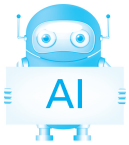
第 3 章



数据可视化

学习目标：数据可视化是将原始数据映射为图形、颜色与位置等视觉元素，使人类一眼即可捕捉模式、趋势与异常，从而高效完成探索、沟通与决策。Matplotlib 是 Python 的 2D 绘图库，Seaborn 是 Python 的数据可视化库，满足数据分析 90%的绘图需求。本章重点介绍 Matplotlib 安装、绘图步骤、绘图设置，包括颜色、标记和线类型、对图标进行刻度、标签和图例等，绘制各种图等。Seaborn 的安装、数据集、绘图设置以及绘图种类。

数据可视化教学应融入科技报国理念：图形映射不仅是技术，更是传递真相、服务决策的责任，引导学生恪守真实、完整、客观的学术诚信，拒绝“美颜”数据、误导公众；掌握 Matplotlib、Seaborn 等开源工具，树立国产替代与自主可控意识，立志贡献中国方案；以美呈现数据，以善沟通结果，培育胸怀家国、德技并修的新时代数据人才。

3.1 数据可视化概述

3.1.1 定义

数据可视化是将数据以图形、图表、地图等视觉形式表现出来的过程，旨在帮助人们更直观地理解数据中的模式、趋势和异常。它是数据分析、商业智能、科研、金融、医疗等多个领域的重要工具。

数据可视化的核心目的如表 3.1 所示。

表 3.1 数据可视化的核心目的

目　　的	说　　明
简化复杂性	将大量复杂数据转换为易懂的图形
发现趋势	识别时间序列中的上升、下降或周期性变化
识别异常	快速发现离群点或异常值
支持决策	为管理层提供直观依据，辅助战略制定
讲故事	用数据讲述一个清晰、有说服力的故事

3.1.2　两支“画笔”

Matplotlib 是 Python 最核心的静态绘图引擎，以面向对象或 MATLAB 式接口，把数值数据映射成 2D/3D 图形，支持从折线、柱状到等高线、动画等所有常见图表，并可精细控制坐标轴、文本、图例和样式，是任何高级可视化库的底层基础。

Seaborn 是在 Matplotlib 之上的高级绘图库，一键生成高颜值的箱线图、小提琴图、回归线、热力图等。

Matplotlib 和 Seaborn 对比如表 3.2 所示。

表 3.2　Matplotlib 和 Seaborn 对比

维　　度	Matplotlib	Seaborn
层级	最底层	高层封装
代码量	多(5～10 行)	少(1～3 行)
默认外观	朴素	好看(配色、网格、字体)
适用范围	出版级微调、3D、动画	EDA、统计图(箱线图、小提琴图、回归线)
中文/负号	手动 rcParams	继承 Matplotlib，同一套设置
与 Pandas 兼容性	只接受 NumPy 数组	原生支持 DataFrame 列名
可自定义程度	无限	有限

3.2　常见图表类型

常见图表类型如表 3.3 所示。

表 3.3　常见图表类型

大类	核心目的	常用几张图	一句话记忆口诀
趋势	随时间变	折线图、面积图、瀑布图	折面瀑布看涨跌
比较	谁大谁小	柱状图、条形图、雷达图	柱条雷达比高低
分布	数据散在哪儿	直方图、散点图、箱线图	直散箱线看分布
构成	占比多少	饼图、环形图、堆叠柱	饼环堆叠看整体

3.2.1　趋势类

1. 折线图

折线图是指用直线段依次连接各数据点，以反映数值随类别或时间变化的走势与对比关系的图表。折线图如图 3.1 所示。

折线图内容如表 3.4 所示。

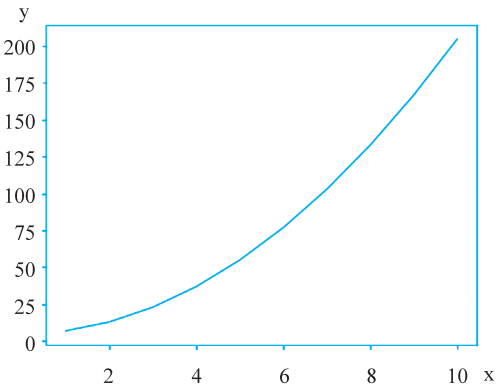


图 3.1 折线图

表 3.4 折线图

项　目	内　容
场合	看“随时间变化”的趋势、序列、走向
优势	时间趋势一眼到底
劣势	类别多于 10 条就成“意大利面”
口诀	线往上走好消息，往下走拉警报；多点交叉看拐点

