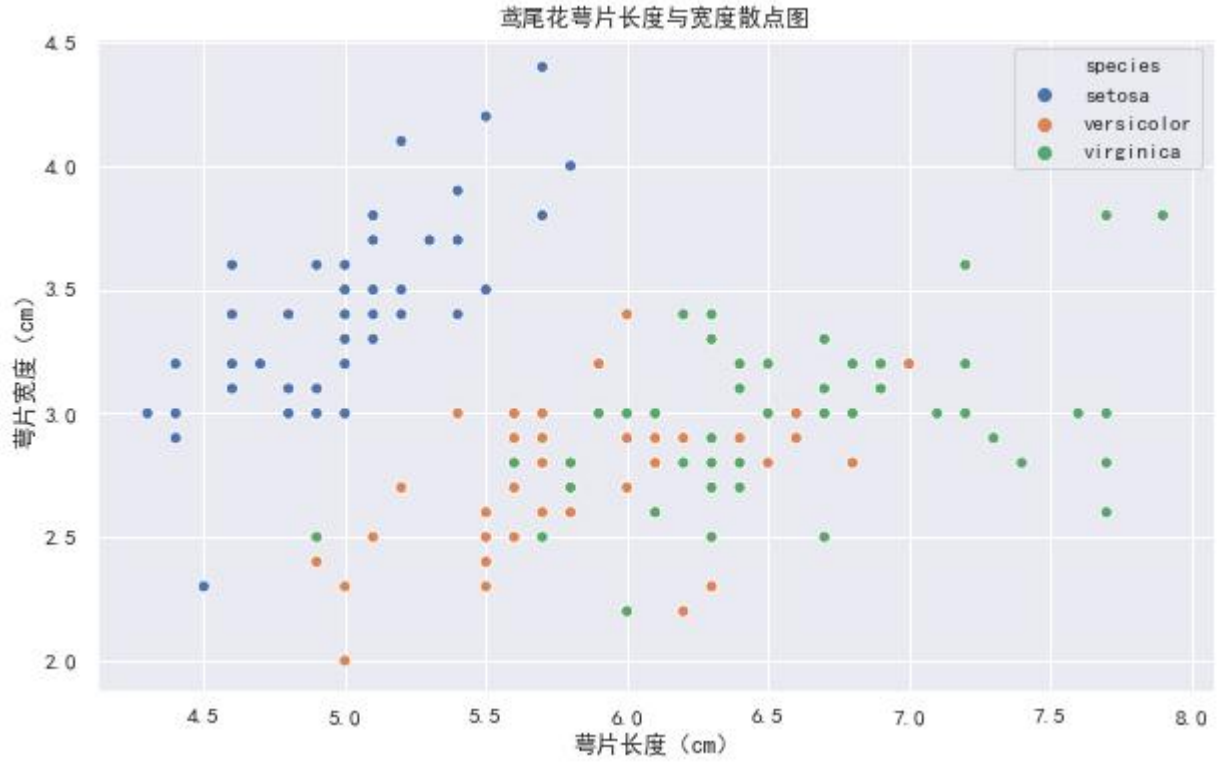
Seaborn 安装与配置¶

Seaborn 创建一个散点图¶

In [1]:

```
import seaborn as snsimport matplotlib.pyplot as pltsns.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()
```

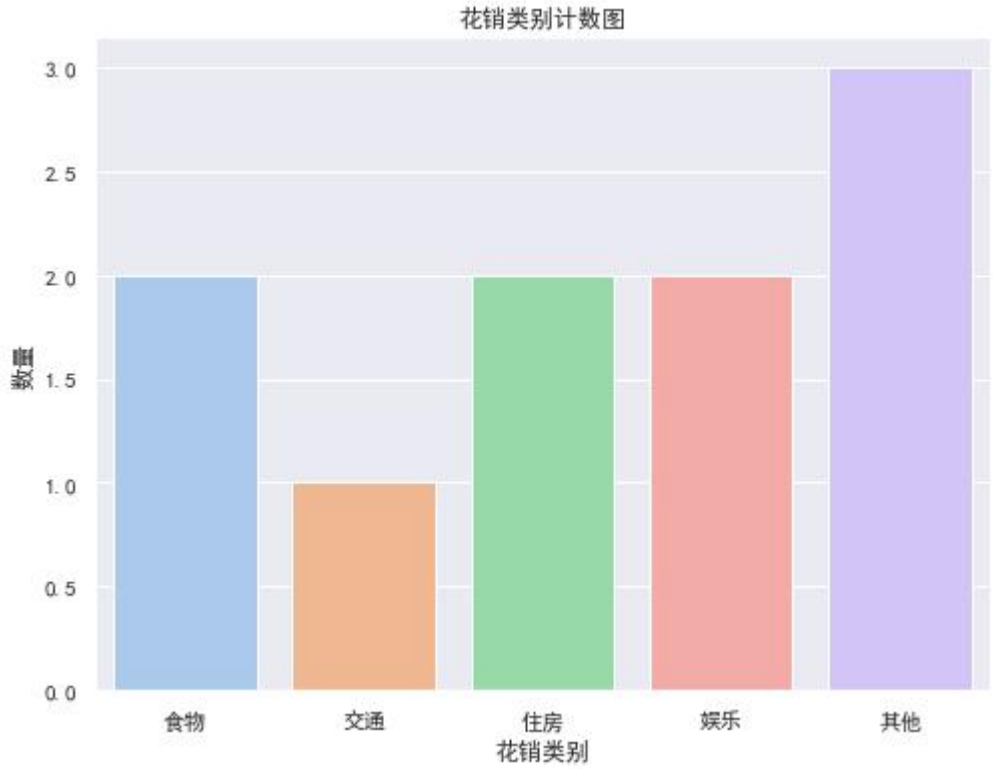


3.1.1 绘制计数图¶

示例 1¶

In [8]:

```
import seaborn as snsimport pandas as pdimport matplotlib.pyplot as pltplt.rcParams['font.sans-serif'] = ['SimHei']
# 解决负号显示问题plt.rcParams['axes.unicode_minus'] = Falsecategories = ['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()
```



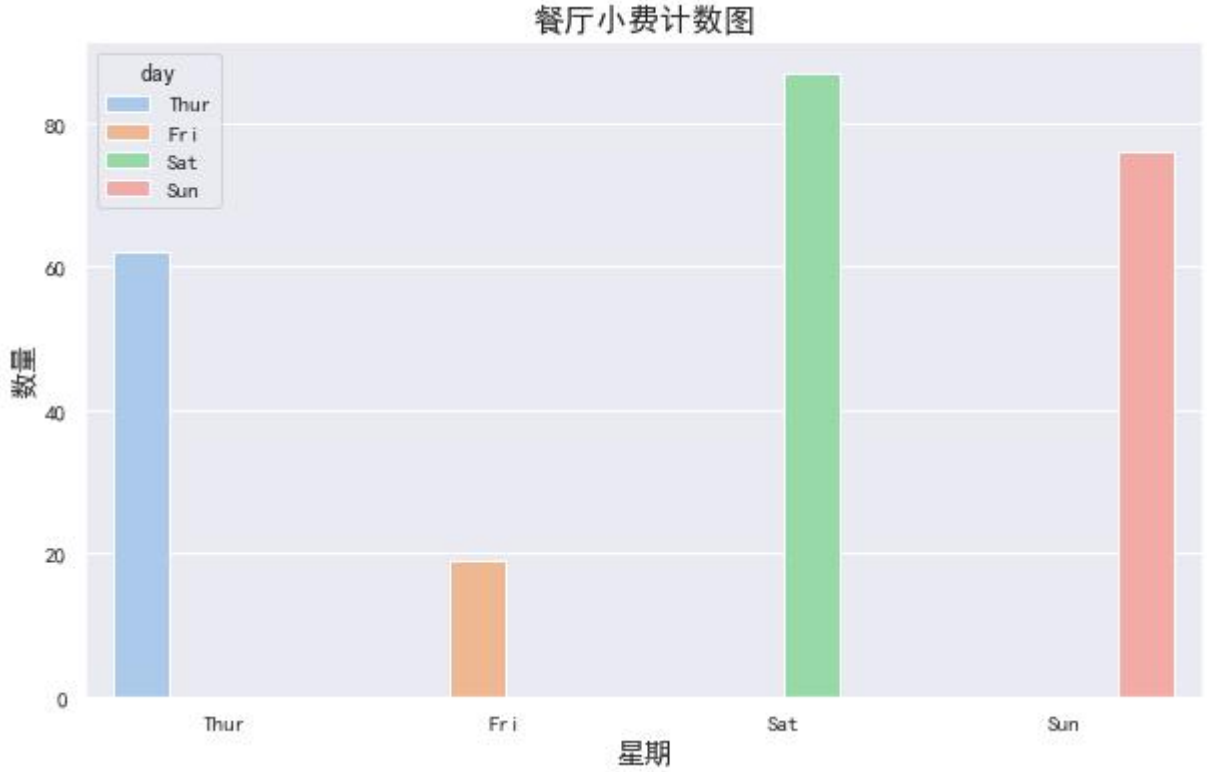
3.1.1 绘制计数图¶

示例 2¶

In [3]:

```
import seaborn as snsimport matplotlib.pyplot as pltplt.rcParams['font.sans-serif'] = 'SimHei'data = sns.load_datas
et('tips')plt.figure(figsize=(10, 6))sns.countplot(x='day', hue='day', data=data, palette='pastel', order=['Thur', '
```

```
Fri', 'Sat', 'Sun'])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()
```

