第 5 章

Seaborn

5.1 Seaborn 简介

在使用 Matplotlib 库绘制图表时，存在两点问题，一是 Matplotlib 库的绘图风格和 MATLAB 类似，即偏古典；二是 Matplotlib 库不能与 Pandas 进行很好结合，进而会导致需要编写较多代码，才可以进行数据展示。

而 Seaborn 库则是在 Matplotlib 库的基础上进行了更高级的 API 封装，可以有效地解决上述两个问题。在大多数情况下，使用 Seaborn 库可以制作出各种具有吸引力的图表，而使用 Matplotlib 库则能制作出具有更多特色的图表，所以 Seaborn 库是 Matplotlib 库的补充，而不是替代。

5.2 图表的背景

可以通过 seaborn 模块中的 set_style()函数对图表的背景进行设置，其语法格式如下：

```
set_style(style,rc)
```

其中，参数 style 表示背景类型，包括 whitegrid(白色网格背景)、darkgrid(灰色网格背景)、white(白色背景)、dark(灰色背景)和 ticks(带刻度线的白色背景)；参数 rc 表示 rc 参数。

5.3 图表的边框

可以通过 seaborn 模块中的 despine()函数对图表的边框进行设置，其语法格式如下：

```
despine(top,right,left,bottom)
```

其中，参数 top 表示是否移除图表的上边框，默认值为 True；参数 right 表示是否移除图表的右边框，默认值为 True；参数 left 表示是否移除图表的左边框，默认值为 False；参数 bottom 表示是否移除图表的下边框，默认值为 False。

5.4 绘制折线图

可以通过 seaborn 模块中的 lineplot()函数绘制折线图，其语法格式如下：

```
lineplot(data,x,y,hue)
```

其中，参数 data 表示数据集；参数 x 为可选参数，并且当参数 data 为长型数据时，用于指定 x 轴对应的数据；参数 y 为可选参数，并且当参数 data 为长型数据时，用于指定 y 轴对应的数据；参数 hue 为可选参数，并且当参数 data 为长型数据时，用于指定数据的分组，其语法格式如下：

```
#资源包\Code\chapter5\5.4\0501.py
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
#设置背景类型
sns.set_style('darkgrid')
#显示中文
plt.rcParams['font.sans-serif'] = 'SimHei'
#显示负号
plt.rcParams['axes.unicode_minus'] = False
#数据集
df_data = pd.read_csv('flights.csv')
flights_wide = df_data.pivot(index = "year", columns = "month", values = "passengers")
#绘制折线图(宽型数据)
#sns.lineplot(data = flights_wide, dashes = False)
#绘制折线图(长型数据)
sns.lineplot(x = "year", y = "passengers", hue = 'month', data = df_data)
```

上面代码的运行结果如图 5-1 和图 5-2 所示。

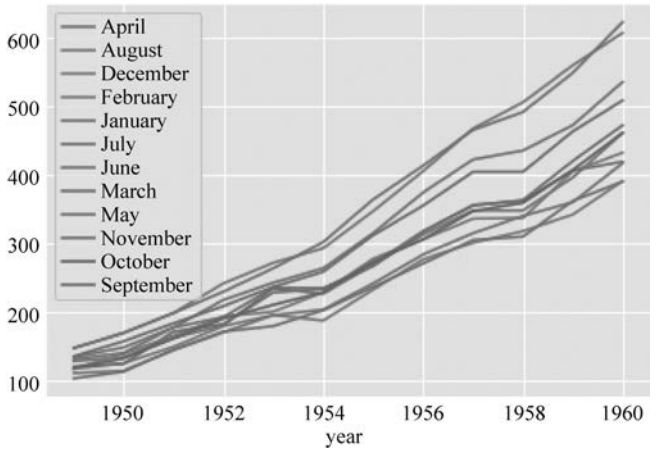

图 5-1 折线图(宽型数据)

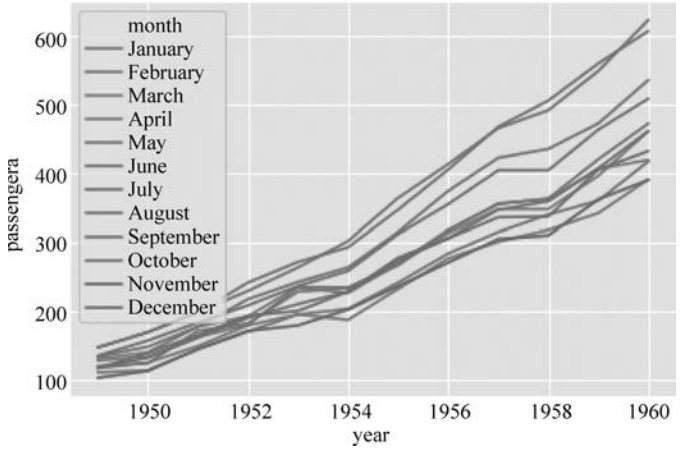

图 5-2 折线图(长型数据)

