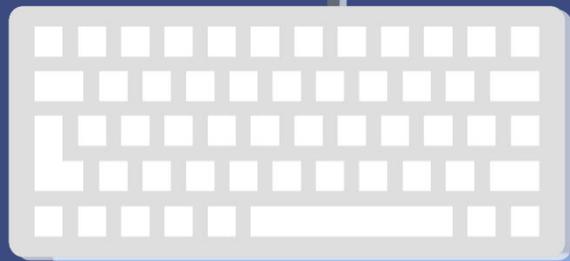




# Seaborn





## 学习目标

- 了解Seaborn库的基本概念和功能。
- 学习如何安装和配置Seaborn库。
- 掌握绘制计数图、箱线图、柱形图等图形的方法。
- 掌握绘制回归关系图、小提琴图、折线图等图形的方法。
- 掌握绘制直方图、饼图、散点图等图形的方法。





# 目录/CONTENTS



Seaborn简介



绘制计数图



绘制箱线图



绘制折线图



绘制柱形图



# Seaborn简介

## Seaborn介绍

- Seaborn是建构在Matplotlib基础上的Python数据可视化库，致力于简化统计图表的创建过程。
- 作为Matplotlib的扩展，Seaborn提供了更高层次的接口，使用户能够轻松地生成专业水平的图形。
- Seaborn支持多种高级统计图表，如散点图、箱线图、直方图和核密度估计图，有助于用户更全面地了解数据的分布、趋势和关系。

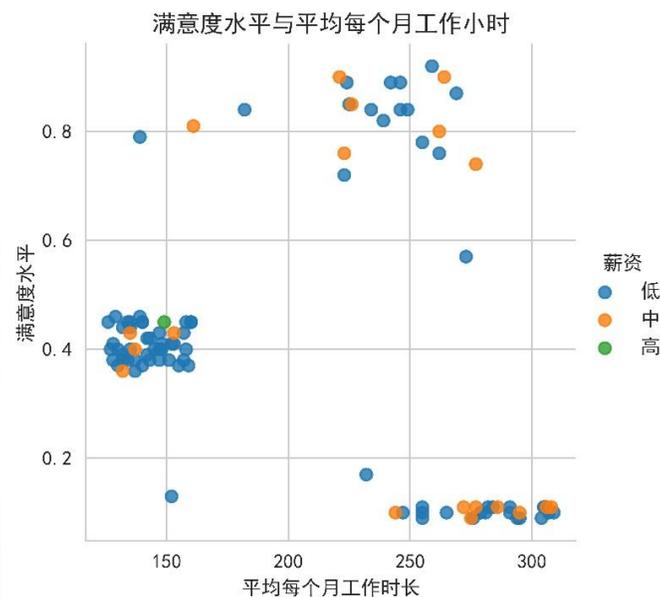
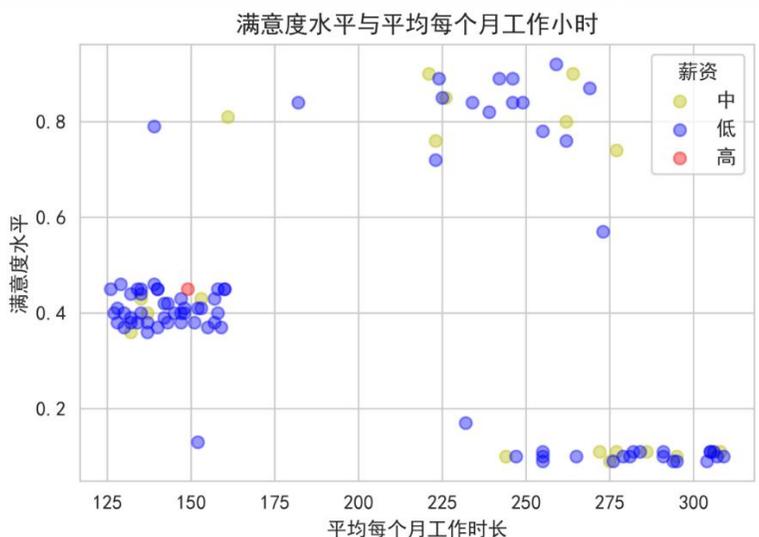




# Seaborn简介

## Seaborn基础图形

- seaborn库绘制的图形色彩、视觉令人耳目一新，通常将它视为Matplotlib库的补充，而不是替代物。seaborn库绘制的常用图形有散点图、折线图、热力图、条形图、箱线图、网格图等，在绘制这些图形之前需要掌握seaborn库的绘图基础，其中包括基础图形、绘图风格和调色板等。
- 在seaborn库中包含了大量常用的基础图形。以某企业人员离职率数据为例，分别使用Matplotlib库与seaborn库绘图不同薪资分布的散点图。





# Seaborn简介

## Seaborn特点

- 美观的默认样式
- 统计数据可视化
- 内置主题和颜色调色板
- 适用于数据框架
- 高级绘图功能
- 可视化统计模型结果
- 强大的色彩控制





## Seaborn与其他可视化库对比

### Seaborn对比Matplotlib

- 高级接口
- 美观性
- 颜色调色板

### Seaborn对比Plotly

- 静态vs交互式
- 语法简洁性
- 在线部署





# Seaborn简介

## Seaborn安装与配置

### 1.安装Seaborn

```
pip install seaborn
```

### 2.配置Seaborn

导入Seaborn

- `import seaborn as sns`

设置图形风格

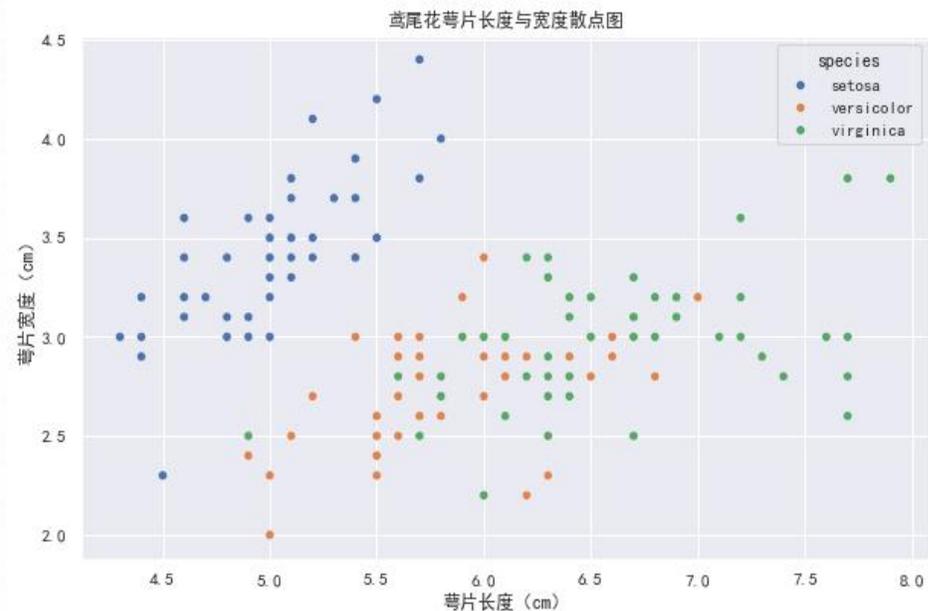
- `sns.set_style("whitegrid")`
- `sns.set(rc={'figure.figsize': (10, 6)})`





