Preface

前言

感谢

首先感谢大家的信任。

作者仅仅是在学习应用数学科学和机器学习算法时，多读了几本数学书，多做了些思考和知识整理而已。知者不言，言者不知。知者不博，博者不知。水平有限，把自己有限所学所思斗胆和大家分享，作者权当无知者无畏。希望大家在 B 站视频下方和 Github 多提意见，让这套书成为作者和读者共同参与创作的优质作品。

特别感谢清华大学出版社的栾大成老师。从选题策划、内容创作、装帧设计，栾老师事无巨细、一路陪伴。每次和栾老师交流，我都能感受到他对优质作品的追求、对知识分享的热情。

出来混总是要还的

曾几何时，考试是我们学习数学的唯一动力。考试是头悬梁的绳，是锥刺股的锥。我们中的绝大多数人从小到大为各种考试埋头题海，数学味同嚼蜡，甚至让人恨之入骨。

数学给我们带来了无尽的折磨。我们憎恨数学，恐惧数学，恨不得一走出校门就把数学抛之脑后、老死不相往来。

可悲可笑的是，我们其中很多人可能会在毕业的五年或十年以后，因为工作需要，不得不重新学习微积分、线性代数、概率统计，悔恨当初没有学好数学、走了很多弯路、没能学以致用，从而迁怒于教材和老师。

这一切不能都怪数学，值得反思的是我们学习数学的方法、目的。

再给自己一个学数学的理由

为考试而学数学，是被逼无奈的举动。而为数学而数学，则又太过高尚而遥不可及。

相信对于绝大部分的我们来说，数学是工具、是谋生手段，而不是目的。我们主动学数学，是想用数学工具解决具体问题。

现在，这套书给大家一个“学数学、用数学”的全新动力——数据科学、机器学习。

数据科学和机器学习已经深度融合到我们生活的方方面面，而数学正是开启未来大门的钥匙。不是所有人生来都握有一副好牌，但是掌握“数学 + 编程 + 机器学习”绝对是王牌。这次，学习数学不再是为了考试、分数、升学，而是投资时间、自我实现、面向未来。

未来已来，你来不来？

本套丛书如何帮到你

为了让大家学数学、用数学，甚至爱上数学，作者可谓颇费心机。在创作这套书时，作者尽量克服传统数学教材的各种弊端，让大家学习时有兴趣、看得懂、有思考、更自信、用得着。

为此，丛书在内容创作上突出以下几个特点：

- **数学 + 艺术**——全彩图解，极致可视化，让数学思想跃然纸上、生动有趣、一看就懂，同时提高大家的数据思维、几何想象力、艺术感；
- **零基础**——从零开始学习 Python 编程，从写第一行代码到搭建数据科学和机器学习应用；
- **知识网络**——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、概率统计等板块之间的联系，编织一张绵密的数学知识网络；
- **动手**——授人以鱼不如授人以渔，和大家一起写代码、用 Streamlit 创作数学动画、交互 App；
- **学习生态**——构造自主探究式学习生态环境“微课视频 + 纸质图书 + 电子图书 + 代码文件 + 可视化工具 + 思维导图”，提供各种优质学习资源；
- **理论 + 实践**——从加减乘除到机器学习，丛书内容安排由浅入深、螺旋上升，兼顾理论和实践；在编程中学习数学，学习数学时解决实际问题。

虽然本书标榜“从加减乘除到机器学习”，但是建议读者朋友们至少具备高中数学知识。如果读者正在学习或曾经学过大学数学 (微积分、线性代数、概率统计)，这套书就更容易读了。

聊聊数学

数学是工具。锤子是工具，剪刀是工具，数学也是工具。

数学是思想。数学是人类思想的高度抽象的结晶体。在其冷酷的外表之下，数学的内核实际上就是人类朴素的思想。学习数学时，知其然，更要知其所以然。不要死记硬背公式定理，理解背后的数学思想才是关键。如果你能画一幅图、用大白话描述清楚一个公式、一则定理，这就说明你真正理解了它。

数学是语言。就好比世界各地不同种族有自己的语言，数学则是人类共同的语言和逻辑。数学这门语言极其精准、高度抽象，放之四海而皆准。虽然我们中绝大多数人没有被数学女神选中，不能为人类的对数学认知开疆扩土；但是，这丝毫不妨碍我们使用数学这门语言。就好比，我们不会成为语言学家，我们完全可以使用母语和外语交流。

数学是体系。代数、几何、线性代数、微积分、概率统计、优化方法等等，看似一个个孤岛，实际上都是数学网络的一条条织线。建议大家学习时，特别关注不同数学板块之间的联系，见树，更要见林。

数学是基石。拿破仑曾说“数学的日臻完善和这个国强民富息息相关。”数学是科学进步的根基，是经济繁荣的支柱，是保家卫国的武器，是探索星辰大海的航船。

数学是艺术。数学和音乐、绘画、建筑一样，都是人类艺术体验。通过可视化工具，我们会在看似枯燥的公式、定理、数据背后，发现数学之美。

数学是历史，是人类共同记忆体。”历史是过去，又属于现在，同时在指引未来。”数学是人类的集体学习思考，她把人的思维符号化、形式化，进而记录、积累、传播、创新、发展。从甲

骨、泥板、石板、竹简、木牍、纸草、羊皮卷、活字印刷、纸质书，到数字媒介，这一过程持续了数千年，至今绵延不息。

数学是无穷无尽的**想象力**，是人类的**好奇心**，是自我挑战的**毅力**，是一个接着一个的**问题**，是看似荒诞不经的**猜想**，是一次次胆大包天的**批判性思考**，是敢于站在前人的臂膀之上的**勇气**，是孜孜不倦地延展人类认知边界的**不懈努力**。

家园、诗、远方

诺瓦利斯曾说：“哲学就是怀着一种乡愁的冲动到处去寻找家园。”

在纷繁复杂的尘世，数学纯粹的就像精神的世外桃源。数学是，一束光，一条巷，一团不灭的希望，一股磅礴的力量，一个值得寄托的避风港。

打破陈腐的锁链，把功利心暂放一边，我们一道怀揣一分乡愁、心存些许诗意、踩着艺术维度，投入数学张开的臂膀，驶入她色彩斑斓、变幻无穷的深港，感受久违的归属，一睹更美、更好的远方。

Acknowledgement

致谢

To my parents.

谨以此书献给我的母亲父亲

How to Use the Book

使用本书

丛书资源

本系列丛书提供的配套资源有以下几个：

- 纸质图书；
- PDF 文件，方便移动终端学习；请大家注意，纸质图书经过出版社五审五校修改，内容细节上会和 PDF 文件有出入。
- 每章提供思维导图，纸质书提供全书思维导图海报；
- Python 代码文件，直接下载运行，或者复制、粘贴到 Jupyter 运行；
- Python 代码中有专门用 Streamlit 开发数学动画和交互 App 的文件；
- 微课视频，强调重点、讲解难点、聊聊天。

在纸质书中为了方便大家查找不同配套资源，作者特别设计了如下几个标识。

微课视频

本书配套微课视频均发布在 B 站——生姜 DrGinger：

- https://space.bilibili.com/513194466

微课视频是以“聊天”的方式，和大家探讨某个数学话题的重点内容，讲讲代码中可能遇到的难点，甚至侃侃历史、说说时事、聊聊生活。

本书配套的微课视频目的是引导大家自主编程实践、探究式学习，并不是“照本宣科”。

纸质图书上已经写得很清楚的内容，视频课程只会强调重点。需要说明的是，图书内容不是视频的“逐字稿”。

代码文件

本系列丛书的 Python 代码文件下载地址为：

- https://github.com/Visualize-ML

Python 代码文件会不定期修改，请大家注意更新。图书配套的 PDF 文件和勘误也会上传到这个 GitHub 账户。因此，建议大家注册 GitHub 账户，给书稿文件夹标星 (star) 或分支克隆 (fork)。

考虑再三，作者还是决定不把代码全文印在纸质书中，以便减少篇幅，节约用纸。

本书编程实践例子中主要使用“鸢尾花数据集”，数据来源是 Scikit-learn 库、Seaborn 库。此外，系列丛书封面设计致敬梵高《鸢尾花》，要是给本系列丛书起个昵称的话，作者乐见“鸢尾花书”。

App 开发

本书几乎每一章都至少有一个用 Streamlit 开发的 App，用来展示数学动画、数据分析、机器学习算法。

Streamlit 是个开源的 Python 库，能够方便快捷搭建、部署交互型网页 App。Streamlit 非常简单易用、很受欢迎。Streamlit 兼容目前主流的 Python 数据分析库，比如 NumPy、Pandas、Scikit-learn、PyTorch、TensorFlow 等等。Streamlit 还支持 Plotly、Bokeh、Altair 等交互可视化库。

本书中很多 App 设计都采用 Streamlit + Plotly 方案。此外，本书专门配套教学视频手把手和大家一起做 App。

大家可以参考如下页面，更多了解 Streamlit：

- https://streamlit.io/gallery
- https://docs.streamlit.io/library/api-reference

实践平台

本书作者编写代码时采用的 IDE (integrated development environment) 是 Spyder，目的是给大家提供简洁的 Python 代码文件。

但是，建议大家采用 JupyterLab 或 Jupyter notebook 作为本系列丛书配套学习工具。

简单来说，Jupyter 集合“浏览器 + 编程 + 文档 + 绘图 + 多媒体 + 发布”众多功能与一身，非常适合探究式学习。

运行 Jupyter 无需 IDE，只需要浏览器。Jupyter 容易分块执行代码。Jupyter 支持 inline 打印结果，直接将结果图片打印在分块代码下方。Jupyter 还支持很多其他语言，比如 R 和 Julia。

使用 markdown 文档编辑功能，可以编程同时写笔记，不需要额外创建文档。Jupyter 中插入图片和视频链接都很方便。此外，还可以插入 Latex 公式。对于长文档，可以用边栏目录查找特定内容。

Jupyter 发布功能很友好，方便打印成 HTML、PDF 等格式文件。

Jupyter 也并不完美，目前尚待解决的问题有几个。Jupyter 中代码调试不方便，需要安装专门插件 (比如 debugger)。Jupyter 没有 variable explorer，要么 inline 打印数据，要么将数据写到 csv 或 Excel 文件中再打开。图像结果不具有交互性，比如不能查看某个点的值，或者旋转 3D 图形，可以考虑安装 (jupyter-matplotlib)。注意，利用 Altair 或 Plotly 绘制的图像支持交互功能。对于自定义函数，目前没有快捷键直接跳转到其定义。但是，很多开发者针对这些问题都开发了插件，请大家留意。

大家可以下载安装 Anaconda，JupyterLab、Spyder、PyCharm 等常用工具都集成在 Anaconda 中。下载 Anaconda 的地址为：

◀ https://www.anaconda.com/

学习步骤

大家可以根据自己的偏好制定学习步骤，本书推荐如下步骤。

学完每章后，大家可以在平台上发布自己的 Jupyter 笔记，进一步听取朋友们的意见，共同进步。这样做还可以提高自己学习的动力。

意见建议

欢迎大家对本系列丛书提意见和建议，丛书专属邮箱地址为：

◀ jiang.visualize.ml@gmail.com

也欢迎大家在 B 站视频下方留言互动。

Contents

目录

Introduction

0 绪论

图解 + 编程 + 实践 + 数学板块融合

0.1 本册在鸢尾花书的定位

首先祝贺大家完成“数学”板块的学习，同时欢迎大家来到鸢尾花书第三板块——实践。

“实践”这个板块，我们将会把学到的编程、可视化，特别是数学工具应用到具体的数据科学、机器学习算法中，并在实践中加深对这些工具的理解。

“实践”这个板块有两本书：《数据有道》、《机器学习》。鸢尾花书读者应该知道机器学习可以大致分为：a) 有监督学习；b) 无监督学习。

有监督学习可以进一步分为：a.1) 分类；a.2) 回归。

无监督学习也可以分为两类：b.1) 聚类；b.2) 降维。

《数据有道》着重讲解 a.2) 回归、b.2) 降维，这两个板块。《机器学习》则强调 a.1) 分类、b.1) 聚类。

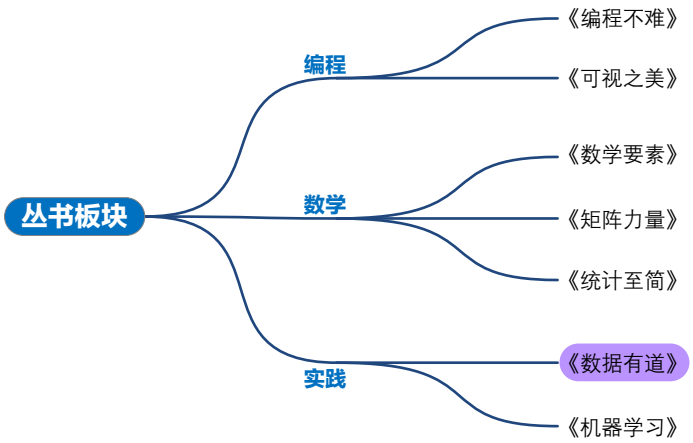


图 1. 鸢尾花书板块布局

0.2 结构：4 大板块

《数据有道》可以归纳为 4 大板块：数据处理、时间数据、回归、降维。

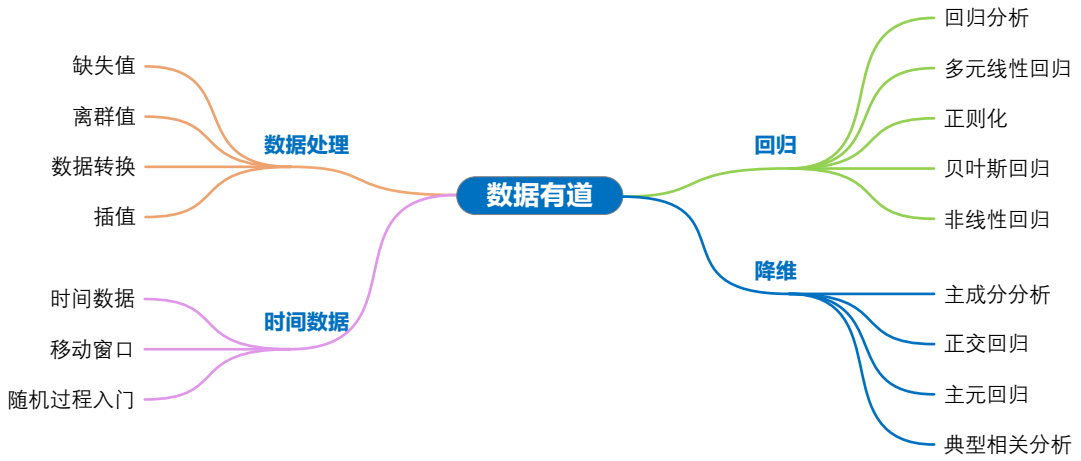


图 2.《数据有道》板块布局

数据处理

第 1 章总括介绍常见数据类型、处理、模型。第 2 章讲解如何处理数据中的缺失值。第 3 章介绍处理离群值的常用工具，这一章和机器学习算法联系紧密。第 4 章讲解常用数据转换方法，本章也相当于对统计知识的回顾。第 5 章特别介绍插值，注意插值和回归的区别。

时间数据

这个板块介绍一类特殊数据——具有时间戳的数据，也叫时间序列。第 6 章讲解如何处理时间数据、发现数据的趋势、时间序列分解等内容。时间数据的特征随时间动态变化，这是第 7 章特别强调的一点。第 7 章中，大家会看到均值、标准差 (波动率)、相关性系数、回归系数都可以随时间变化。第 8 章是随机过程入门，介绍布朗运动、几何布朗运动，以及用几何布朗运动完成股价走势的蒙特卡罗模拟。这一章是《统计至简》第 15 章的延伸。

回归

这个版块都和回归有关。第 9 章首先利用一元 OLS 线性回归讲解回归分析，本章中大家会学到方差分析、拟合优度、F 检验、t 检验、置信区间、预测区间、对数似然函数、信息准则等概念。这一章相对较为无聊，建议大家学习时没有必要全部掌握。实践时再回来有针对性地学习。

第 10 章讲解多元线性回归，回归分析的维度提高。这一章请大家多从几何、数据视角思考回归分析。第 11 章利用正则化解决多元线性回归过拟合、多重共线性的问题。这一章一共介绍三种正则化：a) 岭回归；b) 套索回归；c) 弹性网络回归。

第 12 章介绍如何将贝叶斯推断用在回归分析中。学习这一章时，建议大家回顾《统计至简》第 20 ~ 22 章。这一章最后从贝叶斯推断视角理解正则化。第 13 章讲解非线性回归，需要大家掌握多项式回归，并理解过拟合。此外，这一章还介绍了逻辑回归，逻辑回归既可以用来回归分析，也可以用来分类。

降维

第 14、15 章讲解主成分分析。第 14 章侧重从应用角度讲解，第 15 章则区分六种不同的技术路线。鸢尾花书在不同的板块都或多或少地介绍过主成分分析，这样安排的目的是当大家从线性代数、概率统计、优化、数据等不同角度透彻理解主成分分析。对读者来说，这种抽丝剥茧、逐层深入的讲解方式，不至于信息过载。

第 16、17 章分别介绍以主成分分析为基础的两种回归方法：正交回归、主元回归。虽然这两章介绍的是回归方法，但是它们都离不开主成分分析。此外，第 17 章还介绍了偏最小二乘回归。

第 18 章介绍典型相关分析。典型相关分析方法的目的是找到两组数据的整体相关性的最大线性组合。

0.3 特点：应用

《数据有道》一册的最大特点就是“应用”。本书除了使用鸢尾花数据之外，本书还经常使用股票数据。

在学习本册时，希望大家不要仅仅满足于“调用”Python 库，要知其然，更要知其所以然，弄清楚这些函数底层的算法逻辑。《数学要素》、《矩阵力量》、《统计至简》这三册介绍的数学工具对于本册至关重要，特别是线性代数、概率统计等数学工具。因此，不建议大家跳过“数学”板块三本书，直接学习本册内容。

《数据有道》相当于《机器学习》的基础。此外，本册的“回归”、“降维”这两个板块还会以“综述”方式出现在《机器学习》一册。此外，在数据科学、机器学习实践中，大家会发现《数据有道》一册的很多工具都可以用在特征工程。

《数据有道》和《机器学习》还给大家更多在线开源资源，帮助大家扩展学习。

掌握数据分析的技能需要长年累月地和浩如烟海的数据“摸爬滚打”，不可能一蹴而就。希望大家在学习本册时，能够一边学理论、一边搞实践。

下面，我们正式开始本册的学习之旅！

1 万物皆数

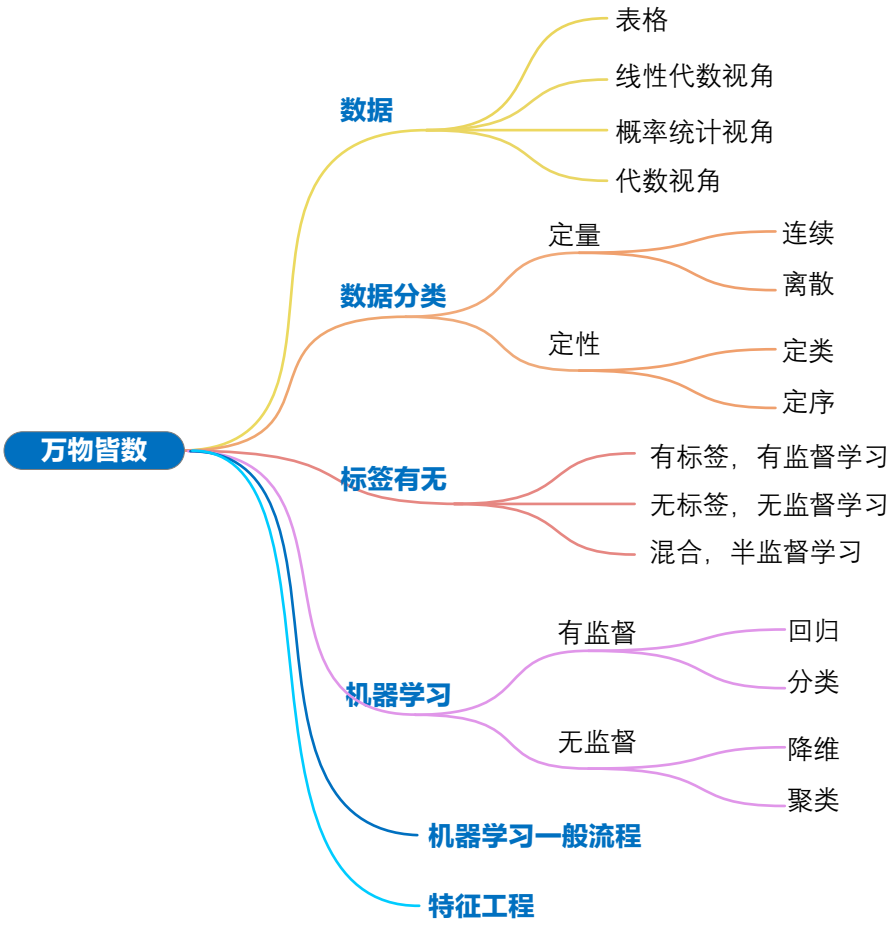
All Is Number

从数据科学、机器学习视角再看数字

但凡满足以下两个条件的理论，便可以称之为优质理论：基于几个有限的变量，准确描述大量观测值；能对对未来观测值做出确定的预测。

A theory is a good theory if it satisfies two requirements: it must accurately describe a large class of observations on the basis of a model that contains only a few arbitrary elements, and it must make definite predictions about the results of future observations.

—— 史蒂芬·霍金 (Stephen Hawking) | 英国理论物理学家、宇宙学家 | 1942 ~ 2018



1.1 从表格说起

四个视角

这是一个有关数字的故事，故事的开端便是形如图 1 所示的表格数据。任何表都可以看成是由**行** (row) 和**列** (column) 构成。

从线性代数角度来看，图 1 这个表格本质上是一个矩阵。《矩阵力量》介绍过矩阵的每一行可以看成是一个**行向量** (row vector)，每一列为**列向量** (column vector)。

比如，将图 1 这个矩阵记做 $\boldsymbol{X}$，$\boldsymbol{X}$ 可以写成一组列向量 $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_D]$。$\boldsymbol{X}$ 当然也可以写成一组行向量 $\boldsymbol{X} = [\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)}]^{\mathrm{T}}$。

⚠ 注意，在《机器学习》一册中，为了方便 $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)}$ 偶尔也会被视作为列向量，会具体说明。

从统计角度来看，表格的每一列可以视作一个随机变量的样本数据。图 1 则代表 D 个随机变量 $(X_1, X_2, \ldots, X_D)$ 的样本数据。

$X_1, X_2, \ldots, X_D$ 可以构成 D 元随机变量列向量 $\boldsymbol{\chi} = [X_1, X_2, \ldots, X_D]^{\mathrm{T}}$。

从代数角度来看，图 1 表格的每一列相当于变量 $(x_1, x_2, \ldots, x_D)$ 的取值。比如，我们会在回归分析的解析式中看到这种记法 $y = b_0 + b_1x_1 + b_2x_2 + \ldots + b_Dx_D$。

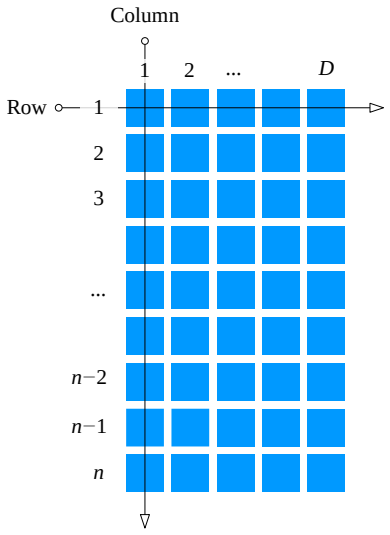


图 1. 表格数据

定量数据、定性数据

数据一般可以分为**定量数据** (quantitative data) 和**定性数据** (qualitative data)，具体分类如图 2 所示。

定量数据指的是，可以采用数值表达的数据，比如股票价格、人体高度、气温等等。

定性数据，也叫**类别数据** (categorical data)，指的是描述事物的特征、属性等文字或符号，比如姓名、颜色、国家、性别、五星评价等等。

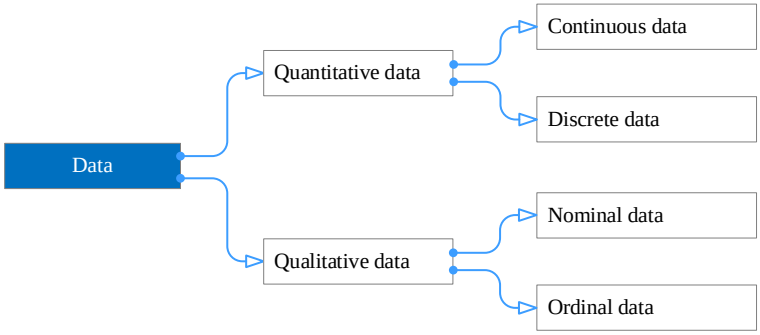


图 2. 数据分类

连续数据、离散数据

定量数据，还可以进一步分为**连续数据** (continuous data) 和**离散数据** (discrete data)。

连续数据是指在一定区间内可以任意取值的数据，比如气温、GDP 数据等等。离散数据只能采取特定值，比如说个数 (整数)、一到五星好评、骰子点数等等。

一天 24 小时之内的温度数据不可能被持续记录，按一定时间频率需要采样。举个例子，比如，每小时记录一个温度数值。图 3 所示为某国家 GDP 数据，虽然为年度数据，当数据量足够大时，GDP 增长曲线看上去是连续曲线；但是，当展开局部数据时，可以发现这条所谓的连续数据实际上是相邻点相连构成的“折线”。

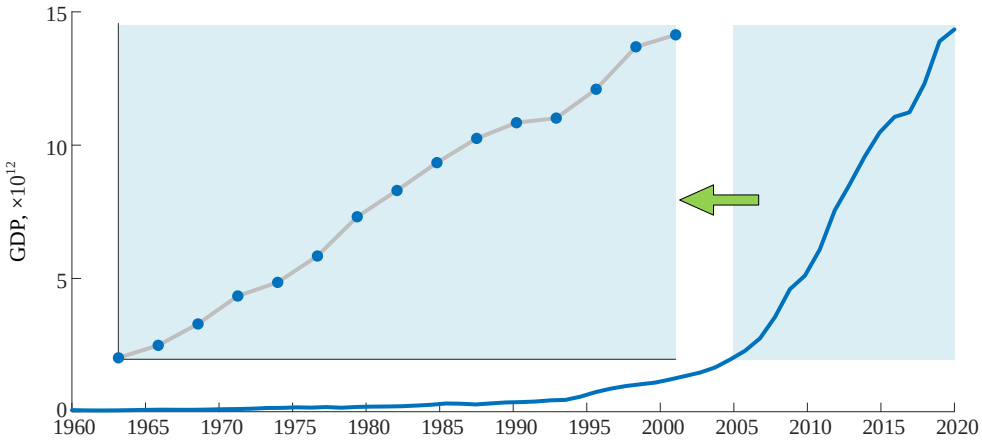


图 3. 采样数据

定类数据、定序数据

定性数据也可以分为**定类数据** (nominal data) 和**定序数据** (ordinal data)。简单来说，定类数据没有任何内在顺序或排序，定序数据指具有内在顺序或排序的数据。

定类数据，也叫名义数据，用来表征事物类别，比如血型 A、B、AB 和 O。

定序数据，也叫有序数据，不仅能够代表事物的类别，还可以据此特征排序，比如学生成绩 A、B、C、D 和 F。此外，区间数据 (interval data) 也可以看做时一种定序数据，比如身高区间数据，160 cm 以下 (包括 160 cm)、160 cm 到 170 cm (包括 170 cm) 、170 cm 到 180 cm (包括 180 cm) 和 180 cm 以上。

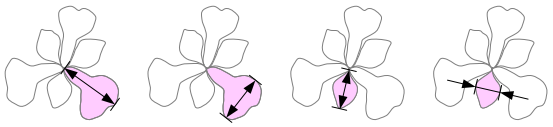
混合

很多时候，一个表格常常是各种数据的集合体。如图 4 所示，表格每一行代表一个学生的某些基本数据。表格第 1 列为为学生姓名，表格第 2 列为性别 (定类数据)，表格第 3 列为身高 (连续定量数据)，第 4 列为成绩 (定序数据)，第 5 列为血型 (定类数据)。

大家已经很熟悉的鸢尾花数据也是混合数据表格。如图 5 所示，表格的第一列为序号，之后四列为花萼长度、花萼宽度、花瓣长度、花瓣宽度四个特征的连续数据。最后一列为鸢尾花分类标签。

Name	Gender	Height	Grade	Blood
James	Male	185	A	AB
Shawn	Male	178	A+	B
Mary	Female	165	A−	O
Alice	Female	175	A+	B
Bill	Male	171	B	A
Julia	Female	168	B+	A

图 4. 学生数据



Index	Sepal length X_1	Sepal width X_2	Petal length X_3	Petal width X_4	Species C
1	5.1	3.5	1.4	0.2	
2	4.9	3	1.4	0.2	
3	4.7	3.2	1.3	0.2	
...	...	...	...	...	Setosa C_1
49	5.3	3.7	1.5	0.2	
50	5	3.3	1.4	0.2	
51	7	3.2	4.7	1.4	
52	6.4	3.2	4.5	1.5	
53	6.9	3.1	4.9	1.5	
...	...	...	...	...	Versicolor C_2
99	5.1	2.5	3	1.1	
100	5.7	2.8	4.1	1.3	
101	6.3	3.3	6	2.5	
102	5.8	2.7	5.1	1.9	
103	7.1	3	5.9	2.1	
...	...	...	...	...	Virginica C_3
149	6.2	3.4	5.4	2.3	
150	5.9	3	5.1	1.8	

图 5. 鸢尾花数据表格，单位为厘米 (cm)

有标签、无标签数据

根据输出值有无标签，如图 6 所示，数据可以分为**有标签数据** (labelled data) 和**无标签数据** (unlabelled data)。鸢尾花数据显然是有标签数据。删去鸢尾花最后一列标签，我们便得到无标签数据。

有标签数据和无标签数据是机器学习中常见的两种数据类型，它们在不同的应用场景中有不同的用途。

简单来说，**有标签数据**是指已经被人工或其他方式标注了类别或标签的数据。在有标签数据中，每个样本都有对应的标签或分类信息。有标签数据通常用于**监督学习** (supervised learning)，即机器学习模型可以利用已知的标签信息进行训练，并在后续的预测过程中使用这些信息进行分类或回归。

无标签数据是指没有标签或分类信息的数据。在无标签数据中，样本只有特征信息，而没有对应的标签信息。无标签数据通常用于**无监督学习** (unsupervised learning)，即机器学习模型需要通过自己的学习过程，从数据中发现并学习出有意义的模式和结构。无监督学习通常包括聚类、降维和异常检测等任务。

在实际应用中，有标签数据和无标签数据往往同时存在。例如，在文本分类任务中，可以有大量已经标注好类别的文本数据 (有标签数据)，但同时还存在大量未分类的文本数据 (无标签数据)，可以利用这些无标签数据进行**半监督学习** (semi-supervised learning)。

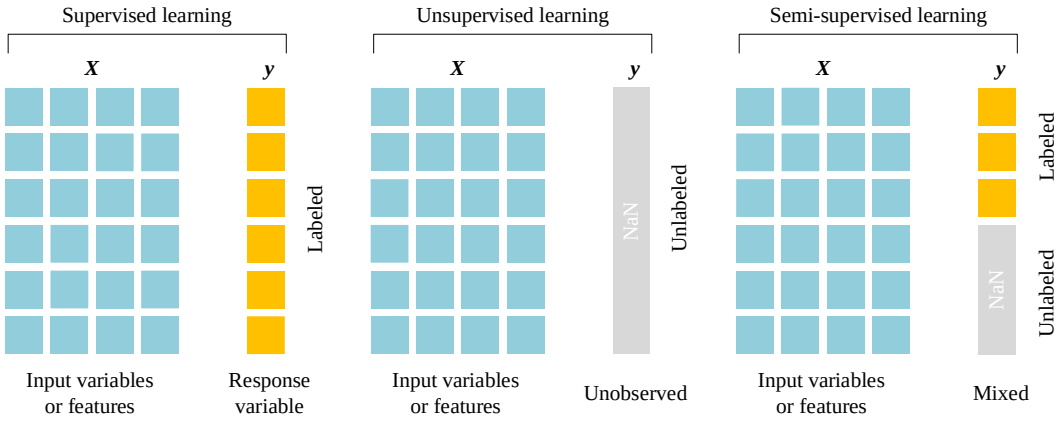


图 6. 根据有无标签分类数据

1.2 机器学习方法分类

人工智能 (Artificial Intelligence, AI) 是一套算法系统，它通过模拟人类智慧，感知环境，经过分析计算，进而可以执行设定的行为动作。

机器学习

机器学习是实现人工智能的一大类方法和技术。机器学习算法的特点是，从样本数据中分析并获得某种规律，再利用这个规律对未知数据进行预测。它是涉及概率、统计、矩阵论、代数学、优化方法、数值方法、算法学等多领域的交叉学科。

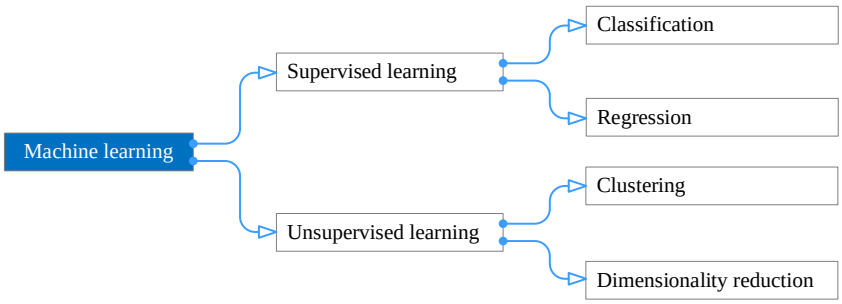


图 7. 机器学习分类

机器学习适合处理的问题有如下特征：(a) 大数据；(b) 黑箱或复杂系统，难以找到**控制方程** (governing equations)。机器学习需要通过数据的训练。

如图 7 所示，简单来说，机器学习可以分为以下两大类：

- **有监督学习**，也叫监督学习，训练练有标签值样本数据并得到模型，通过模型对新样本进行推断。
- **无监督学习**训练没有标签值的数据，并发现样本数据的结构和分布。

此外，**半监督学习**结合无监督学习和监督学习。

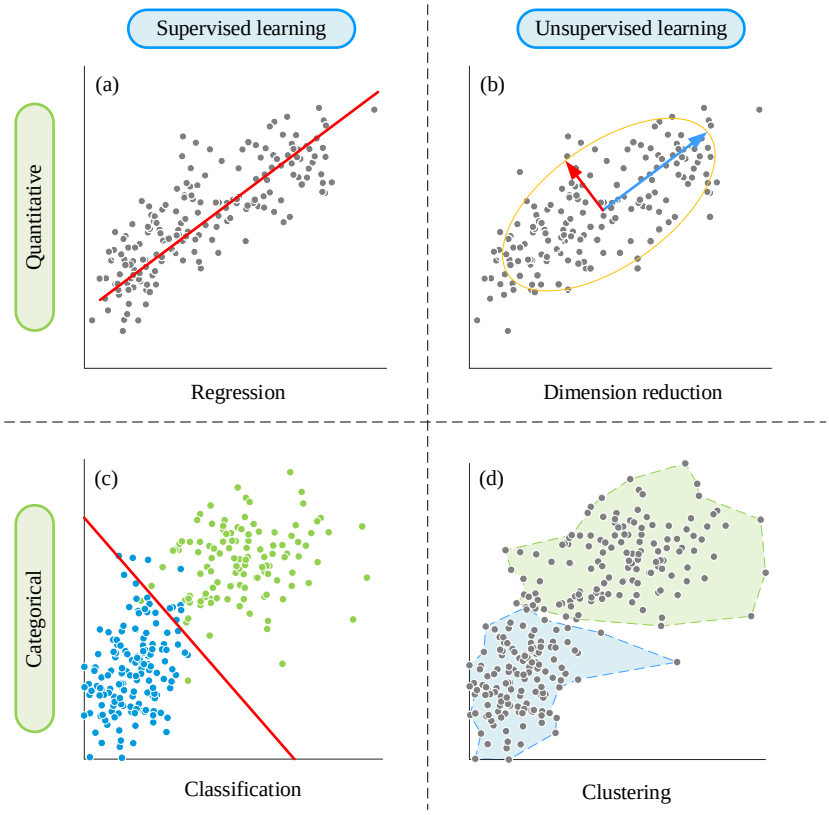


图 8. 根据数据是否有标签、标签类型细分机器学习算法，图片来自《矩阵力量》第 25 章

有监督学习

如图 8 所示，有监督学习可以进一步分为**分类** (classification)、**回归** (regression)。

分类问题是指将数据集划分为不同的类别或标签。给定一个输入，分类模型的目标是预测它所属的类别，如垃圾邮件分类、图像识别和情感分析等。分类问题的输出是一个离散值或类别标签。

回归问题是指根据已知的输入和输出数据，建立一个数学模型来预测输出值。给定一个输入，回归模型的目标是预测它的输出值，如房价预测、股票价格预测和天气预测等。回归问题的输出是一个连续的值或数值。

总的来说，分类问题与离散的输出相关，目标是将数据划分为不同的类别或标签，而回归问题与连续的输出相关，目标是预测数值型数据的结果。

本书将介绍如下几种回归算法：

- **线性回归** (linear regression)，本书第 10、11 章；
- **贝叶斯回归** (Bayesian regression)，本书第 12 章；
- **岭回归** (ridge regression)，本书第 13 章；
- **套索回归** (LASSO regression)，本书第 13 章；
- **弹性网络回归** (elastic net regression)，本书第 13 章；
- **多项式回归** (Polynomial regression)，本书第 14 章；
- **逻辑回归** (logistic regression)，本书第 15 章；
- **正交回归** (orthogonal regression)，本书第 18 章；
- **主元回归** (principal component regression)，本书第 19 章；
- **偏最小二乘回归** (partial least squares regression)，本书第 19 章。

《机器学习》一册将将专门介绍分类算法。

⚠ 注意，很多分类算法也可以用来完成回归分析，这也是《机器学习》一册要介绍的内容。

无监督学习

如图 8 所示，无监督学习主要分为绍**聚类** (clustering)、**降维** (dimensionality reduction)。

降维是指将高维数据映射到低维空间的过程，以便更好地理解和分析数据。通常情况下，高维数据在进行可视化、建模和处理时都会面临计算资源、时间复杂度和维数灾难等问题。通过降维可以减少数据维度，压缩数据，去除冗余信息，提高模型效率和准确度。

聚类是指将数据集中相似的数据分为一类的过程，以便更好地分析和理解数据。聚类分析是一种无监督学习方法，它不需要标记的训练数据，而是根据数据点之间的相似性或距离关系将它们分为不同的簇或群组。聚类可以用于数据挖掘、图像处理、文本分类、市场细分和生物信息学等领域。常见的聚类算法包括 K 均值聚类、层次聚类和 DBSCAN 等。

总的来说，降维是指将高维数据映射到低维空间的过程，目的是减少数据维度、压缩数据、去除冗余信息，而聚类是指将相似的数据分为一类的过程，目的是更好地分析和理解数据。

本书将主要介绍如下降维算法：

- **主成分分析** (principal component analysis)，本书第 15、16 章；
- **因子分析** (Factor Analysis)，本书第 19 章；
- **典型相关分析** (canonical correlation analysis)，本书第 20 章。

《机器学习》一册将将专门介绍聚类算法。

1.3 机器学习流程

图 9 所示为机器学习的一般流程。具体分步流程通常包括以下步骤：

- **收集数据**：从数据源获取数据集，这可能包括数据清理、去除无效数据和处理缺失值等。
- **特征工程**：对数据进行预处理，包括数据转换、特征选择、特征提取和特征缩放等。
- **数据划分**：将数据集划分为训练集、验证集和测试集等。训练集用于训练模型，验证集用于选择模型并进行调参，测试集用于评估模型的性能。
- **选择模型**：选择合适的模型，例如线性回归、决策树、神经网络等。
- **训练模型**：使用训练集对模型进行训练，并对模型进行评估，可以使用交叉验证等方法进行模型选择和调优。
- **测试模型**：使用测试集评估模型的性能，并进行模型的调整和改进。
- **应用模型**：将模型应用到新数据中进行预测或分类等任务。
- **模型监控**：监控模型在实际应用中的性能，并进行调整和改进。

以上是机器学习的一般分步流程，不同的任务和应用场景可能会有一些变化和调整。在实际应用中，还需要考虑数据的质量、模型的可解释性、模型的复杂度和可扩展性等问题。

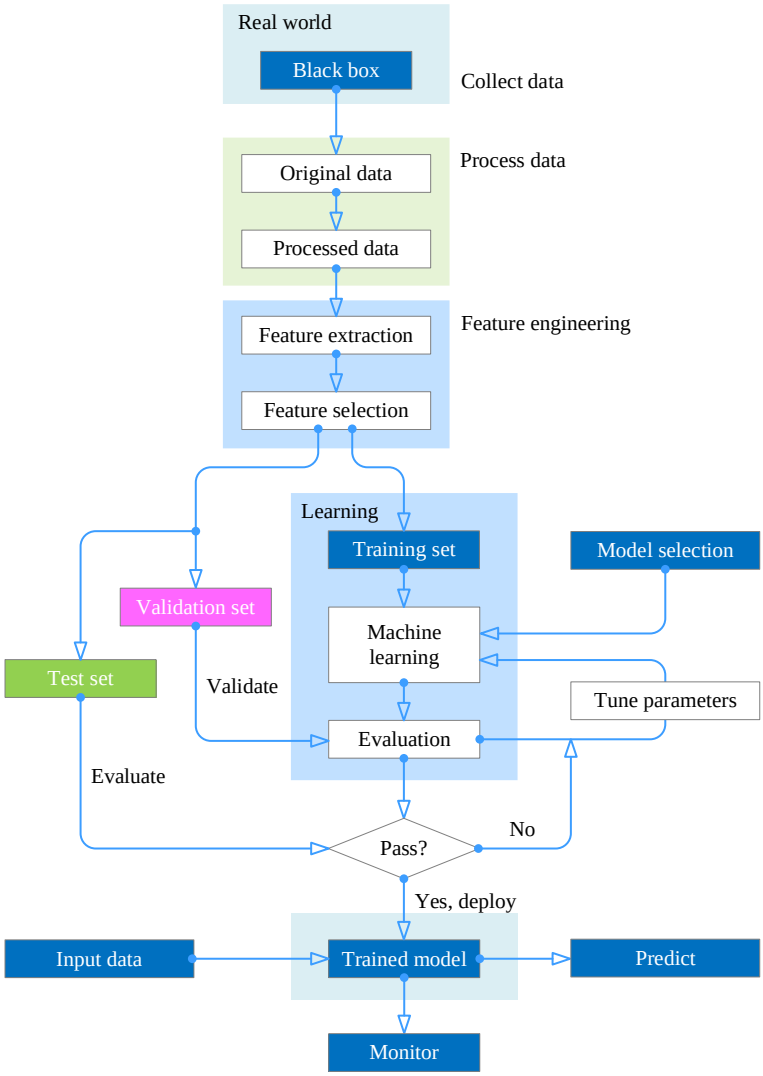


图 9. 机器学习一般流程

1.4 特征工程

从原始数据中最大化提取可用信息的过程就叫做**特征工程** (feature engineering)。特征很好理解，比如鸢尾花花萼长度宽度、花瓣长度宽度，人的性别、身体、体重等，都是特征。

特征工程是机器学习中非常重要的一个环节，指的是对原始数据进行特征提取、特征转换、特征选择和特征创造等一系列操作，以便更好地利用数据进行建模和预测。

具体来说，特征工程包括以下方法：

- **特征提取** (Feature Extraction)：将原始数据转换为可用于机器学习算法的特征向量。注意，这个特征向量不是特征值分解中的特征向量。

- **特征转换** (Feature Transformation)：对原始特征进行数值变换，使其更符合算法的假设。例如，在回归问题中，可以对数据进行对数转换或指数转换等。
- **特征选择** (Feature Selection)：选择最具有代表性和影响力的特征。例如，可以使用相关性分析、PCA 等方法选择最相关或最重要的特征。
- **特征创造** (Feature Creation)：根据原始特征创造新的特征。例如，在房价预测问题中，可以根据房屋面积和房龄创建新的特征。
- **特征缩放** (Feature Scaling)：将特征缩放到相同的尺度或范围内，避免某些特征对模型训练的影响过大。例如，在神经网络中，可以使用标准化或归一化等方法对数据进行缩放。

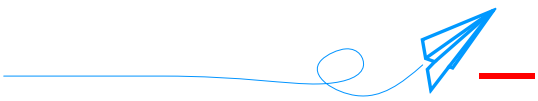
特征工程在机器学习中扮演着至关重要的角色，它可以提高模型的精度、泛化能力和效率。在实际应用中，需要根据具体问题选择合适的特征工程方法，并不断尝试和改进以达到最佳效果。

相信大家都听过“**垃圾进，垃圾出** (garbage in, garbage out, GIGO)”。这句话的含义很简单，将错误的、无意义的数据输入计算机系统，计算机自然也一定会输出错误、无意义的结果。在数据科学、机器学习领域，很多时候数据扮演核心角色。以至于在数据分析建模时，大部分的精力都花在了处理数据上。

特征工程很好的混合了专业知识、数学能力。虽然丛书不会专门讲解特征工程，但是本书的很多内容都可以用于特征工程。

本书第一个板块“数据处理”中介绍的缺失值、离散值处理可以视作特征预处理。而缺失值、离散值也经常使用各种机器学习算法。

本书中的数据转换、插值、正则化、主成分分析、因子分析、典型性分析也都是特征工程的利器。此外，《统计至简》一册中的统计描述、统计推断，《机器学习》一册的**独立成分分析** (independent component analysis, ICA)、**线性判别分析** (linear discriminant analysis, LDA)、**聚类算法**等也都可以用于特征工程。



本章首先简要介绍了观察数据的不同视角 (表格、线性代数、概率统计、代数)。然后，讲解了数据分类。

大家特别需要注意根据数据有无标签可以把机器学习分成两个大类——有监督学习、无监督学习。而有监督学习又可以细分为回归、分类。无监督学习则进一步分为降维、聚类。《数据有道》主要讲解回归、降维，《机器学习》则介绍分类、聚类。

本章最后又聊了聊机器学习的一般流程，以及特征工程。本书几乎所有内容都可以服务特征工程。

有关特征工程，大家可以参考这本开源专著：

http://www.feat.engineering/

Scikit-learn 也有大量特征工程工具，请大家参考：

https://scikit-learn.org/stable/modules/feature_selection.html

2 Dealing with Missing Data

缺失值

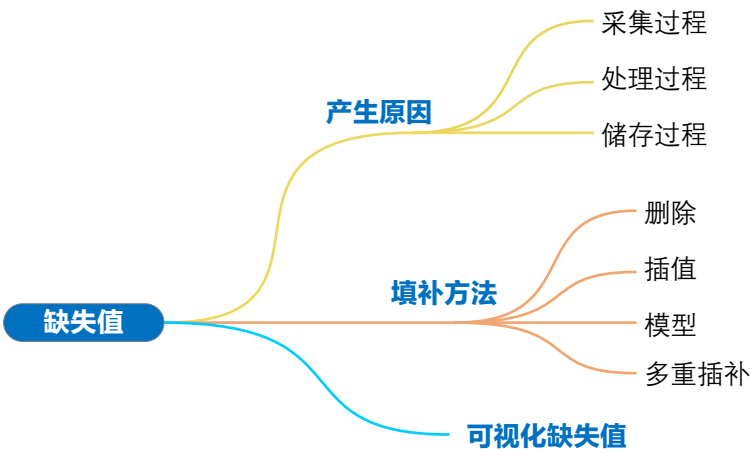
用代数、统计、机器学习算法补齐缺失值

若上天再给一次机会，让我重新开始学业，我定会听从柏拉图，先学数学。

If I were again beginning my studies, I would follow the advice of Plato and start with mathematics.