5.5 绘制柱状图

可以通过 seaborn 模块中的 barplot()函数绘制柱状图,其语法格式如下:

```
barplot(data ,x,y,hue)
```

其中,参数 data 表示数据集; 参数 x 为可选参数,并且当参数 data 为长型数据时,用于指定 x 轴对应的数据; 参数 y 为可选参数,并且当参数 data 为长型数据时,用于指定 y 轴对应的数据; 参数 hue 为可选参数,并且当参数 data 为长型数据时,用于指定数据的分组,其语法格式如下:

```
#资源包\Code\chapter5\5.5\0502.py
import pandas as pd
```

```
import matplotlib.pyplot as plt
import seaborn as sns
#设置背景类型
sns.set_style('darkgrid')
#显示中文
plt.rcParams['font.sans-serif'] = 'SimHei'
#显示负号
plt.rcParams['axes.unicode_minus'] = False
#创建画布
plt.figure(figsize = (10, 8))
#折线图标题
plt.title('flights 数据集')
#数据集
df_data = pd.read_csv('flights.csv')
flights_wide = df_data.pivot(index = "year", columns = "month", values = "passengers")
#绘制柱状图(宽型数据)
#sns.barplot(data = flights_wide)
#绘制柱状图(长型数据)
sns.barplot(x = "year", y = "passengers", hue = 'month', data = df_data)
```

上面代码的运行结果如图 5-3 和图 5-4 所示。

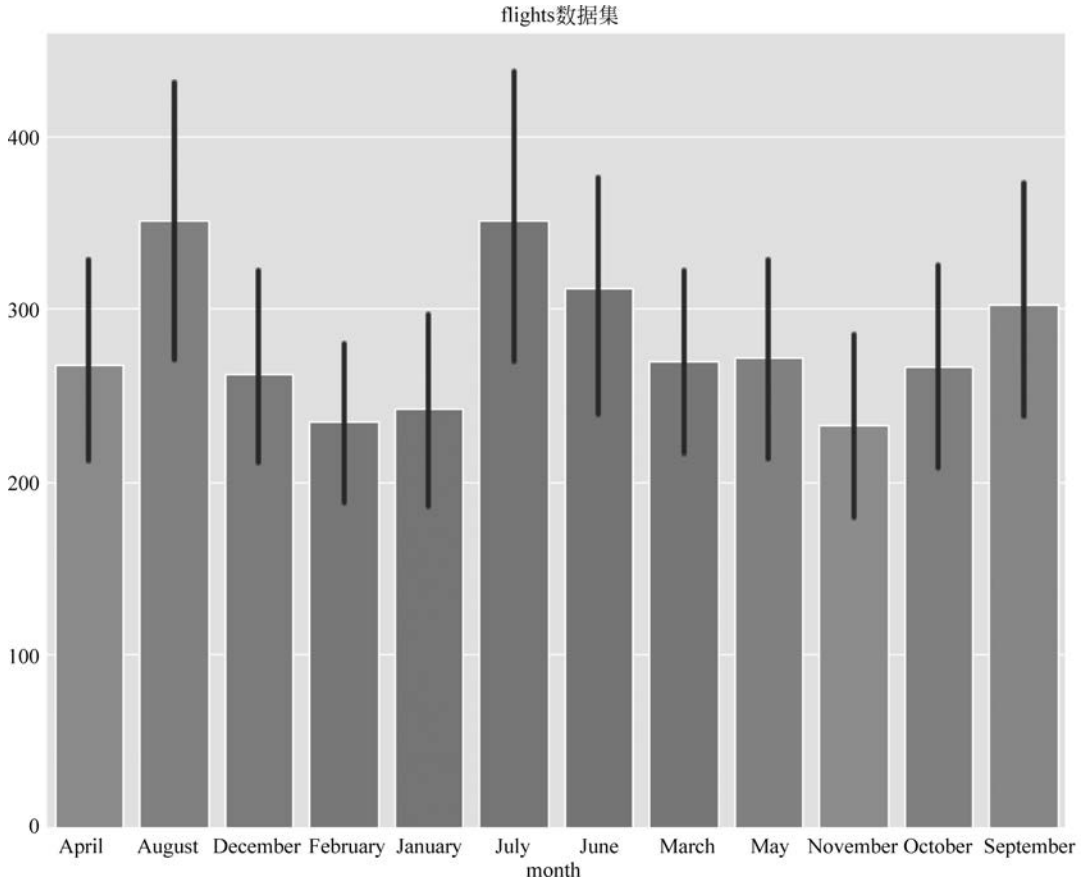

图 5-3　柱状图(宽型数据)