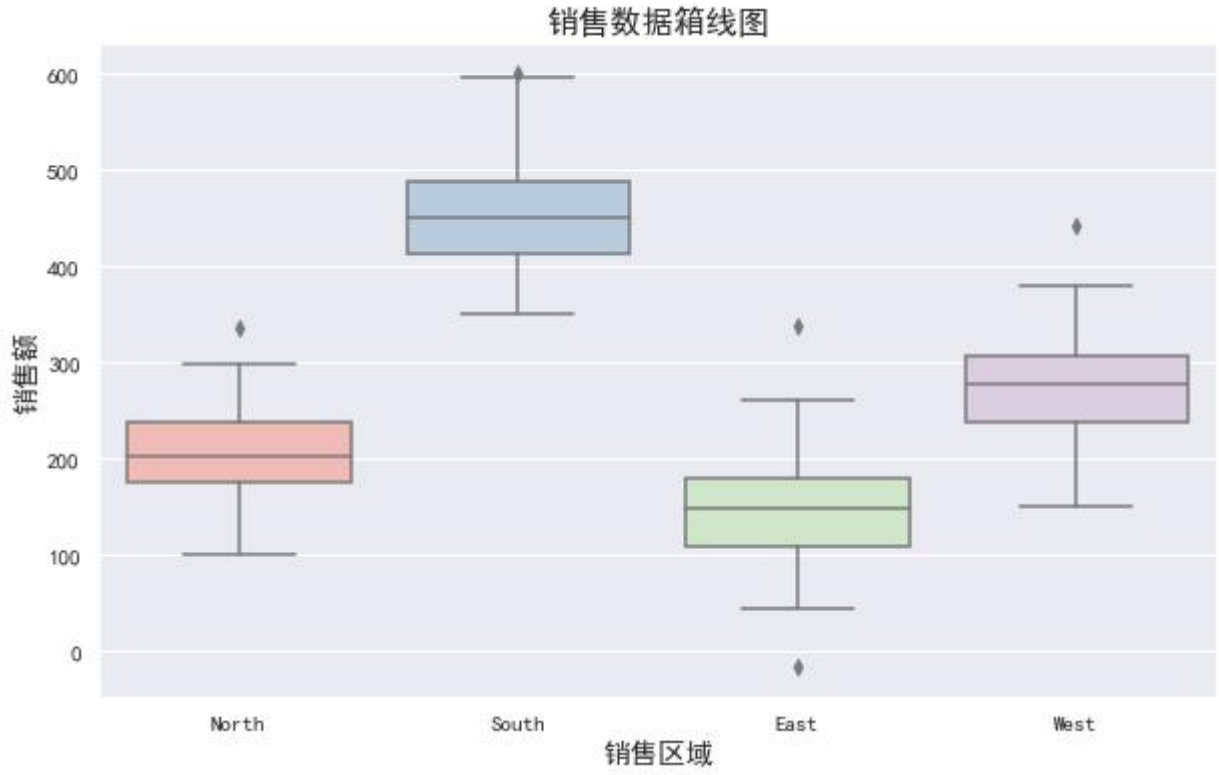


3.1.2 绘制箱线图¶

示例 1¶

In [4]:

```
import seaborn as snsimport matplotlib.pyplot as pltimport pandas as pdimport numpy as npnp.random.seed(42)regions = ['North', 'South', 'East', 'West']sales_data = {region: np.random.normal(np.random.randint(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()
```

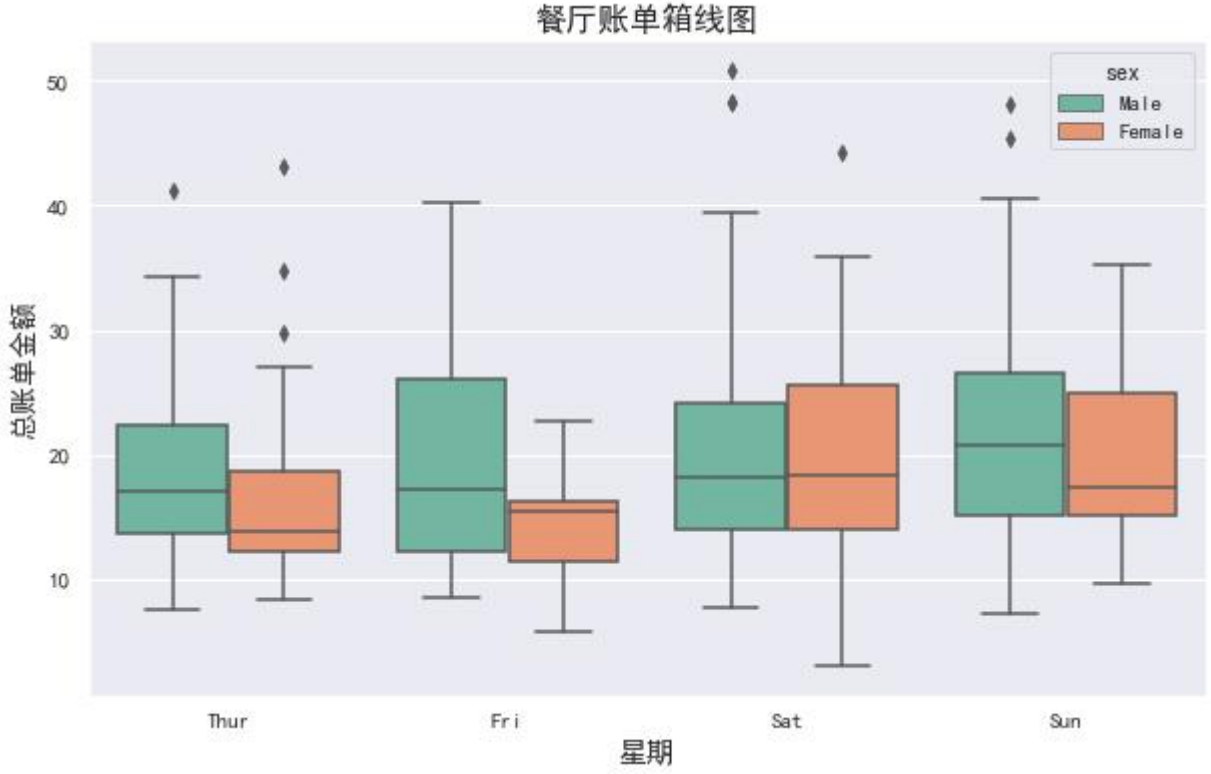


3.1.2 绘制箱线图¶

示例 2¶

In [5]:

```
import seaborn as sns import matplotlib.pyplot as pltdata = sns.load_dataset("tips")plt.figure(figsize=(10, 6))sns.boxplot(x="day", y="total_bill", data=data, hue="sex", palette="Set2")plt.title("餐厅账单箱线图", fontsize=16)plt.xlabel("星期", fontsize=14)plt.ylabel("总账单金额", fontsize=14)plt.show()
```

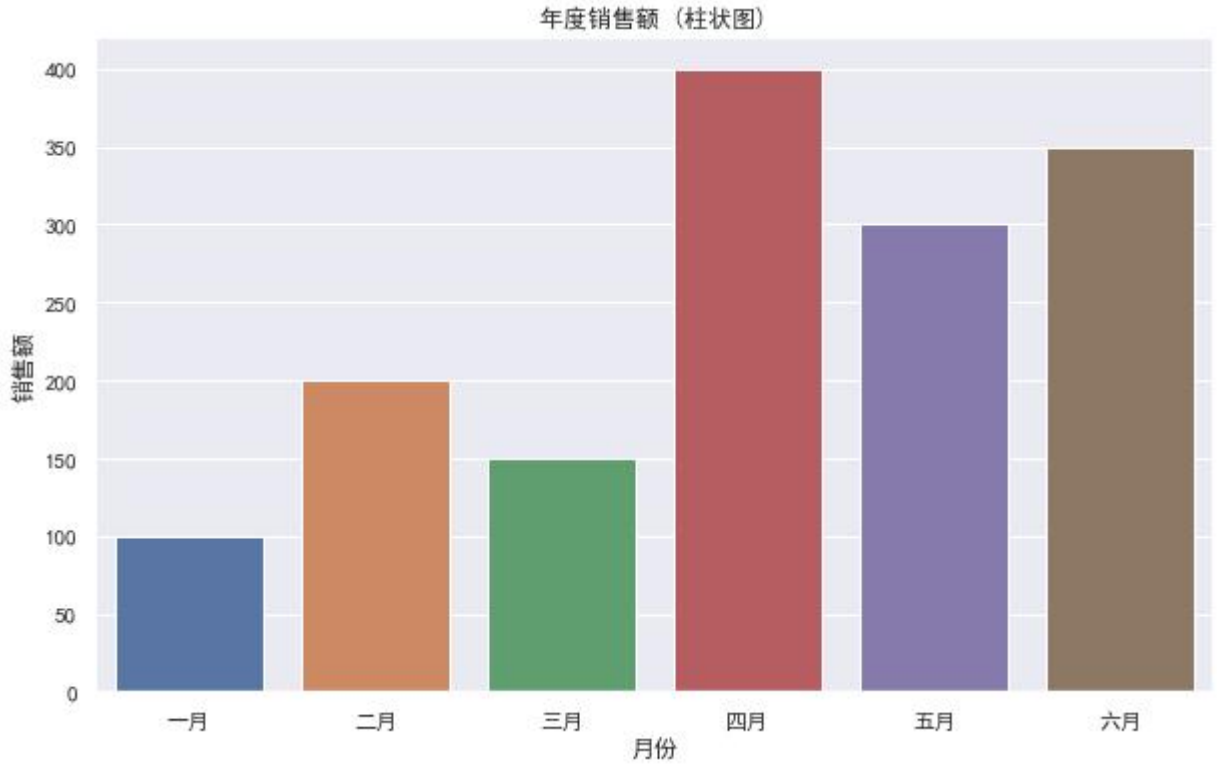


3.1.3 绘制柱形图¶

示例 1：¶

In [6]:

```
import seabornimport matplotlib.pyplot as pltmonths = ['一月', '二月', '三月', '四月', '五月', '六月']sales = [100, 200, 150, 400, 300, 350]plt.xlabel('月份')plt.ylabel('销售额')plt.title('年度销售额（柱状图）')seaborn.barplot(x=months, y=sales)  # 使用 seaborn 配置柱状图的 X 轴和 Y 轴plt.show()
```