—— 伽利略·伽利莱 (Galilei Galileo) | 意大利物理学家、数学家及哲学家 | 1564 ~ 1642

- `df.dropna(axis = 0, how = 'any')` 中 `axis = 0` 为按行删除，设置 `axis = 1` 表示按列删除。`how = 'any'`时，表示某行或列只要有一个缺失值，就删除该行或列；当 `how = 'all'`，表示该行或列全部都为缺失值时，才删除该行或列
- `df.isna()` 判断 Pandas 数据帧是否为缺失值，是便用 `True` 占位，否便用 `False` 占位
- `df.notna()` 判断 Pandas 数据帧是否为非缺失值，是缺失值使用 `False` 占位，不是缺失值采用 `True` 占位
- `missingno.matrix()` 绘制缺失值热图
- `numpy.NaN` 产生 NaN 占位符
- `numpy.random.uniform()` 产生满足连续均匀分布的随机数
- `seaborn.heatmap()` 绘制热图
- `seaborn.pairplot()` 绘制成对特征分析图
- `sklearn.impute.KNNImputer()` 使用 k 近邻插补
- `sklearn.impute.MissingIndicator()` 将数据转换为相应的二进制矩阵 (`True` 和 `False`)，以指示数据中缺失值的存在位置
- `sklearn.impute.SimpleImputer()` 使用缺失值所在的行/列中的统计数据平均值 (`'mean'`)、中位数 (`'median'`) 或者众数 (`'most_frequent'`) 来填充，也可以使用指定的常数 `'constant'`



2.1 缺失值小传

在数据分析中，缺失值是指数据集中某些观测值或属性值没有被记录或采集到的情况。由于各种原因，数据中缺失值不可避免。缺失值通常被编码为空白，NaN 或其他占位符。处理缺失值是数据预处理中重要一环。

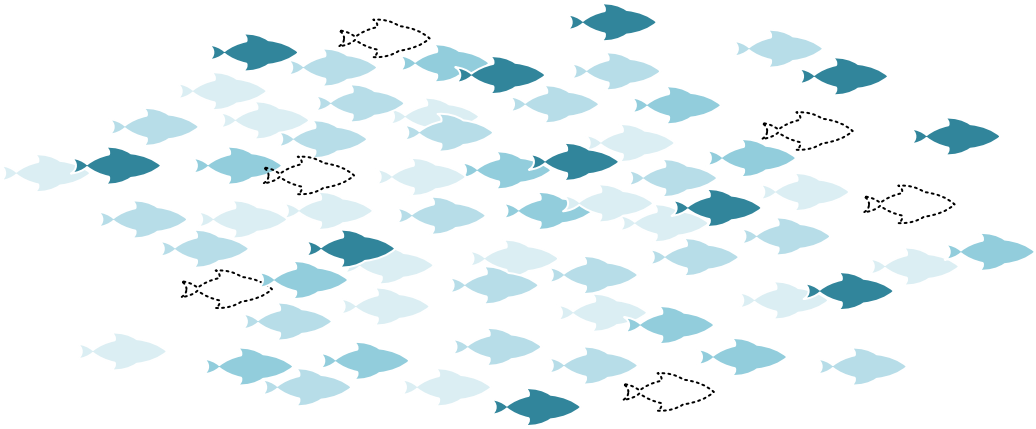


图 1. 缺失值

数据中缺失值产生的原因有很多。比如，在数据采集阶段，设备故障、人为失误、方法局限、拒绝参与调查、信息不完整等等可以造成数据缺失。另外，数据数据存储阶段也可能引入缺失值；比如，数据存储失败、存储器故障等等。

填补缺失值的方法有很多种，包括：

- **删除缺失值**：直接删除缺失值所在的行或列，但这可能会导致数据的丢失和分析结果的偏差。
- **插值法**：通过插值方法填补缺失值，如均值插值、中位数插值、最近邻插值、多项式插值等。
- **模型法**：使用回归、决策树或神经网络等模型预测缺失值，但需要先对数据进行训练和测试，可能会导致模型的过拟合和不准确。
- **多重填补法**：使用多个模型进行填补，可以提高填补缺失值的准确性和可靠性。

本章后文将专门介绍常见填补缺失值的方法。

NaN：非数

NaN 常用于表示缺失值。NaN 是 not a number 的缩写，中文含义是“非数”。numpy.nan 可以用来产生 NaN。举个例子，如果想要在已知数据帧 df 中，增加用 NaN 做占位符一列，就可以用 df['holder'] = np.nan，其中'holder'为这一列的标题 (header)。

一些 Numpy 函数在统计计算时，遇到缺失值会报错。表 1 第二列 Numpy 函数遇到缺失值 NaN，会直接报错。而表 1 第三列函数，计算时忽略 NaN。

表 1. 比较 Numpy 函数处理缺失值差异

	遇到 NaN，报错	计算时，忽略 NaN
均值	numpy.mean()	numpy.nanmean()
中位数	numpy.median()	numpy.nanmedian()
最大值	numpy.max()	numpy.nanmax()
最小值	numpy.min()	numpy.nanmin()
方差	numpy.var()	numpy.nanvar()
标准差	numpy.std()	numpy.nanstd()
分位	numpy.quantile()	numpy.nanquantile()
百分位	numpy.percentile()	numpy.nanpercentile()

原始数据中缺失值的样式没有特定标准，利用 pandas 读取数据时，可以设置缺失值样式。比如 read_csv() 读取 CSV 文件时，可以利用 na_values 设置缺失值样式，比如 na_values = 'Null'，再如 na_values = '?' 等等。在 Pandas 数据帧中，也用 NaT 表达缺失值。

以鸢尾花数据为例

本章以鸢尾花数据讲解如何处理缺失值。图 2 所示为完整的鸢尾花数据成对特征分析图，其中有 150 个数据点。

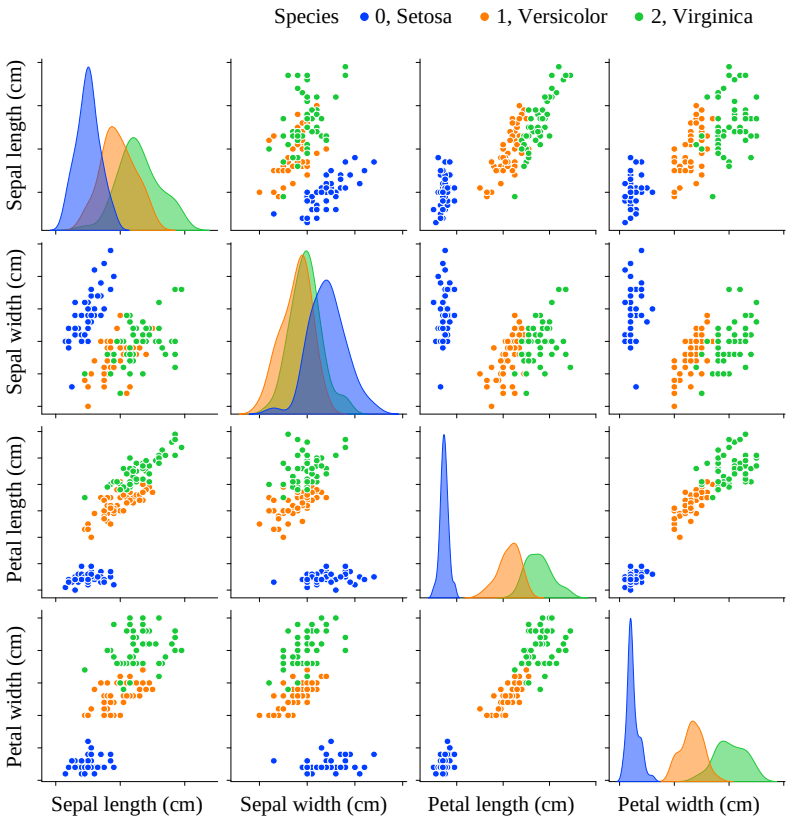


图 2. 鸢尾花原始数据，成对特征分析图

在鸢尾花原始数据中完全随机引入缺失值 NaN，将数据存为 iris_df_NaN，数据的形式如图 3 所示。图 4 所示为含有缺失值得鸢尾花可视化图像。

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	NaN	NaN	0.2
1	NaN	NaN	1.4	0.2
2	4.7	3.2	1.3	0.2
3	NaN	NaN	NaN	NaN
4	NaN	NaN	1.4	NaN
..	...	...	...	...
145	6.7	NaN	5.2	2.3
146	6.3	2.5	5.0	NaN
147	6.5	3.0	5.2	NaN
148	6.2	NaN	NaN	2.3
149	5.9	3.0	NaN	1.8

图 3. 鸢尾花样本数据，随机引入缺失值

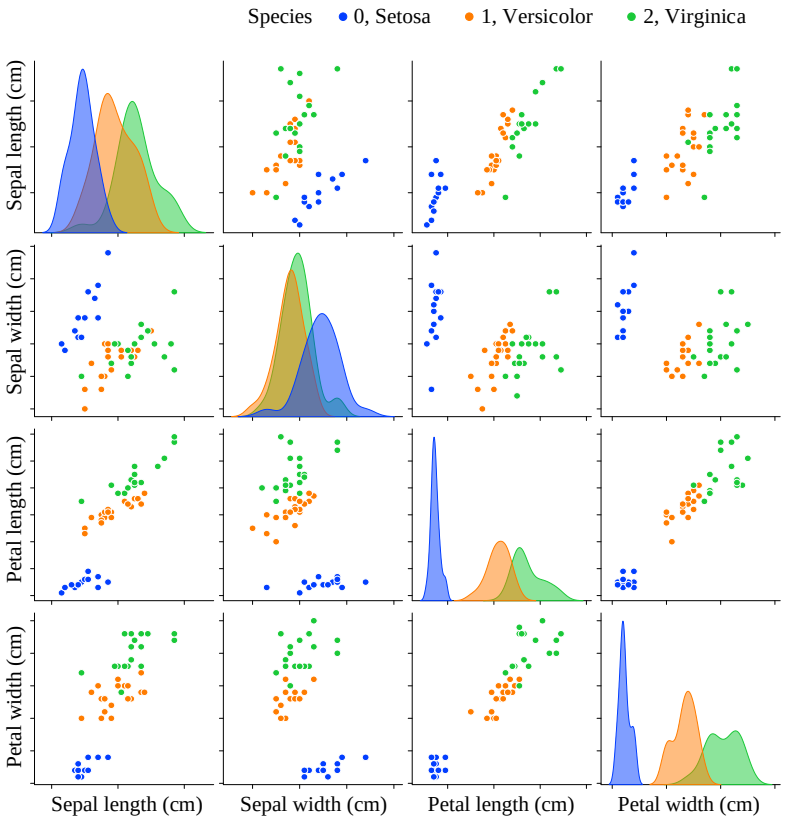


图 4. 鸢尾花数据可视化，引入缺失值

2.2 可视化缺失值位置

为了准确获取缺失值位置、数量等信息，对于 Pandas 数据帧数据可以采用 isna() 或 notna() 方法。

查找缺失值

采用 iris_df_NaN.isna()，返回具体位置数据是否为缺失值。数据缺失的话，为 True；否则，为 False。图 5 所示为 iris_df_NaN.isna() 结果。

	sepal length (cm)	sepal width (cm)	...	petal width (cm)	species
0	False	True	...	False	False
1	True	True	...	False	False
2	False	False	...	False	False
3	True	True	...	True	False
4	True	True	...	True	False
..	...	...	...	...	...
145	False	True	...	False	False
146	False	False	...	True	False
147	False	False	...	True	False
148	False	True	...	False	False
149	False	False	...	False	False

图 5. 判断数据是否为缺失值

图 6 所示为采用 seaborn.heatmap() 可视化数据缺失值，热图的每一条黑色条带代表一个缺失值。使用缺失值热图可以粗略观察得到缺失值分布情况。

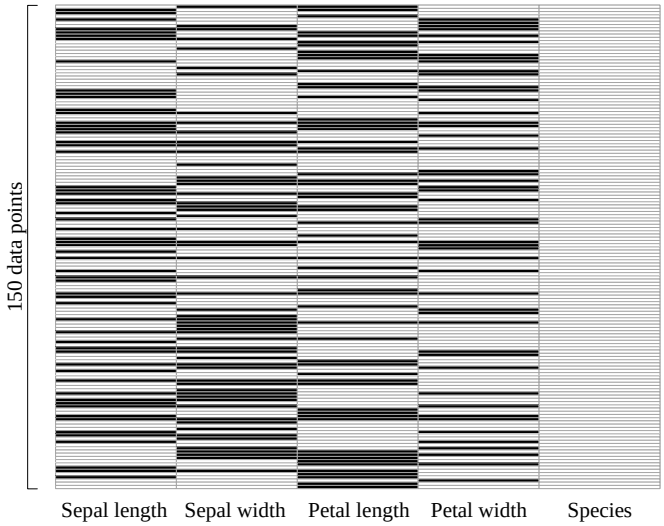


图 6. 缺失值可视化，每条黑带代表缺失值

查找非缺失值

方法 notna() 正好和 isna() 相反，iris_df_NaN.notna()判断数据是否为“非缺失值”；如果数据没有缺失，则为 True。图 7 所示为 iris_df_NaN.notna() 结果。

	sepal length (cm)	sepal width (cm)	...	petal width (cm)	species
0	True	False	...	True	True
1	False	False	...	True	True
2	True	True	...	True	True
3	False	False	...	False	True
4	False	False	...	False	True
..	...	...	...	...	...
145	True	False	...	True	True
146	True	True	...	False	True
147	True	True	...	False	True
148	True	False	...	True	True
149	True	True	...	True	True

图 7. 判断数据是否为“非缺失值”

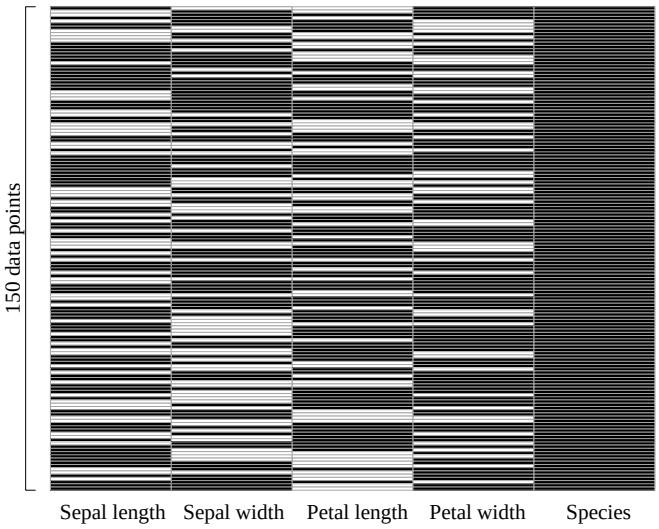


图 8. 缺失值可视化，每条白带代表缺失值

非缺失值变化线图

另外，可以安装 missingno，并调用 missingno.matrix() 绘制缺失值热图，具体如图 9 所示。这幅图最右侧还展示每行非缺失值数据数量的变化线图，线图最小取值为 1，最大取值为 5。取值为 1 时，每行只有一个非缺失值；取值为 5 时，该行不存在缺失值。观察这幅线图，可以帮助我们解读缺失值分布特征。

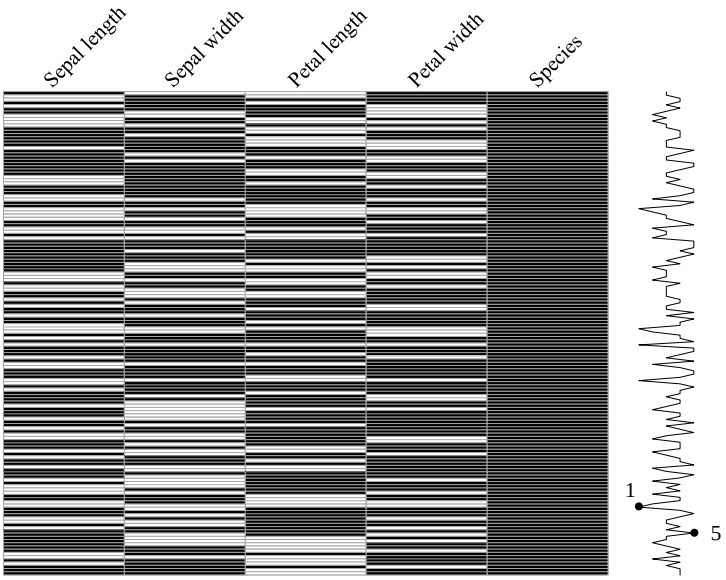


图 9. missingno.matrix()绘制缺失值热图，每条白带代表缺失值

总结缺失值信息

对于 pandas 数据帧，也可以采用 info() 显示数据非缺失值数量和数据类型。图 10 所示为 iris_df_NaN.info() 结果。df.isnull().sum() * 100 / len(df) 则计算每列缺失值的百分比。

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   sepal length (cm)  85 non-null     float64
 1   sepal width (cm)   94 non-null     float64
 2   petal length (cm)  91 non-null     float64
 3   petal width (cm)   84 non-null     float64
 4   species            150 non-null    int32
dtypes: float64(4), int32(1)
memory usage: 5.4 KB
```

图 10. pd.info() 总结样本数据特征

也可以采用 sklearn.impute.MissingIndicator() 函数将数据转换为相应的二进制矩阵 (True 和 False，相当于 1 和 0)，以指示数据中缺失值的存在位置。

2.3 处理缺失值

图 11 总结常用处理缺失值的方法。

对于表格数据，一般情况，每一行代表一个样本数据，每一列代表一个特征。处理存在缺失值数据集的基本策略是舍弃包含缺失值的整行或整列。但是，这是以丢失可能有价值的数据为代价的。

更好的策略是估算缺失值，即从数据的已知部分推断出缺失值，这种方法统称**插补**(imputation)。本章后续主要介绍连续数据的删除和插补方法。

本书第 6 章将专门介绍时间序列数据的插补。

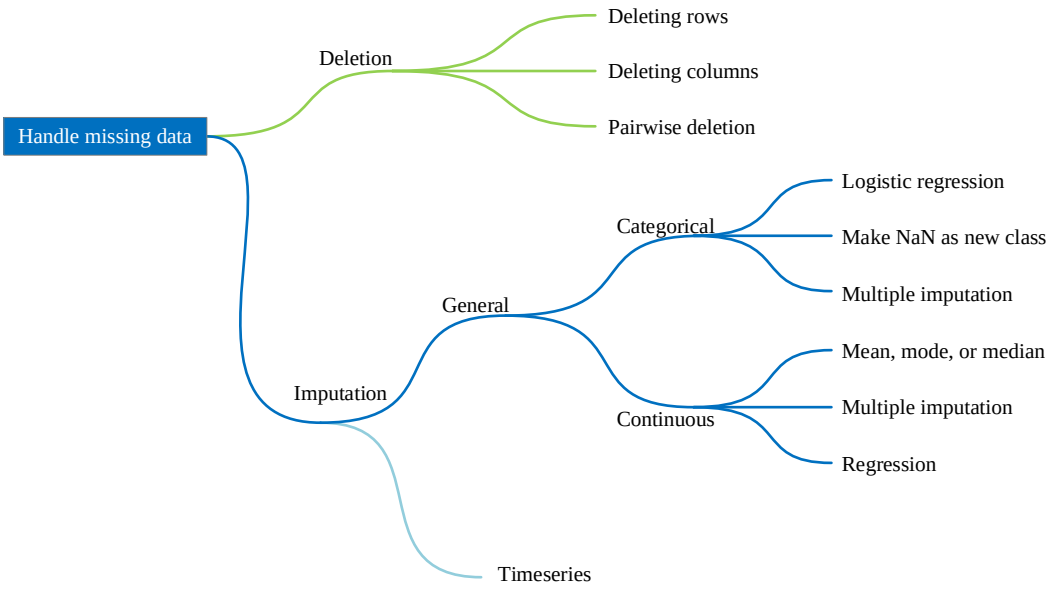


图 11. 处理缺失值的方法分类

2.4 删除：最基本方法

本节简单介绍 Pandas 数据帧 dropna() 方法。

对于某一个数据帧 df，df.dropna(axis = 0, how = 'any') 中 axis = 0 为按行删除，设置 axis = 1 表示按列删除。how = 'any'时，表示某行或列只要有一个缺失值，就删除该行或列，如图 12 所示。

如图 13 所示，当 how = 'all'，表示该行或列全部都为缺失值时，才删除该行或列。dropna()方法默认设置为 axis = 0，how = 'any'。

```
df.dropna(axis=0, how='any')
```

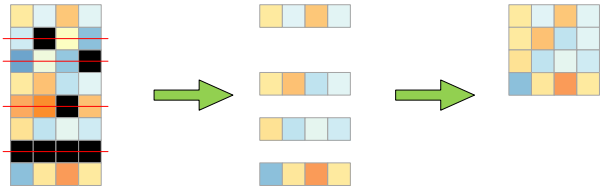


图 12. Pandas 数据帧中删除含有至少一个缺失值所在的行