## Seaborn示例代码

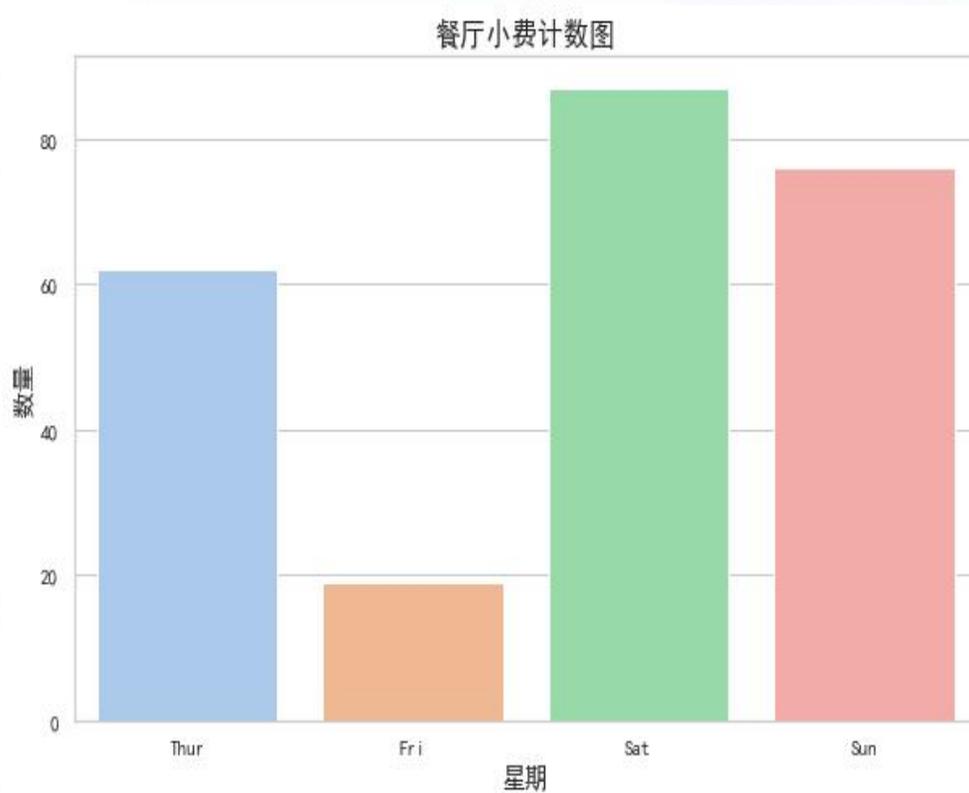
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_style("whitegrid")
sns.set(rc={'figure.figsize': (10, 6)})
data = sns.load_dataset("iris")
plt.rcParams['font.sans-serif'] = ['simhei']
sns.scatterplot(x="sepal_length",
y="sepal_width", hue="species", data=data)
plt.title('鸢尾花萼片长度与宽度散点图')
plt.xlabel('萼片长度 (cm)')
plt.ylabel('萼片宽度 (cm)')
plt.show()
```





## 计数图概述

- 在Seaborn中，计数图是一种用于显示分类变量的频率分布的图形。
- 它简单地计算数据集中每个类别的出现次数，并将结果可视化为条形图或柱形图。
- 计数图通常用于探索数据集中各个类别的分布情况，以便快速了解各类别的相对频率。





# 绘制计数图

## 计数图语法

- Seaborn中用于绘制计数图的主要函数是`countplot()`函数，以下是`countplot()`函数的完整形式及一些常用参数：

```
sns.countplot(x=None, hue=None, data=None, order=None, hue_order=None, orient=None, color=None, palette=None, saturation=0.75, dodge=True, ax=None, **kwargs)
```

参数	说明
x	指定X轴数据
hue	可选参数，用于对数据进行分组，可以是另一个分类变量，会以不同颜色表示
data	数据集，可以是 DataFrame 或其他数据结构
order	可选参数，指定X轴上分类变量的顺序
hue_order	可选参数，指定 hue 分组的顺序
orient	可选参数，控制绘图的方向，可以是 "v"（垂直）或 "h"（水平）
color	可选参数，用于设置图形的颜色
palette	可选参数，用于设置颜色调色板



# 绘制计数图

## 计数图案例实战1

**案例：**花销类别计数可视化

**背景：**分析一组关于个人月度花销的数据，数据包括不同的花销类别及其对应的金额。为了更好地理解花销的分布情况，需要统计每个花销类别的出现次数，并将结果可视化。

**任务：**

**数据准备：**创建一个包含花销类别和相应金额的数据集。花销类别包括“食物”、“交通”、“住房”、“娱乐”和“其他”。请自行设计一组示例数据，确保每个类别至少出现一次。

**数据处理：**使用Pandas库将数据组织成一个DataFrame。

**可视化：**使用Seaborn库绘制一个条形图，显示每个花销类别的出现次数。要求图形的大小为8x6英寸，颜色方案选择“pastel”。

**图表美化：**为图表添加标题“花销类别计数图”，并设置x轴标签为“花销类别”，y轴标签为“数量”。

**展示结果：**显示图表。

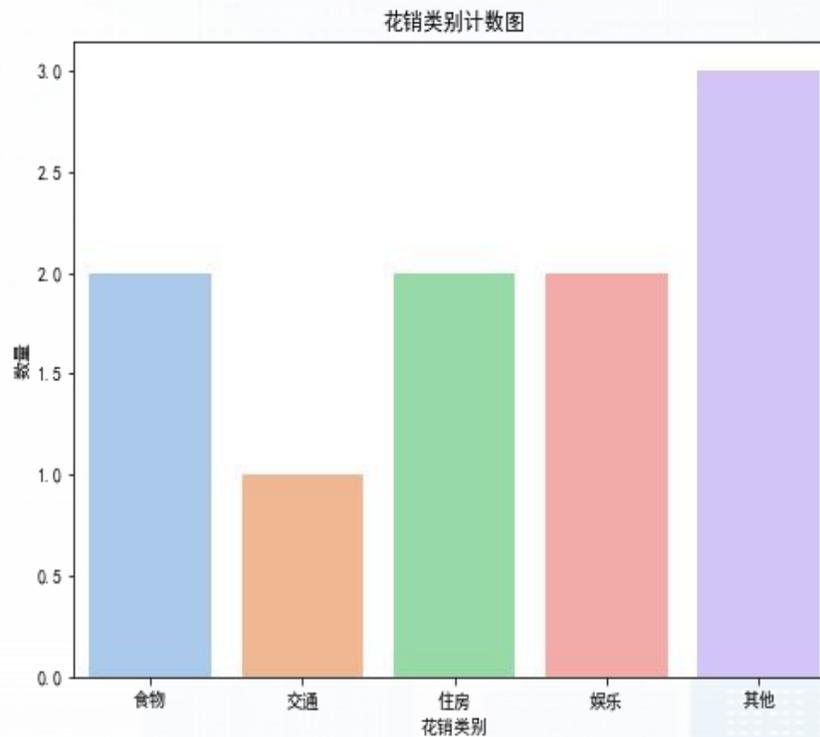


# 绘制计数图

## 计数图案例实战1