2. 面积图

面积图是指用填充了颜色或纹理的折线与坐标轴围成的面积，来强调数值随类别或时间变化的总量与趋势的图表。面积图如图 3.2 所示。

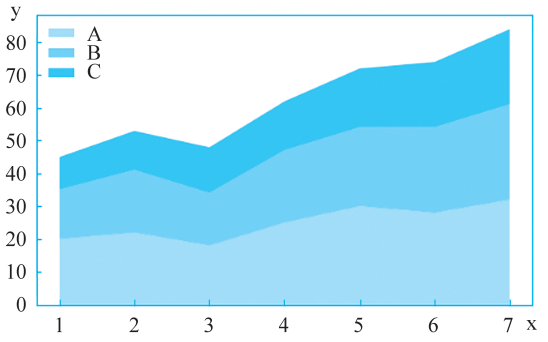


图 3.2 面积图

面积图内容如表 3.5 所示。

表 3.5 面积图

项　目	内　容
场合	看“总量”随时间变化
优势	总量趋势一眼可见

续表

项　目	内　容
劣势	系列过多层层叠，底层难精读
口诀	时序总量选面积，系列别超五；零基线，莫乱改，颜色透明层层露

3.2.2　比较类

1. 柱状图

柱状图(Bar Chart)是一种常用的数据可视化图表，用于比较不同类别之间的数值大小。它通过矩形柱子的高度或长度来表示数据的大小，柱子可以是垂直(柱状图)或水平(条形图)排列。柱状图如图 3.3 所示。

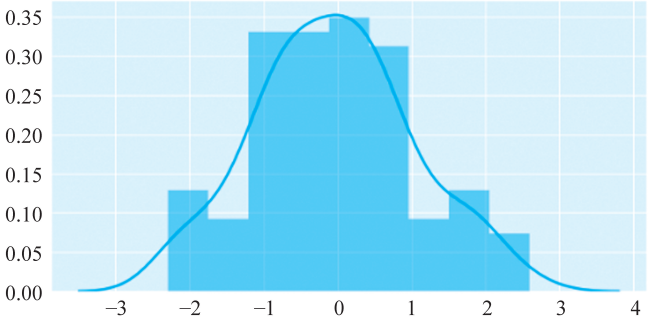


图 3.3　柱状图

柱状图内容如表 3.6 所示。

表 3.6　柱状图内容

项　目	内　容
场合	对比不同类别的数值大小
优势	谁高谁低，长度即大小，零门槛
劣势	类别＞20 根柱子挤成“条形码”；Y 轴截断会误导
口诀	柱长即量，一眼排序；横着看叫条，竖着看叫柱

2. 条形图

条形图是指用水平矩形条的长度表示各类别数值大小的对比图表。条形图如图 3.4 所示。

条形图内容如表 3.7 所示。

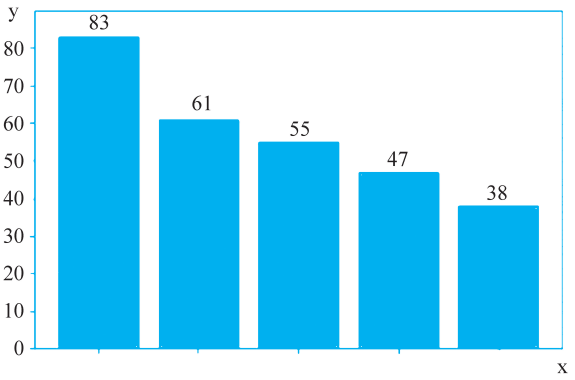


图 3.4 条形图

表 3.7 条形图内容

项　目	内　容
场合	类别对比（国家、产品、品牌等）
优势	一眼看出谁高谁低
劣势	太多条 → 视觉疲劳
口诀	类别多，先排序，零基线，莫乱移；横条长，标签亮，一眼高低全亮相

3.2.3 分布类

1. 箱线图

箱线图又称为箱形图或盒式图，不同于折线图、柱状图或饼图等传统图表只是数据大小、占比、趋势的呈现，箱线图包含统计学的均值、分位数、极值等统计量，用于分析不同类别数据平均水平差异，展示属性与中位数离散速度，并揭示数据间离散程度、异常值、分布差异等。箱线图是一种基于“五位数”摘要显示数据分布的标准化方法。

箱形图判断异常值的标准以四分位数和四分位距为基础，当数据在箱线图中超过上四分位 1.5 倍四分位距或下四分位 1.5 倍距离时，即小于 Q1－1.5IQR 或大于 Q3＋1.5IQR 的值时被认为是异常值。箱线图如图 3.5 所示。

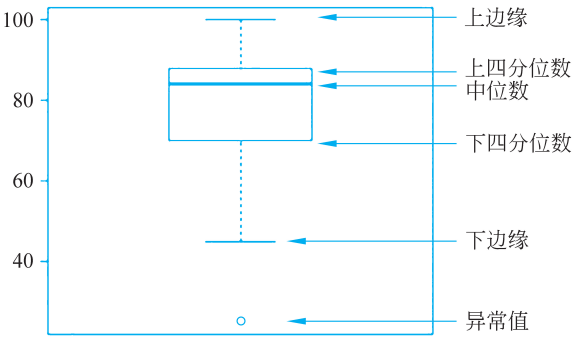


图 3.5 箱线图

图中元素解释如表 3.8 所示。

表 3.8　箱线图元素

元　素	含　义	计 算 方 式
箱体下沿(Q1)	第 25%分位数	np.percentile(x, 25)
箱体中线(Q2)	中位数	np.percentile(x, 50)
箱体上沿(Q3)	第 75%分位数	np.percentile(x, 75)
IQR	四分位距	IQR = Q3−Q1
上须末端	非异常最大值	min(Q3 + 1.5IQR, 数据最大值)
下须末端	非异常最小值	max(Q1−1.5IQR, 数据最小值)
离群点(Outliers)	超出须的点	单独描红