```
df.dropna(axis=0, how='all')
```

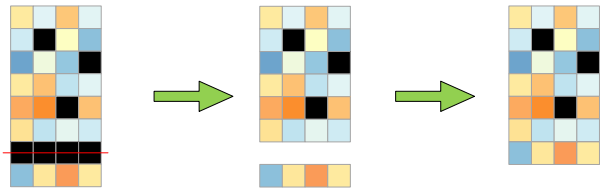


图 13. Pandas 数据帧中删除全为缺失值行

图 14 所示为删除缺失值后的鸢尾花数据，规则为删除含有至少一个缺失值所在的行。对比图 4，可以发现非缺失数据点明显减小。图 14 中所剩数据便是图 9 中最右侧线图值为 5 对应的数据点。

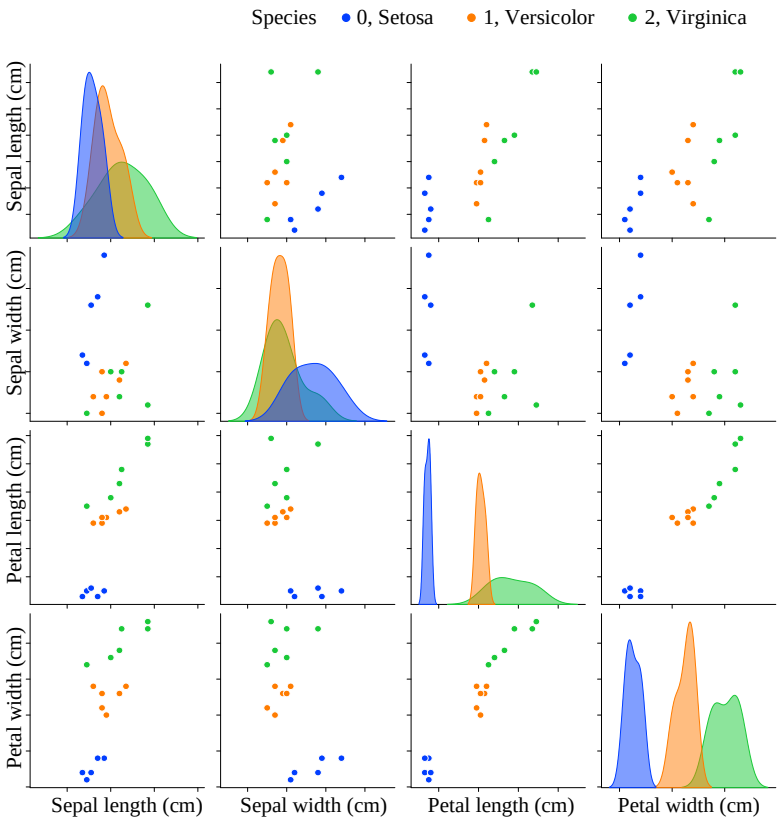


图 14. 鸢尾花数据，删除含有至少一个缺失值所在的行

一般情况每列数据代表一个特征，删除整列特征的情况也并不罕见。不管是删除缺失值所在的行或列，都会浪费大量有价值的信息。

成对删除

成对删除 (pairwise deletion) 是一种特别的删除方式，进行多特征联立时，成对删除只删除掉需要执行运算特征包含的缺失数据；以估算方差协方差矩阵为例，如图 15 所示，计算 X_1 和 X_3 的相关性，只需要删除 X_1 和 X_3 中缺失值对应的数据点。

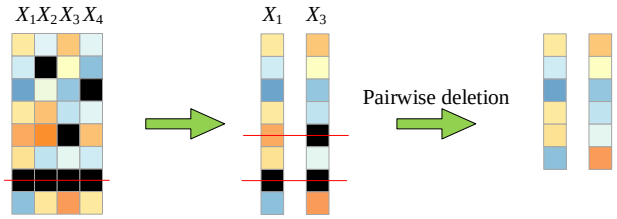


图 15. 成对删除

2.5 单变量插补

相对删除缺失值，更常用的方法是，采用一定的方法补全缺失值，我们称之为**插补** (imputation)。如图 11 所示，分类数据和连续数据采用的方法也稍有差别。

⚠ 注意，选取采用插补方法要格外小心，如果填充方法不合理，会引入数据噪音，并造成数据分析结果不准确。

时间数据采用的插补方法不同于一般数据。Pandas 数据帧有基本插补功能，特别是对于时间数据，可以采用**插值** (interpolation)、向前填充、向填充。这部分内容，我们将在本书插值和时间序列部分详细介绍。

单变量插补：统计插补

本节专门介绍，单变量插补。单变量插补也称统计插补，仅使用第 j 个特征维度中的非缺失值插补该特征维度中的缺失值。本节采用的函数是 sklearn.impute.SimpleImputer()。

SimpleImputer() 可以使用缺失值所在的行/列中的统计数据平均值 ('mean')、中位数 ('median') 或者众数 ('most_frequent') 来填充，也可以使用指定的常数 'constant'。

如果某个特征是是连续数据，可以根据在其他所有非缺失值平均值或中位数来填充该缺失值。

如果某个特征是是分类数据，则可以利用该特征非缺失值的众数，即出现频率最高的数值来补齐缺失值。

图 16 所示为采用中位数插补鸢尾花缺失值。观察图 16，可以发现插补得到的数据形成“十字”图案。

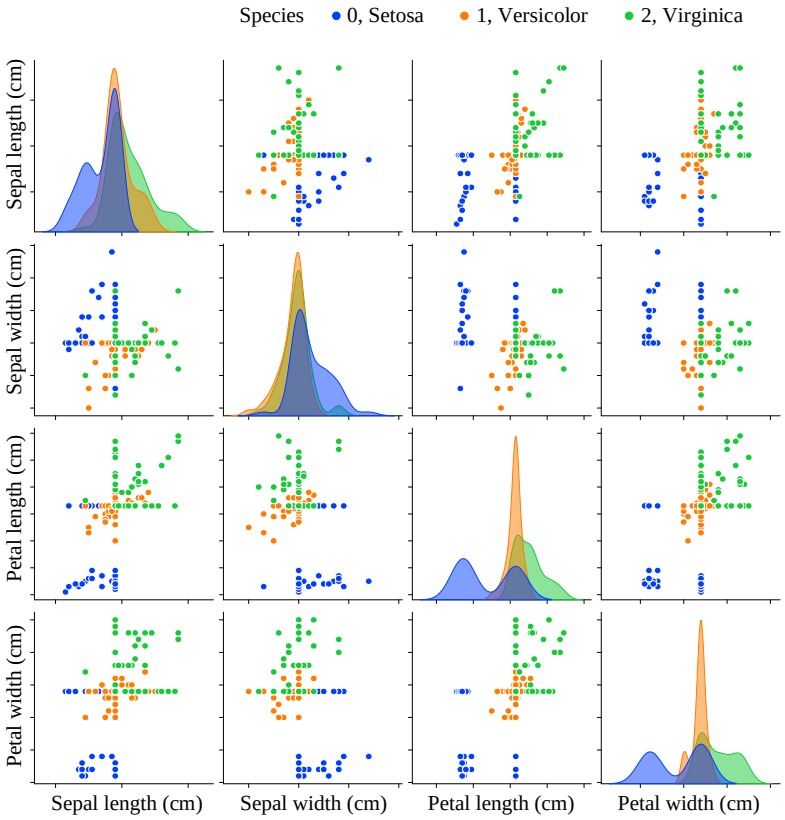


图 16. 鸢尾花数据，采用中位数插补缺失值

2.6 k 近邻插补

本节介绍 *k* 近邻插补。**k 近邻算法** (*k*-nearest neighbors algorithm, *k*-NN) 是最基本有监督学习方法之一，*k*-NN 中的 *k* 指的是“近邻”的数量。*k*-NN 思路很简单——“近朱者赤，近墨者黑”。更准确地说，小范围投票，**少数服从多数** (majority rule)。

《机器学习》第 2 章专门介绍 *k* 近邻算法这种监督学习方法。

本节介绍 *k* 近邻插补的函数为 sklearn.impute.KNNImputer()。利用 KNNImputer 插补缺失值时，先给定距离缺失值数据最近的 *k* 个样本，将这 *k* 个值等权重平均或加权平均来插补缺失值。图 17 所示为采用 *k* 近邻插补鸢尾花数据结果。

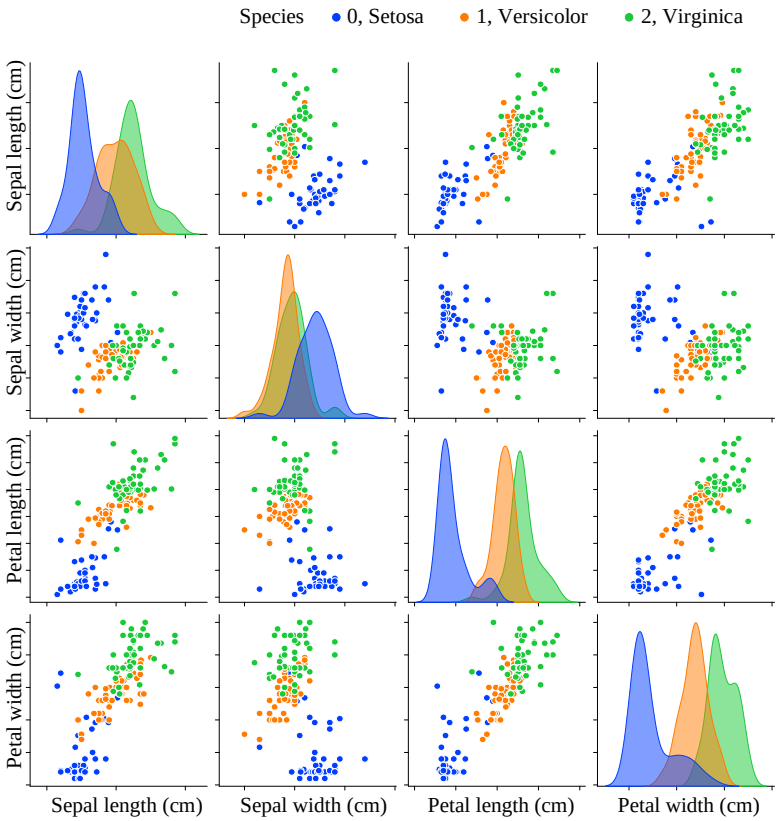


图 17. 鸢尾花数据，最近邻插补

2.7 多变量插补

多变量插补，利用其它特征数据来填充某个特征内的缺失值。具体来说，多变量插补将缺失值所在变量视为预测目标变量，使用其他已知变量作为预测变量，通过建立回归或分类模型来预测缺失值，并进行填补。相比于单变量插补方法，多变量插补能够更充分地利用数据集中的信息，从而提高填补结果的准确性和可靠性。多变量插补的常见方法包括线性回归、随机森林、神经网络等。

多变量插补通常将缺失值建模为其他特征的函数，用该函数估算合理的数值，以填充缺失值。整个过程可以用迭代循环方式进行。比较来看，单变量插一般仅考虑单一特征进行插补，而多变量插补考虑不同特征数据的联系。

图 18 所示为采用 sklearn.impute.IterativeImputer() 函数完成多变量插补，补齐鸢尾花数据中缺失值。

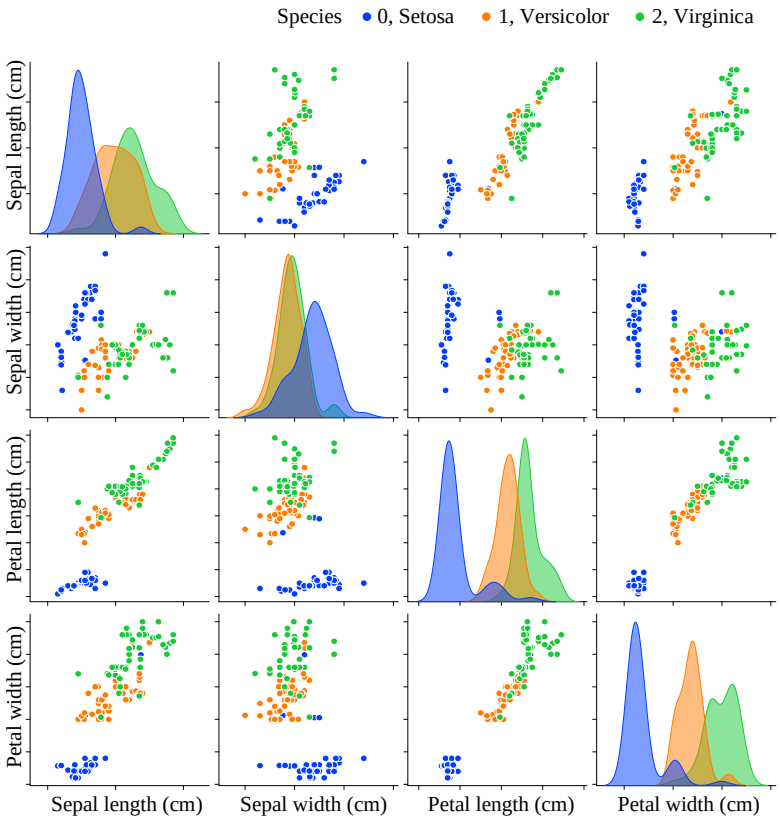


图 18. 鸢尾花数据，多变量插补

Bk6_Ch02_01.py 绘制本章大部分图像。

缺失值是指数据集中某些观测值或属性值没有被记录或采集到的情况。缺失值可能会影响数据分析结果的准确性和偏差，产生原因包括数据采集问题、处理问题、参与者拒绝回答等。解决方法包括删除缺失值、插值法、模型法和多重填补法。注意要根据具体情况选择最合适的处理方法，以确保数据分析的准确性和可靠性。

有关数据帧处理缺失值，请大家参考：

https://pandas.pydata.org/pandas-docs/stable/user_guide/missing_data.html

sklearn.impute.IterativeImputer() 函数非常灵活，可以和各种估算器联合使用，比如决策树回归、贝叶斯岭回归等等。感兴趣的读者可以参考：

https://scikit-learn.org/stable/modules/impute.html

3 离群值

Detecting Outliers

利用统计方法和机器学习算法发现、处理离群值

数学领域，提出问题比解决问题，更珍贵。

In mathematics the art of proposing a question must be held of higher value than solving it.

—— 格奥尔格·康托尔 (Georg Cantor) | 德国数学家 | 1845 ~ 1918

- `numpy.percentile()` 计算百分位
- `pandas.DataFrame()` 构造 pandas 数据帧
- `seaborn.boxplot()` 绘制箱型图
- `seaborn.histplot()` 绘制直方图
- `seaborn.kdeplot()` 绘制概率密度估计曲线
- `seaborn.pairplot()` 绘制成对分析图
- `seaborn.rugplot()` 绘制 rug 图像
- `seaborn.scatterplot()` 绘制散点图
- `sklearn.covariance.EllipticEnvelope()` 协方差椭圆法检测离群值
- `sklearn.ensemble.IsolationForest()` 孤立森林检测离群值
- `sklearn.svm.OneClassSVM()` 支持向量机检测离群值
- `stats.probplot()` 绘制 QQ 图

3.1 离群值小传

离群值 (outlier)，又称逸出值、离群值，是指数据集中与其他数据点有显著差异的数据点，也就是说明显地偏大或偏小。离群值可能是由于异常情况、错误测量、数据录入错误或意外事件等原因而产生。离群值可能会对数据分析和建模造成问题，因为它们可能导致误差或偏差，并降低模型的准确性。因此，数据分析师通常会对数据集中的离群值进行检测和处理。

常见的离群值检测方法包括基于统计学的方法、基于距离的方法、基于密度的方法和基于模型的方法。处理离群值的方法包括删除、替换、调整或利用异常值建立新的模型等。

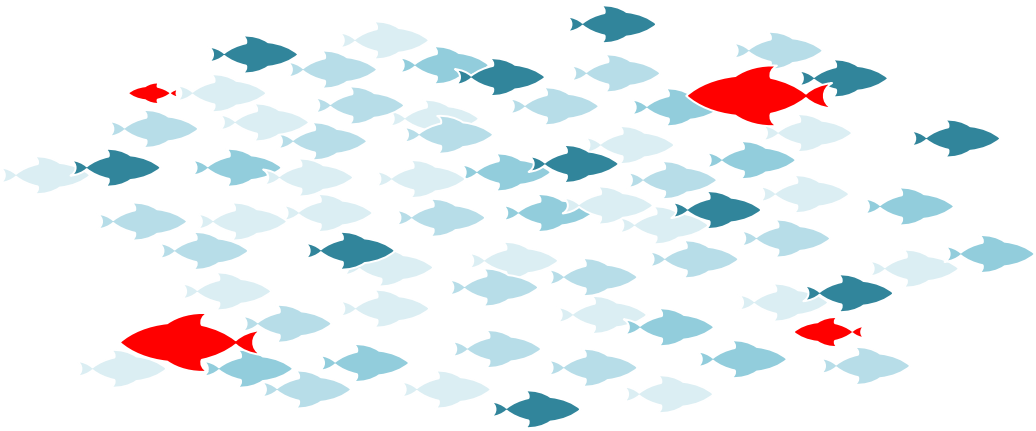


图 1. 离群点

离群值破坏力

离群值可以具有很强的破坏力。比如，离群值可能给最大值、最小值、极差、平均值、方差、标准差、线性相关性系数、分位等统计量计算带来偏差。

图 2 所示为离群值对**线性回归** (linear regression) 的影响。再举个例子，实践中，大家会发现离群值对于时间序列相关性系数计算破坏力更大。这一章专门介绍各种发现离群值的工具。

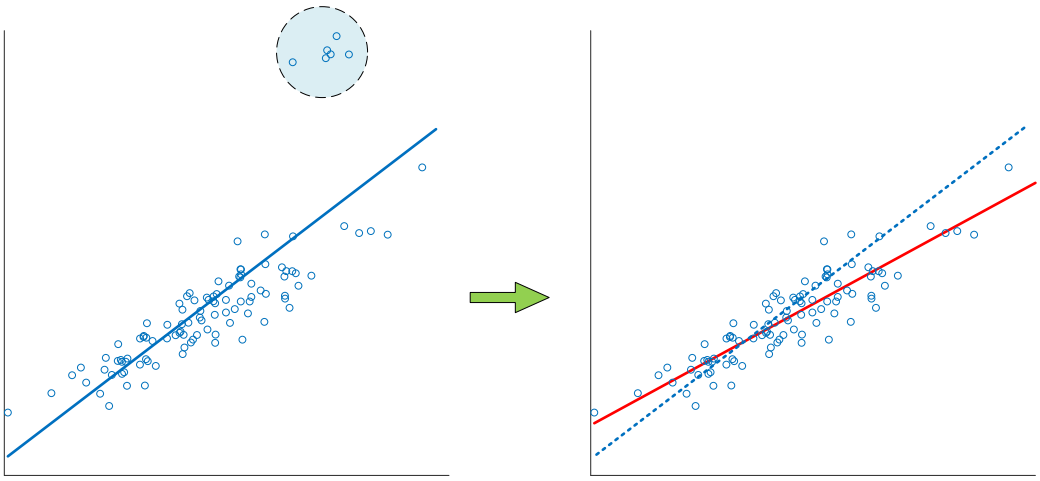


图 2. 离群点对回归分析的影响

工具

如图 3 所示，判断离群值的方法有很多。本章将围绕图 3 中主要方法展开。这幅图也相当于是本章的思维导图。

最简单的方法是，观察样本数据的最大值和最小值，根据生活常识或专业知识判断，取值范围是否合理。比如，鸢尾花数据集中，如果出现某个样本点的花萼长度为 5.2 米，这显然是个离群点。再举例，鸢尾花任何特征数值肯定不能是负数。

确定离群值之后，需要合理处理。常见的办法有，比如通过设为 NaN 将其删除，或者填充。填充的方法很多，可以参考上一章内容。

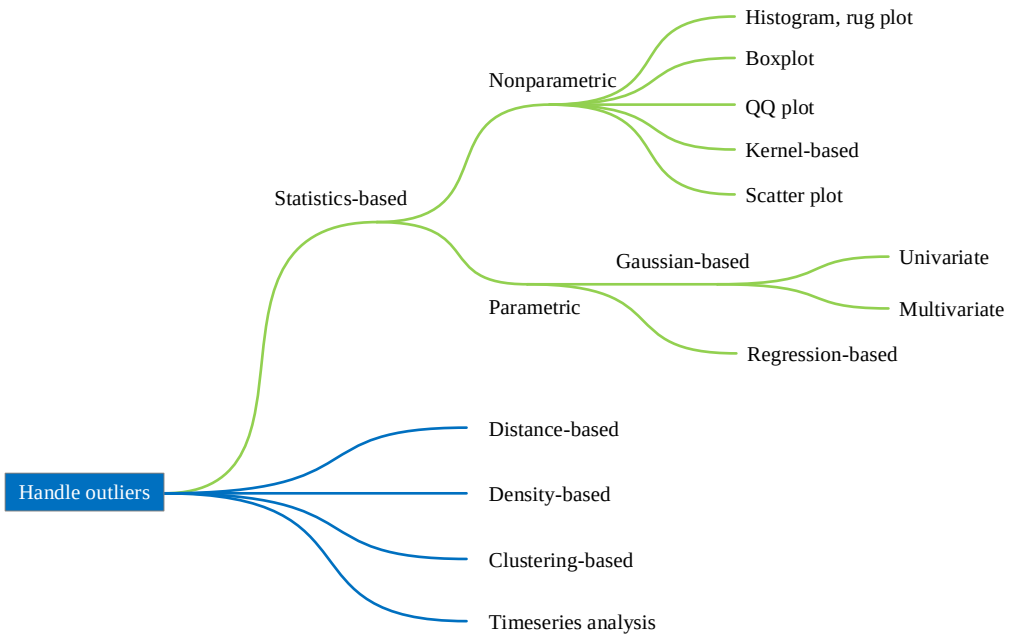


图 3. 处理离群点的常见方法

3.2 直方图：单一特征分布

鸢尾花书《统计至简》第 2 章专门介绍过**直方图** (histogram)。

可以通过观察数据的直方图来初步判断单一特征的分布情况以及可能存在的离群值。

百分位

图 4 所示鸢尾花四个特征数据的直方图。将数据顺序排列，离群值肯定出现分布的两端。比如，在图 4 上，绘制 1%和 99%百分位所在位置。可以 1%和 99%百分位用来界定数据分布的“左尾”和“右尾”。

回顾一下，百分位 (percentile) 是指一个数值在一组数据中的排名位置，表示该数值小于等于百分位数的观测值所占的百分比。例如，50%百分位数是中位数，表示一半的数据小于等于中位数，另一半的数据大于等于中位数。

另外，25%、50%和 75%这三个百分位也同样重要，图 5 给出了鸢尾花四个特征的这三个百分位所在位置。下一节讲解箱型图时，将使用 25%、50%和 75%这三个百分位。

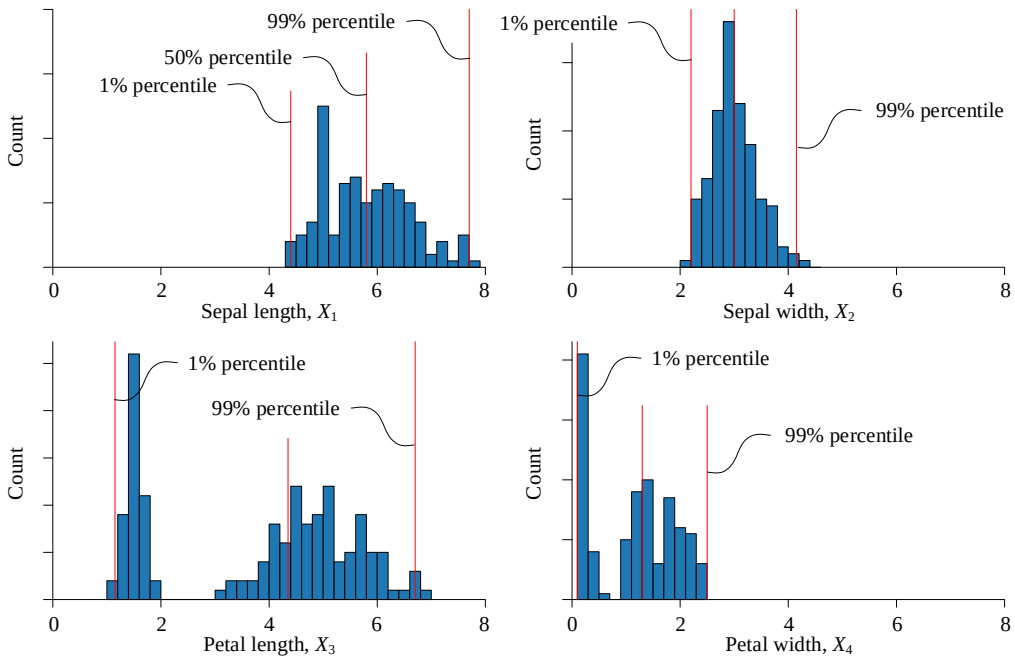


图 4. 鸢尾花数据直方图，以及 1%和 99%百分位

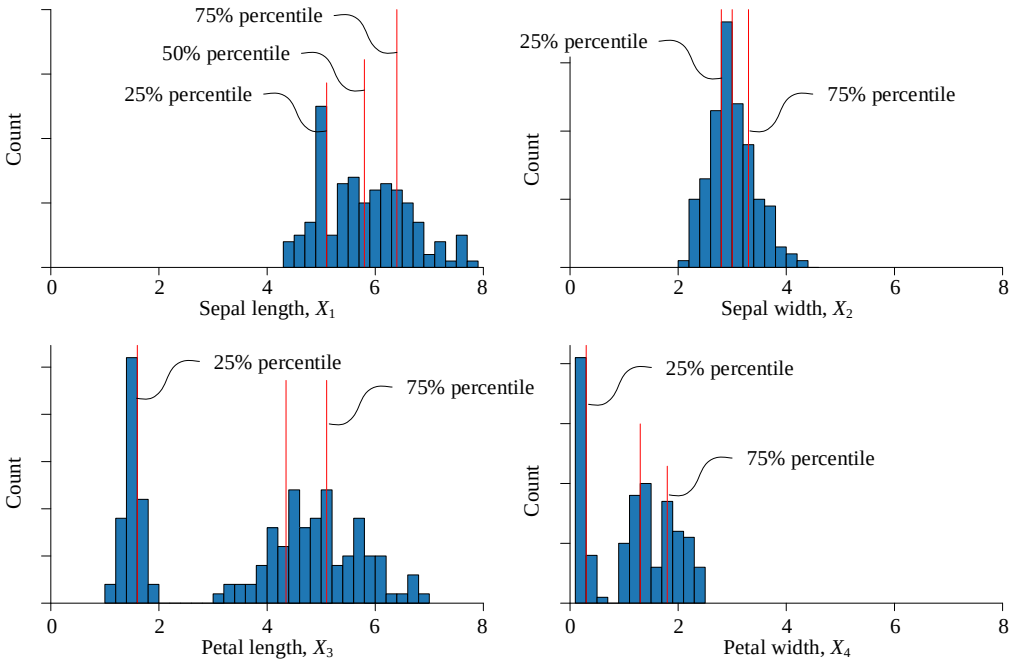


图 5. 鸢尾花数据直方图，以及 25%、50%和 75%百分位

山脊图

图 6 所示为采用 joypy 绘制的山脊图，也可以用来发现分类数据中潜在离群值。

《可视之美》曾专门介绍过山脊图。

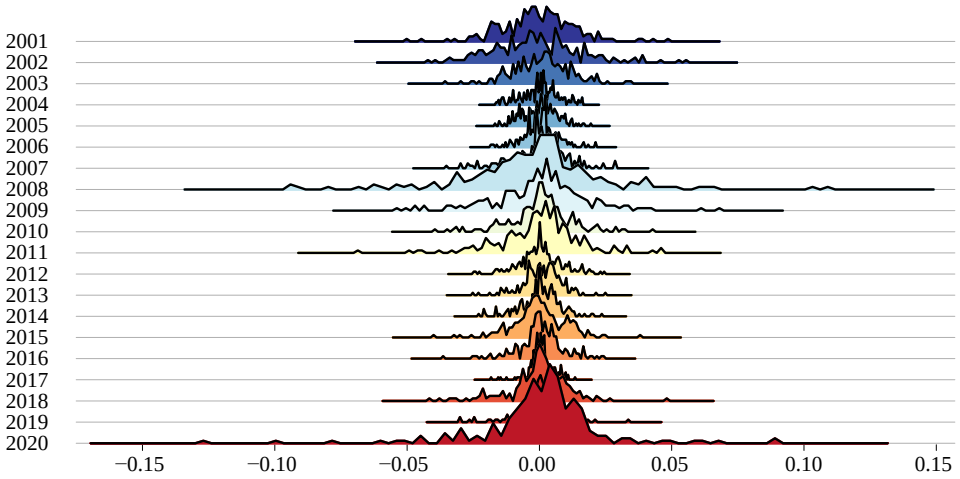


图 6. 标普 500 日收益率数据

概率密度估计 + rug 图

概率密度估计图像也可以用来观察异常值。概率密度估计 (Probability Density Estimation) 是指根据有限样本数据推断出未知概率密度函数的过程，常用于探索性数据分析和模型构建中。通过估计概率密度函数，可以更好地理解数据的分布特征、模型参数和模型拟合度。

高斯核密度估计 (Gaussian Kernel Density Estimation)，或高斯 KDE，是一种常用的概率密度估计方法，基于高斯核函数对数据进行平滑处理，估计未知的概率密度函数。该方法对连续变量的数据有较好的适用性，可以用于探索数据分布、识别离群值和构建概率模型等任务。

图 7 所示为高斯 KDE 图像，叠加 rug 图。图上同样标出 1%和 99%百分位点位置。rug 图是一种数据可视化方法，用于展示数据分布和密度。它将每个数据点在 x 轴上表示为一条短线，形成了数据点的密度分布图。rug 图通常与直方图或核密度图结合使用，可以更直观地显示数据集的分布情况。

《统计至简》专门讲解概率密度估计，请大家回顾高斯核密度估计。

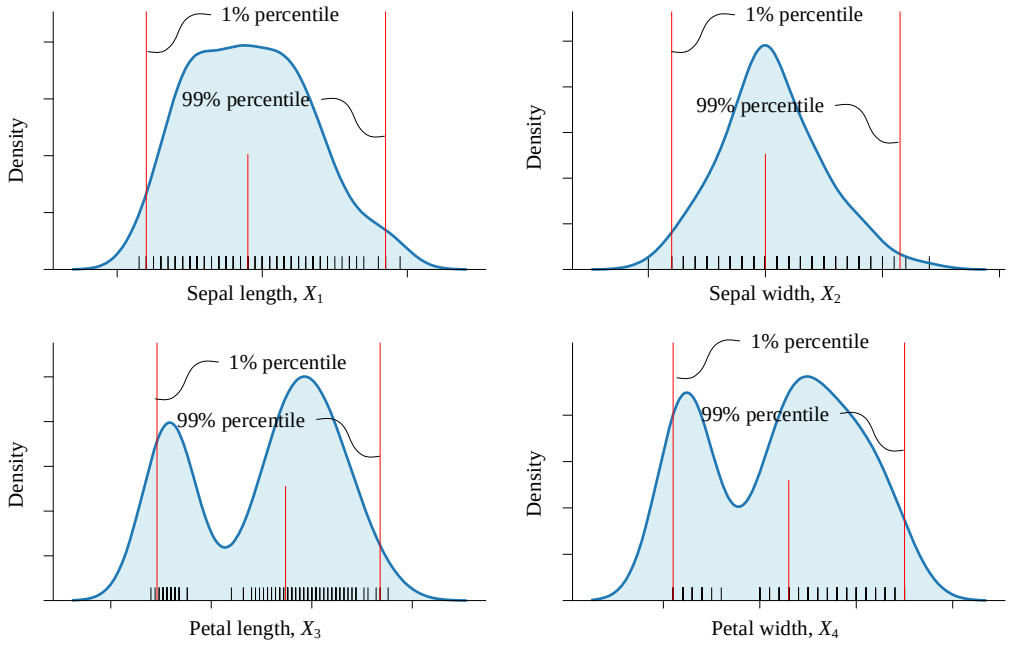


图 7. KDE 密度估计，叠加 rug 图

缩尾调整

缩尾调整 (winsorize) 是将超出变量特定百分位范围的数值替换为其特定百分位数值的方法。缩尾调整通过截断分布的长尾部分来减少异常值对估计结果的影响。在实际应用中，我们可以根据领域知识或经验选择合适的截断点，并将超出截断点的异常值设置为固定的截断值。缩尾调整

可以改善分布拟合和参数估计的稳定性和精度，但也可能引入信息损失和偏差。在选择截断点时需要谨慎，并在分析前后进行敏感性分析。

请参考如下链接学习如何使用 scipy.stats.mstats.winsorize() 函数进行缩尾调整：

https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.mstats.winsorize.html

3.3 散点图：成对特征分布

本章前文所讲的可视化方案均用来发现单一特征可能存在的离群值。采用散点图，发现成对特征数据可能存在的离散点。鸢尾花书读者对散点图肯定很熟悉。**散点图** (scatter plot) 是一种常用的数据可视化方法，用于展示两个变量之间的关系。散点图将每个数据点表示为一个点，在二维坐标系上绘制，其中一个变量在横轴上表示，另一个变量在纵轴上表示。

散点图可以帮助我们直观地观察变量之间的相关性、趋势和异常值，是探索性数据分析和建模中不可或缺的工具。散点图还可以用于比较不同组之间的变化和趋势，或者用不同的颜色或形状表示不同的组或类别。

图 8 所示为鸢尾花数据花萼长度、花萼宽度散点图。图 8 中还绘制了单一特征的 rug 图。

此外，也可以使用如图 9 成对特征数据来观察数据分布，以及可能存在的离群值。

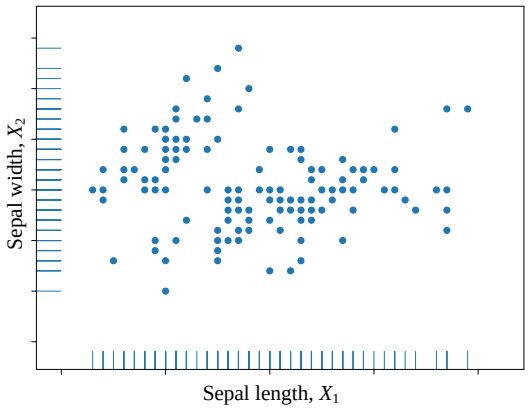


图 8. 散点图，横轴花萼长度，纵轴花萼宽度

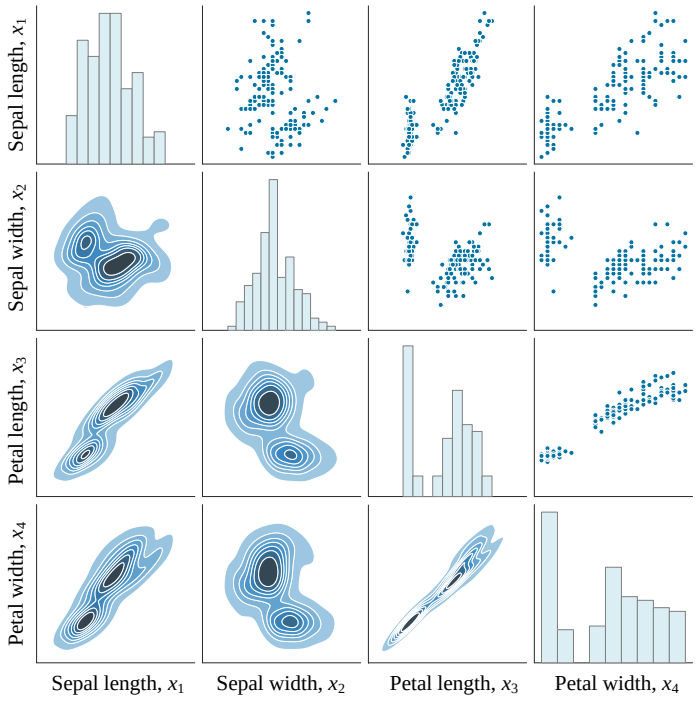


图 9. 鸢尾花数据成对特征分析图

3.4 QQ 图：分位数–分位数

《统计至简》第 9 章专门介绍过 QQ 图。

QQ 图 (Quantile-Quantile plot) 是一种用于检查数据是否符合某种理论分布的数据可视化方法。QQ 图将样本数据的分位数与理论分布的分位数进行比较，并将它们绘制在同一坐标系中。如果数据符合理论分布，则点将沿着一条直线分布。如果数据偏离理论分布，则点将偏离直线。通过观察点的分布情况，我们可以判断数据是否符合某种理论分布，或者是否存在偏差或离群值等问题。QQ 图常用于正态性检验、分布拟合和模型诊断等任务。

QQ 图的横坐标通常是理论分布的分位数，纵坐标通常是样本数据的分位数。在正态 QQ 图中，横坐标通常是标准正态分布的分位数，或 Z 分数；纵坐标是样本数据的分位数。在其他类型的 QQ 图中，横坐标和纵坐标的标尺将取决于所使用的理论分布和样本数据的类型。

图 10 所示为 QQ 图原理，图中横轴为正态分布的分位数。

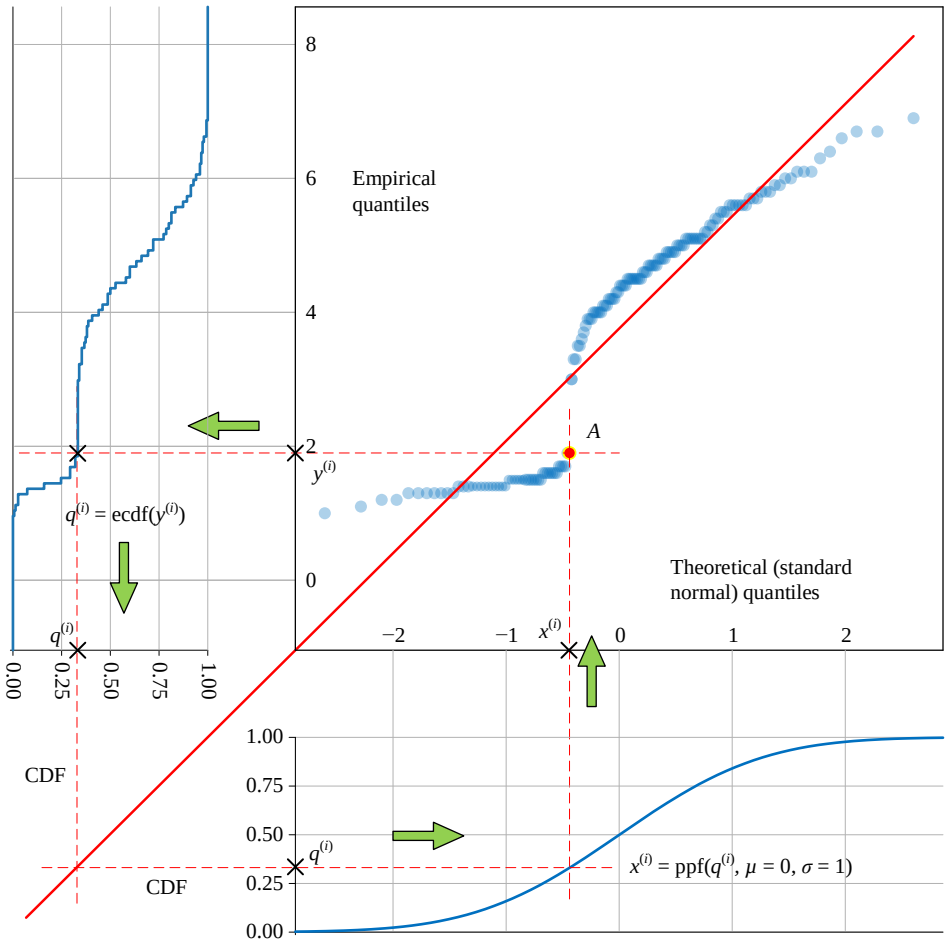


图 10. QQ 图原理，横轴为正态分布，图片来自《统计至简》第 9 章

图 11 到图 14 分别给出鸢尾花四个特征数据的直方图和 QQ 图。容易发现不同的数据分布，对应特定的 QQ 图分布特点。

《统计至简》第 9 章介绍过如何通过 QQ 图形态判断原始数据分布特点，请大家自行回顾，本节不再重复。

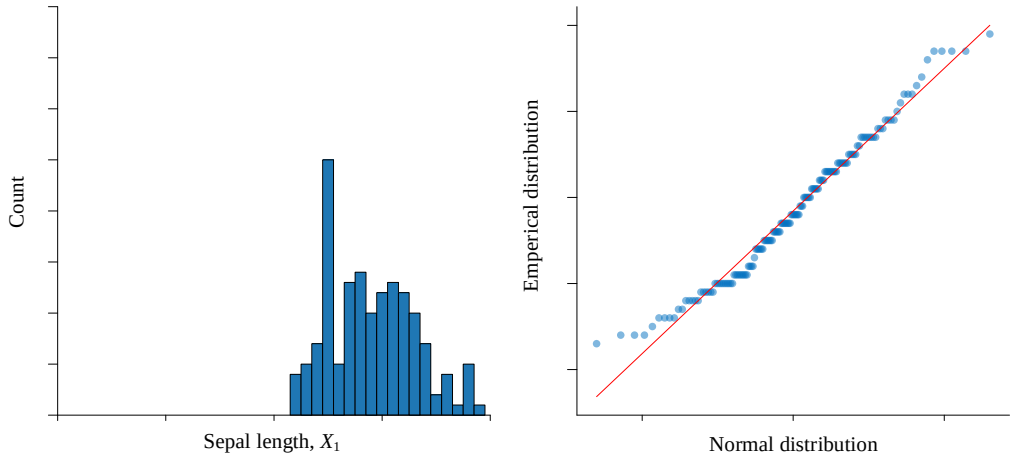


图 11. 花萼长度直方图和 QQ 图

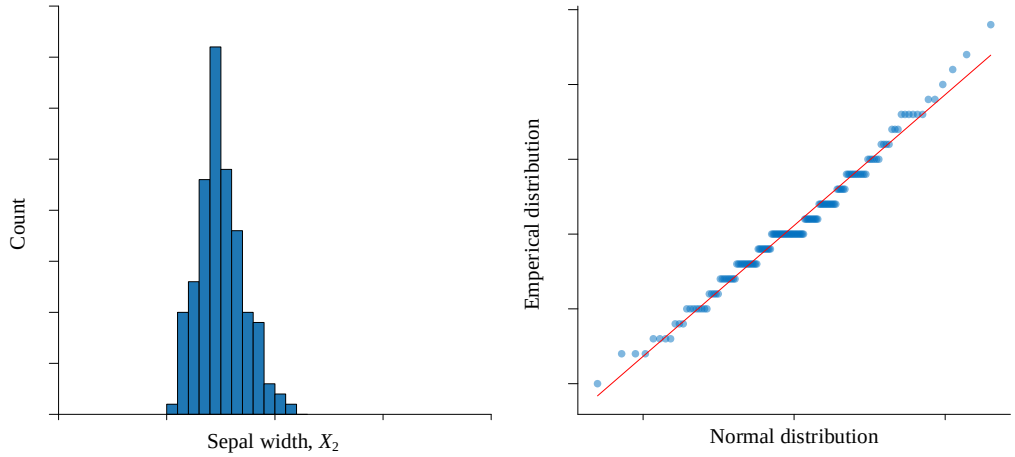


图 12. 花萼宽度直方图和 QQ 图

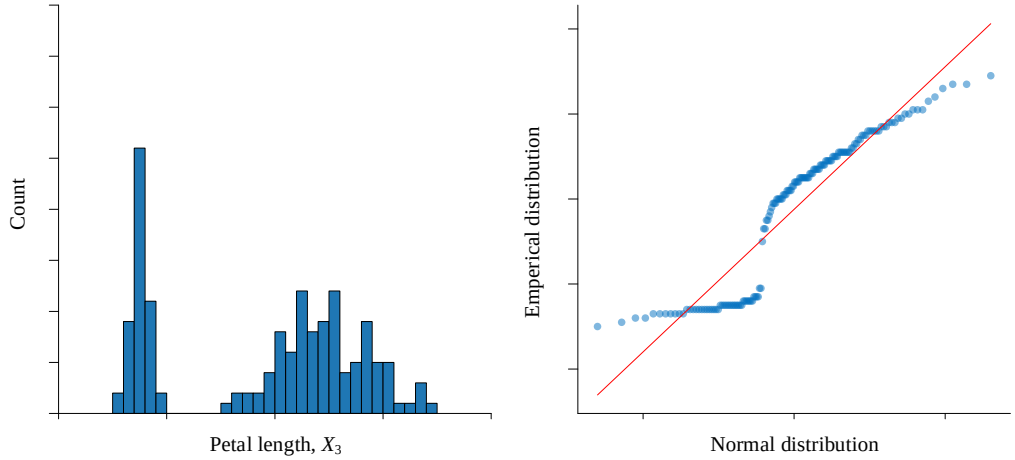


图 13. 花瓣长度直方图和 QQ 图

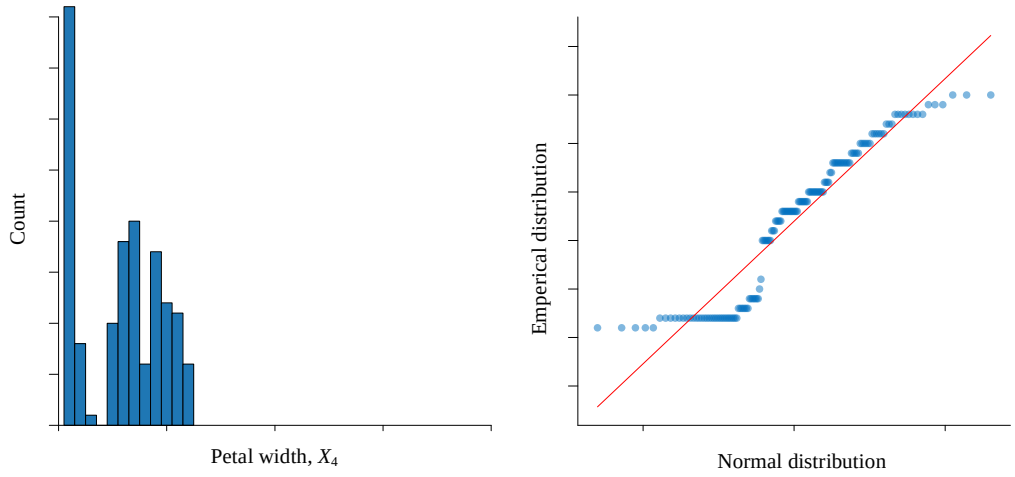


图 14. 花瓣宽度直方图和 QQ 图

3.5 箱型图：上界、下界之外样本

《统计至简》第 2 章专门介绍箱型图。

箱型图 (box plot) 是一种展示数据分布和离群值的方法。箱型图通过绘制数据的四分位数 (Q_1、Q_2、Q_3) 和可能的离群值来呈现数据的位置和离散程度。箱型图常用于探索性数据分析和统计推断，可用于比较不同组之间的数据分布和趋势。

图 15 所示为箱型图原理。Q_1 也叫下四分位，Q_2 也叫中位数，Q_3 也称上四分位。

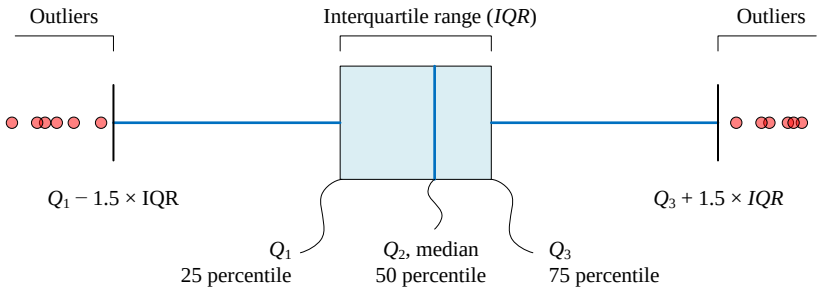


图 15. 箱型图原理

箱型图的**四分位间距** (interquartile range) 的定义为：

$$IQR = Q_3 - Q_1 \tag{1}$$

在 $[Q_1 - 1.5 \times IQR, Q_3 + 1.5 \times IQR]$ 之外的样本数据则可能是离群点。图 16 所示为鸢尾花数据的箱型图。$Q_3 + 1.5 \times IQR$ 也称上界，$Q_1 - 1.5 \times IQR$ 叫下界。

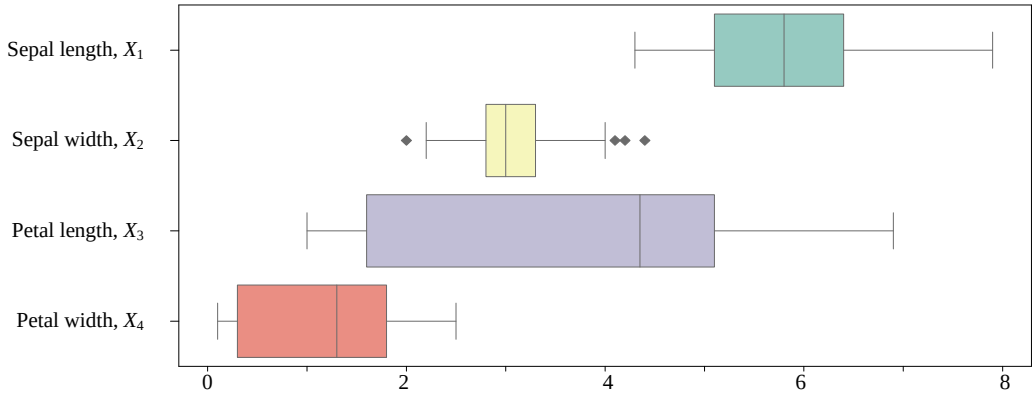


图 16. 鸢尾花箱型图

3.6 Z 分数：样本数据标准化

从大到小排列一组 n 个样本数据，离群值肯定出现在序列的两端。首先计算出数据的样本均值 $\bar{x}$，和样本标准差 s。若任何数据点与均值的偏差绝对值大于三倍标准差，则可以判定数据点为离群点，即满足下式的 x 可能是离群值：

$$|x - \bar{x}| > 3s \tag{2}$$

⚠ 大家需要注意极大的离群值会“污染”样本均值。因此，实践中，也常用样本中位数作为基准。

三倍标准差±3s 相当于 99.7%置信度，对应显著性水平 $\alpha = 0.003$。此外，也可以采用两倍标准差±2s，这相当于 95%置信度，即 $\alpha = 0.05$。

➡ 图 17 展示了《统计至简》第 9 章介绍的 68–95–99.7 法则，请大家回顾。

⚠ 注意，图 17 中并不区分总体标准差 σ 和样本标准差 s，并假设均值为 0。

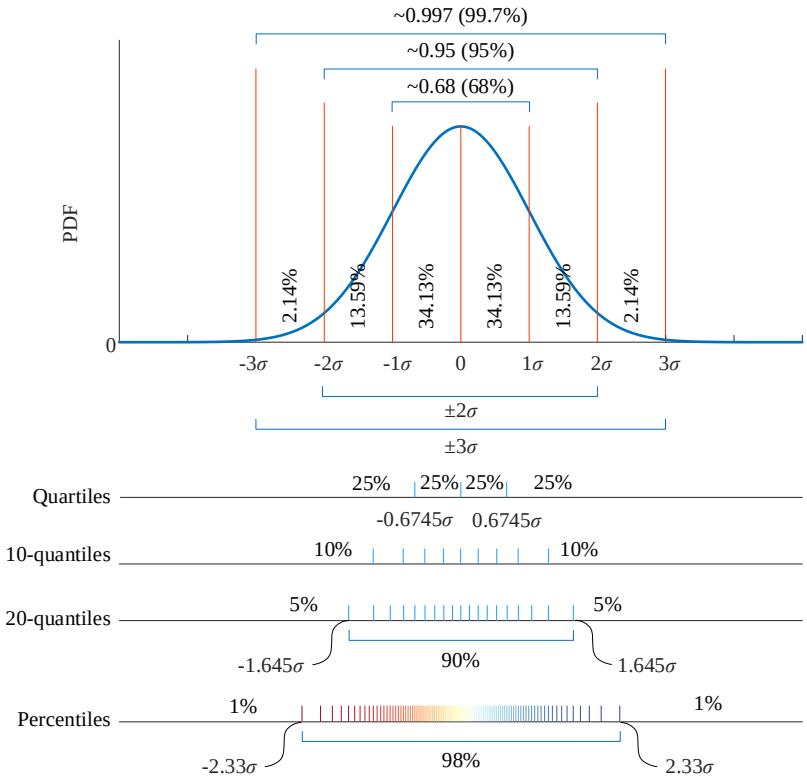


图 17. 标准差，注意图中并不区分总体标准差 σ 和样本标准差 s

Z 分数

Z 分数 (Z score) 是一种用于标准化数据的方法。Z 分数表示一个数据点距离均值的标准差数目，通常用于将不同尺度和分布的数据标准化为标准正态分布。Z 分数可以帮助我们比较不同数据点之间的相对位置和大小，判断数据是否偏离均值，并进行异常值检测和离群值分析。在实际应用中，Z 分数也经常用于构建模型、计算概率和决策阈值等任务。

从 Z 分数角度，(2) 相当于：

$$z = \frac{|x - \bar{x}|}{s} > 3 \tag{3}$$

也就是任何数据点的 Z 分数绝对值大于 3，即 z 分数大于 3 或小于−3，可以判定数据点为离群点。图 18 所示为鸢尾花数据四个特征的 Z 分数。

《统计至简》第 9 章还介绍过 Z 分数。

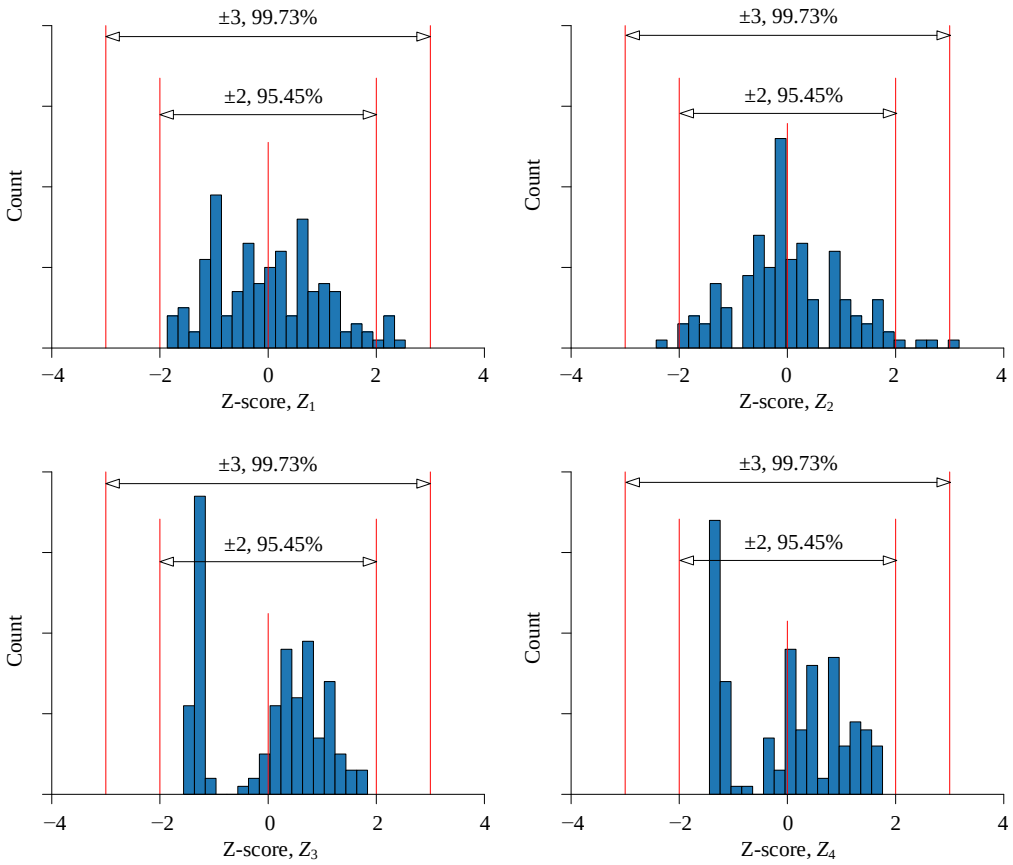


图 18. 鸢尾花 Z 分数

3.7 马氏距离和其他方法

对于二维乃至多维的情况，我们也可以使用 Z 分数。这个 Z 分数就是**马氏距离** (Mahalanobis distance)。马氏距离是一种考虑不同特征之间相关性的距离度量方法。马氏距离可以通过将样本点与数据集的均值向量进行比较，并考虑数据集的协方差矩阵来计算。与欧几里得距离不同，马氏距离可以捕捉不同特征之间的相关性和尺度差异，因此更适用于高维数据或特征相关的数据分析任务。马氏距离常用于聚类、分类、异常检测和模式识别等任务。

马氏距离定义如下：

$$d(\boldsymbol{x},\boldsymbol{q}) = \sqrt{(\boldsymbol{x}-\boldsymbol{q})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{q})} \tag{4}$$

其中，查询点 $\boldsymbol{q}$ 一般为数据质心，$\boldsymbol{\Sigma}$ 为样本数矩阵 $\boldsymbol{X}$ 方差协方差矩阵。

如果样本数据分布近似服从多元高斯分布，马氏距离则可以作为判定离群值的有效手段。图 19 (a) 所示为，不同的马氏距离等高线对应不同的置信区间。图 19 (b) 而所示为$\pm\sigma \sim \pm 4\sigma$ 置信区间。

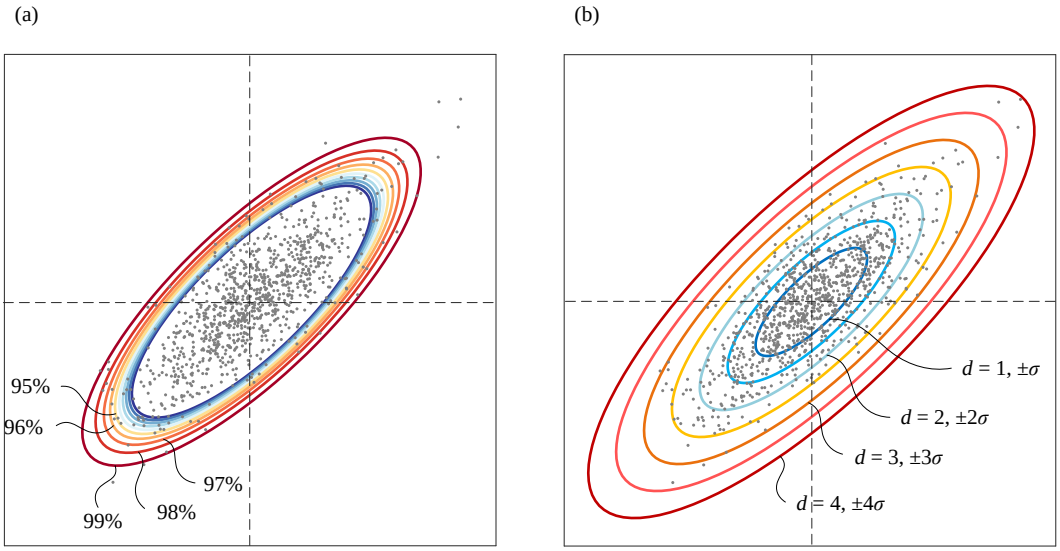


图 19. 协方差椭圆：(a) 95% ~ 99% 置信区间；(b) ±σ ~ ±4σ 置信区间

Scikit-learn 提供一个 covariance.EllipticEnvelope 对象，它就是利用马氏距离椭圆来判断离群点。图 20 所示为鸢尾花花萼长度、花萼宽度的散点图，和马氏距离为 2 的旋转椭圆。这个旋转椭圆之外的样本点可能是离群值。