```
import seaborn as sns
import matplotlib.pyplot as plt

categories = ['食物', '交通', '住房', '娱乐', '其他', '食物', '住房', '娱乐', '其他', '其他']
expenses = [500, 200, 800, 300, 100, 400, 700, 150, 200, 300]
data = {"Category": categories, "Expenses": expenses}
df = pd.DataFrame(data)
plt.figure(figsize=(8, 6))
sns.countplot(x='Category', data=df, palette='pastel')
plt.title("花销类别计数图")
plt.xlabel("花销类别")
plt.ylabel("数量")
plt.show()
```



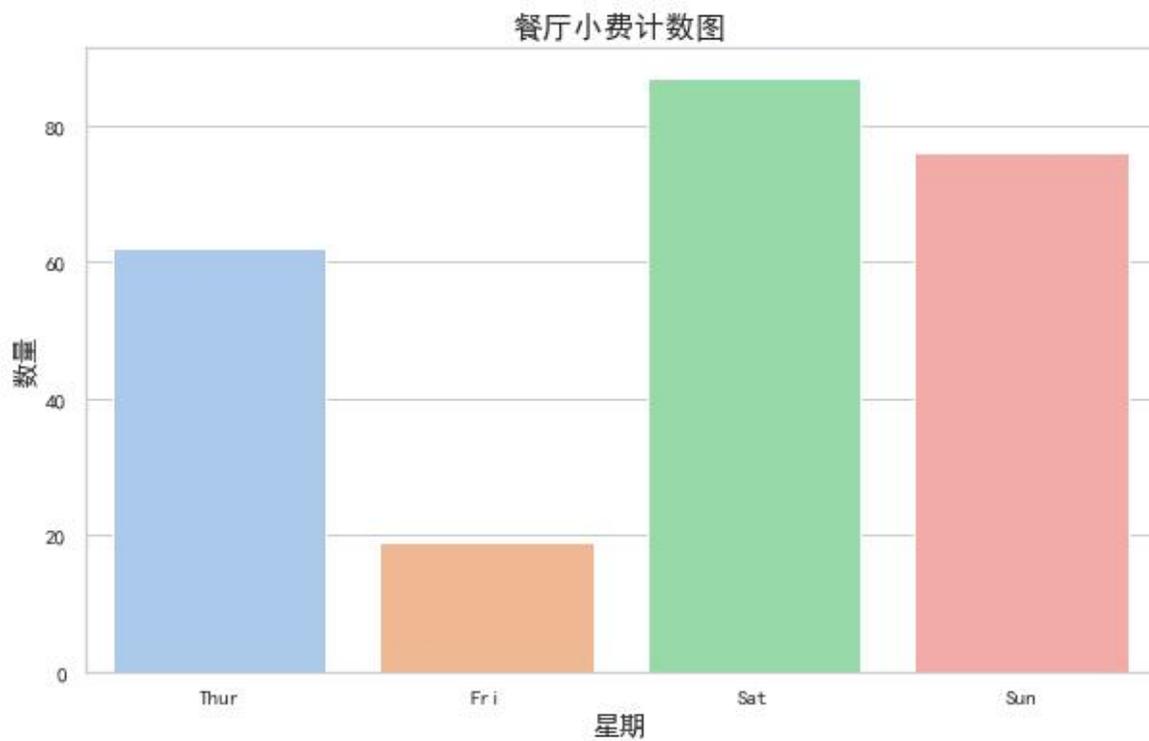


# 绘制计数图

## 计数图案例实战2

案例：餐厅小费计数

描述：通过计数图，我们可以看到每个星期的小费记录数量，从而对小费的分布情况有一个直观的认识。





## >>> 计数图案例实战2

```
import seaborn as sns
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = 'SimHei'
data = sns.load_dataset('tips')
plt.figure(figsize=(10, 6))
sns.countplot(x='day', hue='day', data=data, palette='pastel', order=['Th
ur', 'Fri', 'Sat', 'Sun'], legend=False)
plt.title("餐厅小费计数图", fontsize=16)
plt.xlabel("星期", fontsize=14)
plt.ylabel("数量", fontsize=14)
plt.show()
```



## 箱线图概述

- 箱线图，也被称为盒须图、盒式图，是一种用于显示数据分布情况的统计图。
- 它主要展示了一组数据的中位数、上下四分位数、最大值、最小值以及异常值（离群点）。





## 箱线图概述

- 箱线图，也被称为盒须图、盒式图，是一种用于显示数据分布情况的统计图。
- 它主要展示了一组数据的中位数、上下四分位数、最大值、最小值以及异常值（离群点）。



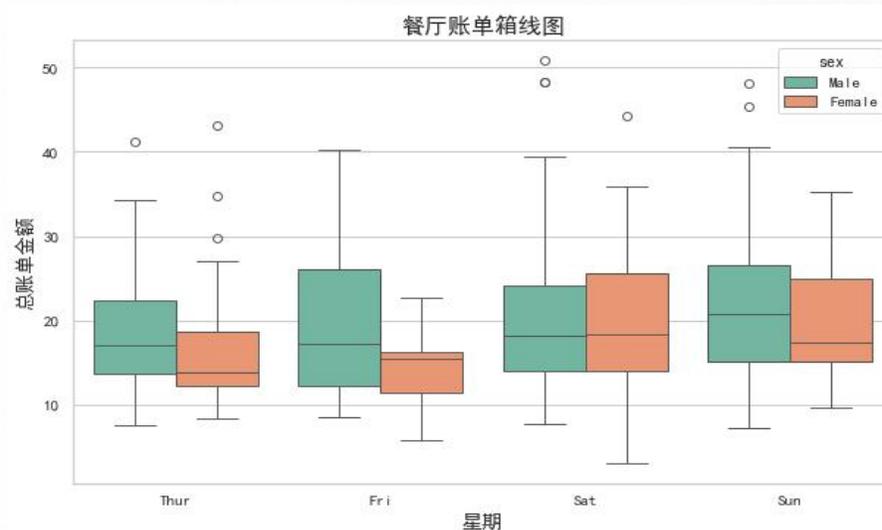


# 绘制箱线图

## 箱线图要素

箱线图的绘制过程包括以下几个要素：

- 箱体： 画出上下四分位数（Q1和Q3）之间的矩形，中间的线表示中位数（Q2）。
- 须： 从箱体延伸出来的线段，表示数据的范围。通常，须的长度由数据的分布情况确定，可以是数据的最小值和最大值，也可以根据一定的标准计算。
- 异常值： 在图表上以点的形式标出的数据点，它们偏离了常规的数据分布。异常值的定义通常是根据一定的统计学标准。





# 绘制箱线图

## 箱线图语法

Seaborn 中用于绘制箱线图的主要函数是 `boxplot ()` 函数，以下是 `boxplot` 函数的完整形式及一些常用参数：

