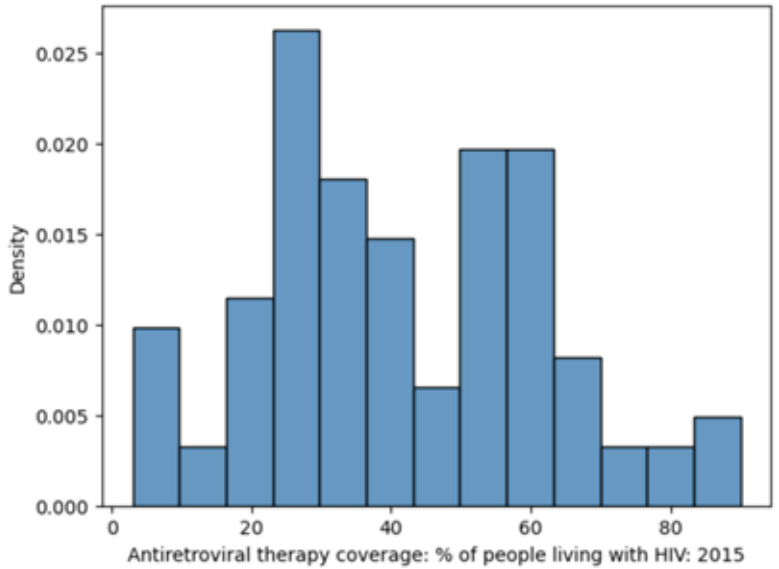
双峰 (Bimodal)：有两个明显的峰值。

多峰 (Multimodal)：有多个峰值。



单峰 (Unimodal)

只有一个明显的峰值



双峰 (Bimodal)