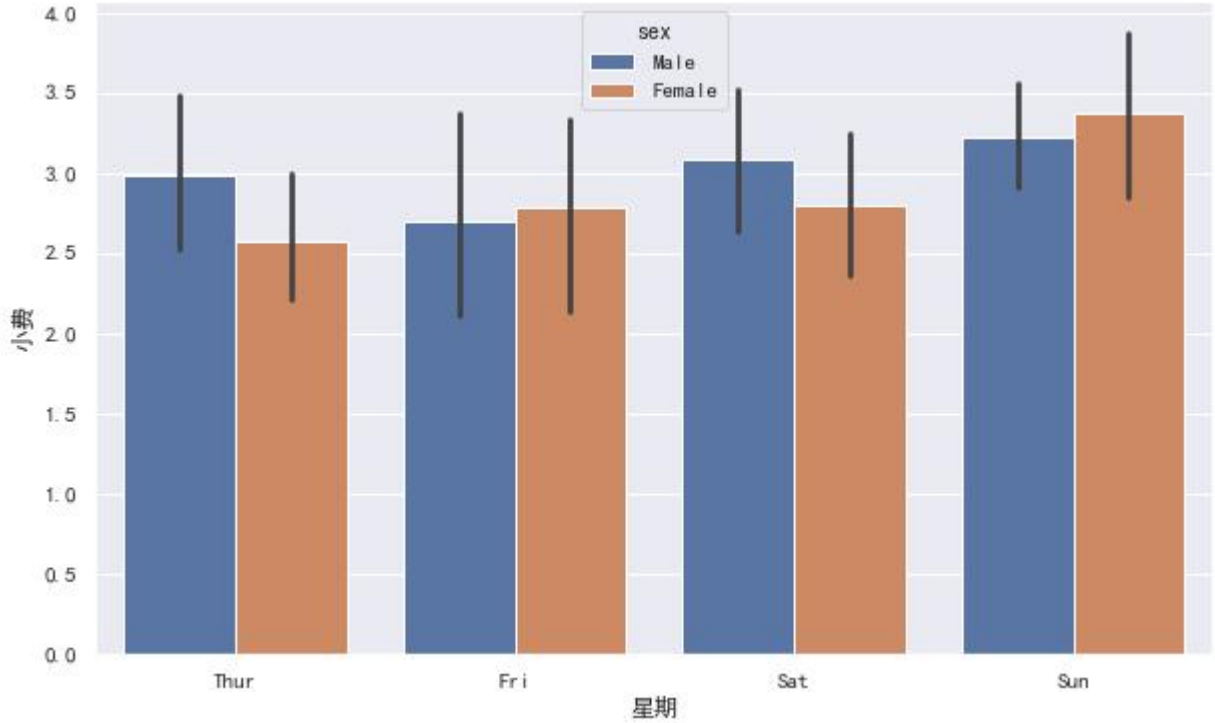


3.1.3 绘制柱形图¶

示例 2：¶

In [7]:

```
import matplotlib.pyplot as pltimport seaborn as snsdata = 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SimHeisns.barplot(x='day', y='tip', hue='sex', data=data)plt.xlabel("星期")plt.ylabel("小费")plt.show()
```

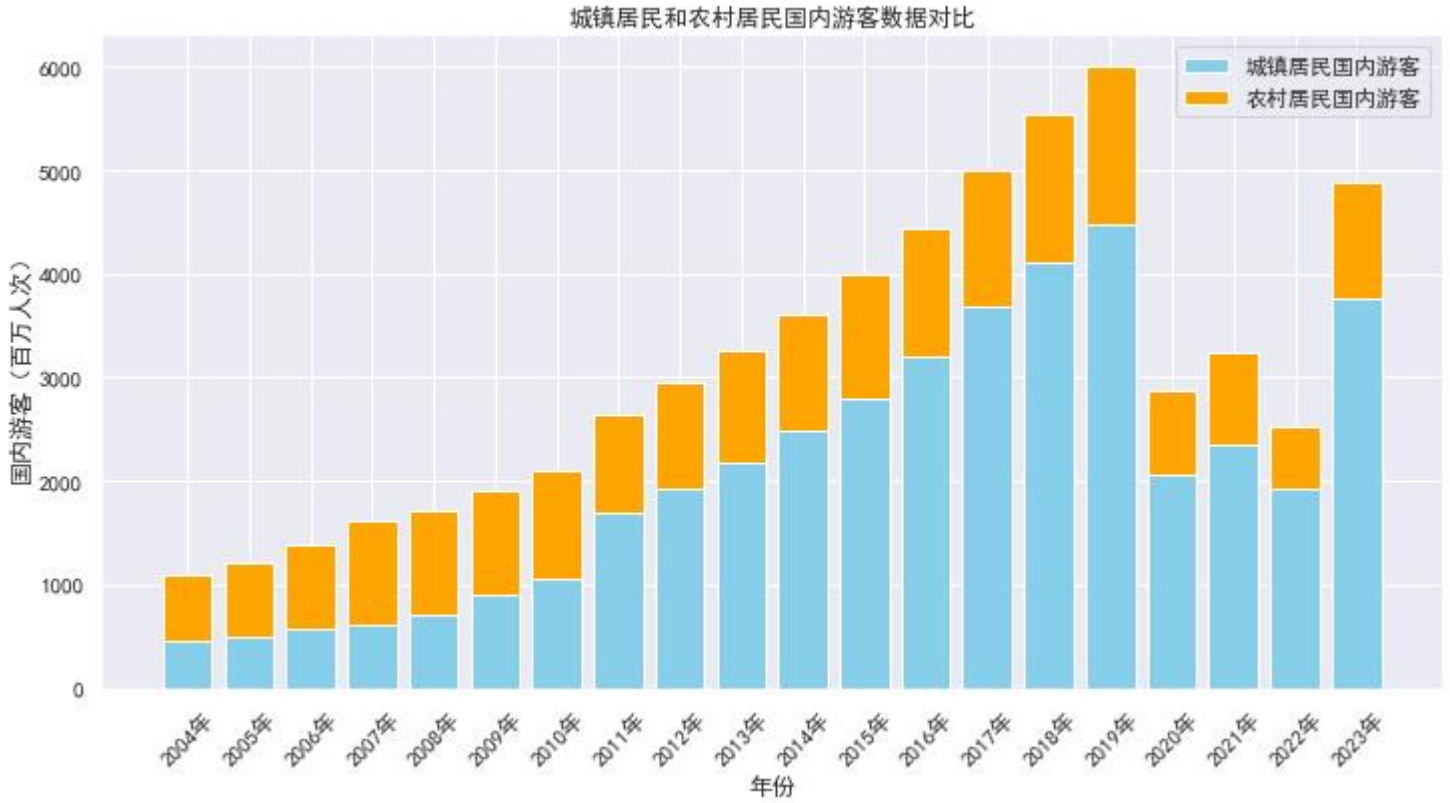


3.1.3 绘制柱形图¶

任务案例：¶

In [9]:

```
import pandas as pdimport matplotlib.pyplot as plt
# 修正函数名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.ilo
c[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()
```



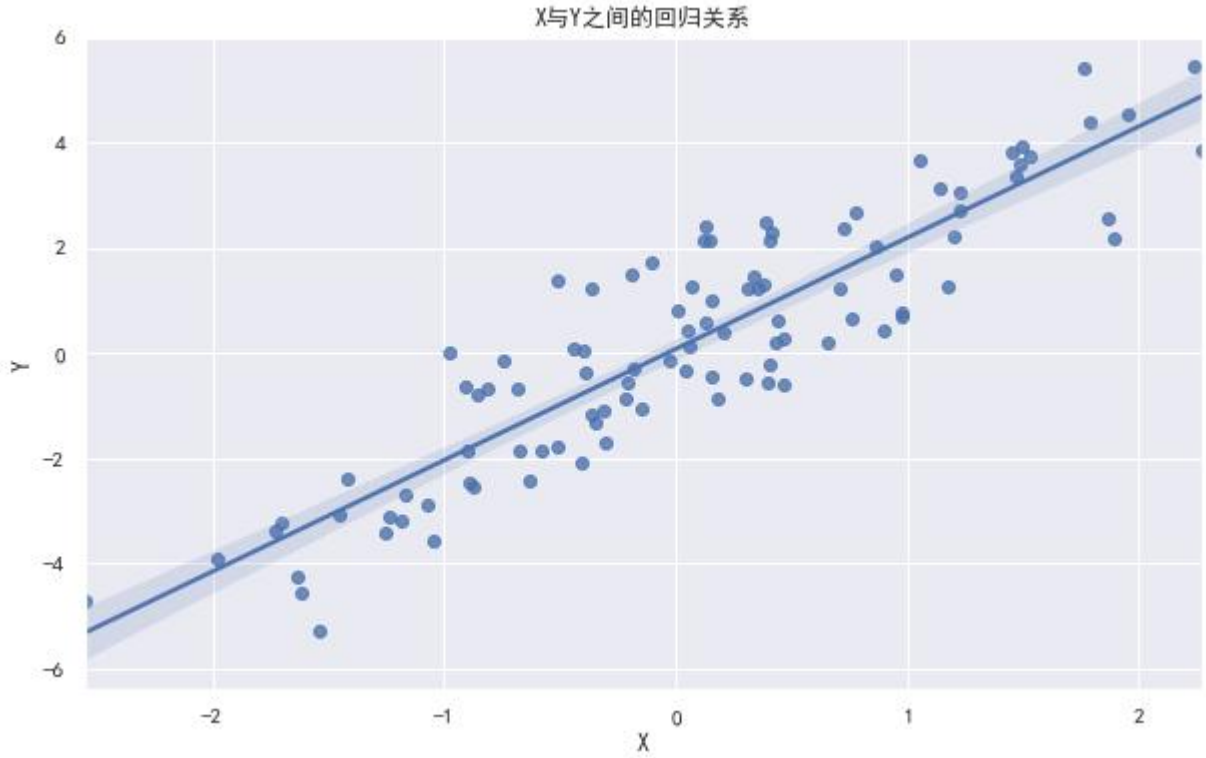
任务 3-2 外国入境游客人次按年龄展示¶

示例 1¶

In [10]:

```
import seaborn as snsimport matplotlib.pyplot as pltimport pandas as pdimport numpy as np# 设置 matplotlib 支持中文显示plt.rcParams['font.sans-serif'] = ['SimHei']  # 使用黑体字体，可根据系统情况替换为其他中文字体，如 'Microsoft YaHei'# 解决负号显示问题plt.rcParams['axes.unicode_minus'] = Falsenp.random.seed(0)x = np.random.randn(100)y = 2 * x
```

```
 + np.random.randn(100)data = pd.DataFrame({'X': x, 'Y': y})plt.rcParams['font.sans-serif'] = ['SimHei']  # 设置中文字体为黑体plt.rcParams['axes.unicode_minus'] = False  # 解决负号显示问题sns.regplot(x='X', y='Y', data=data)plt.title('X 与 Y 之间的回归关系')plt.xlabel('X')plt.ylabel('Y')plt.show()
```