```
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)
```

参数	说明
x	指定X轴数据
hue	可选参数，用于对数据进行分组，可以是另一个分类变量，会以不同颜色表示
data	数据集，可以是 DataFrame 或其他数据结构
order, hue_order	可选参数，用于指定X轴或 hue 分组的顺序
orient	可选参数，控制绘图的方向，可以是 "v"（垂直）或 "h"（水平）
color	可选参数，用于设置箱线图的颜色
palette	可选参数，用于设置颜色调色板
saturation	控制颜色的饱和度
width	控制箱线图的箱体宽度

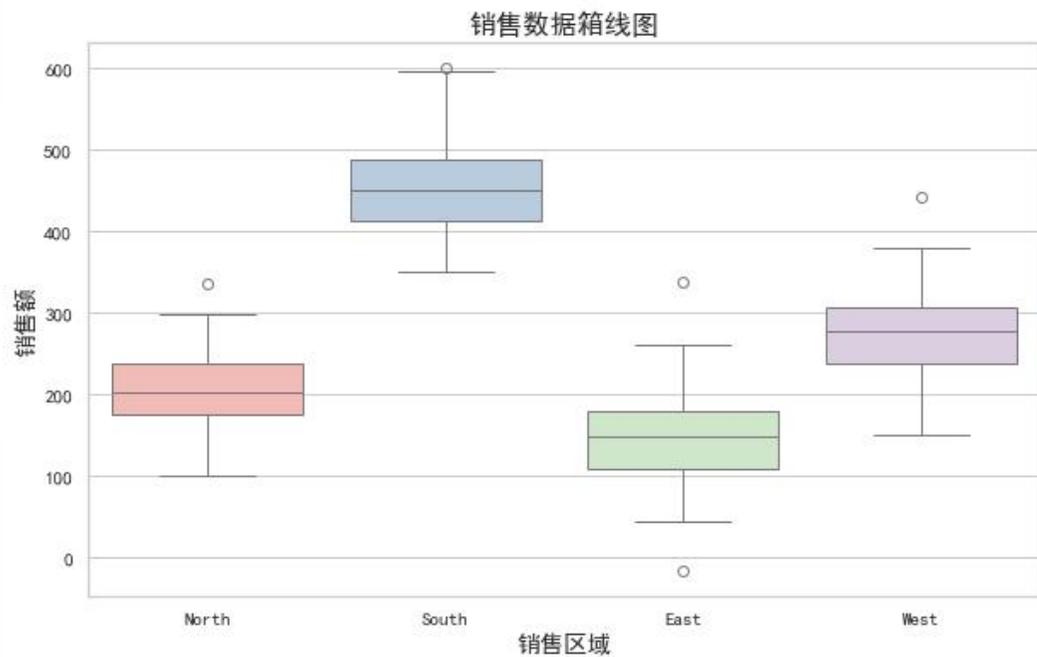


# 绘制箱线图

## 箱线图案例实战

### 案例介绍:

- 模拟包含不同销售区域的销售数据，并使用 Seaborn 的 `boxplot` 函数绘制了箱线图。箱线图展示了每个销售区域销售额的中位数、分布范围和可能的异常值。



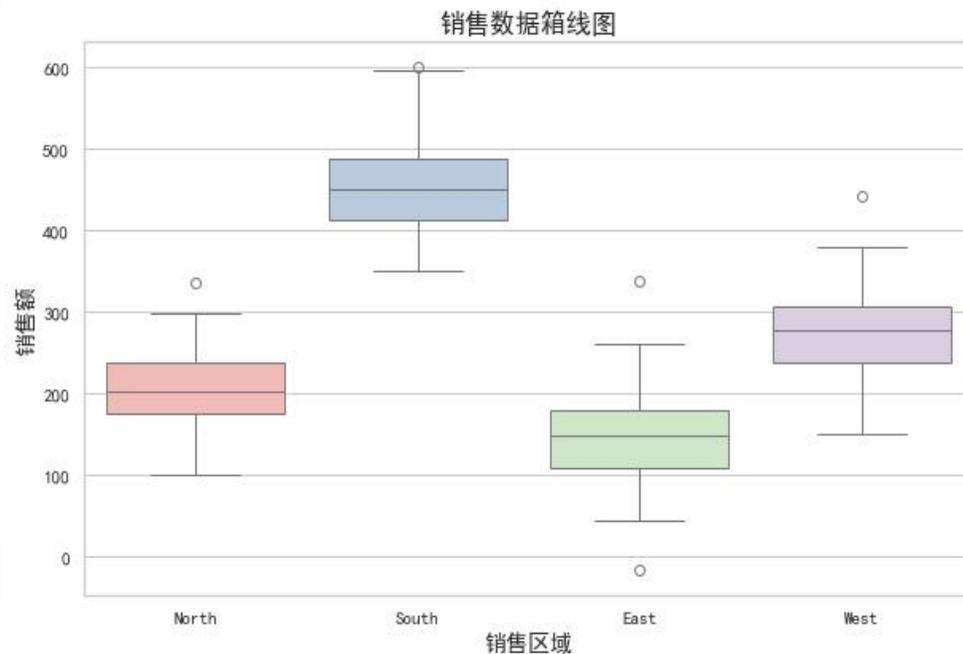


# 绘制箱线图

## 箱线图案例实战

案例代码

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
np.random.seed(42)
regions = ['North', 'South', 'East', 'West']
sales_data = {region: np.random.normal(np.random.r
andint(100, 500), 50, 100) for region in regions}
df_sales = pd.DataFrame(sales_data)
plt.figure(figsize=(10, 6))
sns.boxplot(data=df_sales, palette="Pastel1")
plt.title("销售数据箱线图", fontsize=16)
plt.xlabel("销售区域", fontsize=14)
plt.ylabel("销售额", fontsize=14)
plt.show()
```





# 绘制柱形图

## 柱形图概述

- 柱形图，又称长条图、柱状统计图、条图、条状图、棒形图，是一种以长方形的长度为变量的统计图表。长条图用来比较两个或以上的价值（不同时间或者不同条件），只有一个变量，通常利用于较小的数据集分析。长条图亦可横向排列，或用多维方式表达。





# 绘制柱形图

## 柱形图语法

Seaborn 中用于绘制散点图的主要函数是 `barplot()`，以下是 `barplot` 函数的完整形式及一些常用参数：  
`seaborn.barplot(x=None, y=None, hue=None, data=None, estimator=np.mean, ci=95, orient=None, color=None, palette=None)`

参数	说明
<code>x</code>	指定 X 轴数据
<code>y</code>	指定 Y 轴的数据
<code>hue</code>	指定数据点颜色的变量（分类变量）
<code>data</code>	数据集
<code>estimator=np.mean</code>	聚合函数
<code>ci=95</code>	置信区间的大小，默认为 95%
<code>orient</code>	条的方向，可以是 'v'（垂直）或 'h'（水平）
<code>color</code>	条的颜色
<code>palette</code>	颜色调色板

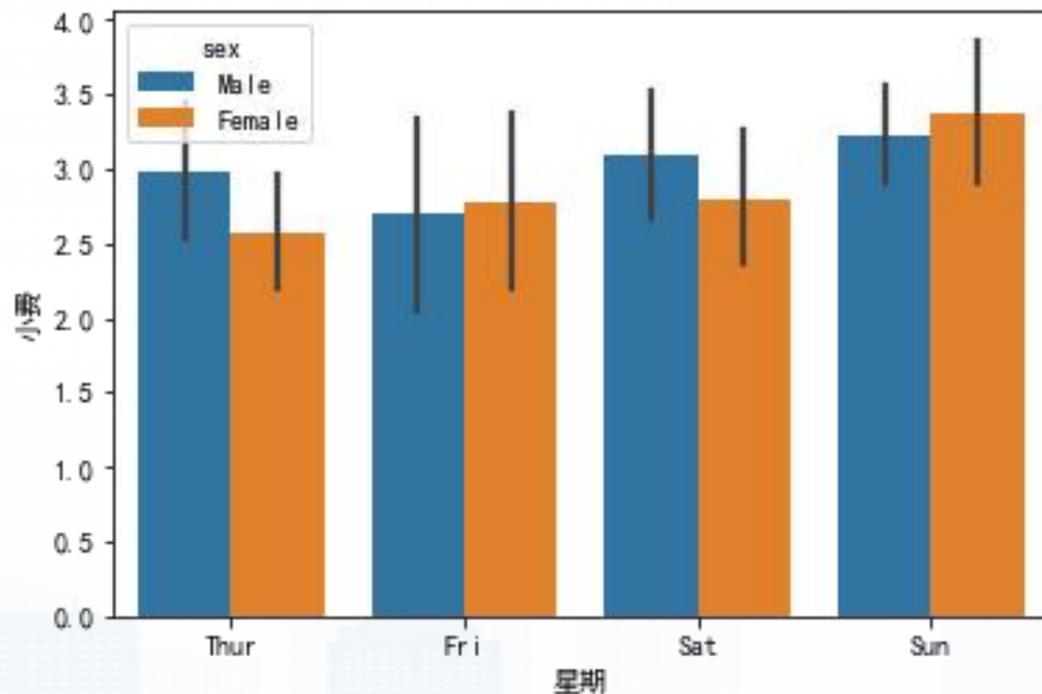


# 绘制柱形图

## 柱形图案例实战1

### 案例：星期与小费金额的分布

**案例描述：** 绘制柱状图展示在不同星期中男性和女性给出的小费金额，以探索这些因素之间的关系游客(百万人次)”行数据绘制柱形图。





# 绘制柱形图

## 柱形图案例实战1

```
import matplotlib.pyplot as plt
import seaborn as sns
data = sns.load_dataset('tips')
# 设置中文显示
plt.rcParams['font.sans-serif'] = 'SimHei' # 选择适合中文显示的字体, 这里使用了SimHei
sns.barplot(x='day', y='tip', hue='sex', data=data)
plt.xlabel("星期")
plt.ylabel("小费")
plt.show()
```