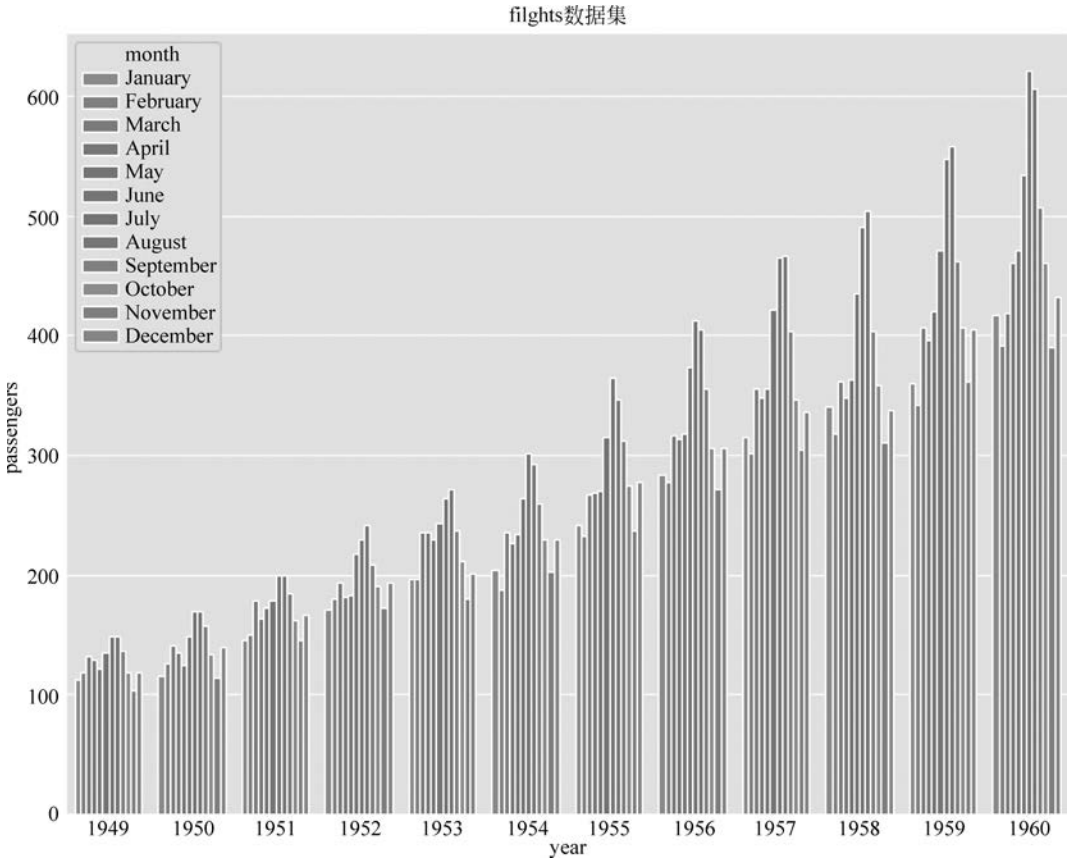

图 5-4 柱状图(长型数据)

5.6 绘制直方图

可以通过 seaborn 模块中的 distplot()函数绘制直方图,其语法格式如下:

```
distplot(a,bins,hist,kde,rug)
```

其中,参数 a 表示数据;参数 bins 表示柱体的数量;参数 hist 表示是否显示柱体;参数 kde 表示是否绘制核密度曲线;参数 rug 表示是否在 x 轴上显示观测值竖线,示例代码如下:

```
#资源包\Code\chapter5\5.6\0503.py
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
#设置背景类型
sns.set_style('darkgrid')
#显示中文
plt.rcParams['font.sans-serif'] = 'SimHei'
#显示负号
```

```
plt.rcParams['axes.unicode_minus'] = False
#数据集
data = np.random.randint(40, 100, (100,))
#绘制直方图
sns.distplot(data, bins = 10)
```

上面代码的运行结果如图 5-5 所示。

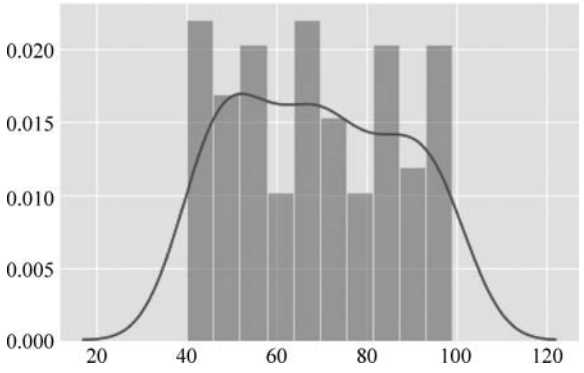

图 5-5　直方图

5.7　绘制散点图

可以通过 seaborn 模块中的 scatterplot()函数绘制散点图，其语法格式如下：

```
scatterplot(data,x,y,hue)
```