有两个明显的峰值（可能暗示来自两个不同群体）

核密度估计

为什么需要核密度估计？

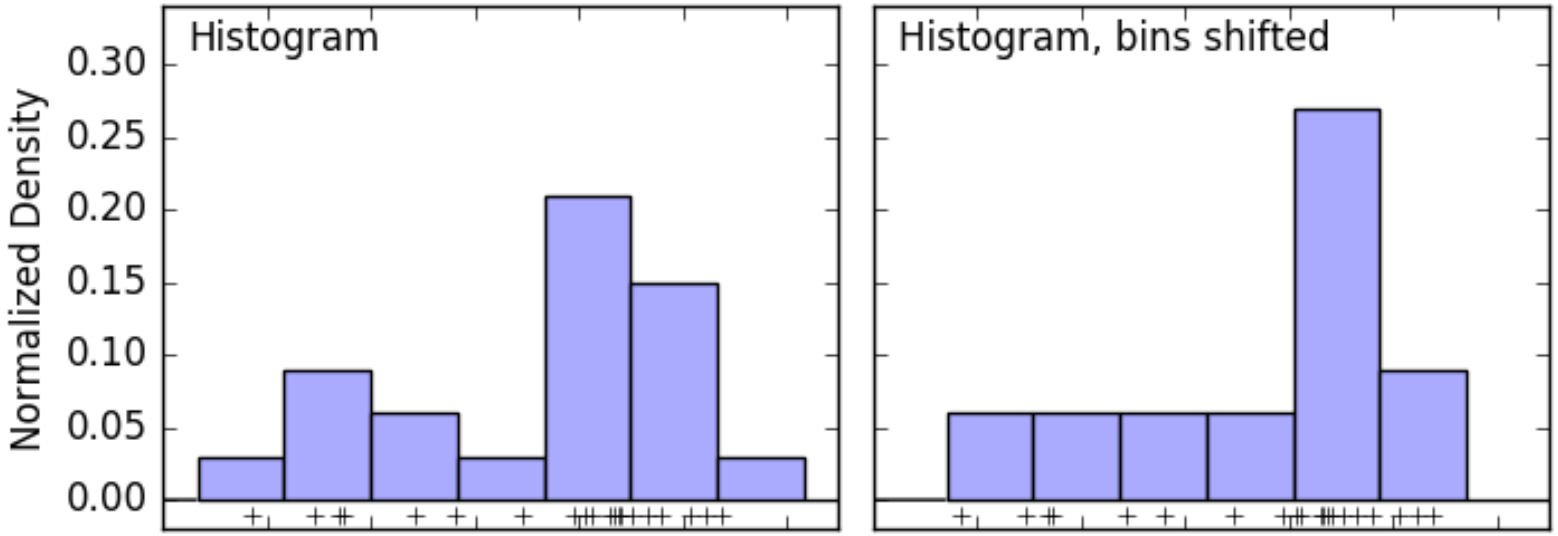
直方图的局限性

形状高度依赖于**箱宽 (Bin Width) 的**选择

呈现不连续的“阶梯状”，**不够平滑**

箱的分界位置可能掩盖数据的真实特征

难以直观看出分布的整体趋势



核密度估计：核心思想

核心目标

用一条平滑的曲线，近似描述数据的**概率分布 (PDF)**。

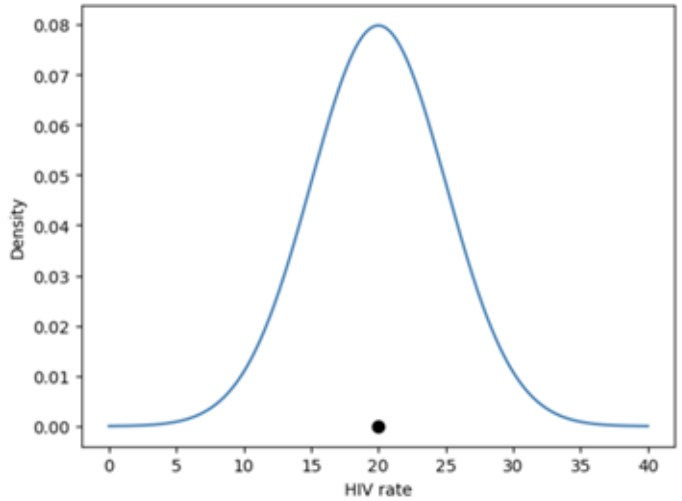
基本理念

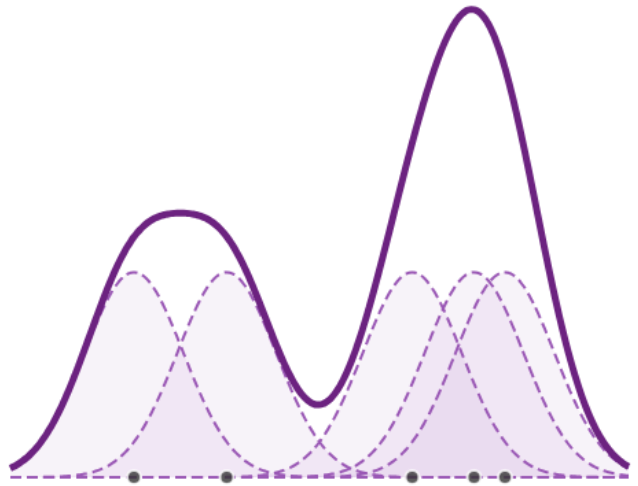
影响范围 (误差范围)： 承认观测值存在随机性。如果重新采样，数值可能会略有不同。因此，我们为每个数据点分配一个**“影响区域”**。

叠加效应： 将所有数据点的局部影响叠加起来，就形成了整体的分布估计。

直观理解

每个数据点就像平地上隆起的一个**“小山丘”**。
当所有小山丘堆叠在一起时，就形成了连绵起伏的**“地形”**（即最终的分布曲线）。





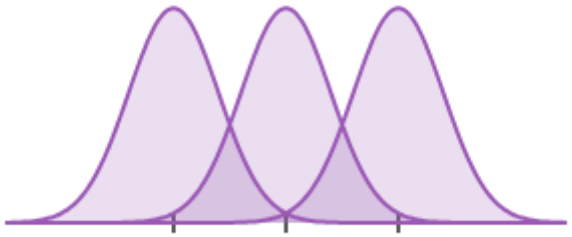
图示：“小山丘”叠加成“地形”