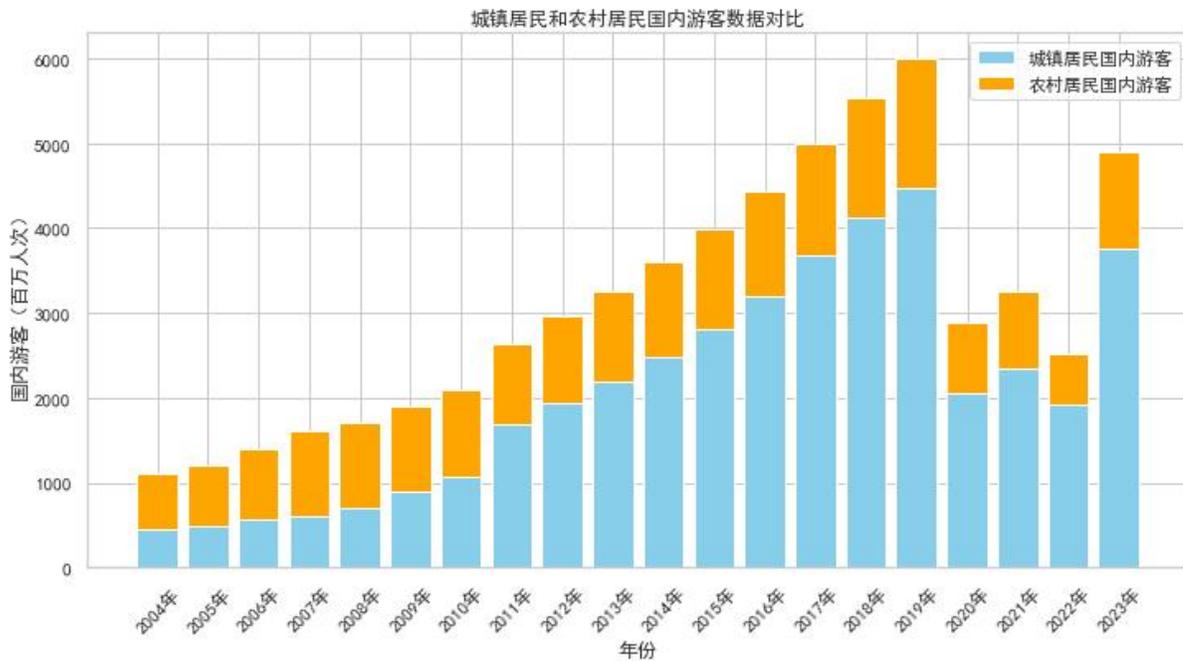
# 绘制柱形图

## 柱形图案例实战2

### 案例：城镇居民和农村居民国内游客数据对比

#### 案例描述：

读取“data.xls”表中“国内旅游情况”表单数据，并根据历年“城镇居民国内游客(百万人次)”和“农村居民国内





# 绘制柱形图

## 柱形图案例实战2

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_Excel('data.xls', sheet_name='国内旅游情况')
subset = data.iloc[1:3, 1:]
plt.figure(figsize=(12, 6))
plt.bar(subset.columns, subset.iloc[0], color='skyblue', label='城镇居民国内游客')
plt.bar(subset.columns, subset.iloc[1], color='orange', label='农村居民国内游客',
bottom=subset.iloc[0])
plt.title('城镇居民和农村居民国内游客数据对比')
plt.xlabel('年份')
plt.ylabel('国内游客（百万人次）')
plt.xticks(rotation=45)
plt.legend()
plt.show()
```





# 绘制回归关系图

## 回归关系图概述

- 回归关系图是一种用于可视化两个变量之间关系的图表，通常用于显示两个连续变量之间的线性关系，并估计它们之间的线性回归模型。这种图表可以帮助我们观察两个变量之间的趋势、相关性以及可能存在的线性关系。





# 绘制回归关系图

## 回归关系图语法

Seaborn 中用于绘制回归关系图的主要函数是 `regplot()` 函数，以下是 `regplot()` 函数的完整形式及一些常用参数：`sns.regplot(x, y, data=None, scatter_kws=None, line_kws=None)`

参数	说明
<code>x</code>	指定回归图中用作 X轴的数据，可以是数据集中的列名或数组
<code>y</code>	指定回归图中用作 Y轴的数据，可以是数据集中的列名或数组
<code>data</code>	DataFrame，可选参数，用于绘图的数据集。如果未提供，则需要在 <code>x</code> 和 <code>y</code> 中指定数组
<code>scatter_kws</code>	字典，可选参数，用于传递给散点图部分的关键字参数。
<code>line_kws</code>	字典，可选参数，用于传递给回归线部分的关键字参数

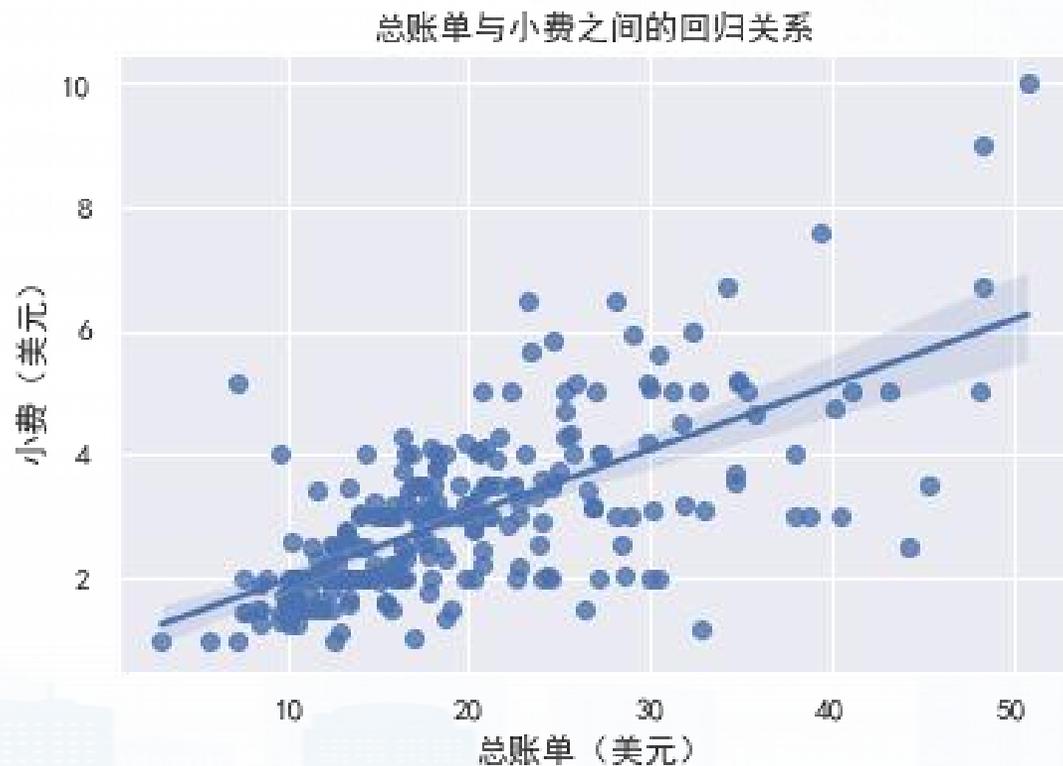


# 绘制回归关系图

## 回归关系图案例实战1

案例：总账单与小费之间的回归关系

案例描述：通过回归关系，用于显示总账单金额与小费金额之间的线性关系。





## 回归关系图案例实战1

