



Python 101

聊聊 “巨蟒”

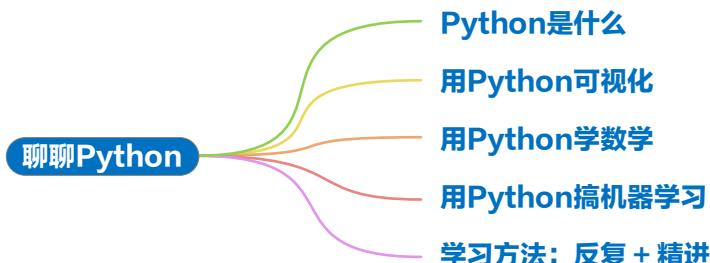
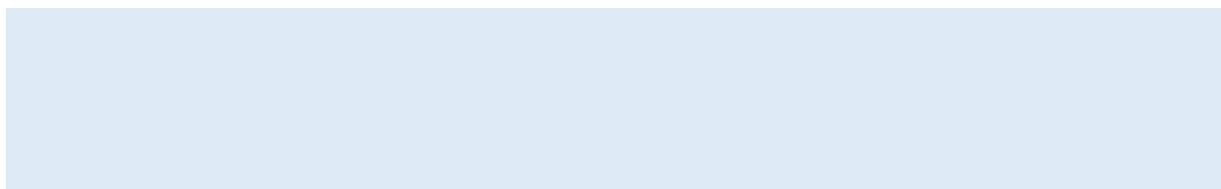
“反复 + 精进”的力量：从加减乘除到机器学习



方悟天地纵横交错，始知万物相生互联。而你我也系其中一环，一念一动皆牵动周身。

There is urgency in coming to see the world as a web of interrelated processes of which we are integral parts, so that all of our choices and actions have consequences for the world around us.

——阿尔弗雷德·怀特海 (Alfred Whitehead) | 英国数学家、哲学家 | 1861 ~ 1947



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

1.1 Python? 巨蟒?

Python 由 Guido van Rossum 于 1991 年正式发布，Python 的首个版本是 0.9.0。

Python 免费开源，语言语法友好，而且社区活跃。Python 的用途极为广泛，特别是在机器学习、深度学习领域。这就是为什么“鸢尾花书”系列会选择 Python 作为编程语言。

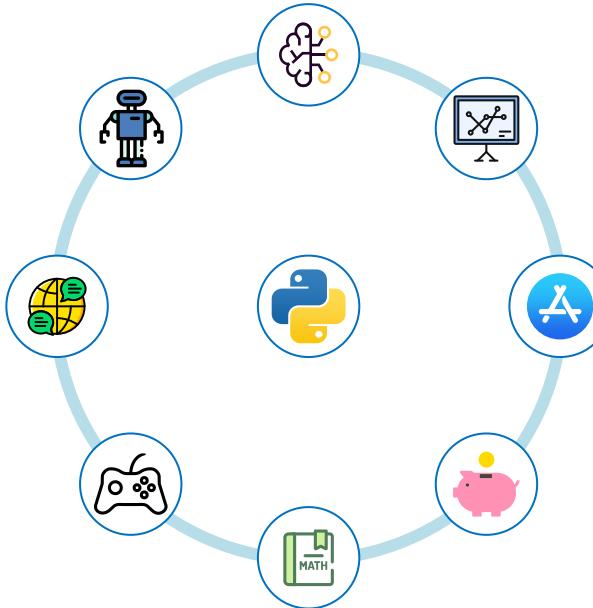


图 1. Python 应用场景

如图 2 所示，Python 的版本持续演进。Python 2.x 和 3.x 系列并存一段时间，但现在 3.x 系列是主要发展方向，建议大家学习时使用最新版本，以便享受最新功能和安全性。

鸢尾花书建议读者通过安装 Anaconda 来安装、管理 Python 环境。下一章会把手教大家如何下载、安装、测试 Anaconda。



什么是 Python?

Python 是一种高级编程语言，使用动态类型系统和自动内存管理。Python 具有简单易学、易于阅读和编写、可移植性强等特点，广泛应用于 Web 开发、数据分析、人工智能、科学计算、自动化等领域。Python 语言具有丰富的标准库和第三方库，可支持各种编程任务，例如文件处理、网络编程、GUI 开发、图像处理、数据可视化、机器学习等。Python 的语法简洁清晰，易于阅读和理解，因此也被广泛应用于教育和科研领域。Python 语言的解释器可用于不同的操作系统，例如 Windows、MacOS、Linux 等，因此 Python 具有很好的跨平台性。

本章很多问题都采用 ChatGPT 答案（有用的废话），作者对回答文本略加编辑。

本书中，ChatGPT 的答案用 标识。建议大家在学习时，不管是概念、程序、数学上遇到问题，都可以使用类似 ChatGPT 工具作为助手。

值得注意的是，在本书创作时，当前版本 ChatGPT 时而废话连篇、胡说八道，请大家注意甄别，切不可不假思索、照单全收。



什么是 ChatGPT?

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

ChatGPT 是一种基于自然语言处理 (Natural Language Processing, NLP) 技术的人工智能 (AI) 应用程序，它是由 OpenAI 公司开发的一种大规模预训练语言模型。ChatGPT 使用深度神经网络来模拟人类的对话过程，它可以理解和生成人类语言，可以用于实现聊天机器人、智能客服、智能助手等应用。ChatGPT 还可以帮助用户进行文本自动生成、文本摘要、文本分类、情感分析等任务。ChatGPT 使用 Python 编程语言进行搭建。在搭建 ChatGPT 时，OpenAI 使用了 Python 的深度学习框架 TensorFlow 和 PyTorch，以及一些其他的 Python 库和工具来完成。

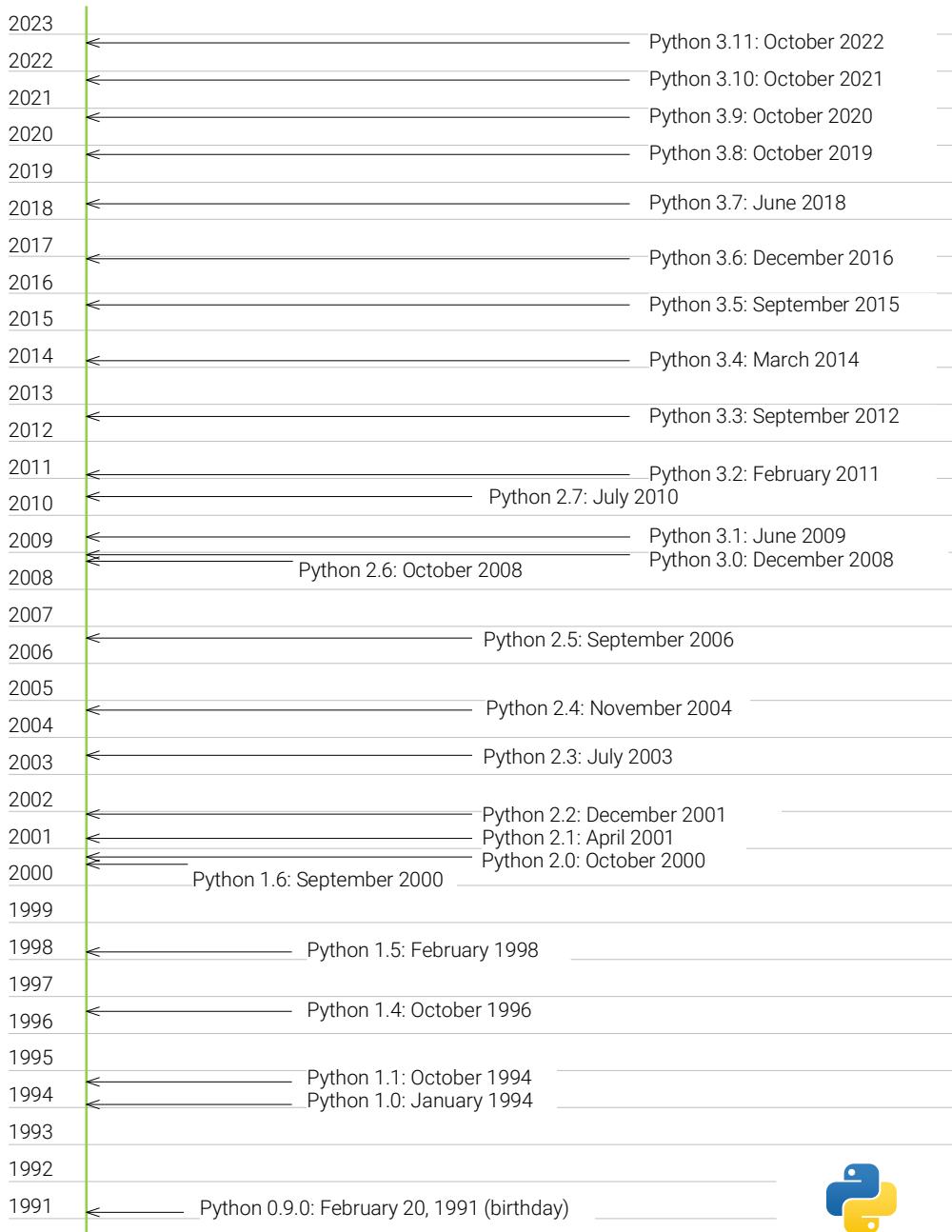


图 2. Python 历史版本时间轴

我们为什么要学 Python?