箱线图内容如表 3.9 所示。

表 3.9　箱线图内容

项　目	内　容
场合	多组**连续数据**“中心-离散-偏态-异常”四连体检
优势	五数概括＋异常值一张图搞定，不假定分布
劣势	看不出双峰、多峰；类别太多胡须打架
口诀	箱体胖则波动大，中线贴底右偏；点出须即异常

2. 散点图

散点图是用二维坐标系中点的位置展示两个变量之间关系与分布形态的图表。散点图如图 3.6 所示。

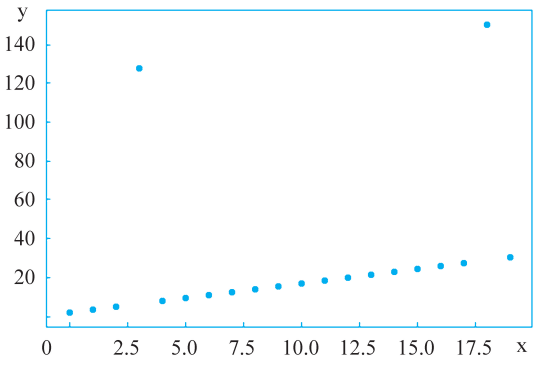


图 3.6　散点图

散点图内容如表 3.10 所示。

表 3.10 散点图内容

项　目	内　容
场合	找**两连续变量**关系：相关、聚类、离群
优势	相关强弱＋离群点一次性暴露
劣势	点太多时会糊成“黑毯”；只能同时看两个变量（第三个变量用颜色/大小）
口诀	点云斜向上正相关，向下负相关；远离团簇是异常

3.2.4 构成类

1. 饼图

饼图是用圆形扇形面积占比展示各部分占总体比例的图表。饼图如图 3.7 所示。饼图内容如表 3.11 所示。

表 3.11 饼图内容

项　目	内　容
场合	只关心**整体内占比**，类别≤5 片
优势	百分比直觉，像切蛋糕
劣势	人眼对角度不敏感；＞5 片或差异＜5%时基本看不清
口诀	饼图不过五，角度难比大小；把 25%当直角，其余靠估

2. 环形图

环形图是用同心圆环的弧长占比展示各部分占总体比例的图表。环形图如图 3.8 所示。

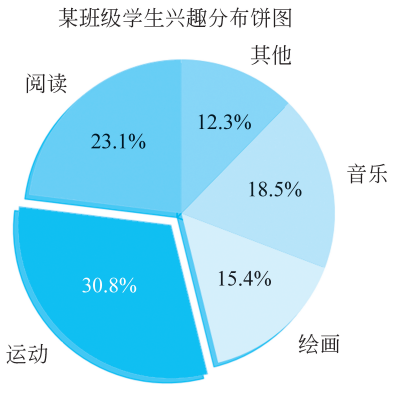


图 3.7 饼图

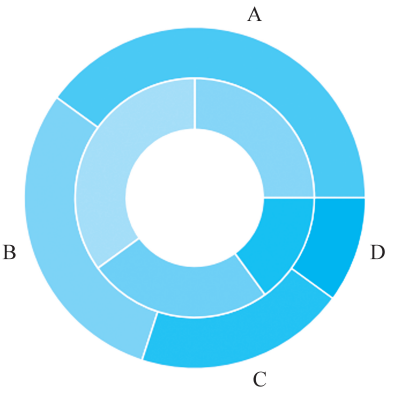


图 3.8 环形图

环形图内容如表 3.12 所示。

表 3.12　环形图内容

项　目	内　容
场合	展示整体与构成比例(仅一个总体)
优势	中心留白可放总计或标题,一眼看整体
劣势	分类过多→角度差异难辨
口诀	分类少,比例昭,中心留白写总指标;环莫多,角莫小,一眼看懂才最好

3.3　Matplotlib

3.3.1　认识 Matplotlib

Matplotlib 诞生于 2003 年,名字中的“Mat”取自 MATLAB(高性能数值计算与可视化软件),“Plot”意为绘图,“Lib”代表函数库集合。Matplotlib 主要用于将 NumPy 统计计算结果可视化,作为 Python 可视化包的鼻祖,是最常用的标准可视化库,功能强大且复杂。

