

10

Fundamentals of Visualization

聊聊可视化

主要了解 Matplotlib、Plotly 如何绘制线图



一个人可以被摧毁，但不能被打败。

A man can be destroyed but not defeated.

—— 欧内斯特·海明威 (Ernest Hemingway) | 美国、古巴记者和作家 | 1899 ~ 1961



- ◀ matplotlib.gridspec.GridSpec() 创建一个规则的子图网格布局
- ◀ matplotlib.pyplot.grid() 在当前图表中添加网格线
- ◀ matplotlib.pyplot.plot() 绘制折线图
- ◀ matplotlib.pyplot.subplot() 用于在一个图表中创建一个子图，并指定子图的位置或排列方式
- ◀ matplotlib.pyplot.subplots() 创建一个包含多个子图的图表，返回一个包含图表对象和子图对象的元组
- ◀ matplotlib.pyplot.title() 设置当前图表的标题，相当于对于特定轴 ax 对象 ax.set_title()
- ◀ matplotlib.pyplot.xlabel() 设置当前图表 x 轴的标签，相当于对于特定轴 ax 对象 ax.set_xlabel()
- ◀ matplotlib.pyplot.xlim() 设置当前图表 x 轴显示范围，相当于对于特定轴 ax 对象 ax.set_xlim()
- ◀ matplotlib.pyplot.xticks() 设置当前图表 x 轴刻度位置，相当于对于特定轴 ax 对象 ax.set_xticks()
- ◀ matplotlib.pyplot.ylabel() 设置当前图表 y 轴的标签，相当于对于特定轴 ax 对象 ax.set_ylabel()
- ◀ matplotlib.pyplot.ylim() 设置当前图表 y 轴显示范围，相当于对于特定轴 ax 对象 ax.set_ylim()
- ◀ matplotlib.pyplot.yticks() 设置当前图表 y 轴刻度位置，相当于对于特定轴 ax 对象 ax.set_yticks()
- ◀ numpy.arange() 创建一个具有指定范围、间隔和数据类型的等间隔数组
- ◀ numpy.cos() 用于计算给定弧度数组中每个元素的余弦值
- ◀ numpy.exp() 计算给定数组中每个元素的 e 的指数值
- ◀ numpy.linspace() 用于在指定的范围内创建等间隔的一维数组，可以指定数组的长度
- ◀ numpy.sin() 用于计算给定弧度数组中每个元素的正弦值
- ◀ numpy.tan() 用于计算给定弧度数组中每个元素的正切值
- ◀ plotly.express.line() 用于创建可交互的线图
- ◀ plotly.graph_objects.Scatter() 用于创建可交互的散点图、线图
- ◀ scipy.stats.norm() 创建一个正态分布对象，可用于计算概率密度、累积分布等



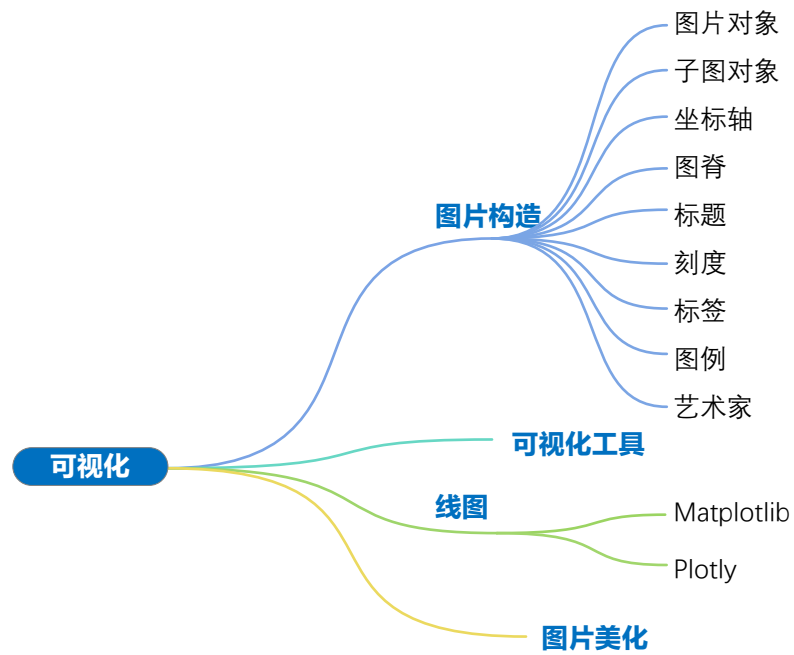
本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com



本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

10.1 解剖一幅图

本章和接下来两章介绍如何实现鸢尾花书中最常见的可视化方案。这三章内容本着“够《编程不难》用就好”为原则，不会特别深究某个具体可视化方案中的呈现细节，也不会探究其他高阶的可视化方案。



鸢尾花书《可视之美》专注提供可视化的“家常菜菜谱”。

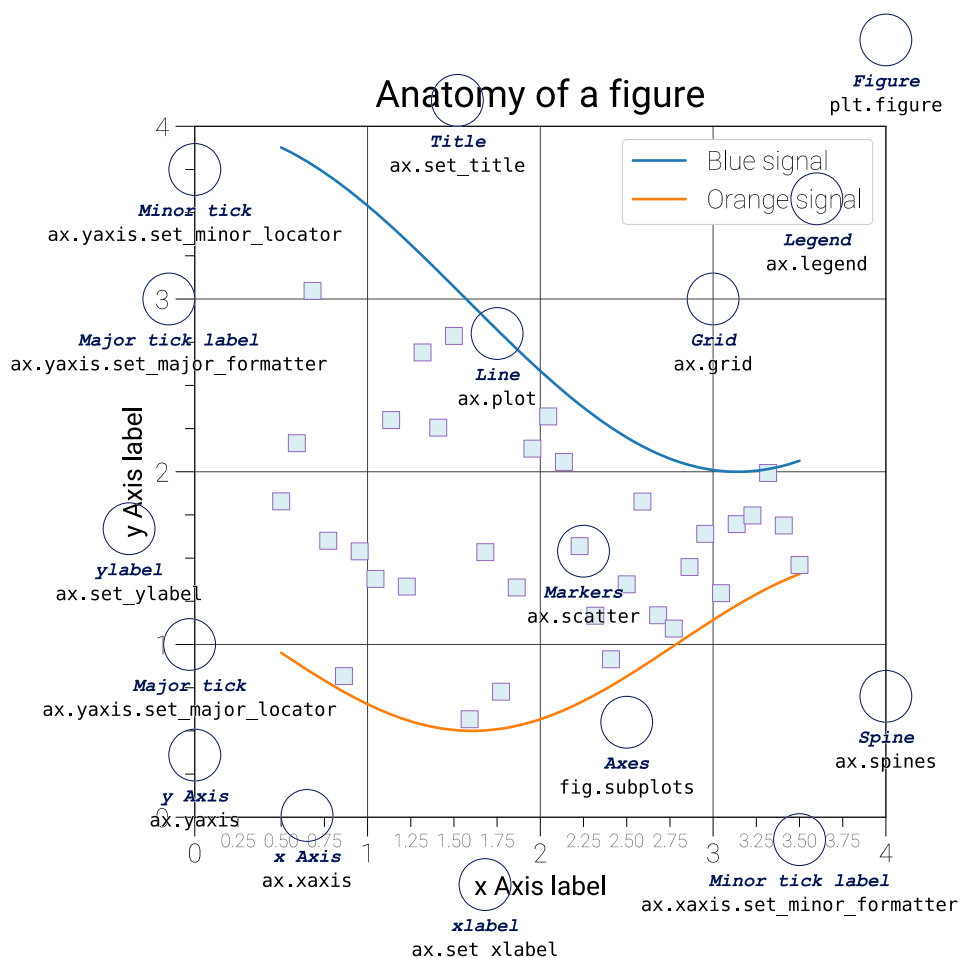


图 1. 解剖一幅图，来源 <https://matplotlib.org/stable/gallery/showcase/anatomy.html>;
Bk1_Ch10_01.ipynb

如图 1 所示，一幅图的基本构成部分包括以下几个部分：

- ▶ **图片对象 (figure)**：整个绘图区域的边界框，可以包含一个或多个子图。
- ▶ **子图对象 (axes)**：实际绘图区域，包含若干坐标轴、绘制的图像和文本标签等。
- ▶ **坐标轴 (axis)**：显示子图数据范围并提供刻度标记和标签的对象。
- ▶ **图脊 (spine)**：连接坐标轴和图像区域的线条，通常包括上下左右四条。

- ▶ **标题 (title)**: 描述整个图像内容的文本标签，通常位于图像的中心位置或上方，用于简要概括图像的主题或内容。
- ▶ **刻度 (tick)**: 刻度标记，表示坐标轴上的数据值。
- ▶ **标签 (label)**: 用于描述坐标轴或图像的文本标签。
- ▶ **图例 (legend)**: 标识不同数据系列的图例，通常用于区分不同数据系列或数据类型。
- ▶ **艺术家 (artist)**: 在 Matplotlib 中，所有绘图元素都被视为艺术家对象，包括图像区域、子图区域、坐标轴、刻度、标签、图例等等。

可视化工具

图 1 这幅图是用 Matplotlib 库绘制。Matplotlib 是 Python 中最基础的绘图工具。鸢尾花书中最常用的绘图库包括：Matplotlib、Seaborn、Plotly。

Matplotlib 可能是 Python 中最常用的绘图库，Matplotlib 具有丰富的绘图功能和灵活的使用方式。Matplotlib 可以绘制多种类型的图形，包括折线图、散点图、柱状图、饼图、等高线图等各种二维、三维图像，还可以进行图像处理和动画制作等。图 21、图 22、图 23 给出 Matplotlib 中常见的可视化方案。