核密度估计：构建过程

核密度估计 (KDE) 的核心思想是近似生成数据的概率分布。
整个构建过程可以分为以下三个关键步骤：

步骤 1

在每个数据点放置核函数



以每个数据点为中心
放置一个核函数（如高斯核）

步骤 2

归一化核函数

缩放每个核函数
使总面积积分为 1

步骤 3

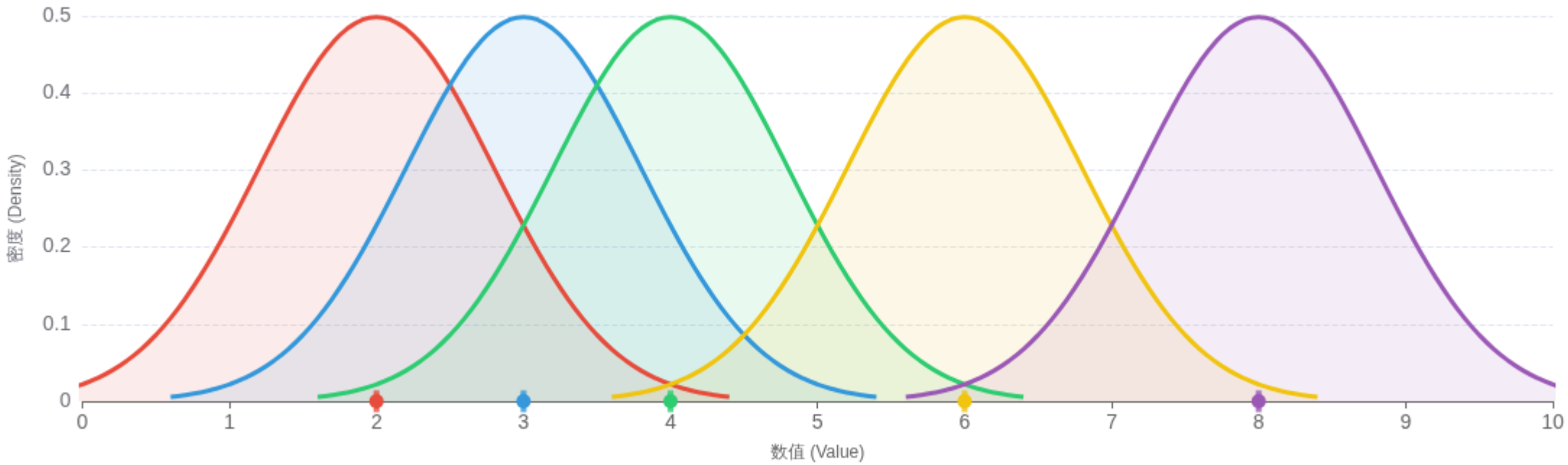
求和所有核函数

叠加所有归一化的核
生成最终的概率密度曲线

步骤 1：在每个数据点放置核函数

假设我们有5个数据点（例如HIV感染率数据）：**2, 3, 4, 6, 8**

我们在每个数据点的位置放置一个**高斯核函数**（钟形曲线）。