如图 1 所示，Python 具有广泛的用途，比如。

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

- ▶ 机器学习：Python 在数据科学和机器学习领域非常受欢迎，因为它有很多强大的库和工具，如 NumPy、Pandas、Scikit-learn 等。
- ▶ 深度学习：Python 的深度学习工具，比如 PyTorch、TensorFlow，常用来开发各种人工智能应用，比如智能设备、无人驾驶、自然语言处理工具等。
- ▶ Web 开发：Python 可以用于 Web 开发，有许多流行的 Web 框架，如 Django、Flask 等。
- ▶ 自动化脚本：Python 可以用于自动化任务，例如自动备份、自动化测试、爬虫等。
- ▶ 游戏开发：Python 可以用于游戏开发，如 Pygame 等库和工具。
- ▶ 系统管理和网络编程：Python 可以用于系统管理和网络编程，例如网络爬虫、服务器开发、安全工具等。

整套鸢尾花书用到的主要是 Python 在可视化、数学、数据分析、机器学习方面的工具。图 3 所示为本书涉及到的 9 个重要的 Python 数学运算和可视化库。



图 3. 《编程不难》涉及到的 Python 库

 **Python 中，什么是模块、包、库？**

在 Python 中，模块、包、库是三个常见的概念。它们的含义如下：

模块 (Module)：是一个 Python 程序文件，包含了一组相关的函数、类、变量和常量等，可供其他程序引用。Python 中的模块是一种可重用的代码组件，可用于将相关的代码组织到一起，以便更好地管理和维护代码。一个模块可以包含多个函数、类、变量和常量等，可以被其他模块或程序引用和调用。

包 (Package)：是一组相关的模块的集合，用于组织 Python 代码的层次结构。一个包是一个文件夹，其中包含其他模块或子包。包是一种通过模块命名空间进行模块组织的方式，可用于更好地组织和管理大型代码库。

库 (Library)：是由一组模块和包组成的软件组件，提供了一系列函数、类、变量和常量等，用于解决特定问题。Python 标准库是 Python 官方提供的一组库，包含了大量的模块和功能，可以直接使用。此外，还有第三方库，如 NumPy、Pandas、Matplotlib 等，用于数据处理、科学计算、可视化等领域。

需要注意的是，模块是最小的可重用代码单元，而包和库是由多个模块组成的更大的结构。在 Python 中，通常使用 `import` 语句来引入需要使用的包、库或模块。

本书每个板块的工具

图 4 所示为《编程不难》每个板块涉及的核心工具。这些工具中有些是 Python 基本语法，有些则是 Python 常用包，剩下一些是 Python 编程工具。

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

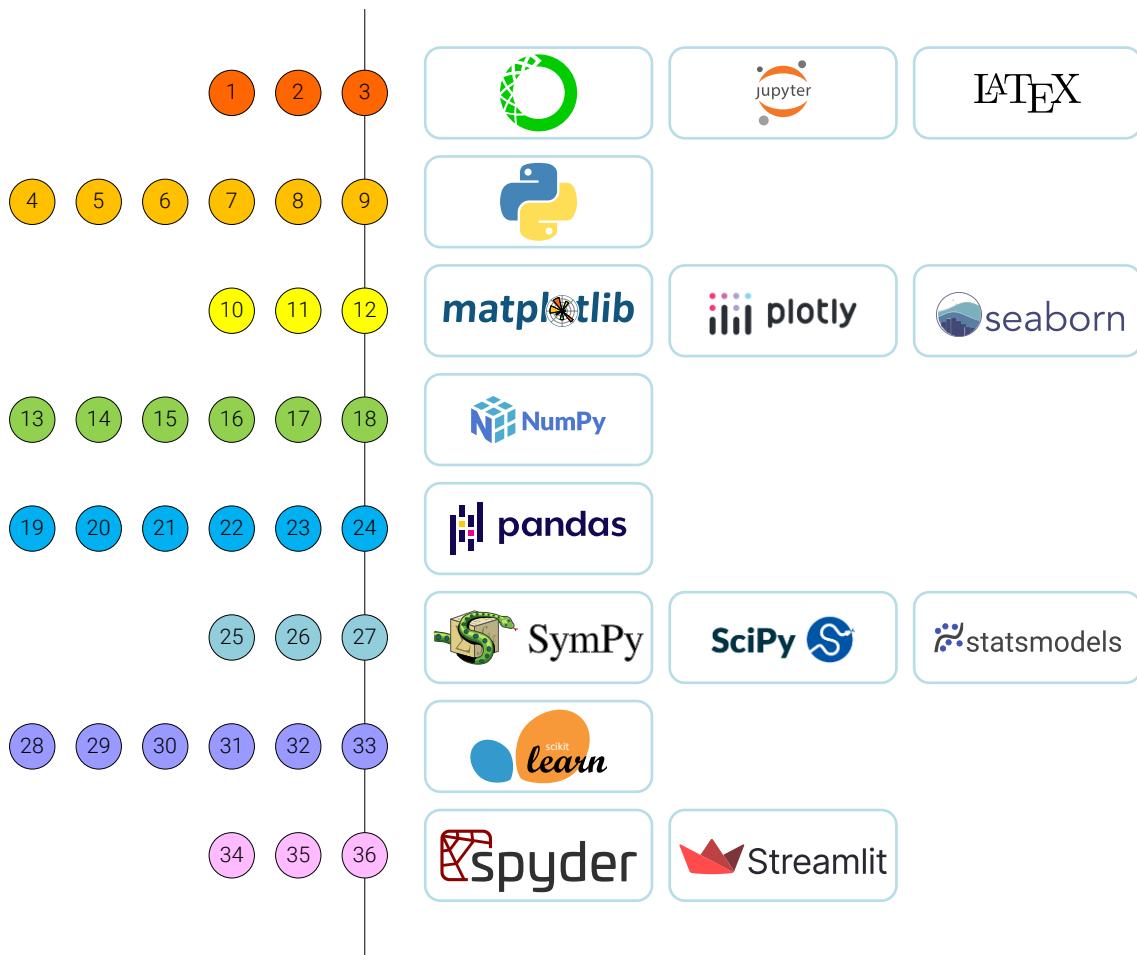


图 4. 《编程不难》每章涉及的核心工具

面向准人工智能时代的教育

作者认为，面向人工智能时代的教育，特别是数学教育，必须结合编程、可视化、实际应用。而 Python 既是编程工具，也拥有大量可视化工具，同时可以用来完成各种数据科学、机器学习任务。

基于这样的考虑，鸢尾花书整套图书在创作时都采用了“编程 + 可视化 + 数学 + 机器学习”这个内核，只不过各个分册的侧重各有不同。

对于初高中生、大学生，学习 Python 有很多好处，比如。

- ▶ 培养编程思维：Python 作为一种编程语言，可以帮助大家培养编程思维能力。大家可以通过编写简单的程序和解决各种问题，锻炼逻辑思考、问题解决和创造力等能力。
- ▶ 高效地学习数学及其他学科：将公式、模型写成 Python 代码的过程，本身就是一种“习题”。而且这类习题比传统课本习题更能激发大家的兴趣。
- ▶ 图形化强化记忆：公式、定理、定义、解题技巧 ... 大家考完试也就忘记了。但是利用 Python 编程，把公式、定理、定义变成一幅幅活生生的图形之后，这些概念将会深深地刻在大家脑中，甚至一辈子不会忘记。

- ▶ 提高学习效率：Python 可以用于自动化各种重复性的任务，如数据处理、文本处理等。大家可以通过编写 Python 程序来自动化这些任务，从而节省时间和精力，提高学习效率。
- ▶ 为未来的学习和职业做准备：Python 是一种非常流行的编程语言，它在数据科学、机器学习、人工智能等领域有很多应用，大家可以通过学习 Python，为未来的学习和职业做准备，提前掌握一些必要的技能和知识。
- ▶ 增强实践能力：学习 Python 可以让大家更容易地将学到的知识应用到实践中，例如编写简单应用程序、游戏、网站等，这有助于大家增强实践能力和探索新领域的能力。
- ▶ 培养团队合作意识：学习 Python 可以让大家更容易地与他人合作，例如在编写程序时可以分工合作、交流想法等，这有助于培养团队合作意识和沟通能力。

未来已来，你来不来？

怎么学 Python？

和汉语、英语一样，Python 也是语言。只不过，Python 是人和机器交互的语言。Python 的语法有绝对的对错，不能模棱两可。错误的命令，要么出 bug，要么得出错误的结果。

我相信鸳尾花书的读者没有任何人是拿字典学会汉语的，同样别指望拿着 Python 语法书学好编程。

在学习 Python 的过程中，大家一定要吸取英语教育的教训。很多人之所以在学了十几年的英语之后仍然不会说，就是因为他们只注重死记硬背词汇、语法，而没有真正把语言用起来，进行口头和书面交流。

因此，在学习 Python 时，大家不妨“功利”一些，边学边练——马上开始编码 code 和调试 debug。先写几段代码，跑起来；现在暂时用不到的语法细枝末节先放到一边。

千万别把 Python 编程当文科来学！语法、指令、函数、逻辑等等，记不住，不要紧！用多了，就好了。无他，但手熟尔。

鸳尾花书有编程、可视化、数学、机器学习这几个重要的元素，而这几个元素都离不开 Python。下面我们逐一聊一下。



学习 Python 和学习英语有什么相似之处？

学习 Python 和学习英语有一些相似之处，以下是其中一些：

都需要掌握基础知识：学习 Python 和学习英语都需要掌握基础知识，Python 的基础语法、数据类型、流程控制语句、函数等，英语的基本词汇、语法、发音等。

都需要不断练习：学习 Python 和学习英语都需要不断地练习，Python 需要编写程序来实践，英语需要口语练习和写作练习。

都需要实践和应用：学习 Python 和学习英语都需要不断地实践和应用，Python 可以应用到数据处理、人工智能、游戏开发等领域，英语可以应用到国际交流、留学、工作等方面。

都需要耐心和坚持：学习 Python 和学习英语都需要耐心和坚持，需要花费大量时间和精力来学习和练习，才能达到良好的掌握和应用水平。

总之，学习 Python 和学习英语都需要掌握基础知识、不断练习、实践和应用，同时也需要耐心和坚持。虽然二者是不同的领域，但都是对自己未来发展非常有帮助的技能。

1.2 Python 和可视化有什么关系？

Python 和可视化有很密切的关系。Python 中有很多强大的可视化库和工具，可以帮助用户对数据进行可视化呈现。大家翻看鸢尾花书的任何一册，会发现大量彩图，其中绝大部分都是用 Python 编码生成。

以下是 Python 和可视化的一些关系。

- ▶ 数据可视化：Python 中有许多数据可视化的库，例如 Matplotlib、Seaborn、Plotly 等，可以帮助用户将数据可视化呈现出来，从而更好地理解数据的分布、趋势等信息。本书的绘图部分将蜻蜓点水地讲解 Matplotlib、Seaborn、Plotly 常用绘图命令。“鸢尾花书”的《可视之美》一册将专门讲解数据可视化这一话题。
- ▶ 图像处理：Python 中有许多图像处理的库，例如 OpenCV 等，可以帮助用户进行图像处理和分析，同时也可将处理后的图像进行可视化呈现。
- ▶ 交互式可视化：Python 中也有许多用于交互式可视化的库，例如 Bokeh、Altair 等，可以帮助用户建立交互式的数据可视化应用程序。
- ▶ 3D 可视化：Python 中也有许多用于 3D 可视化的库，例如 Mayavi、VisPy 等，可以帮助用户对三维数据进行可视化呈现。

1.3 Python 和数学有什么关系？

Python 和数学有着密切的关系。Python 是一种非常适合数学建模和数据分析的编程语言，拥有大量的数学计算库和工具。

以下是 Python 和数学的一些关系。

- ▶ 数学计算：Python 中有很多用于数学计算的库和工具，例如 NumPy、SciPy 等，可以帮助用户进行矩阵运算、微积分、最优化、统计分析等数学计算任务。
- ▶ 数据分析：Python 中有很多用于数据分析的库和工具，例如 Pandas、Matplotlib、Seaborn 等，可以帮助用户对数据进行统计分析、可视化呈现等。
- ▶ 数学建模：Python 中还有很多用于数学建模的库和工具，例如 SymPy 等，可以帮助用户进行数学建模和优化任务。
- ▶ 教学和研究：Python 也被广泛应用于数学教学和研究领域，例如用 Python 实现数学实验、数学模型的探索、算法的实现等。