使用 pip install matplotlib 命令进行安装,如图 3.9 所示。

```
(base) C:\Users\Administrator>pip install matplotlib
Requirement already satisfied: matplotlib in c:\programdata\anaconda3\lib\site-p
ackages
Requirement already satisfied: numpy>=1.7.1 in c:\programdata\anaconda3\lib\site
-packages (from matplotlib)
Requirement already satisfied: six>=1.10 in c:\programdata\anaconda3\lib\site-pa
ckages (from matplotlib)
Requirement already satisfied: python-dateutil>=2.1 in c:\programdata\anaconda3\
lib\site-packages (from matplotlib)
Requirement already satisfied: pytz in c:\programdata\anaconda3\lib\site-package
s (from matplotlib)
Requirement already satisfied: cycler>=0.10 in c:\programdata\anaconda3\lib\site
-packages (from matplotlib)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in c:\pr
ogramdata\anaconda3\lib\site-packages (from matplotlib)
You are using pip version 9.0.3, however version 10.0.0 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' comm
and.
```

图 3.9　Matplotlib 安装

3.3.2　图表基本结构

Matplotlib 生成的图形主要由以下几个部分构成,如图 3.10 所示。

- figure:定义图像窗口,相当于画布。
- axes:绘制 2D 图像的实际区域,又称为轴域区或者绘图区。
- axis:指坐标系中的垂直轴与水平轴,包含轴的长度大小、轴标签(指 x 轴、y 轴)和刻度标签。

Matplotlib 的 pyplot 模块中的 figure 方法用于创建图像,语法如下。

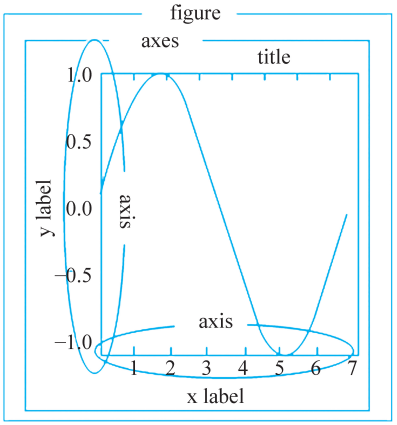


图 3.10　Matplotlib 图像组成

```
from matplotlib import pyplot as plt
plt.figure(num, figsize, dpi, facecolor, edgecolor, clear)      #创建图形对象
```

参数如下。

- num：figure 的编号，可选。
- figsize：设置画布的尺寸，默认为[6.4，4.8]，单位为英寸。
- dpi：分辨率，默认 100。
- facecolor：背景颜色。
- edgecolor：边线颜色。
- clear：如果 num 代表的 figure 已经存在，是否将其清空。

一个画布包含一个或多个 Axes 对象，Axes 参数如表 3.13 所示。

表 3.13　Axes 参数

函数名称	描　　述	函数名称	描　　述
axes	在画布中添加轴	xticks	获取或设置 x 轴的刻标和相应标签
text	向轴上添加文本	ylabel	设置 y 轴的标签
title	设置当前轴的标题	ylim	获取或设置 y 轴的区间大小
xlabel	设置 x 轴的标签	yscale	设置 y 轴的缩放比例
xlim	获取或者设置 x 轴的区间大小	yticks	获取或设置 y 轴的刻标和相应标签
xscale	设置 x 轴的缩放比例		

plot 方法用来指定线型、标记颜色、样式以及大小，语法如下。

```
plt.plot(x, y, 'xxx', label='', linewidth='')
```

（1）点线颜色 color 取值如表 3.14 所示。

表 3.14　颜色取值

字　　符	颜　　色	字　　符	颜　　色
'b'	蓝色	'm'	品红色
'g'	绿色	'y'	黄色
'r'	红色	'k'	黑色
'c'	青色	'w'	白色

（2）点的形状 marker 标记字符如表 3.15 所示。

表 3.15　点的形状标记字符

标记字符	说　明	标记字符	说　明	标记字符	说　明
.	点标记	1	下花三角标记		星形标记
,	像素标记	2	上花三角标记	h	竖六边形标记
O	实心圈标记	3	左花三角标记	H	横六边形标记
V	倒三角标记	4	右花三角标记	+	十字标记
^	正三角标记	8	八角形标记	x	X 标记
<	左三角标记	S	实心方形标记		
>	右三角标记	P	实心五角标记		

参数如下：

- label 设置图例需要调用 plt 或子图的 legend 方法。
- linewidth 设置线的粗细。

Matplotlib 默认不支持中文显示，需要属性 rcParams 设置字体，语法如下。

```
matplotlib.rcParams['font.family']='SimHei'   #支持汉字
```

rcParams 的参数如表 3.16 所示。

表 3.16　中文字体

中文字体	说明	中文字体	说明	中文字体	说明
'SimHei'	中文黑体	'LiSu'	中文隶书	'YouYuan'	中文幼圆
'Kaiti'	中文楷体	'FangSong'	中文仿宋	'STSong'	华文宋体

Matplotlib 可以采用 plt 和 ax 两种方式绘图。plt 在画布上隐式生成一个画图区域进行绘图。而 ax 在画布上显式地选定绘图区域绘图。ax 比 plt 在图像修饰上较为方便。