核函数捕捉了采样的随机性。每个核函数的面积都积分为 1。



步骤 2：归一化核函数

为什么需要归一化？

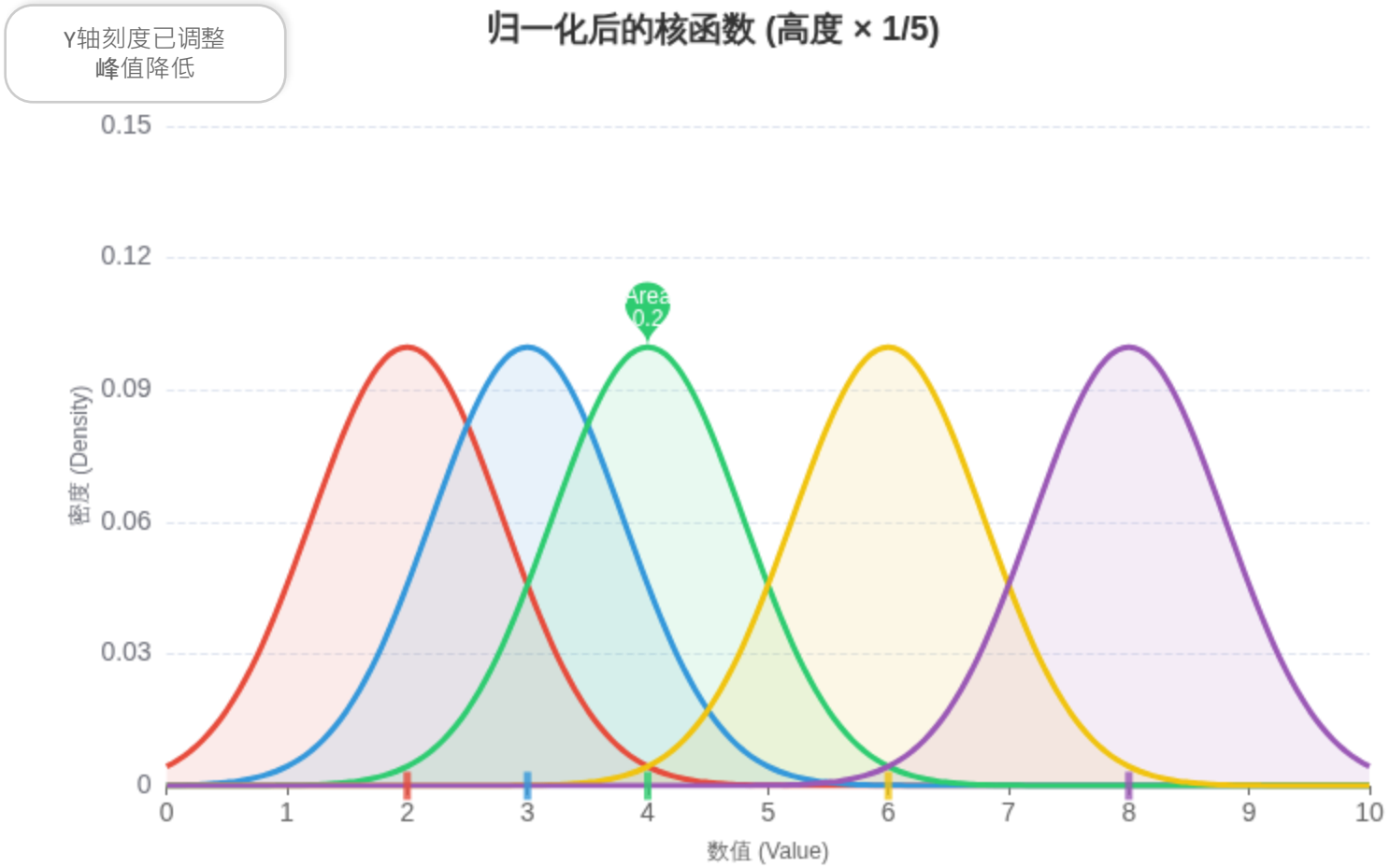
在下一步骤中，我们将对所有核函数求和以生**成最**终分布。

为了确保最终结果是一个有效的概率分布（**曲**线下的总面积必须等于 **1**），**我**们需要先对**每个核函数**进行缩放。

计算方法：

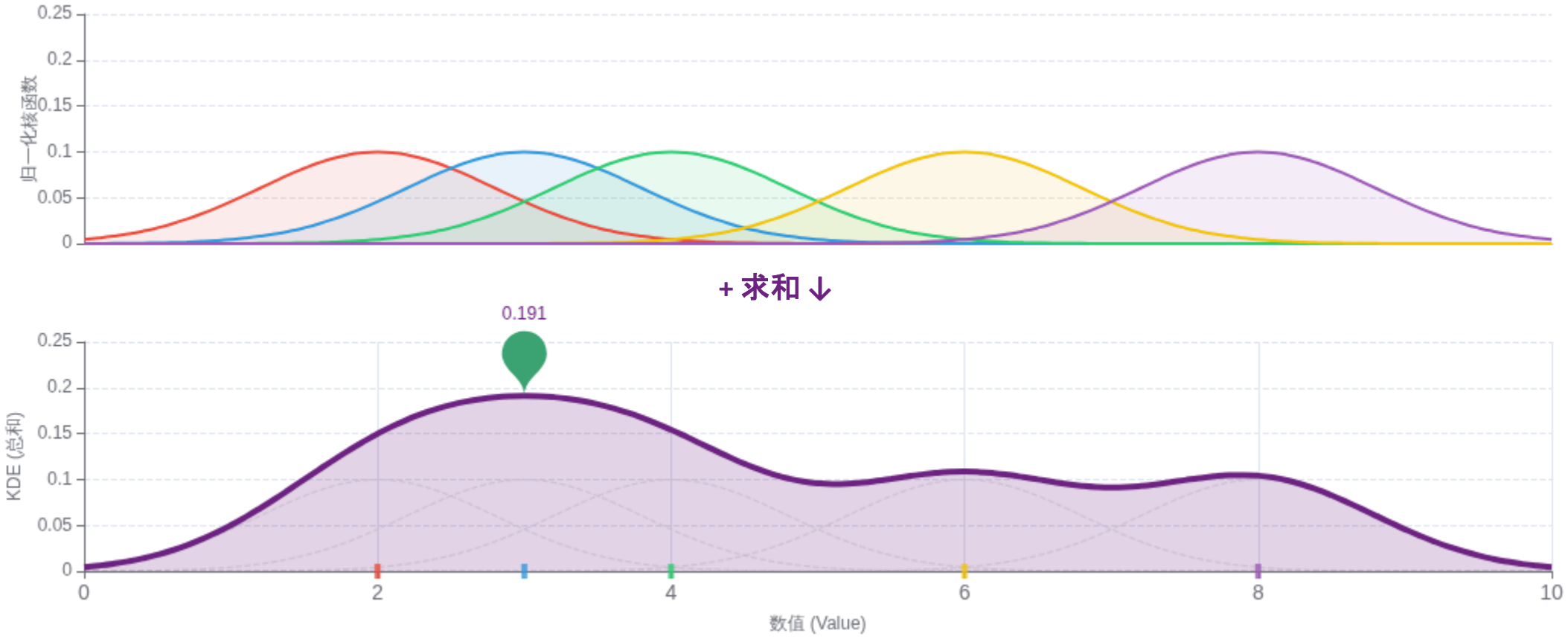
我们有 **5 个核函数**。
每个原始核的面积 = 1。
归一化因子 = **1/5**。