以二元高斯分布为例

下面给大家举个例子。下式是大名鼎鼎的**二元高斯分布** (bivariate Gaussian distribution) **概率密度函数** (Probability Density Function, PDF)。

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

如果大家在这之前没有接触过这个式子，不要紧！

大家仅仅需要知道二元高斯分布不仅仅是概率统计的重要知识点，也和**几何** (geometry)、**微积分** (calculus)、**线性代数** (linear algebra) 有关，更是机器学习各种算法的常客。鸢尾花书会在本册以及其余分册中以各种视角帮大家剖析这个式子。

下面，我们来聊聊 Python 编程对理解这个“让人头大”式子有什么帮助。

$$f(x, y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho_{x,y}^2}} \exp\left(-\frac{1}{2(1-\rho_{x,y}^2)} \left[\left(\frac{x-\mu_x}{\sigma_x}\right)^2 - 2\rho_{x,y} \left(\frac{x-\mu_x}{\sigma_x}\right)\left(\frac{y-\mu_y}{\sigma_y}\right) + \left(\frac{y-\mu_y}{\sigma_y}\right)^2 \right]\right)$$

首先，借助 NumPy 之类的 Python 库，我们可以自己能写代码计算上述函数的数值。更方便的是，SciPy 库就有二元高斯分布现成的函数。当然，自己编码自定义函数肯定印象更深刻。

然后，利用 Matplotlib 等可视化工具，我们可以“看见”这个函数，如图 5 所示。大家可能惊奇地发现，等高线呈现的形状是一组同心椭圆！

大家很快就会发现，这个椭圆和**线性回归** (linear regression)、**主成分分析** (principal component analysis) 有直接关系。

如图 6 和图 7 所示，我们还可以看到不同参数对这些图形的影响。

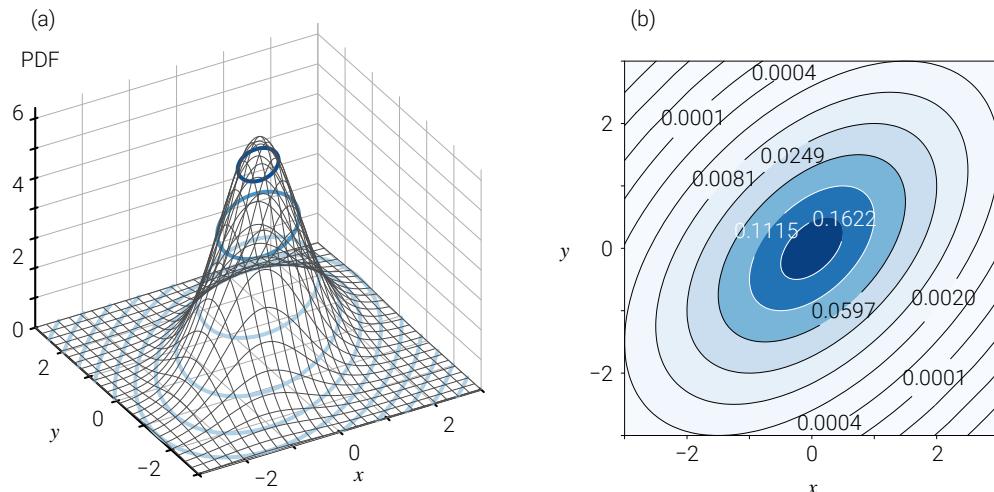
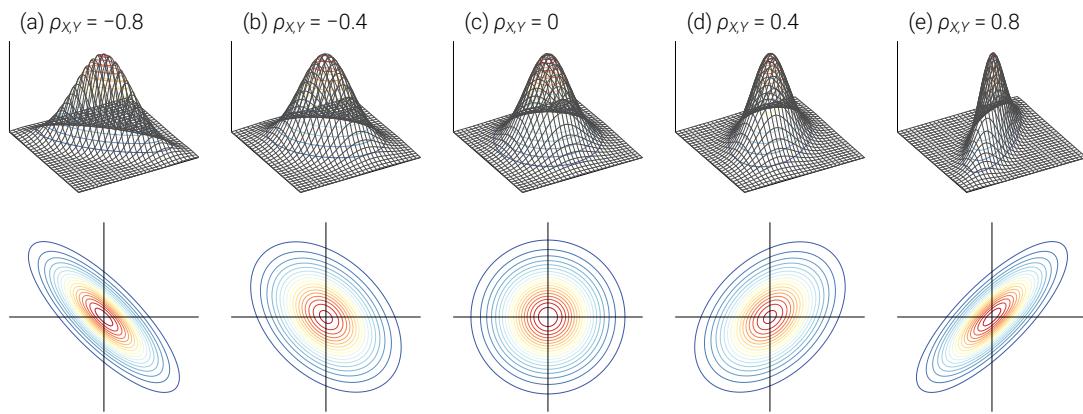
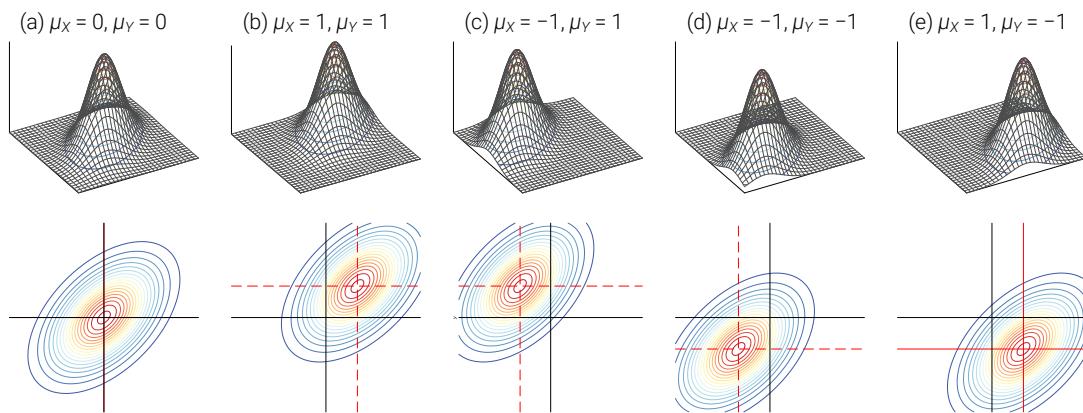


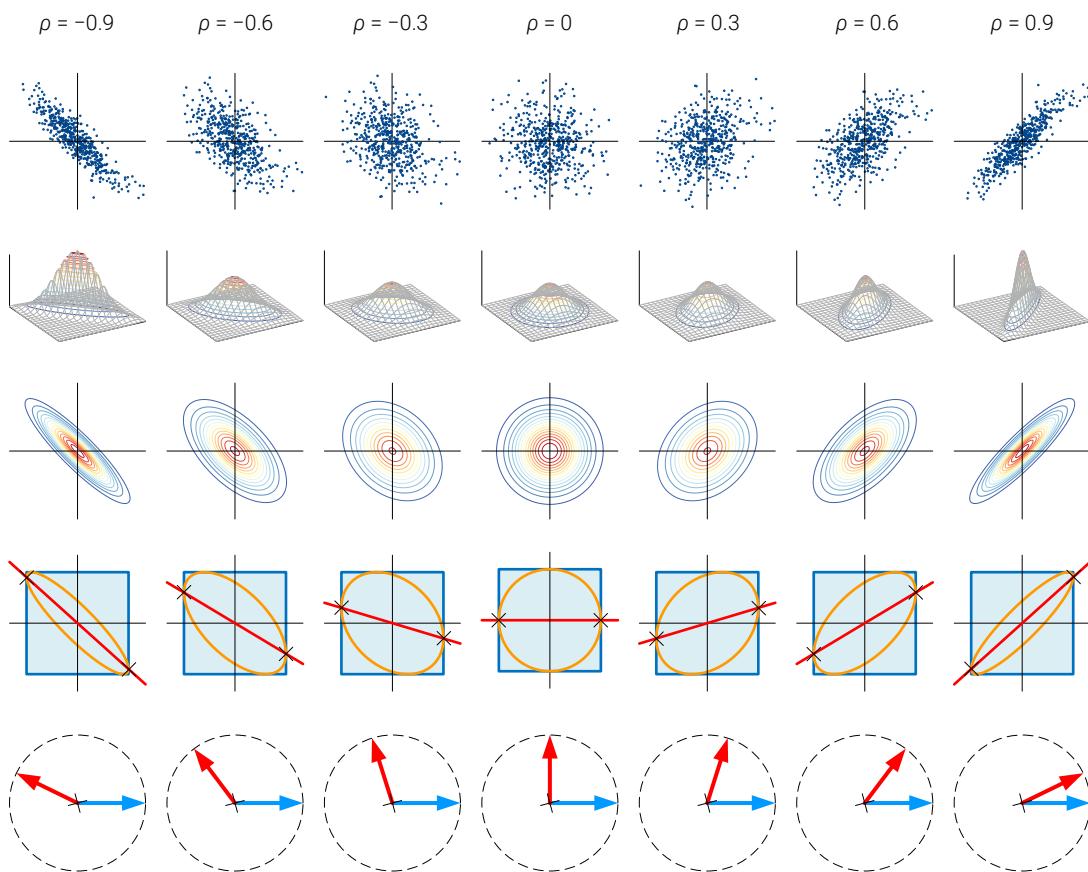
图 5. 一组特定参数下的二元高斯分布概率密度函数

图 6. 不同相关性系数 $\rho_{X,Y}$, 二元高斯分布 PDF 曲面和等高线, $\sigma_x = 1$, $\sigma_y = 1$ 图 7. 不同质心位置 (μ_x, μ_y) , 二元高斯分布 PDF 曲面和等高线, $\sigma_x = 1$, $\sigma_y = 1$, $\rho_{X,Y} = 0.4$

多视角

“可视化”在鸢尾花书系列每一册都是重头戏。因为大家很快就会发现，可视化让很多困扰我们多年的问题迎刃而解。不同的可视化方案就像是一束束光从不同角度射向同一个问题，这些丰富的视角可以帮助我们更深入地理解同一个问题。