Seaborn 是基于 Matplotlib 的高级绘图库，专注于统计数据可视化。它提供了多种高级数据可视化技术，包括分类散点图、热图（热力图）、箱线图、分布图等，可以快速生成高质量的统计图表。Seaborn 适用于数据分析、数据挖掘和机器学习等领域。

⚠ 注意，Matplotlib 和 Seaborn 生成的都是静态图，即图片。

Plotly 是一个交互式可视化库，可以生成高质量的静态和动态图表。它提供了丰富的图形类型和交互式控件，可以通过滑块、下拉列表、按钮等方式动态控制图形的显示内容和样式。Plotly 适用于 Web 应用、数据仪表盘和数据科学教育等领域。类似 Plotly 的 Python 库还有 Bokeh、Altair、Pygal 等。


鸢尾花书中，大家会发现 PDF 书稿、纸质书图片一般会使用 Matplotlib、Seaborn 生成的矢量图，配套的 JupyterLab Notebook、Streamlit 则倾向于采用 Plotly。



本书第六大板块“数据”会介绍 Pandas 本身、Seaborn 的统计描述可视化方案。

10.2 使用 Matplotlib 绘制线图

下面我们聊一下如何用 Matplotlib 可视化正弦、余弦函数，图 2 所示代码生成图 3。下面我们逐块讲解这段代码；此外，请大家在 JupyterLab 中复刻这段代码，并绘制图 3。

虽然相信大家对  这句导入已经不陌生，但是还是要“反复”简单讲一下。import (i 小写) 导入语句库、模块、函数。pyplot 是 matplotlib 的一个模块，我们将 matplotlib.pyplot 模块导入并简作 plt。这样我们可以使用 plt 来调用 matplotlib.pyplot 模块中的函数，而不需要每次都输入较长的模块名。

当然大家可以给这个模块起个其他名字，比如 p、mp、pl 等等；但是，对于初学者，建议大家采用约定成俗的简写方式，别在这些细枝末节上浪费精力。

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

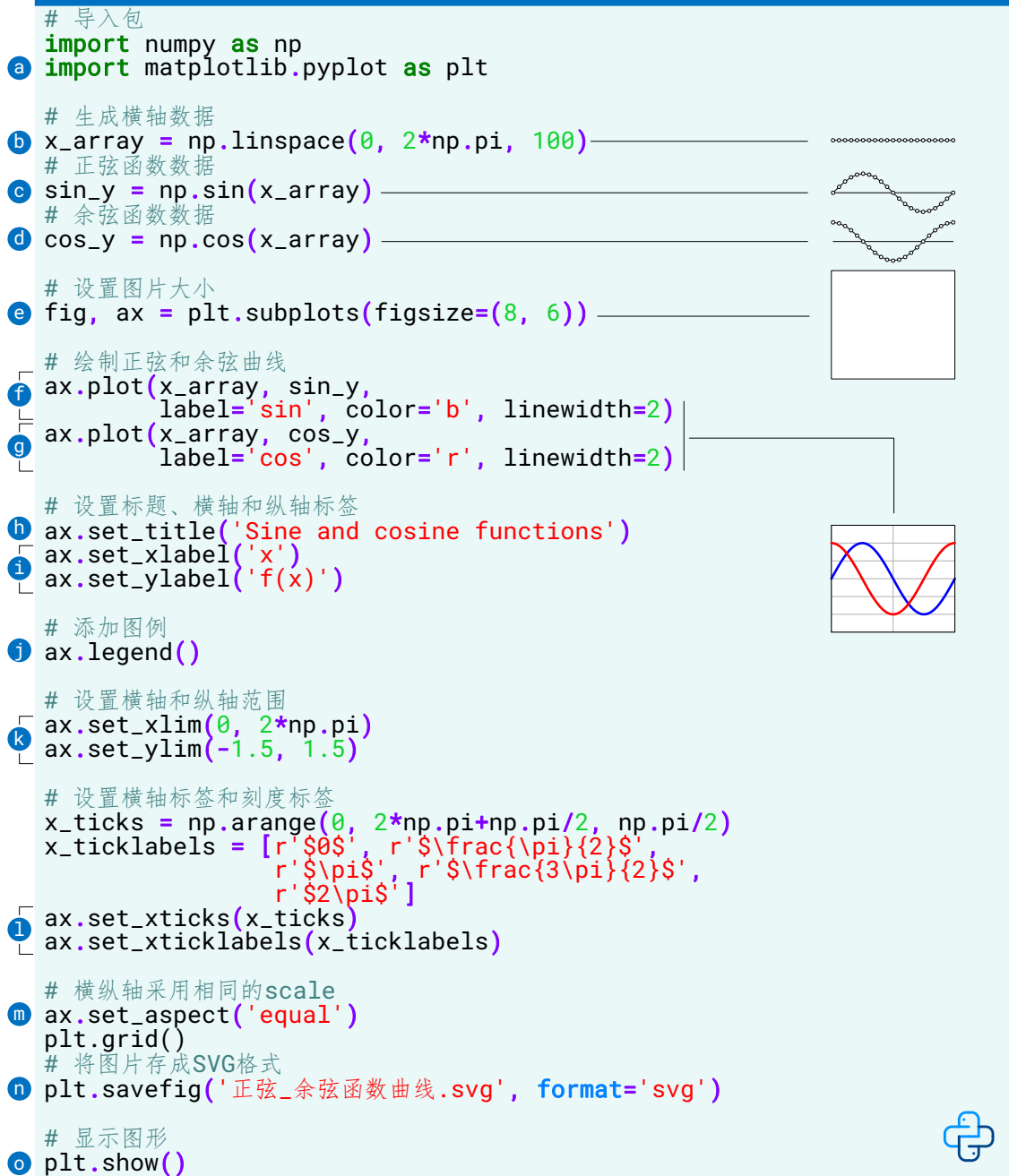


图 2. 用 Matplotlib 绘制正弦、余弦线图; Bk1_Ch10_02.ipynb

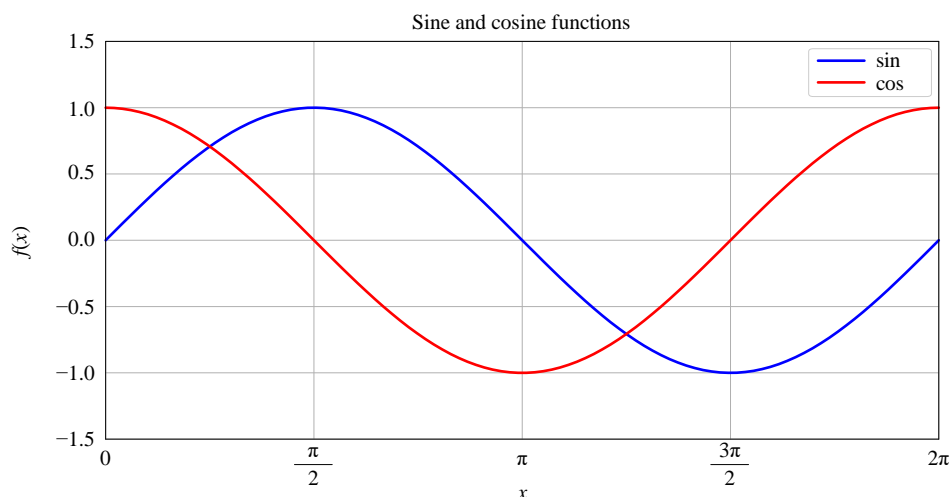


图 3. 正弦、余弦函数线图

产生等差数列

图 2 代码第一句，先用 `import numpy as np` 将 NumPy (Python 代码中叫 `numpy`) 库导入到当前的 Python 程序中，并为其取一个简短的别名 `np`。

再次强调，这意味着我们可以使用 `np` 来代替 `numpy` 来调用 NumPy 库中的函数和方法，例如 `np.linspace()`，`np.sin()`，`np.cos()` 等。这样做的好处是可以简化代码，减少打字量，并且提高代码的可读性。再次强调，人们约定成俗将 `numpy` 取别名为 `np`，不建议自创其他简写。

b 利用 `numpy.linspace()` 生成在给定范围内等差数列。由于在导入 `numpy` 时，我们将其命名为 `np`，因此代码中大家看到的是 `np.linspace()`。

图 4. 用 `numpy.linspace()` 生成等差数列

在 **b** `numpy.linspace()` 的输入中，位置参数 `0` 是数值序列的起始值，`2*np.pi` 是数值序列的结束值，`100` 是数值序列的数量。`numpy.linspace()` 函数默认包含右端点，即 `2*np.pi`。因此，`x_array = np.linspace(0, 2*np.pi, 100)` 在 $[0, 2\pi]$ 闭区间内生成一个 100 个数值等差数列。



本书后续第 13 ~ 18 章将专门讲解 NumPy 库的常用函数、方法。

fx

`numpy.linspace(start, stop, num=50, endpoint=True)`

这个函数的重要输入参数：

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

- **start**: 起始点的值。
- **stop**: 结束点的值。
- **num**: 要生成的数据点数量, 默认为 50。
- **endpoint**: 布尔值, 指定是否包含结束点。如果为 **True**, 则生成的数据点包括结束点; 如果为 **False**, 则生成的数据点不包括结束点。默认为 **True**。

请大家在 JupyterLab 中自行学习下列。

```
import numpy as np

arr = np.linspace(0, 1, num=11)
print(arr)

arr_no_endpoint = np.linspace(0, 1, num=10, endpoint=False)
print(arr_no_endpoint)
```



什么是 NumPy 数组 Array?