新面积 = 1 × (1/5) = 0.2



步骤 3：求和所有归一化核函数

在分布的每一点，将所有 5 个归一化核函数的值**相加**，得到一条平滑曲线。



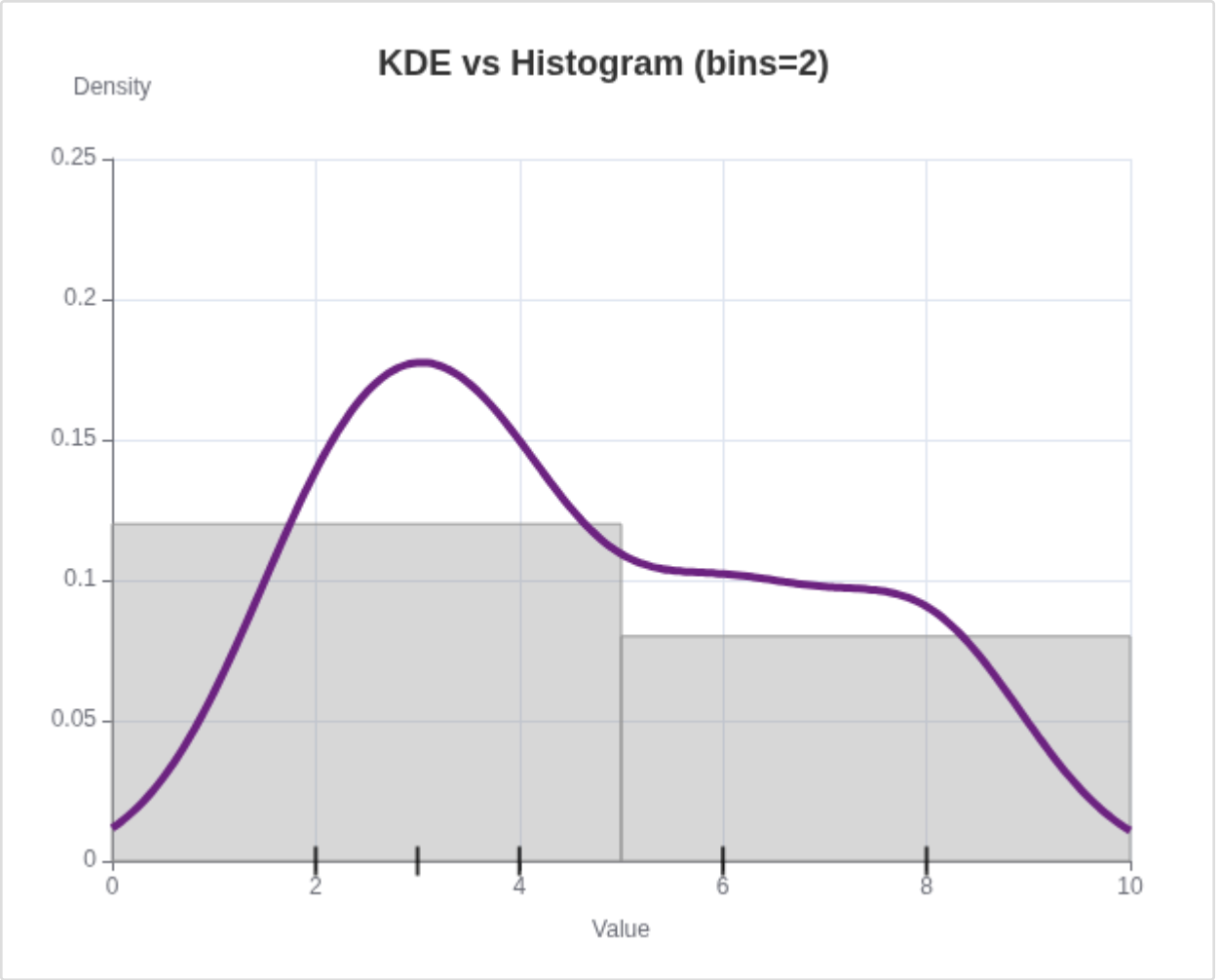
KDE结果：平滑的分布估计

- 最终的 **KDE 曲线** 是一条平滑的概率密度估计曲线
- 曲线在任意一点的高度（密度），对应于该点处所有归一化核函数值的**总和**。

```
# Seaborn 实现代码

sns.kdeplot(points, bw_method=0.65)

sns.histplot(
  points,
  stat="density",
  bins=2
 )
```