【例 3.1】　Matplotlib 绘图。

```
import matplotlib
matplotlib.use('TkAgg')
import matplotlib.pyplot as plt
import numpy as np
#让中文正常显示
matplotlib.rcParams['font.family'] = 'KaiTi'
matplotlib.rcParams['axes.unicode_minus'] = False      #解决负号显示成方块的问题
#π×2 的正确写法是 2 * np.pi 或 2 * math.pi
x = np.arange(0, 2 * np.pi, 0.05)
y = np.sin(x)
fig = plt.figure()
ax = fig.add_axes([0.1, 0.1, 0.8, 0.8])          #坐标轴范围不要贴着边界，留一点边距
ax.plot(x, y)
ax.set_title('正弦曲线示例')
ax.set_xlabel('横轴值')
```

```
ax.set_ylabel('纵轴值')
#y 轴范围应该是[-1, 1],而不是[1, 1]
ax.set_xlim(0, 2 * np.pi)
ax.set_ylim(-1, 1)
plt.show()
```

程序运行结果如图 3.11 所示。

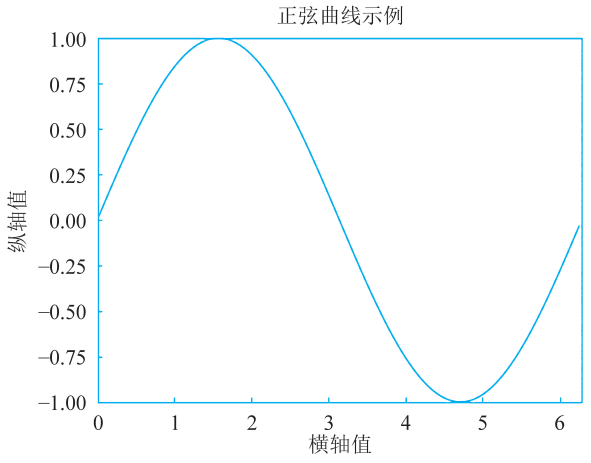


图 3.11　程序运行结果

3.3.3　子图基本操作

将多张子图显示到一个界面可以进行数据对比，Matplotlib 提供如下方法实现。

(1) plt.subplot 方法。

subplot 函数用于均等划分画布，语法如下。

```
plt.subplot(numRows, numCols, plotNum)
```

参数说明如下：整个绘图区域被分成 numRows 行和 numCols 列，按照从左到右，从上到下的顺序对每个子区域进行编号，左上的子区域的编号为 1，plotNum 参数指定创建的 Axes 对象所在的区域。例如，subplot(2,3,3)表示在当前画布的右上角创建一个两行三列的绘图区域，选择在第 3 个位置绘制子图，如图 3.12 所示。

1	2	3
4	5	6

图 3.12　子图示意图

【例 3.2】　subplot 举例。

```
import matplotlib
matplotlib.use('TkAgg')
import matplotlib.pyplot as plt
import numpy as np
fig=plt.figure(num=1,figsize=(4,4))
plt.subplot(221)                                    #画布分为 2×2 的区域
```

```
plt.plot([1,2,3,4],[1,2,3,4])
plt.subplot(222)
plt.plot([1,2,3,4],[2,2,3,4])
plt.subplot(223)
plt.plot([1,2,3,4],[1,2,2,4])
plt.subplot(224)
plt.plot([1,2,3,4],[1,2,3,3])
plt.show()
```

程序运行结果如图 3.13 所示。

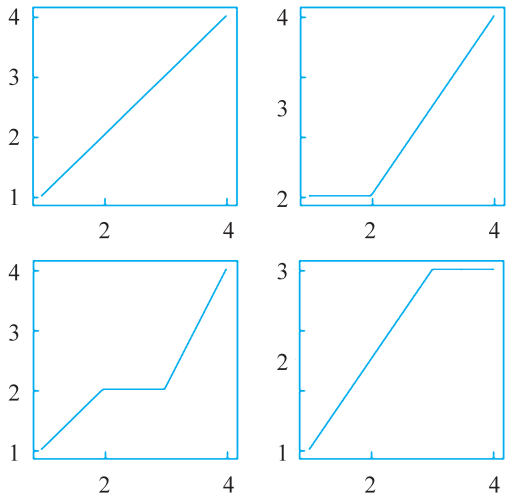


图 3.13　程序运行结果

(2) plt.subplots 方法。

subplots 使用方法和 subplot 函数类似。其不同之处在于 subplots 既创建了一个包含子图区域的画布，又创建 figure 图形对象。而 subplot 只是创建一个包含子图区域的画布。

subplots 的格式如下。

```
fig, ax = plt.subplots(nrows, ncols)
import numpy as np
import matplotlib
matplotlib.use('TkAgg')                    #强制走 Tk 图形后端,保证弹出独立窗口
from matplotlib import pyplot as plt

x = np.linspace(0, 2 * np.pi, 100)
y = np.sin(x)

fig, ax = plt.subplots()                   #创建画布 + 坐标系
ax.plot(x, y, label='sin(x)')
ax.set_title('Simple sin curve')
ax.legend()
plt.show()
```

