第3章

Python表格处理分析


视频讲解

3.1 背景介绍

Office办公软件在日常工作学习中的应用可以说是无处不在，其中Excel是可编程性最好的办公软件，使用Excel时经常会要读取、修改和创建大数据量的Excel表格，纯粹依靠手工完成这些工作十分耗时，而且操作的过程中十分容易出错。本章将介绍如何借助Python的openpyxl模块完成这些工作，提升工作效率。Python中的openpyxl模块能够对Excel文件进行创建、读取和修改，让计算机自动进行大量烦琐重复的Excel文件处理成为可能。本章将围绕以下3个重点内容展开。

(1) 修改已有的Excel表单。

(2) 从Excel表单中提取信息。

(3) 创建更为复杂的Excel表单，为表格添加样式、图表等。

在此之前，读者应该熟知Python的基本语法，能够熟练使用Python的基本数据结构，包括dict、list等，并且理解面向对象编程的基本概念。

在开始之前，读者可能会有疑问：什么时候应该选择使用openpyxl这样的编程工具，而不是直接使用Excel的操作界面来完成工作呢？虽然这样的实际场景数不胜数，但以下这几个例子十分有代表性，提供给读者参考。

假设你在经营一个网店，当你每次需要将新商品上架到网页上时，需要将相应的商品信息填入店铺的系统，而所有的商品信息一开始都记录在若干Excel表格中。如果你需要将这些信息导入系统，就必须遍历Excel表格的每一行，并在店铺系统中重新输入。我们将这种场景抽象成从Excel表单中导出信息。

假设你是一个用户信息系统的管理员，公司在某次促销活动中需要导出所有用户的

联系方式到可打印的文件中，并交给销售人员进行电话营销。显然 Excel 表单是可视化呈现这些信息的不二之选。这样的场景可以称为向 Excel 表单中导入信息。

假设你是一所中学的数学教师，一次期中测验后你需要整理汇总 20 个班级的成绩，并制作相应的统计图表。而令人绝望的是，你发现每个班级的成绩散落在不同的表单文件中，无法使用 Excel 内置的统计工具汇总。我们将这种场景称为 Excel 表单内部的信息聚合与提取。

类似的问题难以枚举，却无不例外地令人头痛。但是，如果学会使用 openpyxl 工具，这些就都不再是问题。

3.2 前期准备与基本操作

3.2.1 基本术语概念说明

在后面章节中将会用表 3-1 中的术语名词来指代表格操作中的具体概念。

表 3-1 基本术语

术 语	含 义
工作簿	指创建或者操作的主要文件对象，通常来讲，一个.xlsx 文件对应一个工作簿
工作表	工作表通常用来划分工作簿中的不同内容，一个工作簿中可以包含多个不同的工作表
列	一列指工作表中垂直排列的一组数据，在 Excel 中，通常用大写字母指代一列，如第一列通常是 A
行	一行指工作表中水平排列的一组数据，在 Excel 中，通常用数字指代一行，如第一行通常是 1
单元格	一个单元格由一个行号和一个列号唯一确定，如 A1 指位于第 A 列第一行的单元格

3.2.2 安装 openpyxl 并创建一个工作簿

如同大多数 Python 模块，我们可以通过 pip 工具安装 openpyxl，只需要在命令行终端中执行以下命令：

```
pip install openpyxl
```

安装完毕之后，就可以用几行代码创建一个十分简单的工作簿了，代码如下所示。

```
1  from openpyxl import Workbook
2
3  workbook = Workbook()
4  sheet = workbook.active
5
6  sheet["A1"] = "hello"
7  sheet["B1"] = "world!"
8
9  workbook.save(filename = "hello_world.xlsx")
```

首先从 openpyxl 包中导入 Workbook 对象，并在第 3 行创建一个实例 workbook。在第 4 行中，通过 workbook 的 active 属性获取默认的工作表。在第 6 行和第 7 行中，向工作表的 A1 和 B1 两个位置分别插入 hello 和 world! 两个字符串。最后，通过 workbook 的 save 方法，将新工作簿存储在名为 hello_world.xlsx 的文件中。打开该文件，可以看到文件内容如图 3-1 所示。

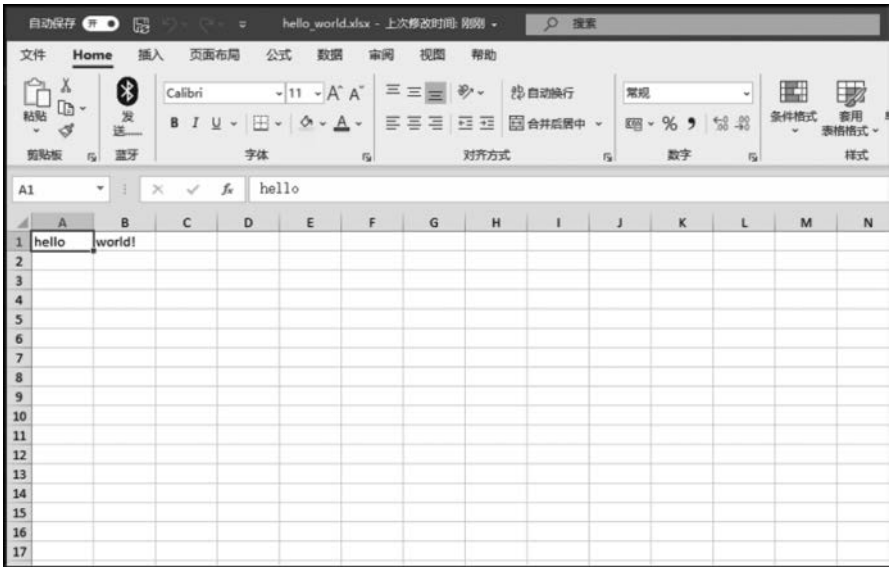


图 3-1　hello_world.xlsx 文件

3.2.3　从 Excel 工作簿中读取数据

本节为读者提供了样例工作簿 sample.xlsx，其中包含了一些亚马逊在线商店的商品评价数据。读者可以在章节对应的附件中找到这个文件，并放置在实验代码的根目录下。之后的样例程序将在该样例工作簿的基础上进行演示。

准备好数据文件后，就可以在 Python 命令行终端中尝试打开并读取一个 Excel 工作簿了。在命令行中输入 Python 命令，进入 Python 命令行终端，接下来的操作代码如下所示。

```
>>> from openpyxl import load_workbook
>>> workbook = load_workbook(filename = "sample.xlsx")
>>> workbook.sheetnames
['Sheet 1']

>>> sheet = workbook.active
>>> sheet
<Worksheet "Sheet 1">

>>> sheet.title
'Sheet 1'
```

为了读取工作簿，需要按照第 1 处的命令从 openpyxl 包中导入 load_workbook 函数。在第 2 行中，通过调用 load_workbook 函数并指定路径名，可以得到一个 workbook 对象。workbook 的 sheetnames 属性为工作簿中所有工作表的名字列表。workbook.active 为当前工作簿的默认工作表，我们用 sheet 变量指向它。sheet 的 title 属性为当前工作表的名称。这个样例是打开工作表的最常见的方式，请读者熟练掌握。

打开工作表后，读者可以检索特定位置的数据，代码如下：

```
1   >>> sheet["A1"]
2   <Cell 'Sheet 1'.A1>
3
4   >>> sheet["A1"].value
5   'marketplace'
6
7   >>> sheet["F10"].value
8   "G-Shock Men's Grey Sport Watch"
```

sheet 对象类似于一个字典，可以通过组合行列序号的方式得到对应位置的键，然后使用键在 sheet 对象中获取相应的值。值的形式为 Cell 类型的对象，如第 1 行和第 2 行所示。如果想要获取相应单元格中的内容，可以通过访问 Cell 对象的 value 字段来完成（第 4～8 行）。除此之外，也可以通过 sheet 对象的 cell()方法获取特定位置的 Cell 对象和对应的值，代码如下所示。

```
>>> sheet.cell(row=10, column=6)
<Cell 'Sheet 1'.F10>

>>> sheet.cell(row=10, column=6).value
"G-Shock Men's Grey Sport Watch"
```

尽管在 Python 中索引的序号总是从 0 开始，但对 Excel 表单而言，行号和列号总是从 1 开始的，在使用 cell()方法时需要留意这一点。

3.2.4 迭代访问数据

本节将会讲解如何遍历访问工作表中的数据，openpyxl 提供了十分方便的数据选取工具，而且使用方式十分接近 Python 语法。依据不同的需求，有如下几种不同的访问方式。

第一种方式是通过组合两个单元格的位置选择一个矩形区域的 Cell，代码如下所示。

```
>>> sheet["A1:C2"]
((<Cell 'Sheet 1'.A1>, <Cell 'Sheet 1'.B1>, <Cell 'Sheet 1'.C1>),
(<Cell 'Sheet 1'.A2>, <Cell 'Sheet 1'.B2>, <Cell 'Sheet 1'.C2>))
```