举个例子，图 8 所示为几个不同视角理解**相关性系数**（Pearson Correlation Coefficient, PCC）。

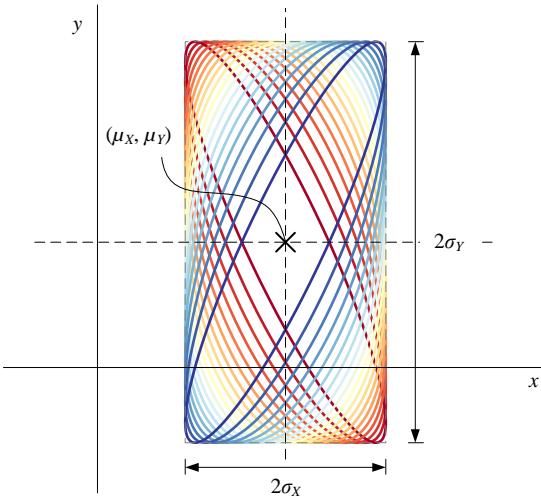
图 8. 相关性系数 $\rho_{X,Y}$ 的几种可视化方案

一组有趣的椭圆

类似地，有了 Python 这个工具，我们可以解剖上述函数。比如，图 9 展示如下等式和一组有趣的椭圆有关。这组椭圆都和同一矩形四个边相切，而这个矩形又和二元高斯分布的参数直接相关。

利用 Python 可视化，我们可以清楚地看到这一点。更重要的是，这个性质又和**条件概率分布** (conditional probability distribution)、**线性回归** (linear regression) 密不可分。

$$\frac{1}{(1-\rho_{X,Y}^2)} \left[\left(\frac{x-\mu_X}{\sigma_X} \right)^2 - 2\rho_{X,Y} \left(\frac{x-\mu_X}{\sigma_X} \right) \left(\frac{y-\mu_Y}{\sigma_Y} \right) + \left(\frac{y-\mu_Y}{\sigma_Y} \right)^2 \right] = 1$$

图 9. 椭圆和中心在 (μ_x, μ_y) 长 $2\sigma_x$ 、宽 $2\sigma_y$ 的矩形相切

马氏距离

给上述等式开个平方根，令其为 d ，我们便得到机器学习中大名鼎鼎的**马氏距离** (Mahalanobis distance)！

$$d = \sqrt{\frac{1}{(1 - \rho_{x,y}^2)} \left[\left(\frac{x - \mu_x}{\sigma_x} \right)^2 - 2\rho \left(\frac{x - \mu_x}{\sigma_x} \right) \left(\frac{y - \mu_y}{\sigma_y} \right) + \left(\frac{y - \mu_y}{\sigma_y} \right)^2 \right]}$$

图 10 所示为一组马氏距离等距线，我们立刻发现了椭圆的存在。

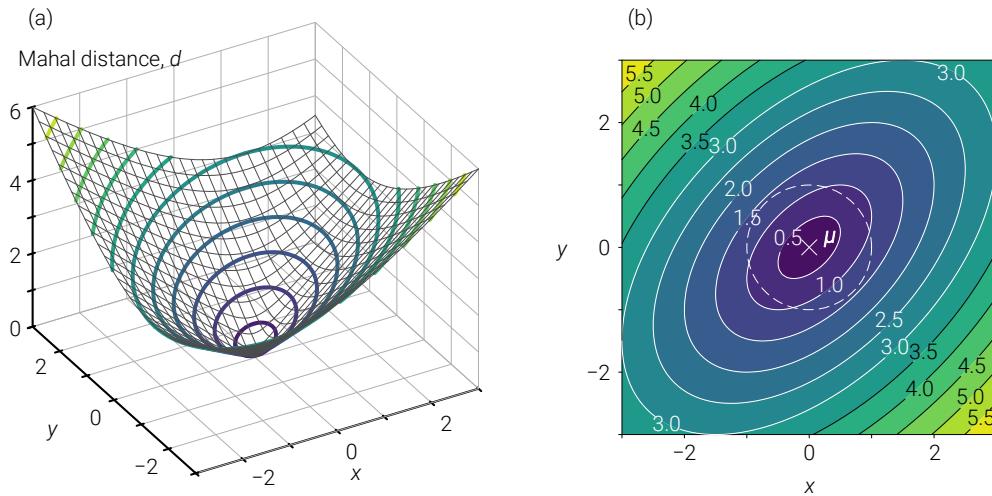


图 10. 马氏距离椭圆等高线

欧氏距离 (Euclidean distance) 就是我们经常说的两点之间线段。而不同于欧氏距离，马氏距离考虑了数据的分布形状。从图 11 中，我们可以看到马氏距离等距线一层层紧紧地包裹着样本散点数据。

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 5 和图 11 的椭圆几何角度存在很多差异，但是两者又存在紧密联系。而两者的联系就是**高斯函数** (Gaussian function)。高斯函数是微积分的重要研究对象之一，也是机器学习各种算法的熟客。

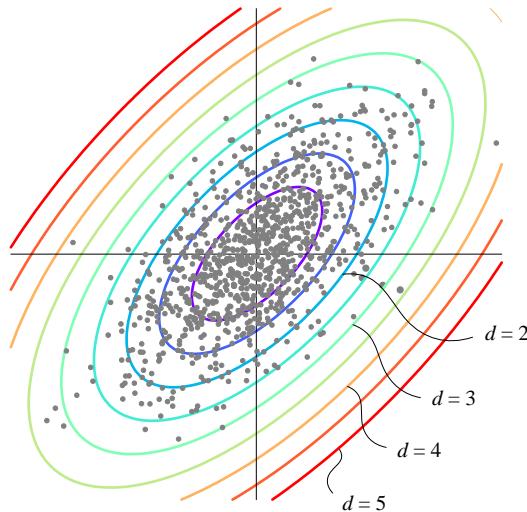


图 11. 马氏距离等距线

几何变换

如图 12 所示，想要更深入理解马氏距离，我们需要借助几何视角，比如**平移** (translation)、**旋转** (rotation)、**缩放** (scaling)。

大家可能会好奇，到底旋转多少角度、缩放多大比例？

想要回答这个问题，这就需要祭出线性代数大杀器——**特征值分解** (Eigen Value Decomposition, EVD)。



鸢尾花书《矩阵力量》会专门介绍特征值分解，现在大家有个印象就好。

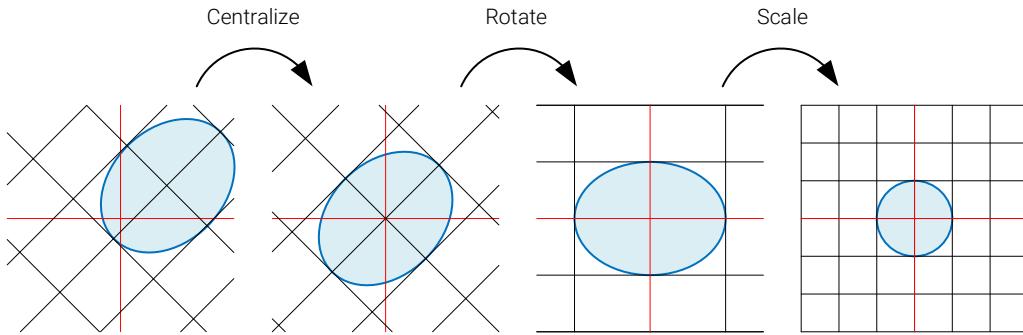


图 12. 通过几何变换理解马氏距离：平移 → 旋转 → 缩放

用 Streamlit 做应用 App

如果大家还觉得不过瘾，《编程不难》最后还介绍如何用 Streamlit 制作如图 13 所示 App。这个 App 采用交互形式让大家更加清楚地理解各种参数对二元高斯分布的影响。

在数学知识可视化方面，3Blue1Brown 绝对是学霸！他们开发的 Python 数学动画工具 manim 更是很多知识类博主的利器。

但是，几经权衡还是没有把 manim 纳入鸢尾花书体系。主要原因是，manim 更适合制作知识类分享视频，代码可迁移性差。哪怕鸢尾花书提供一些用 manim 制作的动画，同学们也是被动观看，不可能主动参与到编程实践中。

鸢尾花书系列考虑再三最后采用了 Streamlit。Streamlit 不但可以做交互式数学演示，还可以做数据分析、机器学习 App。

大家会在鸢尾花书各个分册经常看到用 Streamlit 做的各种应用 App。这些 App 一方面帮大家理解各种数学工具、算法逻辑，还可以帮大家学会用 Streamlit 快速搭建可交互应用 App。

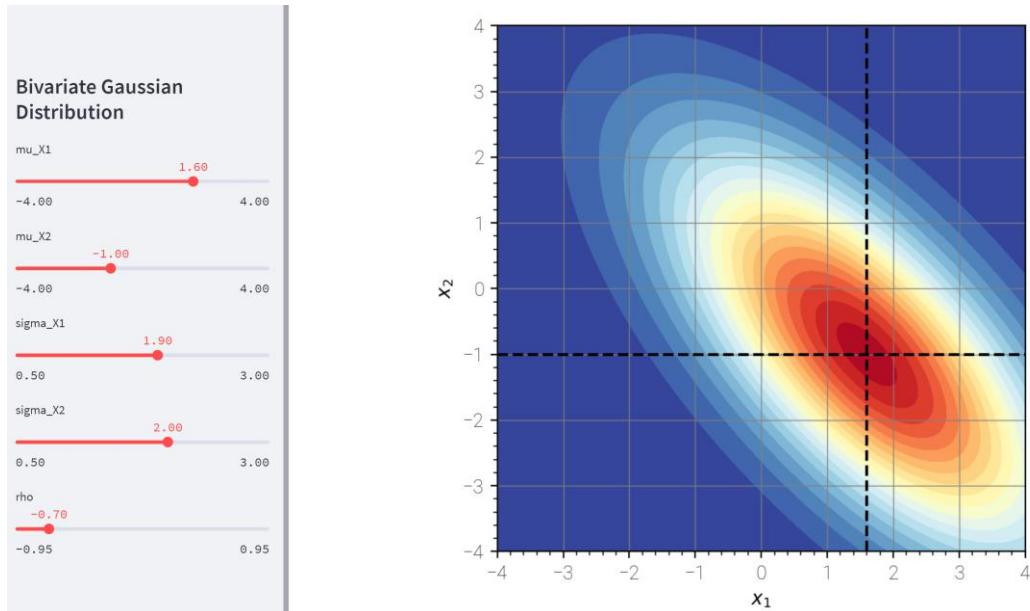


图 13. 二元高斯分布 App, Streamlit 创建

三元、多元高斯分布

大家可能会问，有了一元、二元高斯分布，就肯定有三元，乃至多元高斯分布 (multi-variate Gaussian distribution)。Python 能帮助我们理解这些高斯分布吗？