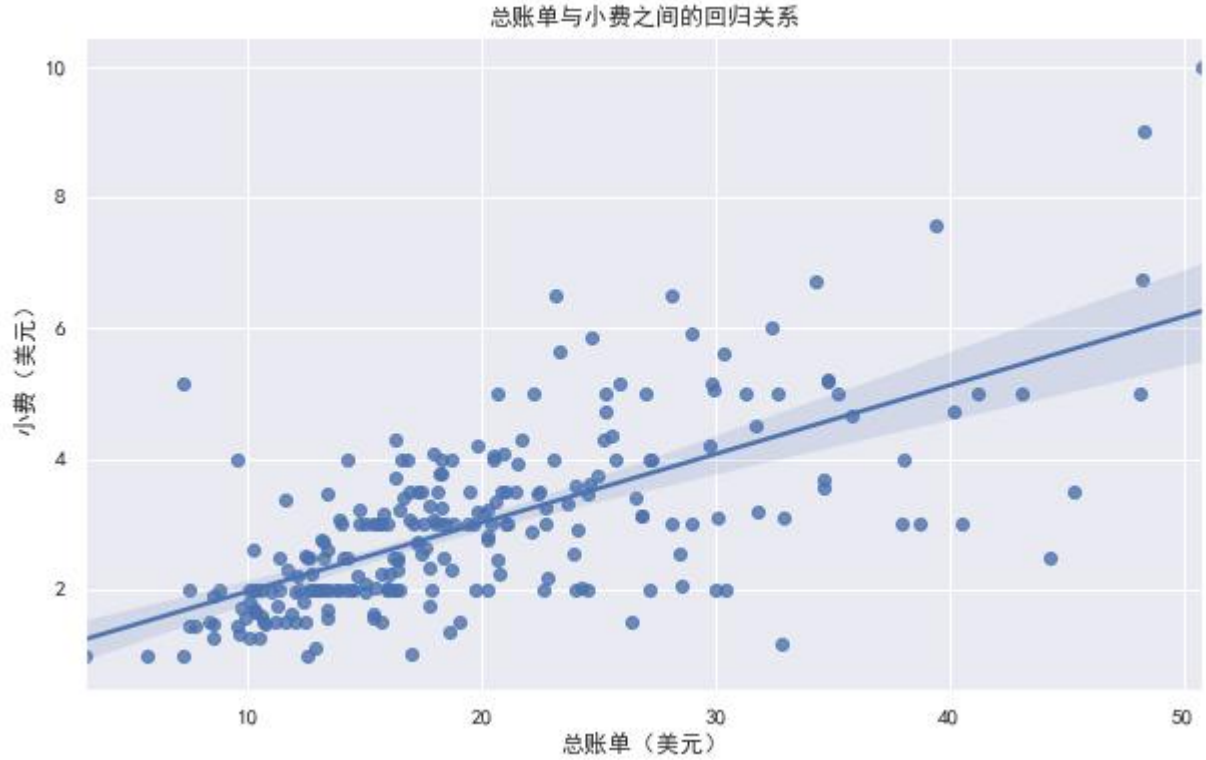


任务 3-2 外国入境游客人次按年龄展示¶

示例 2¶

In [11]:

```
import seaborn as snsimport matplotlib.pyplot as pltips = 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SimHeisns.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()
```

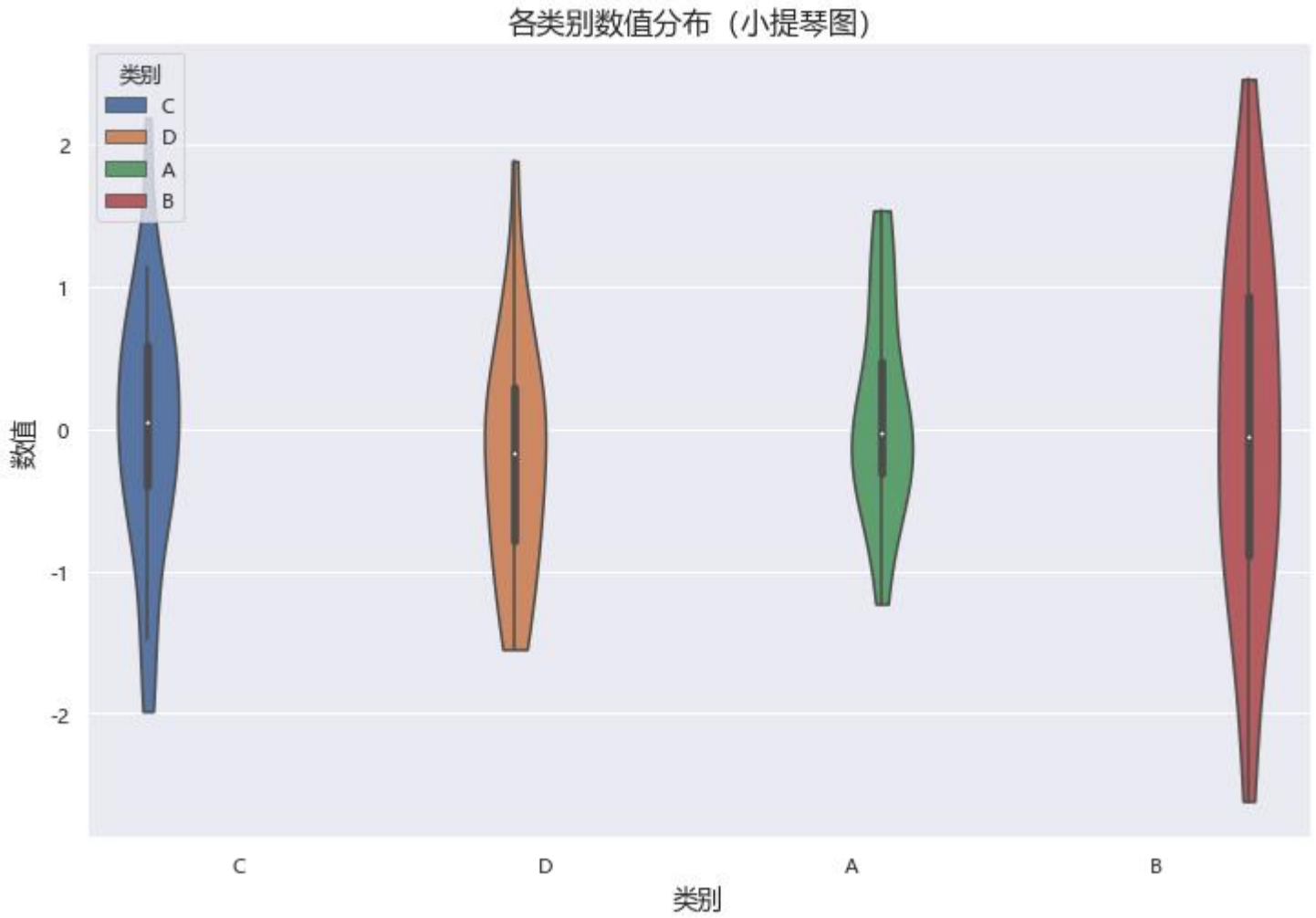


3.2.2 绘制小提琴图¶

示例 1¶

In [12]:

```
import seaborn as snsimport matplotlib.pyplot as pltimport pandas as pdimport numpy as npnp.random.seed(42)plt.rcParams['font.sans-serif'] = ['simhei']plt.rcParams['font.sans-serif'] = ['Microsoft YaHei']data = {
    '类别': np.random.choice(['A', 'B', 'C', 'D'], size=100),
    '数值': np.random.normal(loc=0, scale=1, size=100)}df = pd.DataFrame(data)plt.figure(figsize=(12, 8))sns.violinplot(x='类别', y='数值', data=df, bw_method=0.2, cut=0, hue='类别', legend=False)plt.title('各类别数值分布（小提琴图）', fontsize=16)plt.xlabel('类别', fontsize=14)plt.ylabel('数值', fontsize=14)plt.show()
```

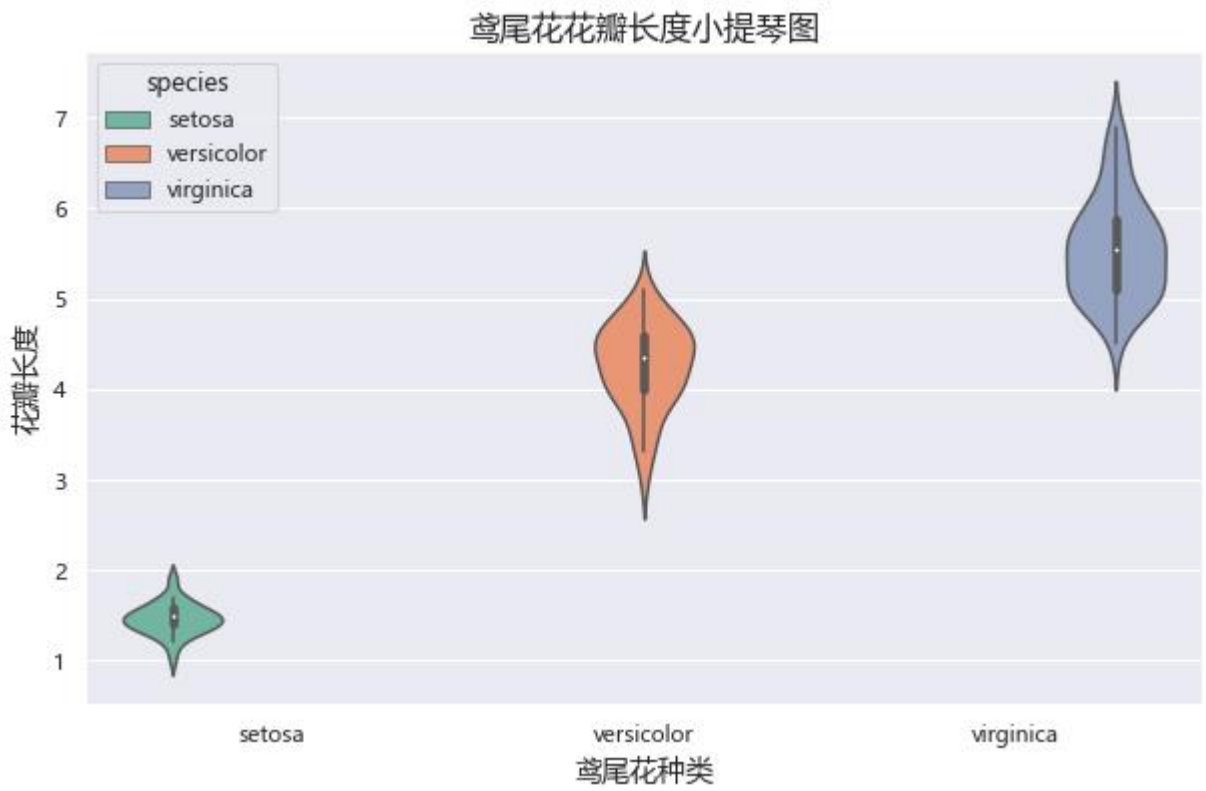


3.2.2 绘制小提琴图¶

示例 2¶

In [13]:

```
import seaborn as snsimport matplotlib.pyplot as pltiris_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()
```