第二种方式是通过指定行号或列号选择一整行或一整列的数据，代码如下所示。

```
>>> # Get all cells from column A
>>> sheet["A"]
(<Cell 'Sheet 1'.A1>,
<Cell 'Sheet 1'.A2>,
...
<Cell 'Sheet 1'.A99>,
<Cell 'Sheet 1'.A100>)

>>> # Get all cells for a range of columns
>>> sheet["A:B"]
((<Cell 'Sheet 1'.A1>,
<Cell 'Sheet 1'.A2>,
...
<Cell 'Sheet 1'.A99>,
<Cell 'Sheet 1'.A100>),
(<Cell 'Sheet 1'.B1>,
<Cell 'Sheet 1'.B2>,
...
<Cell 'Sheet 1'.B99>,
<Cell 'Sheet 1'.B100>))

>>> # Get all cells from row 5
>>> sheet[5]
(<Cell 'Sheet 1'.A5>,
<Cell 'Sheet 1'.B5>,
...
<Cell 'Sheet 1'.N5>,
<Cell 'Sheet 1'.O5>)

>>> # Get all cells for a range of rows
>>> sheet[5:6]
((<Cell 'Sheet 1'.A5>,
<Cell 'Sheet 1'.B5>,
...
<Cell 'Sheet 1'.N5>,
<Cell 'Sheet 1'.O5>),
(<Cell 'Sheet 1'.A6>,
<Cell 'Sheet 1'.B6>,
...
<Cell 'Sheet 1'.N6>,
<Cell 'Sheet 1'.O6>))
```

第三种方式是通过如下基于 Python generator 的两个函数来获取单元格：

(1) iter_rows()。

(2) iter_cols()。

这两个函数都可以接收以下 4 个参数：

(1) min_row。

（2）max_row。

（3）min_col。

（4）max_col。

使用方式如下所示。

```
>>> for row in sheet.iter_rows(min_row = 1,
...                         max_row = 2,
...                         min_col = 1,
...                         max_col = 3):
...       print(row)
(<Cell 'Sheet 1'.A1>, <Cell 'Sheet 1'.B1>, <Cell 'Sheet 1'.C1>)
(<Cell 'Sheet 1'.A2>, <Cell 'Sheet 1'.B2>, <Cell 'Sheet 1'.C2>)


>>> for column in sheet.iter_cols(min_row = 1,
...                         max_row = 2,
...                         min_col = 1,
...                         max_col = 3):
...       print(column)
(<Cell 'Sheet 1'.A1>, <Cell 'Sheet 1'.A2>)
(<Cell 'Sheet 1'.B1>, <Cell 'Sheet 1'.B2>)
(<Cell 'Sheet 1'.C1>, <Cell 'Sheet 1'.C2>)
```

如果在调用函数时将 values_only 设置为 True，将只返回每个单元格的值，代码如下所示。

```
>>> for value in sheet.iter_rows(min_row = 1,
...                         max_row = 2,
...                         min_col = 1,
...                         max_col = 3,
...                         values_only = True):
...       print(value)
('marketplace', 'customer_id', 'review_id')
('US', 3653882, 'R3O9SGZBVQBV76')
```

同时，sheet 对象的 rows 对象和 columns 对象本身即是迭代器，如果不需要指定特定的行列，而只是想遍历整个数据集，可以使用如下代码访问数据。

```
>>> for row in sheet.rows:
...       print(row)
(<Cell 'Sheet 1'.A1>, <Cell 'Sheet 1'.B1>, <Cell 'Sheet 1'.C1>
...
<Cell 'Sheet 1'.M100>, <Cell 'Sheet 1'.N100>, <Cell 'Sheet 1'.O100>)
```

通过使用上述的方法，相信你已经学会如何读取 Excel 表单中的数据了，以下实例代

码展示了一个完整的读取数据并转化为 json 序列的流程。

```
import json
from openpyxl import load_workbook

workbook = load_workbook(filename = "sample.xlsx")
sheet = workbook.active

products = {}

# values_only 参数要设为 True,因为这里想返回单元格的数值
for row in sheet.iter_rows(min_row = 2,
                           min_col = 4,
                           max_col = 7,
                           values_only = True):
    product_id = row[0]
    product = {
        "parent": row[1],
        "title": row[2],
        "category": row[3]
    }
    products[product_id] = product

# 使用 json 库,以便之后呈现更好的输出格式
print(json.dumps(products))
```

3.2.5 插入数据

以下为示例代码，当向 B10 单元格中添加了数据之后，openpyxl 会自动插入 10 行数据，中间未定义的位置的值为 None。

```
>>> def print_rows():
...       for row in sheet.iter_rows(values_only = True):
...           print(row)

>>> # 在行代码之前,表格中仅有 1 行
>>> print_rows()
('hello', 'world!')

>>> # 这行代码尝试往第 10 行添加一个新值
>>> sheet["B10"] = "test"
>>> print_rows()
('hello', 'world!')
(None, None)
(None, None)
(None, None)
```

```
(None, None)
(None, None)
(None, None)
(None, None)
(None, None)
(None, 'test')
```

接下来介绍如何插入和删除行列，openpyxl 模块提供了以下非常直观的 4 个函数。

(1) insert_rows()。

(2) delete_rows()。

(3) insert_cols()。

(4) delete_cols()。

每个函数接受两个参数：idx 和 amount。idx 指明了从哪个位置开始插入和删除，amount 指明了插入或删除的数量。示例程序如下所示。

```
>>> print_rows()
('hello', 'world!')

>>> # 在已存在的 A 列后插入新的一列
>>> sheet.insert_cols(idx = 1)
>>> print_rows()
(None, 'hello', 'world!')

>>> # 在 B 列和 C 列之间插入新的 5 列
>>> sheet.insert_cols(idx = 3, amount = 5)
>>> print_rows()
(None, 'hello', None, None, None, None, None, 'world!')

>>> # 删掉之前插入的 5 列
>>> sheet.delete_cols(idx = 3, amount = 5)
>>> sheet.delete_cols(idx = 1)
>>> print_rows()
('hello', 'world!')

>>> # 在表格最上面插入新的一行
>>> sheet.insert_rows(idx = 1)
>>> print_rows()
(None, None)
('hello', 'world!')

>>> # 在表格最上面插入新的 3 行
>>> sheet.insert_rows(idx = 1, amount = 3)
>>> print_rows()
(None, None)
(None, None)
```

```
(None, None)
(None, None)
('hello', 'world!')

>>> # 删掉前 4 行
>>> sheet.delete_rows(idx = 1, amount = 4)
>>> print_rows()
('hello', 'world!')
```

注意，当使用函数插入数据时，插入实际发生在 idx 参数所指特定行或列的前一个位置，例如调用 insert_rows(1)，新插入的行将会在原先的第一行之前，成为新的第一行。

3.3 进阶内容

3.3.1 为 Excel 表单添加公式

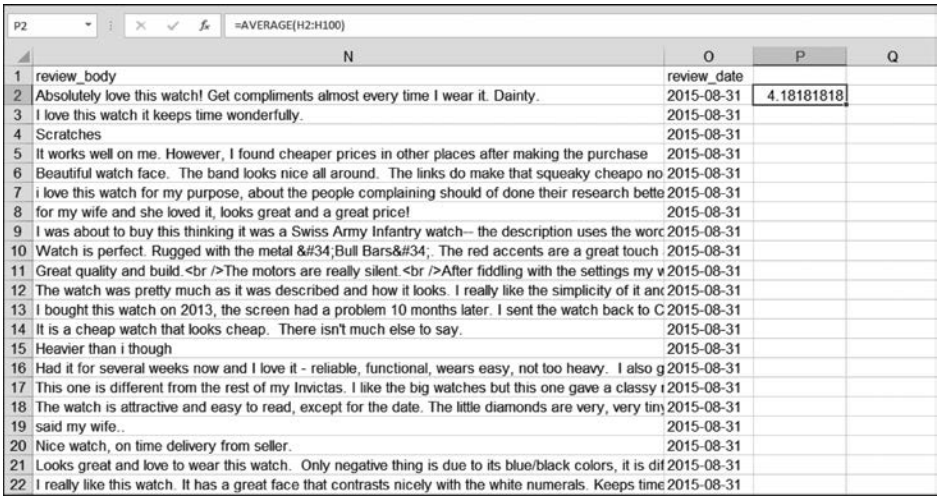
公式计算可以说是 Excel 中最重要的功能，也是 Excel 表单相比其他数据记录工具最为强大的地方。通过使用公式，可以在任意单元格的数据上应用数学方程，得到期望的统计或计量结果。在 openpyxl 中使用公式和在 Excel 中编辑公式一样简单，以下示例程序展示了如何查看 openpyxl 中支持的公式类型。

```
>>> from openpyxl.utils import FORMULAE
>>> FORMULAE
frozenset({'ABS',
         'ACCRINT',
         'ACCRINTM',
         'ACOS',
         'ACOSH',
         'AMORDEGRC',
         'AMORLINC',
         'AND',
         ...
         'YEARFRAC',
         'YIELD',
         'YIELDDISC',
         'YIELDMAT',
         'ZTEST'})
```