答案是肯定的！

这就需要我们进一步借助各种数学工具和可视化手段继续升维！如图 14 所示，三元高斯分布就变成了椭球！

而这些椭球在平面的投影得到椭圆，对应的就是二元高斯分布。这些都是借助 Python 这个工具达成知识“升维”！

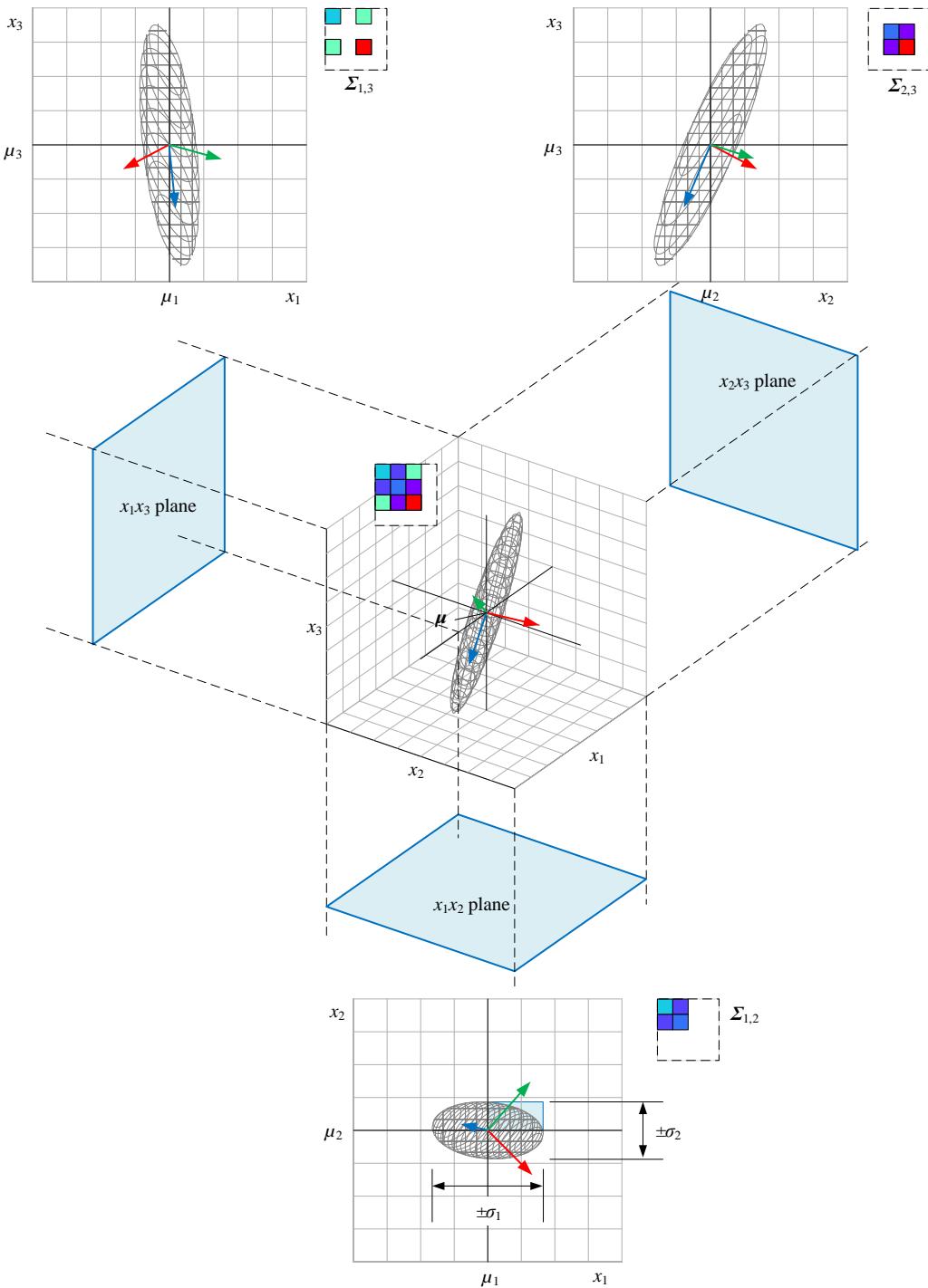


图 14. “旋转”椭球投影到三个二维平面

看到这里，大家如果觉得有点吃不消，不要怕。一步一个脚印，对于 Python 零基础的读者，请先耐心读完本册《编程不难》和下一册《可视化之美》。

紧接着，鸢尾花书“数学三剑客”给大家提供了大量的“编程 + 可视化”方案来帮大家深入理解这些数学工具。

通过上述例子，大家可能已经发现 Python 对于学习数学的意义。“鸢尾花书”整个系列丛书希望给大家提供一个学习、理解、掌握、应用数学工具的全新路径。

用习题集学习数学给大家养成一个坏习惯——期待标准答案，指望解题技巧！

而真实世界面对的各种问题根本没有标准答案，也不存在什么阶梯技巧。大家需要利用“编程 + 可视化 + 数学 + 机器学习”自主探索。因此，培养大家的自主探究学习能力也是鸢尾花书的目的之一，这就是为什么我们要在整套书都引入 JupyterLab 作为学习平台的原因。

1.4 Python 和机器学习有什么关系？

Python 与机器学习有非常密切的关系。Python 是一种简单易学、可读性强的编程语言，同时也拥有丰富的第三方库和工具，这使得 Python 成为机器学习领域的重要工具之一。

机器学习是一种应用人工智能的技术，通过让计算机从数据中学习并改善性能，来实现对未知数据的预测和决策。

Python 在机器学习领域的应用非常广泛，主要有以下几个方面。

- ▶ 数据处理和分析：Python 中有许多用于数据处理和分析的库，例如 Pandas、NumPy 和 SciPy，这些库能够帮助用户轻松地处理和分析数据。
- ▶ 机器学习框架：Python 中也有许多用于机器学习的框架，例如 TensorFlow、PyTorch 和 Scikit-Learn 等，这些框架可以帮助用户更加高效地进行机器学习建模和预测。
- ▶ 可视化工具：Python 中的 Matplotlib 和 Seaborn 等可视化库，可以帮助用户更加清晰地理解数据和模型，以及呈现结果。
- ▶ 自然语言处理：Python 中的自然语言处理库，例如 NLTK 和 Spacy 等，可以帮助用户进行文本数据的处理、分析和预测。



什么是机器学习？

机器学习是一种人工智能技术，它使计算机系统能够通过数据和经验自主学习和改进，而无需显式地编程指令。简单来说，机器学习是通过训练算法从数据中学习模式和规律，然后利用这些模式和规律来进行预测或决策。在机器学习中，模型是通过训练算法从大量数据中学习而来的，这些数据被称为训练数据集。训练数据集包含已知结果的输入输出对，这些输入输出对用于训练模型来预测未知数据的输出。训练数据集中的数据越多，训练时间越长，模型就越准确。机器学习可以应用于各种领域，例如语音识别、图像识别、自然语言处理、推荐系统和金融分析等。它已成为当今科技领域中最热门和最具前途的领域之一。

当然，不管是数据分析，还是机器学习，我们到处都可以看到各种各样的数学工具。

还是以高斯分布为例，我们可以在很多算法中都看到高斯的名字，比如**高斯朴素贝叶斯**（Gaussian Naive Bayes）、**高斯判别分析**（Gaussian discriminant analysis）、**高斯过程**（Gaussian process）、**高斯混合模型**（Gaussian mixture model）等等。

1.5 相信“反复 + 精进”的力量！

反复，不是机械重复，不是当一天和尚撞一天钟。而是在反复中，日拱一卒，不断精进！

鸢尾花书几乎所有的知识点都是采用这种“反复 + 精进”的模式编写的。比如，大家会在鸢尾花书的几乎每一分钟都看到“回归分析”的影子。

下面，我们就以“回归分析”为例聊聊“反复 + 精进”的力量！

一组散点

图 15 所示平面有一组散点。从数据角度，我们面对的无非就是两列数字。把每行看成坐标画在平面直角坐标系上便得到平面散点图。这幅图中，我们似乎看到了某种“线性”关系。

换个角度，图 15 这个简简单单的散点图也让我们看到了可视化的力量。通过各种可视化方案，我们可以呈现数据、发现规律。然后再用各种数学工具来量化分析这些可能存在的关系。

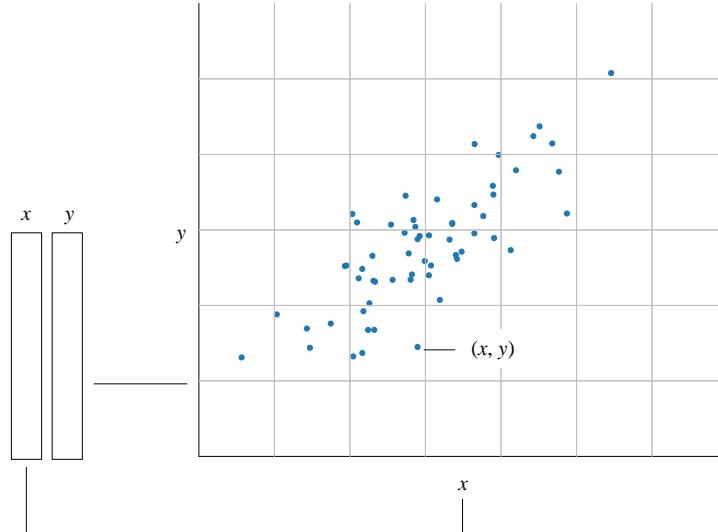


图 15. 平面散点图

画一条直线

然后，利用 Python 第三方库函数，比如：

- ▶ SciPy (`scipy.stats.linregress`),
- ▶ Statsmodels (`statsmodels.regression.linear_model.OLS`),
- ▶ Scikit-Learn (`sklearn.linear_model.LinearRegression`)。

我们可以很轻松获得图 16 这条一元线性回归直线。调用 Python 的各种包完成计算的过程简称“调包”。

从代数角度来看，这条直线不过就是一元一次函数。它的两个重要参数可以是斜率（slope）和截距（intercept）。

我相信所有读者在初中阶段一定接触过一元一次函数。这个函数看上去简单，但是在数据分析、机器学习领域却很实用。

我们通过“调包”的确获得了图 16 这条直线，计算得到了它的斜率和截距。但是，希望鸢尾花读者能够多问几个问题。比如，图 16 这条直线怎么算出来的？用到了什么数学工具？怎么评价这个回归结果的好坏？

问这些问题有很多好处。

第一，机器学习模型不是黑盒子。只有能合理解释的模型才让人信服，想调参训练模型的话，就必须要了解其背后的数学原理和算法逻辑。调包虽然方便，但有时会导致对模型的理解不足。这种情况下，无法解释模型的决策过程，无法识别模型的潜在偏差或不适用性。了解数学背后的工具和算法可以帮助你理解模型的内部机制，提高模型的可解释性