NumPy 中最重要的数据结构是 **ndarray (n-dimensional array)**, 即多维数组。一维数组是最简单的数组形式, 类似于 Python 中的列表。它是一个有序的元素集合, 可以通过索引访问其中的元素。一维数组只有一个轴。二维数组是最常见的数组形式, 可以看作是由一维数组组成的表格或矩阵。它有两个轴, 通常称为行和列。我们可以使用两个索引来访问二维数组中的元素。多维数组是指具有三个或更多维度的数组。

正弦、余弦

③ 中 `numpy.sin()` 和 ④ 中 `numpy.cos()` 是 NumPy 库中的数学函数, 用于计算给定角度的正弦和余弦值, 具体如图 5 所示。这两个函数的输入既可以是单个弧度值 (比如 `numpy.pi/2`), 也可以是数组 (一维、二维、多维)。比如 ③ 和 ④ 中, 两个函数的输入都是一维 NumPy 数组。从这一点上来看, 利用 NumPy 数组向量化运算, 要比 Python 的列表方便得多。

⚠ 注意, NumPy 中 `numpy.deg2rad()` 将角度转换为弧度, `numpy.rad2deg()` 将弧度转换为角度。

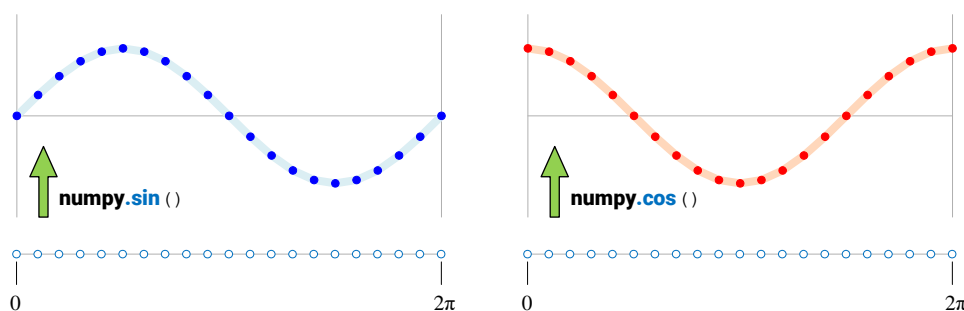


图 5. 生成正弦、余弦数据

创建图形、轴对象

⑤ 中 `fig, ax = plt.subplots(figsize=(8, 6))` 用于创建一个新的 Matplotlib 图形 `fig` 和一个轴 `ax` 对象, 并设置图形的大小为 (8, 6), 单位为英寸。

通过创建图形和轴对象，我们可以在轴上绘制图表、设置轴的标签和标题、调整轴的范围等。`fig, ax = plt.subplots()` 这一句代码常常是开始绘图的第一步，它创建了一个具有指定大小的图形和轴对象，为后续绘图操作提供了一个可用的基础。

⚠ 再次强调，`plt` 是 `Matplotlib` 的一个常用的别名，前文已经通过 `import matplotlib.pyplot as plt` 引入。所以在使用 `plt.subplots()` 函数之前，需要确保已经正确导入了 `Matplotlib` 库的 `pyplot` 模块。