有关马氏距离、卡方分布、置信区间关系，请大家参考《统计至简》第 23 章。

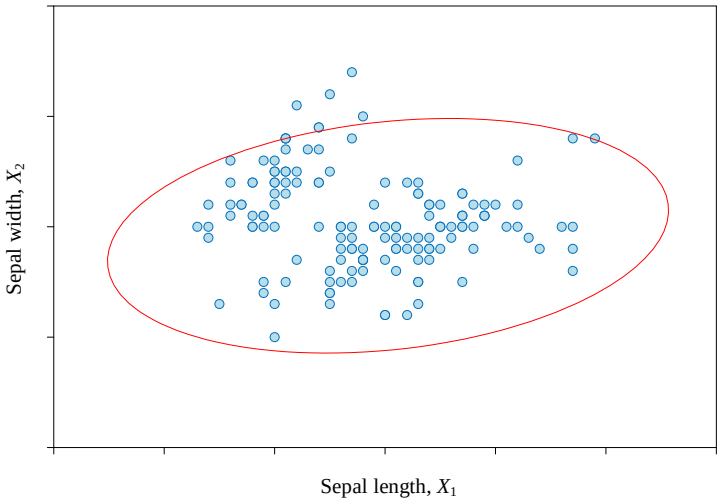


图 20. 鸢尾花数据前两个特征构造的协方差椭圆，马氏距离为 2

代码 Bk6_Ch03_01.py 绘制本章前文主要图片。

概率密度估计检测离群值

马氏距离实际上假设数据服从多元正态分布。当多特征数据分布情况较大偏离多元正态分布，马氏距离就会失效。这时我们可以用概率密度估计来检测离群值。如图 21 所示，KDE 概率密度估计没有预设数据分布假设。

有关 KDE 概率密度估计，大家可以回顾《统计至简》第 18 章。

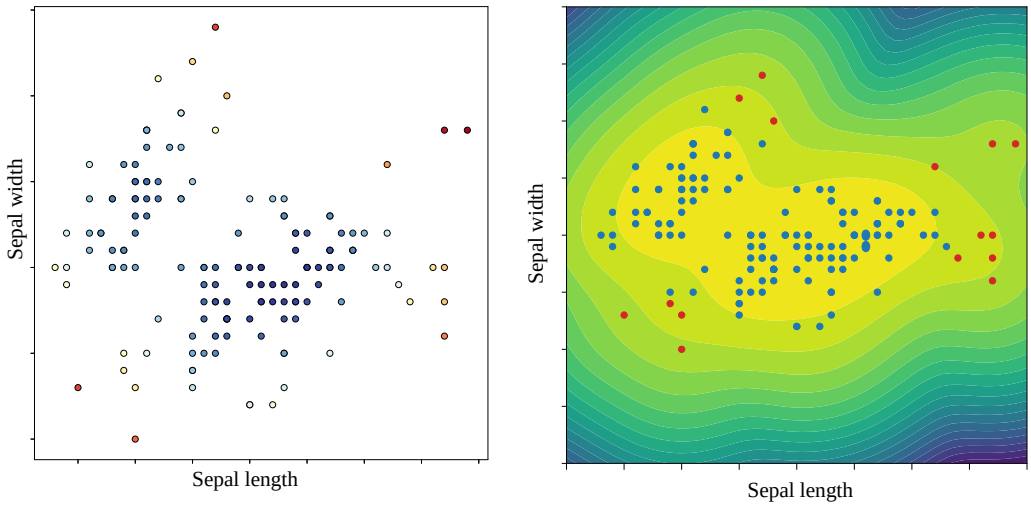


图 21. 概率密度估计判断离群值，左图散点颜色对应数据 KDE 概率密度估算值

机器学习方法

机器学习中很多算法都可以用来判断离群值。图 22 所示为用支持向量机和孤立森林算法判断鸢尾花数据中可能存在的离群值。

更多机器学习算法，请大家参考《机器学习》一书。

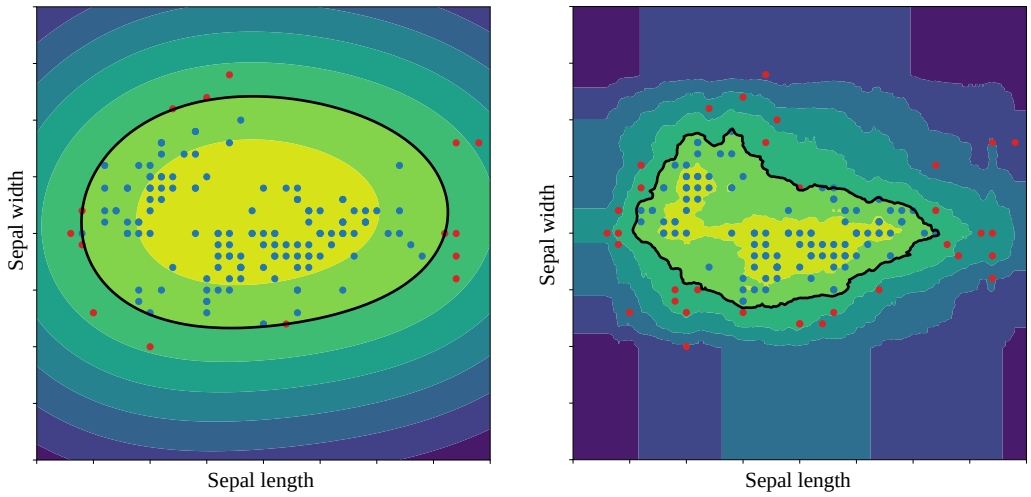


图 22. 支持向量机和孤立森林算法判定离群值

Bk6_Ch03_02.py 绘制图 21 和图 22。

离群值指的是数据集中与其他值相差较远的异常值。离群值可能会对数据分析结果产生较大的影响，导致模型不准确或偏差。离群值的产生原因包括测量误差、数据录入错误、采集异常、样本选择偏差等。

解决方法包括删除离群值、修正离群值、分别分析离群值等。注意事项包括要对数据进行探索性分析，了解数据分布和异常值的特点，合理处理离群值，避免对分析结果造成负面影响。同时，在进行离群值处理时需要谨慎，避免过度修正，影响数据的真实性和可靠性。

Scikit-learn 中有更多利用机器学习方法检测离群值的方法，请参考下例。

https://scikit-learn.org/stable/modules/outlier_detection.html

建议大家学完丛书《机器学习》一册内容，再回过头来自学这几个例子。

4 Data Transformations
数据转换

代数和统计方法处理数据，以便后续回归、分类或聚类

没有数据，就得出结论，这是大错特错。

It is a capital mistake to theorize before one has data.

—— 阿瑟·柯南·道尔 (Arthur Conan Doyle) | 英国小说作家、医生 | 1859 ~ 1930

- `numpy.random.exponential()` 产生满足指数分布随机数
- `pandas.plotting.parallel_coordinates()` 绘制平行坐标图
- `scipy.stats.boxcox()` Box-Cox 数据转换
- `scipy.stats.probplot()` 绘制 QQ 图
- `scipy.stats.yeojohnson()` Yeo-Johnson 数据转换
- `seaborn.distplot()` 绘制概率直方图
- `seaborn.heatmap()` 绘制热图
- `seaborn.jointplot()` 绘制联合分布和边际分布
- `seaborn.kdeplot()` 绘制 KDE 核概率密度估计曲线
- `seaborn.violinplot()` 绘制数据小提琴图
- `sklearn.preprocessing.MinMaxScaler()` 归一化数据
- `sklearn.preprocessing.PowerTransformer()` 广义幂变换
- `sklearn.preprocessing.StandardScaler()` 标准化数据

4.1 数据转换

本章介绍**数据转换** (data transformation) 的常见方法。数据转换是数据预处理的重要一环，用来转换要分析的数据集，使其更方便后续建模，比如回归分析、分类、聚类、降维。注意，数据预处理时，一般先处理缺失值、离群值，然后再数据转换。

数据转换的外延可以很广。函数 (比如指数函数、对数函数)、中心化、标准化、概率密度估计、插值、回归分析、主成分分析、时间序列分析、平滑降噪等，某种意义上都可以看做是数据转换。比如，经过主成分分析处理过的数据可以成为其他算法的输入。

图 1 总结本章要介绍的几种主要数据转换方法。下一章专门介绍插值。图 1 可以用作本章思维导图。

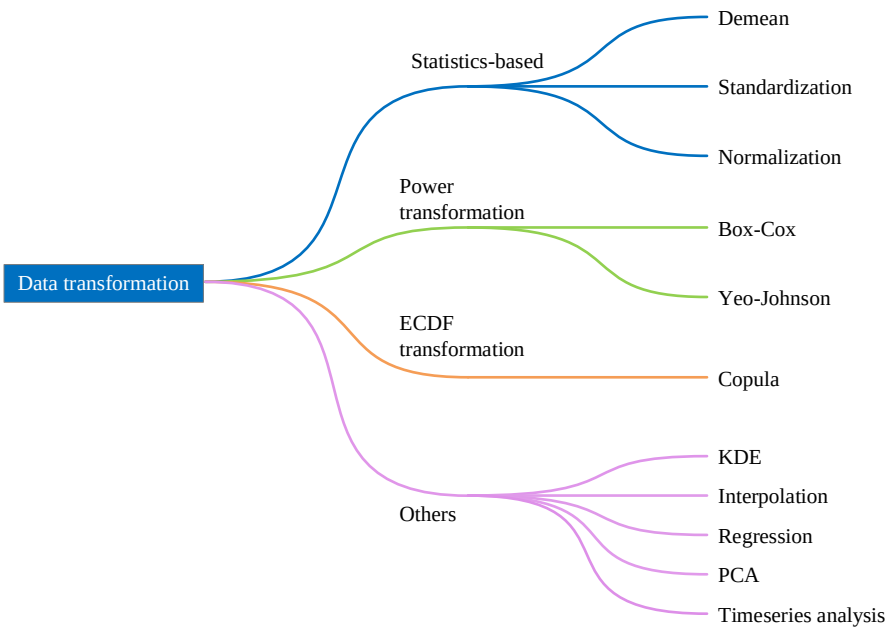


图 1. 常见数据转换方法

4.2 中心化：去均值

数据**中心化** (centralize, demean)，也叫去均值，是基于统计最基本的数据转换。

对于一个给定特征，**去均值数据** (demeaned data, centered data) 的定义为：

$$Y = X - \text{mean}(X) \tag{1}$$

其中，mean(X) 计算期望值或均值。

一般情况，多特征数据每一列数据代表一个特征。多特征数据的中心化，相当于每一列数据分别去均值。对于均值几乎为 0 的数据，去均值处理效果肯定不明显。

原始数据

本节用四种可视化方案展示数据，它们分别是热图、KDE 分布、小提琴图和平行坐标图。图 2 ~ 图 5 所示为这四种可视化方案展示的鸢尾花原始四个特征数据。

相信丛书读者对前三种可视化方案应该很熟悉。这里简单介绍图 5 所示**平行坐标图** (parallel coordinate plot)。

一个正交坐标系可以用来展示二维或三维数据，但是对于高维多元数据，正交坐标系则显得无力。而平行坐标图，可以用来可视化多特征数据。平行坐标图采用多条平行且等间距的轴，以折线形式呈现数据。图 5 还用不用颜色折线代表分类标签。

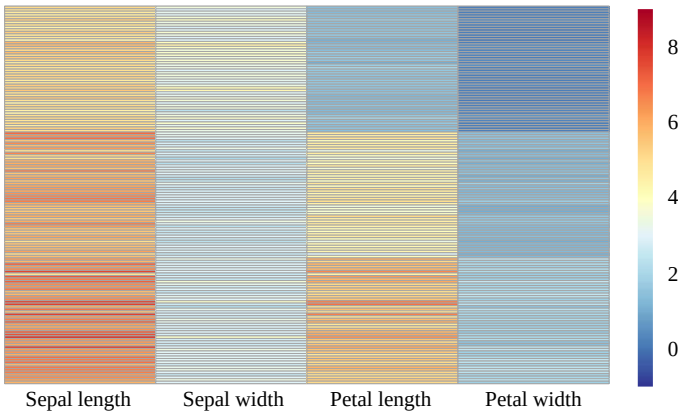


图 2. 鸢尾花数据，原始数据矩阵 $\boldsymbol{X}$

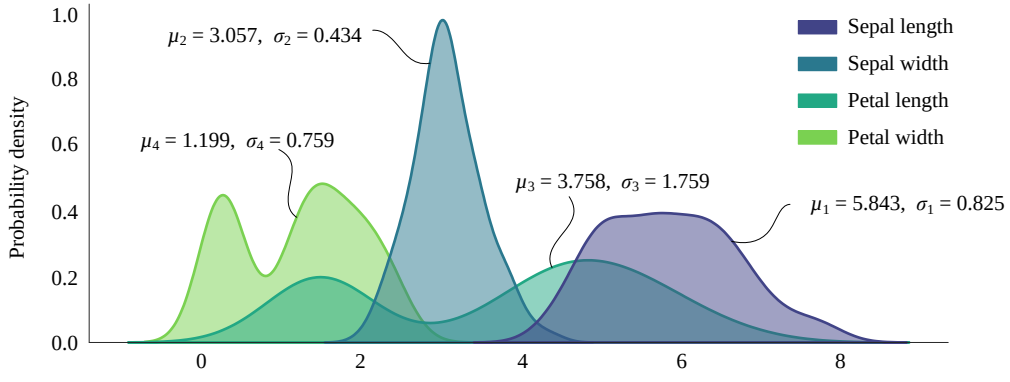


图 3. 鸢尾花数据四个特征上分布，KDE 估计

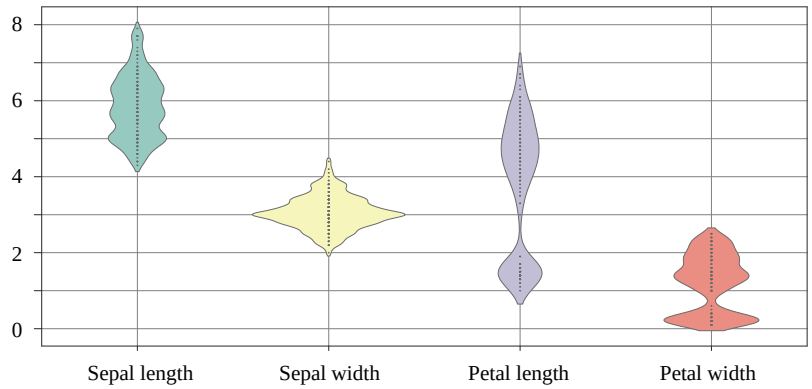


图 4. 鸢尾花原始数据，小提琴图

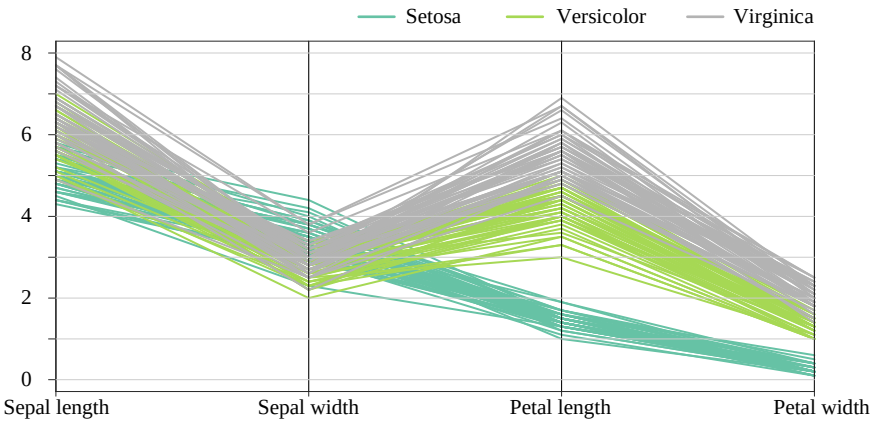


图 5. 鸢尾花数据，平行坐标图

中心化数据

图 6 ~ 图 9 则用这四种可视化方案展示去均值后鸢尾花数据。

《矩阵力量》介绍过，对于多特征数据，去均值相当于将数据质心移动到 $\boldsymbol{0}$，但是对数据分布的离散度没有任何影响。

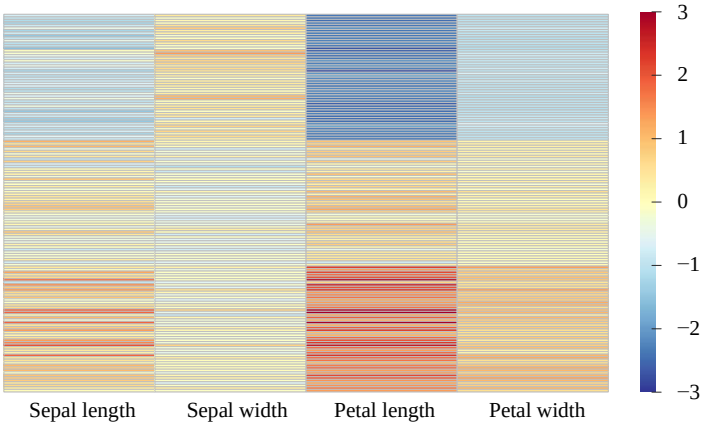


图 6. 数据热图，去均值

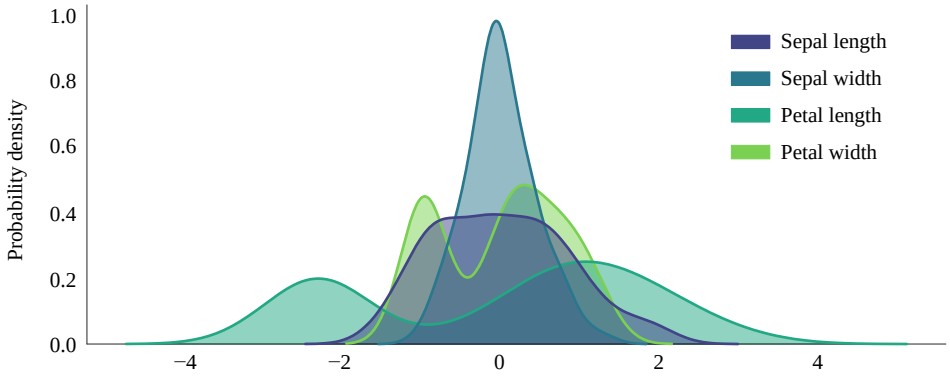


图 7. 数据 KDE 分布估计，去均值

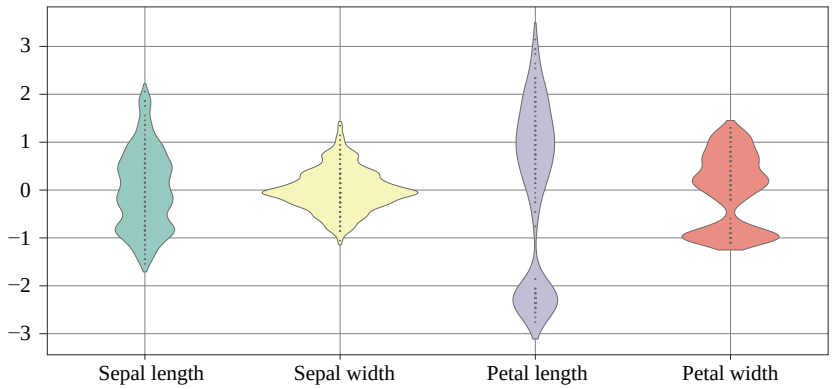


图 8. 小提琴图，去均值

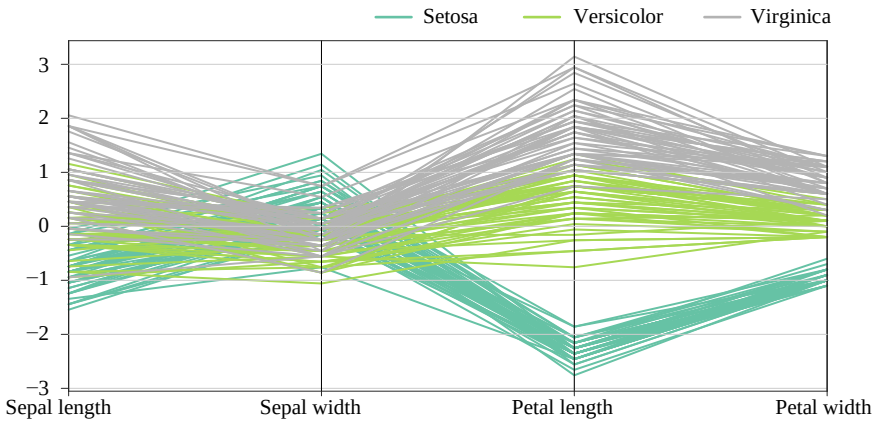


图 9. 平行坐标图，去均值

4.3 标准化：Z 分数

标准化 (standardization) 对原始数据先去均值，然后再除以标准差：

$$Z = \frac{X - \text{mean}(X)}{\text{std}(X)} \tag{2}$$

处理得到的数值实际上是原始数据的 Z 分数，表达若干倍的标准差偏移。比如，某个数值处理后结果为 3，这代表数据距离均值 3 倍标准差偏移。

⚠ 注意，Z 分数的正负代表偏离均值的方向。

在机器学习中，standardization 和 normalization 通常分别翻译为标准化和归一化。这两种预处理方法的主要区别在于对数据的缩放方式不同。

标准化通常是指将数据缩放到均值为 0，标准差为 1 的标准正态分布上。标准化可以通过先减去均值，再除以标准差来实现。标准化可以使得不同特征之间的数值尺度相同，避免某些特征对模型的影响过大，从而提高模型的鲁棒性和稳定性。

归一化 (normalization) 通常是指将数据缩放到[0,1]或[-1,1]的区间上。归一化可以通过线性变换、MinMaxScaler 等方法来实现。归一化可以使得不同特征的权重相同，避免某些特征对模型的影响过大，从而提高模型的准确性和泛化能力。

⚠ 很多文献混用 standardization 和 normalization，大家注意区分。

图 10、图 11 和图 12 分别展示的是经过标准化处理的鸢尾花数据的热图、KDE 分布曲线和平行坐标图。

《统计至简》一册讲过，**主成分分析** (Principal Component Analysis, PCA) 之前，一般会先对数据进行标准化。经过标准化后的数据，再求协方差矩阵，得到的实际上是原始数据的相关性系数矩阵。

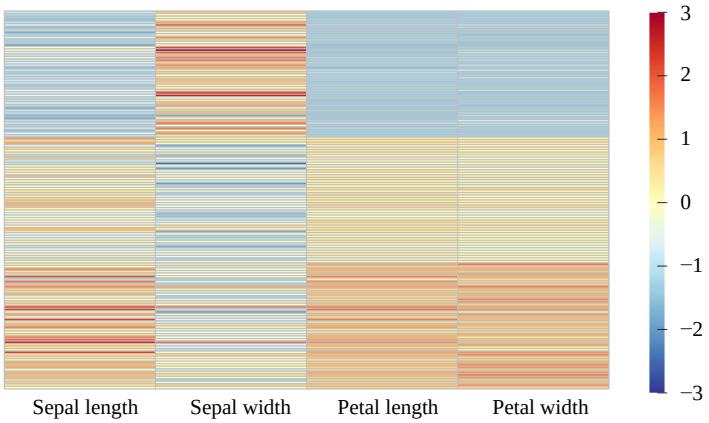


图 10. 热图，标准化

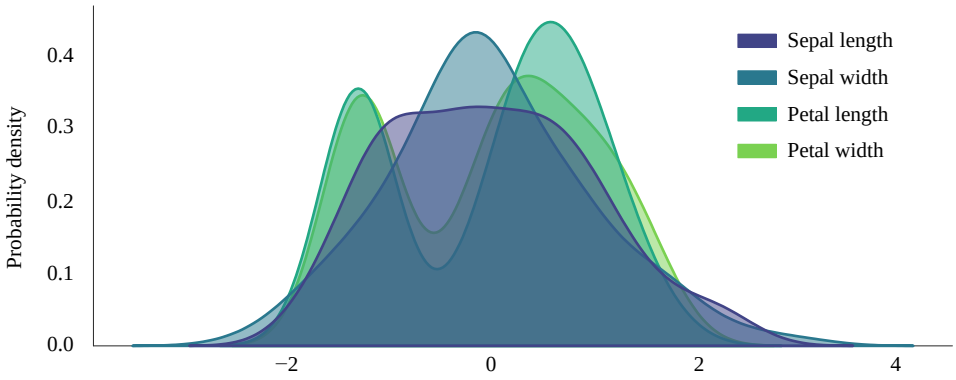


图 11. KDE 分布估计，标准化

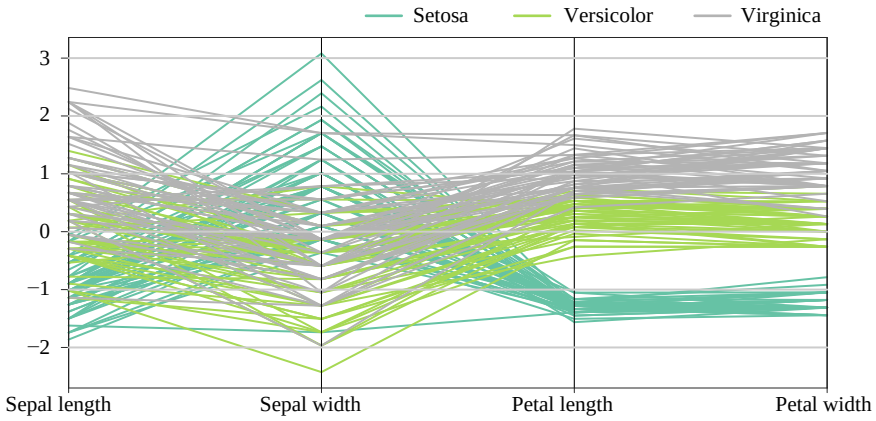


图 12. 平行坐标图，标准化

4.4 归一化：取值在 0 和 1 之间

归一化 (normalization) 常指数据首先减去其最小值，然后再除以 range(X)，即 max(X) – min(X)：

$$\frac{X-\min(X)}{\max(X)-\min(X)} \tag{3}$$

通过上式归一化得到的数据取值范围在 [0, 1] 之间。

图 13、图 14 分别展示归一化鸢尾花数据的小提琴图和平行坐标图。

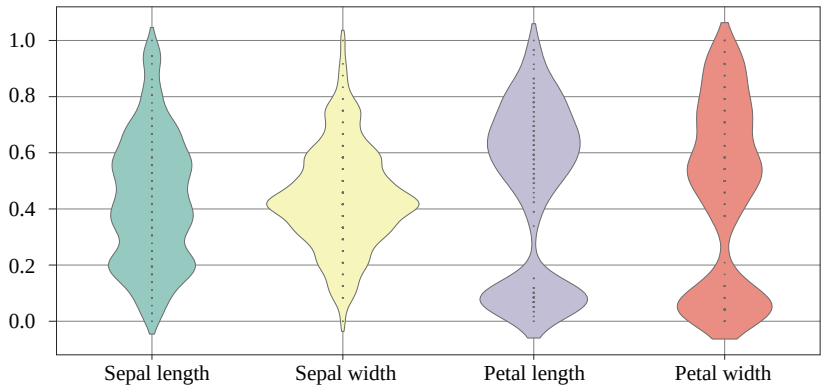


图 13. 小提琴图，归一化

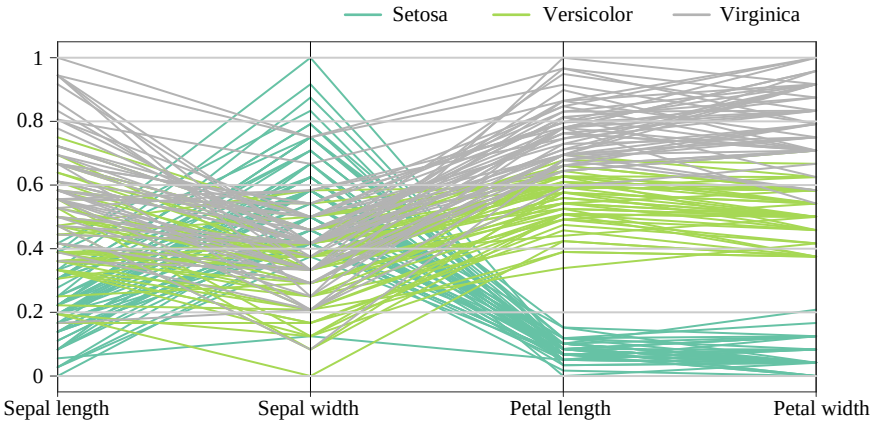


图 14. 平行坐标图，归一化

其他转换

另外一种类似归一化的数据转换方式，数据先去均值，然后再除以 range(X)：

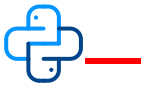
$$\tilde{x} = \frac{x - \text{mean}(X)}{\max(X) - \min(X)} \tag{4}$$

这种数据处理的特点是，处理得到的数据取值范围约在 [−0.5, 0.5] 之间。

还有一种数据转换使用箱型图的**四分位间距** (interquartile range) 作为分母，来缩放数据：

$$\frac{X - \text{mean}(X)}{IQR(X)} \tag{5}$$

其中 $IQR = Q_3 - Q_1$。



Bk6_Ch04_01.py 绘制本章之前几乎所有图像。

4.5 广义幂转换

广义幂转换 (power transform)，也称 Box-Cox，是一种用于对非正态分布数据进行转换的方法。Box-Cox 转换通过一系列参数 λ 的取值，将数据的概率密度函数进行幂函数变换，使得变换后的数据更加接近正态分布。

Box-Cox 转换可以通过最大似然估计或数据探索的方式来确定最优的λ值。Box-Cox 转换可以帮助我们改善非正态分布数据的统计性质，如方差齐性、线性关系和偏度等，从而提高模型的准确性和稳定性。Box-Cox 转换广泛应用于回归分析、时间序列分析、贝叶斯分析等领域。

Box-Cox 转换具体为：

$$x^{(\lambda)} = \begin{cases} \dfrac{x^{\lambda} - 1}{\lambda} & \lambda \neq 0 \\ \ln x & \lambda = 0 \end{cases} \tag{6}$$

其中，x 为原始数据，$x^{(\lambda)}$ 代表经过 Box-Cox 转换后的新数据，λ 为转换参数。

⚠ 注意，Box-Cox 转换要求参与转换的数据为正数。

在进行 Box-Cox 转换之前，需要确保数据都是正数。如果数据包含负数或零，可以先对数据进行平移或加上一个较小的正数，使得数据都变成正数，然后再进行 Box-Cox 转换。另外，如果数据中存在较小的负数或零，也可以考虑使用其他的转换方法，如 Yeo-Johnson 转换，它可以处理包含负数的数据。

实际上，Box-Cox 转换代表一系列转换。其中，$\lambda = 0.5$ 时，叫平方根转换；$\lambda = 0$ 时，叫对数转换；$\lambda = -1$ 时，为倒数转换。大家观察上式可以发现，它无非就是两个单调递增函数。

Box-Cox 转换通过优化 λ 参数，让转换得到的新数据明显地展现出**正态性** (normality)。

正态性指的是一个随机变量服从高斯分布的特性。正态分布是一种常见的概率分布，其概率密度函数呈钟形曲线，具有单峰性、对称性和连续性。如果一个数据集或随机变量的分布近似于正态分布，那么它就具有正态性，也称为正态分布性。正态性在统计分析中非常重要，因为很多经典的统计方法，如 t 检验、方差分析等，都基于正态分布的假设。如果数据不服从正态分布，可能会影响到模型的可靠性和精度，需要采取相应的数据预处理或选择适当的非参数方法。

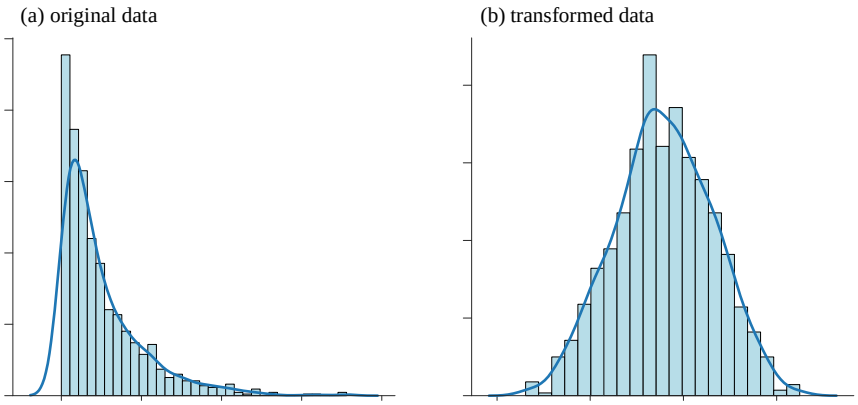


图 15. 原始数据和转换数据的直方图

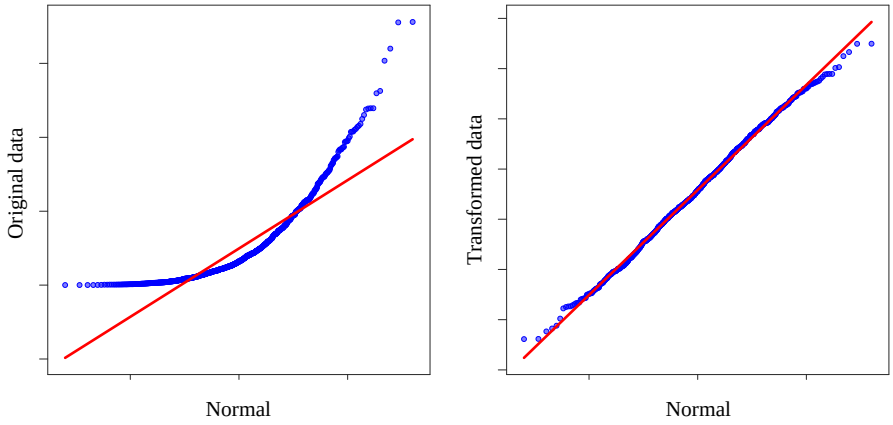


图 16. 原始数据和转换数据的 QQ 图

Yeo-Johnson 转换

前文提过 Yeo-Johnson 可以处理负值，具体数学工具为：

$$x^{(\lambda)}=\begin{cases}\dfrac{(x+1)^{\lambda}-1}{\lambda} & \lambda\neq 0, x\geq 0\\ \ln(x+1) & \lambda=0, x\geq 0\\ \dfrac{-\left((-x+1)^{2-\lambda}-1\right)}{2-\lambda} & \lambda\neq 2, x<0\\ -\ln(-x+1) & \lambda=2, x<0\end{cases} \tag{7}$$

Bk6_Ch04_02.py 绘制图 15 和图 16。sklearn.preprocessing.PowerTransformer() 函数同时支持‘yeo-johnson’和‘box-cox’ 两种方法。

4.6 经验累积分布函数

《统计至简》第 9 章一册提到，经验累积分布函数 (Empirical Cumulative Distribution Function, ECDF) 实际上也是一种重要的数据转换函数。ECDF 是一种非参数的数据转换方法。

ECDF 的特点是简单易懂，不需要对数据进行任何假设或参数估计，适用于任何类型的数据分布，包括连续型和离散型数据。通过将原始数据转换为概率分布函数，可以更好地理解数据的分布情况，并与理论分布进行比较，从而判断数据是否符合某种分布模型。

图 17 所示为样本数据和其经验累积分布的关系。

如图 18 所示，u = ECDF(x) 代表经验累积分布函数；其中，x 为原始样本数值，u 为其 ECDF 值。u 的取值范围为 [0, 1]。u = ECDF(x)具有单调递增特性。

u = ECDF(x) 对应 Scikit-learn 中的 sklearn.preprocessing.QuantileTransformer() 函数。

图 19 所示为鸢尾花数据四个特征的 ECDF 图像。

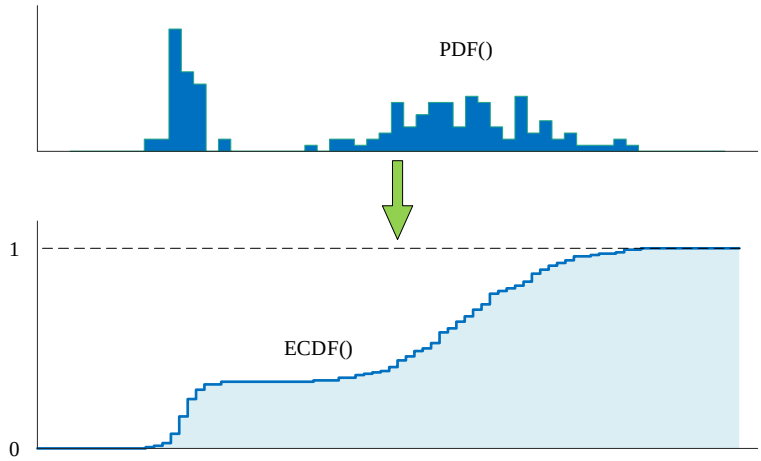


图 17. ECDF 函数转换样本数据

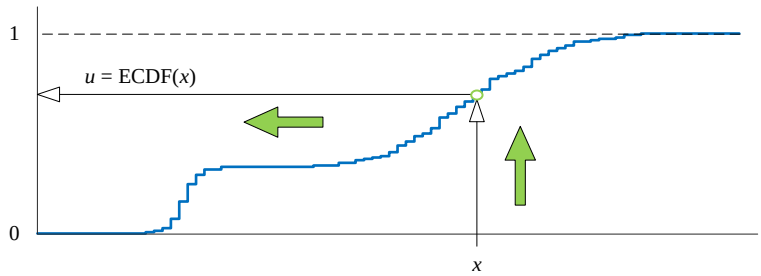


图 18. ECDF 函数原理

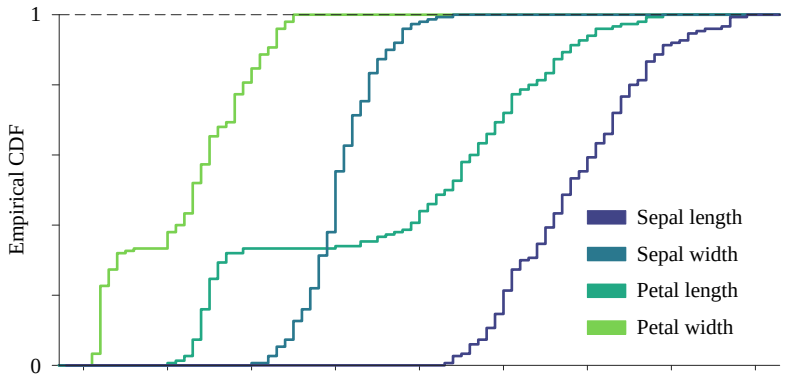


图 19. 鸢尾花数据四个特征的 ECDF

散点图

如图 19 所示，经过 ECDF 转换，鸢尾花四个特征的样本数据都变成了 [0, 1] 区间的数据。这组数据肯定也有自己的分布特点。

图 20 所示为花萼长度、花萼宽度 ECDF 散点图和概率密度等高线。

图 21 所示为鸢尾花数据 ECDF 的成对特征图。

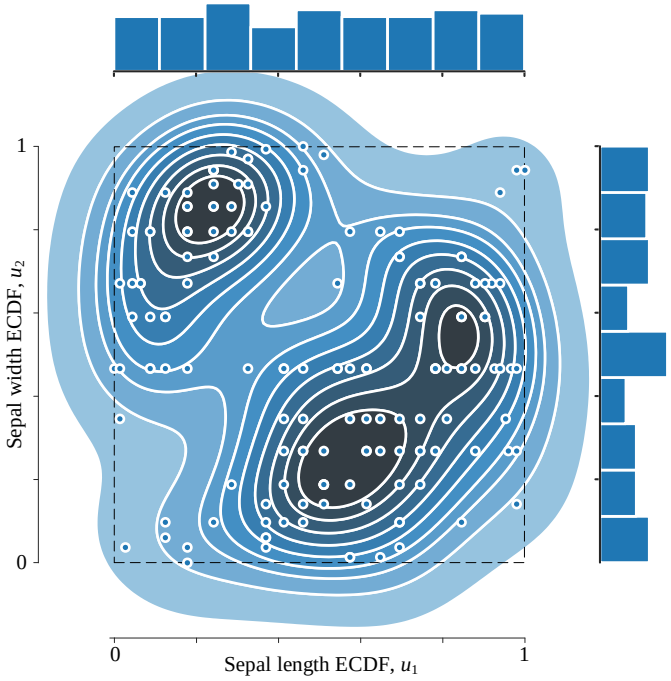


图 20. 鸢尾花花萼长度、花萼宽度 ECDF 散点图

容易发现 parametric (theoretical) CDF 和 empirical CDF 的取值范围都是 [0, 1]，而且是一一对应关系，这就是我们反复提到过的，CDF 曲线是很好的映射函数，可以将任意取值范围的数值映射到 (0, 1) 区间，而且得到的具体数值有明确的含义，即累积概率值，可以解释。

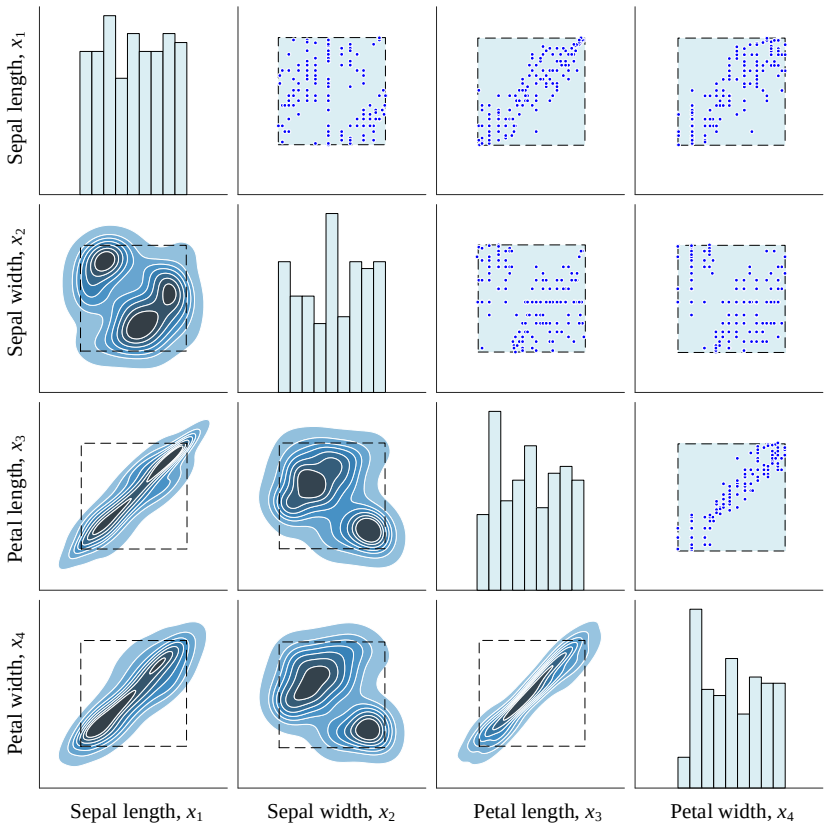


图 21. 鸢尾花数据 ECDF 的成对特征图

Bk6_Ch04_03.py 绘制图 20 和图 21。

连接函数

大家肯定会问，有没有一种分布可以描述图 20、图 21 所示概率分布？答案是肯定的！

这就是**连接函数** (copula)。连接函数是一种描述**协同运动** (co-movement) 的方法。定义向量：

$$\begin{bmatrix} x_1 & x_2 & \cdots & x_D \end{bmatrix} \tag{8}$$

它们各自的边缘经验累积概率分布值可以构成如下向量：

$$\begin{bmatrix} u_1 & u_2 & \cdots & u_D \end{bmatrix} = \begin{bmatrix} \text{ECDF}_1(x_1) & \text{ECDF}_2(x_2) & \cdots & \text{ECDF}_D(x_D) \end{bmatrix} \tag{9}$$

其中 $u_j = \text{ECDF}_j(x_j)$ 为 X_j 的边缘累积概率分布函数，u_j 的取值范围为 [0, 1]。

图 22 所示为以二元为例展示原数据和 ECDF 的关系。

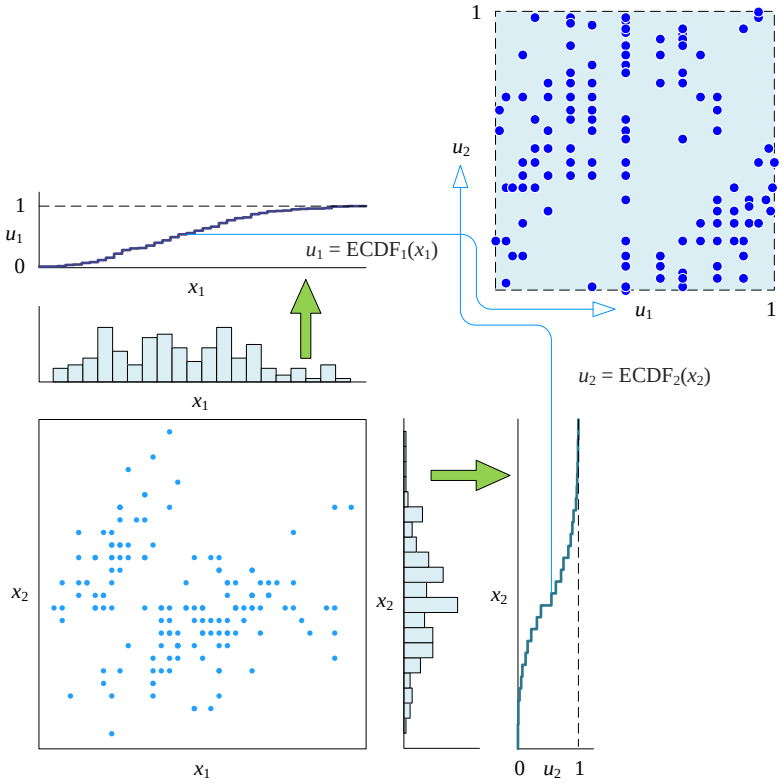


图 22. x_1 和 x_2，和 u_1 和 u_2 的关系

反方向来看 (9)：

$$\begin{bmatrix} x_1 & x_2 & \cdots & x_D \end{bmatrix} = \begin{bmatrix} \mathrm{ECDF}_1^{-1}(u_1) & \mathrm{ECDF}_2^{-1}(u_2) & \cdots & \mathrm{ECDF}_D^{-1}(u_D) \end{bmatrix} \tag{10}$$

其中，$x_j = \mathrm{ECDF}_j^{-1}(u_j)$ 为**逆累积概率分布函数** (inverse empirical cumulative distribution function)，也就是累积概率分布函数 $u_j = \mathrm{ECDF}_j(x_j)$ 的反函数。

连接函数 C 可以被定义为：

$$C(u_1, u_2, ...u_D) = \mathrm{ECDF}\left(\mathrm{ECDF}_1^{-1}(u_1), \mathrm{ECDF}_2^{-1}(u_2), ..., \mathrm{ECDF}_D^{-1}(u_D)\right) \tag{11}$$

连接函数的概率密度函数，也就是 copula PDF 可以通过下式求得：

$$c(u_1, u_2, ...u_D) = \frac{\partial^D}{\partial u_1 \cdot \partial u_2 \cdot ... \cdot \partial u_D} C(u_1, u_2, ...u_D) \tag{12}$$

图 23 展示的是几种常见连接函数，其中最常用的是**高斯连接函数** (Gaussian copula)。本书不做展开讲解，请感兴趣的读者自行学习。

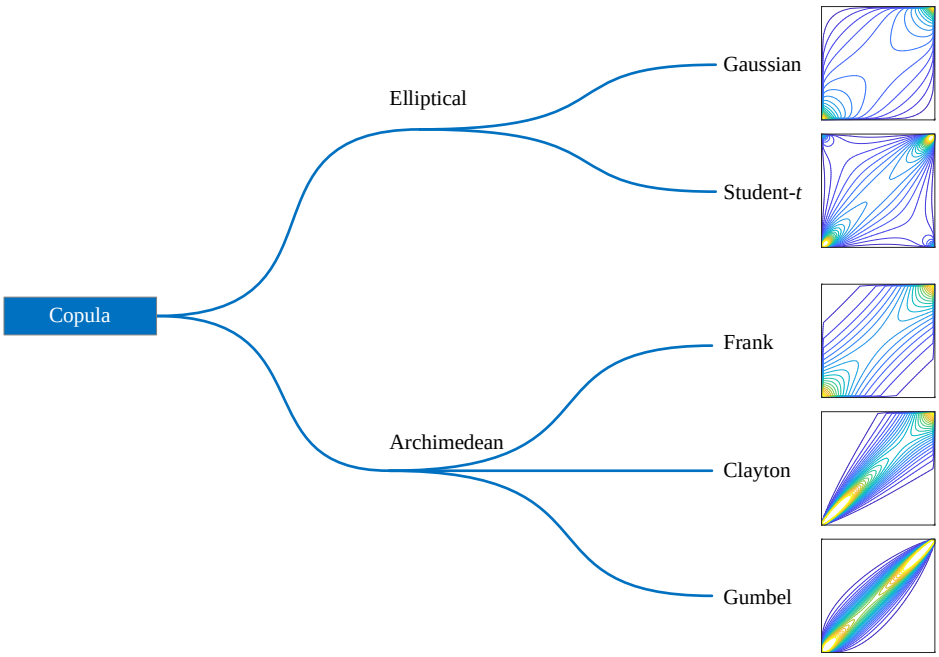


图 23. 常见连接函数

在机器学习中，数据转换是将原始数据进行处理或转换，以更好地适应模型的需求。常用的数据转换方法包括中心化、标准化、归一化、对数转换、指数转换和广义幂转换等方法。这些方法可以根据数据的分布特点、度量单位、取值范围和变量之间的关系进行选择和应用。

正确的数据转换可以提高模型的预测精度，从而提高模型的应用效果。然而，不同的数据转换方法可能对同一数据集产生不同效果，需要进行比较和评估。

如下网页专门介绍 Scikit-learn 预处理，请大家参考：