第二，机器学习的算法层出不穷。知道数学工具和算法逻辑的局限性和适用性有助于选择合适的算法，避免不必要的错误。

第三，很多时候标准模型不可能解决你的“定制化”问题，我们常常需要根据问题的具体特征改进、创新算法。

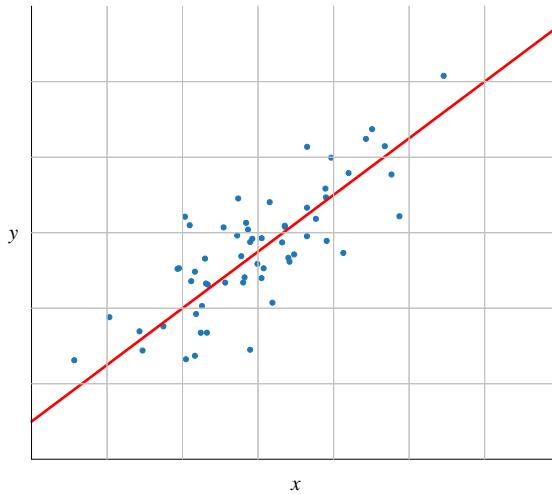


图 16. 平面上，一元线性回归

一个优化问题

对于一元线性回归，我们可以利用最小二乘法（Ordinary Least Square, OLS）来求解模型参数。

简单来说，最小二乘法的核心思想是通过最小化观测数据与模型预测值之间的残差平方和来找到最优的模型参数。图 17 利用线段展示残差项。

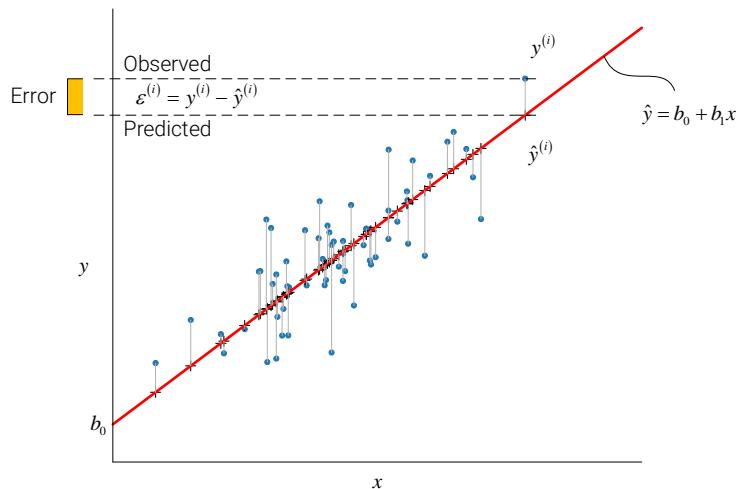


图 17. 一元线性回归中的残差项

利用可视化方案，残差的平方和就更容易理解了。如图 18 所示，残差的平方和无非就是图中所有正方形的面积之和。这样，又“精进”一步，我们就把代数和几何联系到了一起。

最小二乘法 OLS 就是找到最合适的一元一次函数斜率和截距让这些正方形的面积之和最小。想要解决这个问题，我们就需要微积分和优化方面的知识。

“日拱一卒”，在扩充自己数学工具箱工具的同时，我们也发现了看似割裂的数学板块，实际上并不是一个个孤岛，它们之间有着千丝万缕的联系。

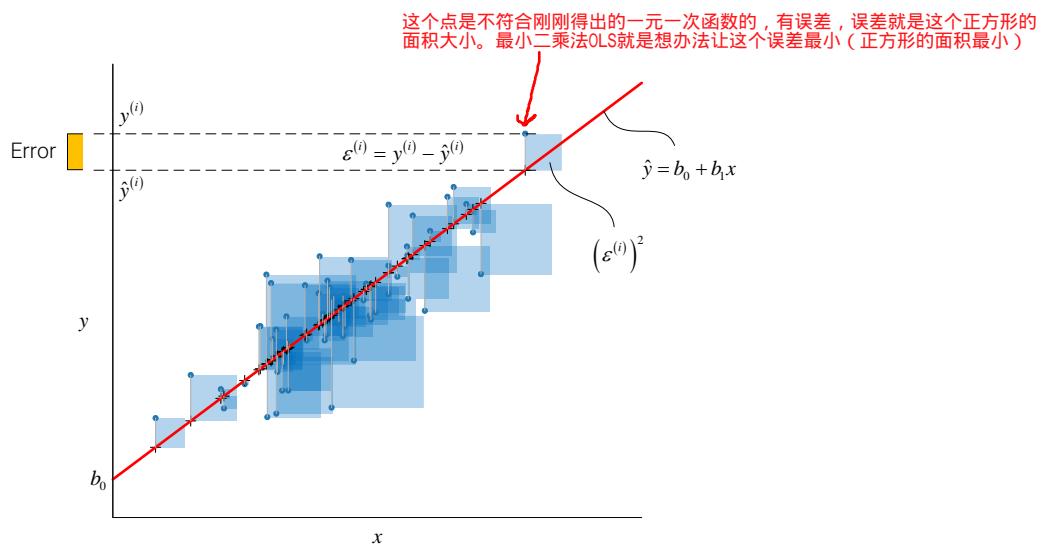


图 18. 残差平方和的几何意义

线性代数

然而，这个“精进”过程远未结束！有了线性代数这个数学工具，我们还能从投影这个视角理解一元线性回归问题，具体如图 19 所示。在线性代数这个百宝箱中，和回归分析相关的数学工具简直不胜枚举，比如范数、超定方程组、伪逆、QR 分解、SVD 分解等等。

线性代数实在太有用，但是很多读者却又学不好！因此从第一册《编程不难》开始，鸢尾花书便不厌其烦地在各个板块见缝插针讲解线性代数知识。

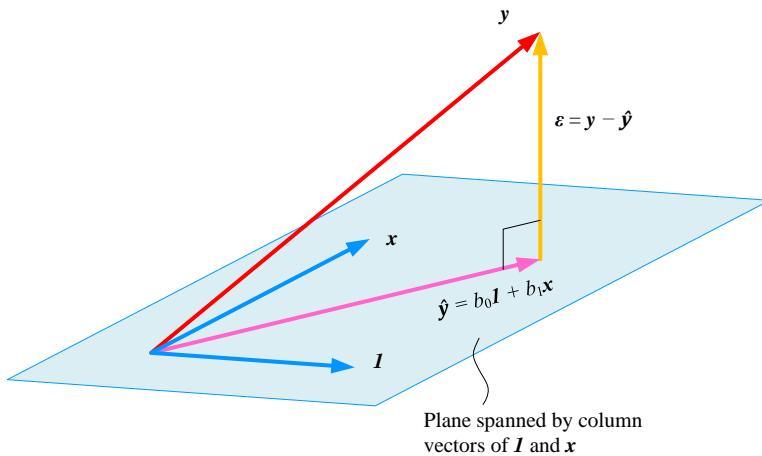


图 19. 几何角度解释一元最小二乘结果，二维平面

概率统计

概率统计怎么能缺席回归分析！如图 20 所示，在《统计至简》中，大家很快就会发现我们还可以从条件概率角度理解线性回归。

大家如果利用 `Statsmodels` 库中函数完成线性分析，一定会看到方差分析 ANOVA、拟合优度、 F 检验、 t 检验等等这些概念。简单来说，这些数学工具从不同角度告诉我们一个线性回归模型的好坏。

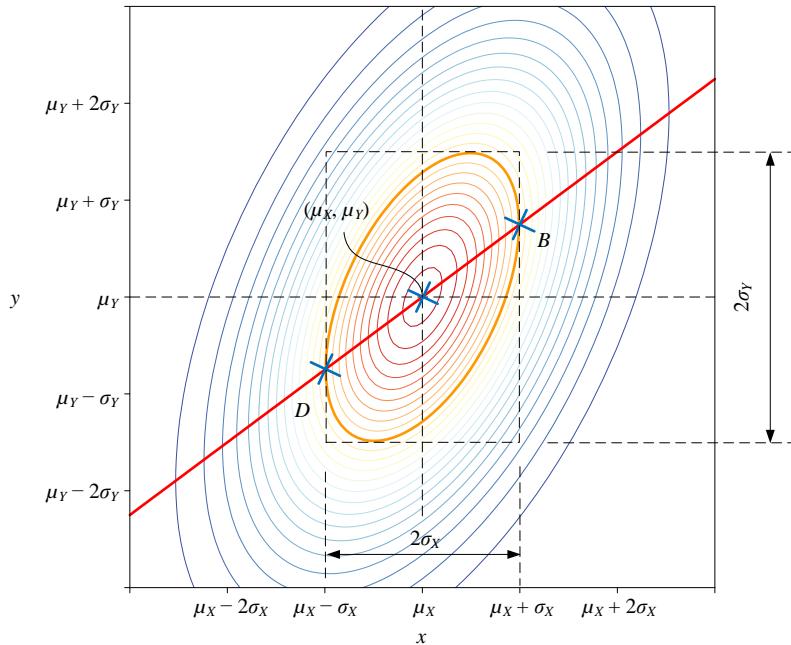


图 20. 从条件期望角度理解线性回归

贝叶斯派

谈到概率统计，怎么能少了贝叶斯派？！

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

统计推断有两大学派——**频率学派推断** (Frequentist inference) 和**贝叶斯学派推断** (Bayesian inference)。

频率学派认为真实参数确定，但一般不可知。真实参数就好比上帝视角能够看到一切随机现象表象下的本质。

贝叶斯学派则认为参数本身也是不确定的，参数本身也是随机变量，因此也服从某种概率分布。很重要的是，贝叶斯派可以引入我们自身经验，是一种“经验 + 数据”的学习模式，类似人脑原理。

用到一元线性回归上，在**贝叶斯派视角下**，我们看到的是图 21 这幅图景。**一元线性回归不再是“一条直线”，而是无数可能直线中的某一条**。有了这个视角，我们的数学工具箱、机器学习工具箱再次装备新工具！
——**一堆一元一次函数的集合？**

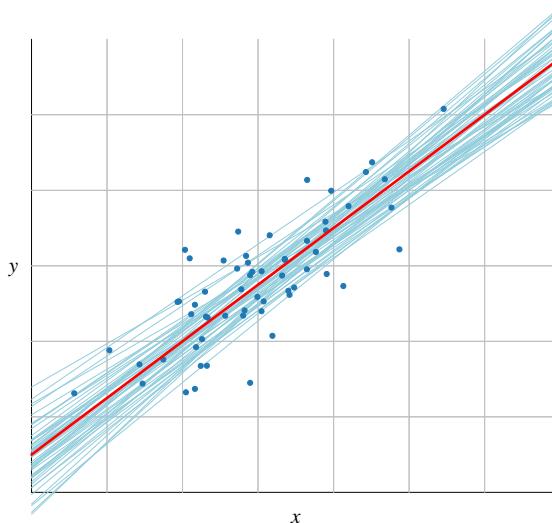


图 21. 贝叶斯统计视角下看线性回归

二元线性回归

我们还可以继续“升维”，将一元线性回归分析提升到如图 22 所示的**二元线性回归** (bivariate linear regression)。这时，我们看到的就不再是一条直线，而是一个平面。而代数角度来看，这个平面是一个二元一次函数。