KDE的另一种实现方法

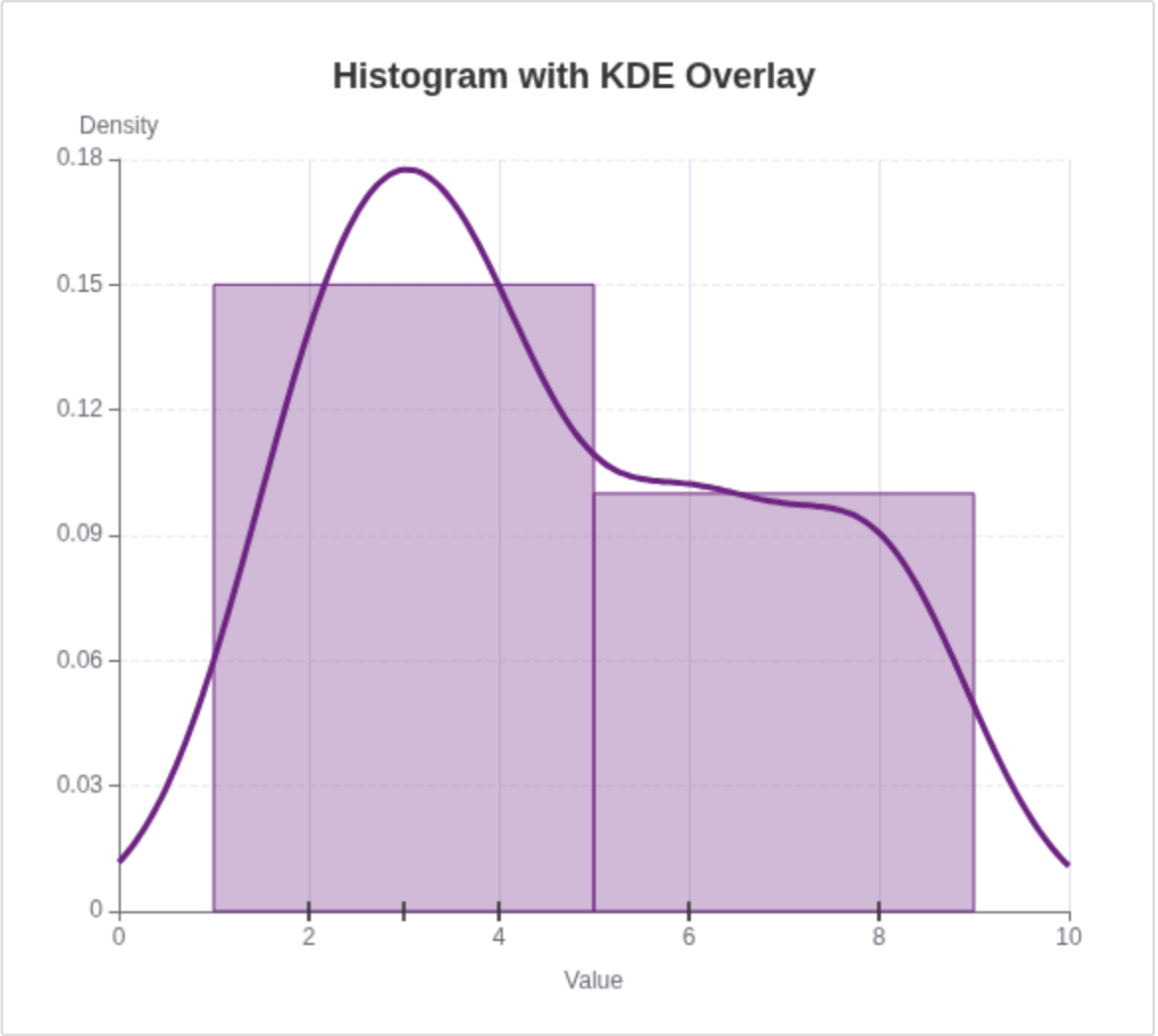
除了专门的 kdeplot，seaborn 的 histplot 函数也可以同时绘制直方图和 KDE 曲线。

```
sns.histplot(
 points,
 bins=2,
 kde=True,
 stat="density",
 kde_kws=dict(cut=3, bw_method=0.65)
 )
```