https://scikit-learn.org/stable/modules/preprocessing.html

此外，Scikit-learn 有大量的数据转换函数，请大家学习如下两例：

https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html

https://scikit-learn.org/stable/auto_examples/preprocessing/plot_map_data_to_normal.html

Statsmodels 支持连接函数，请大家参考：

https://www.statsmodels.org/dev/examples/notebooks/generated/copula.html

5 Interpolation 插值

分段插值函数，通过已知数据点

人们思考皆，浮皮潦草，泛泛而谈；现实世界却，盘根错节，千头万绪。

We think in generalities, but we live in details.

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

- `scipy.interpolate.interp1d()` 一维插值
- `scipy.interpolate.lagrange()` 拉格朗日多项式插值
- `scipy.interpolate.interp2d()` 二维插值，网格化数据
- `matplotlib.pyplot.pcolormesh()` 绘制填充颜色网格数据
- `scipy.interpolate.griddata()` 二维插值，散点化数据
- `matplotlib.pyplot.imshow()` 绘制数据平面图像

5.1 插值

插值是通过已知数据点之间的值来估计未知点的值的方法，它可以用于填补数据缺失或者进行数据平滑处理。插值方法通常基于已知数据点之间的关系，通过数学函数或者曲线拟合等方法来预测未知数据点的值。

如图 1 所示的蓝色点为已知数据点，插值就是根据这几个离散的数据点估算其他点对应的 y 值。

插值可分为**内插** (interpolation) 和**外插** (extrapolation)。内插是在已知数据点之间进行插值，估计出未知点的值。而外插则是在已知数据点的范围之外进行插值，从而预测超出已知数据点范围的未知点的值。在进行外插时，需要考虑插值函数是否能够正确地拟合未知数据点，并且需要注意不要过度依赖插值函数来进行预测，以免导致不可靠的预测结果。

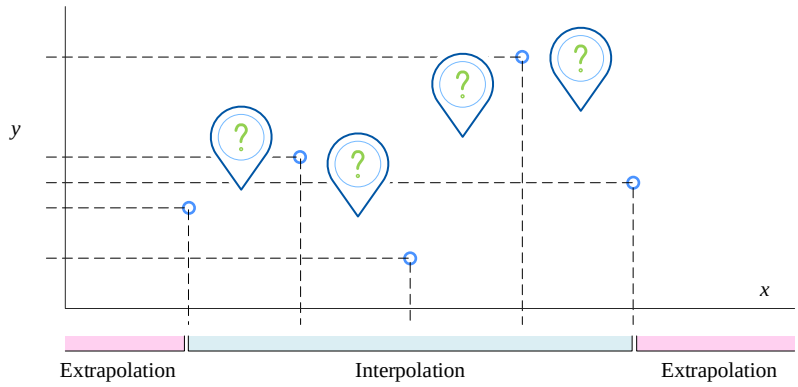


图 1. 插值的意义

常见插值方法

图 2 总结常用的插值的算法。图 2 相当于本章的思维导图。

本章主要介绍如下几种方法：

- **常数插值** (constant interpolation)，比如**向前** (previous 或 forward)、**向后** (next 或 backward)、**最邻近** (nearest)；
- **线性插值** (linear interpolation)；
- **二次插值** (quadratic interpolation)，本章不做介绍；
- **三次插值** (cubic interpolation)；
- **拉格朗日插值** (Lagrange polynomial interpolation)。

本章最后还要介绍**二维插值** (bivariate interpolation)，二维插值将一元插值的方法推广到二维。

此外，对于时间序列，处理缺失值或者获得颗粒度更高的数据，都可以使用插值。图 3 所示为利用线性插值插补时间序列数据中的缺失值。

《可视之美》介绍的贝塞尔曲线本质上也是插值。贝塞尔曲线是一种通过一系列控制点来定义曲线形状的数学函数。在计算机图形学和计算机辅助设计中，常使用贝塞尔曲线来生成平滑的曲线形状。

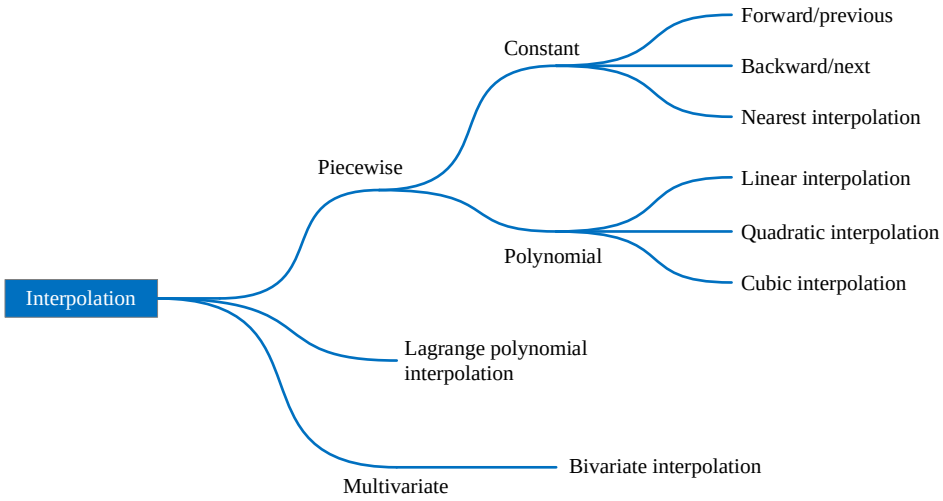


图 2. 插值的分类

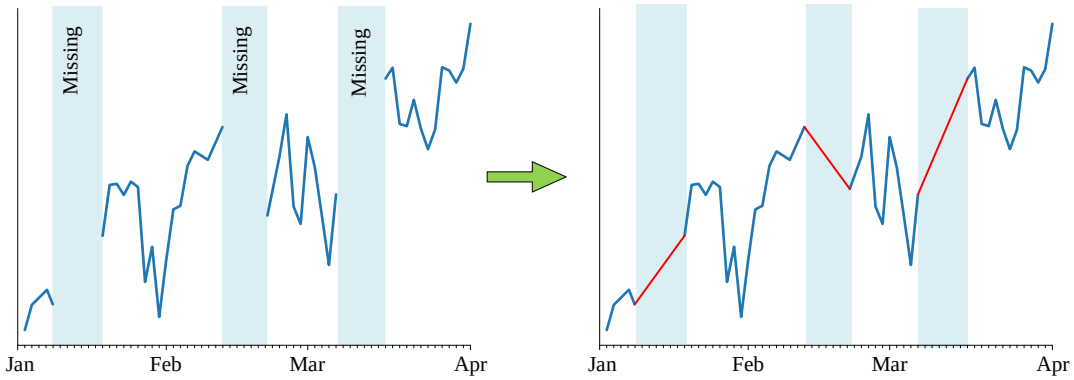


图 3. 时间序列插值

分段函数

虽然，一些插值分段函数构造得到的曲线整体看上去平滑。但是绝大多数情况，插值函数是分段函数，因此插值也称**分段插值** (piecewise interpolation)。

《数学要素》第 11 章介绍过分段函数。

对于一元函数 $f(x)$，分段函数是指自变量 x 在不同取值范围对应不同解析式的函数。

每两个相邻的数据点之间便对应不同解析式：

$$f(x)=\begin{cases} f_1(x) & x^{(1)} \le x < x^{(2)} \\ f_2(x) & x^{(2)} \le x < x^{(3)} \\ \cdots & \cdots \\ f_{n-1}(x) & x^{(n-1)} \le x < x^{(n)} \end{cases} \tag{1}$$

其中，n 为已知点个数。

注意，上式中 $f_i(x)$ 代表一个特定解析式。分段函数虽然由一系列解析式构成，但是分段函数还是一个函数。

如图 4 所示，已知数据点一共有五个——$(x^{(1)}, y^{(1)})$、$(x^{(2)}, y^{(2)})$、$(x^{(3)}, y^{(3)})$、$(x^{(4)}, y^{(4)})$、$(x^{(5)}, y^{(5)})$。比如，分段函数 $f(x)$ 在 $[x^{(1)}, x^{(2)}]$ 区间的解析式为 $f_1(x)$。$f_1(x)$ 通过 $(x^{(1)}, y^{(1)})$、$(x^{(2)}, y^{(2)})$ 两个已知数据点。图 4 实际上就是线性插值。

(1) 还告诉我们，对于内插，n 个已知点可以构成 $n-1$ 个区间，即分段函数有 $n-1$ 个解析式。

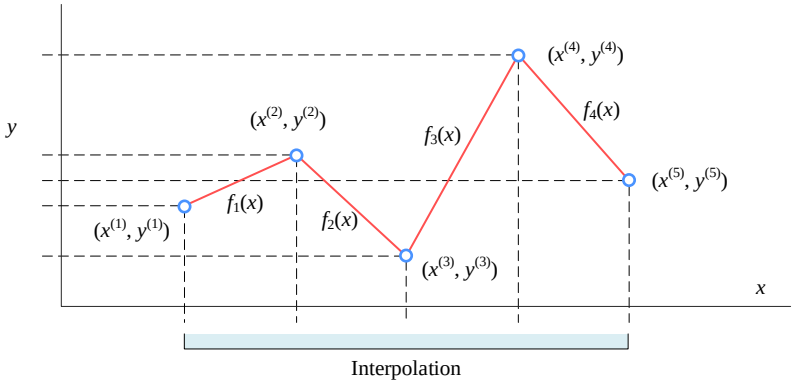


图 4. 分段函数

拟合、插值

大家经常混淆拟合和插值这两种方法。插值和拟合有一个相同之处，它们都是根据已知数据点，构造函数，从而推断得到更多数据点。

插值和回归都是用于对数据进行预测的方法，但两者有明显的区别。插值是用于填补已有数据点之间的空缺，预测未知点的值。回归则是预测自变量和因变量之间的关系。插值通常使用插值函数，如多项式插值；而回归则通过拟合数据点的回归方程来预测因变量的值。插值通常用于数据平滑处理、数据填补等。回归则常用于预测和建模。插值要求原始数据点之间要有一定的连续性和平滑性；而回归则对数据点的分布没有明显要求。插值得到的是精确的函数值，但在超出已有数据范围时可能不准确；而回归得到的是变量之间的大致关系，可以预测未来的趋势。

需要根据具体情况选择合适的方法。当数据缺失或需要平滑处理时，可以使用插值方法；当需要建立模型并预测未来趋势时，可以使用回归方法。

插值一般得到分段函数，分段函数通过所有给定的数据点，如图 5 (a)、(b) 所示。回归拟合得到的函数尽可能靠近样本数据点，如图 5 (c)、(d) 所示。图 6 比较二维插值和二维回归。

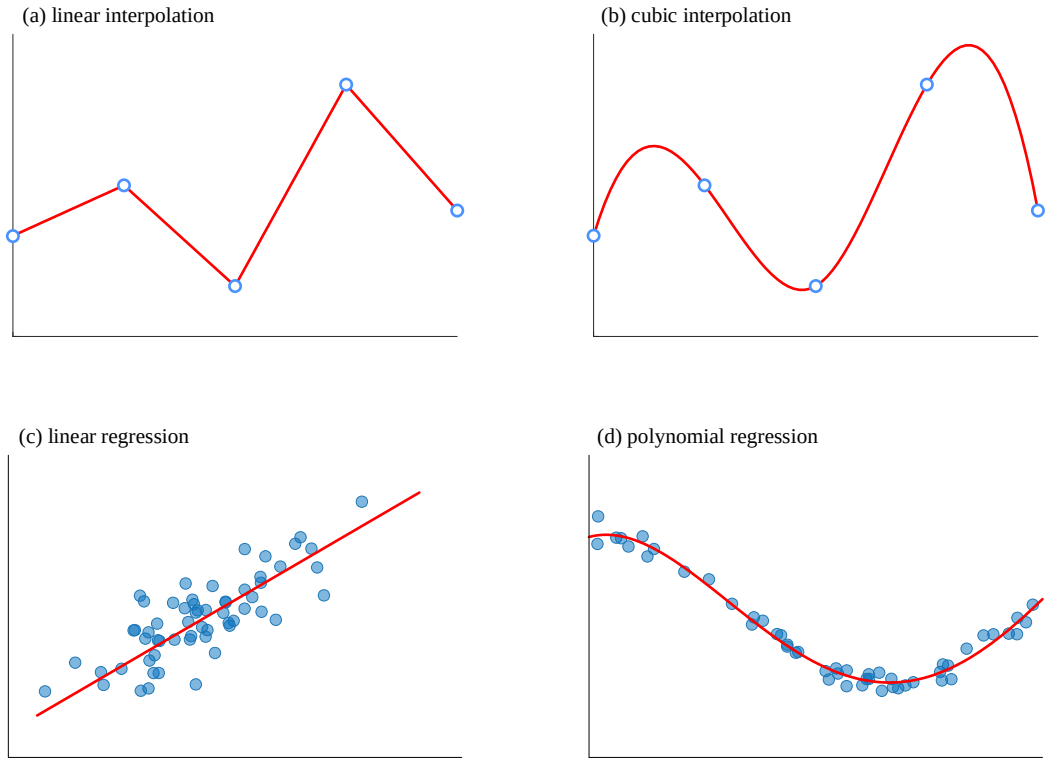


图 5. 比较一维插值和回归

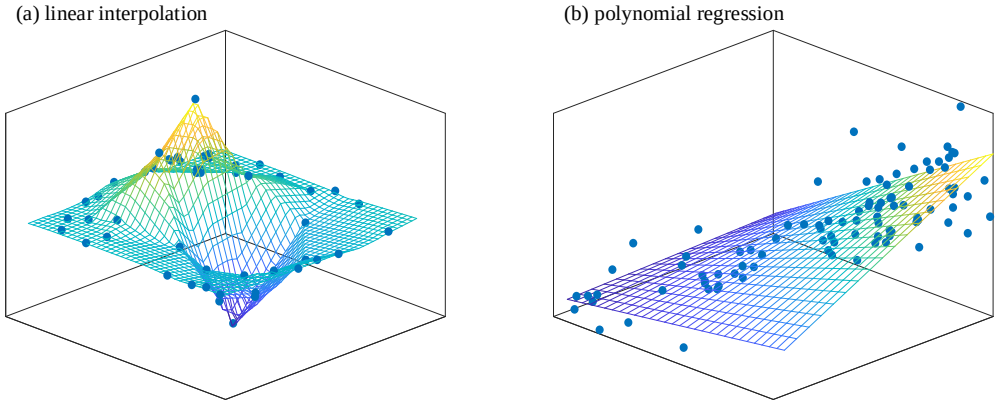


图 6. 比较二维插值和二维回归

5.2 常数插值：分段函数为阶梯状

本节介绍常用的三种常数插值方法。

向前

向前常数插值对应的分段函数为：

$$f(x)=\begin{cases} f_1(x)=x^{(1)} & x^{(1)}\le x<x^{(2)} \\ f_2(x)=x^{(2)} & x^{(2)}\le x<x^{(3)} \\ \cdots & \cdots \\ f_{n-1}(x)=x^{(n-1)} & x^{(n-1)}\le x<x^{(n)} \end{cases} \tag{2}$$

如图 7 所示，向前常数插值用区间 $[x^{(i)}, x^{(i+1)}]$ 左侧端点，即 $x^{(i)}$，对应的 $y^{(i)}$，作为常数函数的取值。图 7 中红色划线为真实函数取值。

对于数据帧 df，如果存在 NaN 的话，df.fillna(method = 'ffill') 便对应向前常数插补。

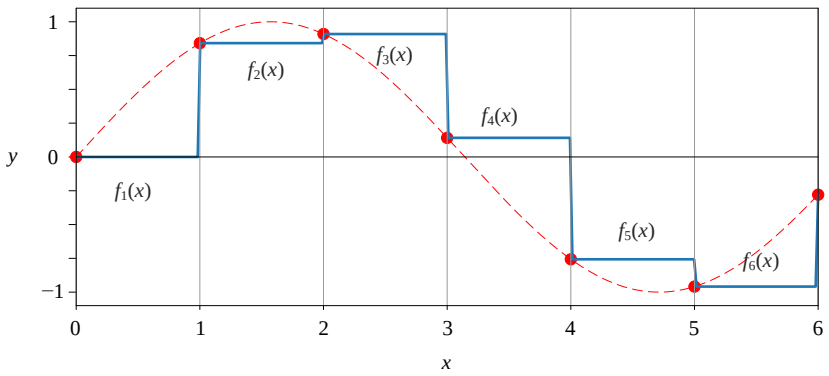


图 7. 向前常数插值

向后

向后常数插值对应的分段函数为：

$$f(x)=\begin{cases} f_1(x)=x^{(2)} & x^{(1)}\le x<x^{(2)} \\ f_2(x)=x^{(3)} & x^{(2)}\le x<x^{(3)} \\ \cdots & \cdots \\ f_{n-1}(x)=x^{(n)} & x^{(n-1)}\le x<x^{(n)} \end{cases} \tag{3}$$

如图 8 所示，向后常数插值和图 7 正好相反。

对于数据帧 df，如果存在 NaN 的话，df.fillna(method = 'bfill') 对应向后常数插补。

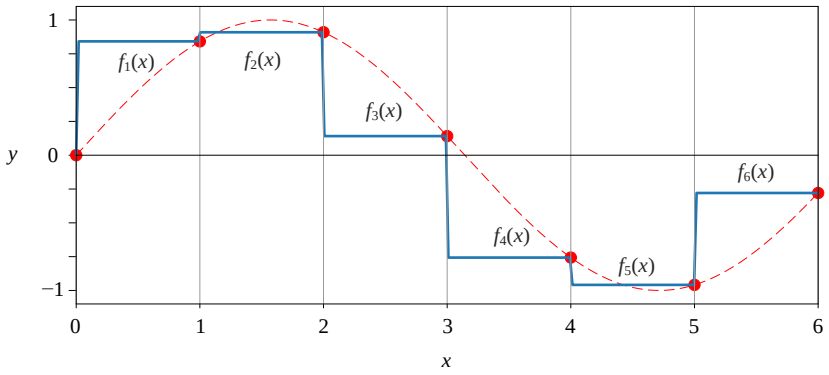


图 8. 向后常数插值

最邻近

最邻近插值的分段函数为：

$$f(x)=\begin{cases} f_1(x)=x^{(1)} & x^{(1)}\le x<\dfrac{x^{(1)}+x^{(2)}}{2} \\ f_2(x)=x^{(2)} & \dfrac{x^{(1)}+x^{(2)}}{2}\le x<\dfrac{x^{(2)}+x^{(3)}}{2} \\ \cdots & \cdots \\ f_n(x)=x^{(n)} & \dfrac{x^{(n-1)}+x^{(n)}}{2}\le x<x^{(n)} \end{cases} \tag{4}$$

如图 9 所示，最邻近常数插值相当于“向前”和“向后”常数插值的“折中”。分段插值函数同样是阶梯状，只不过阶梯发生在两个相邻已知点中间处。

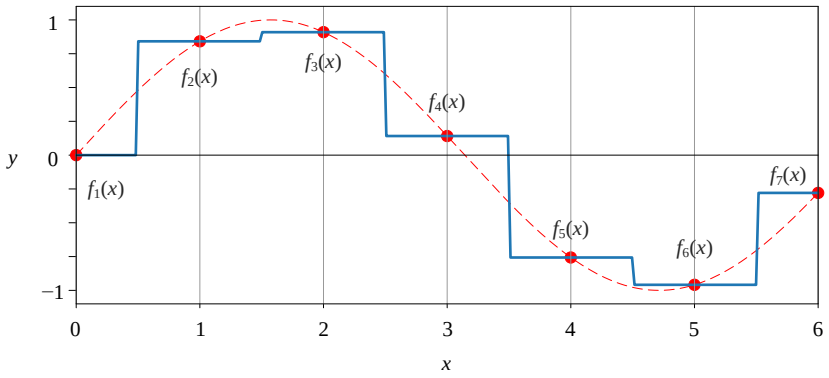


图 9. 最邻近常数插值

5.3 线性插值：分段函数为线段

对于线性插值，区间 $[x^{(i)}, x^{(i+1)}]$ 对应的 $f_i(x)$ 为：

$$f_i(x)=\underbrace{\left(\frac{y^{(i)}-y^{(i+1)}}{x^{(i)}-x^{(i+1)}}\right)}_{\text{slope}}\left(x-x^{(i+1)}\right)+y^{(i+1)} \tag{5}$$

容易发现，上式就是《数学要素》第 11 章介绍的一元函数的点斜式。

也就是说，不考虑区间的话，上式代表通过 $(x^{(i)}, y^{(i)})$、$(x^{(i+1)}, y^{(i+1)})$ 两点的一条直线。

图 10 所示为线性插值结果。白话说，线性插值就是用任意两个相邻已知点连接成的线段来估算其他未知点的值。

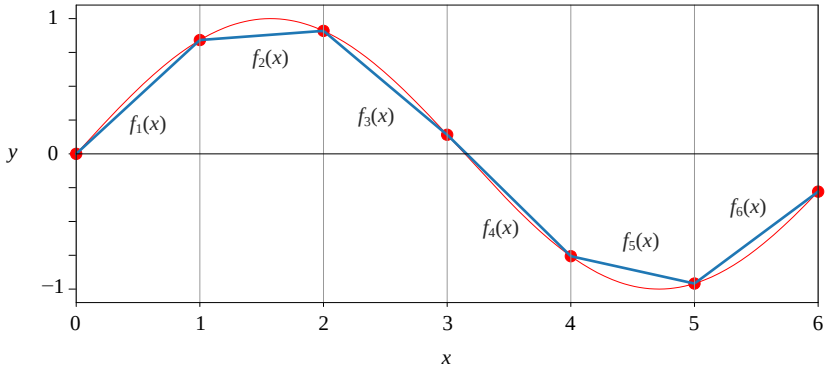


图 10. 线性插值

5.4 三次样条插值：光滑曲线拼接

图 11 所示为三次样条插值的结果。虽然，整条曲线看上去连续、光滑，实际上它是由四个函数拼接起来的分段函数。

对于三次样条插值，每一段的分段函数是三次多项式：

$$f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i \tag{6}$$

其中，a_i、b_i、c_i、d_i 为需要求解的系数。

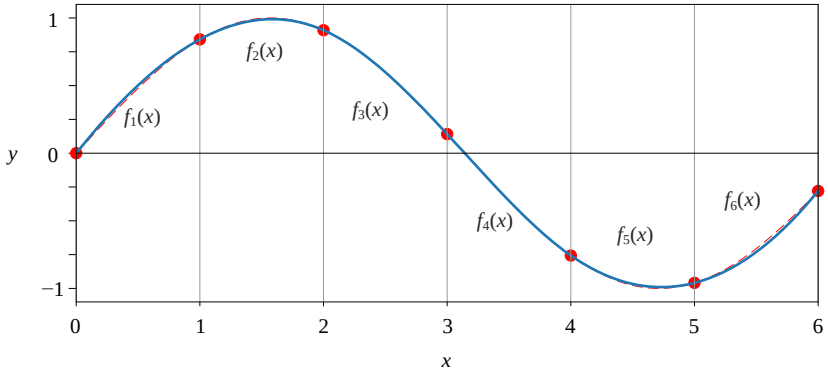


图 11. 三次样条插值

为了求解系数，我们需要构造一系列等式。类似线性插值，每一段三次函数通过区间 $[x^{(i)}, x^{(i+1)}]$ 左右两点，即：

$$\begin{cases} f_i\left(x^{(i)}\right)=y^{(i)} & i=1,2,...,n-1 \\ f_i\left(x^{(i+1)}\right)=y^{(i+1)} & i=1,2,...,n-1 \end{cases} \tag{7}$$

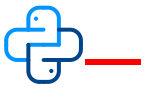
曲线之所以看起来很平滑是因为，除两端样本数据点以外，内部数据点处，一阶和二阶导数等值：

$$\begin{cases} f_i'\left(x^{(i+1)}\right)=f_{i+1}'\left(x^{(i+1)}\right) & i=1,2,...,n-2 \\ f_i''\left(x^{(i+1)}\right)=f_{i+1}''\left(x^{(i+1)}\right) & i=1,2,...,n-2 \end{cases} \tag{8}$$

对于三次样条插值，一般还设定两端样本数据点处二阶导数为 0：

$$\begin{cases} f_1''\left(x^{(1)}\right)=0 \\ f_{n-1}''\left(x^{(n)}\right)=0 \end{cases} \tag{9}$$

插值中系数求解一般都是用矩阵运算完成。举个例子，在三次样条插值中，需要解出一个三对角线方程组，这个方程组可以用矩阵形式表示。具体来说，需要先确定每个小区间内的多项式系数，然后利用这些系数和每个小区间的边界点，构造一个三对角矩阵方程组，利用三对角矩阵求解方法，可以得到每个小区间内的多项式系数，从而得到整个分段函数。本章不展开讲解。



Bk6_Ch05_01.py 完成插值并绘制图 7 ~ 图 11。Python 进行一维插值函数为 scipy.interpolate.interp1d()，二维插值的函数为 scipy.interpolate.interp2d()。

5.5 拉格朗日插值

拉格朗日插值 (Lagrange interpolation) 不同于本章前文介绍的插值方法。前文介绍的插值方法得到的都是分段函数，而拉格朗日插值得到的是一个高次多项式函数 $f(x)$。$f(x)$ 相当是由若干多项式函数叠加而成：

$$f\left(x\right)=\sum_{i=1}^{n} f_i\left(x\right) \tag{10}$$

其中，

$$f_i\left(x\right)=y^{(i)}\cdot\prod_{k=1,k\neq i}^{n}\frac{x-x^{(k)}}{x^{(i)}-x^{(k)}} \tag{11}$$

$f_i(x)$ 展开来写：

$$f_i(x)=y^{(i)}\cdot\frac{\left(x-x^{(1)}\right)\left(x-x^{(2)}\right)\ldots\left(x-x^{(i-1)}\right)\left(x-x^{(i+1)}\right)\ldots\left(x-x^{(n)}\right)}{\left(x^{(i)}-x^{(1)}\right)\left(x^{(i)}-x^{(2)}\right)\ldots\left(x^{(i)}-x^{(i-1)}\right)\left(x^{(i)}-x^{(i+1)}\right)\ldots\left(x^{(2)}-x^{(n)}\right)} \tag{12}$$

比如，$f_1(x)$ 展开来写：

$$f_1(x)=y^{(1)}\cdot\frac{\left(x-x^{(2)}\right)\left(x-x^{(3)}\right)\ldots\left(x-x^{(n)}\right)}{\left(x^{(1)}-x^{(2)}\right)\left(x^{(1)}-x^{(3)}\right)\ldots\left(x^{(1)}-x^{(n)}\right)} \tag{13}$$

$f_2(x)$ 展开来写：

$$f_2(x)=y^{(2)}\cdot\frac{\left(x-x^{(1)}\right)\left(x-x^{(3)}\right)\ldots\left(x-x^{(n)}\right)}{\left(x^{(2)}-x^{(1)}\right)\left(x^{(2)}-x^{(3)}\right)\ldots\left(x^{(2)}-x^{(n)}\right)} \tag{14}$$

举个例子

比如，$n=3$，也就是有三个样本数据点 $\{(x^{(1)},y^{(1)}),(x^{(2)},y^{(2)}),(x^{(3)},y^{(3)})\}$ 的时候，$f(x)$ 为：

$$f(x)=\underbrace{y^{(1)}\cdot\frac{\left(x-x^{(2)}\right)\left(x-x^{(3)}\right)}{\left(x^{(1)}-x^{(2)}\right)\left(x^{(1)}-x^{(3)}\right)}}_{f_1(x)}+\underbrace{y^{(2)}\cdot\frac{\left(x-x^{(1)}\right)\left(x-x^{(3)}\right)}{\left(x^{(2)}-x^{(1)}\right)\left(x^{(2)}-x^{(3)}\right)}}_{f_2(x)}+\underbrace{y^{(3)}\cdot\frac{\left(x-x^{(1)}\right)\left(x-x^{(2)}\right)}{\left(x^{(3)}-x^{(1)}\right)\left(x^{(3)}-x^{(2)}\right)}}_{f_3(x)} \tag{15}$$

观察上式，$f(x)$ 相当于三个二次函数叠加得到。

将三个数据点 $\{(x^{(1)},y^{(1)}),(x^{(2)},y^{(2)}),(x^{(3)},y^{(3)})\}$，逐一代入上式，可以得到：

$$f\left(x^{(1)}\right)=y^{(1)},\quad f\left(x^{(2)}\right)=y^{(2)},\quad f\left(x^{(3)}\right)=y^{(3)} \tag{16}$$

也就是说，多项式函数 $f(x)$ 通过给定的已知点。

图 12 所示为拉格朗日插值结果。

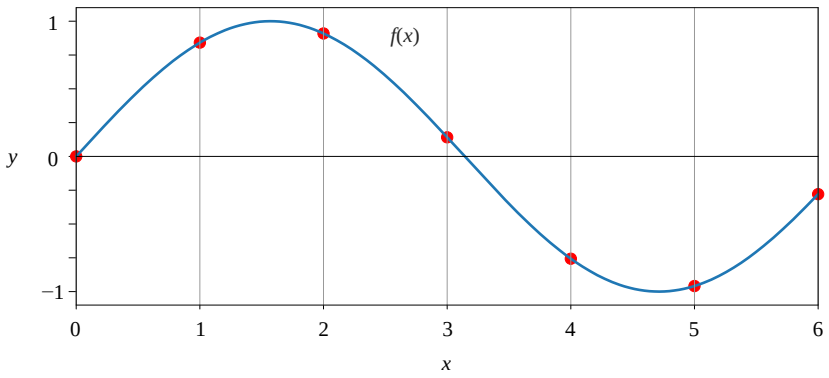


图 12. 拉格朗日插值

龙格现象

有一点需要大家注意的是，已知点数量 n 不断增大，拉格朗日插值函数多项式函数次数不断提高，插值多项式的插值逼近效果未必好。如图 13 所示，插值多项式 (红色曲线) 区间边缘处出现振荡问题，这一现象叫做**龙格现象** (Runge's phenomenon)。

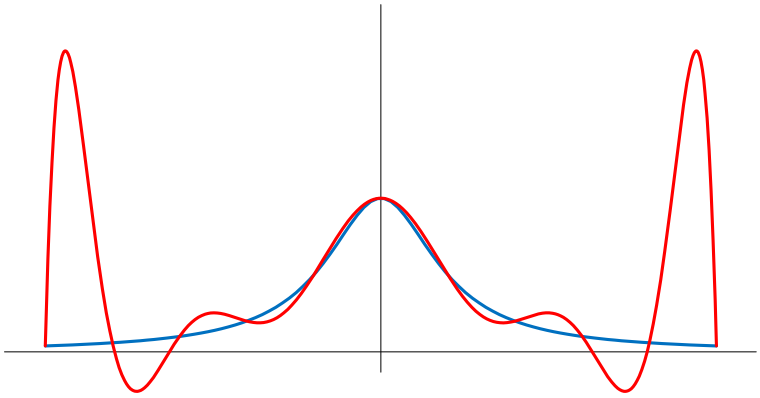


图 13. 龙格现象

Bk6_Ch05_02.py 完成拉格朗日插值，并绘制图 12。

5.6 二维插值

如图 14 所示，以二维线性插值为例，二维线性插值相当于处理了三个一维线性插值。

对于二维线性插值，先将二维坐标系中的点分别按照横坐标和纵坐标排序。然后，找到待插值点所在的四个相邻的点。分别对这四个点在横坐标和纵坐标上进行一维线性插值，得到在横向和纵向上的两个插值结果。将上述两个插值结果加权平均，作为待插值点的二维线性插值结果。其中，权重的计算基于待插值点相对于四个相邻点在横向和纵向上的距离，距离越远的点权重越小。

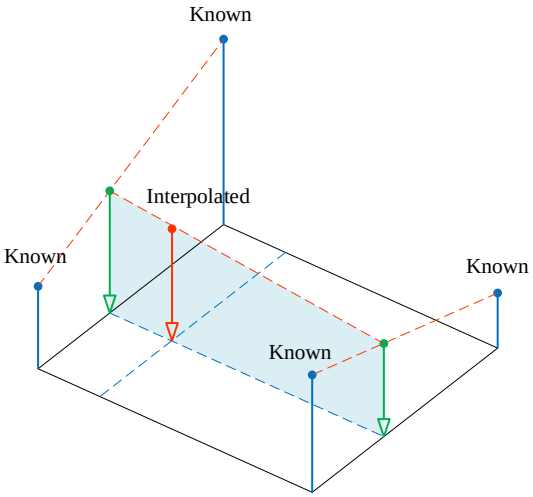


图 14. 二维线性插值原理

举个例子

图 15 中 × 为给定的已知数据。图 16 和图 17 所示为分别通过线性插值、三次样条插值完成的二维插值结果。二维插值用到的函数是 scipy.interpolate.interp2d()。

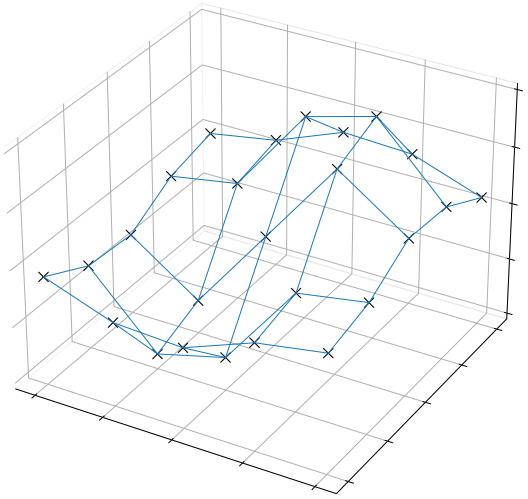


图 15. 已知数据点

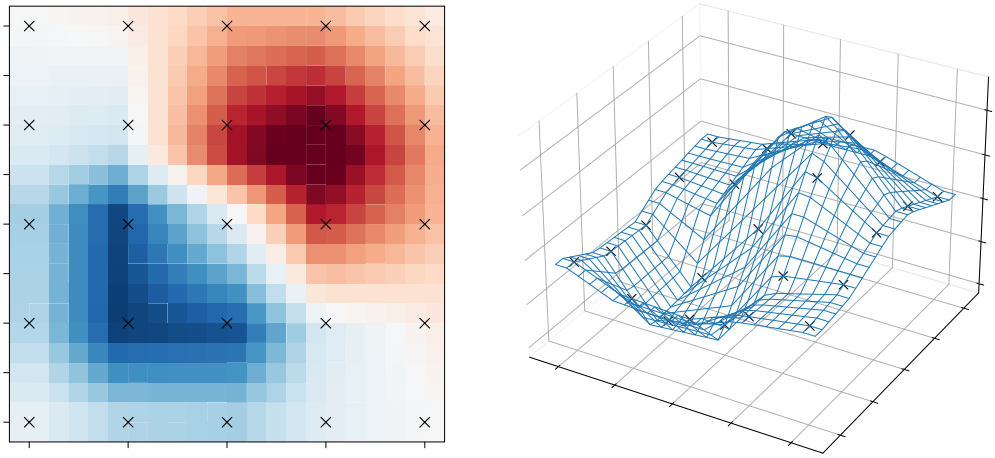


图 16. 二维插值，规则网格，线性插值

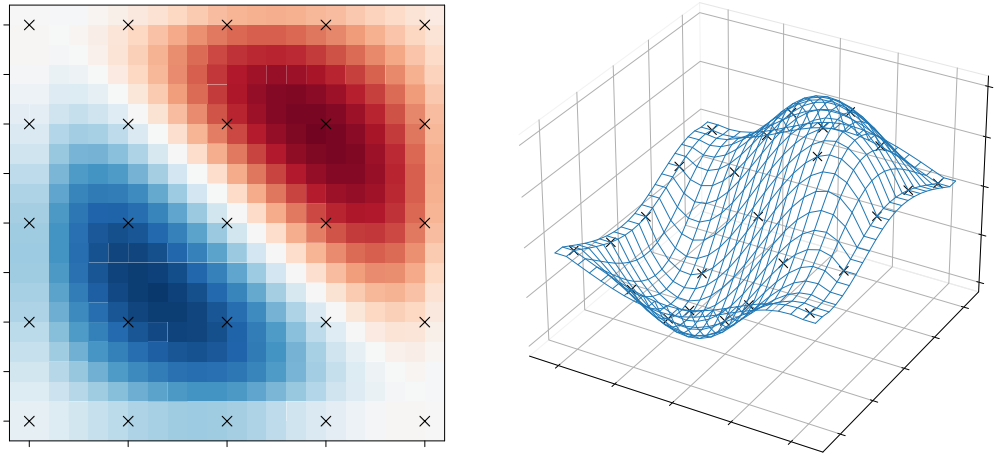


图 17. 二维插值，规则网格，三次样条

Bk6_Ch05_03.py 完成二维插值，并绘制图 16 和图 17。

不规则散点

大家可能已经注意到，图 15 给定的已知数据是规整的网格数据。当数据并不是规整的网格数据，而是不规则的散点时，我们也可以用 scipy.interpolate.griddata() 完成二维插值。图 18、图 19、图 20 分别所示为利用最邻近、线性、三次样条方法完成不规则散点的二维插值。

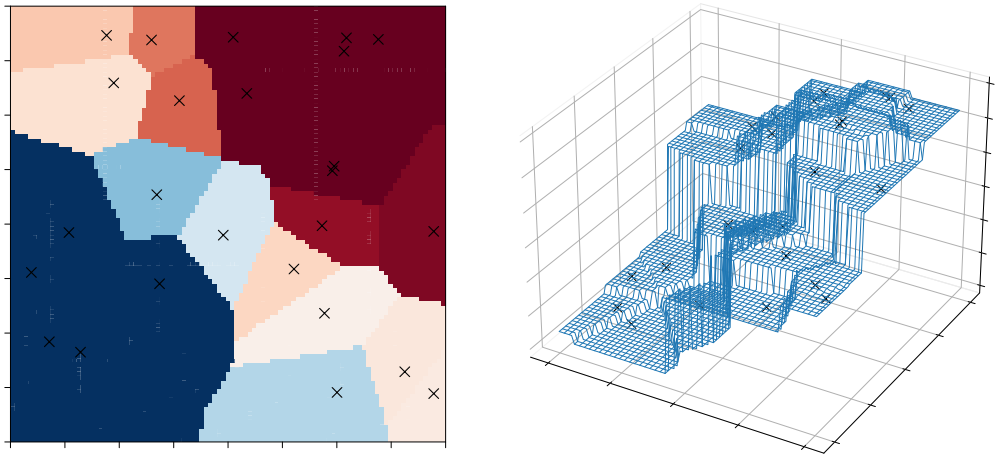


图 18. 二维插值，不规则散点，最近邻

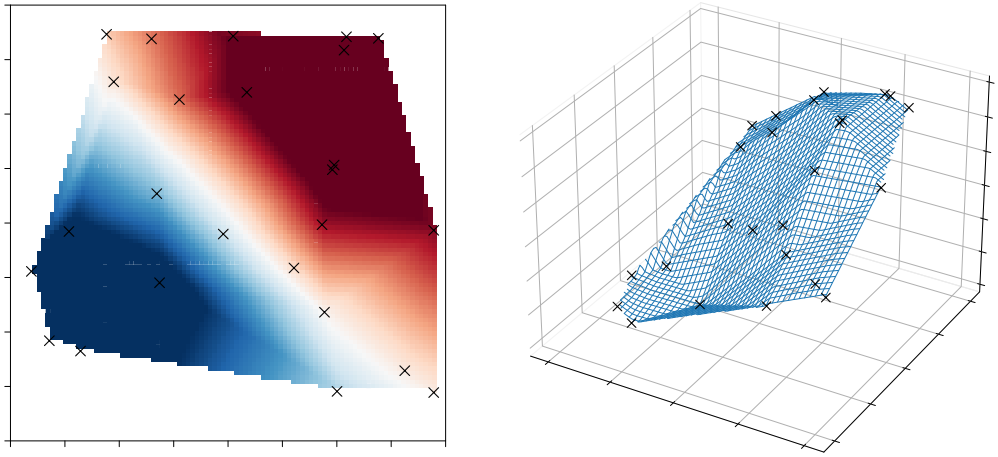