(3) add_subplot 方法。

【例 3.3】 add_subplot 举例。

```
import matplotlib
matplotlib.use('TkAgg')
import matplotlib.pyplot as plt
import numpy as np
#让中文正常显示
fig=plt.figure(num=1,figsize=(4,4))
ax1=fig.add_subplot(221)                          #画布分为 2×2 的区域
rect = plt.Rectangle((0.1,0.2),0.3,0.6,color='r')
                                                  #创建一个矩形,参数:(x,y),width,height
ax1.add_patch(rect)                               #将形状添加到子图上
ax2=fig.add_subplot(222)
circ = plt.Circle((0.5,0.3),0.2,color='r',alpha=0.3)
#创建一个椭圆,参数:中心点,半径,默认圆形随窗口大小进行长宽压缩
ax2.add_patch(circ)                               #将形状添加到子图上
ax3=fig.add_subplot(223)
pgon = plt.Polygon([[0.2,0.2],[0.65,0.6],[0.2,0.6]])
                                                  #创建一个多边形,参数:每个顶点坐标
ax3.add_patch(pgon)                               #将形状添加到子图上
fig.canvas.draw()                                 #子图绘制
plt.show()
```

程序运行结果如图 3.14 所示。

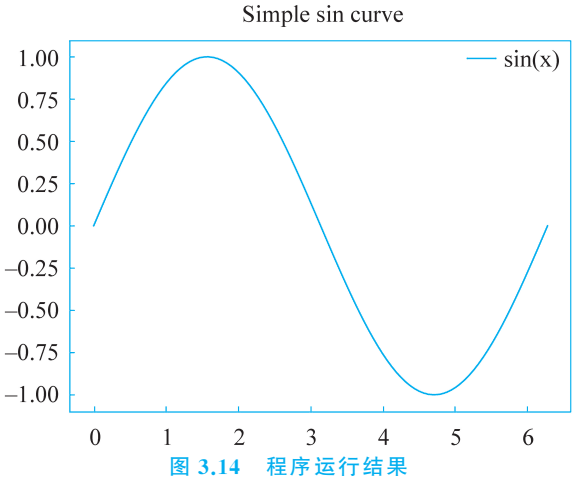


图 3.14 程序运行结果

3.3.4 二维图

pyplot 模块提供如下各类图形，如表 3.17 所示。

表 3.17 pyplot 模块提供的图形说明

函数名称	描　述	函数名称	描　述
Bar	绘制条形图	Polar	绘制极坐标图

续表

函数名称	描　　述	函数名称	描　　述
Barh	绘制水平条形图	Scatter	绘制 x 与 y 的散点图
Boxplot	绘制箱形图	Stackplot	绘制堆叠图
Hist	绘制直方图	Stem	绘制二维离散数据(“火柴图”)
his2d	绘制 2D 直方图	Step	绘制阶梯图
Pie	绘制饼图	Quiver	

1. 折线图

Matplotlib 绘制折线图的函数为 plot,常用参数及说明如表 3.18 所示。

表 3.18　plot 常用参数及说明

参数	接收值	说　　明	默认值
x,y	Array	表示 x 轴与 y 轴对应的数据	无
color	String	表示折线的颜色	None
marker	String	表示折线上数据点处的类型	None
linestyle	String	表示折线的类型	—
linewidth	数值	线条粗细	1
alpha	0～1 小数	表示点的透明度	None
label	String	数据图例内容：label=‘实际数据’	None

【例 3.4】　折线图。

```
import numpy as np
import matplotlib
matplotlib.use('TkAgg')                          #强制走 Tk 图形后端,保证弹出独立窗口
from matplotlib import pyplot as plt
x = np.arange(1, 11)
y = 2 * x * x + 5
plt.title("plot figure")
plt.xlabel("x axis")
plt.ylabel("y axis")
plt.plot(x, y)
plt.show()
```

2. 散点图

Matplotlib 提供 scatter 函数绘制散点图。

【例 3.5】　散点图。

```
import numpy as np
import matplotlib.pyplot as plt
```

```
x = np.arange(1,10)                              #产生测试数据
y = x
fig = plt.figure()
ax1 = fig.add_subplot(111)
ax1.set_title('Scatter Plot')                    #设置标题
plt.xlabel('X')                                  #设置 X 轴标签
plt.ylabel('Y')                                  #设置 Y 轴标签
ax1.scatter(x,y,c = 'r',marker = 'o')            #画散点图
plt.legend('x1')                                 #设置图标
plt.show()                                       #显示所画的图
```

3. 饼图

Matplotlib 提供 pie 函数绘制饼图。

【例 3.6】 饼图。

```
import matplotlib.pyplot as plt
import numpy as np
labels = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
data = np.random.rand(7) * 100
plt.pie(data, labels=labels, autopct='%1.1f%%')
plt.axis('equal')
plt.legend()
plt.show()
```