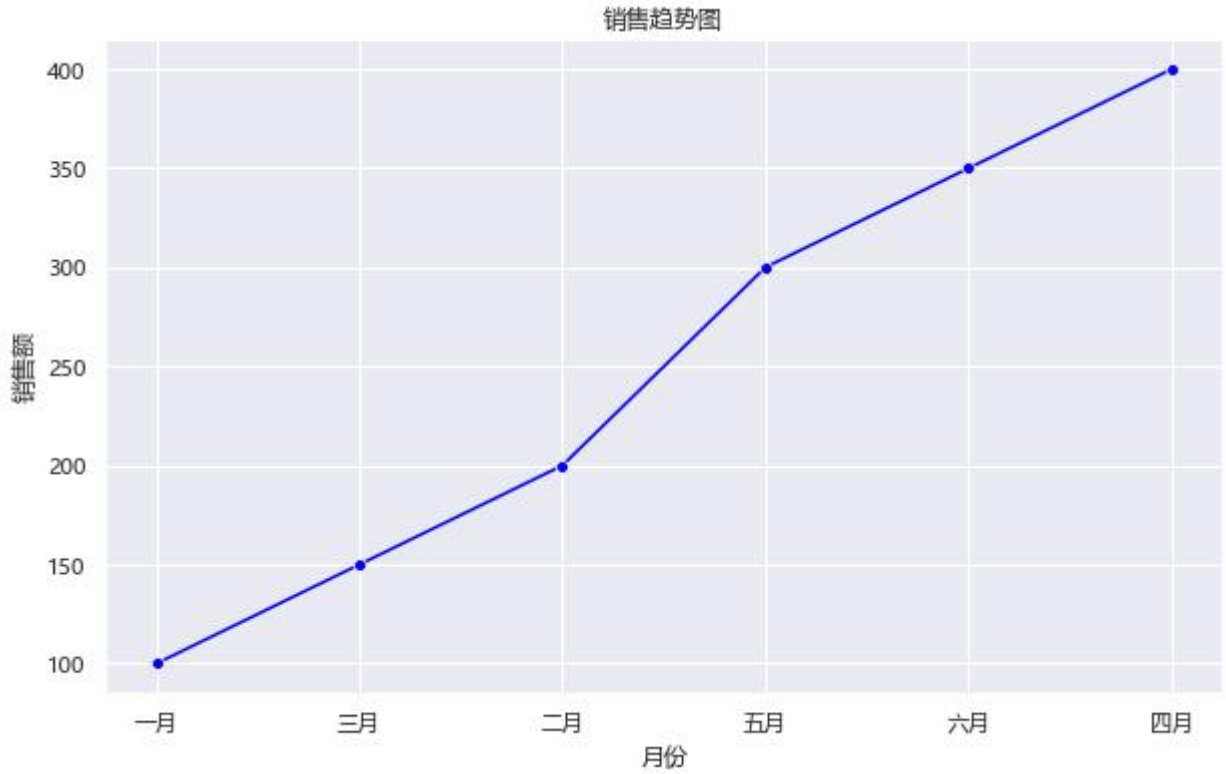


3.2.3 绘制折线图¶

示例 1¶

In [14]:

```
import seaborn as snsimport matplotlib.pyplot as pltmonths = ['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()
```

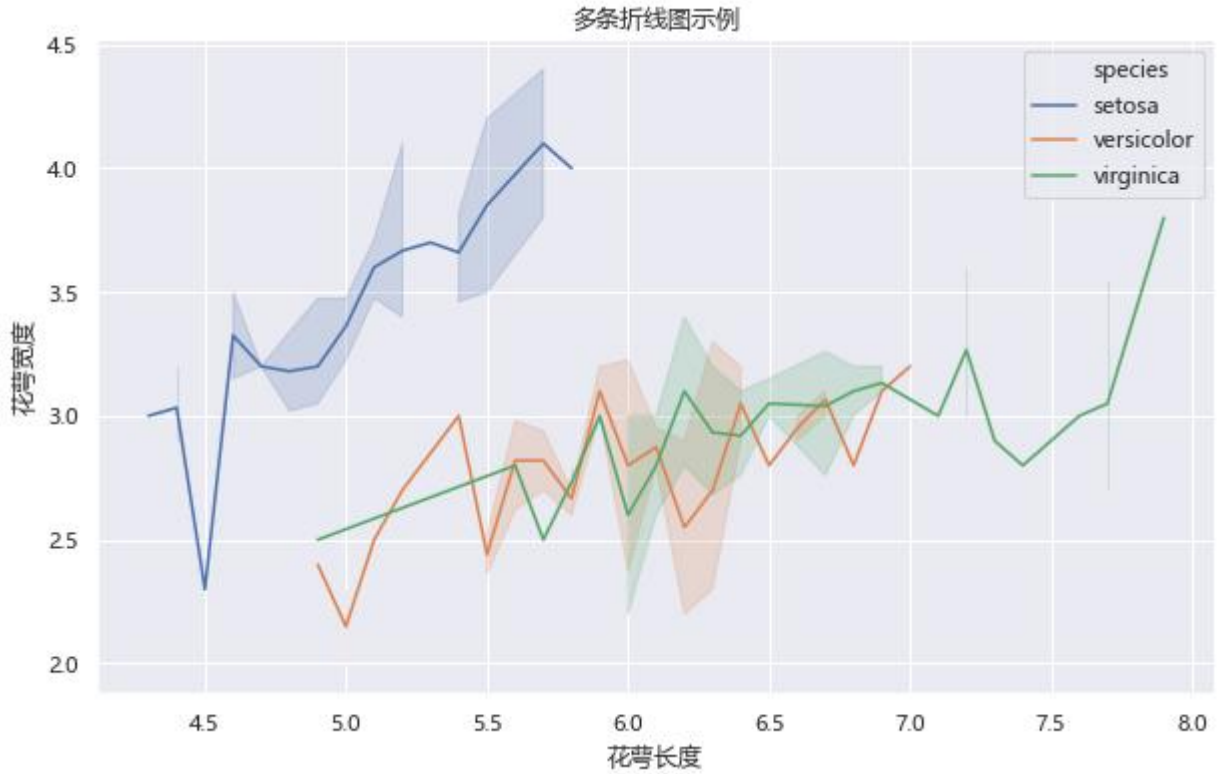


3.2.3 绘制折线图¶

示例 1¶

In [15]:

```
import seaborn as snsimport matplotlib.pyplot as pltdata = 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")plt.ylabel("花萼宽度")plt.show()
```



3.2.3 绘制折线图¶

任务案例¶

In [29]:

```
import pandas as pdimport matplotlib.pyplot as pltfrom 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()
```

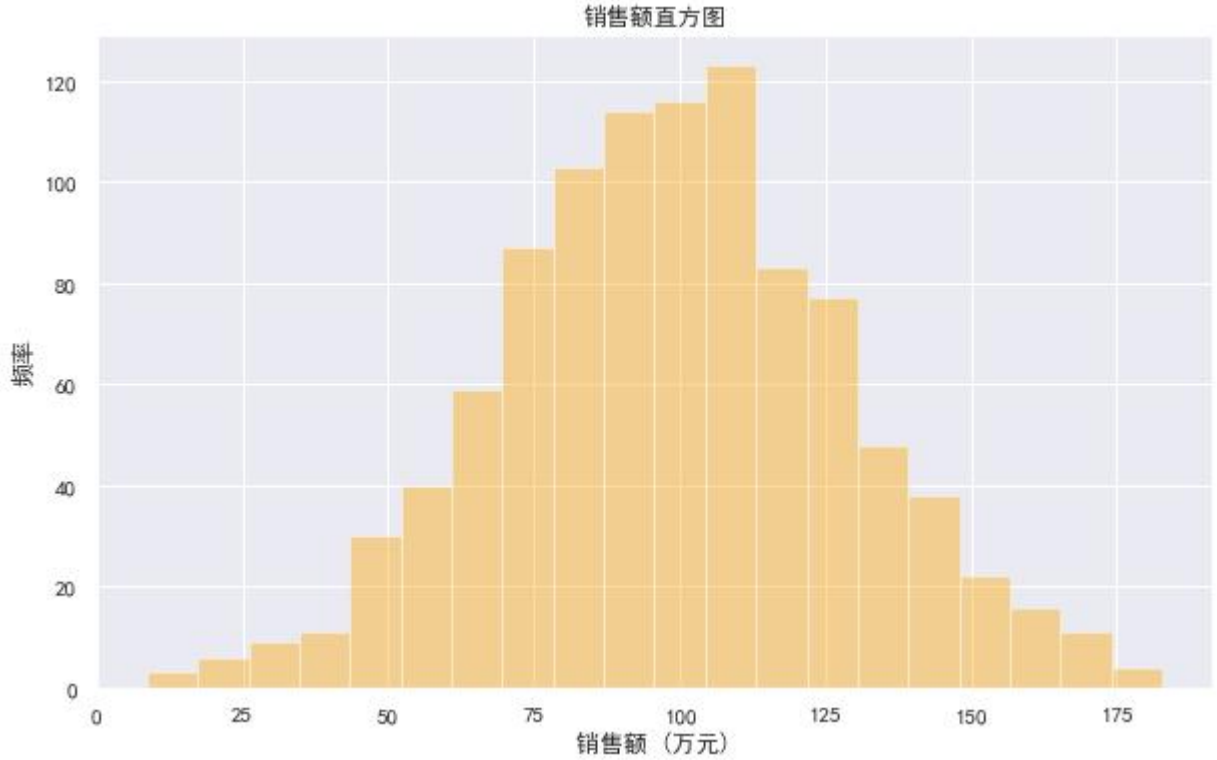
3.3.1 直方图¶

示例 1¶

In [32]:

```
import seaborn as snsimport matplotlib.pyplot as pltimport numpy as np
```

```
np.random.seed(0)sales = np.random.normal(loc=100, scale=30, size=1000)plt.rcParams['font.sans-serif'] = ['SimHei']sns.histplot(data=sales, bins=20, kde=False, color='orange')plt.title('销售额直方图')plt.xlabel('销售额（万元）')plt.ylabel('频率')plt.show()
```