添加子图

此外，如图 6 代码所示，我们还可以使用 `add_subplot()` 方法创建一个新的子图对象，并指定其所在的行、列、编号等属性。

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

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

a fig = plt.figure()
b ax = fig.add_subplot(1, 1, 1)
ax.plot(x, y)
plt.show()
```

图 6. 用 `add_subplot()` 方法创建一个新的子图对象;  Bk1_Ch10_03.ipynb

在这个例子中，**a** 先用 `plt.figure()` 生成了一个 `Figure` 对象，然后使用 `add_subplot()` 方法创建了一个新的子图轴对象 `ax`，并将其添加到 `Figure` 对象中。

其中，`1,1,1` 参数表示子图在 1 行 1 列的第 1 个位置，即占据整个 `Figure` 对象的空间。然后，我们在子图中绘制了一个正弦曲线。注意，`1,1,1` 也可以写作 `111`。此外，若是想要添加若干子图，比如两行一列，可以分别用 `ax1 = fig.add_subplot(2,1,1)`、`ax2 = fig.add_subplot(2,1,2)` 生成两个子图的轴对象 `ax1`、`ax2`。

最后，使用 `plt.show()` 函数显示 `Figure` 对象，即可在屏幕上显示绘制的图像。

请大家参考图 2 代码进一步装饰图 6。

绘制曲线

回到图 2 代码，**f** 中 `ax.plot(x_array, sin_y, label='sin', color='blue', linewidth=2)` 用于在轴对象 `ax` 上绘制正弦曲线。`x_array` 为 `x` 轴数据，`sin_y` 为 `y` 轴数据。

参数 `label='sin'` 设置了曲线的标签为 `'sin'`。

参数 `color='blue'` 设置曲线的颜色为蓝色。

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

参数 `linewidth=2` 设置曲线的线宽为 2。线宽的单位是点 (point, pt)，通常用于测量线条、字体等绘图元素的大小。在 Matplotlib 中，默认情况下，一个点等于 1/72 inch。

在 Matplotlib 中，`linewidth` 参数表示线条的宽度，可以简作 `lw`。

类似地，参数 `color`，可以简作 `c`；参数 `linestyle` 可以简作 `ls`；参数 `markeredgecolor` 可简作 `mec`；`markeredgewidth` 可简作 `mew`；`markerfacecolor` 简作 `mfc`；`markersize` 简作 `ms`。

请大家自行分析⁹。

其他“艺术家”

图 2 代码还采用了各种图片装饰命令，下面逐一说明。

- ▶ `ax.set_title('Sine and cosine functions')` 设置图表的标题为 "Sine and cosine functions"，即正弦和余弦函数。
- ▶ `ax.set_xlabel('x')` 设置横轴标签为 "x"。`ax.set_ylabel('f(x)')` 设置纵轴标签为 "f(x)"。
- ▶ `ax.legend()` 添加图例 legend，用于标识不同曲线或数据系列。
- ▶ `ax.set_xlim(0, 2*np.pi)` 设置横轴范围从 0 到 2π 。`ax.set_ylim(-1.5, 1.5)` 设置纵轴范围从 -1.5 到 1.5。
- ▶ `x_ticks = np.arange(0, 2*np.pi+np.pi/2, np.pi/2)` 生成横轴刻度的位置，从 0 到 2π ，间隔为 $\pi/2$ 。
- ▶ `x_ticklabels = [r'0', r'\frac{\pi}{2}$', r'\pi$', r'\frac{3\pi}{2}$', r'$2\pi$']` 设置横轴刻度的标签，分别为 0, $\pi/2$, π , $3\pi/2$, 2π 。在代码中，`r'\frac{\pi}{2}$'` 是一个特殊的字符串，用于表示数学公式中的文本。在这个字符串前面的 `r` 前缀表示该字符串是一个“原始字符串”，即不对字符串中的特殊字符进行转义。
- ▶ 在这个特殊字符串中，使用了 LaTeX 符号来表示一个分数。具体来说，`\frac{\pi}{2}` 表示一个分数，分子是 π ，分母是 2。当这个字符串被用作横轴刻度的标签时，它会在图表中显示为 " $\pi/2$ " 的形式。这种表示方法可以用于在图表中显示复杂的数学公式或符号。
- ▶ `ax.set_xticks(x_ticks)` 设置横轴刻度的位置。
- ▶ `ax.set_xticklabels(x_ticklabels)` 设置横轴刻度的标签。
- ▶ `ax.set_aspect('equal')` 设置横纵轴采用相同的比例，保持图形在绘制时不会因为坐标轴的比例问题而产生形变。

图片输出格式

ⁿ 采用 `matplotlib.pyplot.savefig()`，简做 `plt.savefig()`，保存图片。

Matplotlib 可以输出多种格式的图片，其中一些是矢量图，比如 SVG。以下是一些常见的输出格式及其特点：

- ▶ PNG (Portable Network Graphics): PNG 是一种常见的位图格式，支持透明度和压缩。PNG 格式输出的图片不是矢量图，因此在放大时会失去清晰度，但是可以保持较高的分辨率和细节。

- ▶ **JPG/JPEG (Joint Photographic Experts Group)**: JPG 是一种常见的有损压缩位图格式，用于存储照片和复杂的图像。与 PNG 不同，JPG 格式输出的图片是有损的，压缩率高时会失去一些细节，但是文件大小通常较小。
- ▶ **EPS (Encapsulated PostScript)**: EPS 是一种矢量图格式，可以在很多绘图软件中使用。EPS 格式输出的图片可以无限放大而不失真，适合于需要高品质图像的打印和出版工作。
- ▶ **PDF (Portable Document Format)**: PDF 是一种常见的文档格式，可以包含矢量图和位图。与 EPS 类似，PDF 格式输出的图片也是矢量图，可以无限放大而不失真，同时具有可编辑性和高度压缩的优势。存成 PDF 很方便插入 Latex 文档。