4. 条形图

Matplotlib 提供 bar 函数绘制条形图。

【例 3.7】 条形图。

```
import matplotlib
matplotlib.use('TkAgg')
import matplotlib.pyplot as plt
import numpy as np
plt.rcParams['font.size'] = 10                   #全局字号
plt.rcParams['axes.unicode_minus'] = False       #解决负号显示为方框
labels = ['Apple', 'Banana', 'Orange', 'Pear', 'Watermelon']
sales  = [83, 61, 55, 47, 38]
fig, ax = plt.subplots(figsize=(6, 4))
bars = ax.bar(labels, sales, color='tomato', edgecolor='k', linewidth=0.8)
#数值标签置顶
for b in bars:
    height = b.get_height()
    ax.text(b.get_x() + b.get_width()/2, height + 1,
        f'{height}', ha='center', va='bottom')
ax.set_ylabel('Sales / boxes')
ax.set_title('Fruit sales')
plt.tight_layout()
plt.show()
```

5. 箱线图

Matplotlib 提供 boxplot 函数绘制箱线图，常用参数及说明如表 3.19 所示。

表 3.19　boxplot 常用参数及说明

参　数	说　明	参　数	说　明
x	指定要绘制箱线图的数据	showcaps	是否显示箱线图顶端和末端的两条线
notch	是否以凹口的形式展现箱线图	showbox	是否显示箱线图的箱体
sym	指定异常点的形状	showfliers	是否显示异常值
vert	是否需要将箱线图垂直摆放	boxprops	设置箱体的属性，如边框色、填充色
whis	指定上下须与上下四分位的距离	labels	为箱线图添加标签
positions	指定箱线图的位置	filerprops	设置异常值的属性
widths	指定箱线图的宽度	medianprops	设置中位数的属性
patch_artist	是否填充箱体的颜色	meanprops	设置均值的属性
meanline	是否用线的形式表示均值	capprops	设置箱线图顶端和末端线条的属性
showmeans	是否显示均值	whiskerprops	设置须的属性

【例 3.8】　箱线图。

```
import matplotlib
matplotlib.use('TkAgg')
import matplotlib.pyplot as plt
import numpy as np
#1. 造数据
np.random.seed(42)
data = np.concatenate([np.random.normal(100, 10, 90),
                       np.random.normal(150, 15, 10)])
#2. 画图
fig, ax = plt.subplots(figsize=(6, 3))
bp = ax.boxplot(data, vert=False, patch_artist=True,
                boxprops=dict(facecolor='lightblue'),
                showmeans=False, widths=0.3)
#3. 关键数值
q1, q2, q3 = np.percentile(data, [25, 50, 75])
iqr = q3 - q1
whis_lo = q1 - 1.5 * iqr
whis_hi = q3 + 1.5 * iqr
#4. 标注(纯英文,位置用 y=0.15 保证在图内)
style = dict(fontsize=9, va='center', ha='center')
y_txt = 0.15
ax.text(q1, y_txt, f'Q1\n{q1:.1f}', **style)
ax.text(q2, y_txt, f'Q2\n{q2:.1f}', **style)
ax.text(q3, y_txt, f'Q3\n{q3:.1f}', **style)
#IQR 箭头
```

```
ax.annotate('', xy=(q3, 0.35), xytext=(q1, 0.35),
            arrowprops=dict(arrowstyle='<->', color='red'))
ax.text((q1 + q3)/2, 0.4, f'IQR={iqr:.1f}', color='red', **style)
#须线
ax.text(whis_lo, y_txt, f'Lower\n{whis_lo:.1f}', **style)
ax.text(whis_hi, y_txt, f'Upper\n{whis_hi:.1f}', **style)
ax.set_xlim(50, 180)
ax.set_yticks([])
ax.set_title('Boxplot Elements')
plt.tight_layout()
plt.show()
```

6. 面积图

Matplotlib 提供 stackplot 函数绘制面积图。

【例 3.9】 面积图。

```
#统一导入
import matplotlib
matplotlib.use('TkAgg')
import matplotlib.pyplot as plt
import numpy as np

plt.rcParams['font.size'] = 10
plt.rcParams['axes.unicode_minus'] = False
x = np.linspace(0, 10, 200)
y = 2 + np.sin(x) * np.exp(-x*0.2)

days = np.arange(1, 8)                          #一周 7 天
A = np.array([20, 22, 18, 25, 30, 28, 32])
B = np.array([15, 19, 16, 22, 24, 26, 29])
C = np.array([10, 12, 14, 15, 18, 20, 23])

labels = ['Product A', 'Product B', 'Product C']
colors = ['#ff7f0e', '#2ca02c', '#1f77b4']

plt.figure(figsize=(6, 4))
plt.stackplot(days, A, B, C, labels=labels, colors=colors, alpha=0.8)
plt.legend(loc='upper left')
plt.title('Stacked Area Chart')
plt.xlabel('Day')
plt.ylabel('Sales')
plt.tight_layout()
plt.show()
```