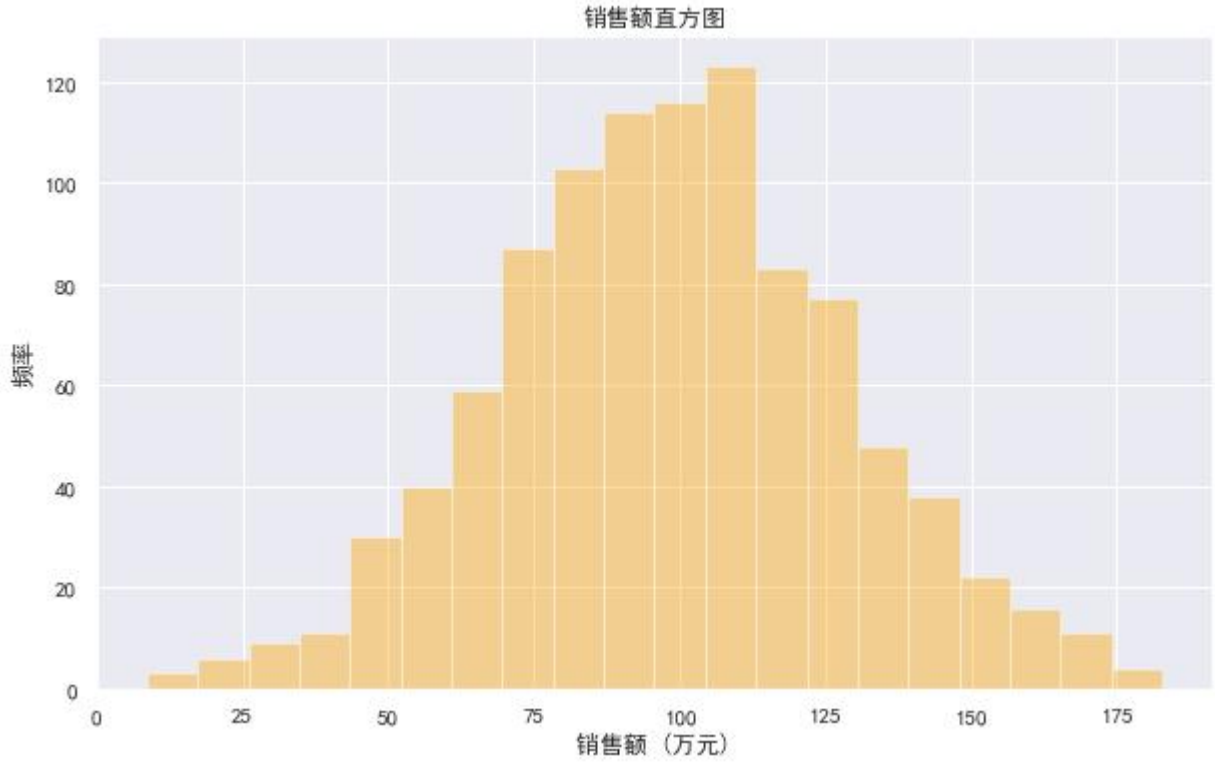


3.3.1 直方图¶

示例 1¶

In [32]:

```
import seaborn as snsimport matplotlib.pyplot as pltimport numpy as np
np.random.seed(0)sales = np.random.normal(loc=100, scale=30, size=1000)plt.rcParams['font.sans-serif'] = ['SimHei']s
ns.histplot(data=sales, bins=20, kde=False, color='orange')plt.title('销售额直方图')plt.xlabel('销售额 (万元)')plt.y
label('频率')plt.show()
```

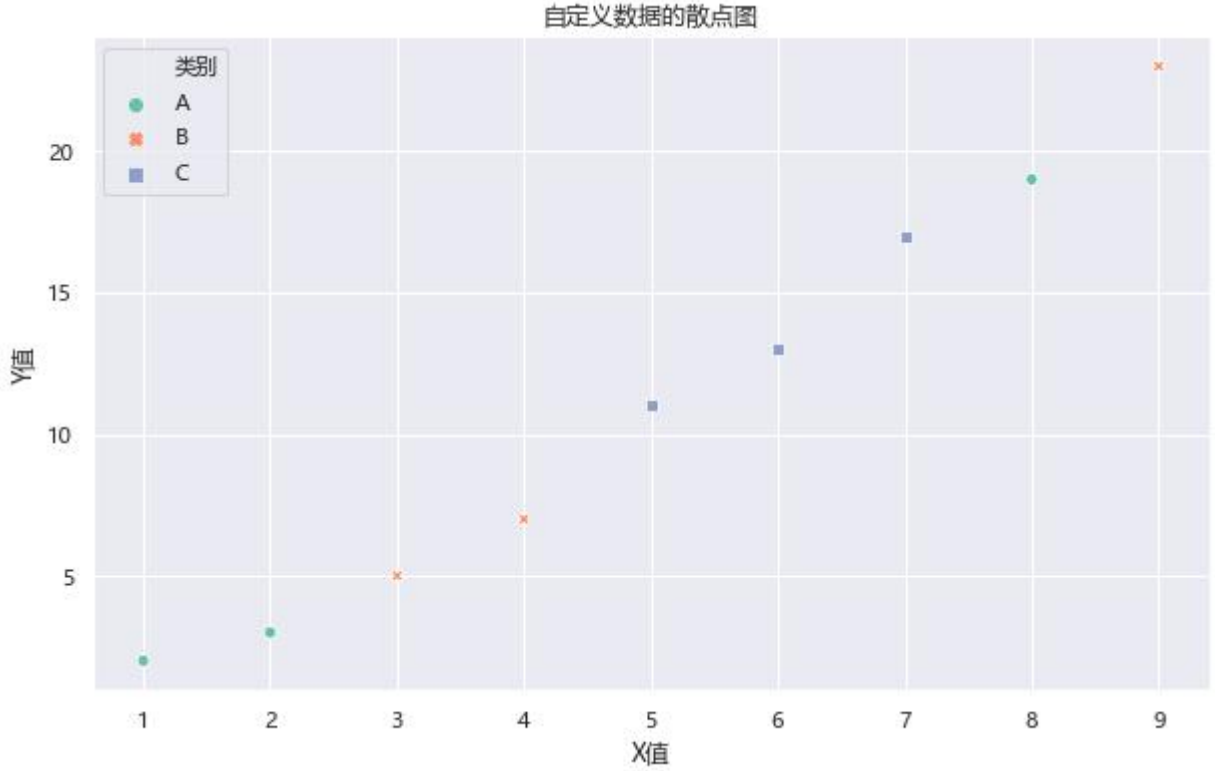


3.3.2 散点图¶

示例 1：¶

In [17]:

```
import pandas as pdimport seaborn as snsimport matplotlib.pyplot as pltdata = {
    'X值': [1, 2, 3, 4, 5, 6, 7, 8, 9],
    'Y值': [2, 3, 5, 7, 11, 13, 17, 19, 23],
    '类别': ['A', 'A', 'B', 'B', 'C', 'C', 'C', 'A', 'B']}df = pd.DataFrame(data)sns.scatterplot(x='X值', y='Y值', data=df, hue='类别', style='类别', palette='Set2')plt.title("自定义数据的散点图")plt.xlabel("X值")plt.ylabel("Y值")plt.show()
```

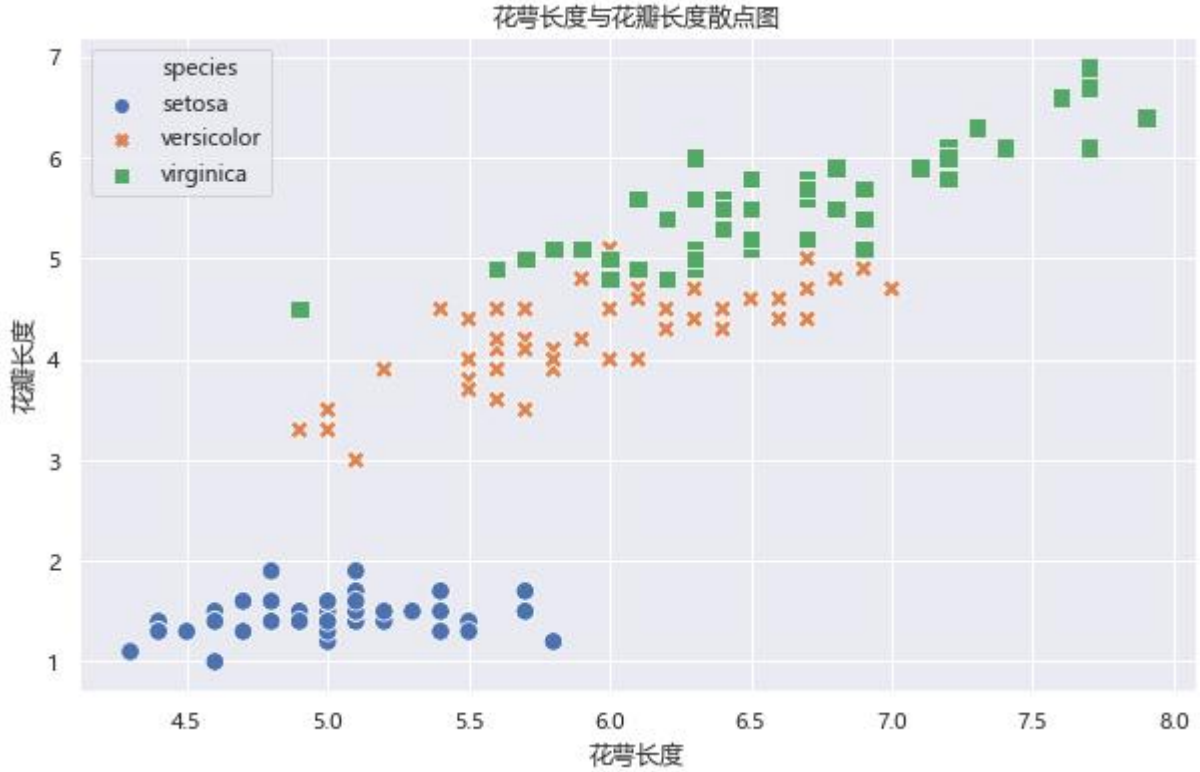


3.3.2 散点图¶

示例 2：¶

In [18]:

```
import seaborn as snsimport matplotlib.pyplot as pltiris = 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()
```