向单元格中添加公式的操作与赋值操作非常类似，示例代码如下所示，计算第 H 列第 2～100 行的平均值。

```
>>> workbook = load_workbook(filename = "sample.xlsx")
>>> sheet = workbook.active
>>> # 给第 H 列排序
>>> sheet["P2"] = " = AVERAGE(H2:H100)"
>>> workbook.save(filename = "sample_formulas.xlsx")
```

操作后的 Excel 表单如图 3-2 所示。



P2　=AVERAGE(H2:H100)

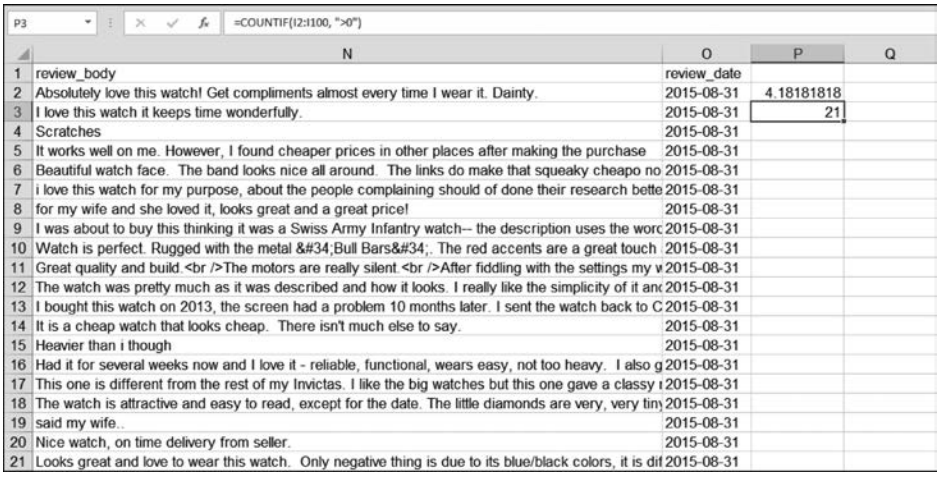
	N	O	P	Q
1	review_body	review_date		
2	Absolutely love this watch! Get compliments almost every time I wear it. Dainty.	2015-08-31	4.18181818	
3	I love this watch it keeps time wonderfully.	2015-08-31		
4	Scratches	2015-08-31		
5	It works well on me. However, I found cheaper prices in other places after making the purchase	2015-08-31		
6	Beautiful watch face. The band looks nice all around. The links do make that squeaky cheapo no	2015-08-31		
7	i love this watch for my purpose, about the people complaining should of done their research bette	2015-08-31		
8	for my wife and she loved it, looks great and a great price!	2015-08-31		
9	I was about to buy this thinking it was a Swiss Army Infantry watch-- the description uses the worc	2015-08-31		
10	Watch is perfect. Rugged with the metal "Bull Bars". The red accents are a great touch	2015-08-31		
11	Great quality and build. The motors are really silent. After fiddling with the settings my v	2015-08-31		
12	The watch was pretty much as it was described and how it looks. I really like the simplicity of it anc	2015-08-31		
13	I bought this watch on 2013, the screen had a problem 10 months later. I sent the watch back to C	2015-08-31		
14	It is a cheap watch that looks cheap. There isn't much else to say.	2015-08-31		
15	Heavier than i though	2015-08-31		
16	Had it for several weeks now and I love it - reliable, functional, wears easy, not too heavy. I also g	2015-08-31		
17	This one is different from the rest of my Invictas. I like the big watches but this one gave a classy r	2015-08-31		
18	The watch is attractive and easy to read, except for the date. The little diamonds are very, very tin	2015-08-31		
19	said my wife..	2015-08-31		
20	Nice watch, on time delivery from seller.	2015-08-31		
21	Looks great and love to wear this watch. Only negative thing is due to its blue/black colors, it is dif	2015-08-31		
22	I really like this watch. It has a great face that contrasts nicely with the white numerals. Keeps time	2015-08-31		

图 3-2　sample_formulas. xlsx

在需要添加的公式中有时候会出现引号包围的字符串，这时需要特别留意。有两种方式应对这个问题：将最外围改为单引号；对公式中的双引号使用转义符。例如我们要统计第 I 列的数据中大于 0 的个数，代码如下所示。

```
>>> # 统计第 I 列中大于 0 的数据个数
>>> sheet["P3"] = '=COUNTIF(I2:I100, ">0")'
>>> # or sheet["P3"] = "=COUNTIF(I2:I100, \">0\")"
>>> workbook.save(filename = "sample_formulas.xlsx")
```

统计结果如图 3-3 所示。



P3　=COUNTIF(I2:I100, ">0")

	N	O	P	Q
1	review_body	review_date		
2	Absolutely love this watch! Get compliments almost every time I wear it. Dainty.	2015-08-31	4.18181818	
3	I love this watch it keeps time wonderfully.	2015-08-31	21	
4	Scratches	2015-08-31		
5	It works well on me. However, I found cheaper prices in other places after making the purchase	2015-08-31		
6	Beautiful watch face. The band looks nice all around. The links do make that squeaky cheapo no	2015-08-31		
7	i love this watch for my purpose, about the people complaining should of done their research bette	2015-08-31		
8	for my wife and she loved it, looks great and a great price!	2015-08-31		
9	I was about to buy this thinking it was a Swiss Army Infantry watch-- the description uses the worc	2015-08-31		
10	Watch is perfect. Rugged with the metal "Bull Bars". The red accents are a great touch	2015-08-31		
11	Great quality and build. The motors are really silent. After fiddling with the settings my v	2015-08-31		
12	The watch was pretty much as it was described and how it looks. I really like the simplicity of it anc	2015-08-31		
13	I bought this watch on 2013, the screen had a problem 10 months later. I sent the watch back to C	2015-08-31		
14	It is a cheap watch that looks cheap. There isn't much else to say.	2015-08-31		
15	Heavier than i though	2015-08-31		
16	Had it for several weeks now and I love it - reliable, functional, wears easy, not too heavy. I also g	2015-08-31		
17	This one is different from the rest of my Invictas. I like the big watches but this one gave a classy r	2015-08-31		
18	The watch is attractive and easy to read, except for the date. The little diamonds are very, very tin	2015-08-31		
19	said my wife..	2015-08-31		
20	Nice watch, on time delivery from seller.	2015-08-31		
21	Looks great and love to wear this watch. Only negative thing is due to its blue/black colors, it is dif	2015-08-31		

图 3-3　添加计数统计的 sample_formulas

3.3.2 为表单添加条件格式

条件格式是指表单根据单元格中不同的数据自动地应用预先设定的不同种类的格式。举一个比较常见的例子，如果想让成绩统计册中所有不及格的学生都被高亮地显示出来，那么条件格式就是最恰当的工具。下面在 sample.xlsx 数据表上演示示例。

以下代码实现了一个简单的功能：将所有评分为 3 以下的行标成红色。

```
>>> from openpyxl.styles import PatternFill, colors
>>> from openpyxl.styles.differential import DifferentialStyle
>>> from openpyxl.formatting.rule import Rule

>>> red_background = PatternFill(bgColor = colors.RED)
>>> diff_style = DifferentialStyle(fill = red_background)
>>> rule = Rule(type = "expression", dxf = diff_style)
>>> rule.formula = ["$H1 < 3"]
>>> sheet.conditional_formatting.add("A1:O100", rule)
>>> workbook.save("sample_conditional_formatting.xlsx")
```

第 1 行 openpyxl.style 中引入了 PatternFill 和 colors 两个对象，这两个对象是为了设定目标数据行的格式属性。在第 2 行中引入了 DifferentialStyle 这个包装类，可以将字体、边界、对齐等多种不同的属性聚合在一起。第 3 行引入了 Rule 类，通过 Rule 类可以设定填充属性需要满足的条件。如第 5～9 行所示，应用条件格式的主要流程为先构建 PatternFill 对象 red_background，再构建 DifferentialStyle 对象 diff_style，diff_style 将作为 rule 对象构建的参数。构建 rule 对象时，需要指明 rule 的类型为 expression，即通过表达式进行选择。在第 8 行，指明了 rule 的公式为满足第 H 列数值小于 3 的相应行，此处的公式语法与 Excel 软件中的公式语法一致。