```
import seaborn as sns
import matplotlib.pyplot as plt
tips = sns.load_dataset("tips")
plt.rcParams['font.sans-serif'] = ['SimHei'] # 选择适合中文显示的字体，这里使用了SimHei
sns.regplot(x="total_bill", y="tip", data=tips)
plt.title('总账单与小费之间的回归关系')
plt.xlabel('总账单（美元）')
plt.ylabel('小费（美元）')
plt.show()
```





# 绘制小提琴图

## 绘制小提琴图概述

- 小提琴图（Violin Plot）是一种用于可视化数值数据分布的统计图表。它结合了箱线图和核密度估计图的特点，能够显示数据的中位数、四分位数、离群值以及整体数据分布的形状。





# 绘制小提琴图

## 小提琴图语法

在Seaborn中，小提琴图（Violin Plot）是一种用于可视化数值数据分布的统计图表。它结合了箱线图和核密度估计图的特点，能够显示数据的中位数、四分位数、离群值以及整体数据分布的形状。

x=None,	数据在X轴上的变量（类别）
y=None,	数据在Y轴上的变量
hue=None,	分组变量，可以根据它分组绘制不同颜色的小提琴
data=None,	数据集
order=None,	控制X轴上类别的顺序
hue_order=None,	控制分组变量的顺序
bw='scott',	控制小提琴图的平滑程度
cut=2,	控制小提琴图的截断位置，可以是数值或字符串
scale='area',	控制小提琴图的宽度
scale_hue=True,	是否对每个分组变量的小提琴进行缩放
gridsize=100,	控制核密度估计的精度
width=0.8,	控制小提琴的宽度
inner='box',	控制小提琴内部的样式，可以是'box'、'quartile'、'point'等

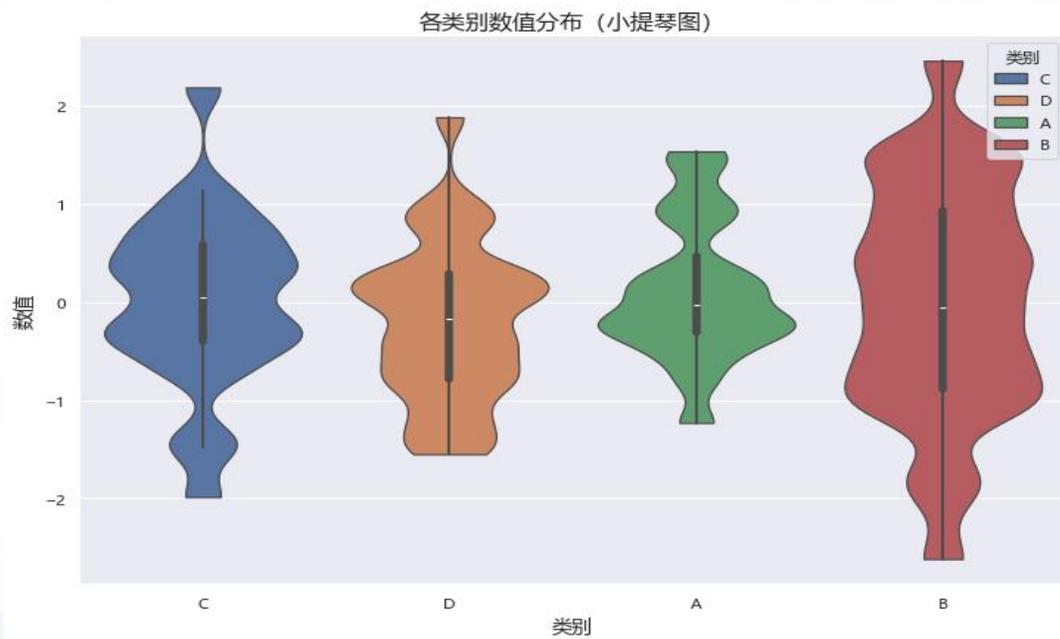


# 绘制小提琴图

## 小提琴图案案例实战1

案例：不同类别数据分布

案例描述：通过关系图绘制一个各类别数值分布的小提琴图，展示了不同类别数据的分布情况。





## 小提琴图案案例实战1

```
import seaborn as sns
import matplotlib.pyplot as plt
iris_data = sns.load_dataset("iris")
plt.figure(figsize=(10, 6))
sns.violinplot(x="species", y="petal_length", hue="species", data=iris_data, palette="Set2", legend=False)
plt.title("鸢尾花花瓣长度小提琴图", fontsize=16)
plt.xlabel("鸢尾花种类", fontsize=14)
plt.ylabel("花瓣长度", fontsize=14)
plt.show()
```





# 绘制折线图

## 折线图概述

- 折线图是一种用于显示数据随着连续变量的变化而变化的图形。在折线图中，数据点通过连接线条来展示趋势、变化或关系。
- 折线图常用于表示时间序列数据，显示变量随时间的演变。它能够清晰地展示数据的趋势和周期性。





# 绘制折线图

## 折线图语法

Seaborn 中用于折线图的主要函数是 `lineplot()`，以下是 `lineplot()` 函数的完整形式及一些常用参数：

```
seaborn.lineplot(x=None, y=None, hue=None, style=None, size=None, data=None, palette=None, markers=True, dashes=True, legend="auto", hue_order=None)
```

参数	说明
x	指定X轴数据
y	指定Y轴的数据
hue	指定数据点颜色的变量（分类变量）
style	指定数据点标记样式的变量（分类变量）
data	指定数据集
palette	指定颜色调色板
markers	是否显示标记，默认为 True
dashes	是否使用虚线
legend	是否显示图例，默认为 "auto"
hue_order	指定颜色变量的排序顺序



# 绘制折线图

## 折线图案例实战1

### 案例要求

- 描述：创建一个折线图来展示某商店半年的月销售额。
- 要求：使用Python的Seaborn和Matplotlib库，根据给定的数据绘制折线图，并添加适当的标题和轴标签。

### 销售数据分析

- 描述：分析并可视化某商店在六个月内的销售额变化。
- 要求：编写一个Python脚本，使用Seaborn库绘制折线图，展示每个月的销售额，并确保图表包含标记点、标题、X轴和Y轴的标签。

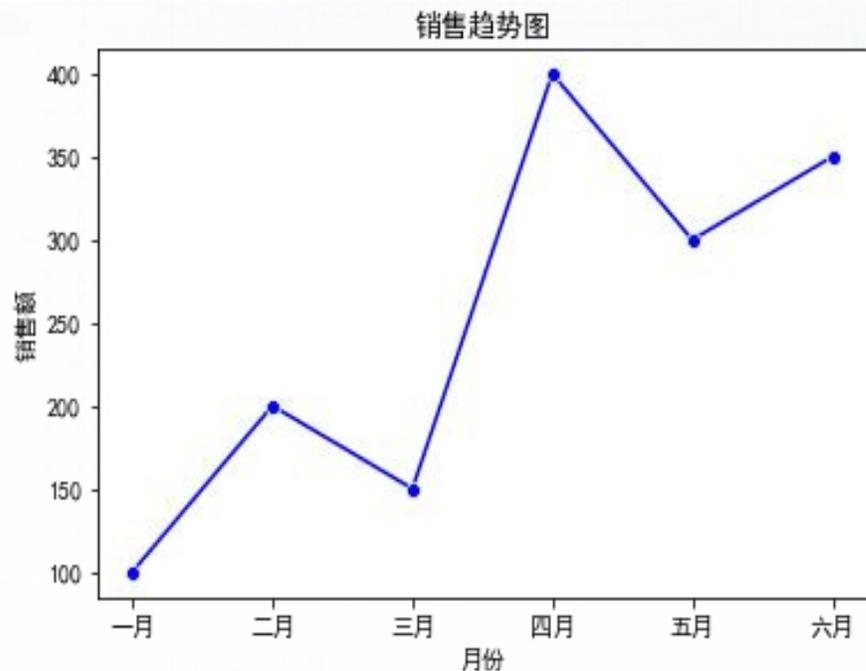




# 绘制折线图

## 折线图案例实战1

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False
months = ['一月', '二月', '三月', '四月', '五月', '六月']
sales = [100, 200, 150, 400, 300, 350]
sales_data = {"Month": months, "Sales": sales}
df_sales = pd.DataFrame(sales_data)
sns.lineplot(x="Month", y="Sales", data=df_sales,
marker='o', color='blue')
plt.title("销售趋势图")
plt.xlabel("月份")
plt.ylabel("销售额")
plt.show()
```