其中，参数 data 表示数据集；参数 x 为可选参数，并且当参数 data 为长型数据时，用于指定 x 轴对应的数据；参数 y 为可选参数，并且当参数 data 为长型数据时，用于指定 y 轴对应的数据；参数 hue 为可选参数，并且当参数 data 为长型数据时，用于指定数据的分组，示例代码如下：

```
#资源包\Code\chapter5\5.7\0504.py
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
#设置背景类型
sns.set_style('darkgrid')
#显示中文
plt.rcParams['font.sans-serif'] = 'SimHei'
#显示负号
plt.rcParams['axes.unicode_minus'] = False
#创建画布
plt.figure(figsize = (10, 8))
#数据集
df_data = pd.read_csv('tips.csv')
#绘制散点图
sns. scatterplot(x = "total_bill", y = "tip", hue = 'sex', data = df_data)
```

上面代码的运行结果如图 5-6 所示。

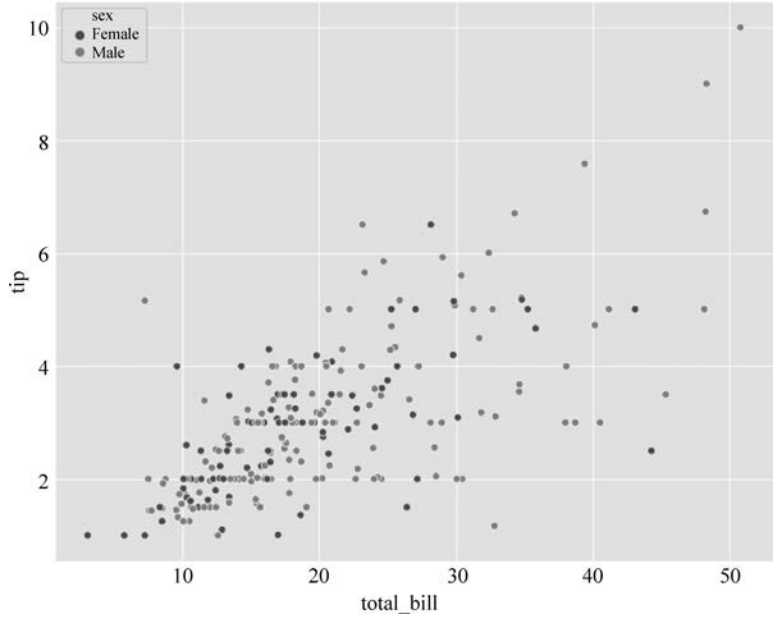

图 5-6 散点图

5.8 绘制分布散点图

可以通过 seaborn 模块中的 stripplot()函数绘制分布散点图，其语法格式如下：

```
stripplot(data,x,y,hue)
```

其中，参数 data 表示数据集；参数 x 为可选参数，并且当参数 data 为长型数据时，用于指定 x 轴对应的数据；参数 y 为可选参数，并且当参数 data 为长型数据时，用于指定 y 轴对应的数据；参数 hue 为可选参数，并且当参数 data 为长型数据时，用于指定数据的分组，示例代码如下：

```
#资源包\Code\chapter5\5.8\0505.py
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
#设置背景类型
sns.set_style('darkgrid')
#显示中文
plt.rcParams['font.sans-serif'] = 'SimHei'
#显示负号
plt.rcParams['axes.unicode_minus'] = False
#创建画布
plt.figure(figsize = (10, 8))
#数据集
```

```
df_data = pd.read_csv('tips.csv')
#绘制小提琴图
sns.stripplot(x = "day", y = "tip", hue = 'sex', data = df_data)
```