评分为 3 以下的条目均被标红，如图 3-4 所示。

	B	C	D	E	F
1	customer_id	review_id	product_id	product_parent	product_title
2	3653882	R3O9SGZBVQBV76	B00FALQ1	937001370	Invicta Women's 15150 "Angel" 18k Yellow Gold Ion-Plated Stainless Steel and Brown Leather Watch
3	14661224	RKH8BNC3L5DLF	B00D3RG(	484010722	Kenneth Cole New York Women's KC4944 Automatic Silver Automatic Mesh Bracelet Analog Watch
4	27324930	R2HLE8WKZSU3NL	B00DKYC7	361166390	Ritche 22mm Black Stainless Steel Bracelet Watch Band Strap Pebble Time/Pebble Classic
5	7211452	R31U3UH5AZ42LL	B000EQS1	958035625	Citizen Men's BM8180-03E Eco-Drive Stainless Steel Watch with Green Canvas Band
6	12733322	R2SV659OUJ945Y	B00A6GFD	765328221	Orient ER27009B Men's Symphony Automatic Stainless Steel Black Dial Mechanical Watch
7	6576411	RA51CP8TR5A2L	B00EYSO5	230493695	Casio Men's GW-9400BJ-1JF G-Shock Master of G Rangeman Digital Solar Black Carbon Fiber Insert Watch
8	11811565	RB2Q7DLDN6TH6	B00WM0Q	549298279	Fossil Women's ES3851 Urban Traveler Multifunction Stainless Steel Watch - Rose
9	49401598	R2RHFJV0UYBK3Y	B00A4EYE	844009113	INFANTRY Mens Night Vision Analog Quartz Wrist Watch with Nato Nylon Watchband-Red.
10	45925069	R2Z6JOQ94LFHEP	B00MAMP(	263720892	G-Shock Men's Grey Sport Watch
11	44751341	RX27XIIWY5JPB	B004LBPB	124278407	Heiden Quad Watch Winder in Black Leather
12	9962330	R15C7QEZT0LGZN	B00KGTV(	28017857	Fossil Women's ES3621 Serena Crystal-Accented Two-Tone Stainless Steel Watch
13	16097204	R361XSS37V0NCZ	B0039UT5	685450910	Casio General Men's Watches Sporty Digital AE-2000W-1AVDF - WW
14	51330346	ROTNLALUAJAUB	B00MPF0X	767769082	2Tone Gold Silver Cable Band Ladies Bangle Cuff Watch
15	4201739	R2DYX7QU6BGOHR	B003P1OH	648595227	Bulova Men's 98B143 Precisionist Charcoal Grey Dial Bracelet Watch
16	26339765	RWASY7FKI7QOT	B00R70YE	457338020	Casio - G-Shock - Gulfmaster - Black - GWN1000C-1A
17	2692576	R2KKYZIN3CCL21	B000FVE3	824370661	Invicta Men's 3329 Force Collection Lefty Watch
18	44713366	R22H4FGVD5O52O	B008X6JB	814431355	Seiko Women's SUT068 Dress Solar Classic Diamond-Accented Two-Tone Stainless Steel Watch
19	32778769	R11UACZERCM4ZY	B0040UOF	187700878	Anne Klein Women's 109271MPTT Swarovski Crystal Accented Two-Tone Multi-Chain Bracelet Watch
20	27258523	R1AT8NQ38UQOL6	B00UR2R5	594315262	Guess U13630G1 Men's day and date Gunmetal dial Gunmetal tone bracelet
21	42646538	R2NCZRQGIF1Q75	B00HFF57	520810507	Nixon Men's Geo Volt Sentry Stainless Steel Watch with Link Bracelet
22	46017899	RJ9HWWMU4IAHF	B00F5O06	601596859	Nautica Men's N14699G BFD 101 Chrono Classic Stainless Steel Watch with Brown Band
23	37192375	R3CNTCKG352GL1	B00CHS39	798261110	HDE Watch Link Pin Remover Band Strap Repair Tool Kit for Watchmakers with Pack of 3 Extra Pins
24	11710007	R9Q2LDSES6NBL	B003OQ4T	557813802	Timex Women's Q7B860 Padded Calfskin 8mm Black Replacement Watchband
25	6673146	R3629T8HDV5VWU	B007X0SY	22870009	Movado Men's 0606545 "Museum" Perforated Black-Rubber Strap Sport Watch
26	7899951	R2CLMKC0IVZ9UX	B005KPL7	269520616	Invicta Men's 6674 Corduba Chronograph Black Dial Polyurethane Watch
27	27979201	R2QGEJRU4ENYN2	B00FNIFI2	330558574	Szanto Men's SZ 2001 2000 Series Classic Vintage-Inspired Stainless Steel Watch with Pebbled Leather Band
28	912779	R2E5STTYU6LTSC	B005JVP0	220345054	Casio Men's MRW200H-7EV Sport Resin Watch
29	44345527	RI97MTVX08KWX	B004M23S	299884359	Casio F-108WH-2AEF Mens Blue Digital Watch
30	2659331	RK20LJG750ERC	B00RV2L8	714311106	August Steiner Men's AS8160TTG Silver And Gold Swiss Quartz Watch with Black Dial and Two Tone Bracelet

图 3-4 评分为 3 以下的条目均被标红

为了方便起见，openpyxl 提供了以下三种内置的格式，可以让使用者快速地创建条件格式。

（1）ColorScale。

（2）IconSet。

（3）DataBar。

ColorScale 可以根据数值的大小创建色阶，使用方法如下所示。

```
>>> from openpyxl.formatting.rule import ColorScaleRule
>>> color_scale_rule = ColorScaleRule(start_type = "num",
...                                   start_value = 1,
...                                   start_color = colors.RED,
...                                   mid_type = "num",
...                                   mid_value = 3,
...                                   mid_color = colors.YELLOW,
...                                   end_type = "num",
...                                   end_value = 5,
...                                   end_color = colors.GREEN)

>>> # 将这个梯度加到第 H 列
>>> sheet.conditional_formatting.add("H2:H100", color_scale_rule)
>>> workbook.save(filename = "sample_conditional_formatting_color_scale_3.xlsx")
```

效果如图 3-5 所示，单元格的颜色随着评分由高到低逐渐由绿变红。

product_title	product_category	star_rating
Invicta Women's 15150 "Angel" 18k Yellow Gold Ion-Plated Stainless Steel and Brown Leather Watch	Watches	5
Kenneth Cole New York Women's KC4944 Automatic Silver Automatic Mesh Bracelet Analog Watch	Watches	5
Ritche 22mm Black Stainless Steel Bracelet Watch Band Strap Pebble Time/Pebble Classic	Watches	2
Citizen Men's BM8180-03E Eco-Drive Stainless Steel Watch with Green Canvas Band	Watches	5
Orient ER27009B Men's Symphony Automatic Stainless Steel Black Dial Mechanical Watch	Watches	4
Casio Men's GW-9400BJ-1JF G-Shock Master of G Rangeman Digital Solar Black Carbon Fiber Insert Watch	Watches	5
Fossil Women's ES3851 Urban Traveler Multifunction Stainless Steel Watch - Rose	Watches	5
INFANTRY Mens Night Vision Analog Quartz Wrist Watch with Nato Nylon Watchband-Red.	Watches	1
G-Shock Men's Grey Sport Watch	Watches	5
Heiden Quad Watch Winder in Black Leather	Watches	4
Fossil Women's ES3621 Serena Crystal-Accented Two-Tone Stainless Steel Watch	Watches	4
Casio General Men's Watches Sporty Digital AE-2000W-1AVDF - WW	Watches	1
2Tone Gold Silver Cable Band Ladies Bangle Cuff Watch	Watches	3
Bulova Men's 98B143 Precisionist Charcoal Grey Dial Bracelet Watch	Watches	5
Casio - G-Shock - Gulfmaster - Black - GWN1000C-1A	Watches	5
Invicta Men's 3329 Force Collection Lefty Watch	Watches	5
Seiko Women's SUT068 Dress Solar Classic Diamond-Accented Two-Tone Stainless Steel Watch	Watches	4
Anne Klein Women's 109271MPTT Swarovski Crystal Accented Two-Tone Multi-Chain Bracelet Watch	Watches	5
Guess U13630G1 Men's day and date Gunmetal dial Gunmetal tone bracelet	Watches	5
Nixon Men's Geo Volt Sentry Stainless Steel Watch with Link Bracelet	Watches	4
Nautica Men's N14699G BFD 101 Chrono Classic Stainless Steel Watch with Brown Band	Watches	4
HDE Watch Link Pin Remover Band Strap Repair Tool Kit for Watchmakers with Pack of 3 Extra Pins	Watches	4
Timex Women's Q7B860 Padded Calfskin 8mm Black Replacement Watchband	Watches	4
Movado Men's 0606545 "Museum" Perforated Black-Rubber Strap Sport Watch	Watches	5
Invicta Men's 6674 Corduba Chronograph Black Dial Polyurethane Watch	Watches	5
Szanto Men's SZ 2001 2000 Series Classic Vintage-Inspired Stainless Steel Watch with Pebbled Leather Band	Watches	5
Casio Men's MRW200H-7EV Sport Resin Watch	Watches	4
Casio F-108WH-2AEF Mens Blue Digital Watch	Watches	3
August Steiner Men's AS8160TTG Silver And Gold Swiss Quartz Watch with Black Dial and Two Tone Bracelet	Watches	5
Invicta Men's 8928OB Pro Diver Gold Stainless Steel Two-Tone Automatic Watch	Watches	5
BOS Men's Automatic self-wind mechanical Pointer Skeleton Watch Black Dial Stainless Steel Band 9008	Watches	3
Luminox Men's 3081 Evo Navy SEAL Chronograph Watch	Watches	5
INFANTRY Mens 50mm Big Face Military Tactical Analog Digital Sport Wrist Watch Black Silicone Band	Watches	5
BUREI Dress Women's Minimalist Wrist Watches with Date Analog Quartz Stainless Steel and Ultra Slim Dial	Watches	5
Motorola Moto 360 Modern Timepiece Smart Watch - Black Leather 00418NARTL	Watches	5
Domire Fashion Accessories Trial Order New Quartz Fashion Weave Wrap Around Leather Bracelet Lady Woman Butterfly Wrist Watch	Watches	5
Casio Women's LQ139B-1B Classic Round Analog Watch	Watches	3

