4.7. Python 编程实践

▲ 4.6.数据可视化的发展趋势

▼ 4.8.继续学习本章知识

Python 编程实践

【分析对象】

CSV 文件—文件名为“salaries.csv”。该数据集主要记录了397个样本的6个属性。主要属性如下。 1．rank：职称，包括正教授（Prof）、副教授（AsstProf）、助理教授（AssocProf）。 2．discipline：学科，分为 A 和 B。3．yrs.since.phd：博士毕业年数。 4．yrs.service：工作年数。 5．sex：性别。6．salary：薪水。

【分析目的与任务】

利用 Python 中的 seaborn 包实现数据可视化：首先，准备好数据。其次，导入可视化需要用到的 seaborn 包。 最后，绘制散点图和箱线图。

【分析方法及工具】 Python 及 seaborn 包。

【主要步骤】

与第 2 章中数据统计建模和第3章中的机器学习类似，数据可视化也需要以业务理解和数据理解为前提。由于所涉及的业务和数据非常简单，本例中略过针对业务理解和数据理解活动的讨论。读者如需要了解相关知识，建议参考本书第 2 章和第 3 章 中对 Python 编程实践的讲解。实现数据可视化的步骤除以上表述外，还有数据准备、导入 Python 包和可视化绘图等步骤，具体内容如下。

Step 1:数据准备

- 在数据科学项目中，基于Python 的**数据可视化**是通过调用Python 第三方包来实现的。常用的Python 第三方数据可视化包有Matplotlib、Seaborn、Bokeh、Basemap、 Plotly 和NetworkX 等。
- 考虑到本书第2 章例子中已用过Matplotlib 进行了数据可视化，本例我们采用另一个常用包Seaborn 进行数据可视化。
- 为了数据准备，我们首先采用 Python 模块 os 中的函数 chdir()和 getcwd()分别进行当前工作目录的查看和更改。其次，从本书配套资源中找到数据文件 salaries.csv，存放在当前工作目录下。接着，采用第三方包 Pandas 提供的函数 pd.read_csv()从当前工作目录中读取数据文件 salaries.csv 到 Pandas 数据框 salaries 中。最后，用 Pandas 包提供的函数head()显示数据框salaries 的前5 行。

- 首先，采用模块os 中的函数getcwd()，查看当前工作目录。示例如下。

```
In[1]: #查看当前工作目录，并将数据文件"salaries.csv"放在当前工作目录中
       import os
       os.getcwd()
       #【提示】读者可以在本书配套资源中找到数据文件"salaries.csv"
       #【注意】读者的“当前工作目录”不一定与本书一样，请以自己Out[1]中的显示结果为准
```

对应输出结果为：

```
C:\Users\soloman\clm
```

- 其次，用第三方包 Pandas 提供的函数 read_csv()从当前工作目录读入文件“salaries. csv”到 Pandas 数据框对象 salaries 中。

```
In[2]: import pandas as pd
       salaries = pd.read_csv('salaries.csv', index_col=0)
       #【提示】index_col=0 的含义为，准备读入的数据文件（salaries.csv）中带有索引列，且索引列位于第 0 列
```

- 接着，用Pandas 数据框的head()方法查看数据框alaries 的前6 行。示例如下。

```
In[3] #查看 Pandas 数据框 salaries 的部分内容
      salaries.head(6)
```

对应输出结果为：

	rank	discipline	yrs.since.phd	yrs.service	sex	salary
1	Prof	B	19	18	Male	139750
2	Prof	B	20	16	Male	173200
3	AsstProf	B	4	3	Male	79750
4	Prof	B	45	39	Male	115000
5	Prof	B	40	41	Male	141500
6	AssocProf	B	6	6	Male	97000

Step 2:导入Python 包

本例准备采用 Python 第三方数据可视化包—seaborn，为此，我们需要用 Python 的import 语句导入Seaborn 包。

```
In[4]: #导入 seaborn 模块，并取别名为 sns
       import seaborn as sns
```

Step 3:可视化绘图

接下来，通过调用 seaborn 包提供的各种功能函数实现数据可视化的目的。本例将采用 seaborn包提供的3个函数，即 set_style()、stripplot()和 boxplot()，分别用于设置 seaborn的绘图样式或主题、绘制分类散点图和绘制箱线图。

（1）绘图样式或主题

其中，设置 seaborn 的绘图样式或主题示例如下。

```
In[5]| #设置 seanborn 的绘图样式或主题为“darkgrid”(灰色+网格)
       sns.set_style('darkgrid')
```

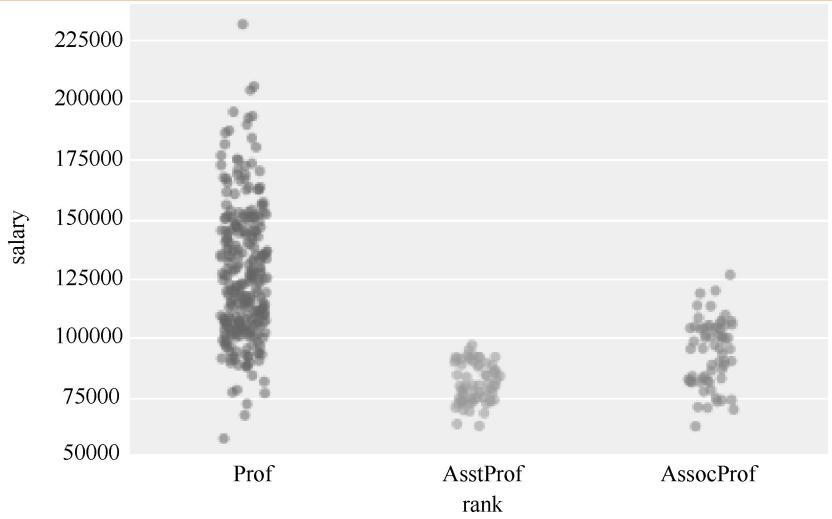
（2）绘制分类散点图

上一行采用了 set_style()函数将 seaborn 的绘图主题改为“带有网格线的灰色背景（darkgrid）”。
在此基础上，我们调用 stripplot()函数继续绘制分类散点图。示例如下。

```
In[6]: #用 stripplot()函数绘制分类散点图
       sns.stripplot(data=salaries, x='rank', y='salary', jitter=True,alpha=0.5)
       # 【提示】data 为数据来源；x 和 y 分别用于设置 x 轴和 y 轴；jitter 的含义为散点是否有抖动（重叠)；alpha 为透明度
```

对应输出结果为：

```
<matplotlib.axes._subplots.AxesSubplot at 0x25f91ea91d0>
```



（3）绘制箱线图

接着，我们在上图显示的结果基础上，增加箱线图。
箱线图的绘制可通过调用boxplot()函数实现。示例如下。

```
In[7]  #继续绘制【箱线图】
       sns.stripplot(data=salaries, x='rank', y='salary', jitter=True,alpha=0.5)
       sns.boxplot(data=salaries, x='rank', y='salary‘)
       # 【提示】data 为数据来源；x 和 y 分别用于设置 x 轴和 y 轴
```

对应输出结果为：

```
<matplotlib.axes._subplots.
AxesSubplot at 0x25f92210e80>
```