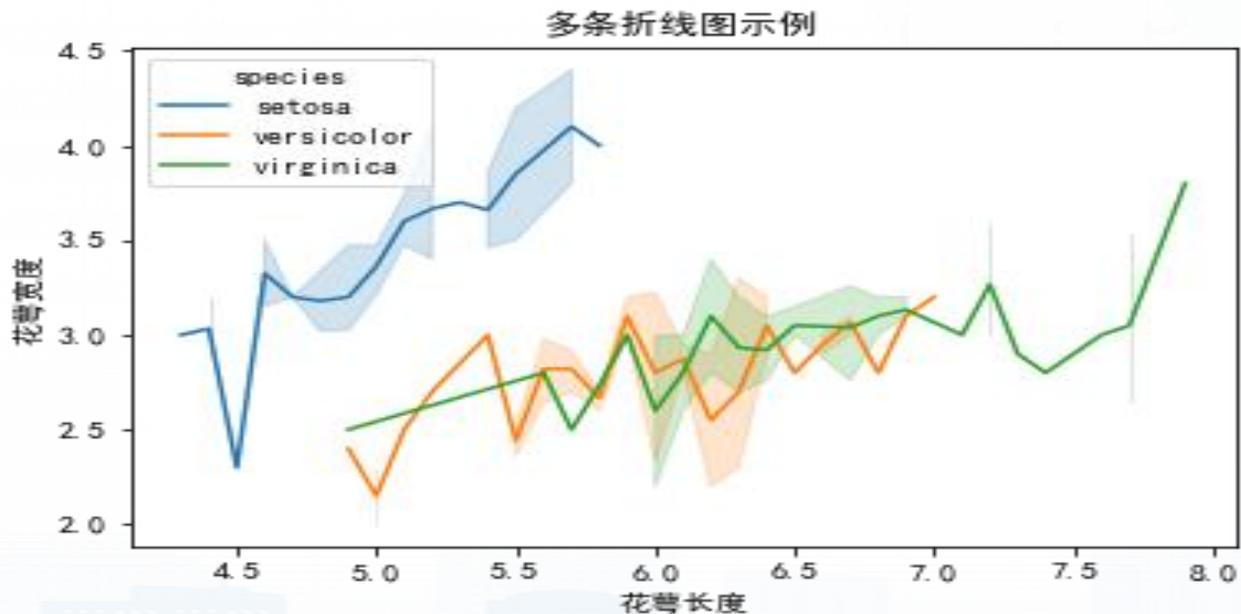


# 绘制折线图

## 折线图案例实战2

### 案例：鸢尾花数据分析

描述：展示鸢尾花花萼长度 (sepal\_length) 和花萼宽度 (sepal\_width) 之间的关系趋势。通过观察折线的走势，可以了解在不同花萼长度下，花萼宽度的变化情况。





## »» 折线图案例实战2

```
import seaborn as sns
import matplotlib.pyplot as plt
data = sns.load_dataset("iris")
sns.lineplot(x="sepal_length", y="sepal_width", hue="species", data=data)
plt.title("多条折线图示例")
plt.xlabel("花萼长度")
```

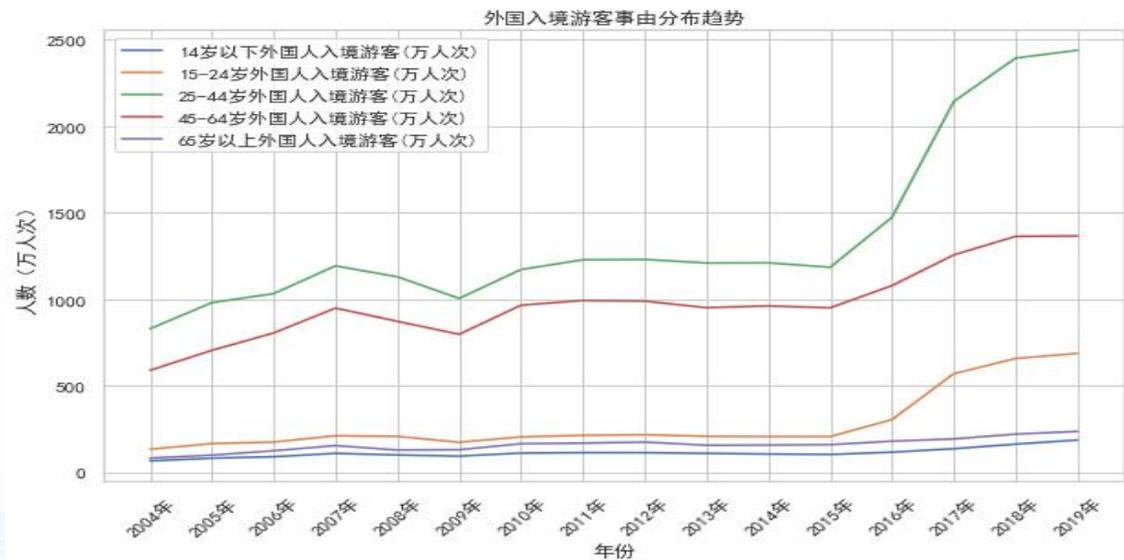


# 绘制折线图

## 折线图案例实战3

### 案例：旅游数据分析

描述：根据历年“14岁以下外国人入境游客(万人次)”、“15-24岁外国人入境游客(万人次)”、“25-44岁外国人入境游客(万人次)”、“45-64岁外国人入境游客(万人次)”、“65岁以上外国人入境游客(万人次)”行数据绘制折线图





### 折线图案例实战3

```
import pandas as pd
import matplotlib.pyplot as plt
from pylab import mpl
# 设置显示中文字体
mpl.rcParams["font.sans-serif"] = ["SimHei"]
data = pd.read_Excel('data.xls', sheet_name='按性别、年龄
和事由分外国入境游客')
subset = data.iloc[3:8]
plt.figure(figsize=(10, 6))
for index, row in subset.iterrows():
    plt.plot(data.columns[1:], row[1:], label=row[0])
plt.title('外国入境游客事由分布趋势')
plt.xlabel('年份')
plt.ylabel('人数 (万人次)')
plt.legend()
plt.xticks(rotation=45)
plt.show()
```



# 绘制直方图

## 直方图概述

- 直方图是一种统计图表，用于展示数据的分布情况。它将数据分成若干个等宽的区间（通常称为“箱子”或“柱子”），然后统计每个区间内数据的频数或频率。直方图的横轴通常表示数据的取值范围，纵轴则表示每个区间内数据的频数或频率。通过绘制这些区间的条形图，直方图能够直观地展示数据的分布特征。