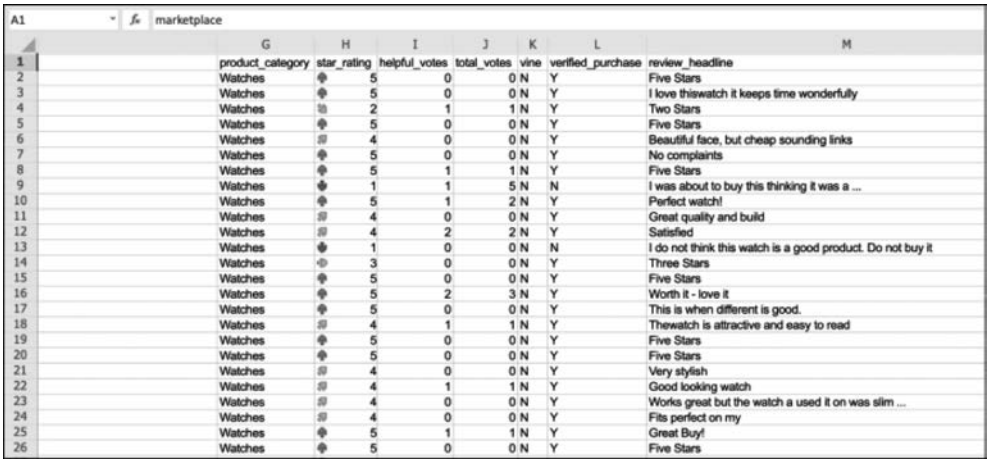
图 3-5　使用 ColorScale 创建色阶

IconSet 可以依据单元格的值来添加相应的图标，代码如下所示，只需要指定图标集合的类别和相应值的范围，就可以直接应用到表格上。完整的图标列表可以在 openpyxl 的官方文档中找到。

```
>>> from openpyxl.formatting.rule import IconSetRule

>>> icon_set_rule = IconSetRule("5Arrows", "num", [1, 2, 3, 4, 5])
>>> sheet.conditional_formatting.add("H2:H100", icon_set_rule)
>>> workbook.save("sample_conditional_formatting_icon_set.xlsx")
```

效果如图 3-6 所示。



A1 marketplace

	G	H	I	J	K	L	M
1	product_category	star_rating	helpful_votes	total_votes	vine	verified_purchase	review_headline
2	Watches	5	0	0	N	Y	Five Stars
3	Watches	5	0	0	N	Y	I love thiswatch it keeps time wonderfully
4	Watches	2	1	1	N	Y	Two Stars
5	Watches	5	0	0	N	Y	Five Stars
6	Watches	4	0	0	N	Y	Beautiful face, but cheap sounding links
7	Watches	5	0	0	N	Y	No complaints
8	Watches	5	1	1	N	Y	Five Stars
9	Watches	1	1	5	N	N	I was about to buy this thinking it was a ...
10	Watches	5	1	2	N	Y	Perfect watch!
11	Watches	4	0	0	N	Y	Great quality and build
12	Watches	4	2	2	N	Y	Satisfied
13	Watches	1	0	0	N	N	I do not think this watch is a good product. Do not buy it
14	Watches	3	0	0	N	Y	Three Stars
15	Watches	5	0	0	N	Y	Five Stars
16	Watches	5	2	3	N	Y	Worth it - love it
17	Watches	5	0	0	N	Y	This is when different is good.
18	Watches	4	1	1	N	Y	Thewatch is attractive and easy to read
19	Watches	5	0	0	N	Y	Five Stars
20	Watches	5	0	0	N	Y	Five Stars
21	Watches	4	0	0	N	Y	Very stylish
22	Watches	4	1	1	N	Y	Good looking watch
23	Watches	4	0	0	N	Y	Works great but the watch a used it on was slim ...
24	Watches	4	0	0	N	Y	Fits perfect on my
25	Watches	5	1	1	N	Y	Great Buy!
26	Watches	5	0	0	N	Y	Five Stars

图 3-6　添加了图标的表格

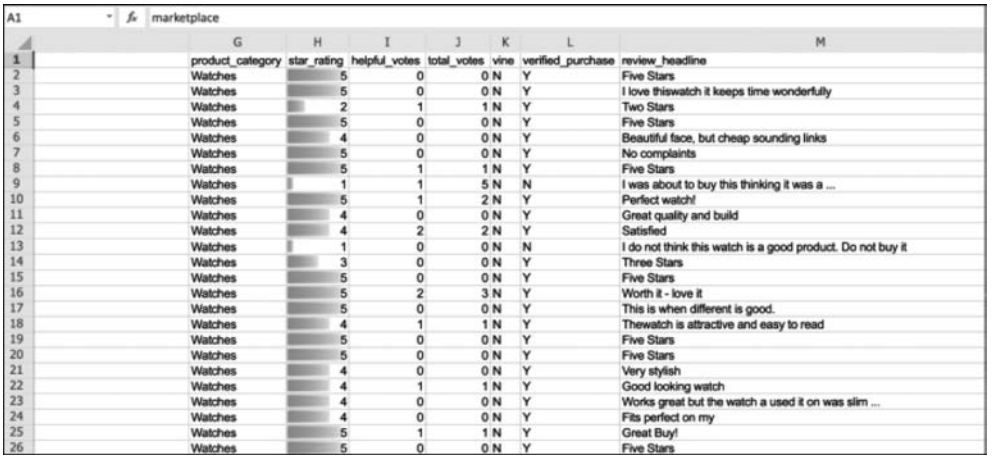
DataBar 允许在单元格中添加类似进度条一样的条带，直观地展示数值的大小，使用方式如下所示。

```
>>> from openpyxl.formatting.rule import DataBarRule

>>> data_bar_rule = DataBarRule(start_type = "num",
...                             start_value = 1,
...                             end_type = "num",
...                             end_value = "5",
...                             color = colors.GREEN)
>>> sheet.conditional_formatting.add("H2:H100", data_bar_rule)
>>> workbook.save("sample_conditional_formatting_data_bar.xlsx")
```

只需要指定规则的最大值和最小值，以及希望显示的颜色，就可以直接使用了。代码执行后的效果如图 3-7 所示。

使用条件格式可以实现很多功能，这里限于篇幅只展示了一部分样例，读者可以查阅 openpyxl 的文档获得更多的信息。



A1 marketplace

	G	H	I	J	K	L	M
1	product_category	star_rating	helpful_votes	total_votes	vine	verified_purchase	review_headline
2	Watches	5	0	0	N	Y	Five Stars
3	Watches	5	0	0	N	Y	I love thiswatch it keeps time wonderfully
4	Watches	2	1	1	N	Y	Two Stars
5	Watches	5	0	0	N	Y	Five Stars
6	Watches	4	0	0	N	Y	Beautiful face, but cheap sounding links
7	Watches	5	0	0	N	Y	No complaints
8	Watches	5	1	1	N	Y	Five Stars
9	Watches	1	1	5	N	N	I was about to buy this thinking it was a ...
10	Watches	5	1	2	N	Y	Perfect watch!
11	Watches	4	0	0	N	Y	Great quality and build
12	Watches	4	2	2	N	Y	Satisfied
13	Watches	1	0	0	N	N	I do not think this watch is a good product. Do not buy it
14	Watches	3	0	0	N	Y	Three Stars
15	Watches	5	0	0	N	Y	Five Stars
16	Watches	5	2	3	N	Y	Worth it - love it
17	Watches	5	0	0	N	Y	This is when different is good.
18	Watches	4	1	1	N	Y	Thewatch is attractive and easy to read
19	Watches	5	0	0	N	Y	Five Stars
20	Watches	5	0	0	N	Y	Five Stars
21	Watches	4	0	0	N	Y	Very stylish
22	Watches	4	1	1	N	Y	Good looking watch
23	Watches	4	0	0	N	Y	Works great but the watch a used it on was slim ...
24	Watches	4	0	0	N	Y	Fits perfect on my
25	Watches	5	1	1	N	Y	Great Buy!
26	Watches	5	0	0	N	Y	Five Stars