上面代码的运行结果如图 5-7 所示。

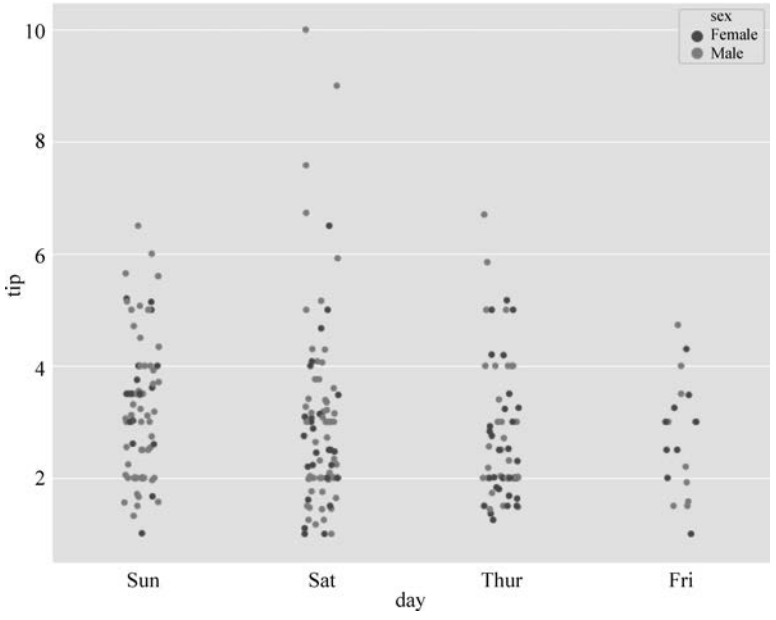

图 5-7　分布散点图

5.9　绘制分簇散点图

可以通过 seaborn 模块中的 swarmplot()函数绘制分簇散点图，其语法格式如下：

```
swarmplot(data,x,y,hue)
```

其中，参数 data 表示数据集；参数 x 为可选参数，并且当参数 data 为长型数据时，用于指定 x 轴对应的数据；参数 y 为可选参数，并且当参数 data 为长型数据时，用于指定 y 轴对应的数据；参数 hue 为可选参数，并且当参数 data 为长型数据时，用于指定数据的分组，示例代码如下：

```
#资源包\Code\chapter5\5.9\0506.py
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
#设置背景类型
sns.set_style('darkgrid')
#显示中文
plt.rcParams['font.sans-serif'] = 'SimHei'
#显示负号
```

```
plt.rcParams['axes.unicode_minus'] = False
#创建画布
plt.figure(figsize = (10, 8))
#数据集
df_data = pd.read_csv('tips.csv')
#绘制散点图
sns.swarmplot(x = "day", y = "tip", hue = 'sex', data = df_data)
```

上面代码的运行结果如图 5-8 所示。

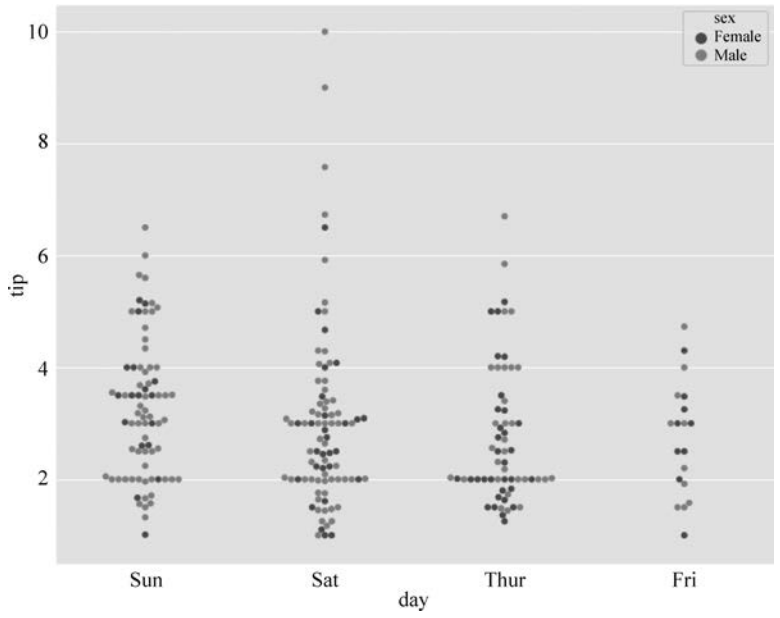

图 5-8 分簇散点图

5.10 绘制箱形图

可以通过 seaborn 模块中的 boxplot()函数绘制箱形图，其语法格式如下：

```
boxplot(data,x,y,hue)
```

其中，参数 data 表示数据集；参数 x 为可选参数，并且当参数 data 为长型数据时，用于指定 x 轴对应的数据；参数 y 为可选参数，并且当参数 data 为长型数据时，用于指定 y 轴对应的数据；参数 hue 为可选参数，并且当参数 data 为长型数据时，用于指定数据的分组，示例代码如下：

```
#资源包\Code\chapter5\5.10\0507.py
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
#设置背景类型
sns.set_style('darkgrid')
```

```
#显示中文
plt.rcParams['font.sans-serif'] = 'SimHei'
#显示负号
plt.rcParams['axes.unicode_minus'] = False
#创建画布
plt.figure(figsize = (10, 8))
#数据集
df_data = pd.read_csv('tips.csv')
#绘制箱形图
sns.boxplot(x = "day", y = "tip", hue = 'sex', data = df_data)
```

上面代码的运行结果如图 5-9 所示。

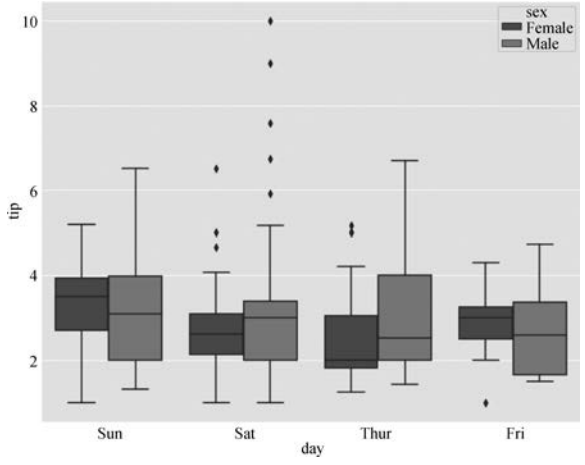

图 5-9　箱形图

5.11　绘制小提琴图

可以通过 seaborn 模块中的 violinplot()函数绘制小提琴图，其语法格式如下：

```
violinplot(data,x,y,hue)
```