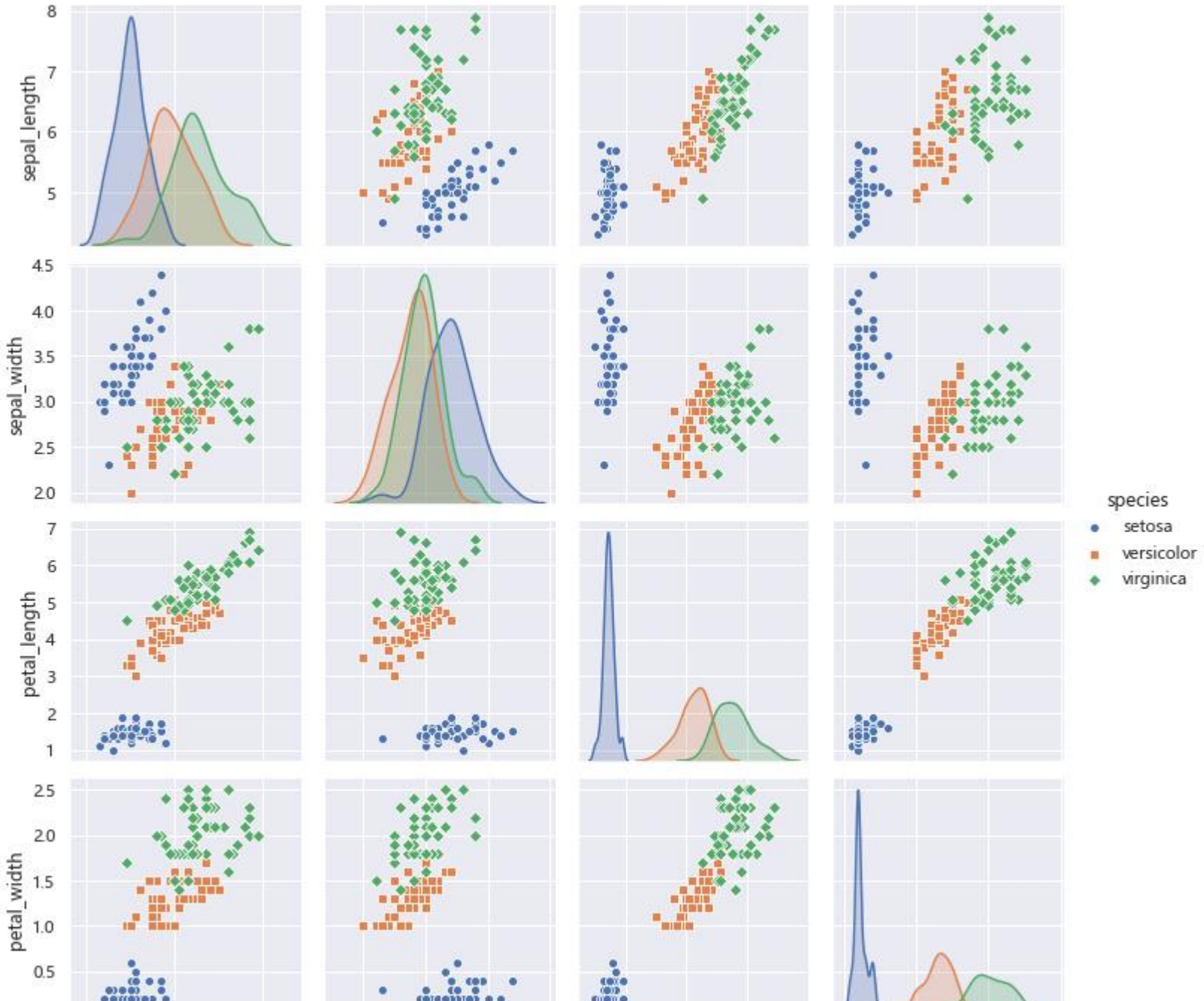


pairplot()函数¶

图 3-23 鸢尾花数据集散点图矩阵¶

In [19]:

```
import seaborn as snsimport matplotlib.pyplot as pltfrom sklearn.datasets import load_iris# 加载鸢尾花数据集iris = load_iris()iris_df = sns.load_dataset('iris')# 创建散点图矩阵，使用散点图表示变量之间的关系，并在对角线上使用直方图sns.pairplot(iris_df, hue="species", diag_kind="kde", markers=["o", "s", "D"])plt.show()
```

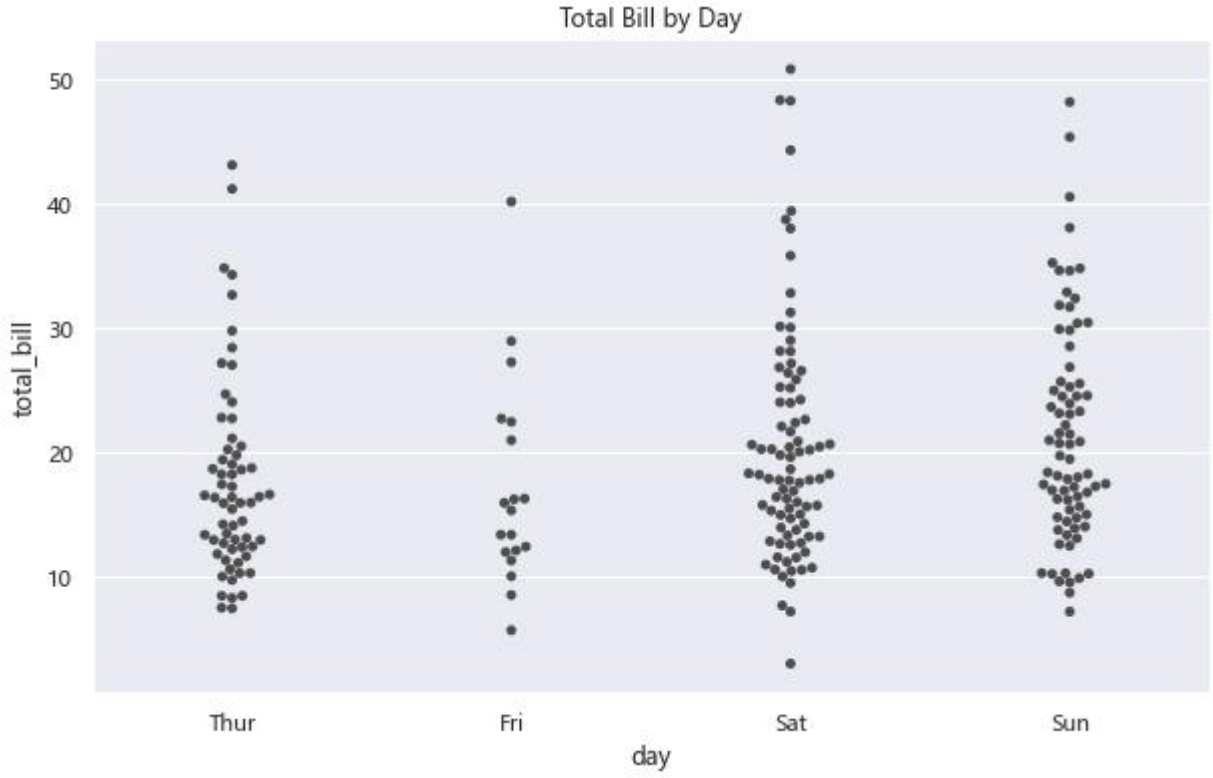

sepal_length
sepal_width
petal_length
petal_width
8
7
6
5
4.5
4.0
3.5
3.0
2.5
2.0
7
6
5
4
3
2
1
2.5
2.0
1.5
1.0
0.5
species
setosa
versicolor
virginica

swarmplot 函数¶

图 3-24 餐厅在不同天总账单分布情况¶

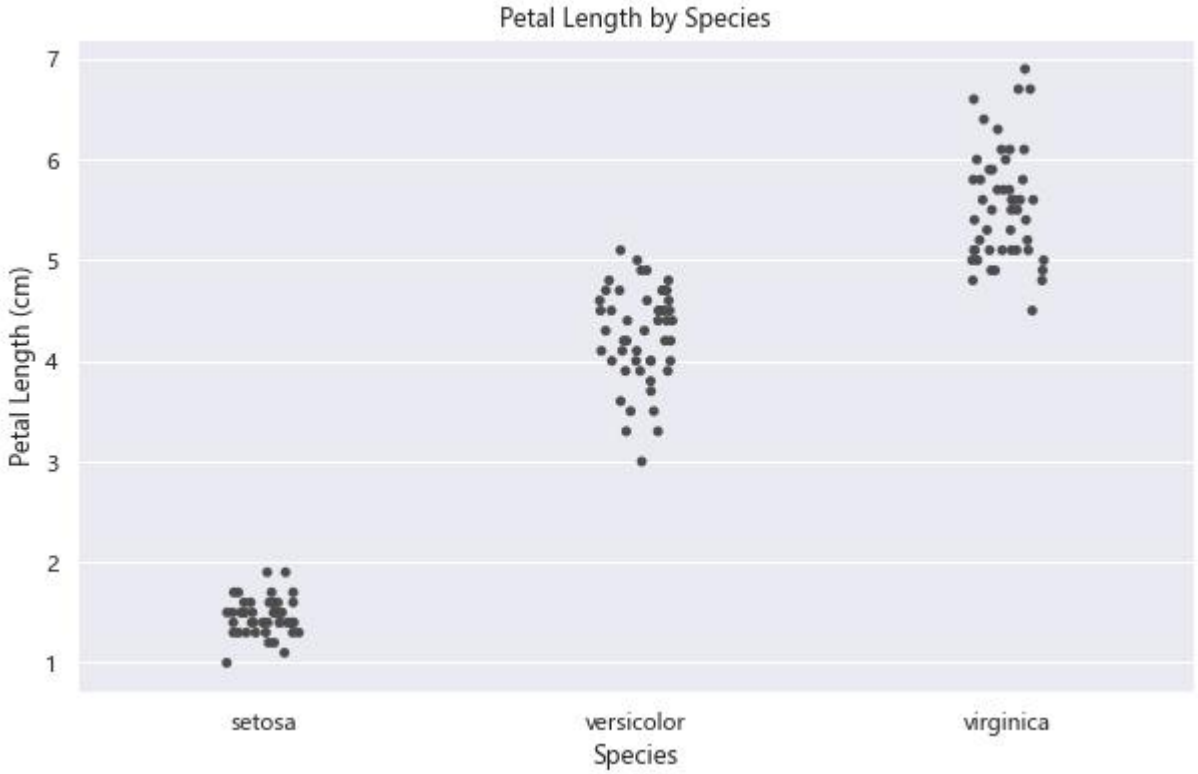
In [20]:

```
import seaborn as snsimport matplotlib.pyplot as plt# 加载示例数据集tips = sns.load_dataset('tips')# 创建散点图（Swarmplot），用于显示不同天的总账单分布sns.swarmplot(x="day", y="total_bill", data=tips, color=".3")# 添加标题plt.title("Total Bill by Day")# 显示图形plt.show()
```



stripplot()函数¶
示例：¶
图 3-25 不同种类鸢尾花的花瓣长度分布¶
In [21]:

```
import seaborn as snsimport matplotlib.pyplot as pltfrom sklearn.datasets import load_iris# 加载鸢尾花数据集iris = load_iris()iris_df = sns.load_dataset('iris')# 创建分类散点图，使用 stripplot 显示不同种类鸢尾花的花瓣长度分布sns.stripplot(data=iris_df, x="species", y="petal_length", jitter=True, color=".3")# 添加标题和标签plt.title("Petal Length by Species")plt.xlabel("Species")plt.ylabel("Petal Length (cm)")# 显示图形plt.show()
```