7. 环形图

Matplotlib 没有提供独立的环形图函数，一般先画饼图，再挖中心。

【例 3.10】　环形图。

```
#统一导入
import matplotlib
matplotlib.use('TkAgg')
import matplotlib.pyplot as plt
import numpy as np

plt.rcParams['font.size'] = 10
plt.rcParams['axes.unicode_minus'] = False

#内环:大类
outer_labels = ['Mobile', 'Desktop']
outer_sizes  = [60, 40]
outer_colors = ['#ff9f43', '#0984e3']

#外环:细分
inner_labels = ['iOS', 'Android', 'Win', 'macOS']
inner_sizes  = [35, 25, 22, 18]
inner_colors = ['#feca57', '#ff7675', '#74b9ff', '#a29bfe']

fig, ax = plt.subplots(figsize=(6, 6), subplot_kw=dict(aspect='equal'))

#外环(细分数)
ax.pie(inner_sizes,
       labels=inner_labels,
       colors=inner_colors,
       autopct='%1.1f%%',
       startangle=90,
       radius=1,                              #外半径
       wedgeprops=dict(width=0.3, edgecolor='w'))

#内环(大类)
ax.pie(outer_sizes,
       labels=outer_labels,
       colors=outer_colors,
       autopct='%1.1f%%',
       startangle=90,
       radius=0.7,                            #内半径
       wedgeprops=dict(width=0.3, edgecolor='w'))

#中心白圆
centre_circle = plt.Circle((0, 0), 0.4, fc='white')
fig.gca().add_artist(centre_circle)

ax.set_title('Device & OS Split Donut Chart')
plt.tight_layout()
plt.show()
```

3.3.5 三维图

1. 两种创建方式

创建三维图主要有如下两种方式：一种是利用关键字 projection='3d'来实现，另一种是通过从 mpl_toolkits.mplot3d 导入对象 Axes3D 来实现。

【例 3.11】 三维图的两种创建方式。

方法一：利用关键字。

```
from matplotlib import pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
#定义坐标轴
fig = plt.figure()
ax1 = plt.axes(projection='3d')
#ax = fig.add_subplot(111,projection='3d')       #可以绘制多个子图
```

方法二：利用三维轴。

```
from matplotlib import pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
#定义图像和三维格式坐标轴
fig=plt.figure()
ax2 = Axes3D(fig)
```

2. 三维曲线图

【例 3.12】 绘制三角螺旋线。

```
import matplotlib
matplotlib.use('TkAgg')
import matplotlib.pyplot as plt
import numpy as np
plt.rcParams['font.size'] = 10
plt.rcParams['axes.unicode_minus'] = False

#1. 创建图 + 3D 子图
fig = plt.figure(figsize=(6, 4))
ax = fig.add_subplot(111, projection='3d')
#2. 数据
zline = np.linspace(0, 15, 1000)
xline = np.sin(zline)
yline = np.cos(zline)
#3. 绘图
ax.plot3D(xline, yline, zline, 'gray')
#4. 显示
plt.show()
```

运行结果如图 3.15 所示。

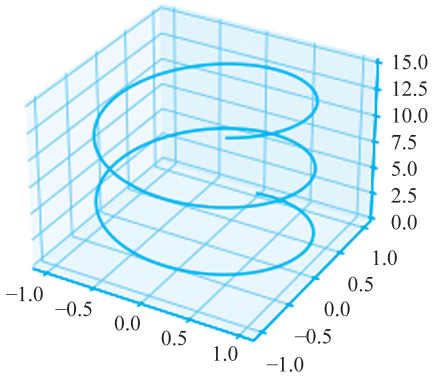


图 3.15　程序运行结果

3. 三维散点图

【例 3.13】　绘制三维散点的数据。

```
import matplotlib.pyplot as plt
import numpy as np

ax = plt.axes(projection='3d')

zdata = 15 * np.random.random(100)
xdata = np.sin(zdata) + 0.1 * np.random.randn(100)
ydata = np.cos(zdata) + 0.1 * np.random.randn(100)
ax.scatter3D(xdata, ydata, zdata, c=zdata, cmap='Reds')
```

运行结果如图 3.16 所示。

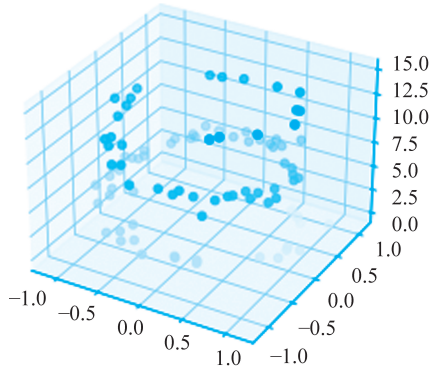


图 3.16　程序运行结果

4. 三维等高线图

【例 3.14】　绘制三维等高线图。

```
from mpl_toolkits import mplot3d
import matplotlib.pyplot as plt
import numpy as np
```