其中，参数 data 表示数据集；参数 x 为可选参数，并且当参数 data 为长型数据时，用于指定 x 轴对应的数据；参数 y 为可选参数，并且当参数 data 为长型数据时，用于指定 y 轴对应的数据；参数 hue 为可选参数，并且当参数 data 为长型数据时，用于指定数据的分组，示例代码如下：

```
#资源包\Code\chapter5\5.11\0508.py
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
#设置背景类型
sns.set_style('darkgrid')
#显示中文
```

```
plt.rcParams['font.sans-serif'] = 'SimHei'
#显示负号
plt.rcParams['axes.unicode_minus'] = False
#创建画布
plt.figure(figsize = (10, 8))
#数据集
df_data = pd.read_csv('tips.csv')
#绘制小提琴图
sns.violinplot(x = "day", y = "tip", hue = 'sex', data = df_data)
```

上面代码的运行结果如图 5-10 所示。

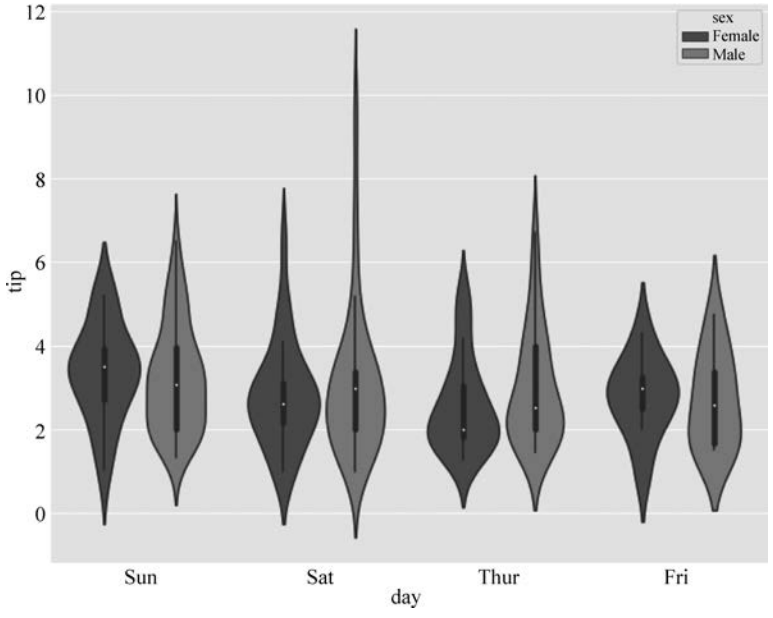

图 5-10 小提琴图

5.12 绘制核密度图

可以通过 seaborn 模块中的 kdeplot()函数绘制核密度图,其语法格式如下:

```
kdeplot(data,shade)
```

其中,参数 data 表示数据集; 参数 shade 表示是否绘制阴影,示例代码如下:

```
#资源包\Code\chapter5\5.12\0509.py
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
#设置背景类型
sns.set_style('darkgrid')
#显示中文
```

```
plt.rcParams['font.sans-serif'] = 'SimHei'
#显示负号
plt.rcParams['axes.unicode_minus'] = False
#创建画布
plt.figure(figsize = (10, 8))
#数据集
data = np.random.randint(40, 100, (100,))
#绘制核密度图
sns.kdeplot(data = data, shade = True)
```