当然，如果二元线性回归不满足需求，我们还可以进一步升维到**多元线性回归** (multi-variate linear regression)。处理这些高维度的回归模型，线性代数工具从未缺席。

但是，不断引入变量会导致模型过于复杂，从而引发过拟合问题。简单来说，如果一个模型在训练数据表现很好，但是在新数据上表现糟糕的话，这就是一个典型的**过拟合** (overfitting) 问题。

我们可以引入线性代数中的范数工具，即**正则化** (regularization)，来解决过拟合问题。

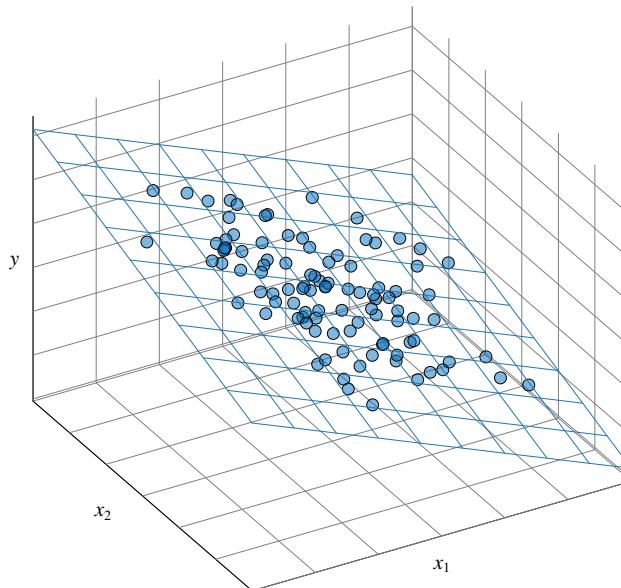


图 22. 二元线性回归

非线性回归

实际应用中，我们会发现很多变量的关系还可能是“非线性”！这时，我们就需要比一次函数更复杂的模型，比如图 23 所示的**多项式回归**（polynomial regression）模型，再如图 24 所示的**逻辑回归**（logistic regression）模型。

我们发现，那些形状千奇百怪的函数原来都有自己的用武之地！

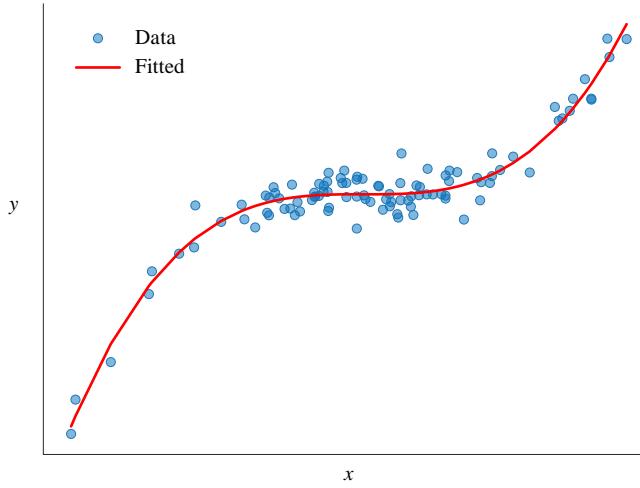


图 23. 多项式回归模型

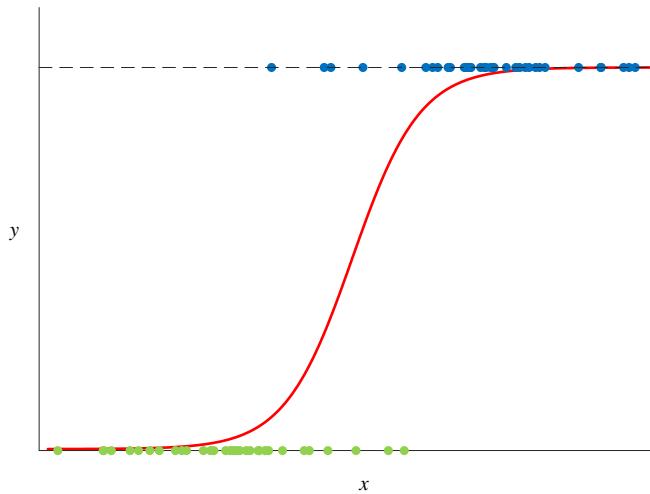


图 24. 逻辑回归模型

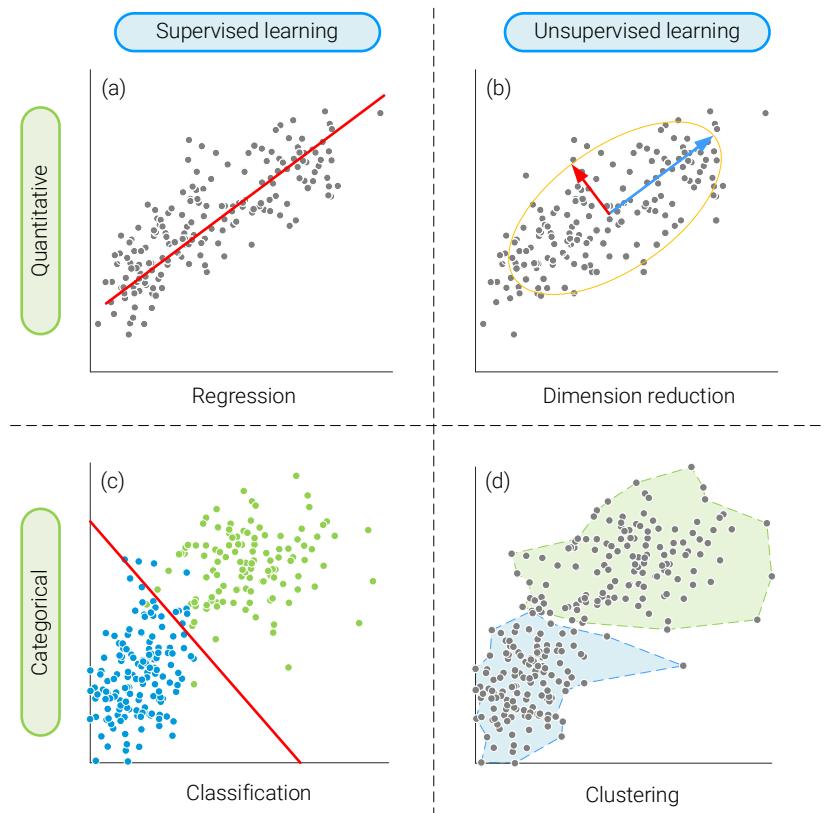
机器学习视角下的回归

读到这里，我们有必要提醒自己一下，回归到底是什么？

图 25 告诉我们，机器学习主要包括两大类问题——**有监督学习** (supervised learning) 和**无监督学习** (unsupervised learning)。

如图 25 所示，站在机器学习的角度来看，回归是有监督学习任务的一种。简单来说，回归用于分析和建模变量之间的关系，通常用来预测或解释一个或多个因变量与一个或多个自变量之间的关联。

简单来讲，就是找因变量和自变量的关系



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 25. 根据数据是否有标签、标签类型细分机器学习算法

和分类算法的联系

很快地，我们就会发现一些回归算法还可以用来解决机器学习的**分类** (classification) 问题。比如，图 26 所示逻辑回归模型用于二分类问题。

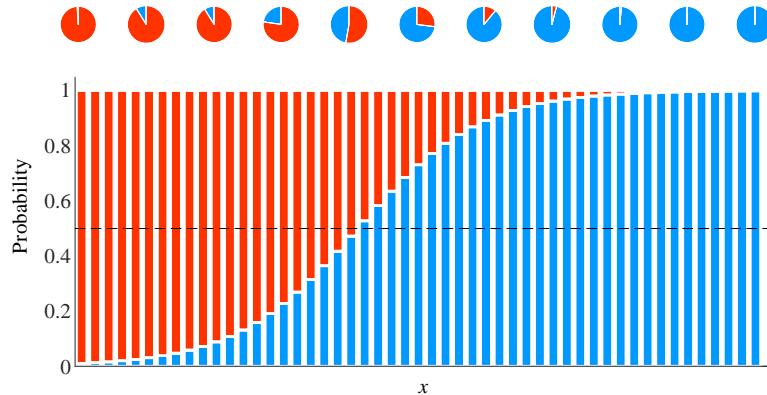


图 26. 逻辑回归模型用于二分类问题

和降维算法的联系

回看图 17，我们会发现，在一元线性回归中，最小二乘法 OLS 定义的残差沿着纵轴。如果我们要是关注，点到直线的距离的话，得到的直线又是什么？

图 27 这幅图便回答了这个问题。图 27 (b) 这种线性回归模型叫做正交回归。

有意思的是，如图 28 所示，解释正交回归的最好办法是机器学习中的一种常用降维算法——**主成分分析** (Principal Component Analysis, PCA)。而基于主成分分析，我们又可以拓展出其他各种回归方法。