# 绘制直方图

## 直方图语法

Seaborn 中用的直方图的主要函数是 `histplot()`，以下是 `histplot` 函数的完整形式及一些常用参数：  
`seaborn.histplot(x=None, y=None, hue=None, multiple='layer', binwidth=None, binrange=None, bins=10, kde=False, stat='count', common_bins=True, element='bars', fill=None)`

参数	说明
x	指定X轴数据
y	指定 Y轴的数据
hue	用于对数据进行分组的变量，根据该变量的不同取值绘制不同颜色的直方图
multiple	确定如何处理包含多个观测值的数据，默认为 'layer'，表示将直方图层叠在一起
binwidth	指定直方图条形的宽度
binrange	指定直方图条形的范围
bins	指定直方图的条形数量
kde	是否显示核密度估计曲线
stat	确定绘图时使用的统计量，默认为 'count'，表示绘制计数直方图
common_bins	是否将所有组使用相同的箱子，默认为 True
element	确定绘图元素的类型，默认为 'bars'，表示绘制条形直方图
fill	确定是否填充直方图条形的颜色。默认为 None，表示不填充

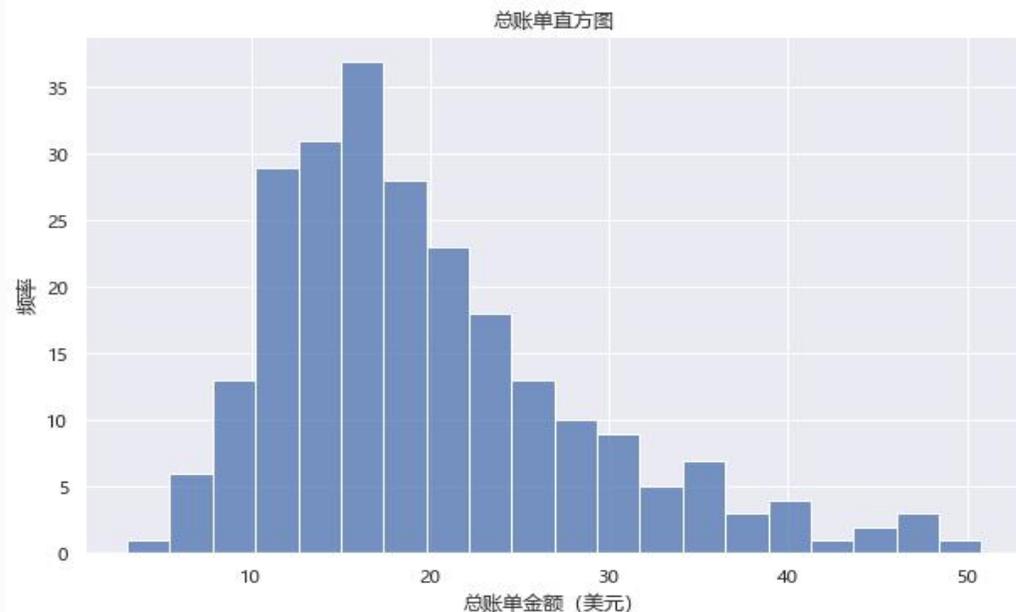


# 绘制直方图

## 直方图案例实战

### 案例：账单金额分布

案例描述：这段代码使用了Seaborn内置的数据集"tips"，该数据集包含了餐厅小费的相关信息。通过绘制直方图，分析了餐厅顾客的总账单金额的分布情况。





# 绘制直方图

## 直方图案例实战

```
import seaborn as sns
import matplotlib.pyplot as plt
tips = sns.load_dataset("tips")
sns.histplot(data=tips, x="total_bill", bins=20, kde=False)
plt.title('总账单直方图')
plt.xlabel('总账单金额 (美元)')
plt.ylabel('频率')
plt.show()
```



## 散点图概述

- 散点图是一种用于显示两个数值变量之间关系的图形。它通过在二维平面上以点的形式表示数据点，其中每个点的 X轴和 Y轴坐标分别对应两个变量的值。散点图可用于观察变量之间的趋势、集中程度、异常值以及变量之间的相关性。





# 绘制散点图

## 散点图语法

Seaborn 中用于绘制散点图的主要函数是 `scatterplot()`，以下是 `scatterplot` 函数的完整形式及一些常用参数：

```
seaborn.scatterplot(x=None, y=None, hue=None, style=None, size=None, data=None, palette=None, hue_order=None, hue_norm=None, sizes=None)
```

参数	说明
x	指定X轴数据
y	指定 Y轴的数据
hue	指定数据点颜色的变量（分类变量）
style	指定数据点标记样式的变量（分类变量）
palette	指定颜色调色板
markers	是否显示标记，默认为 True
sizes	指定不同大小的数据点
alpha	点的透明度，默认为 "auto"
legend	是否显示图例，默认为 "auto"，自动根据 hue 和 style 参数决定

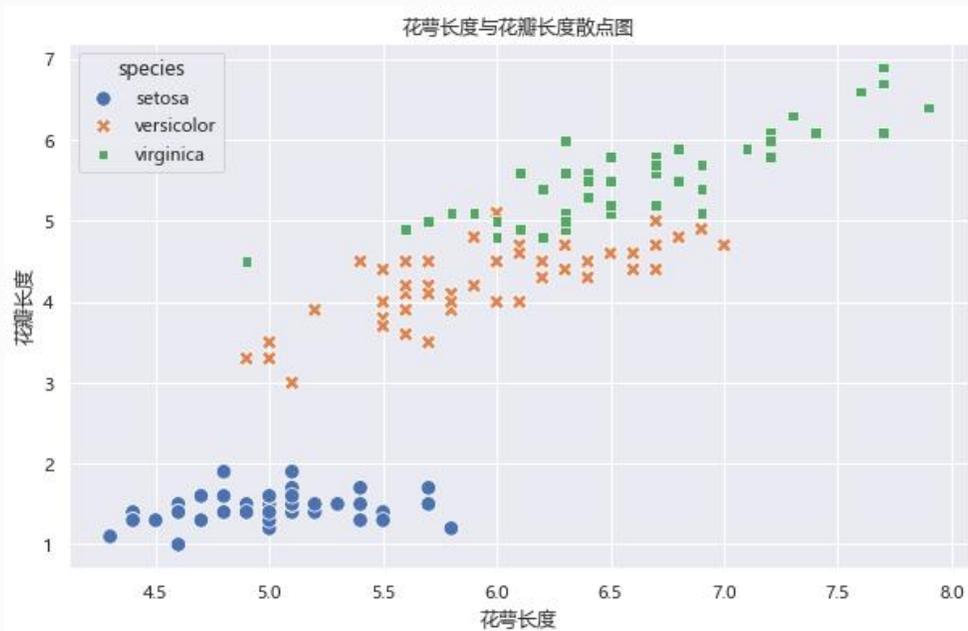


# 绘制散点图

## 散点图案例实战

### 案例：鸢尾花数据

案例描述：展示花萼长度与花瓣长度之间的关系，并通过颜色和符号区分不同种类的花。





## 绘制散点图

### 散点图案例实战

```
import seaborn as sns
import matplotlib.pyplot as plt
iris = sns.load_dataset("iris")
sns.scatterplot(x='sepal_length', y='petal_length', hue='
species', style='species', s=90, data=iris)
plt.title("花萼长度与花瓣长度散点图")
plt.xlabel("花萼长度")
plt.ylabel("花瓣长度")
plt.show()
```



# 总结

-  Seaborn简介
-  绘制计数图: `seaborn.countplot`
-  绘制箱线图: `seaborn.boxplot`
-  绘制柱形图: `seaborn.barplot`
-  绘制回归关系图: `seaborn.regplot`
-  绘制小提琴图: `seaborn.violinplot`
-  绘制折线图: `seaborn.lineplot`
-  绘制直方图: `seaborn.histplot`
-  绘制散点图: `seaborn.scatterplot`