图 19. 二维插值，不规则散点，线性插值

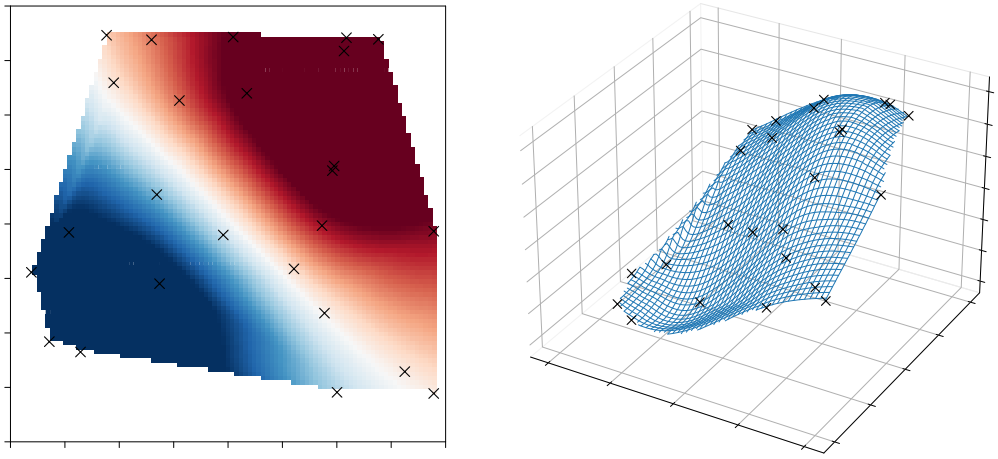


图 20. 二维插值，不规则散点，三次样条插值

Bk6_Ch05_04.py 完成不规则散点插值，并绘制图 18、图 19、图 20。

更多插值方法

matplotlib.pyplot.imshow() 绘图函数自带大量二维插值方法，请大家参考图 21。

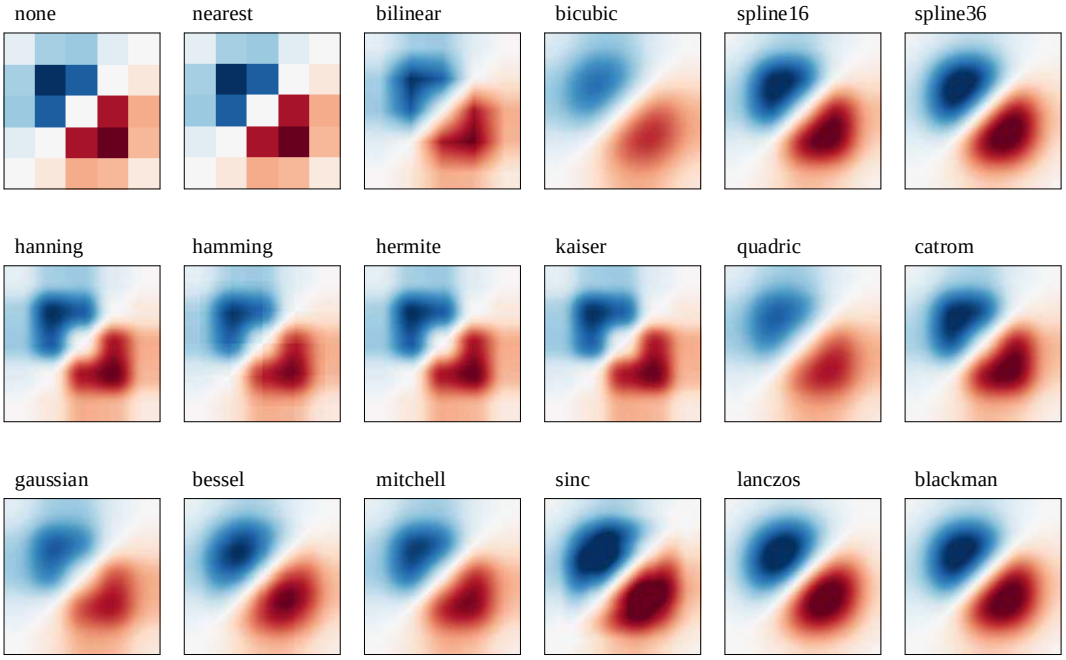


图 21. imshow() 函数插值方法

Bk6_Ch05_05.py 绘制图 21。

插值是一种通过已知数据点推断出连续函数在其他位置上取值的方法。在实际问题中，我们常常只知道一些离散的数据点，但需要通过这些数据点来推断出函数在其他位置的取值。插值可以通过拟合一条曲线、平面或者高维曲面来达到这个目的，从而实现对函数的估计。在机器学习中，插值可以用于对数据进行处理和预处理。插值可以通过拟合一条平滑的曲线或者曲面来填充数据中的缺失值，从而获得完整的数据集，这可以提高模型的准确性和可靠性。

请大家格外注意，插值和回归都是处理数据的方法，但插值是通过已知的数据点之间的值来估计未知点的值，而回归是通过已知的数据点来拟合一个函数，预测未知点的值。插值的目的是将数据点之间的缺失值或噪声进行平滑处理，而回归的目的是对数据进行预测和建模。虽然两者都是通过已知数据点来估计未知点的值，但它们的目的和使用场景是不同的。

6 Time Series 时间数据

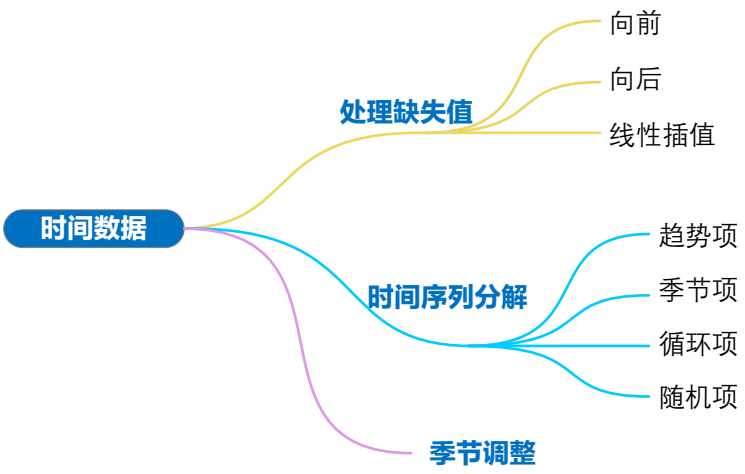
具有时间戳的数据序列

我们能看到的有限长的未来，但是面对无限多的问题。

We can only see a short distance ahead, but we can see plenty there that needs to be done.

—— 艾伦·图灵 (Alan Turing) | 英国计算机科学家、数学家，人工智能之父 | 1912 ~ 1954

- `statsmodels.api.tsa.seasonal_decompose()` 季节性调整
- `numpy.random.uniform()` 生成满足均匀分布的随机数
- `df.ffill()` 向前填充缺失值
- `df.bfill()` 向后填充缺失值
- `df.interpolate()` 插值法填充缺失值
- `seaborn.boxplot()` 绘制箱型图
- `seaborn.lineplot()` 绘制线图



6.1 时间序列数据

时间序列 (timeseries) 是一种特殊的数据类型，是指按照时间顺序排列的数据集合，其中每个数据点都与特定的时间点相关联。**时间戳** (timestamp) 可以精确到年份，月份，日期，甚至是小时、分、秒。

简单来说，时间序列可以用来描述某个变量随时间变化的趋势和模式。例如，一支股票的价格随时间变化的数据集就是一个时间序列，每个数据点对应着一个特定的日期和该日期下的股票价格。另一个例子是天气数据，例如每小时记录的温度、湿度和风速，它们也可以被组织成时间序列，以便分析和预测气象变化趋势。

如图 1 所示，**历史数据** (historical data) 是指已经发生的数据，它们是用来分析和理解过去发生的事件和趋势的。**预测数据** (forecasted data) 是指未来可能发生的数据，它们是根据历史数据和模型进行推算得出的。

历史数据可以用来训练模型，帮助模型学习过去的规律和趋势，从而提高预测的准确性。预测数据则可以用来制定决策、规划资源和制定策略。

历史数据和预测数据是相互依存的，历史数据是预测数据的基础，预测数据又可以帮助我们更好地理解历史数据。在时间序列分析中，历史数据和预测数据是两个不可或缺的部分。

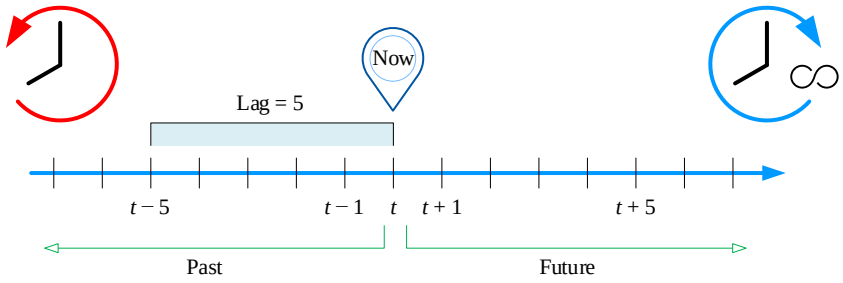


图 1. 时间轴

图 2 所示为 2020 年度中 9 支股票的每个营业日股价数据。图 2 中数据共有 253 行，每行代表一个日期及当日股价水平；时间数据表格共有 10 列，第 1 列为时间戳，其余 9 列每列为股价数据。除去时间戳一列和表头，图 2 可以看成一个矩阵。

Date	TSLA	TSM	COST	NVDA	FB	AMZN	AAPL	NFLX	GOOGL
2-Jan-2020	86.05	58.26	281.10	239.51	209.78	1898.01	74.33	329.81	1368.68
3-Jan-2020	88.60	56.34	281.33	235.68	208.67	1874.97	73.61	325.90	1361.52
6-Jan-2020	90.31	55.69	281.41	236.67	212.60	1902.88	74.20	335.83	1397.81
7-Jan-2020	93.81	56.60	280.97	239.53	213.06	1906.86	73.85	330.75	1395.11
8-Jan-2020	**98.43**	**57.01**	**284.19**	**239.98**	**215.22**	**1891.97**	**75.04**	**339.26**	**1405.04**
9-Jan-2020	96.27	57.48	288.75	242.62	218.30	1901.05	76.63	335.66	1419.79
...	...	...	...	...	...	...	...	...	...
21-Dec-2020	649.86	104.44	364.25	533.29	272.79	3206.18	128.04	528.91	1734.56
22-Dec-2020	640.34	103.55	361.32	531.13	267.09	3206.52	131.68	527.33	1720.22
23-Dec-2020	645.98	103.37	361.18	520.37	268.11	3185.27	130.76	514.48	1728.23
24-Dec-2020	661.77	105.57	363.86	519.75	267.40	3172.69	131.77	513.97	1734.16
28-Dec-2020	663.69	105.75	370.33	516.00	277.00	3283.96	136.49	519.12	1773.96
29-Dec-2020	665.99	105.16	371.99	517.73	276.78	3322.00	134.67	530.87	1757.76
30-Dec-2020	694.78	108.49	373.71	525.83	271.87	3285.85	133.52	524.59	1736.25
31-Dec-2020	705.67	108.63	376.04	522.20	273.16	3256.93	132.49	540.73	1752.64

图 2. 股票收盘股价数据

图 3 利用线图可视化股票收盘股价走势。图 3 (b) 右图初始股价归一化处理，这些曲线更容易比较不同股票的涨跌情况。

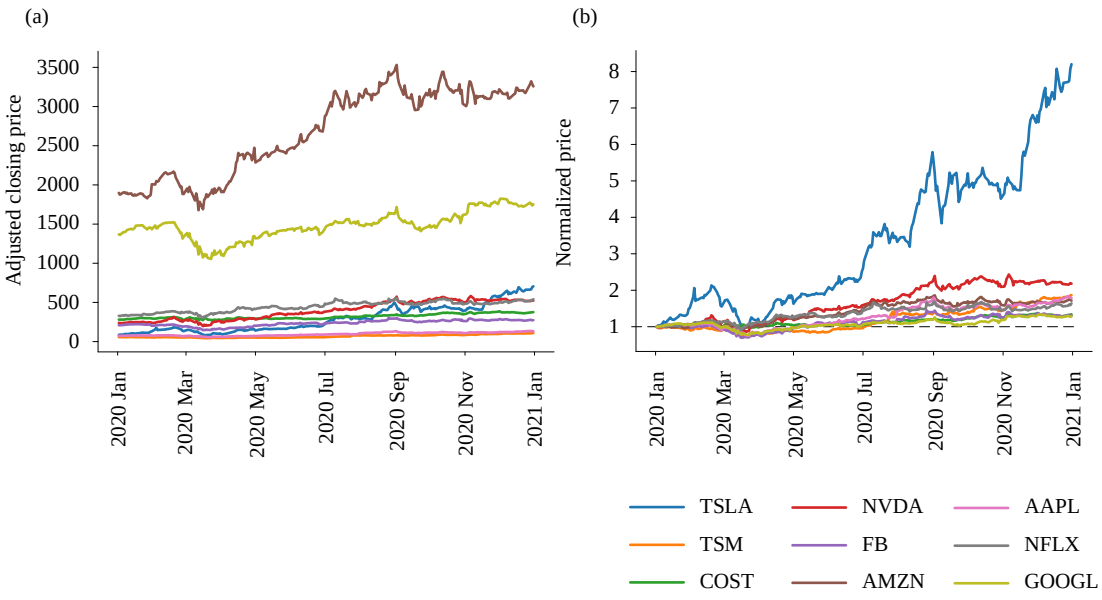


图 3. 股票收盘股价走势，和初始值归一化，时间序列数据

我们先介绍**损益** (Profit and Loss, PnL) 这个概念。损益 PnL 是指某个交易或投资策略在一定时期内的总收益或总损失。它是通过将所有交易的盈利和亏损加总起来得出的。正的 PNL 表示盈利，负的 PNL 表示亏损。如图 4 所示，只考虑某只股票收盘价 S 在 t 时刻和 $t-1$ 时刻 (工作日) 的变动，通过如下公式计算出 t 时刻的日损益：

$$\mathrm{PnL}_t = S_t - S_{t-1} \tag{1}$$

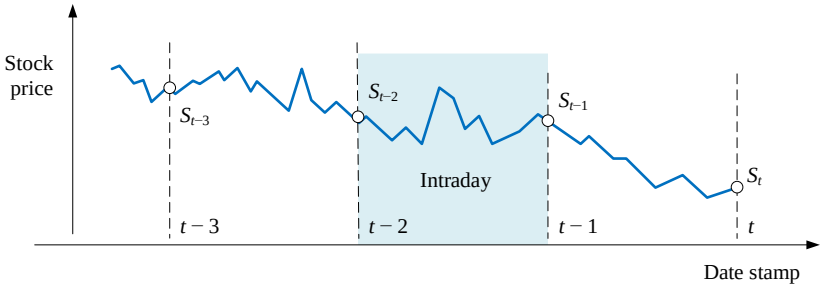


图 4. 某股票的价格变动

下面介绍**收益率** (return) 这个概念。在不考虑**分红** (dividend) 的条件下，单日简单回报率 (daily simple return) 可以这样计算：

$$r_t = \frac{S_t - S_{t-1}}{S_{t-1}} \tag{2}$$

股票分红是指上市公司根据其盈利情况，在向股东分配利润之后，以现金或股票形式再次向股东发放一部分盈利的行为。这种行为使得持有公司股票的股东可以从公司利润中获得收益，同时也是上市公司回报投资者、增强投资者信心的一种方式。分红通常以每股派息或每股送股的形式实施，也可以同时采用这两种方式。

量化金融建模还经常使用**日对数回报率** (daily log return)：

$$r_t = \ln\left(\frac{S_t}{S_{t-1}}\right) \tag{3}$$

对数收益率的计算结果具有可加性，也就是说，多个时间段的对数收益率之和等于总时间段的对数收益率。这个特性在计算投资组合收益率时非常有用。

量化金融建模时，一般会假设股价服从对数正态分布，这样对数收益率的分布更加接近正态分布，这对于一些金融模型的应用很实用，例如对冲基金、风险管理和投资组合优化等。本书后续经常使用日对数收益率。

图 5 所示为只股票在不同年份的日收益率分布，利用高斯分布估计样本分布多数情况下似乎是个不错的选择。图 6 所示为利用 KDE 估算得到概率密度。大家可以发现数据的统计量 (均值、方差、均方差、偏度、峰度) 随着时间变化。

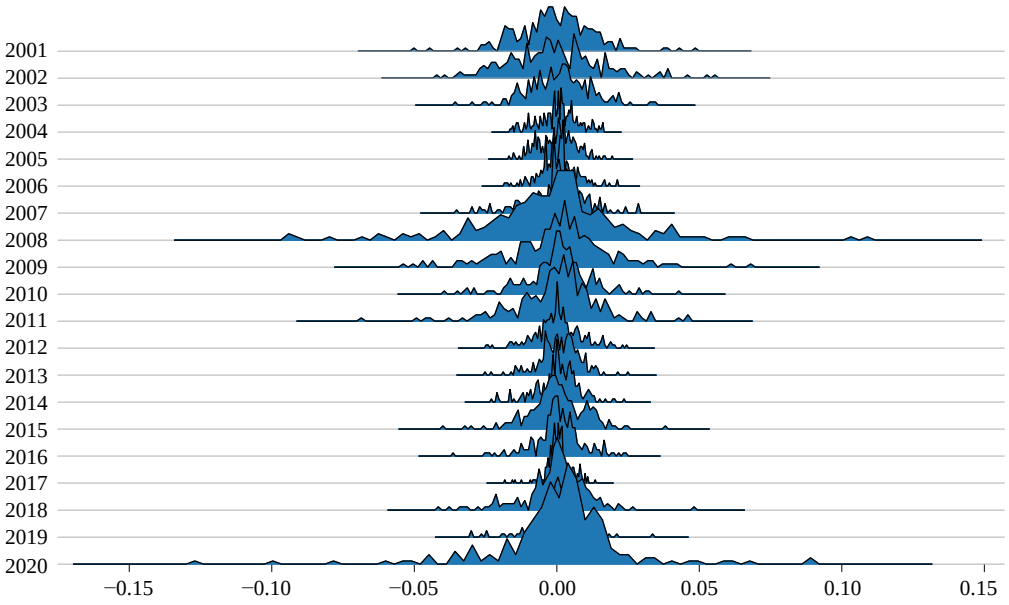


图 5. 收益率数据山脊图，按年分类

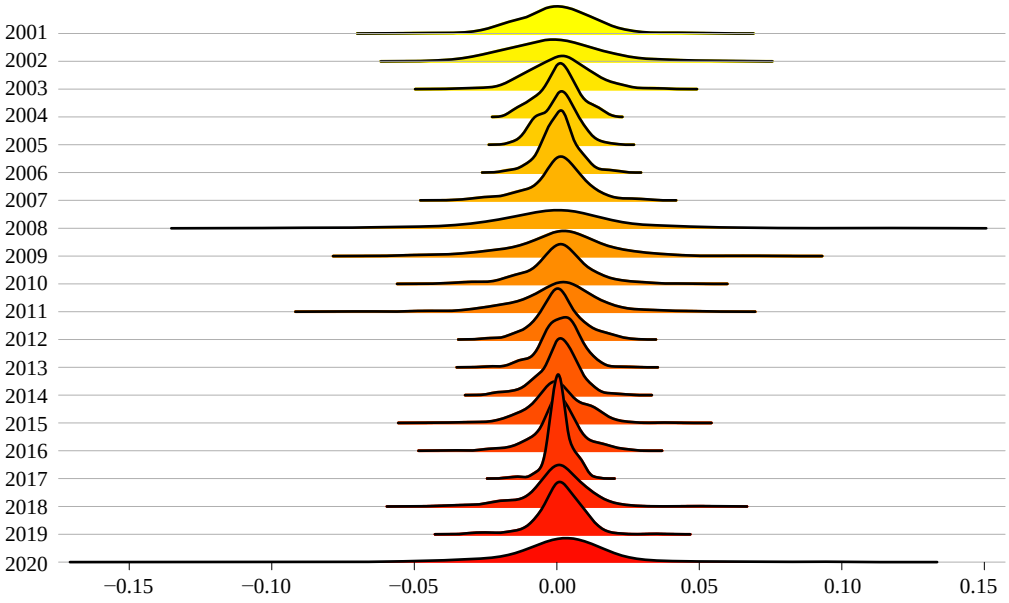


图 6. 收益率数据 KDE 山脊图，按年分类

鸢尾花数据，我们可以打乱数据的先后排列。但是时间序列是一个顺序序列，数据的先后顺序一般情况是不允许打乱的。有些情况，我们可以不考虑数据点的时间，比如图 7 所示回归分析中的散点图。

本书第 10、11 章将介绍线性回归模型。

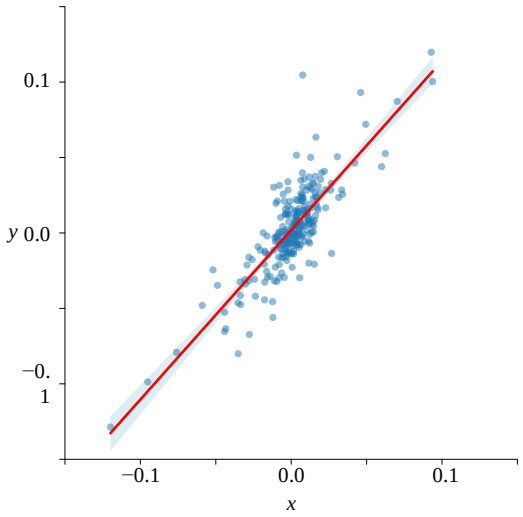


图 7. 线性 OLS 回归分析和散点图

6.2 处理时间序列缺失值

时间数据序列在分析建模之前，也需要注意数据中的缺失值和异常值处理。

本节从时间序列角度加以补充缺失值处理。

本书第 2、3 章分别介绍如何处理缺失值和异常值。

前文强调，时间序列数据是顺序观察的数据；因此在处理缺失值时，有其特殊性。比如，时间序列出具可以采用均值、众数、中位数、插值等一般方法，也可以采用如向前、向后这种方法。

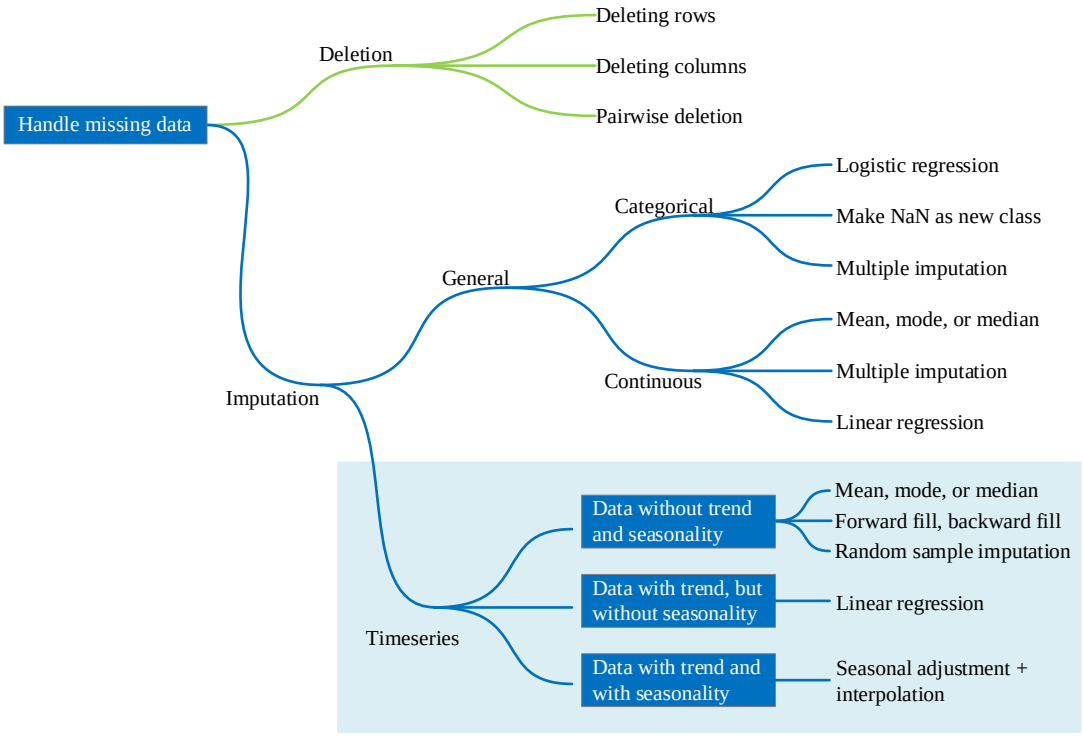


图 8. 处理缺失值

图 9 ~ 图 11 比较三种不同处理时间序列缺失值的基本方法。

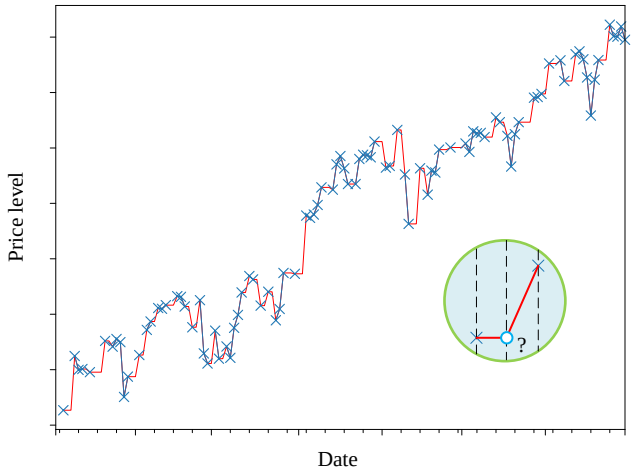


图 9. 向前插值填充缺失值

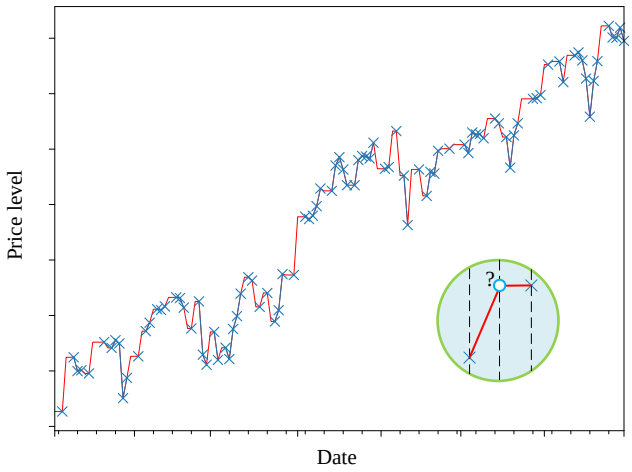


图 10. 向后插值填充缺失值

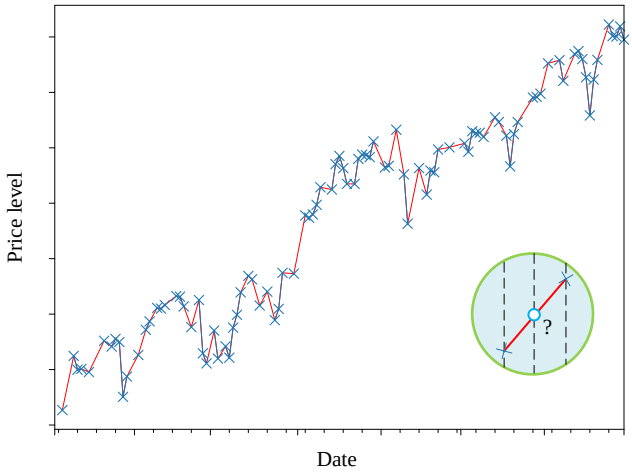


图 11. 线性插值填充缺失值

Bk6_Ch06_01.py 绘制图 9 ~ 图 11。

6.3 从时间数据中发现趋势

本节利用美国失业率数据介绍如何从时间数据中发现趋势。图 12 所示为失业率的原始数据。数据从 1950 年开始到 2021 年，每月有一个数据点。

观察图 12 这幅图，虽然存在“噪音”，我们已经能够大致看到失业率的按照年份的大致走势。下一章会介绍移动平均的方法来消除“噪音”。

观察图 12 的局部图中，我们还发现不同年份中一年内失业率存在某种特定的“模式”。也就是说，图中的“噪音”可能存在重要的价值！

图 13 所示为按月同比规律。同比是一种比较方式，用于比较同一时间段内两年或多年的某项指标的变化情况。同比通常表示为百分比或比率，可以用来分析和评估一个公司或经济指标在不同年份间的表现。

同比的计算方法是，将当前时间段的指标值减去同一时间段上一年的指标值，然后将差值除以上一年的指标值，再乘以 100%。这个计算结果就是同比指标，可以表示为百分比。

与历史同时期比较，例如 2005 年 7 月份与 2004 年 7 月份相比称其为同比。相比图 12，图 13 更容易发现失业率变化规律。

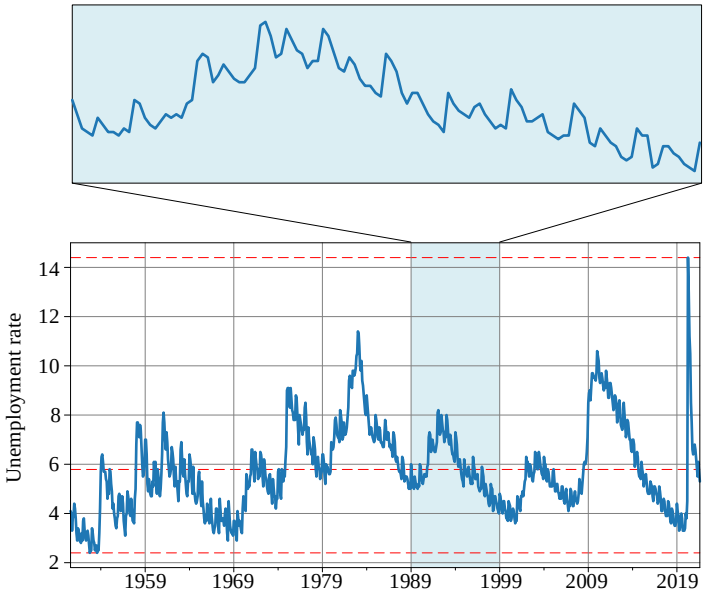


图 12. 原始失业率数据和局部放大图

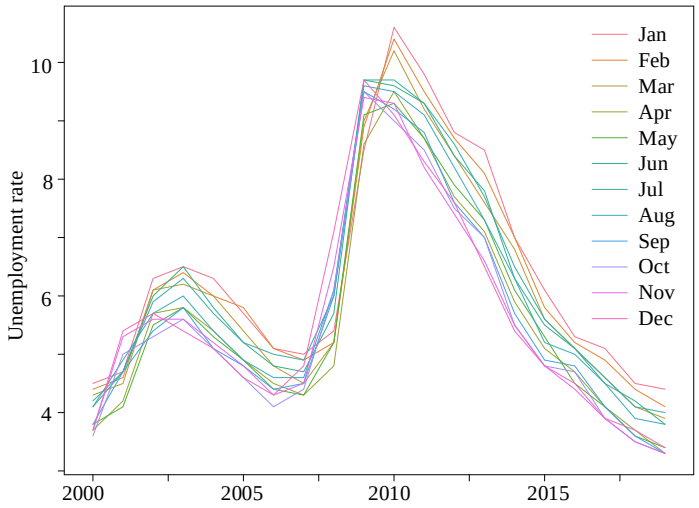


图 13. 失业率，按月同比

图 14 所示为年内环比数据。环比是一种比较方式，用于比较相邻两个时间段内某项指标的变化情况。环比指标通常表示为百分比或比率，可以用来分析和评估一个公司或经济指标在不同时间段内的表现。

环比的计算方法是，将当前时间段的指标值减去上一个时间段的指标值，然后将差值除以上一个时间段的指标值，再乘以 100%。这个计算结果就是环比指标，可以表示为百分比。

与上一统计段比较，例如 2005 年 7 月份与 2005 年 6 月份相比较称其为环比。我们似乎发现失业率存在某种年度周期规律。一年之内春天的失业率往往较低，这似乎和春天农业生产用工有关。而每一年的一月份的失业率显著提高，这可能和圣诞节、新年节庆之后用工下降有关。

为了进一步看到失业率随年度变化，我们可以用箱型图对年内失业率数据加以归纳，如图 15 所示。箱型图的均值代表年度失业率的平均水平。箱型图的四分位间距 IQR 告诉我们年度失业率的变化幅度。显然，失业率在 2020 年出现“前所未闻”的大起大落。

图 16 所示为月份失业率箱型图。比较月份失业率的平均值变化，一月份的平均失业率确实陡然升高，这也印证了之前的猜测。下一节，我们就介绍如何将不同的成分从原始时间数据从分离出来。

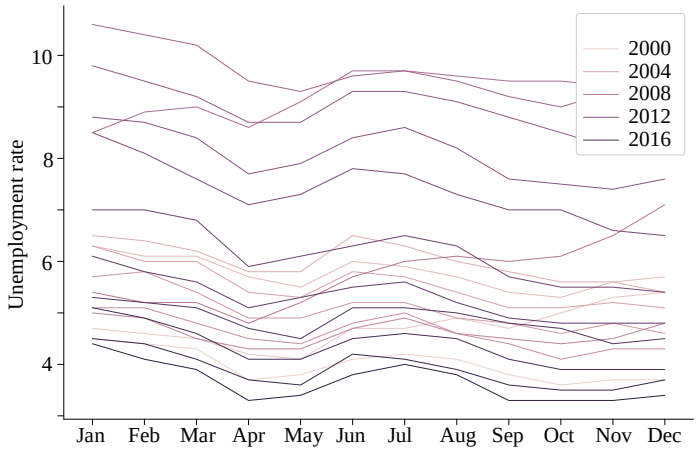


图 14. 失业率，年内环比

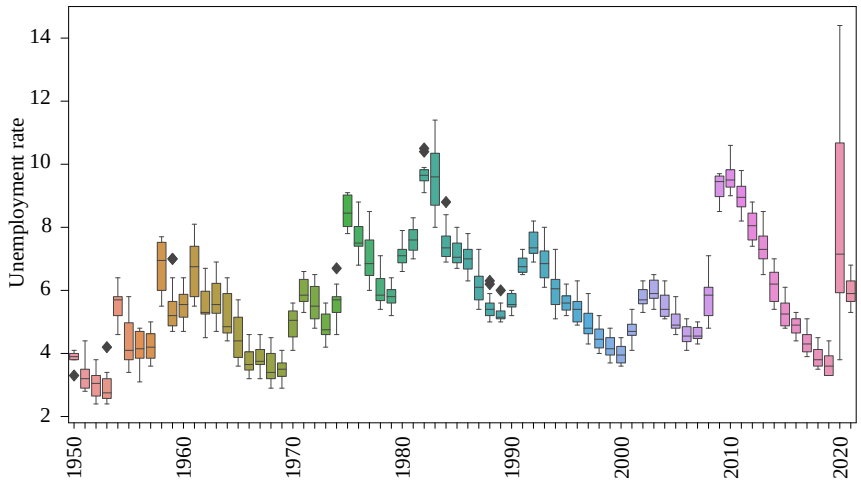


图 15. 年度失业率数据箱型图

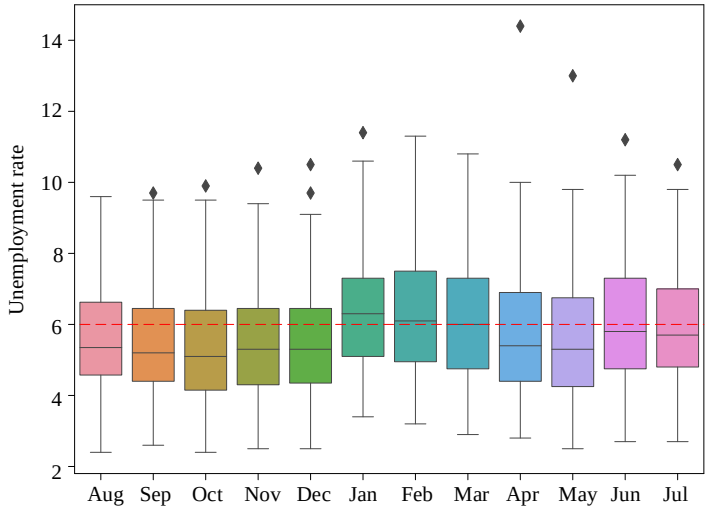


图 16. 月份失业率数据箱型图

Bk6_Ch06_02.py 绘制本节图像。

6.4 时间序列分解

时间序列有如图 17 所示的几种主要的组成部分。具体定义如下：

- **趋势项** (trend component) $T(t)$，表征时间序列中确定性的非季节性长期总体趋势，通常呈现出线性或非线性的持续上升或者持续下降。当一个时间序列数据长期增长或者长期下降时，表示该序列有趋势。在某些场合，趋势代表着“转换方向”。例如从增长的趋势转换为下降趋势。
- **季节项** (seasonal component) $S(t)$，表征时间序列中确定性的周期季节性成分，是在连续时间内 (例如连续几年内) 在相同时间段 (例如月或季度) 重复性的系统变化。当时间序列中的数据受到季节性因素的影响时，表示该序列具有季节性。季节性总是一个已知并且固定的频率。
- **循环项** (long-run cycle component) $C(t)$。循环项代表是相对周期更长 (例如几年或者十几年) 的重复性变化，但一般没有固定的平均周期，往往与大型经济体的经济周期息息相关。有时由于时间跨度较短，循环项很难体现出来，这时可能就被当作趋势项来分析了。当时间序列数据存在不固定频率的上升和下降时，表示该序列有周期性。这些波动经常由经济活动引起，并且与“商业周期”有关。周期波动通常至少持续两年。
- **随机项** (stochastic component) $I(t)$，表征时间序列中随机的不规则成分，体现出一定的自相关性以及持续时间内无法预测的周期。该成分可以是噪声，但不一定是。往往认为随机项包含有与业务自身密切相关的信息。

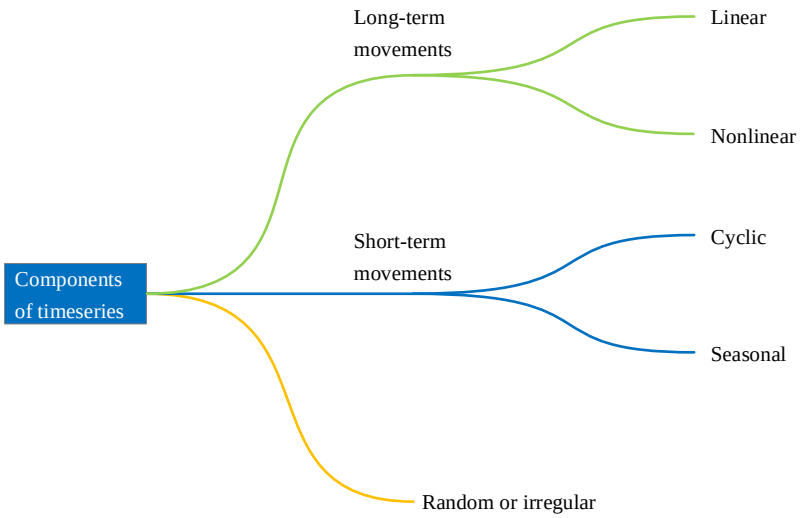


图 17. 时间序列成分

许多时间序列同时包含趋势、季节性以及周期性。基于以上的主要成分，一个时间序列可以有以下几种组合模型。

加法模型

加法模型 (additive model)，各个成分直接相加得到：

$$X(t)=T(t)+S(t)+C(t)+I(t) \tag{4}$$

这可能是最常用的时间序列分解方式。如果一个时间序列仅仅由趋势项 $T(t)$ 和随机项 $I(t)$ 构成：

$$X(t) = T(t) + I(t) \tag{5}$$

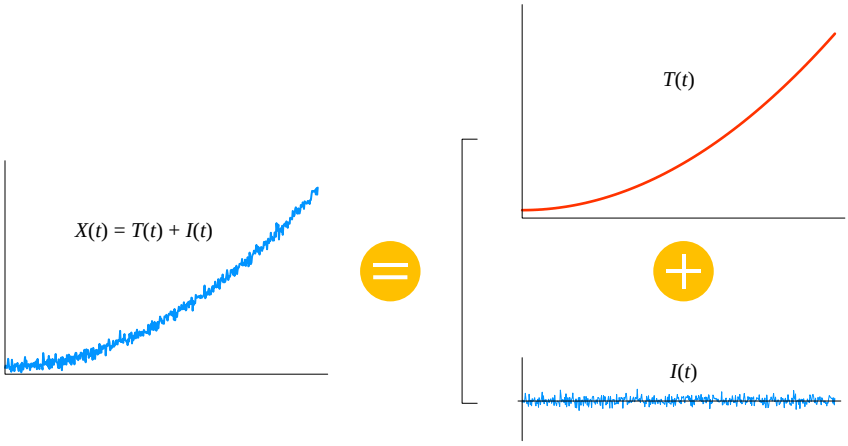


图 18. 累加分解，原始数据 X(t)被分解为趋势成分 $T(t)$和噪音成分 $I(t)$

标普 500 指数长期来看随时间增长，按照经济周期涨跌，短期来看指数每天波动不止。长期**趋势成分** (trend component) $TR(t)$ 就可以描述这种时间序列的长期行为，而**不规则成分** (irregular component) $IR(t)$ 描述的就是噪音成分，或者说是随机运动成分。

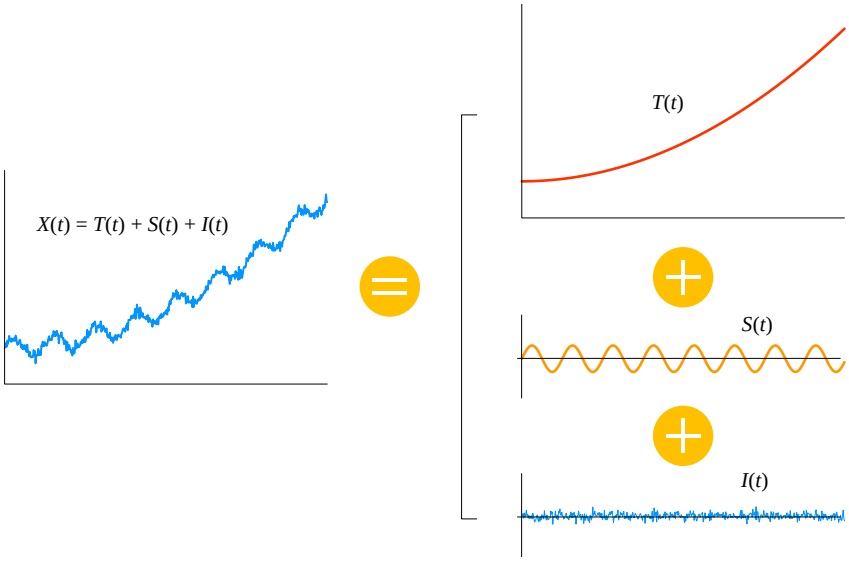


图 19. 累加分解，原始数据 X(t)被分解为趋势成分 $T(t)$、季节成分 $S(t)$ 和噪音成分 $I(t)$

乘法模型

乘法模型 (multiplicative model)，各个成分直接相乘得到：

$$X(t)=T(t)\cdot S(t)\cdot C(t)\cdot I(t) \tag{6}$$

如果只考虑趋势项 $T(t)$ 和随机项 $I(t)$：

$$X(t)=T(t)\cdot I(t) \tag{7}$$

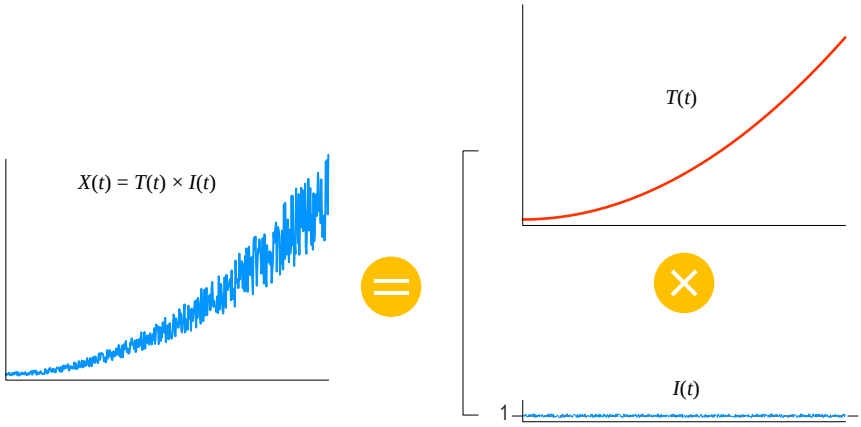


图 20. 累乘分解，原始数据 X(t)被分解为趋势成分 $T(t)$和噪音成分 $I(t)$

考虑季节成分的乘法模型：

$$X(t)=T(t)\cdot S(t)\cdot I(t) \tag{8}$$

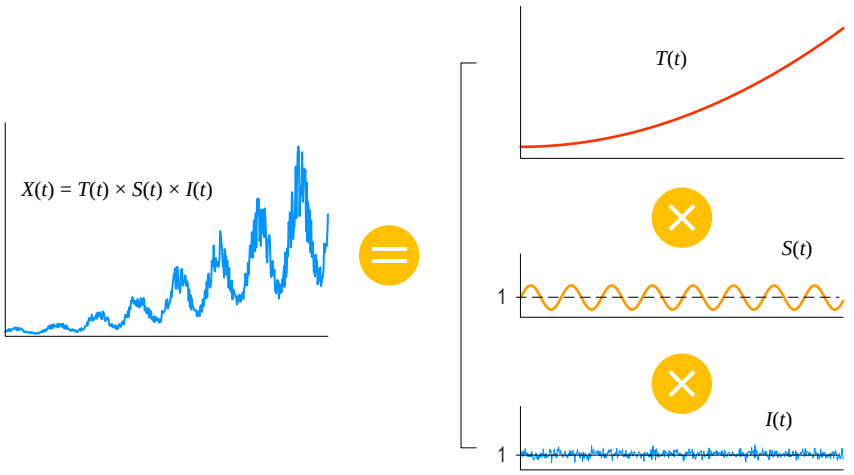


图 21. 累乘分解，原始数据 X(t)被分解为趋势成分 $T(t)$和噪音成分 $I(t)$

当然，时间序列还可以存在其他分解模型。比如**对数加法模型** (log-additive model)，时间序列取对数后由各个成分相加得到：

$$\ln X(t)=T(t)+S(t)+C(t)+I(t) \tag{9}$$

上式相当于对 $X(t)$ 进行对数转换。对于更复杂的时间序列分解模型，本书不做介绍。

6.5 季节调整

季节性调整 (seasonal adjustment) 是一种经济学上的数据处理技术，用于消除某些变量在特定季节内的周期性波动。季节性调整的目的是将原始数据中的季节性因素剔除，从而更准确地了解某个经济指标的实际趋势。

季节性调整通常应用于具有季节性波动的经济指标，例如销售额、就业率、消费水平等。由于不同季节的天气、节日、促销活动等因素都会影响这些指标的变化，因此原始数据往往会出现季节性波动。

季节性调整的方法通常是通过构建季节性模型来预测和剔除季节性波动，常用的方法包括移动平均法、指数平滑法和回归分析等。调整后的数据更能反映出经济指标的实际趋势，有助于进行更准确的分析和决策。

本节利用 scipy.stats.tsa.seasonal_decompose() 函数完成本章前文失业率数据的季节性调整。这个函数同时支持加法模型，seasonal_decompose(series, model='additive')，和乘法模型，seasonal_decompose(series, model='multiplicative')。本节采用的是默认的加法模型。

图 22 所示为失业率数据的分解。图 22 (a) 为原始数据，图 22 (b) 为趋势成分，图 22 (c) 为季节成分，图 22 (d) 为噪音成分。

注意，图 22 四副子图的纵轴尺度完全不同。

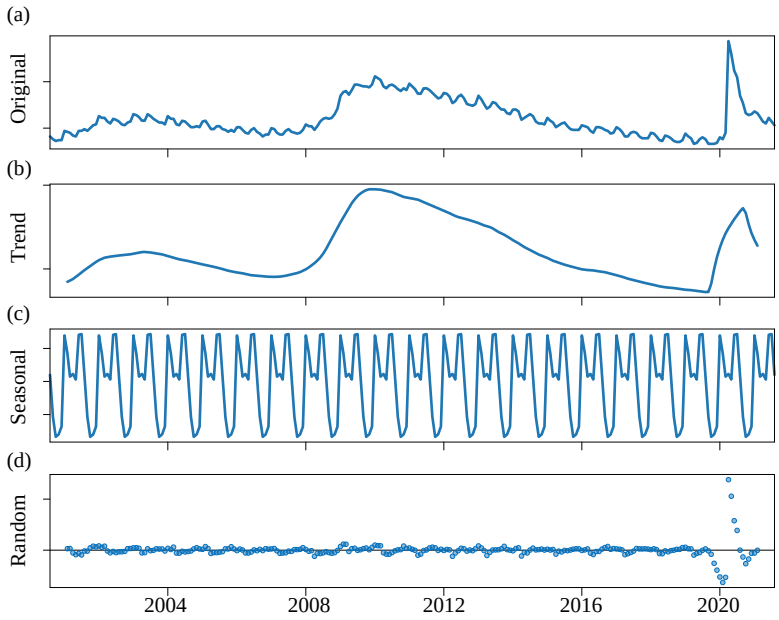


图 22. 失业率数据的分解

图 23、图 24、图 25 三幅图分别展示这四种成分。

scipy.stats.tsa.seasonal_decompose() 函数采用比较简单卷积方法进行季节调整，对于更复杂的季节性调整，建议大家了解 X11 模型。

X11 模型是一种用于季节性调整的统计方法，它是 Census Bureau 在 1967 年开发的，是 ARIMA 模型的一种扩展。X11 模型能够预测和剔除原始数据中的季节性因素，从而更准确地反映某个经济指标的趋势。本书不展开讲解 X11 模型。

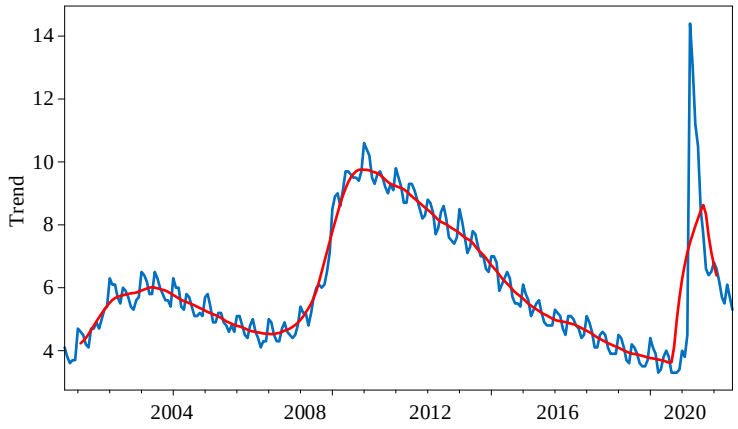


图 23. 比较原始数据和趋势成分

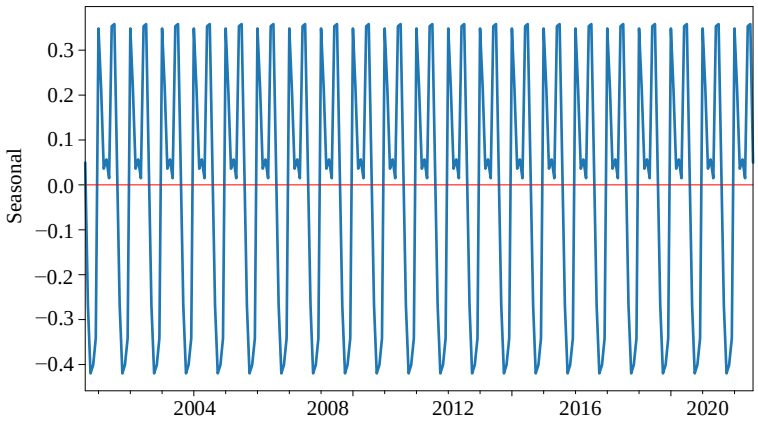