- ▶ **SVG (Scalable Vector Graphics)**: SVG 是一种基于 XML 的矢量图格式，可以用于网页和打印等多种用途。SVG 格式输出的图片可以无限放大而不失真，且文件大小通常较小。鸢尾花书的图片首选 SVG 格式保存。

⚠ 注意，EPS、PDF 和 SVG 是矢量图格式，可以无限放大而不失真（比如图 7 (b)），适合于需要高品质图像的打印和出版工作。在需要高品质图像的场合，最好使用这些矢量图格式。

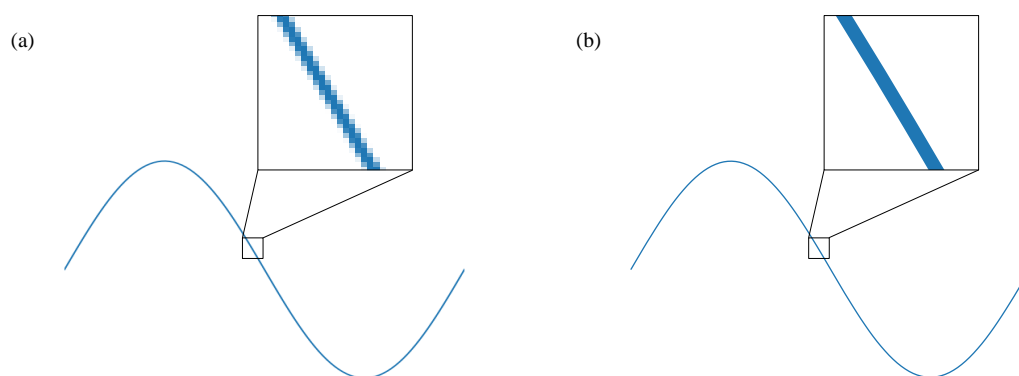


图 7. 比较非矢量、矢量图

后期处理

大家会在鸢尾花书中发现，我们用 Python 代码生成的图像和书中的图像很多细节上并不一致。产生这种偏差的原因有很多。

首先，为了保证矢量图像质量及可编辑性，每幅 Python 代码生成的图形都会经过多道后期处理。也就是说，鸢尾花书中每一幅图都经过“千锤百炼”。前期需要构思创意，然后 Python 编码，矢量出图之后还要一张张后期制作。在出版社排版老师手里，草稿中的图形对象还要再经过多轮制作才定型。

草稿阶段用到的后期处理的工具包括（但不限于）Inkscape、MS Visio、Adobe Illustrator。使用怎样的工具要根据图片类型、图片大小等因素考虑。

出版社排版老师用的排版工具为 Adobe InDesign。

Inkscape 是开源免费的矢量图形编辑软件，支持多种矢量图形格式，适用于绘制矢量图形、图标、插图等。

MS Visio 特别适合做示意图、流程图等矢量图像。

Adobe Illustrator 是 Adobe 公司开发的专业矢量图形编辑软件，功能强大，广泛用于图形设计、插图、标志设计等。比如鸢尾花书的封面都是用 Adobe Illustrator 设计，鸢尾花书中复杂的图像也都是在这个软件设计生成。

此外，也推荐大家使用 CorelDRAW。CorelDRAW 是 Corel 公司开发的矢量图形编辑软件，具有类似于 Adobe Illustrator 的功能，是一种流行的矢量图形处理工具。

⚠ 请大家务必注意，图片后期加工过程仅仅是为了美化图像，并没有篡改数据本身。特别是在科学研究中，不篡改数据是一条铁律，希望大家谨记。

也就是说哪怕图 2 这种简单的线图中的所有“艺术家 (artist)”，即所有元素，都被加工过。比如，图中的数字、英文、希腊字母都是作者手动添加上去的（为了保证文本可编辑）。

此外，从时间成本角度来看，一些标注、艺术效果用 Python 写代码方生成并不“划算”；鸢尾花书中，诸如箭头、指示线、注释等元素也都是后期处理时作者手动添加。

有一种特殊情况，就是同一类图形将会反复代码出图，这样的话为了节省后期制作时间，我们可以考虑写代码“自动化”某些标注、艺术效果。举个例子，如果我们需要用 Python 代码生成 50 张直方图 (histogram)，用来展示不同特征数据分布。在这些图上，我们要打印数据的基本统计数据（均值、众数、中位数、最大值、最小值、四分位点、5%和 95%百分位、峰度、偏度等等），这时手动添加的时间成本太高。莫不如在代码中写几句话将这些数值直接打印到图片上。

子图

图 8 所示一行两列子图，分别展示正弦、余弦函数曲线。图 9 代码绘制图 8，下面分析其中关键语句。

a 创建一个一行两列子图的图形对象。

位置参数“1,2”代表 1 行、2 列子图布局。

参数 `figsize=(10, 4)` 指定了整个图形的大小为宽度、高度。

参数 `sharey=True` 表示两个子图共享相同的 y 轴，这意味着它们在垂直方向上具有相同的刻度和范围。

而 `fig, (ax1, ax2)` 将 `plt.subplots` 返回值解包，其中 `fig` 是整个图形对象，而 `(ax1, ax2)` 是一个包含两个子图对象的元组。这样，我们可以分别通过 `ax1` 和 `ax2` 来操作这两个子图。

b 在 `ax1` 轴对象上，用 `plot()` 方法绘制正弦曲线线图。

c 对 `ax1` 进行装饰，请大家逐行注释。

d 在 `ax2` 轴对象上，用 `plot()` 方法绘制余弦曲线线图。

e 对 `ax2` 进行装饰，请大家逐行注释。

f 自动调整子图或图形的布局，使其更加紧凑。在创建包含多个子图的图形时，有时候可能会出现重叠的标签或坐标轴，`tight_layout()` 就是为了解决这个问题而设计。

g 打印图像。

请大家在 JupyterLab 中给图 9 代码逐行添加注释，并复刻图 8。



《可视之美》将介绍更多子图可视化方案。

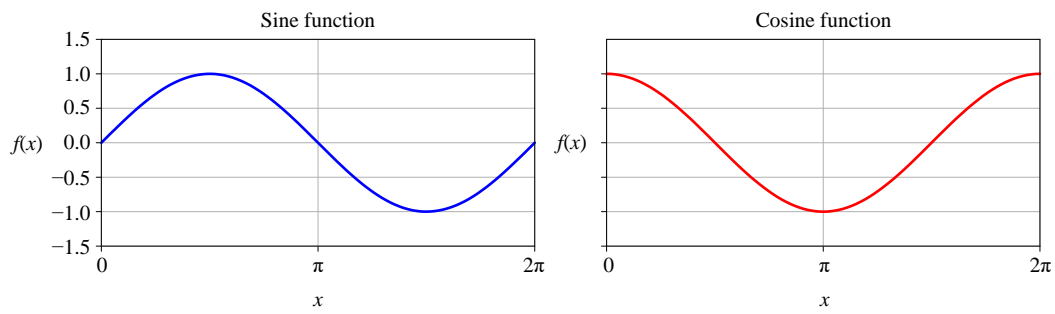


图 8. 一行、两列子图

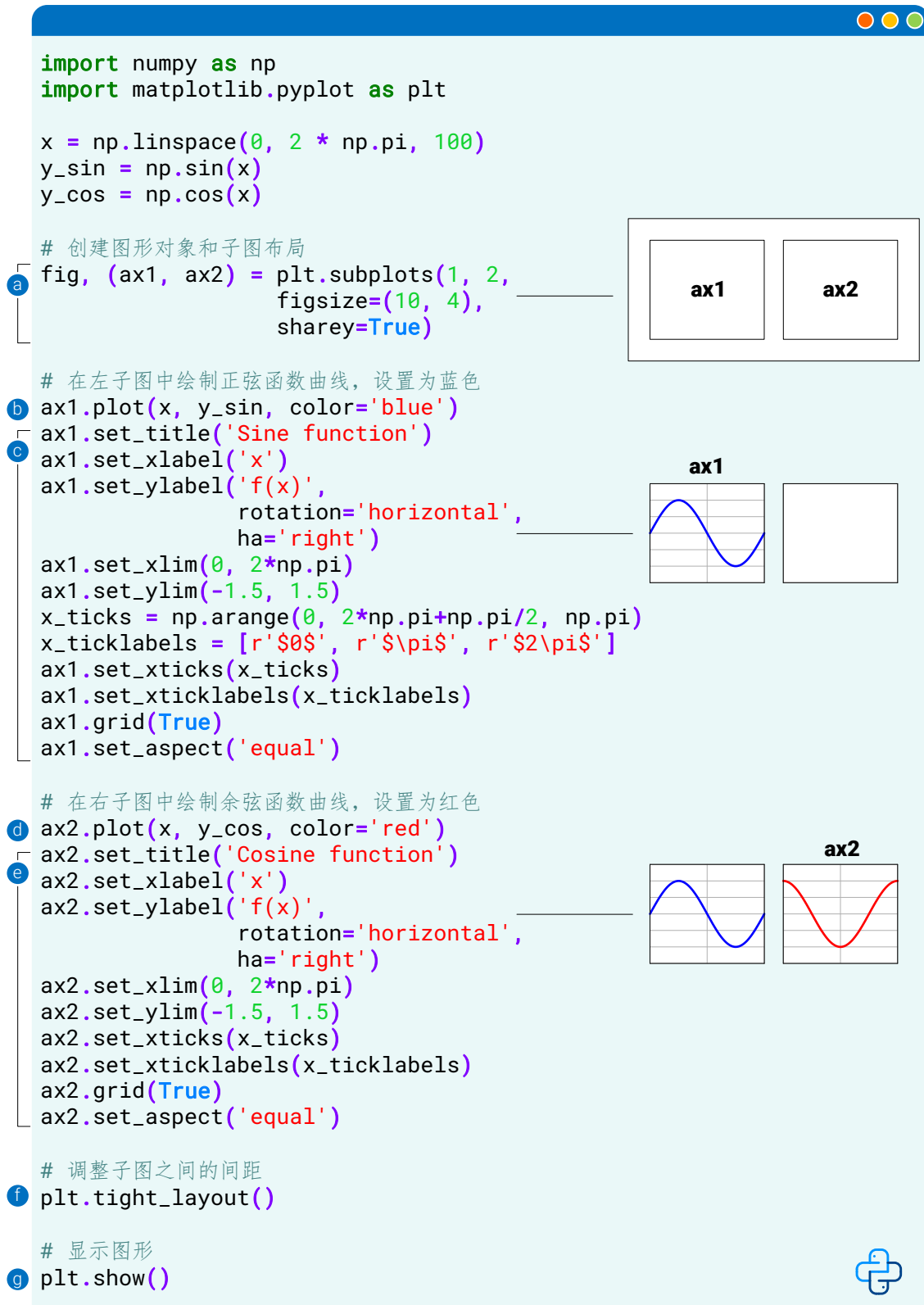


图 9. 绘制一行两列子图; Bk1_Ch10_04.ipynb

以利用 Matplotlib 工具绘制线图为例，大家会发现，有些时候我们利用 `plt.plot()`，有些时候用 `ax.plot()`。注意，`plt` 相当于“提笔就画”，`ax` 是在指定轴对象操作。如果只需要绘制简单的

图形，使用 `plt` 函数就足够了；但是如果需要更复杂的图形布局或多个子图，使用 `ax` 函数会更方便。表 1 比较各种常用的 `plt` 和 `ax` 函数。

表 1. 比较 `plt` 和 `ax` 函数

功能	plt 函数 <code>import matplotlib.pyplot as plt</code>	ax 函数 <code>fig, ax = plt.subplots()</code> <code>fig, axes = plt.subplots(n_rows, n_cols)</code> <code>ax = axes[row_num][col_num]</code>
创建新的图形	<code>plt.figure()</code>	
创建新的子图	<code>plt.subplot()</code>	<code>ax = fig.add_subplot()</code>
创建折线图	<code>plt.plot()</code>	<code>ax.plot()</code>
添加横轴标签	<code>plt.xlabel()</code>	<code>ax.set_xlabel()</code>
添加纵轴标签	<code>plt.ylabel()</code>	<code>ax.set_ylabel()</code>
添加标题	<code>plt.title()</code>	<code>ax.set_title()</code>
设置横轴范围	<code>plt.xlim()</code>	<code>ax.set_xlim()</code>
设置纵轴范围	<code>plt.ylim()</code>	<code>ax.set_ylim()</code>