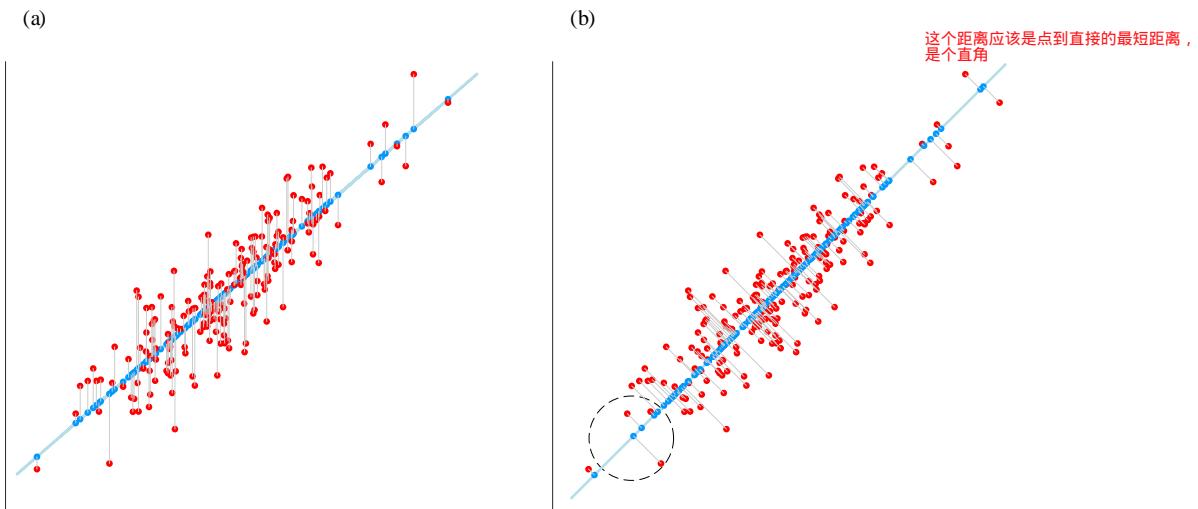


图 27. 对比最小二乘回归和正交回归

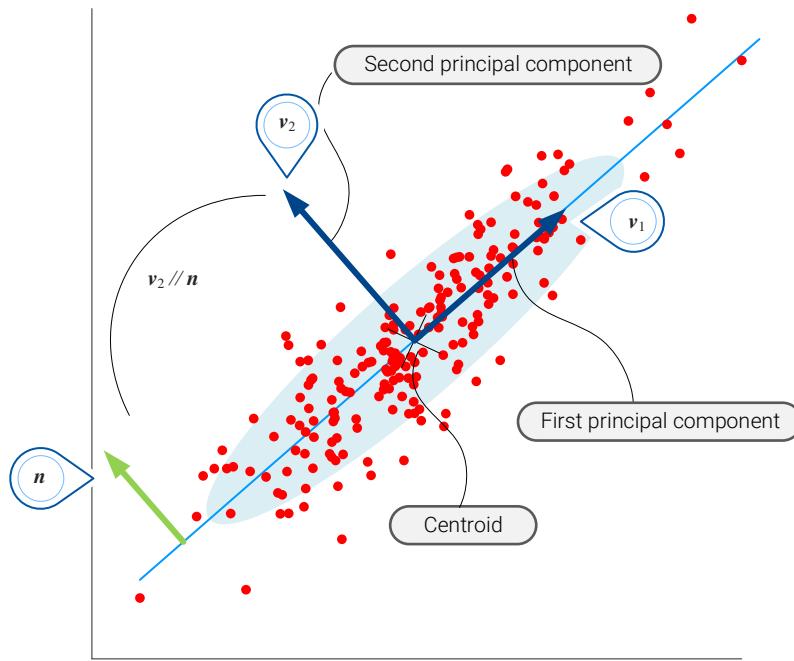


图 28. 正交回归和主成分分解的关系

从点到线、由线及面

很难想象，从图 15 这幅平淡无奇的散点图走来，我们竟然走了这么远！

而且图 29 告诉我们脚下的路还在沿着各个方向蜿蜒...

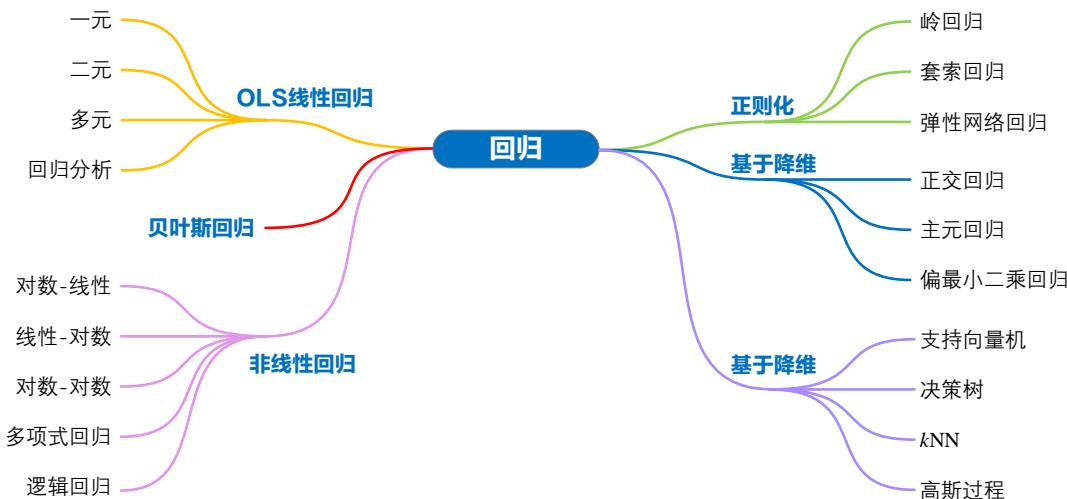


图 29. 从一元线性回归开始，不断“反复 + 精进”

这是一个从点到线、由线及面的故事。从一个个孤岛开始，不断扩展直到整片海洋。

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

正如本书前言写的那样，利用“编程 + 可视化”，我们可以打破数学板块之间的壁垒，让大家看到数学代数、几何、线性代数、微积分、概率统计、优化、数据分析、机器学习等板块之间的联系，编织一张绵密的数学知识网络。

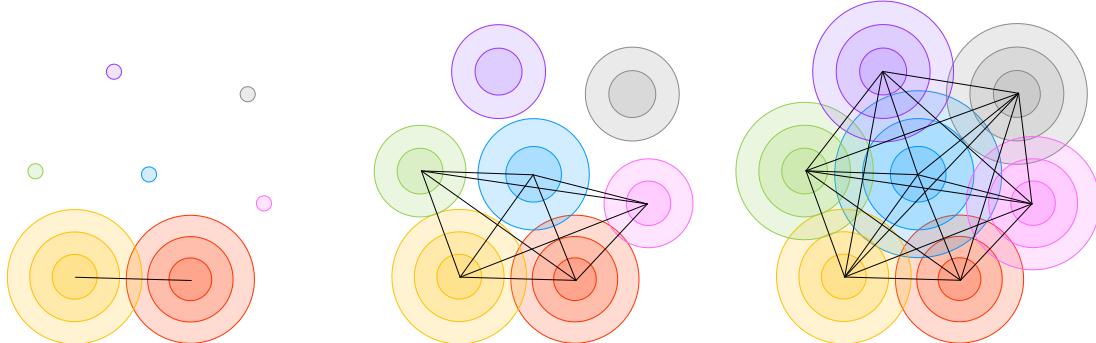


图 30. 编程 + 可视化 + 数学 + 机器学习：从点到线，由线及面

一张学习地图

在我们一起踏上这段奇妙旅途伊始，再次强烈建议各位鸢尾花书读者不要仅仅满足于“调包”。希望大家在“调包”时，更要了解这些工具背后的数学原理、算法流程。

虽然《编程不难》《可视之美》仅仅要求大家知其然，不需要知其所以然；但是，鸢尾花书《数学要素》《矩阵力量》《统计至简》专门介绍常见各种数学工具原理。

《数据有道》《机器学习》则介绍机器学习各种常用算法原理。这五本书会循序渐进给大家解释很多 Python 工具背后的原理。

虽然鸢尾花书每一册自成体系，但又相互高度依赖，难以避免给大家造成“套娃”“挤牙膏”的既视感，希望大家体谅。鸢尾花书全系列免费开源，大家可以从中 GitHub 下载草稿和 Python 文件，根据自己的节奏、偏好自主探索。

希望鸢尾花书不仅仅给大家提供“Python 编程 + 可视化 + 数学 + 机器学习”这套强有力的合作拳，还给大家提供一种自主探究的学习方法。

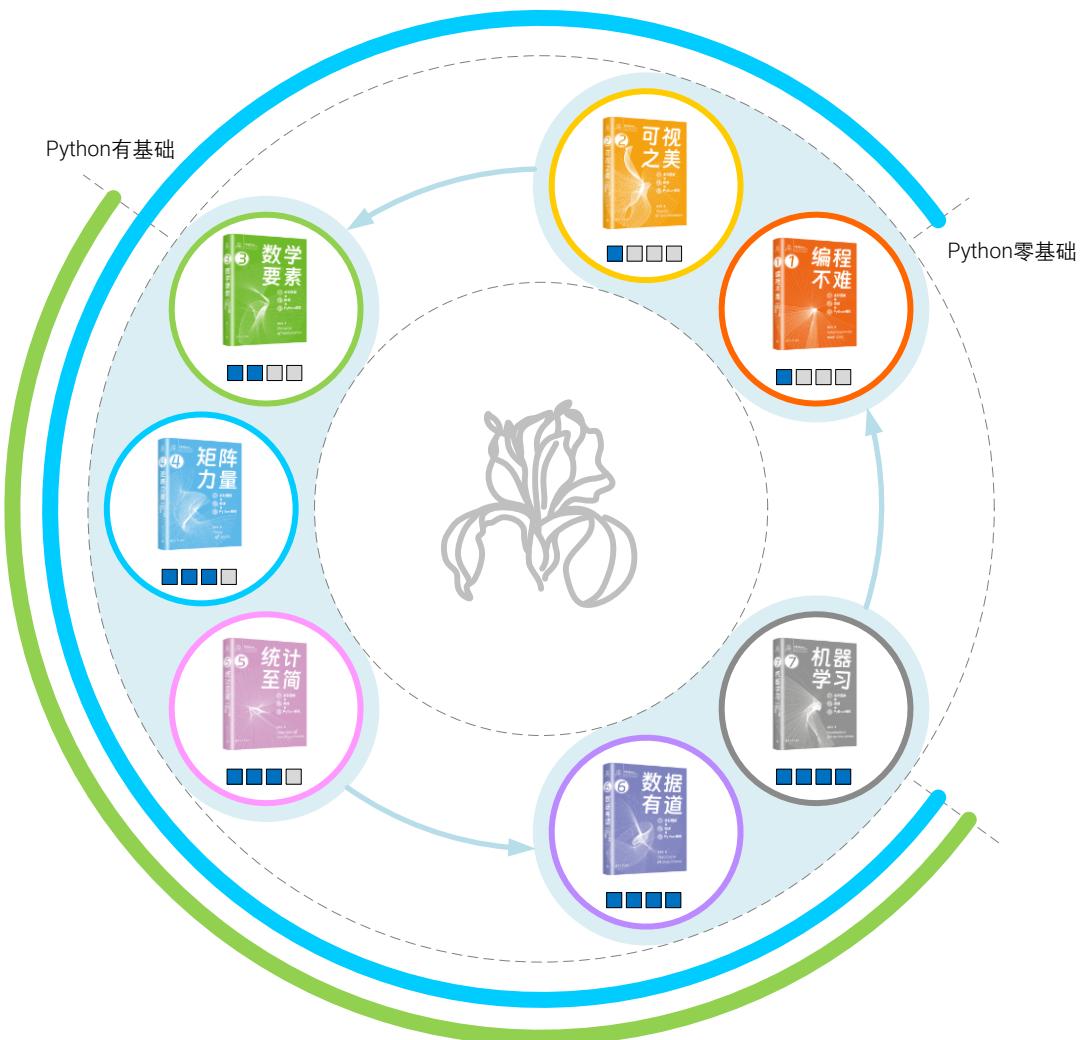
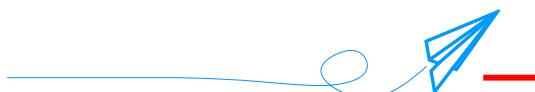


图 31. 整套鸢尾花书和大家一起持续“重复 + 精进”



相信滴水穿石的力量！“反复 + 精进”会把最陡峭的学习曲线拉平，推动我们一步步登上看似无路可爬的山峰。

不积跬步，无以至千里；不积小流，无以成江海。

脚下的路沿着四面八方伸延而去…

从今天起，做一个旅人，日拱一卒，功不唐捐。