图 3-7 添加了 DataBar 的表格

3.3.3 为 Excel 表单添加图表

Excel 表单可以生成具有表现力的数据图表，包括柱状图、饼图、折线图等，使用 openpyxl 一样可以实现对应的功能。

在展示如何添加图表之前，需要先构建一组数据作为实例，代码如下所示。

```
from openpyxl import Workbook
from openpyxl.chart import BarChart, Reference

workbook = Workbook()
sheet = workbook.active

rows = [
    ["Product", "Online", "Store"],
    [1, 30, 45],
    [2, 40, 30],
    [3, 40, 25],
    [4, 50, 30],
    [5, 30, 25],
    [6, 25, 35],
    [7, 20, 40],
]

for row in rows:
    sheet.append(row)
```

接下来，就可以通过 BarChart 类对象来为表格添加柱状图，我们希望柱状图展示每类商品的总销量，代码如下所示。

```
chart = BarChart()
data = Reference(worksheet = sheet,
                 min_row = 1,
                 max_row = 8,
                 min_col = 2,
                 max_col = 3)

chart.add_data(data, titles_from_data = True)
sheet.add_chart(chart, "E2")

workbook.save("chart.xlsx")
```

简洁的柱状图已经生成，并且插入了表格，如图 3-8 所示。

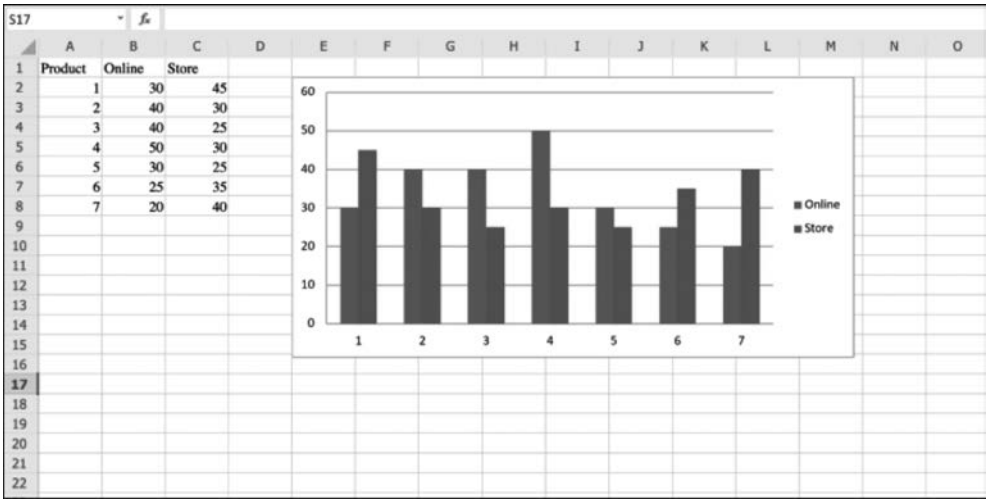


图 3-8　插入了柱状图的表格

插入图表的左上角将和代码指定的单元格对齐，样例将图表对齐在了 E2 处。

如果想绘制一个折线图，可以简单修改代码，然后使用 LineChart 类，代码如下所示。

```
import random
from openpyxl import Workbook
from openpyxl.chart import LineChart, Reference

workbook = Workbook()
sheet = workbook.active

# 创建一些示例销售数据
rows = [
    ["", "January", "February", "March", "April",
    "May", "June", "July", "August", "September",
    "October", "November", "December"],
```

```
    [1, ],
    [2, ],
    [3, ],
]

for row in rows:
    sheet.append(row)

for row in sheet.iter_rows(min_row = 2,
                       max_row = 4,
                       min_col = 2,
                       max_col = 13):
    for cell in row:
        cell.value = random.randrange(5, 100)

chart = LineChart()
data = Reference(worksheet = sheet,
             min_row = 2,
             max_row = 4,
             min_col = 1,
             max_col = 13)

chart.add_data(data, from_rows = True, titles_from_data = True)
sheet.add_chart(chart, "C6")

workbook.save("line_chart.xlsx")
```

效果如图 3-9 所示。

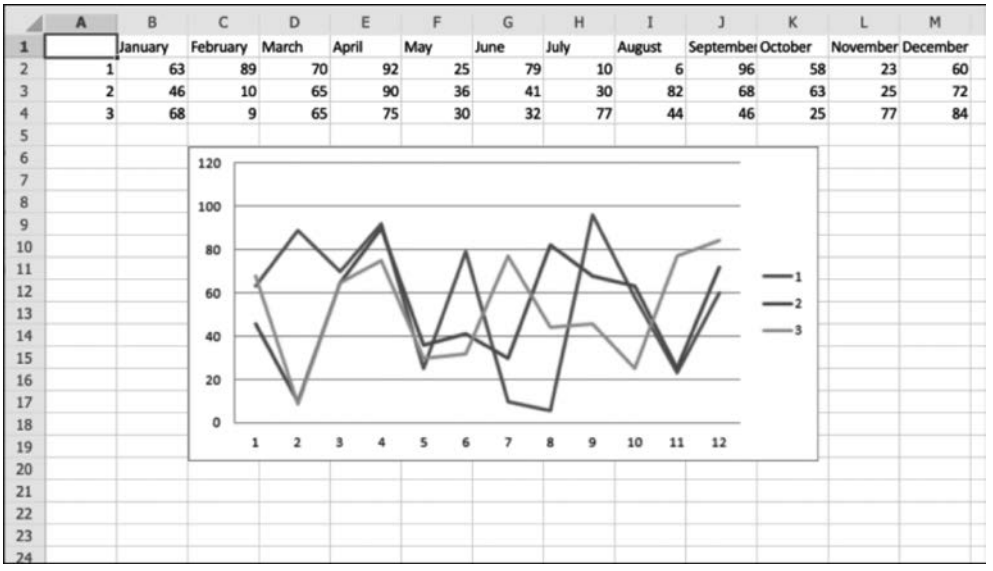


图 3-9　添加了折线图的表格

3.4 数据分析实例

3.4.1 背景与前期准备

本实例中使用的数据为 Consumer Reviews of Amazon Dataset 中的一部分，读者可以在随书的资料中找到名为 Consumer_Reviews_of_Amazon.xlsx 的文件。Consumer Reviews of Amazon Dataset 有超过 34 000 条针对 Amazon 产品（如 Kindle、Fire TV Stick 等）的消费者评论，以及 Datafiniti 产品数据库提供的更多评论。数据集中包括基本产品信息、评分、评论文本等相关信息。本节提供的数据截取了数据集中的一部分，完整的数据集可从 Datafiniti 的网站获得。

通过这些数据，读者可以了解亚马逊的消费电子产品销售情况，分析每次交易中消费者的评论，甚至可以进一步构建机器学习模型对产品的销售情况进行预测，如：

（1）最受欢迎的亚马逊产品是什么？

（2）每个产品的初始和当前顾客评论数量是多少？

（3）产品发布后的前 90 天内的评论与产品价格相比如何？

（4）产品发布后的前 90 天内的评论与整个销售周期相比如何？

将评论文本中的关键字与评论评分相对应来训练情感分类模型。

本节主要聚焦于数据的可视化分析，展示如何使用 openpyxl 读取数据，如何与 Pandas、Matplotlib 等工具交互，以及如何将其他工具生成的可视化结果重新导回到 Excel 中。

首先新建一个工作目录，并将 Consumer_Reviews_of_Amazon.xlsx 复制到当前的工作目录下，然后使用如下命令安装额外的环境依赖。

```
pip install numpy matplotlib sklearn pandas Pillow
```

3.4.2 使用 openpyxl 读取数据并转为 DataFrame

代码如下所示。

```
1   import pandas as pd
2   from openpyxl import load_workbook
3
4   workbook = load_workbook(filename = "Consumer_Reviews_of_Amazon.xlsx")
5   sheet = workbook.active
6
7   data = sheet.values
8
9   # 将第一行作为 DataFrame 结构的第一列
10  cols = next(data)
11  data = list(data)
12
13  df = pd.DataFrame(data, columns = cols)
```

在第 4 行中，加载准备好的文件。在第 5 行中，获得默认工作表 sheet。在第 7 行中，通过 sheet 的 values 属性提取工作表中所有的数据。在第 10 行中，将 data 的第一行单独取出，作为 Pandas 中 DataFrame 的列名，然后在 11 行中将 data 生成器转化为 Python List(注意，这里的 Python List 中不包含原工作表中的第一行，请读者自行思考原因)。最后，在第 13 行中将数据转化为 DataFrame，留作下一步使用。