添加图例	<code>plt.legend()</code>	<code>ax.legend()</code>
添加文本注释	<code>plt.text()</code>	<code>ax.text()</code>
添加注释	<code>plt.annotate()</code>	<code>ax.annotate()</code>
添加水平线	<code>plt.axhline()</code>	<code>ax.axhline()</code>
添加垂直线	<code>plt.axvline()</code>	<code>ax.axvline()</code>
添加背景网格	<code>plt.grid()</code>	<code>ax.grid()</code>
保存图形到文件	<code>plt.savefig()</code>	通常使用 <code>fig.savefig()</code>

10.3 图片美化

颜色

在 `Matplotlib` 中，可以使用多种方式指定线图的颜色，包括 RGB 值、预定义颜色名称、十六进制颜色码和灰度值。

可以使用 RGB（R 是 red，G 是 green，B 是 blue）来指定颜色，其中每个元素的值介于 0 到 1 之间。例如，(1, 0, 0) 表示纯红色，(0, 1, 0) 表示纯绿色。使用 RGBA 值指定“颜色 + 透明度 (A)”。

如图 11 所示，RGB 三原色模型实际上构成了一个色彩“立方体”——一个色彩空间。也就是说在这个立方体中藏着无数种色彩。

➡ 鸢尾花书《矩阵力量》将会用 RGB 三原色模型讲解线性代数中向量空间（vector space）这个重要概念。

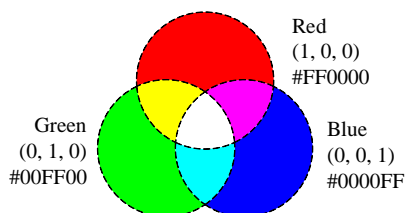


图 10. RGB 三原色模型

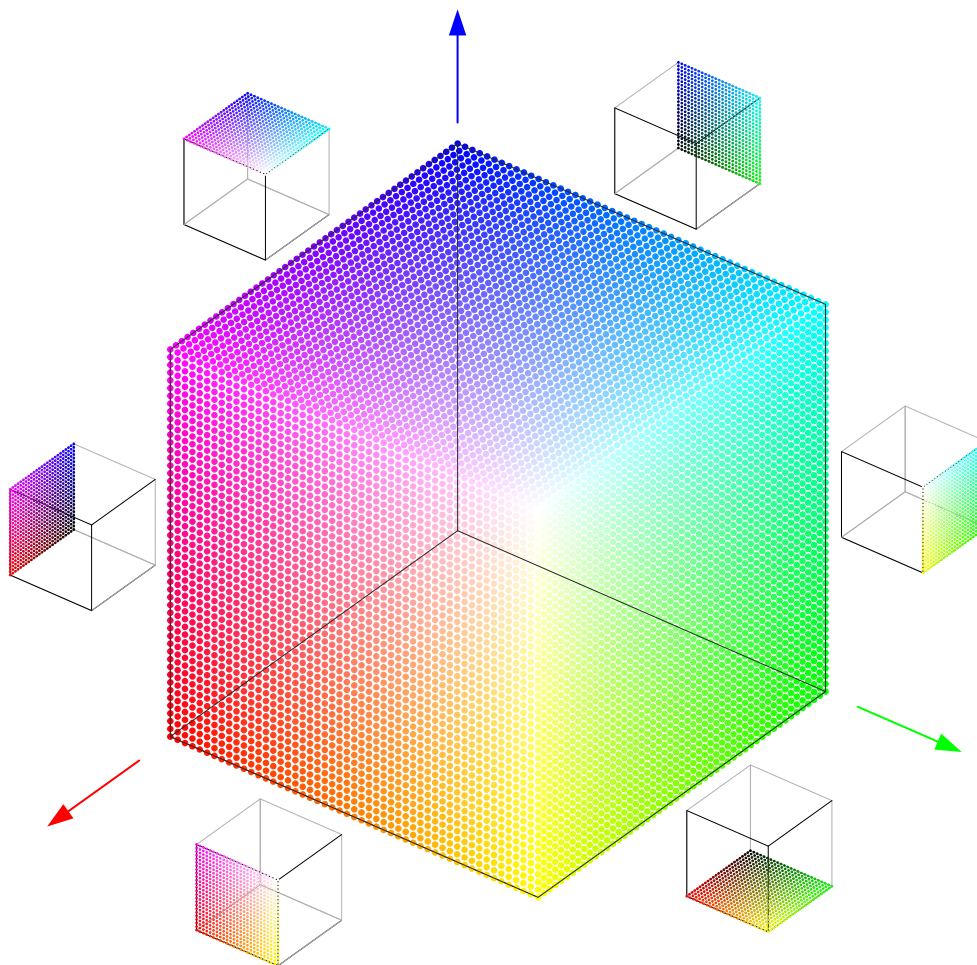


图 11. RGB 三原色模型“立方体”



什么是 RGB 颜色模式？

RGB（红绿蓝）颜色模式是一种使用红、绿、蓝三个基本颜色通道来表示颜色的方法。在 RGB 模式中，通过调整每个通道的强度（从 0 到 255 的值，Matplotlib 中 0 到 1 的值）来创建各种颜色。通过组合不同强度的红、绿和蓝，可以形成几乎所有可见光颜色。RGB 颜色模式被广泛应用于计算机图形、数字图像处理和网页设计等领域，它提供了一种直观、灵活且广泛支持的方式来表示和操作颜色。

Matplotlib 提供了一些常见颜色的预定义名称，例如 'red'、'green'、'blue' 等。图 20 所示为在 Matplotlib 中已经预定义名称的颜色。

本书前文介绍过，大家还可以使用十六进制颜色码字符串来指定颜色。它以 '#' 开头，后面跟着六位十六进制数。例如，'#FF0000' 表示纯红色，'#00FF00' 表示纯绿色。

我们还可以使用灰度值来指定颜色，取值介于 0 到 1 之间，表示不同的灰度级别。'0' 表示黑色，'1' 表示白色。比如，color='0.5' 代表灰度值为 0.5 的灰色。

使用色谱

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

Matplotlib 中还有一种渐变配色方案——颜色映射（colormap）。在 Matplotlib 中，colormap 用于表示从一个端到另一个端的颜色变化。这个变化可以是连续的，也可以是离散的。Colormap 可以直译为“颜色映射”“色彩映射”，鸢尾花书一般称之为“色谱”。

图 12 所示为几种常见的色谱。鸢尾花书中最常用的色谱为 RdYlBu。

《可视之美》将专门讲解色谱。

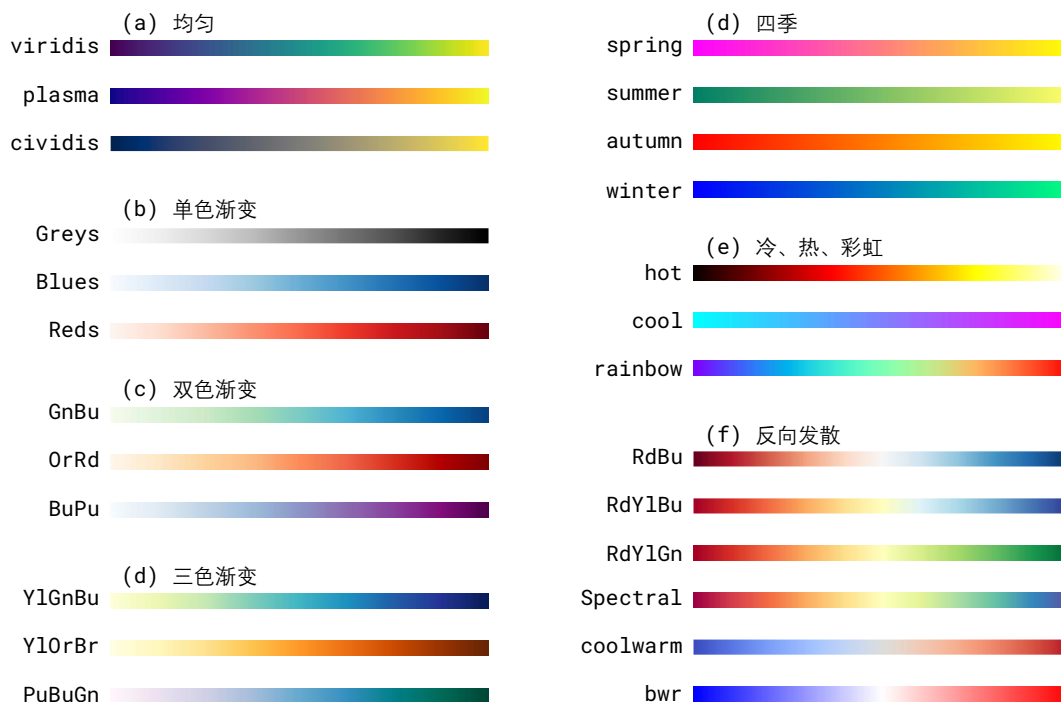


图 12. 几种常用色谱

在 Matplotlib 中，colormap 主要用于绘制二维图形，如热图、散点图、等高线图等。它用于将数据值映射到不同的颜色，以显示数据的变化和模式。图 13 展示使用色谱的几个场合。

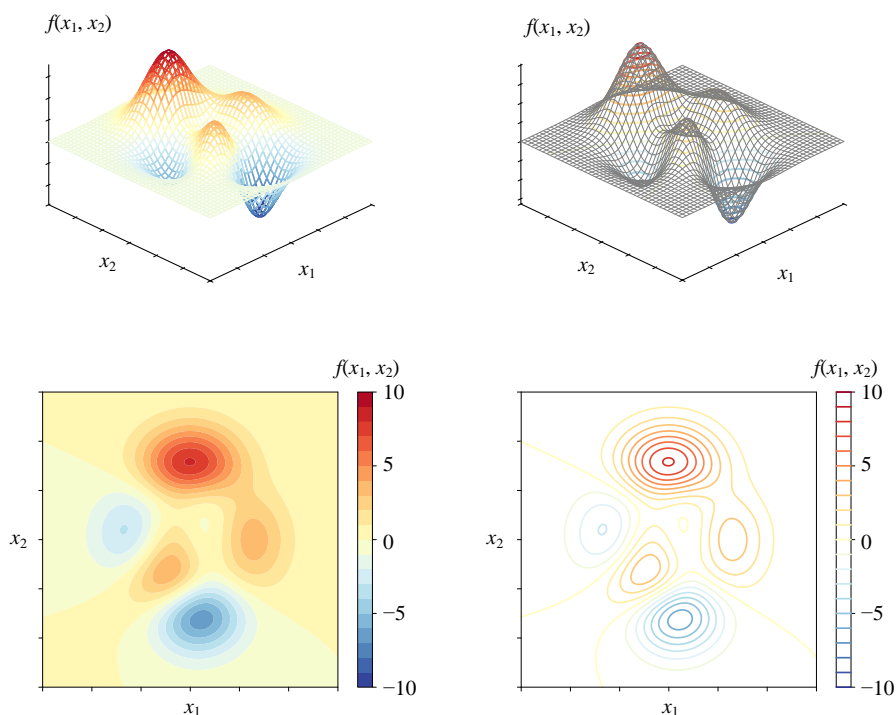


图 13. 使用色谱的几个场合

图 14 所示为利用色谱渲染一组曲线。图 14 左图所示为一元高斯概率密度分布曲线随均值 μ 变化，图 14 右图所示为曲线随标准差 σ 变化。



本书第 26 章会介绍获得图 14 两幅子图的代码。

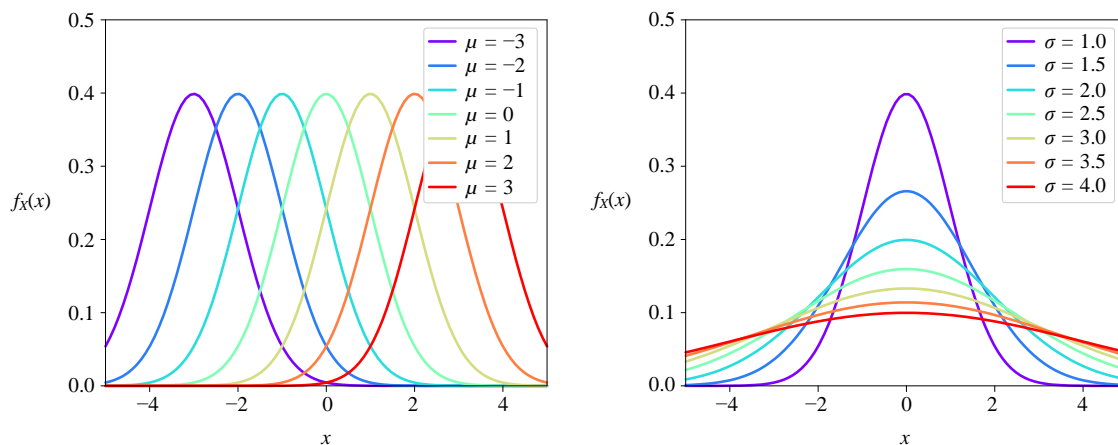


图 14. 用色谱渲染一组曲线



什么是高斯分布？

高斯分布 (Gaussian distribution)，也称为正态分布 (Normal distribution)，是统计学中常用的概率分布模型之一。它具有钟形曲线的形状，呈对称分布。高斯分布的概率密度函数可以由两个参数完全描述：均值 (mean) 和标准差 (standard deviation)。均值决定了分布的中心位置，标准差决定了分布的展开程度。

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

高斯分布在自然界和社会现象中广泛存在，例如身高、体重、温度等连续型随机变量常常服从高斯分布。中心极限定理也说明了许多独立同分布的随机变量的总和趋向于高斯分布。

高斯分布在统计学和数据分析中有着重要的应用，可用于描述数据集的分布特征、进行假设检验、构建回归模型等。在机器学习和人工智能领域，高斯分布在概率密度估计、聚类分析、异常检测等算法中被广泛使用。



什么是概率密度函数？

概率密度函数 (Probability Density Function, 简称 PDF) 是概率论和统计学中用于描述连续型随机变量的概率分布的函数。它表示了变量落在某个特定取值范围内的概率密度，而不是具体的概率值。

一元连续随机变量的概率密度函数是非负函数，并且在整个定义域上的积分等于 1。对于给定的连续型随机变量，通过 PDF 可以计算出在不同取值范围内的概率密度值，从而了解变量的分布特征和概率分布形状。

以正态分布为例，其概率密度函数即高斯函数，可以描述变量取值的概率密度。在某个特定取值处，概率密度函数的值越高，表示该取值的概率越大。概率密度函数在统计分析、数据建模、概率推断等领域广泛应用，可用于计算概率、推断参数、生成模拟数据等。

默认设置

Matplotlib 提供了许多配置参数，用于控制图形的默认设置。这些默认设置包括图形大小、颜色、字体、线条样式等。我们可以通过修改这些配置参数来自定义 Matplotlib 图形的外观和行为。

图 15 代码可以用来查看 Matplotlib.pyplot 绘图时的全套默认设置；同时，我们还可以通过列表给出的关键字修改默认设置。

由于列表过长，为了节省用纸，请大家在配套代码中查看。下面，我们挑几个常用设置简单介绍。