图 24. 季节成分

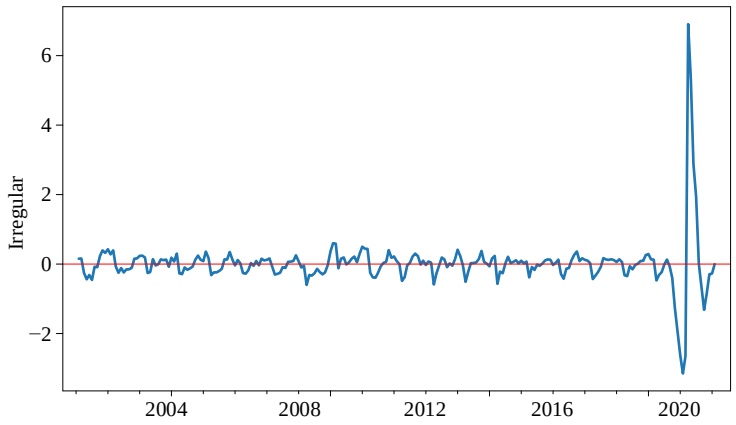


图 25. 噪音成分

Bk6_Ch06_03.py 绘制本节图像。

时间序列是一种按时间顺序排列的数据序列，用于描述某个现象、变量或指标随时间变化的规律。时间序列常用于经济学、金融学、气象学、医学等领域，例如股票价格、气温、血压等指标。

时间序列中可能存在缺失值和离群值，这些异常值可能会影响时间序列分析的准确性。处理缺失值的方法包括插值法、回归法、拉格朗日插值法等。处理离群值的方法包括删除、替换、缩尾等，具体选择哪种方法需要根据实际情况来确定。

时间序列分解是一种将时间序列分解为趋势项、季节项、循环项、随机项等等成分方法。季节调整是时间序列分析的一种重要应用，用于消除时间序列中的季节性因素，以便更好地分析序列的趋势和周期性。

时间序列分析是一种非常重要的统计方法，可以帮助我们了解和预测经济、自然和社会现象的趋势和变化规律，对于决策和规划具有重要意义。

7 Rolling Window
移动窗口

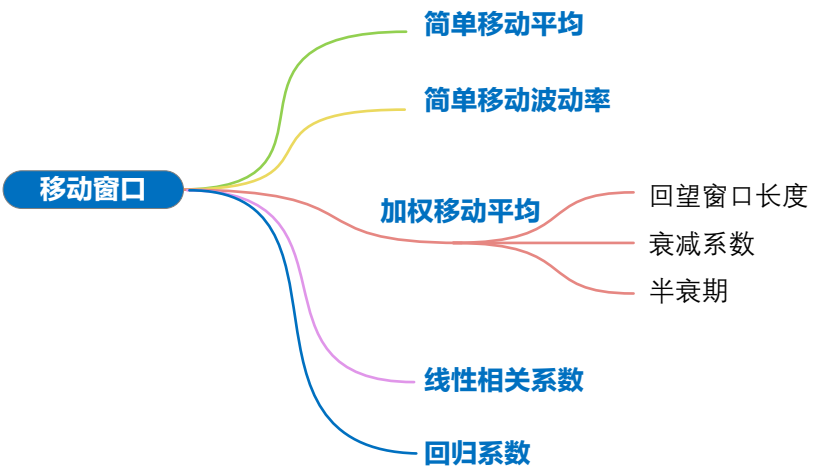
移动窗口展示数据之间动态关系

没有一种语言比数学更普遍、更简单、更没有错误、更不晦涩……更容易表达所有自然事物的不变关系。它用同一种语言解释所有现象，仿佛要证明宇宙计划的统一性和简单性，并使主导所有自然原因的不变秩序更加明显。

There cannot be a language more universal and more simple, more free from errors and obscurities...more worthy to express the invariable relations of all natural things than mathematics. It interprets all phenomena by the same language, as if to attest the unity and simplicity of the plan of the universe, and to make still more evident that unchangeable order which presides over all natural causes.

—— 约瑟夫·傅里叶 (Joseph Fourier) | 法国数学家、物理学家 | 1768 ~ 1830

- `df.ewm().mean()` 计算数据帧 `df` EWMA 平均值
- `df.ewm().std()` 计算数据帧 `df` EWMA 标准差/波动率
- `df.rolling().corr()` 计算数据帧 `df` 的移动相关性
- `df.rolling().kurt()` 计算数据帧 `df` 滚动峰度
- `df.rolling().max()` 计算数据帧 `df` 滚动最大值
- `df.rolling().mean()` 计算数据帧 `df` 滚动均值
- `df.rolling().min()` 计算数据帧 `df` 滚动最小值
- `df.rolling().quantile()` 计算数据帧 `df` 滚动百分位值
- `df.rolling().skew()` 计算数据帧 `df` 滚动偏度
- `df.rolling().std()` 计算数据帧 `df` MA 平均值
- `statsmodels.regression.rolling.RollingOLS()` 计算移动 OLS 线性回归系数



7.1 移动窗口

移动窗口 (rolling window, moving window) 是一种重要的时间序列统计计算方法。移动窗口按照一定规律沿着历史数据移动，每一个位置都产生一个统计量，比如最大值、最小值、平均值、加权平均值、标准差等等。移动窗口方法可以消除时间序列中的随机噪声，减少数据波动，更好地反映数据的趋势和周期性。

随着移动窗口不断滚动，特定统计量不断产生；因此，通过移动窗口得到的数据是序列数据，也就是时间序列。移动窗口的宽度叫做**回望窗口长度** (lookback window length)。

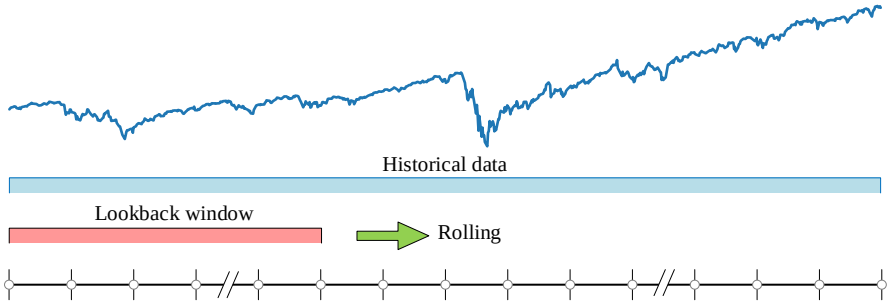


图 1. 移动窗口

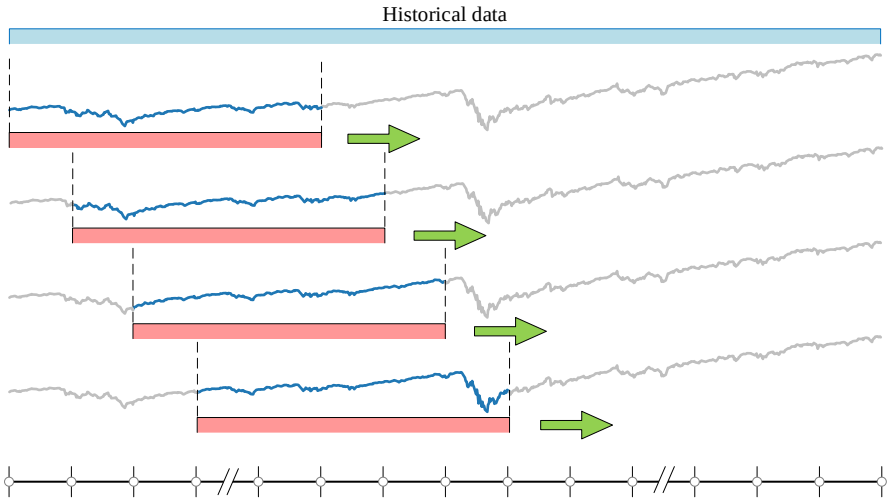


图 2. 移动窗口不断移动产生新的时间序列

最大、最小

如图 3 所示，利用长度为 100 营业日的回望窗口，我们可以得到移动最大值 (橙色) 和移动最小值 (绿色) 曲线。随着移动窗口移动到每一个位置，便利用回望窗口内的数据产生一个最大值和最小值。当移动窗口最左端和历史数据的最左端对齐时，产生第一个数据；而这个数据位于移动

窗口的最右端。因此，移动窗口数据长度比历史数据长度短。对于某个数据帧数据 df，移动最大值和最小值时间序列可以利用 df.rolling().max() 和 df.rolling().min()两个函数计算得到。

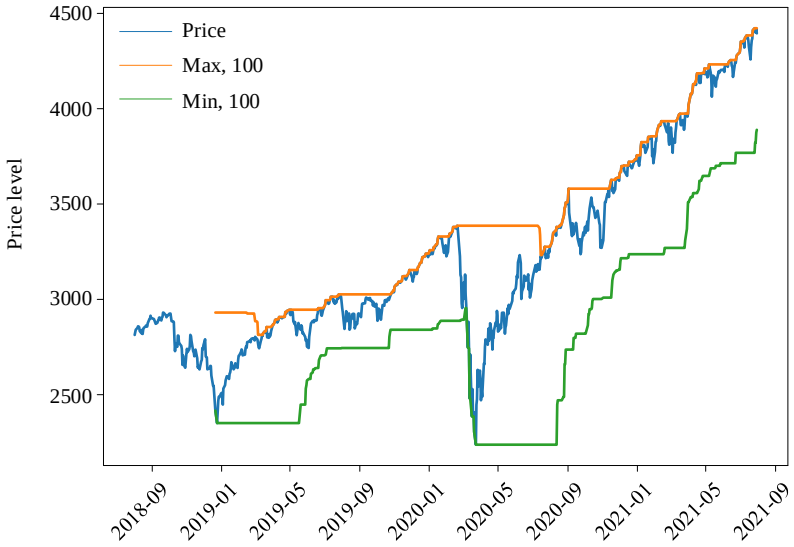


图 3. 移动最大和最小，回望窗口长度为 100

简单移动平均

简单移动平均数 (simple moving average, SMA)，是时间序列分析中常用的一种方法，用于平滑时间序列数据。SMA 的计算方法是将某一时间段内的数据求平均值，然后移动到下一个时间段内，继续计算平均值，如此重复直到计算完整个时间序列。SMA 具体运算如下：

$$\begin{aligned}\bar{x}_{\text{SMA\_}k} &= \frac{x_{k-L+1} + x_{k-L+2} + \ldots + x_{k-2} + x_{k-1} + x_k}{L} \\ &= \frac{x_{(k-L)+1} + x_{(k-L)+2} + \ldots + x_{k-2} + x_{k-1} + x_k}{L} \\ &= \frac{1}{L}\sum_{i=1}^{L} x_{(k-L)+i}\end{aligned} \tag{1}$$

SMA 有助于消除短期波动带来的数据噪音，突出长期趋势。移动平均相当于一个滤波器；回望窗口长度影响着统计量数据平滑度。SMA 的计算过程中，每个数据点的权重相等，因此对于较短的时间段，SMA 能够更好地反映数据的短期趋势和波动性，但对于长期趋势和周期性较弱的数据，则可能不太准确。

图 5 比较回望窗口分别为 50、100 和 150 三种情况的移动平均值。可以发现，回望窗口越长，得到的统计量时间序列看起来越平滑。

对于数据帧数据 df，移动平均可以用 df.rolling().mean() 计算得到。对于采样频率为营业日的数据，常见的移动窗口回望长度可以是 5 天 (一周)、10 天 (两周)、20 天 (一个月)、60 天 (一个季度)、125/126 天 (半年) 或 250/252 天 (一年) 等等。

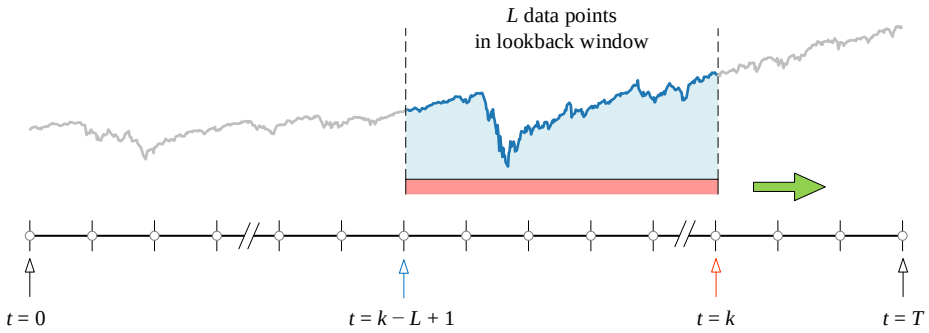


图 4. 回望窗口内数据序号

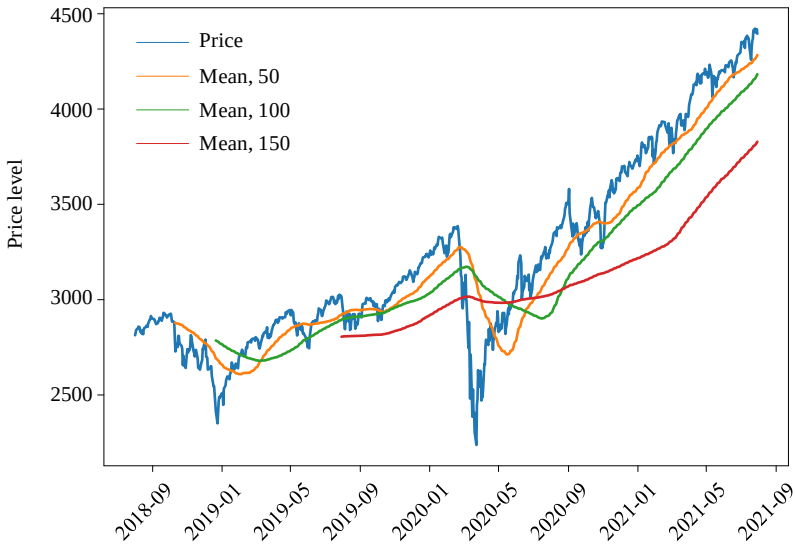


图 5. 移动平均，不同窗口长度

其他统计量

此外，移动窗口还可以帮助我们理解数据统计特点的动态特征。图 6 所示为日收益率的移动期望、波动率、偏度和峰态。**波动率** (volatility) 就是标准差。可以发现数据的统计特征随着时间移动不断改变。

对于数据帧数据 df，df.rolling().std()、df.rolling().skew() 和 df.rolling().kurt() 可以分别计算滚动标准差、偏度和峰度。

请大家改变回望窗口长度比较结果。

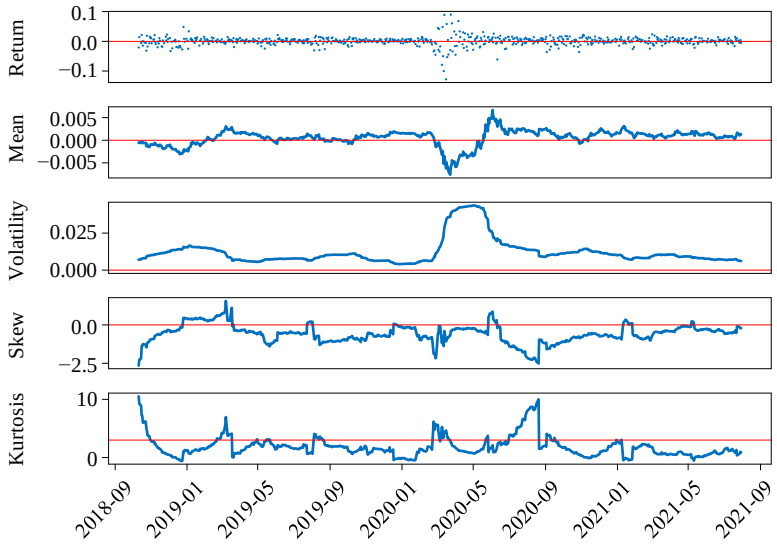


图 6. 日收益率的移动期望、波动率 (标准差)、偏度和峰态

类似地，图 7 所示为日收益率的 95%和 5%移动百分位变化。对于数据帧数据 df，df.rolling().quantile() 计算滚动百分位值。

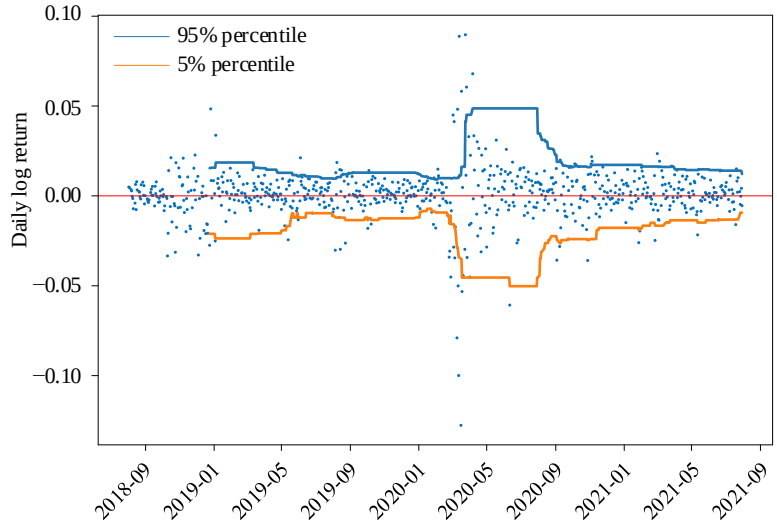


图 7. 移动百分位，95%和 5%

7.2 移动波动率

回望窗口长度为 L 的条件下，时间序列移动波动率为：

$$\sigma_{\text{daily\_}k} = \sqrt{\frac{1}{L-1}}\sqrt{\sum_{i=1}^{L}\left(x_{(k-L)+i}-\mu\right)^2} \tag{2}$$

其中，μ 为回望窗口内数据 x_i 的平均值。时间序列波动率的大小可以反映时间序列数据的风险程度，即数据变化的不确定性程度。通常情况下，波动率越大，数据变化的不确定性越高，风险也就越大。在金融市场分析中，时间序列波动率被广泛应用于风险管理和投资决策。例如，股票的波动率可以帮助投资者评估其风险水平，从而做出更明智的投资决策。

当 L 足够大，且 μ 几乎为 0 时，(2) 可以简化为：

$$\sigma_{\text{daily}} = \sqrt{\frac{\sum_{i=1}^{L}\left(x_{(k-L)+i}\right)^2}{L}} \tag{3}$$

(3) 相当于对回望窗口内 $(x_i)^2$ 数据，施加完全相同的权重 $1/L$；因此，这种波动率也被叫做**移动平均波动率** (moving average volatility)。

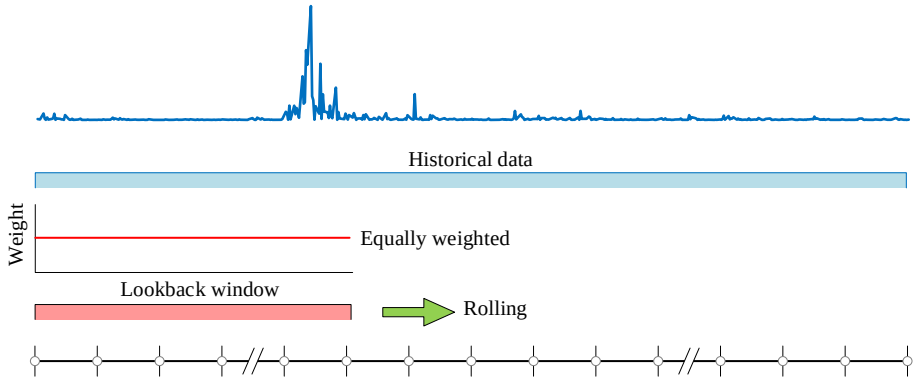


图 8. 移动平均

(3) 常用来计算股票收益率的波动率。图 10 所示为不同窗口长度条件下得到的移动平均波动率。可以发现，窗口长度越长数据越平缓，但是对数据变化响应越缓慢。

白话说，回望窗口长度越长，窗口内相对更具影响力的“陈旧”数据越尾大不掉，代谢的周期越长。下一节介绍的指数加权移动平均 EWMA，便很好地解决这一问题；哪怕回望窗口越长，EWMA 计算得到的波动率也能更快地跟踪数据变化规律。

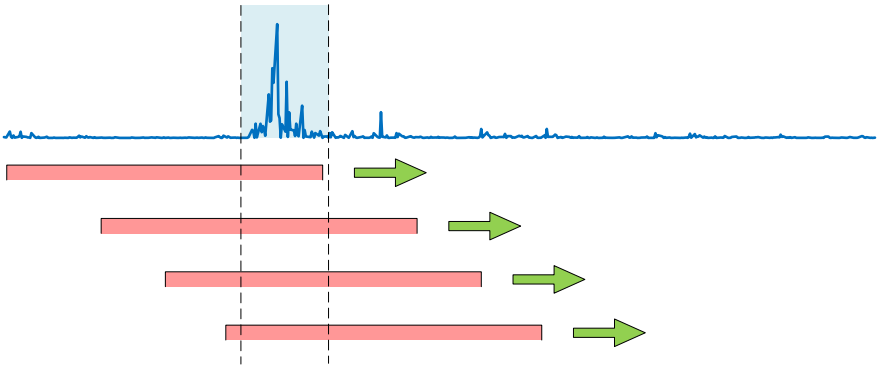


图 9. 尾大不掉的“陈旧”数据

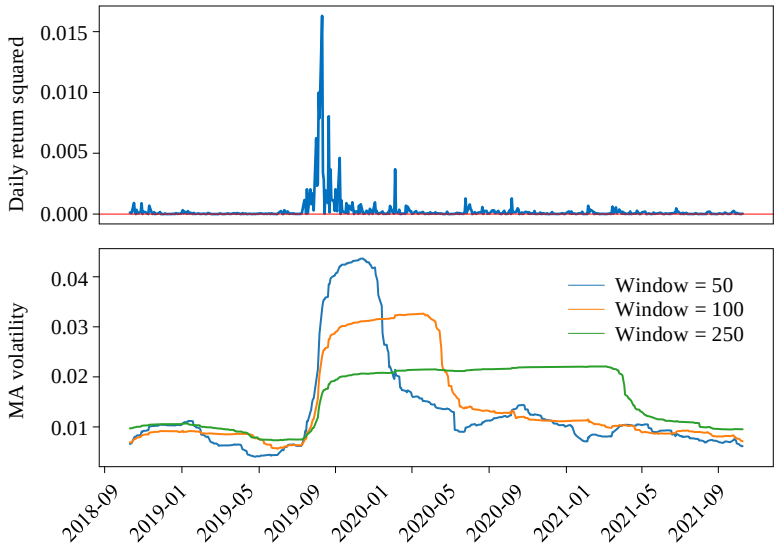


图 10. 移动平均 MA 单日波动率，不同窗口长度

此外，$\pm 2\sigma$ 波动率带常用来检测时间数据中可能存在的异常值。+2σ 曲线被称之为+2σ 上轨，−2σ 曲线常被称之为−2σ 下轨。图 11~图 13 分别展示窗口长度为 50 天、100 天和 250 天的$\pm 2\sigma$ 移动平均 MA 波动率带宽。

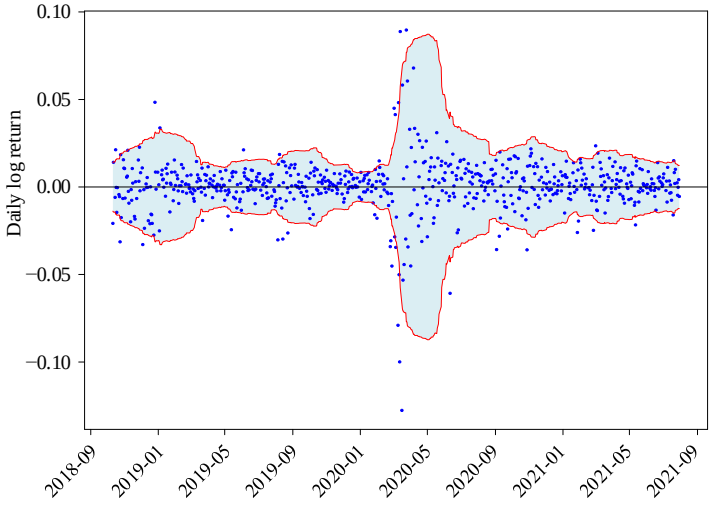


图 11. ±2σ 移动平均 MA 波动率带宽，窗口长度 50 天

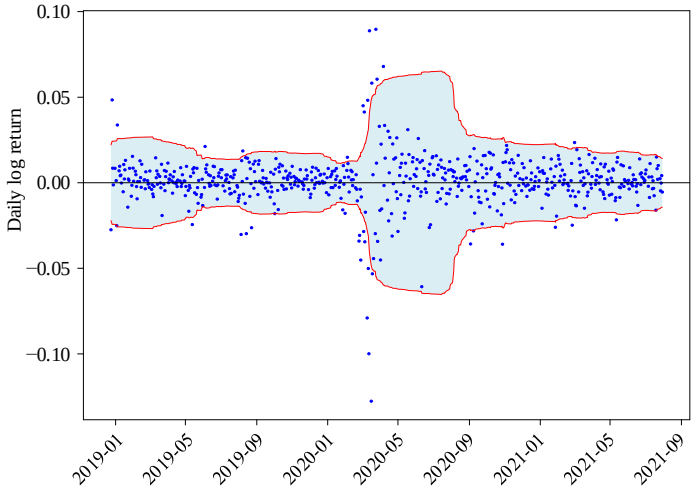


图 12. ±2σ 移动平均 MA 波动率带宽，窗口长度 100 天

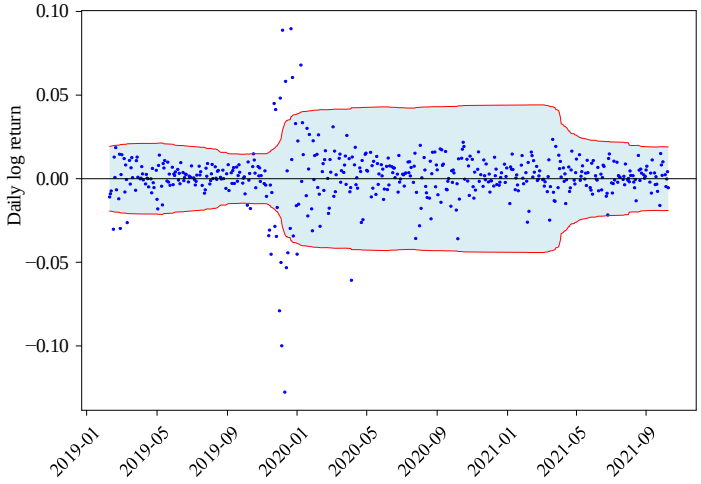


图 13. ±2σ 移动平均 MA 波动率带宽，窗口长度 250 天

时间平方根法则

时间平方根法则 (square root of time) 可以将日波动率转化为年化波动率：

$$\sigma_{annual} = \sqrt{250} \cdot \sigma_{daily} \tag{4}$$

式中的 250 代表假设一年有 250 营业日。图 14 所示为不同窗口长度条件下，移动平均 MV 年化波动率随时间变化情况。

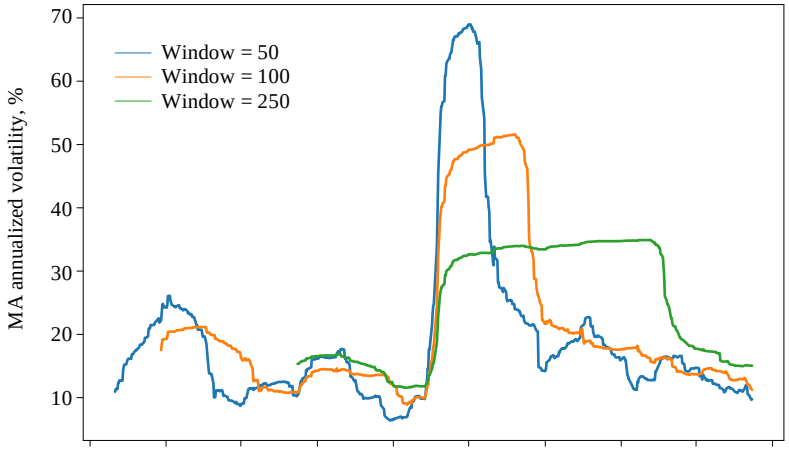


图 14. 移动平均 MV 年化波动率百分数，不同窗口长度

Bk6_Ch07_01.py 绘制上一节和本节主要图像。

7.3 指数加权移动平均

指数加权移动平均 (exponentially-weighted moving average, EWMA) 可以用来计算平均值、标准差、方差、协方差和相关性等等。EWMA 是对前文的简单移动平均的改进。EWMA 方法的特点是，对窗口内越近期的数据给予更高权重，越陈旧数据越低权重。权重的衰减过程为指数衰减。这种方法可以在平滑数据的同时保留较新数据的影响。

指数加权移动平均数 (exponential moving average, EMA, or exponentially weighted moving average) 定义为：

$$\bar{x}_{\text{EWMA\_}k} = \left(\frac{1-\lambda}{1-\lambda^{L}}\right)\frac{x_{k-L+1}\lambda^{L-1} + x_{k-L+2}\lambda^{L-2} + \ldots + x_{k-2}\lambda^{2} + x_{k-1}\lambda^{1} + x_{k}\lambda^{0}}{L} \tag{5}$$

其中，λ 为**衰减系数** (decay factor)。

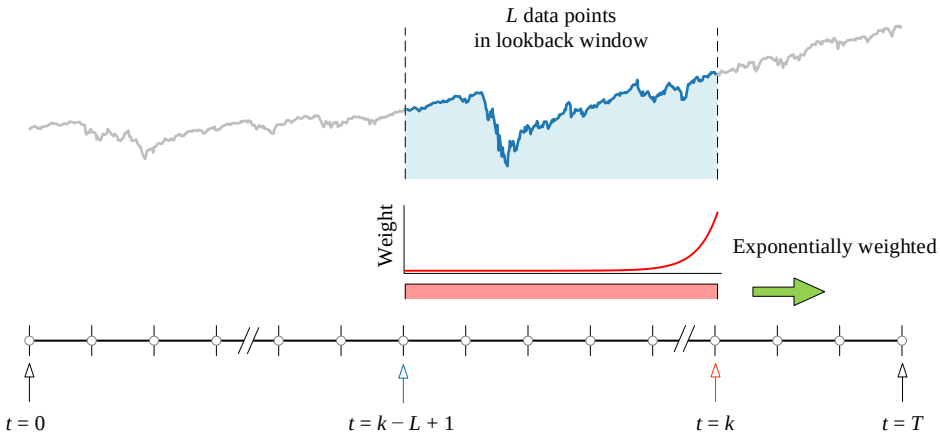


图 15. 回望窗口内数据指数加权移动平均

图 16 所示为 EWMA 权重随衰减系数变化。

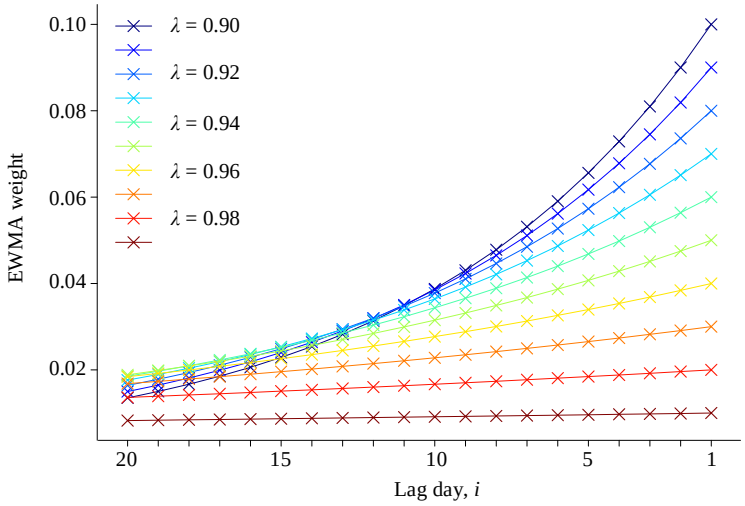


图 16. EWMA 权重随衰减系数变化

EWMA 中**半衰期** (half life, HL) 指的是权重衰减一半的时间，具体定义如下：

$$\lambda^{HL} = \frac{1}{2} \Leftrightarrow HL = \frac{\ln(1/2)}{\ln(\lambda)} \tag{6}$$

图 17 所示为半衰期 HL 随衰减系数 λ 变化。

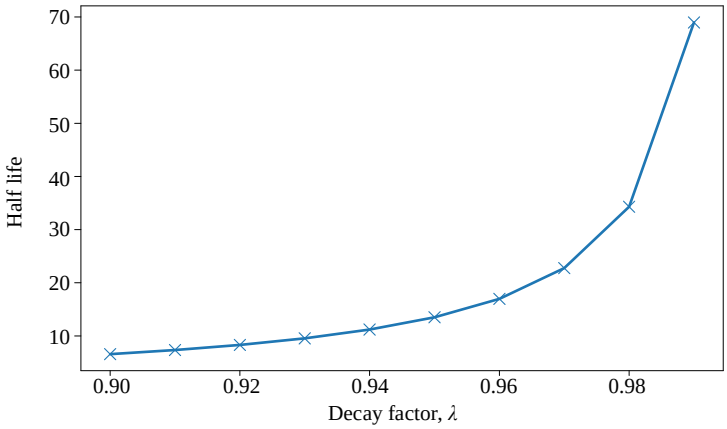


图 17. 半衰期随衰减系数变化

Bk6_Ch07_02.py 绘制图 16 和图 17。

图 18 所示为衰减因子不同条件下，EWMA 平均值变化情况。对比三条曲线，不难发现衰减系数 λ 越小 (比如红线)，EWMA 平均值更贴近真实趋势 (蓝线)，但是平滑度降低。

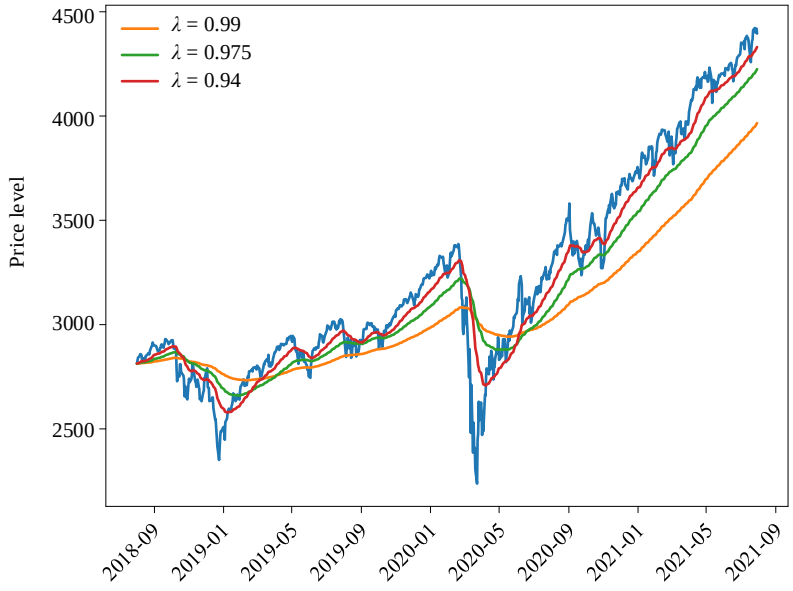


图 18. 指数加权移动平均

给定数据帧数据 df，df.ewm().mean() 可以用来计算指数加权移动平均。这个函数可以还是使用平滑系数 α。衰减因子 λ 与平滑系数 α 有关系如下：

$$\lambda = 1 - \alpha \tag{7}$$

容易得到 α 和半衰期 HL 关系：

$$\alpha = 1 - \exp\left(\frac{\ln(0.5)}{HL}\right) \tag{8}$$

7.4 EWMA 波动率

用 EWMA 方法计算波动率时，常使用如下迭代公式：

$$\sigma_n^2 = \lambda\sigma_{n-1}^2 + (1-\lambda)r_{n-1}^2 \tag{9}$$

其中，λ 为**衰减因子** (decay factor)；σ_n 是当前时刻的波动率；σ_{n-1} 是上一时刻的波动率；r_{n-1} 是上一时刻的回报率。

如下所示，列出四个时间点 n、$n-1$、$n-2$ 和 $n-3$ 的 EWMA 波动率计算式：

$$\begin{cases} \sigma_n^2 = \lambda\sigma_{n-1}^2 + (1-\lambda)r_{n-1}^2 \\ \sigma_{n-1}^2 = \lambda\sigma_{n-2}^2 + (1-\lambda)r_{n-2}^2 \\ \sigma_{n-2}^2 = \lambda\sigma_{n-3}^2 + (1-\lambda)r_{n-3}^2 \\ \sigma_{n-3}^2 = \lambda\sigma_{n-4}^2 + (1-\lambda)r_{n-4}^2 \end{cases} \tag{10}$$

将 (10) 几个算式依次迭代，可以得到：

$$\sigma_n^2 = (1-\lambda)\left(r_{n-1}^2 + \lambda r_{n-2}^2 + \lambda^2 r_{n-3}^2 + \lambda^3 r_{n-4}^2\right) + \lambda^4\sigma_{n-4}^2 \tag{11}$$

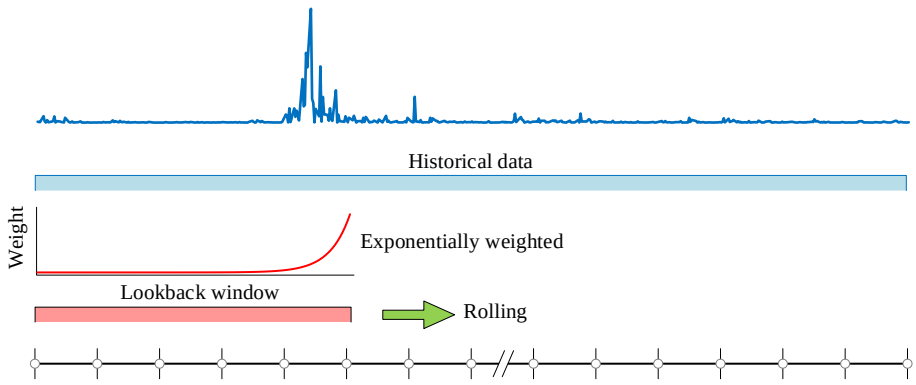


图 19. 指数加权移动平均计算波动率

图 20 所示为不同衰减因子条件下 EWMA 单日波动率。相比 MA 方法，EWMA 可以更快跟踪数据变化。衰减因子越小，跟踪速度越快。

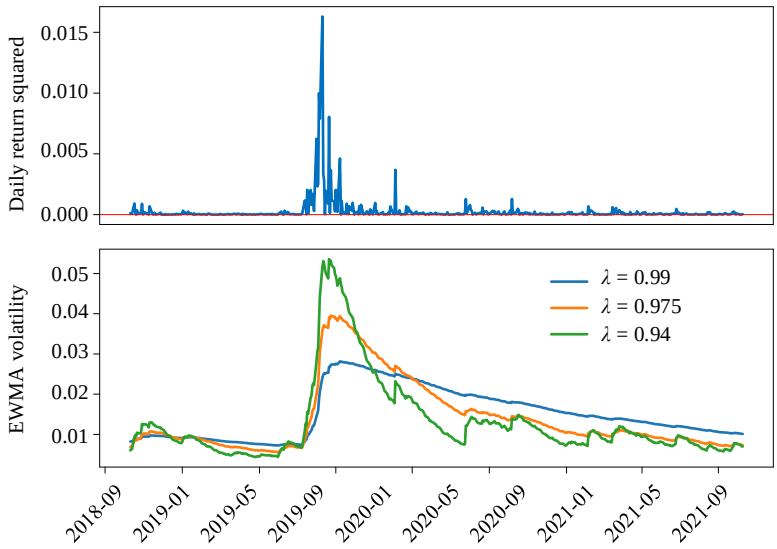


图 20. EWMA 单日波动率，不同衰减因子

图 21~图 23 分别展示衰减因子为 0.99、0.975 和 0.94 的$\pm 2\sigma$移动平均 MA 波动率带宽。

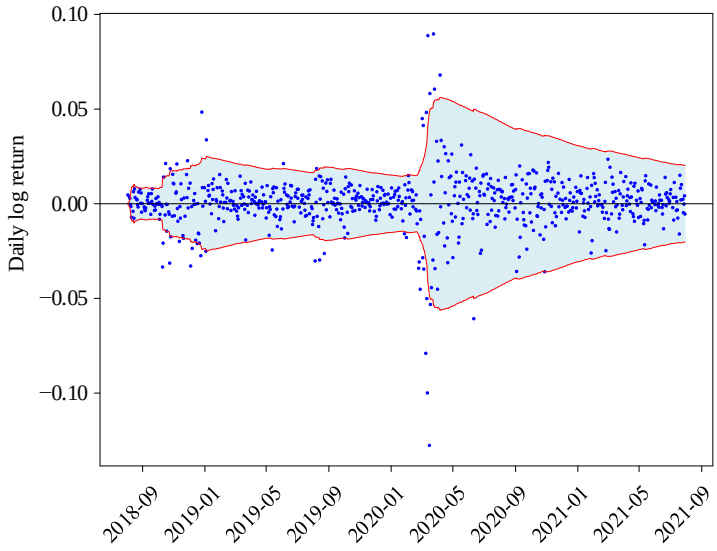


图 21. $\pm 2\sigma$ EWMA 波动率带宽，$\lambda = 0.99$

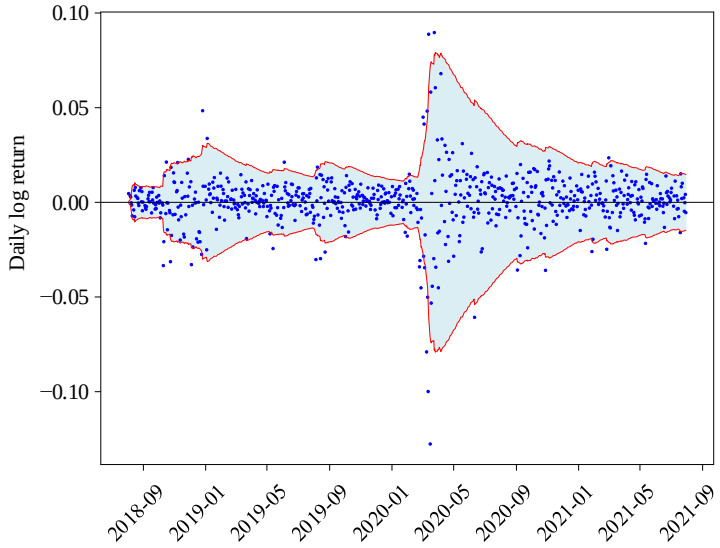


图 22. $\pm 2\sigma$ EWMA 波动率带宽，$\lambda = 0.975$

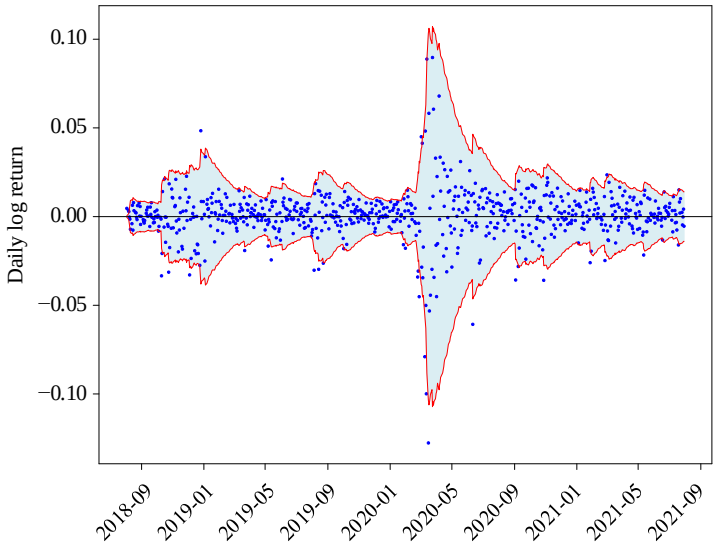


图 23. $\pm 2\sigma$ EWMA 波动率带宽，$\lambda = 0.94$

时间平方根法则将 EWMA 日波动率得到年化波动率。图 24 比较六个年化波动率。

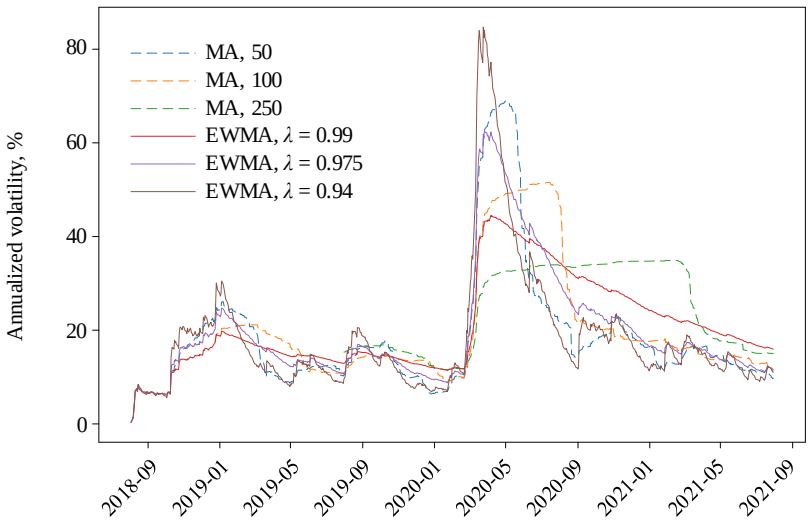


图 24. 比较 6 个年化波动率

7.5 相关性系数

除了平均值、波动率等，相关性系数也随着时间不断变化。df.rolling().corr() 可以计算数据帧 df 的移动相关性。图 25 所示为移动相关性系数。在处理数据时，但凡发现移动相关性系数发生剧烈波动时，都需要大家格外小心。因为移动相关性系数的陡然增大、降低，都可能是由为数不多的几个数据点造成的。而这几个数据点有可能是离群值，值得我们深入探究。

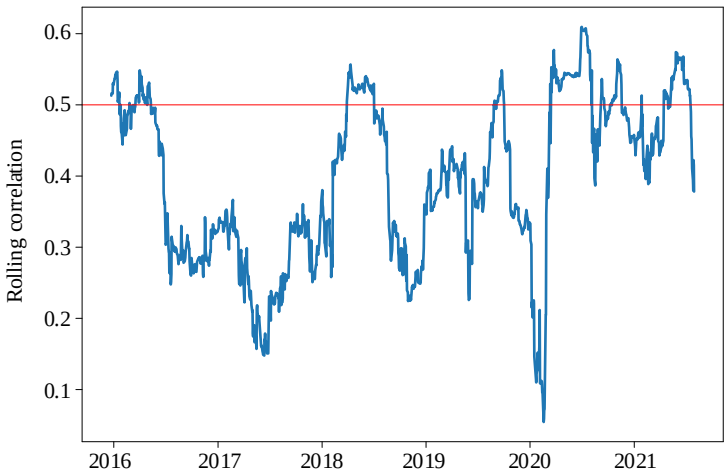


图 25. 移动相关性

Bk6_Ch07_03.py 绘制图 25。

7.6 回归系数

类似地，回归系数也随着移动窗口数据不断变化。

图 26 和图 27 用 statsmodels.regression.rolling.RollingOLS() 计算移动 OLS 线性回归系数。

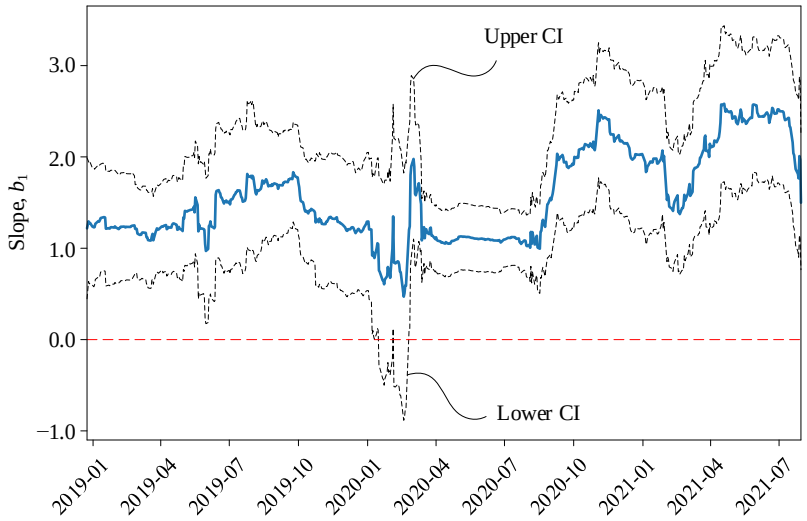


图 26. 回归斜率系数，移动窗口长度 100

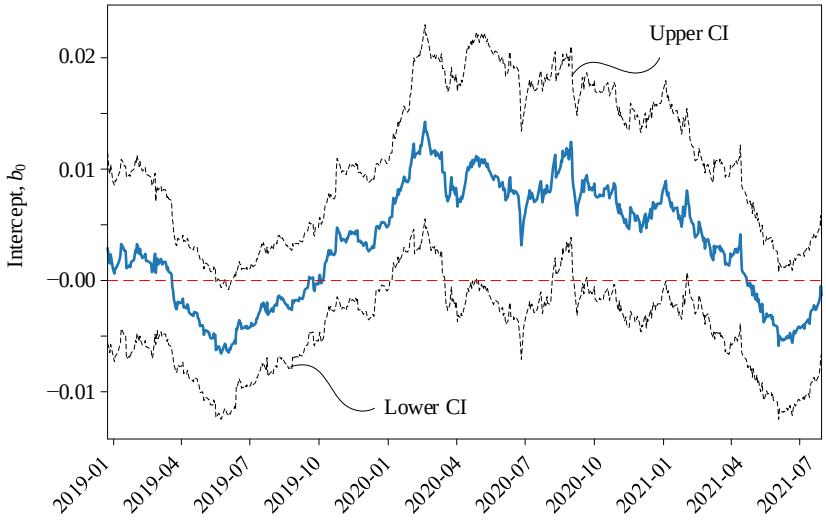
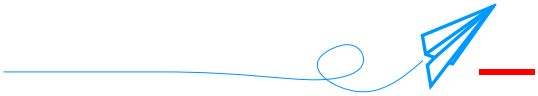


图 27. 回归截距系数，移动窗口长度 100

Bk6_Ch07_04.py 绘制图 26 和图 27。



总结来说，时间序列分析中，移动窗口是一种常用的技术，用于对时间序列数据进行平滑处理和预测分析。通过在时间序列上滑动固定大小的窗口，计算每个窗口中数据点的平均值或加权平均值来平滑数据。简单移动平均法 SMA 是最基本的移动窗口方法，它将窗口内的数据点简单平均处理，对于时间序列的短期波动有较好的平滑效果。移动波动率是指在移动窗口内计算的标准差或方差，它通常用于评估时间序列的波动性。指数加权移动平均法 EWMA 是一种加权移动平均方法，它通过指数函数来计算每个数据点的权重，使得较近期的数据点的权重更大，从而更好地捕捉跟踪到时间序列变化趋势。此外，相关性系数、线性回归系数也都随时间 (移动窗口变化)。

8 随机过程入门

Fundamentals of Stochastic Processes

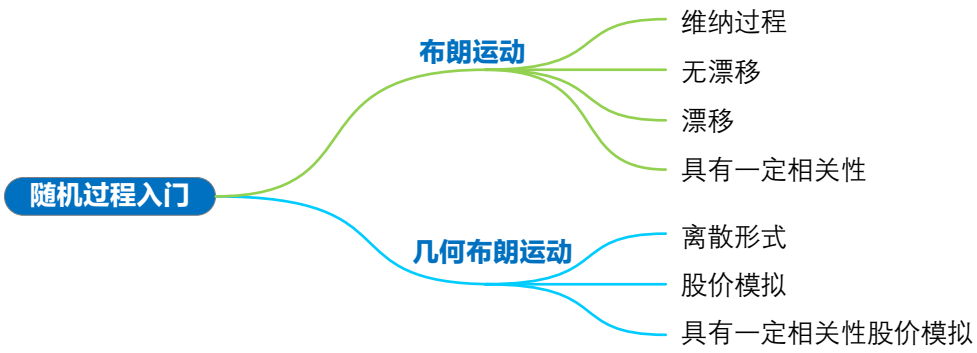
一连串随机事件动态关系的定量描述

不断重复地观察这些运动给我极大的满足；它们并非来自水流，也不是源于水的蒸发，这些运动的源头是颗粒自发的行为。

These motions were such as to satisfy me, after frequently repeated observation, that they arose neither from currents in the fluid, nor from its gradual evaporation, but belonged to the particle itself.