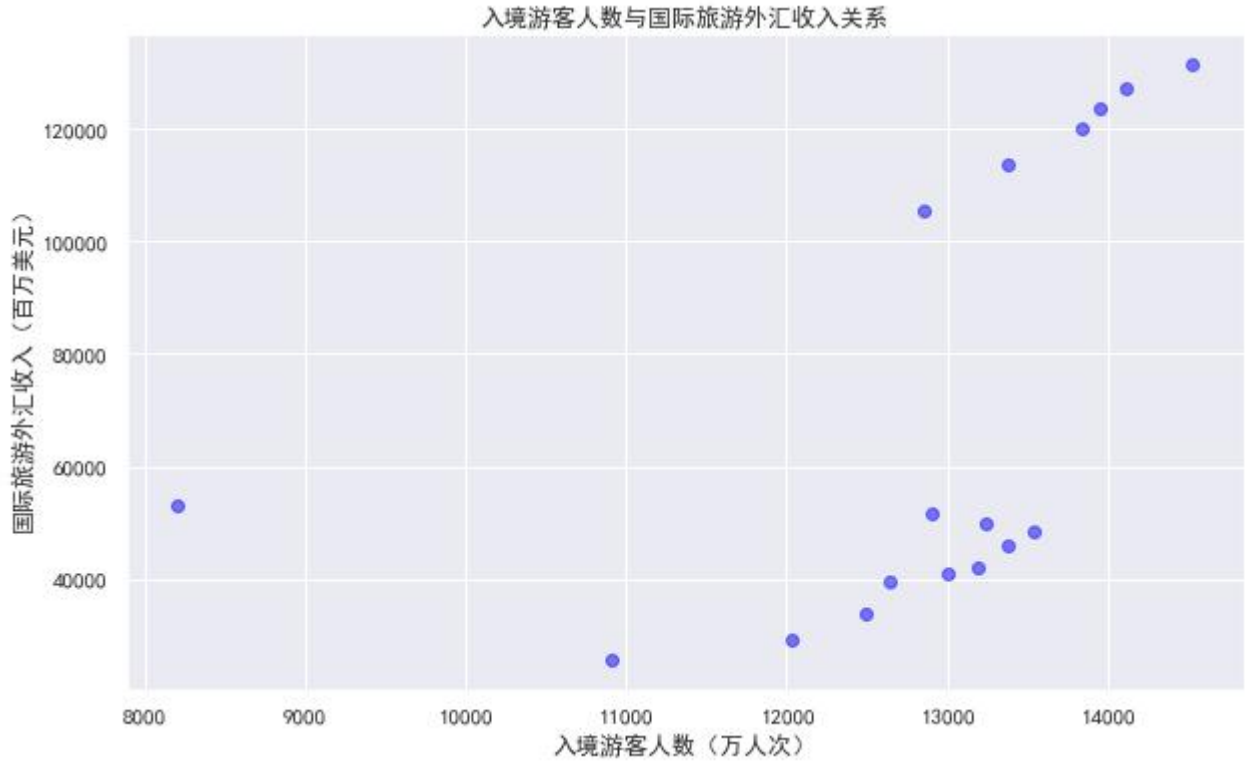
任务案例：¶

读取“data.xls”表中“旅游业发展情况”表单数据，并根据历年“入境游客人数（万人次）”、“国际旅游外汇收入（百万美元）”数据绘制散点图。¶

In [22]:

```
import pandas as pdimport matplotlib.pyplot as pltplt.rcParams['font.sans-serif'] = ['SimHei']  plt.rcParams['axes.unicode_minus'] = False data = pd.read_excel('data.xls', sheet_name='旅游业发展情况')tourist_arrivals = data.iloc[2, 1:]foreign_exchange_income = data.iloc[10, 1:]plt.figure(figsize=(10, 6))plt.scatter(tourist_arrivals, foreign_exch
```

```
ange_income, color='blue', alpha=0.5)plt.title('入境游客人数与国际旅游外汇收入关系')plt.xlabel('入境游客人数（万人次）')plt.ylabel('国际旅游外汇收入（百万美元）')plt.show()
```



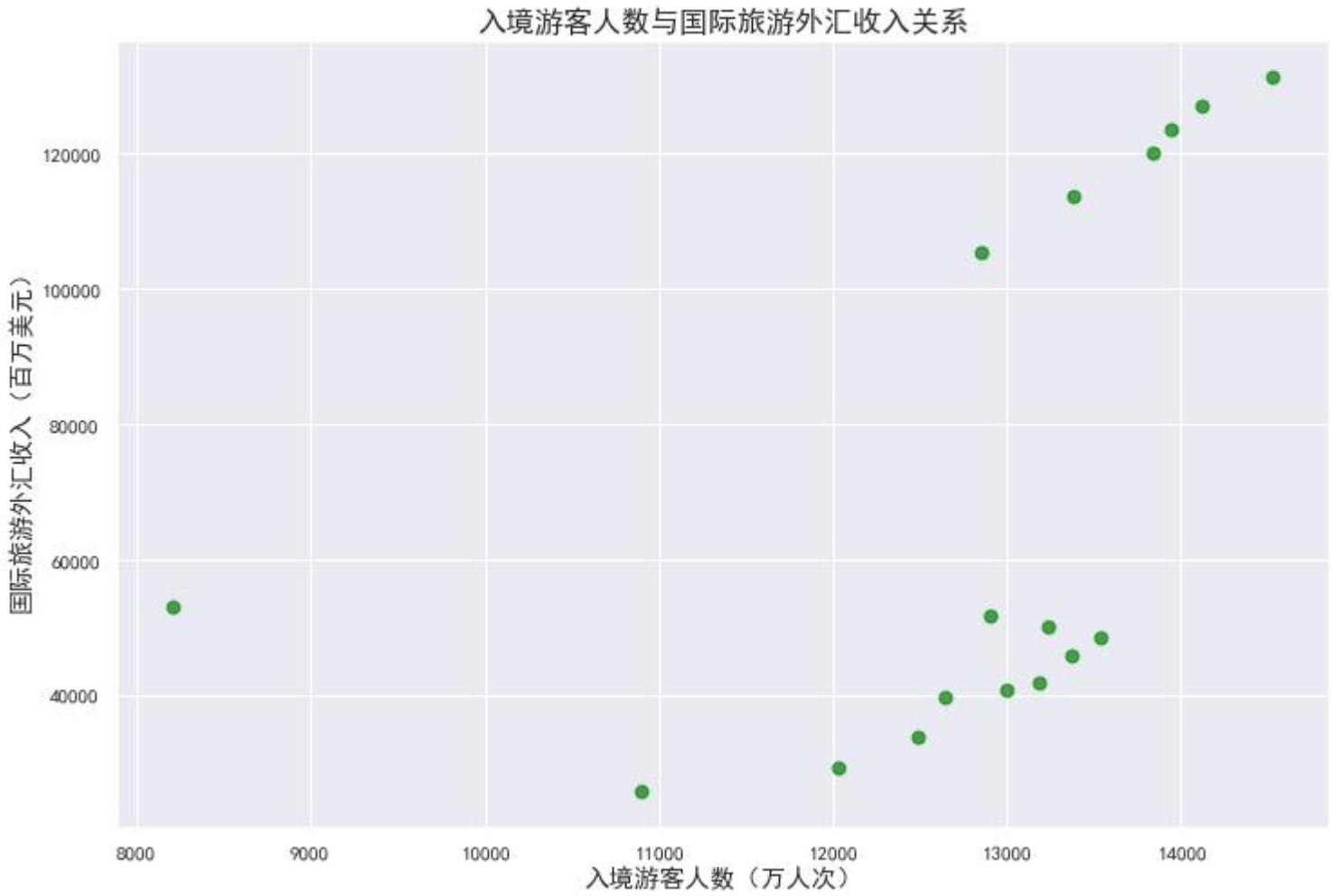
任务案例：¶

读取“data.xls”表中“旅游业发展情况”表单数据，并根据历年“入境游客人数（万人次）”、“国际旅游外汇收入（百万美元）”数据绘制散点图。 为了实现不同的视觉效果，可以通过调整一些参数来改变图表的外观。例如，可以通过 color 参数来改变点的颜色，alpha 参数调整点的透明度，s 参数改变点的大小，如果想要改变点的形状，可以使用 marker 参数，它允许选择不同的

标记样式，如圆形、方形或星形等。图表的整体大小可以通过 figsize 参数来调整。通过参数的调整，可以灵活地改变图表的视觉效果，使其更加符合特定的展示需求。¶

In [36]:

```
import pandas as pdimport matplotlib.pyplot as plt# 设置 matplotlib 字体，确保支持中文plt.rcParams['font.sans-serif'] = ['SimHei'] plt.rcParams['axes.unicode_minus'] = False # 假设 data 是已经加载的 DataFramedata = pd.read_excel('data.xls', sheet_name='旅游业发展情况')tourist_arrivals = data.iloc[2, 1:]foreign_exchange_income = data.iloc[10, 1:]plt.figure(figsize=(12, 8)) plt.scatter(tourist_arrivals, foreign_exchange_income, color='green', alpha=0.7, s=50, marker='o') plt.title('入境游客人数与国际旅游外汇收入关系', fontsize=16) plt.xlabel('入境游客人数（万人次）', fontsize=14) plt.ylabel('国际旅游外汇收入（百万美元）', fontsize=14) plt.grid(True) plt.show()
```


入境游客人数与国际旅游外汇收入关系
国际旅游外汇收入（百万美元）
120000
100000
80000
60000
40000
8000
9000
10000
11000
12000
13000
14000
入境游客人数（万人次）