```
a import matplotlib.pyplot as plt
b p = plt.rcParams # 全局配置参数
  print(p)
  # plt.rcParams 配置参数的当前默认值
```

图 15. 查看 Matplotlib 图片默认设置； Bk1_Ch10_05.ipynb

比如，默认图片大小为 'figure.figsize': [6.4, 4.8]。通过 `plt.rcParams['figure.figsize'] = (8, 6)` 可以修改图片大小。默认线宽为 'lines.linewidth': 1.5。 `plt.rcParams['lines.linewidth'] = 2` 将线宽设置为 2 pt。

再如， `axes.prop_cycle: cycler('color', ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd', '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf'])` 告诉我们在绘制线图时，如果不指定具体颜色，在绘制若干线图时，会采用如图 16 右侧由上至下颜色依次渲染。颜色不够用时，重复颜色序列循环。

如果你不满意这组颜色循环，可以在绘制线图时，像前文介绍的那样分别指定颜色。或者，直接修改 `cycler`。这是《可视之美》要介绍的话题之一。

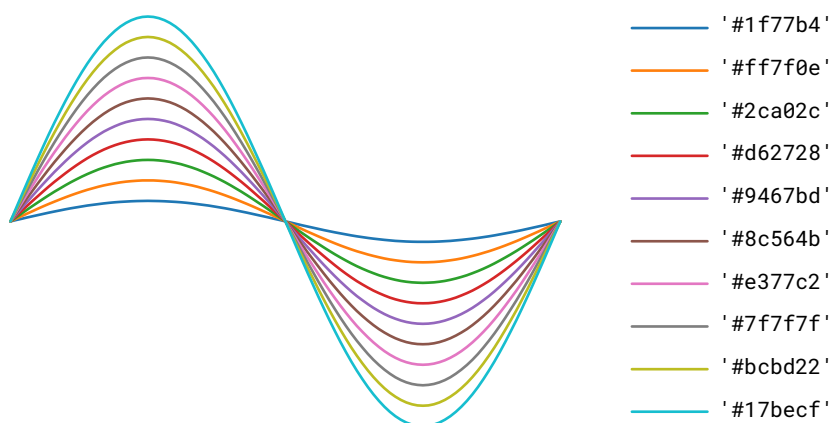


图 16. Matplotlib 线图默认颜色序列

10.4 使用 Plotly 绘制线图

我们还可以用 Plotly 绘制具有交互属性的图形，比如图 17。下面介绍两不同的方法绘制图 17。

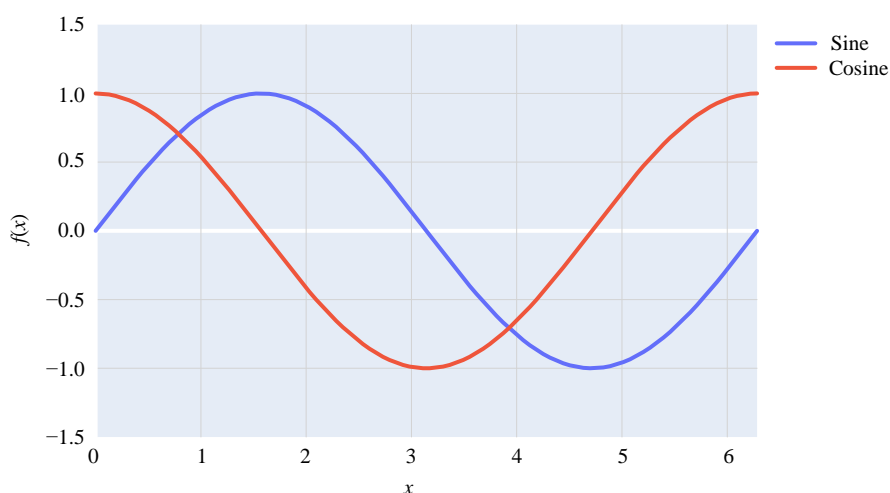


图 17. 用 Plotly 绘制具有交互性质的曲线

首先聊一聊图 18 代码的关键语句。

- 将 `plotly.express` 模块导入，简作 `px`。模块 `plotly.express` 中可视化方案很丰富，比如散点图、面积图、饼图、太阳爆炸图、直方图、冰柱图等等。
- 导入 `numpy`，简作 `np`。
- 用 `plotly.express.line()`，简作 `px.line()`，绘制线图。参数变量 `x` 为横轴坐标数据，参数变量 `y` 为纵轴坐标数据。参数 `labels` 用于设置图表的标签。字典中键值对 `'y': 'f(x)'` 指定了纵轴的标签为 `'f(x)'`。键值对 `'x': 'x'` 指定了横轴的标签为 `'x'`。
- 这两句修改了图例中两条线的标签。由于绘图时输入为一维 NumPy 数组，需要额外语句设定图例标签。下一段中，绘图时采用的数据类型是 `Pandas DataFrame`，就没有这个问题。
- 展示交互图片。

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