上面代码的运行结果如图 5-11 所示。

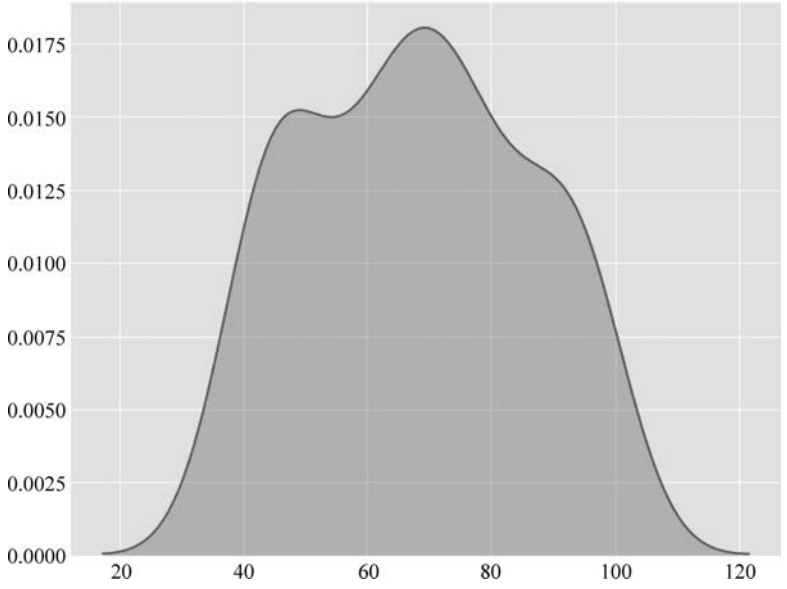

图 5-11　核密度图

5.13　绘制热力图

可以通过 seaborn 模块中的 heatmap()函数绘制热力图，其语法格式如下：

```
heatmap(data,cmap,annot)
```

其中，参数 data 表示数据集；参数 cmap 表示单元格的颜色；参数 annot 表示是否显示单元格中的数据值，示例代码如下：

```
#资源包\Code\chapter5\5.13\0510.py
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
#设置背景类型
```

```
sns.set_style('darkgrid')
#显示中文
plt.rcParams['font.sans-serif'] = 'SimHei'
#显示负号
plt.rcParams['axes.unicode_minus'] = False
#创建画布
plt.figure(figsize = (10, 8))
#数据集
df = pd.DataFrame(np.random.rand(10, 10), columns = list('abcdefghij'))
#绘制热力图
sns.heatmap(data = df, cmap = "Greens", annot = True)
```

上面代码的运行结果如图 5-12 所示。

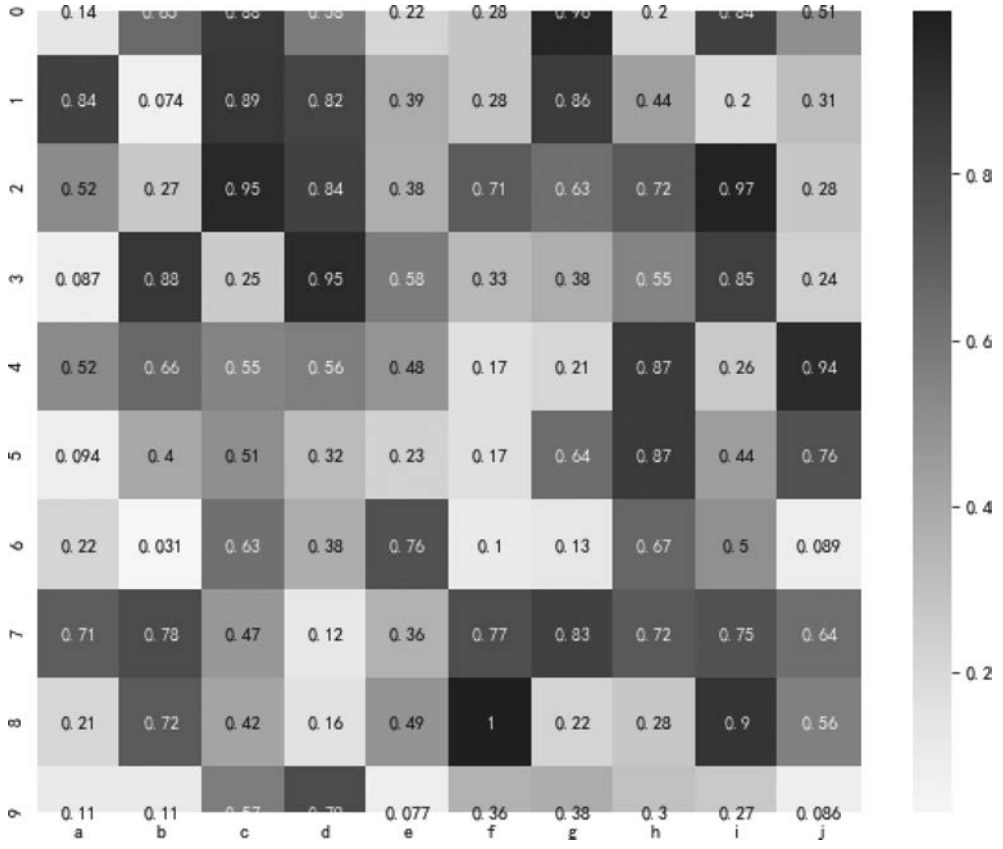

图 5-12 热力图

5.14 绘制聚类热图

可以通过 seaborn 模块中的 clustermap()函数绘制聚类热图，其语法格式如下：

```
clustermap(data,cmap,annot,method)
```

其中，参数 data 表示数据集；参数 cmap 表示单元格的颜色；参数 annot 表示是否显示单元

格中的数据值；参数 method 表示聚类算法，示例代码如下：

```
#资源包\Code\chapter5\5.14\0511.py
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
#设置背景类型
sns.set_style('darkgrid')
#显示中文
plt.rcParams['font.sans-serif'] = 'SimHei'
#显示负号
plt.rcParams['axes.unicode_minus'] = False
#创建画布
plt.figure(figsize = (10, 8))
#数据集
df = pd.DataFrame(np.random.rand(10, 10), columns = list('abcdefghij'))
#绘制聚类热图
sns.clustermap(data = df, cmap = "Greens", annot = True)
```