kde=True：启用 KDE 曲线叠加，在直方图上绘制平滑的密度估计。

cut：控制曲线延伸范围（超出数据极值的距离）。

bw_method：控制带宽（平滑程度）。

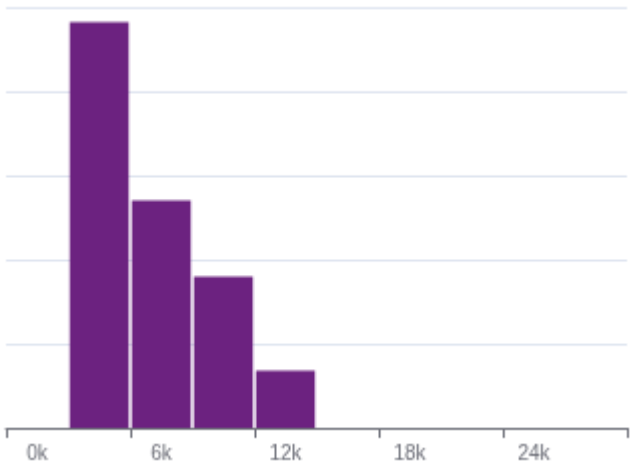


displot：绘制分布的通用函数

displot 是 Seaborn 中绘制分布的通用包装函数，通过调整 `kind` **参数，可以灵活**绘制直方图、KDE 和 ECDF。

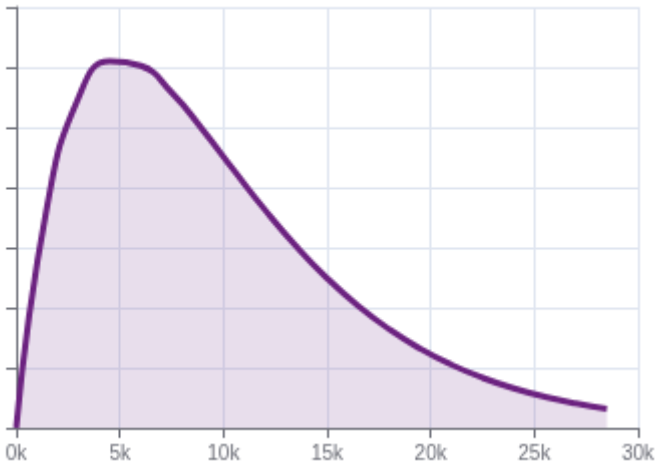
1. 直方图 (Histogram)

```
sns.displot(
data=wb, x="gni",
kind="hist",
stat="density")
```



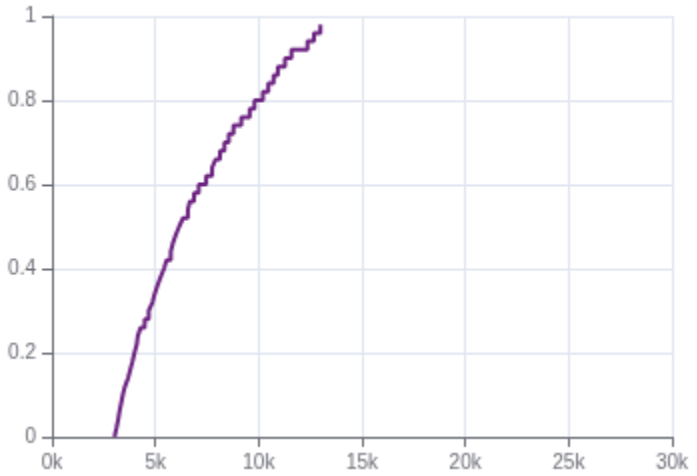
2. 核密度估计 (KDE)

```
sns.displot(
data=wb, x="gni",
kind="kde")
```