```

# 导入包
a import plotly.express as px
b import numpy as np

# 生成横轴数据
x = np.linspace(0, 2 * np.pi, 100)

# 生成正弦和余弦曲线数据
y_sin = np.sin(x)
y_cos = np.cos(x)

# 创建图表
c fig = px.line(x=x, y=[y_sin, y_cos],
                labels={'y': 'f(x)', 'x': 'x'})
# 修改图例
d fig.data[0].name = 'Sine'
  fig.data[1].name = 'Cosine'
# 显示图表
e fig.show()

```

图 18. 用 `plotly.express.plot()` 绘制线图，输入数据类型为 NumPy Array; Bk1_Ch10_06.ipynb

图 19 中代码 **a** 将 `pandas` 库导入，简做 `pd`。

b 利用 `pandas.DataFrame()` 构造数据帧。

`{'X': x, 'Sine': y_sin, 'Cosine': y_cos}` 是一个字典，其中键是数据帧中的列名，而值是对应列的数据。具体来说，`'x'` 列包含了 `x` 数组的数据，`'Sine'` 列包含了 `y_sin` 数组的数据，`'Cosine'` 列包含了 `y_cos` 数组的数据。

新创建的数据帧对象叫做 `df`。

c 调用 `plotly.express.line()` 绘制正弦、余弦曲线。输入的数据为新创建的数据帧 `df`，然后，我们直接可以通过数据帧列标签，比如 `'x'`、`'Sine'`、`'Cosine'` 调用数据帧具体数据。

实践中，大家会发现可视化库 `Seaborn` 和 `Plotly` 和 `Pandas DataFrame` 的结合更为密切。

本书第 19 ~ 24 章专门介绍 `Pandas` 库；其中，第 23 章将专门介绍“`Pandas + Plotly`”相结合用数据可视化讲故事的强大力量！

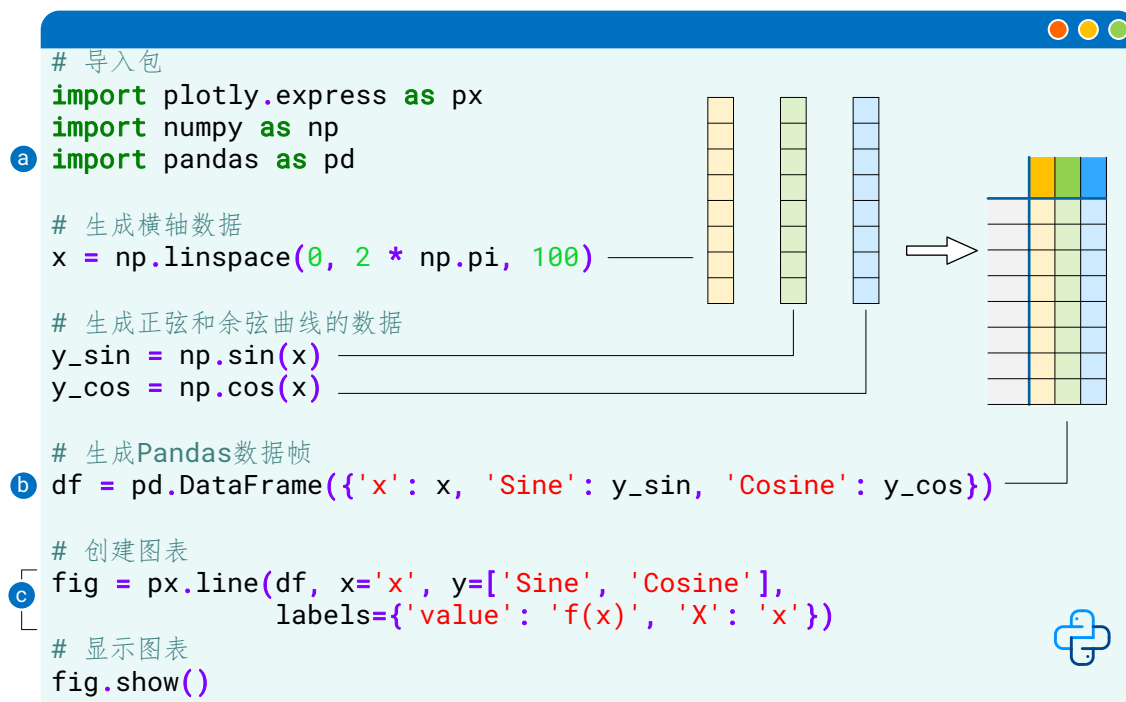


图 19. 用 `plotly.express.plot()` 绘制线图, 输入数据类型 Pandas DataFrame; Bk1_Ch10_07.ipynb



请大家完成下面 3 道题目。

Q1. 大家可以在本章配套代码中找到图 1 对应的 Matplotlib 官方提供的代码文件。本书将 Python 代码文件命名为 `Q1_Assignment_Anatomy_of_a_figure.py`。请大家给这个代码文件中的代码逐行中文注释, 并在 JupyterLab 中进行探究式学习。

Q2. Matplotlib 提供丰富的可视化方案实例, 图 21、图 22、图 23 大部分子图对应的代码都在如下链接中, 请大家在 JupyterLab 复刻每幅子图, 并补充必要注释。

https://matplotlib.org/stable/plot_types/index.html

* 本章习题不提供答案。



鸢尾花书的内核是“编程 + 可视化 + 数学 + 机器学习”, “可视化”是系列图书四根支柱之一! 鸢尾花书中的任何一幅图片起到的作用并不是单纯的“装饰”。

我们想用各种丰富的可视化方案帮助大家理解数学工具原理, 搞懂机器学习算法。从图片创意, 到编程实现, 最后后期处理, 整个过程也是一次“美学实践”。

Python 提供大量第三方可视化工具助力我们的“美学实践”! 本书中仅有三章内容专门介绍可视化, 而鸢尾花书《可视之美》整本就专注于一件事——如何画好图。



图 20. Matplotlib 已定义名称的颜色

整页排版，背景色采用奇数页网格笔记纸背景色

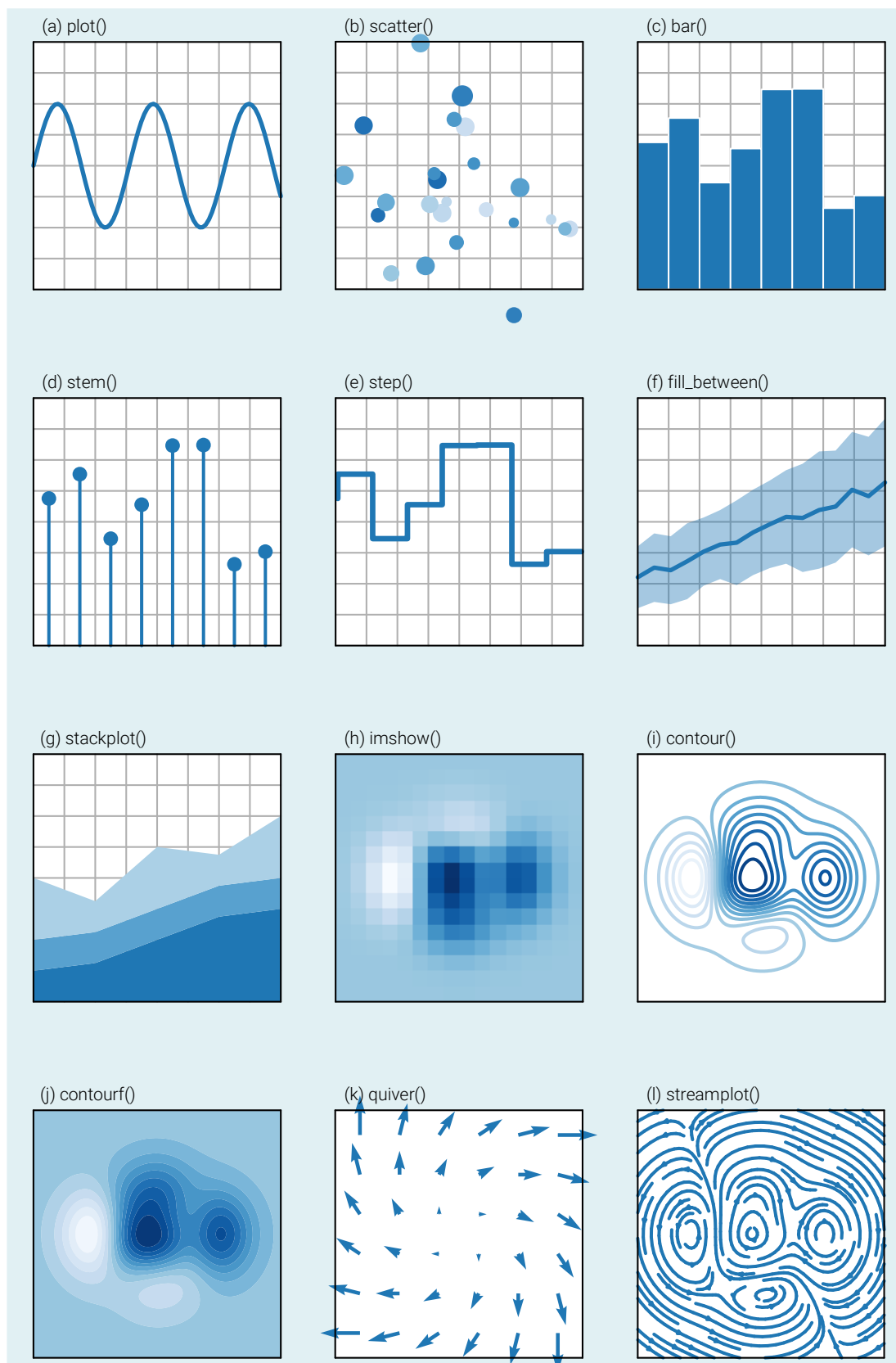


图 21. Matplotlib 常见可视化方案，第一组

整页排版，背景色采用奇数页网格笔记纸背景色，四面出血

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

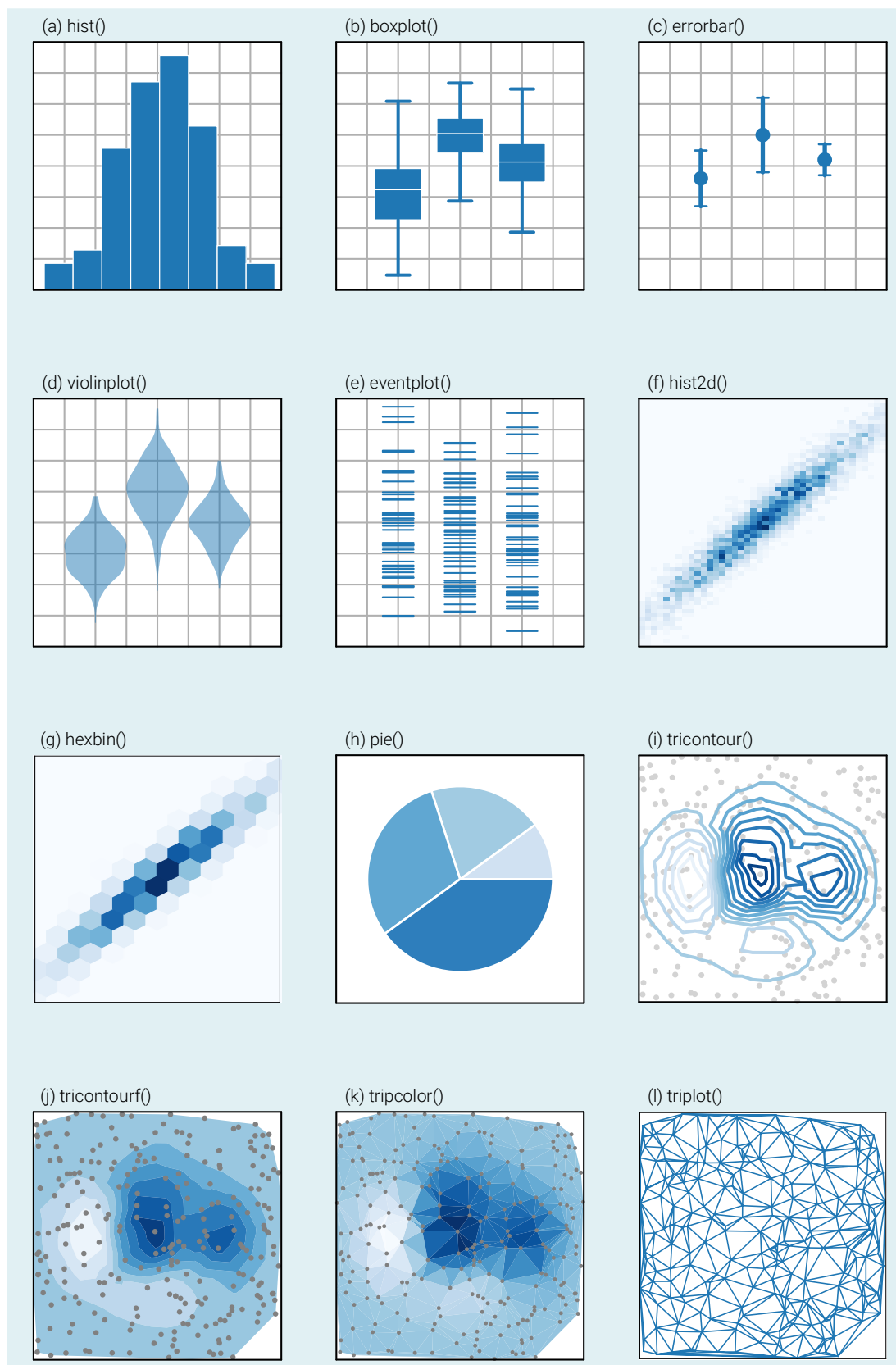


图 22. Matplotlib 常见可视化方案，第二组

整页排版，背景色采用奇数页网格笔记纸背景色，四面出血

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

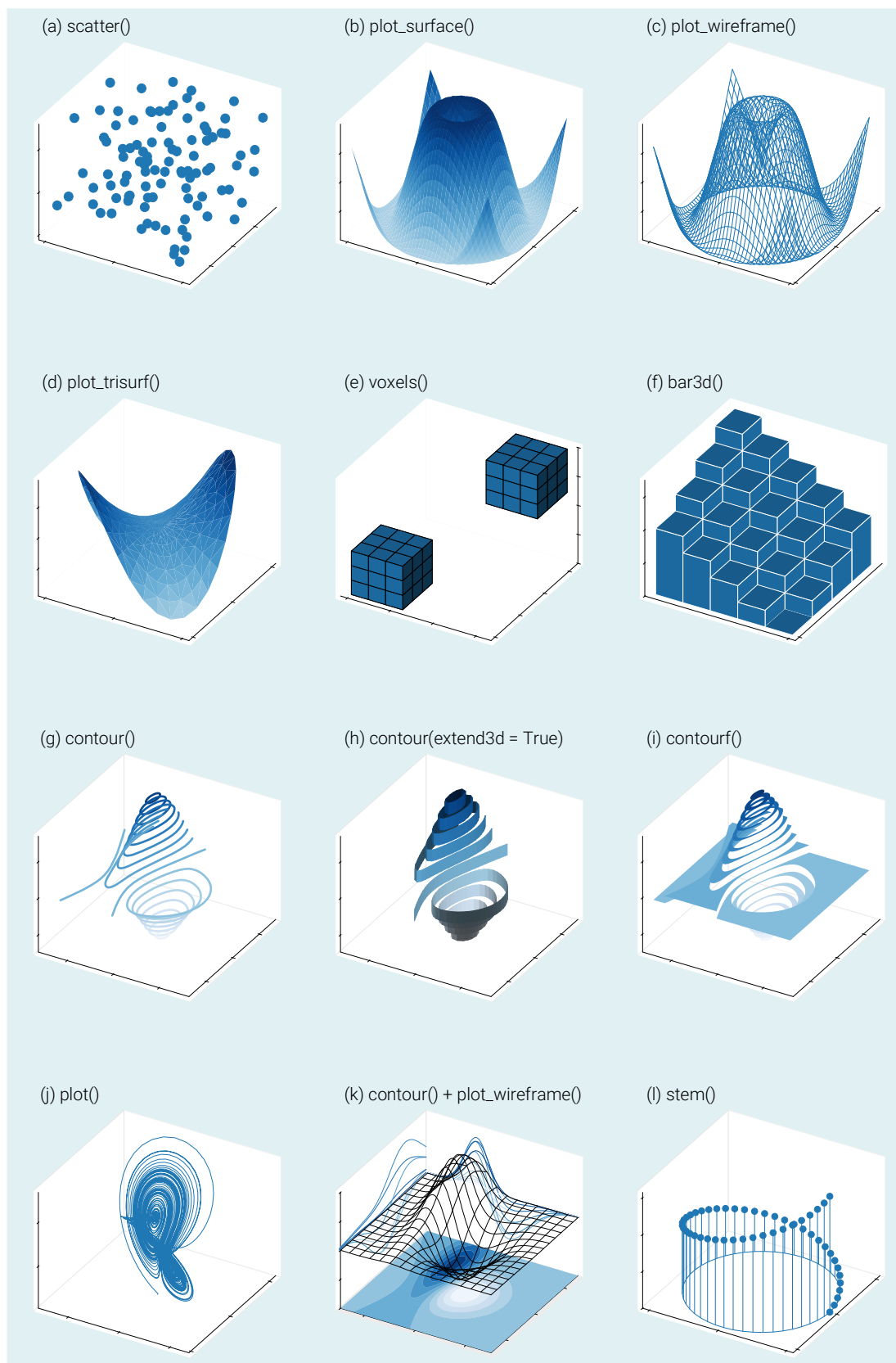


图 23. Matplotlib 常见可视化方案，第三组

整页排版，背景色采用奇数页网格笔记纸背景色，四面出血

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com