上面代码的运行结果如图 5-13 所示。

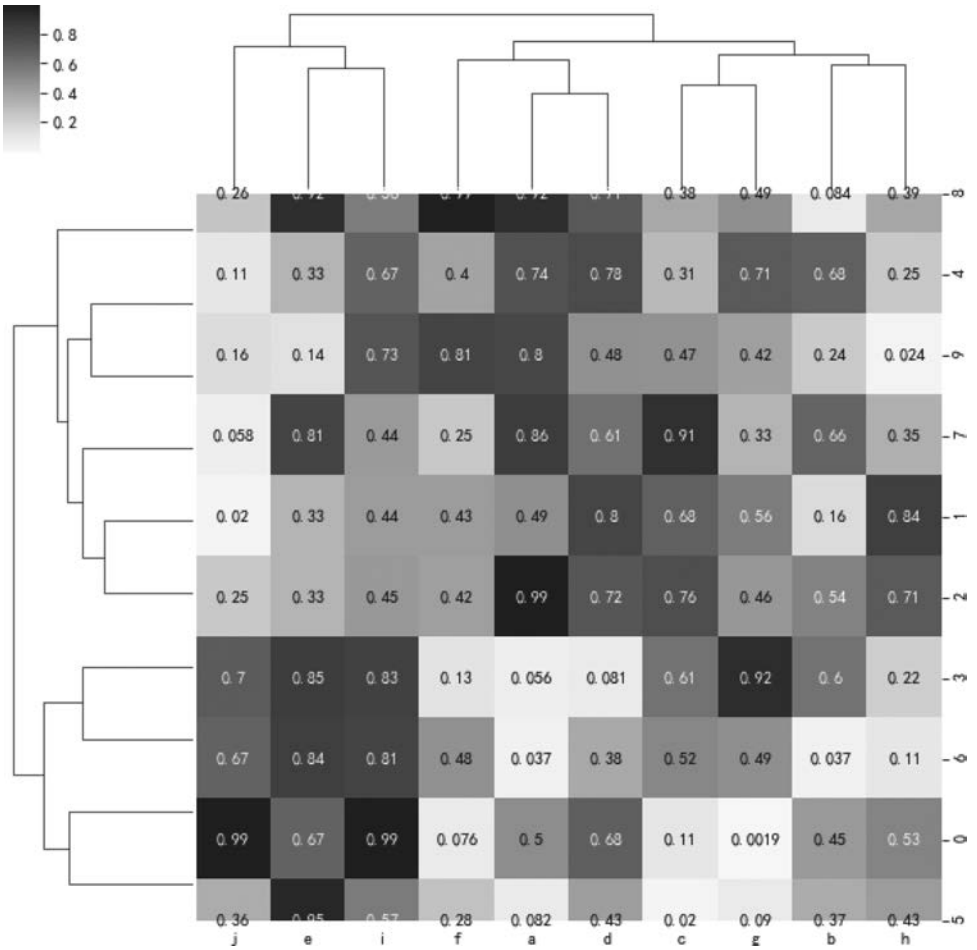

图 5-13　聚类热图

5.15 绘制线性回归图

可以通过 seaborn 模块中的 regplot()函数绘制线性回归图，其语法格式如下：

```
regplot(data,x,y)
```

其中，参数 data 表示数据集；参数 x 表示 x 轴对应的数据；参数 y 表示 y 轴对应的数据，示例代码如下：

```
#资源包\Code\chapter5\5.15\0512.py
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
#设置背景类型
sns.set_style('darkgrid')
#显示中文
plt.rcParams['font.sans-serif'] = 'SimHei'
#显示负号
plt.rcParams['axes.unicode_minus'] = False
#创建画布
plt.figure(figsize = (10, 8))
#数据集
df_data = pd.read_csv('tips.csv')
#绘制线性回归图
sns.regplot(x = 'total_bill', y = 'tip', data = df_data)
```

上面代码的运行结果如图 5-14 所示。

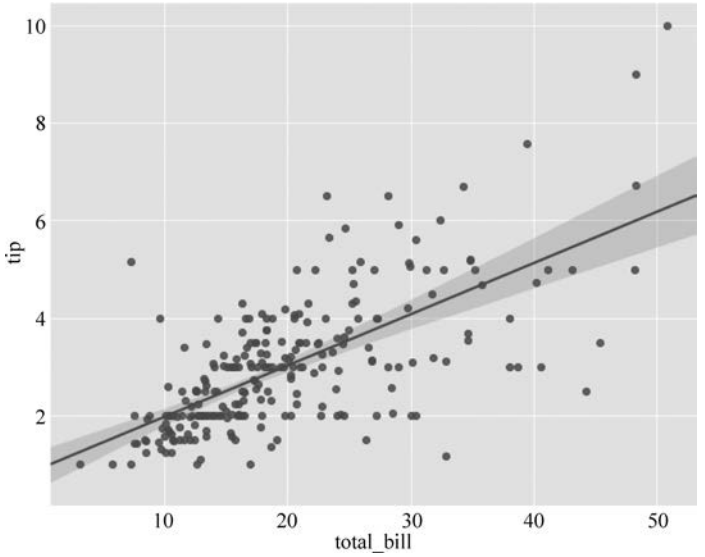

图 5-14 线性回归图