3. 经验累积分布 (ECDF)

```
sns.displot(
data=wb, x="gni",
kind="ecdf")
```



核密度估计：数学公式

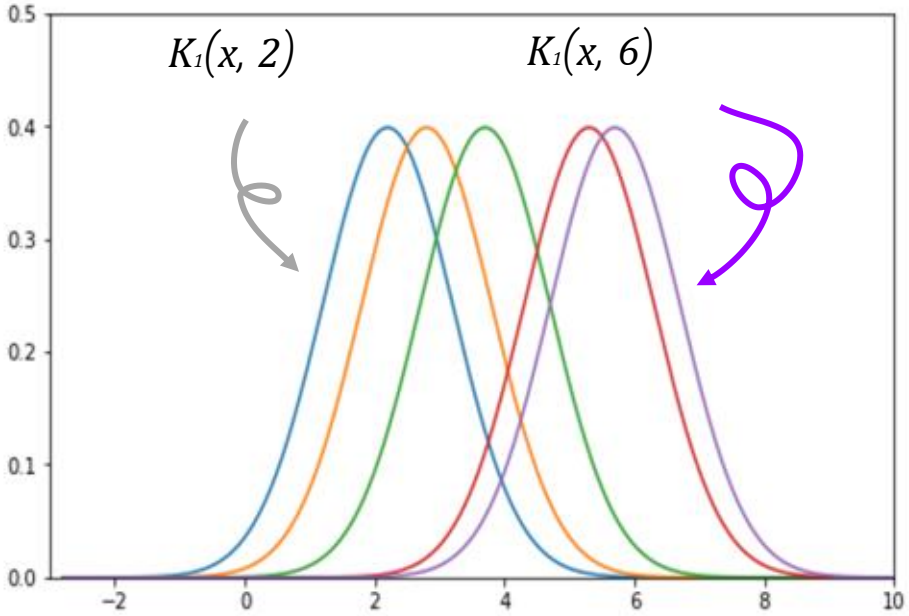
这是一个通用的 KDE **公式函数**：

$$f_\alpha(x) = \frac{1}{n}\sum_{i=1}^{n} K_\alpha(x, x_i)$$

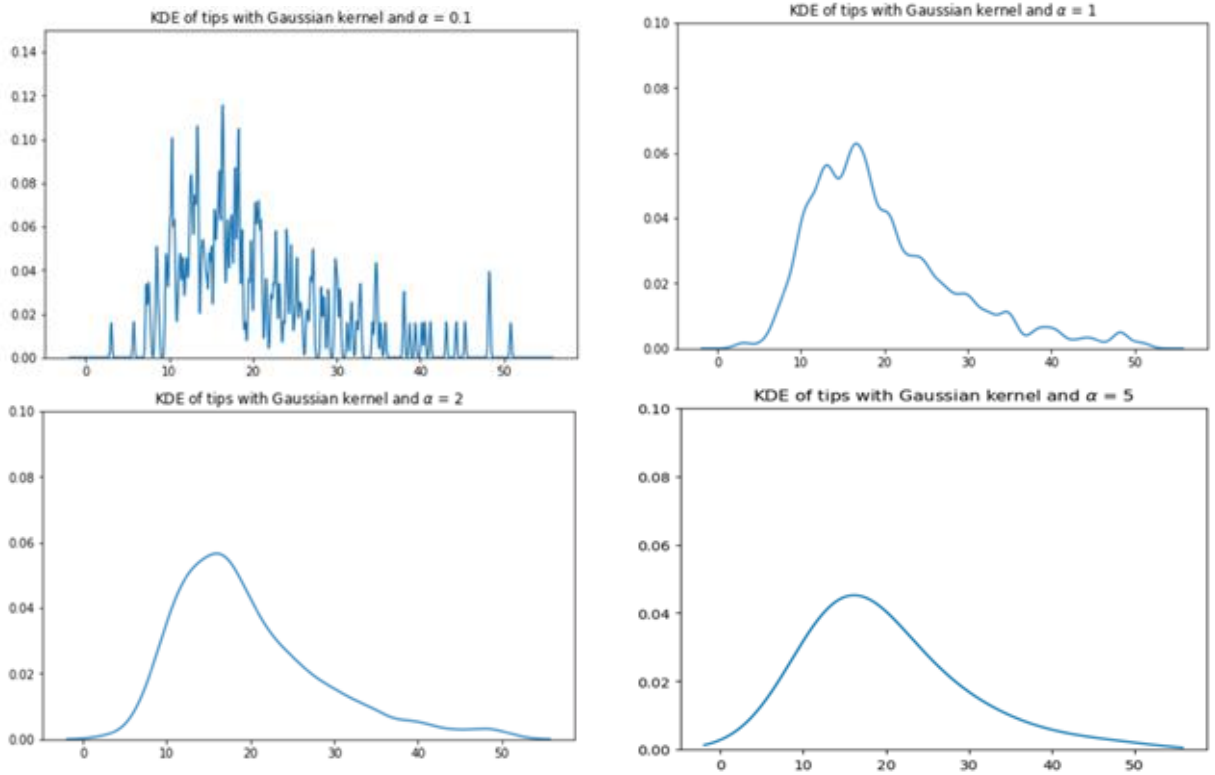
$K\alpha(x, xi)$ **是以**观测点 xi 为中心的核函数。

每个核函数单独的面积积分为 1。

K **代表我**们选择的核函数（稍后讨论数学形式）。



带宽参数的影响



带宽 (α)类比于直方图中的**箱宽 (Bin Width)**。
它控制了核函数的宽度，直接决定了最终KDE曲线的**平滑程度**。

大带宽 (α Large)

效果： 曲线非常平滑。
优点： 简单易懂，消除噪声。
缺点：丢失重要信息。例如图中的双峰结构被掩盖，看起来像单峰分布。

小带宽 (α Small)

效果： 曲线起伏剧烈。
优点： 保留所有细节。
缺点：过拟合噪声。曲线看起来像锯齿，难以分辨真实的分布特征。