3.4.3 绘制数值列直方图

得到待分析的数据后，通常要做的第一步就是统计各列的数值分布，使用直方图直观地展示出来。下面将自定义一个较为通用的直方图绘制函数，这个函数使用直方图将表中所有数值可枚举(1～50 种)的列展示出来，代码如下所示。

```
1   from mpl_toolkits.mplot3d import Axes3D
2   from sklearn.preprocessing import StandardScaler
3   import matplotlib.pyplot as plt          # 绘制
4   import numpy as np                       # 线性代数
5   import os                                # 访问目录结构
6
7   # 列数据的柱形分布图
8   def plotPerColumnDistribution(df, nGraphShown, nGraphPerRow):
9       nunique = df.nunique()
10      df = df[[col for col in df if nunique[col] > 1 and nunique[col] < 50]]
    # 为了显示,选择具有 1～50 个唯一值的列
11      nRow, nCol = df.shape
12      columnNames = list(df)
13      nGraphRow = (nCol + nGraphPerRow - 1) / nGraphPerRow
14      plt.figure(num = None, figsize = (6 * nGraphPerRow, 8 * nGraphRow), dpi = 80,
    facecolor = 'w', edgecolor = 'k')
15      for i in range(min(nCol, nGraphShown)):
16        plt.subplot(nGraphRow, nGraphPerRow, i + 1)
17        columnDf = df.iloc[:, i]
18        if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):
19          valueCounts = columnDf.value_counts()
20          valueCounts.plot.bar()
21        else:
22          columnDf.hist()
23        plt.ylabel('counts')
24        plt.xticks(rotation = 90)
25        plt.title(f'{columnNames[i]} (column {i})')
26      plt.tight_layout(pad = 1.0, w_pad = 1.0, h_pad = 1.0)
27      plt.show()
28      plt.savefig('./ColumnDistribution.png')
29
30  plotPerColumnDistribution(df, 10, 5)
```

plotPerColumnDistribution 函数接受 3 个参数：df 为 DataFrame 数据集；nGraphShown 为图总数的上限；nGraphPerRow 为每行的图片数。在第 9 行中使用 Pandas 的 nunique 方法获得每一列的不重复值的总数。在第 10 行中将不重复值总数为 1～50 的列保留，其余剔除。第 11～14 行计算总行数，并设置 Matplotlib 的画布尺寸和排布。从第 15 行开

始依次绘制每个子图。绘制过程中需要区分值的类型，如果该列不是数值类型，则需要对各种值的出现数量进行统计，并通过 plot.bar()方法绘制到画布上（第 18～20 行）；如果该列是数值类型，则只需要调用 hist()函数即可完成绘制（第 22 行）。在第 23～25 行中设置图题以及坐标轴标签。在第 26 行和第 27 行中调整布局后即可通过 plt.show()查看绘制结果，如图 3-10 所示。

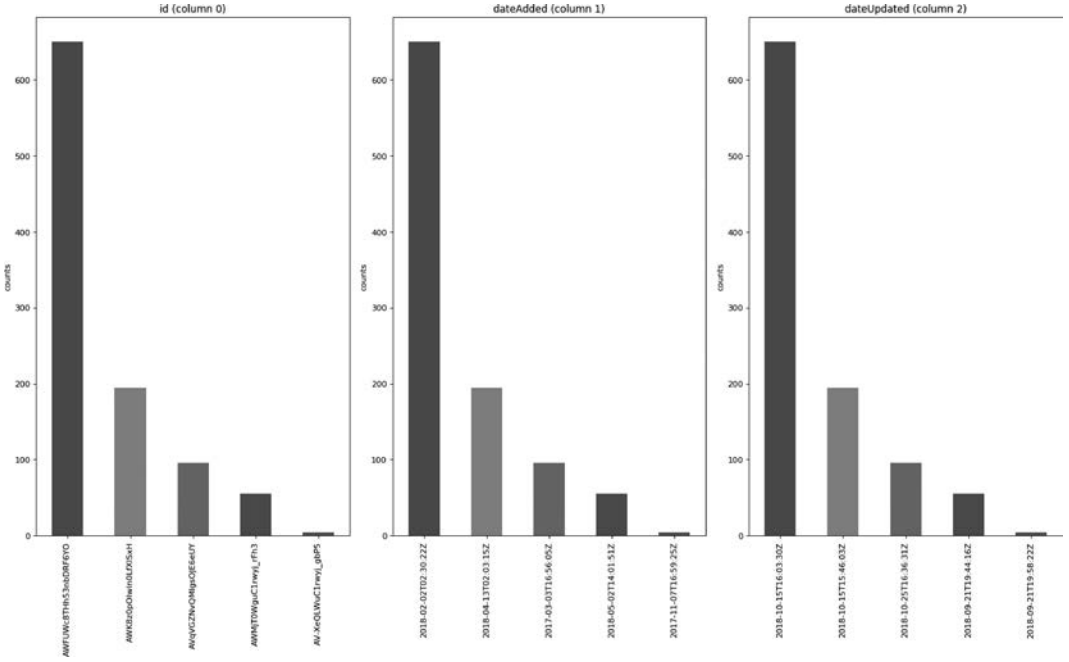


图 3-10　ColumnDistribution

3.4.4　绘制相关性矩阵

相关性矩阵是表示变量之间的相关系数的表。表格中的每个单元格均显示两个变量之间的相关性。通常在进行数据建模之前需要计算相关性矩阵，主要原因有以下 3 个。

（1）通过相关性矩阵图表，可以较为清晰直观地看出数据中的隐藏特征。

（2）相关性矩阵可以作为其他分析的输入特征。例如，使用相关矩阵作为探索性因素分析、确认性因素分析、结构方程模型的输入，或者在线性回归时用来成对排除缺失值。

（3）作为检查其他分析结果时的诊断因素。例如，对于线性回归，变量间相关性过高则表明线性回归的估计值是不可靠的。

同样地，本节将会定义一个较为通用的相关性矩阵构建函数，代码如下所示。

```
1   def plotCorrelationMatrix(df, graphWidth):
2       filename = df.dataframeName
3       df = df.dropna('columns')                                  # 去除值为 NaN 的列
4       df = df[[col for col in df if df[col].nunique() > 1]] # 保留超过 1 个唯一值的列
5       if df.shape[1] < 2:
```

```
6            print(f'No correlation plots shown: The number of non-NaN or constant columns
  ({df.shape[1]}) is less than 2')
7            return
8        corr = df.corr()
9        plt.figure(num=None, figsize=(graphWidth, graphWidth), dpi=80,
  facecolor='w', edgecolor='k')
10       corrMat = plt.matshow(corr, fignum = 1)
11       plt.xticks(range(len(corr.columns)), corr.columns, rotation=90)
12       plt.yticks(range(len(corr.columns)), corr.columns)
13       plt.gca().xaxis.tick_bottom()
14       plt.colorbar(corrMat)
15       plt.title(f'Correlation Matrix for {filename}', fontsize=15)
16       plt.show()
17       plt.savefig('./CorrelationMatrix.png')
18
19  df.dataframeName = 'CRA'
20  plotCorrelationMatrix(df, 8)
```

在第 2 行中获得当前表名(注意,手动构建的 Dataframe 需要手动指定 dataframeName,见第 19 行)。在第 3 行中将表中的空值全部丢弃。在第 4 行中将所有值都相同的列全部丢弃。这时,如果列数小于 2,则无法进行相关性分析,打印警告并直接返回。在第 8 行中通过 corr()方法获得相关性矩阵的原始数据。在第 10～17 行中设置画布并绘制,最终的效果如图 3-11 所示。

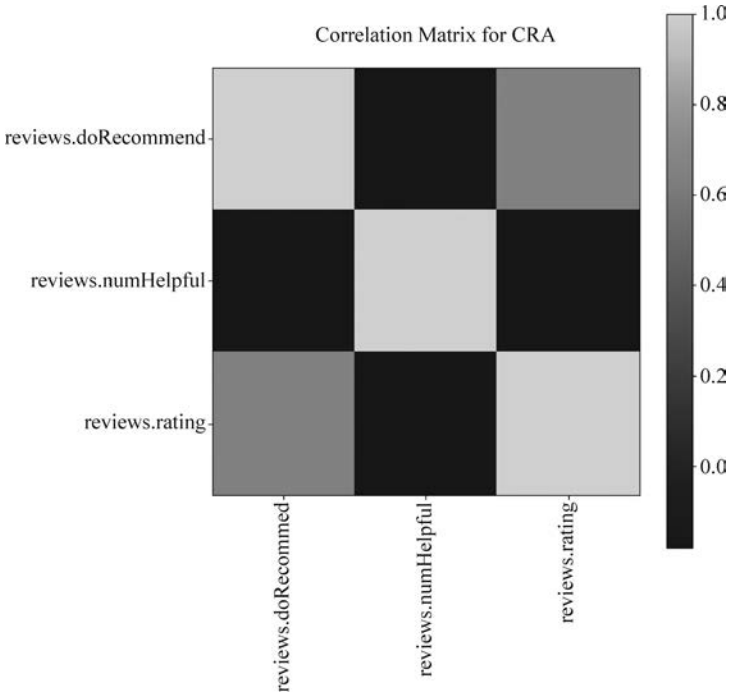


图 3-11　相关性矩阵

在图 3-11 中,颜色越浅则相关性越高。可以看到,用户是否对商品进行打分与是否进行评论的相关性很强。这表明评论与打分是两个关联极强的因素,可以进一步设计模

型根据其中一个来预测另一个。

3.4.5 绘制散布矩阵

散布矩阵(Scatterplot Matrix)又叫 scagnostic,是一种常用的高维度数据可视化技术,最初是由 John 和 Paul Turkey 提出的。它将高维度的数据的每两个变量组成一个散点图,再将它们按照一定的顺序组成散点图矩阵。通过这样的可视化方式,能够将高维度数据中所有的变量两两之间的关系展示出来。

下面将介绍如何构建一个简单的散布矩阵函数,代码如下所示。

```
1   def plotScatterMatrix(df, plotSize, textSize):
2       df = df.select_dtypes(include =[np.number])          # 保留类型为数字的列
3       # Remove rows and columns that would lead to df being singular
4       df = df.dropna('columns')
5       df = df[[col for col in df if df[col].nunique() > 1]] # 保留超过 1 个唯一值的列
6       columnNames = list(df)
7       if len(columnNames) > 10:                            # 减少矩阵求逆的列数
8           columnNames = columnNames[:10]
9       df = df[columnNames]
10      ax = pd.plotting.scatter_matrix(df, alpha = 0.75, figsize = [plotSize, plotSize],
    diagonal = 'kde')
11      corrs = df.corr().values
12      for i, j in zip( * plt.np.triu_indices_from(ax, k = 1)):
13          ax[i, j].annotate('Corr. coef = %.3f' % corrs[i, j], (0.8, 0.2), xycoords =
    'axes fraction', ha = 'center', va = 'center', size = textSize)
14      plt.suptitle('Scatter and Density Plot')
15      plt.show()
16      plt.savefig('./ScatterMatrix.png')
17
18  plotScatterMatrix(df, 9, 10)
```

在第 2 行中去除所有非数字类型的列。在第 4 行中将表中的空值全部丢弃。在第 5 行中将所有值都相同的列全部丢弃。第 6、7 行截取了前 10 列进行展示,这是因为如果列数过多会超出屏幕的显示范围,读者可以自行选择需要绘制的特定列。在第 10 行中通过 pd.plotting.scatter_matrix 初始化画布。在第 11 行中获取相关性系数。第 12、13 行将依次获取不同的列组合,并绘制该组合的相关性图表。在第 14～16 行中绘制并保存图片。最终的可视化结果如图 3-12 所示。

在图 3-12 中,左上和右下展示了 numHelpful 和 rating 的数据分布,可以看到绝大多数商品的 numHelpful 数量为 0,而其他数量的分布比较平均。绝大部分商品的 rating 为 5 分,20%左右的商品是 4 分,低于 4 分的数量较少。左下和右上的散点图展示了数据在交叉的两个维度上的分布,绝大部分的 numHelpful 评价都来源于打分为 5 分的商品,且分数越低,出现 numHelpful 评价的概率越小,这符合日常生活的直觉。

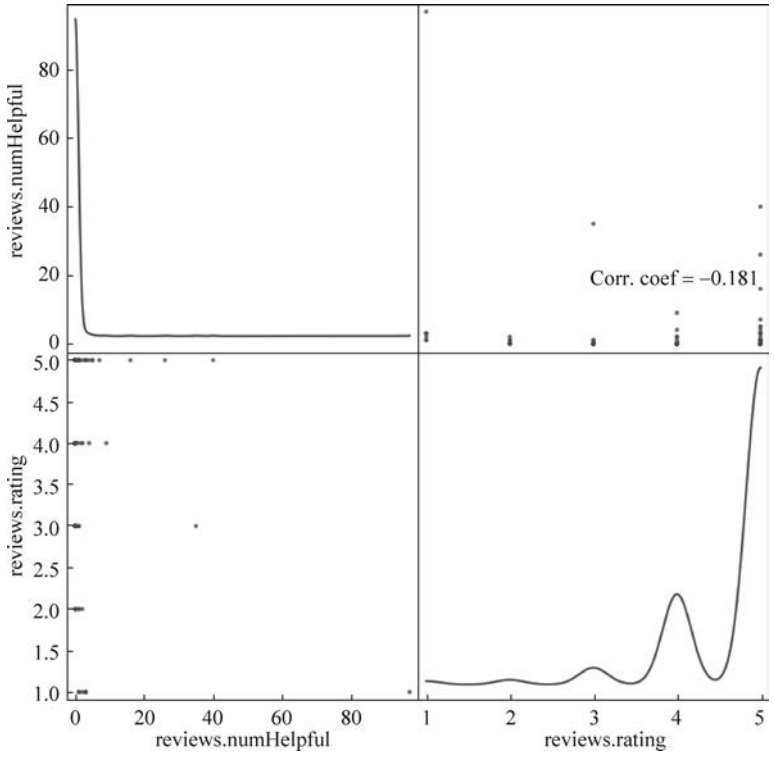


图 3-12　散布矩阵

3.4.6　将可视化结果插入 Excel 表格

前面的可视化图表都以 PNG 图片格式存储在工作路径中，下面介绍如何将图片插入 Excel 工作簿，代码如下所示。

```
from openpyxl import Workbook
from openpyxl.drawing.image import Image

workbook = Workbook()
sheet = workbook.active

vis = Image("ScatterMatrix.png")

# 改变形状，避免 logo 占据整个表格
vis.height = 600
vis.width = 600

sheet.add_image(vis, "A1")
workbook.save(filename = "visualization.xlsx")
```

在上述代码中，首先创建了一个新的工作簿，而后通过 openpyxl 的 Image 模块加载了已经预先生成的 ScatterMatrix.png。在调整了图片的大小后，将其插入 A1 单元格，最后保存工作簿。流程十分清晰简单，最终的效果如图 3-13 所示。

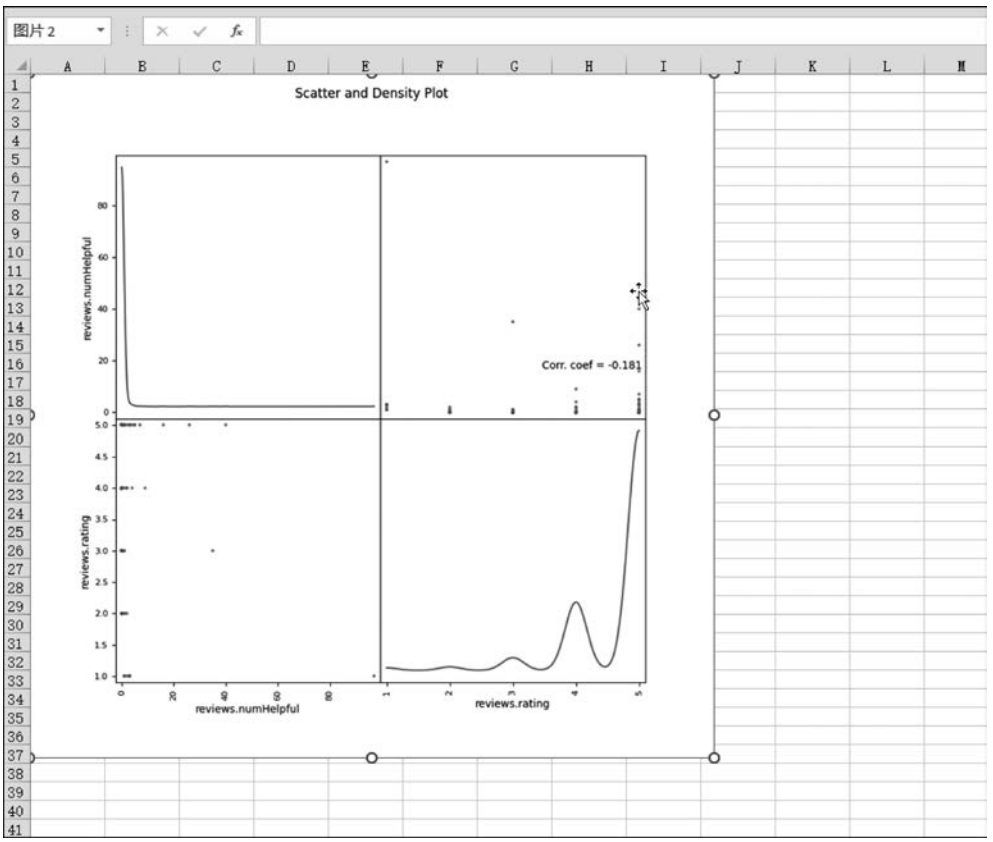


图 3-13　visualization.xlsx