—— 罗伯特·布朗 (Robert Brown) | 英国植物学家 | 1773 ~ 1858

- `matplotlib.patches.Circle()` 绘制正圆
- `np.random.normal()` 产生服从正态分布随机数
- `numpy.cumsum()` 累加
- `numpy.flipud()` 上下翻转矩阵
- `seaborn.distplot()` 绘制频率直方图和 KDE 曲线



8.1 布朗运动：来自花粉颗粒无规则运动

1827 年，英国著名植物学家罗伯特·布朗通过显微镜观察悬浮于水中的花粉，发现花粉颗粒迸裂出的微粒呈现出不规则的运动，后人称之为**布朗运动** (Brownian motion)。这里一个有趣的细节是，实际上花粉自身在水中并没有呈现出布朗运动，而是其崩裂出的微粒。爱因斯坦在 1905 年第一个解释布朗运动现象。

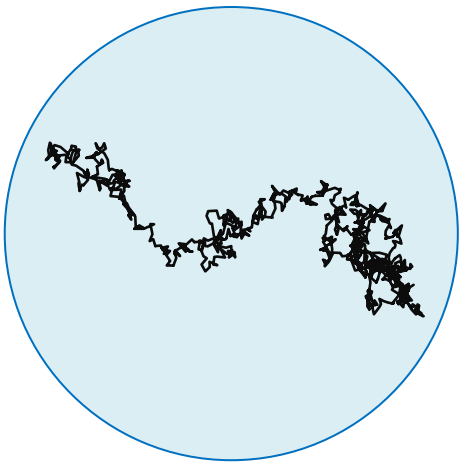


图 1. 平面上的随机运动

罗伯特·布朗 (Robert Brown)
英国植物学家 | 1773 ~ 1858
丛书关键词：● 随机 ● 布朗运动 ● 几何布朗运动 ● 蒙特卡罗模拟

布朗运动定义

如果一个过程满足如下性质，则称 $X(t)$ 为**布朗运动** (Brownian motion)。

过程初始值为 0：

$$X(0)=0 \tag{1}$$

$X(t)$ 几乎处处连续。布朗运动是一种连续时间的运动，其轨迹是连续的，并且其微小变化是连续的。

$X(t)$ 布朗运动的增量是相互独立的，并且服从正态分布。对于所有 $0 \leqslant s < t$，

$$X(t)-X(s) \sim N\left(0,(t-s)\sigma^2\right) \tag{2}$$

对于 $t > 0$，$X(t)$ 是均值为 0，方差为 $\sigma^2 t$ 的正态随机变量。也就是说，$X(t)$ 的密度函数为：

$$f_{X(t)}(x)=\frac{1}{\sqrt{2\pi\sigma^2 t}}\exp\left(\frac{-x^2}{2\sigma^2 t}\right) \tag{3}$$

⚠ 注意，布朗运动在时间上是平稳的，即均值和方差不随时间的推移而改变。此外，布朗运动的变化是由随机因素驱动的，因此变化不可预测。

维纳过程

特别地，如果 $\sigma = 1$，这个过程被称作**标准布朗运动过程** (standard Brownian motion process)，也叫做维纳过程，本章用大写 B 表示。**维纳过程** (Wiener process) 得名于诺伯特·维纳 (Norbert Wiener)。

诺伯特·维纳 (Norbert Wiener)
美国数学家 | 1894 ~ 1964
丛书关键词：● 维纳过程 ● 蒙特卡罗模拟

假设 $t = 0$ 时，$B(0) = 0$，微粒位置在原点处。在 t 时刻，如果 x 为微粒所在位置，对应的概率密度为：

$$f_{B(t)}(x)=\frac{1}{\sqrt{2\pi t}}\exp\left(\frac{-x^2}{2t}\right) \tag{4}$$

$B(t)$ 也可以描述为：

$$B(t)\sim N(0,t) \tag{5}$$

这说明 $B(t)$ 服从均值为 0、方差为 t 的正态分布。图 2 所示为标准差随 t 变化。

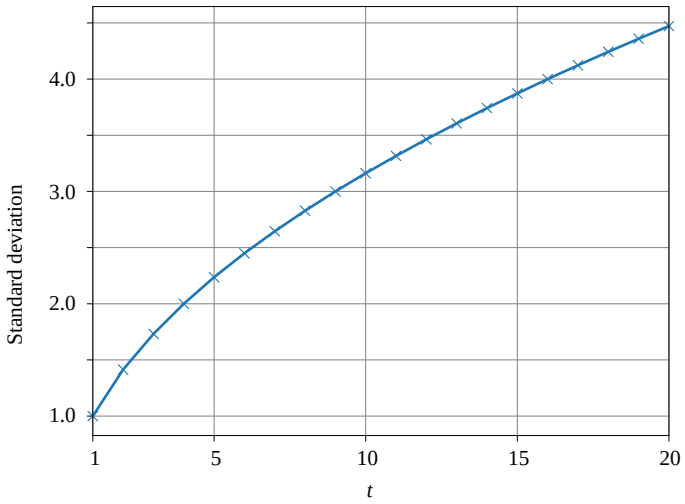


图 2. 维纳过程标准差随时间 t 变化

图 3 所示为 (4) 所示概率密度随 x、t 变化曲面，图中仅仅保留曲线随位置 x 变化曲线。可以这样理解图 3 中的曲线，随着时间不断推移，微粒的运动范围不断扩大。也就是说，随着 t 增大，微粒出现在远离原点的“偏远”位置的可能性增大。

⚠ 注意，图 3 的纵轴是概率密度，不是概率值；但是，概率密度也代表可能性。

如果把视角换成时间 t，我们得到图 4。原点是微粒出发的位置，我们发现随着 t 增大，概率密度值不断减小。这说明微粒位于原点及其附近的可能性随着 t 增大而减小。而远离原点的位置，微粒出现的可能性却随着时间 t 增大而增大。介于其间的位置，概率密度先增大后减小，可以用“涟漪”形容这种现象，微粒从原点汹涌而至，而又倏忽散去，雨散云飞。

图 5 所示为维纳过程概率密度随 x、t 变化等高线。由于维纳过程概率密度函数期望值为 0，大家可以发现当 t 为定值时，概率密度的最大值出现在 $x = 0$ 处。这就是为什么图 5 (b) 的平面等高线关于 $x = 0$ 对称。

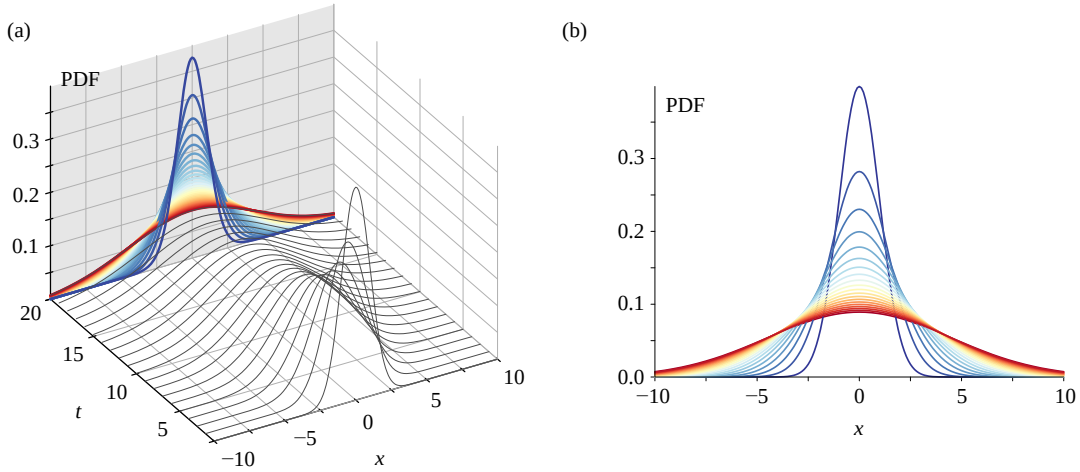


图 3. 维纳过程概率密度曲线随 x 变化，t 快照

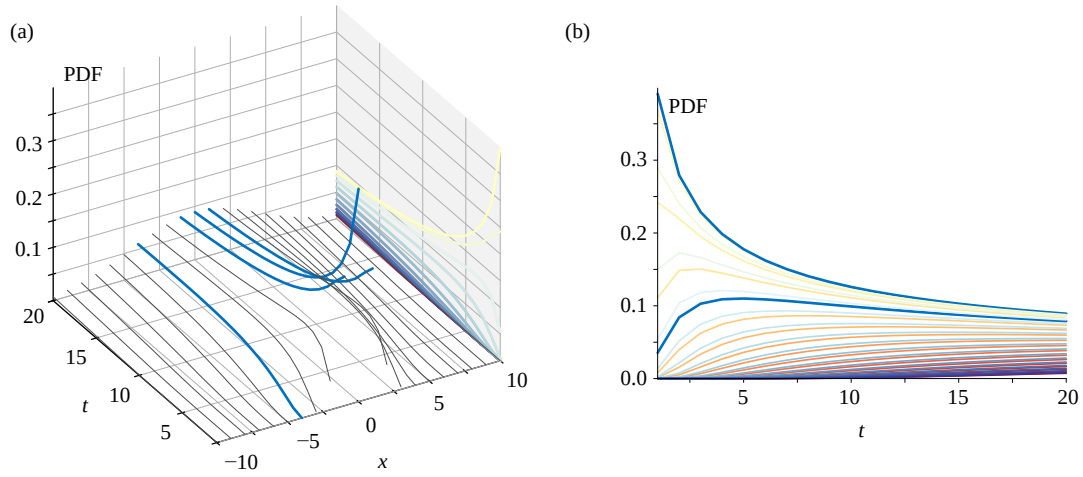


图 4. 维纳过程概率密度曲线随 t 变化，x 快照

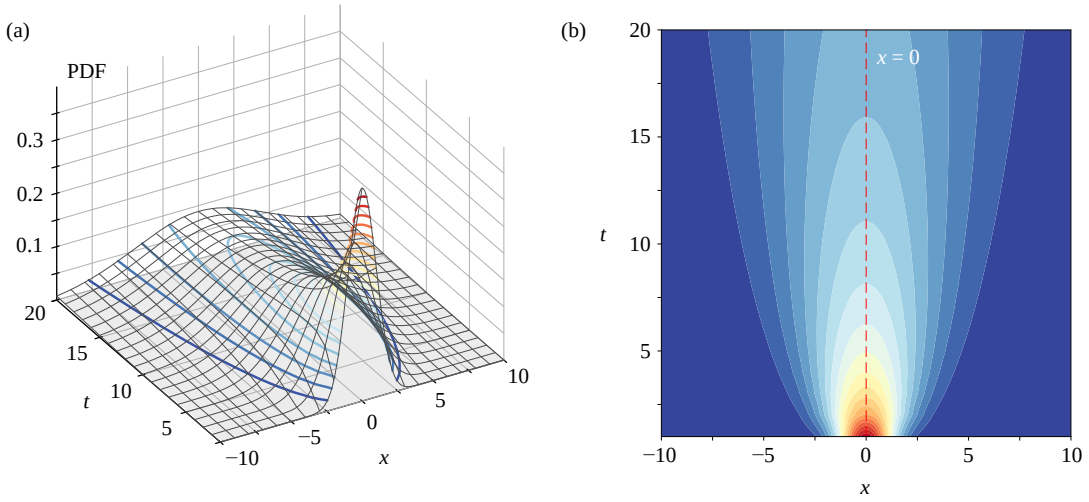


图 5. 维纳过程概率密度随 x、t 变化等高线

Bk6_Ch08_01.py 绘制图 2。请大家自行绘制本节其他图像。

8.2 无漂移布朗运动

一维

无漂移布朗运动 (zero-drift Brownian motion) 和标准布朗运动的关系为：

$$X(t)=\sigma B(t) \tag{6}$$

上式相当于漂移项为 0。漂移项通常是指趋势项，即随机过程的长期趋势。在无漂移布朗运动中，随机游走的漂移项为 0，因此其表现为在一条平均线附近上下波动。

ΔX 为 X 在小段时间 Δt 内位置变化：

$$\Delta X=\varepsilon\sigma\sqrt{\Delta t} \tag{7}$$

其中，随机数 ε 服从标准正态分布 $N(0,1)$，这说明 $X(t)\sim N\left(0,\sigma^2 t\right)$。

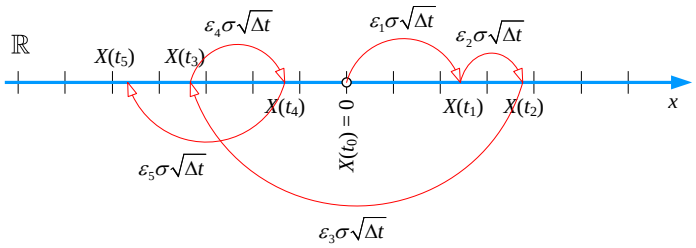


图 6. 某个微粒的一维无漂移布朗运动

在 $t_0=0$ 时刻，微粒的位移 $X(t_0)=0$。如图 6 所示，t_n 时刻，微粒的位移为 $X(t_n)$ 可以写成一系列微小移动之和：

$$\begin{aligned}X(t_n)&=X(t_{n-1})+\Delta X(t_{n-1})\\&=X(t_{n-1})+\varepsilon_n\sigma\sqrt{\Delta t}\\&=X(t_{n-2})+\varepsilon_{n-1}\sigma\sqrt{\Delta t}+\varepsilon_n\sigma\sqrt{\Delta t}\\&\cdots\cdots\\&=X(t_0)+\varepsilon_1\sigma\sqrt{\Delta t}+\varepsilon_2\sigma\sqrt{\Delta t}+\ldots+\varepsilon_{n-1}\sigma\sqrt{\Delta t}+\varepsilon_n\sigma\sqrt{\Delta t}\\&=\sigma\sqrt{\Delta t}\sum_{i=1}^{i=n}\varepsilon_i\end{aligned} \tag{8}$$

其中，$\sqrt{\Delta t}=t_n-t_{n-1}$。

图 7 给出的是 100 个微粒的的 200 步无漂移布朗运动轨迹。这就好比在 $t=0$ 时刻，在数轴原点同时释放 100 个微粒，让它做沿着 x 轴无漂移布朗运动。图 7 右侧直方图为 $t=200$ 时刻，微粒在 x 轴上所处位置的分布。

同时图 7 也绘制出 $\pm\sigma\sqrt{t}$ 和 $\pm2\sigma\sqrt{t}$ 这四条曲线。

图 7 就可以用《统计至简》第 9 章讲过的 68-95-99.7 法则，请大家思考。

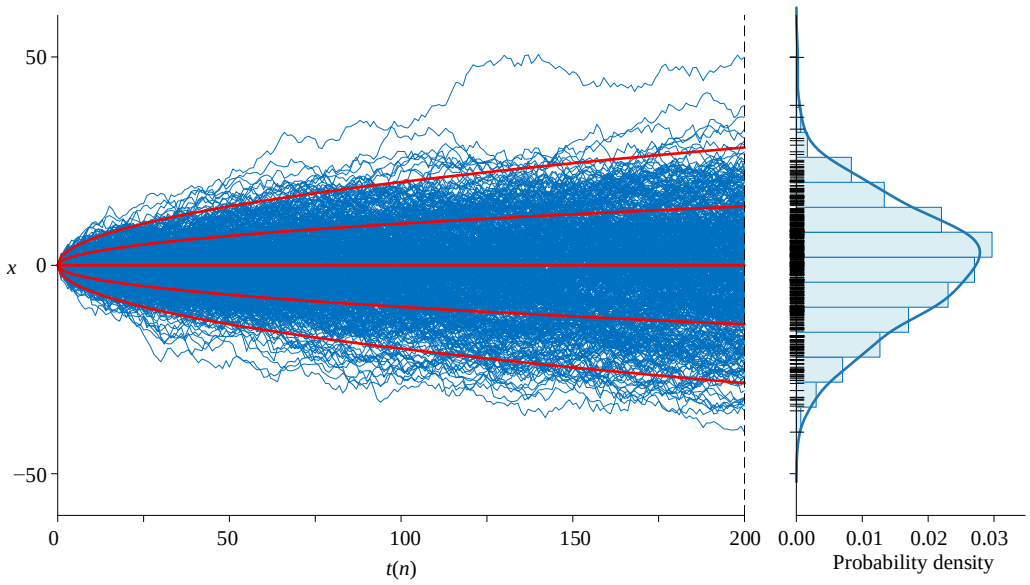


图 7. 100 个微粒一维无漂移布朗运动轨迹和运动范围

图 7 的每一个微粒随机漫步的路径，都是不同的。换句话说，任意两个微粒的运动轨迹相同的概率几乎为零。

图 8 所示为微粒在不同 t 在 x 轴上分布的快照，图中我们也可以看到 68-95-99.7 法则。

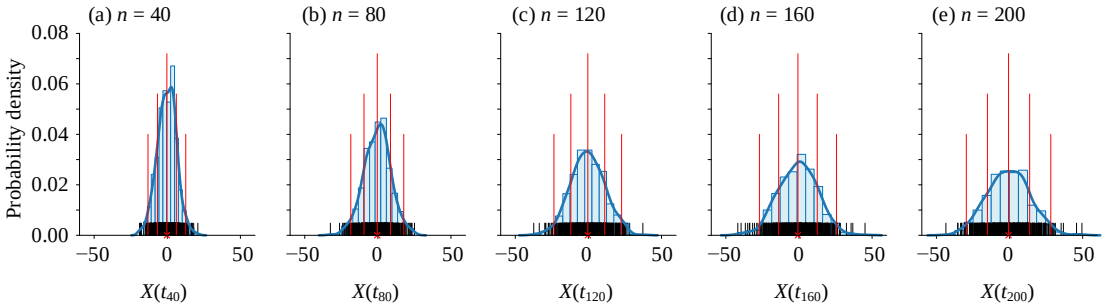


图 8. 100 个微粒无漂移布朗运动轨迹在不同时刻位置分布的快照

Bk6_Ch08_02.py 绘制图 7 和图 8。

二维

在二维平面里，微粒的随机漫步更像布朗运动中炸裂的花粉颗粒一样。在 t_n 时刻，$X(t_n)$ 为微粒的横坐标值，$Y(t_n)$ 为微粒的纵坐标值：

$$\begin{cases} X(t_n) = \sigma\sqrt{\Delta t}\sum_{i=1}^{i=n}\varepsilon_i \\ Y(t_n) = \sigma\sqrt{\Delta t}\sum_{j=1}^{j=n}\varepsilon_j \end{cases} \tag{9}$$

图 9 所示为某个微粒从原点出发完全的二维无漂移布朗运动，运动过程显得“浑浑噩噩”、“生无可恋”。

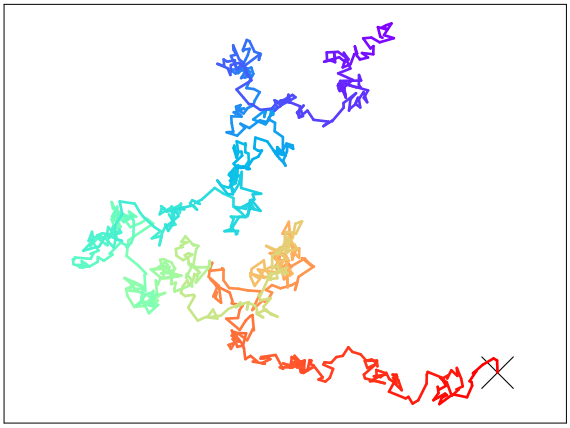


图 9. 平面二维无漂移随机漫步

Bk6_Ch08_03.py 绘制图 9。

8.3 漂移布朗运动：确定 + 随机

前面介绍了零漂移布朗运动，微粒的运动只具有随机成分，而没有确定成分。如果在零漂移布朗运动基础上，引入确定成分，我们便得到**漂移布朗运动** (Brownian motion with drift)：

$$X(t) = \underbrace{\mu t}_{\text{Drift}} + \underbrace{\sigma B(t)}_{\text{Random}} \tag{10}$$

其中，μ 为漂移率，σ 为标准差。这说明 $X(t) \sim N\left(\mu t, \sigma^2 t\right)$。

如果把上式看做是物体直线运动的话，μt 相当于是匀速运动部分，也就是漂移，确定的成分。如图 10 所示，漂移率 μ 可以为正，可以为负，当然也可以为 0 (无漂移)。

$\sigma B(t)$ 相当于随机漫步，可以理解为噪音，即随机成分，代表不确定性。

打个比方，μt 就是浩浩汤汤的历史进程，大势所趋。$\sigma B(t)$ 就是时时刻刻的生活细节，琐碎繁杂。

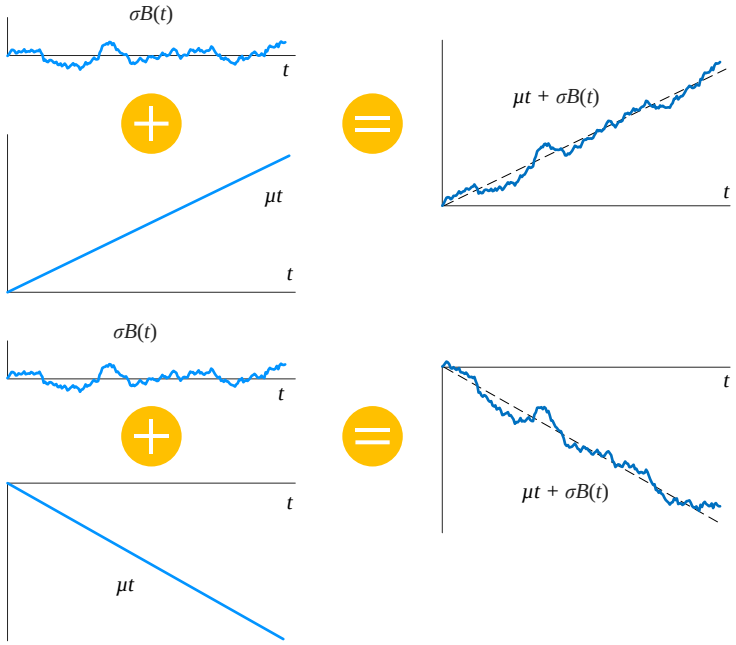


图 10. 解构定向漂移布朗运动

图 11 所示为漂移布朗运动概率密度随 x、t 变化曲面。类似图 3，图 11 中仅仅保留曲线随位置 x 变化曲线。类似无漂移布朗运动，随着时间不断推移，漂移布朗运动微粒的运动范围不断扩大。同时，我们能够看到概率密度的对称轴随着时间增大而移动。

图 12 所示为含漂移布朗运动概率密度曲线随 t 变化，在不同 x 点上的快照。

图 13 所示为含漂移布朗运动概率密度随 x、t 变化等高线，图中能够明显地看到 (10) 漂移项。

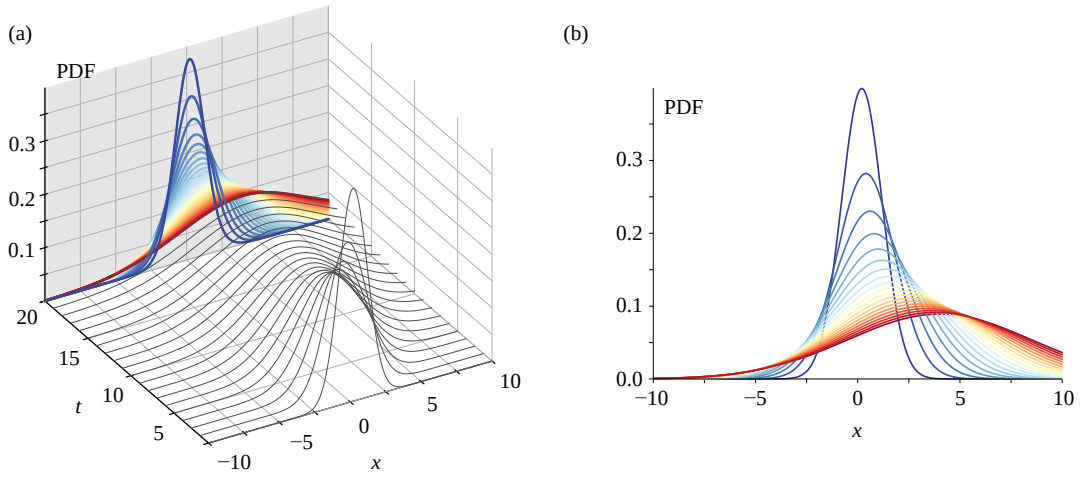


图 11. 漂移布朗运动概率密度曲线随 x 变化，t 快照

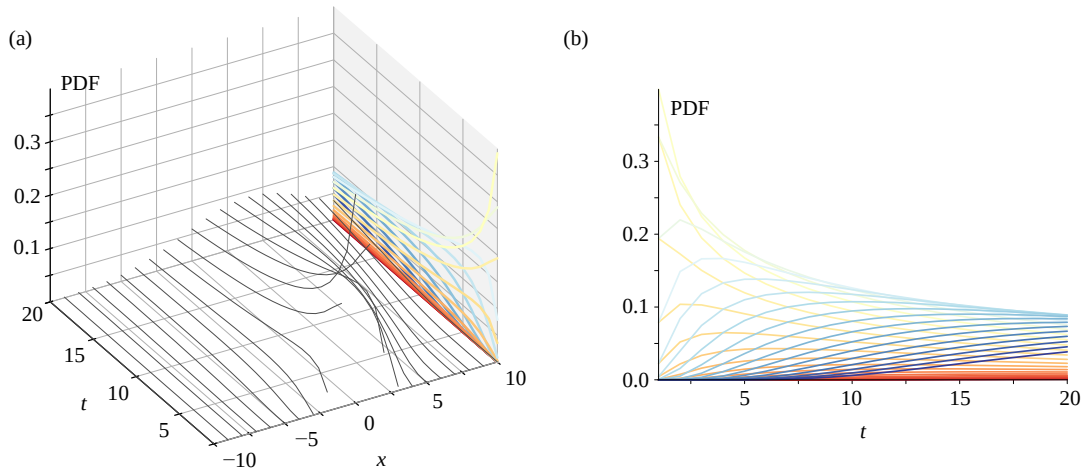


图 12. 含漂移布朗运动概率密度曲线随 t 变化，x 快照

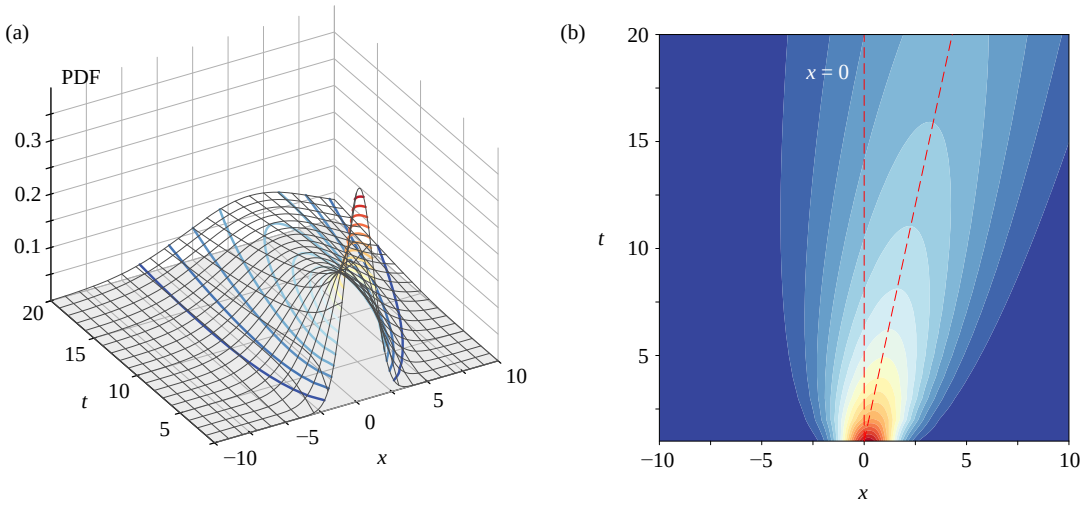


图 13. 含漂移布朗过程概率密度随 x、t 变化等高线

离散形式

为了方便蒙特卡洛模拟，我们也需要得到含漂移布朗过程的离散形式。

首先，写出 (10) 的微分形式：

$$\mathrm{d}X(t) = \mu \mathrm{d}t + \sigma \mathrm{d}B(t) \tag{11}$$

这样，(10) 的离散化形式可以写成：

$$\Delta X(t) = \mu \cdot \Delta t + \sigma \sqrt{\Delta t} \cdot \varepsilon \tag{12}$$

然后，把上式写成累加形式：

$$X\left(t_n\right)=\mu\cdot n\Delta t+\sigma\sqrt{\Delta t}\sum_{i=1}^{i=n}\varepsilon_i \tag{13}$$

图 14 给出的是 100 个微粒的的 200 步含漂移布朗运动轨迹。能够明显地看到运动轨迹“整体”表现出“向上”的运动趋势，这来自于定向漂移成分 μt。此外，这些轨迹在时间 t 处的期望值就是 μt 。图 14 右侧直方图为 t = 200 时刻，微粒在 x 轴上所处位置的分布。

图 7 也绘制出 $\mu t\pm\sigma\sqrt{t}$ 和 $\mu t\pm 2\sigma\sqrt{t}$ 这四条曲线。图 15 所示为微粒在不同 t 在 x 轴上分布的快照，图中我们也可以看到 68-95-99.7 法则。

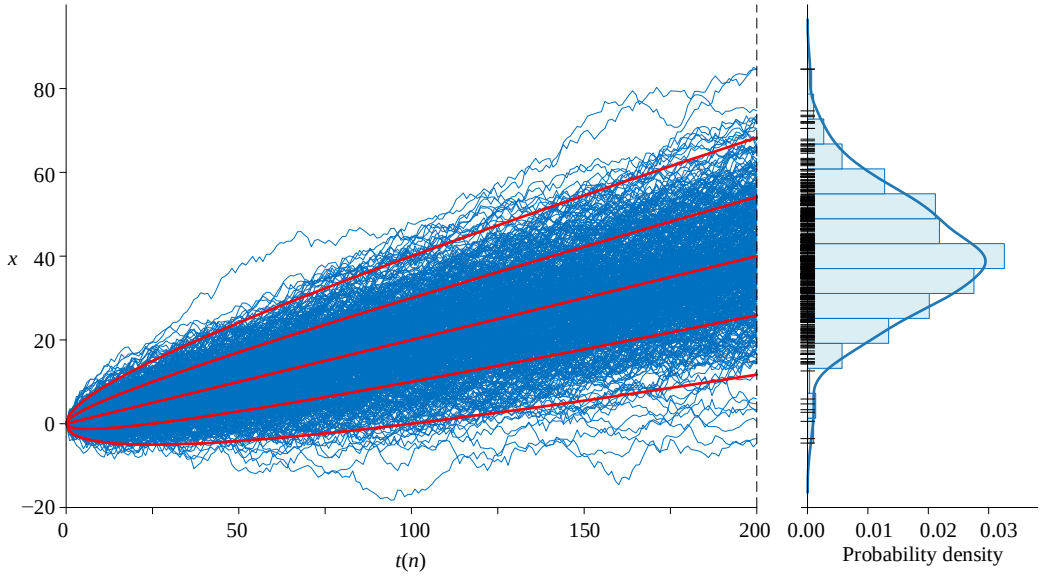


图 14. 100 个微粒一维含漂移布朗运动轨迹和运动范围

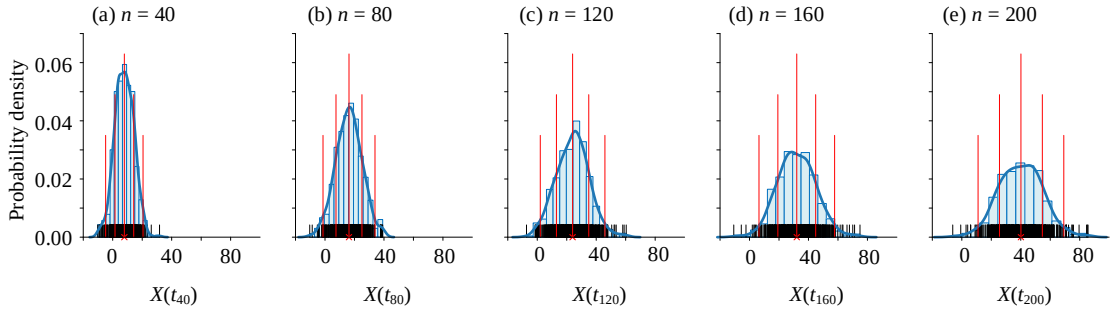


图 15. 100 个微粒含漂移布朗运动轨迹在不同时刻位置分布的快照

Bk6_Ch08_04.py 绘制图 14 和图 15。

8.4 具有一定相关性布朗运动

本节介绍如何据此产生满足一定相关性的布朗运动。

《统计至简》第 15 章介绍如何产生具有一定相关性的随机数，请大家回顾。

如图 16 所示，给定固定时间间隔 Δt，$\Delta \boldsymbol{X}(t)$ 为在 Δt 满足一定相关性布朗运动分步步长构成的矩阵为：

$$\Delta \boldsymbol{X}(t) = \mathrm{E}(\boldsymbol{X})\Delta t + \boldsymbol{Z}\boldsymbol{R}\sqrt{\Delta t} \tag{14}$$

也就是说，$\boldsymbol{X}(t) \sim N(\mathrm{E}(\boldsymbol{X})t, \boldsymbol{\Sigma} t)$。而是 $\boldsymbol{R}$ 是 $\boldsymbol{\Sigma}$ 的 Cholesky 分解的三角矩阵。图 16 中，矩阵 $\boldsymbol{Z}$ 为随机数矩阵，服从 $N(0, \boldsymbol{I})$。

图 17、图 18 所示为具有正相关的两条漂移布朗运动蒙特卡洛模拟结果。图 19、图 20 所示为具有负相关的两条漂移布朗运动蒙特卡洛模拟结果。

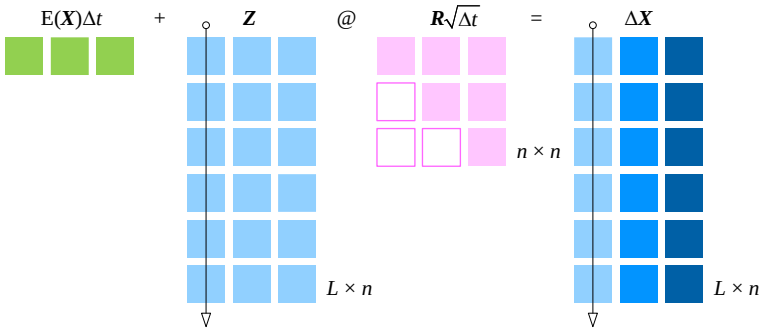


图 16. 计算具有一定相关性布朗运动矩阵运算

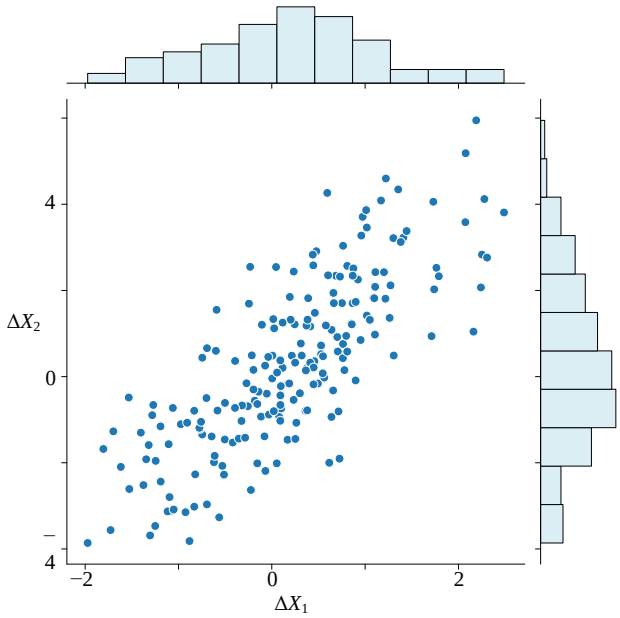


图 17. 分步步长的散点图，$\rho = 0.8$

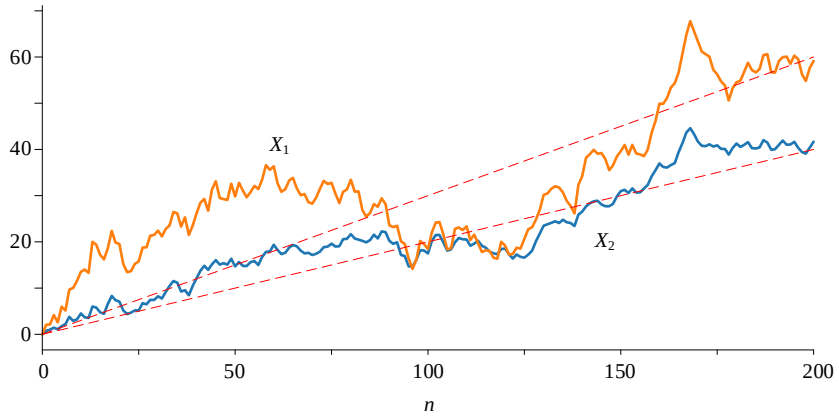


图 18. 两条具有正相关关系的行走轨迹，$\rho = 0.8$

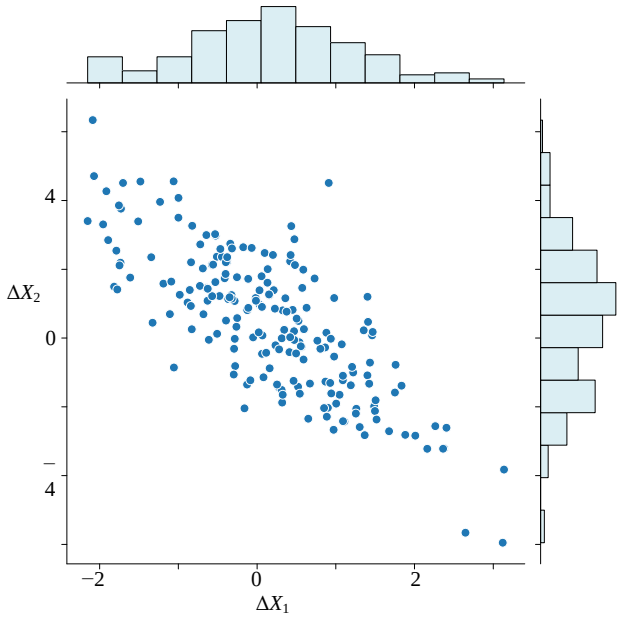


图 19. 分步步长的散点图，$\rho = -0.8$

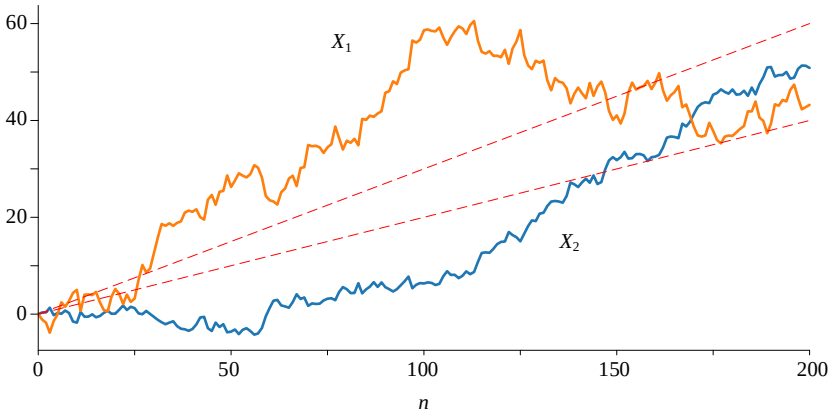


图 20. 两条具有正相关关系的行走轨迹，$\rho = -0.8$

Bk6_Ch08_05.py 绘制图 17 ~ 图 20。

8.5 几何布朗运动

满足下式的随机微分方程的过程，被称作**几何布朗运动** (Geometric Brownian motion, GBM)：

$$\mathrm{d}X(t) = \mu X(t)\mathrm{d}t + \sigma X(t)\mathrm{d}B(t) \tag{15}$$

其中，$X(t) > 0$。

上式也可以写成：

$$\frac{\mathrm{d}X(t)}{X(t)} = \mu \mathrm{d}t + \sigma \mathrm{d}B(t) \tag{16}$$

利用**伊藤引理** (Ito's Lemma)，求解得到 $X(t)$：

$$X(t) = X(0)\exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma B(t)\right) \tag{17}$$

伊藤引理是随机微积分的重要定理之一，用于计算随机过程的微分。简单来说，伊藤引理是泰勒展开在随机微积分中的应用，它是通过将泰勒展开的思想推广到随机微积分中，得到了一般的随机微分方程的解析式。泰勒展开是将一个函数展开成一个多项式的形式，而伊藤引理是将一个随机过程展开成一个多项式 (一般为二阶) 加一个随机项的形式。

《数学要素》第 17 章介绍过泰勒展开，请大家回顾。

$X(t)$ 的期望值为：

$$\mathrm{E}\left(X(t)\right) = X(0)\exp(\mu t) \tag{18}$$

$X(t)$ 的方差：

$$\mathrm{var}\left(X(t)\right) = X(0)^2 \exp(2\mu t)\left(\exp\left(\sigma^2 t\right) - 1\right) \tag{19}$$

$X(t)$ 的标准差为：

$$\mathrm{std}\left(X(t)\right) = X(0)\exp(\mu t)\sqrt{\exp\left(\sigma^2 t\right) - 1} \tag{20}$$

对 $X(t)$ 求对数得到：

$$\begin{aligned} \ln X(t) &= \ln\left(X(0)\exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma B(t)\right)\right) \\ &= \ln X(0) + \left(\mu - \frac{\sigma^2}{2}\right)t + \sigma B(t) \end{aligned} \tag{21}$$

可以发现 $\ln X(t)$ 为布朗运动，也就是说 $\ln X(t)$ 的概率密度服从高斯分布。

离散形式

(21) 的离散形式为：

$$\ln\left(X(t+\Delta t)\right) - \ln\left(X(t)\right) = \left(\mu - \frac{\sigma^2}{2}\right)\Delta t + \sigma\varepsilon\sqrt{\Delta t} \tag{22}$$

有了上式，我们就可以进行蒙特卡洛模拟。图 21 所示为 100 个微粒几何布朗运动轨迹。图 22 所示为微粒在不同时刻位置分布的快照。

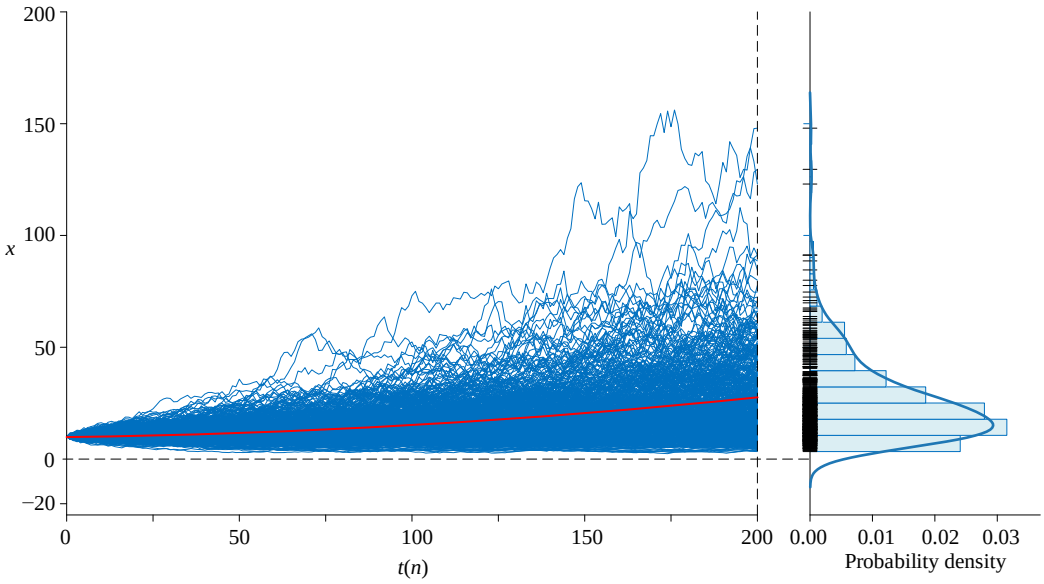


图 21. 100 个微粒几何布朗运动轨迹

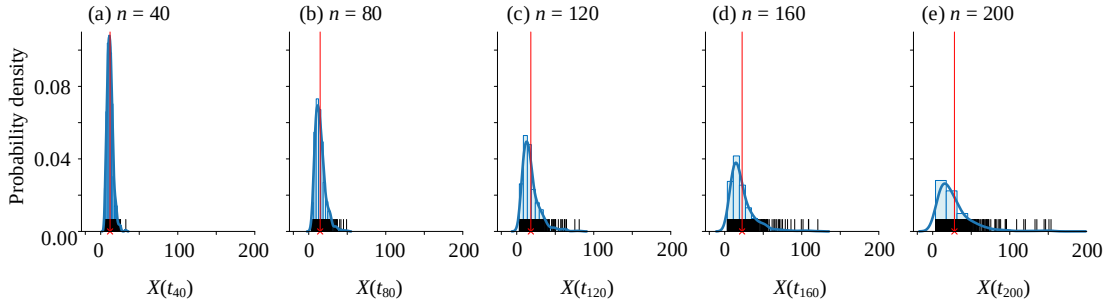


图 22. 100 个微粒几何布朗运动轨迹在不同时刻位置分布的快照

Bk6_Ch08_06.py 绘制图 21 和图 22。

模拟股票股价走势

实践中，几何布朗运动常用来模拟股票股价走势。如图 23 所示，长期观察股票股价，可以发现走势，而且股价不能为负值。更重要的是，股价对数收益率分布可以用高斯分布来描述。

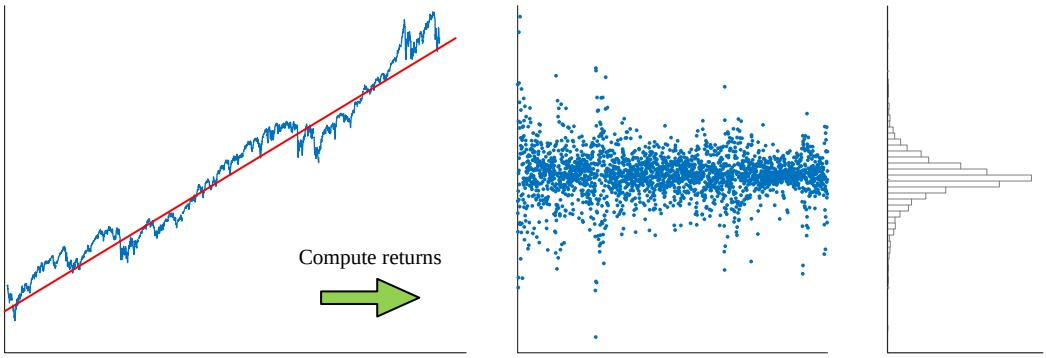


图 23. 某只股价走势、收益率

8.6 股价模拟

S_0 为初始股价，经过一小段时间 Δt，股价变化 ΔS：

$$\Delta S = S_0 \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)\Delta t + \sigma\varepsilon\sqrt{\Delta t}\right) \tag{23}$$

μ 收益率期望值，σ 为收益率波动率，ε 随机数服从标准正态分布。图 24 总结整个蒙特卡洛模拟股价走势过程。历史数据用来校准模型。图 25 所示为 S&P 500 指数在一段时间内的走势。图 26 所示为其日对数回报率。图 27 给出日对数回报率的分布情况，我们可以计算得到均值和方差，这些参数可以用来校准模型。图 28 所示为蒙特卡洛模拟结果。

这种方法缺陷很明显，历史数据未必能够代表未来趋势。此外，由于假设回报率服从正态分布，没有考虑到“厚尾”问题，也就是所谓的“黑天鹅”问题。

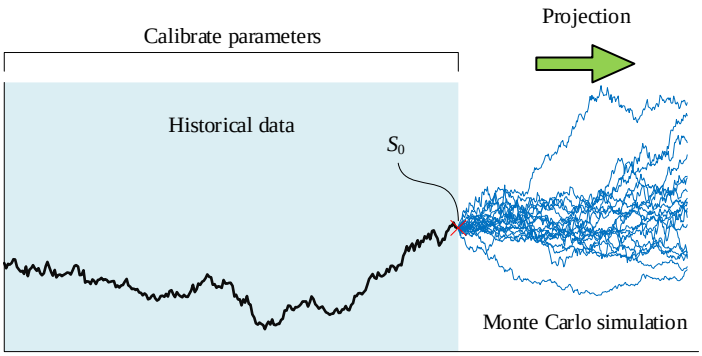


图 24. 基于历史数据估计参数，和蒙特卡洛模拟预测未来股价可能走势

图 25. S&P 500 价格水平数据

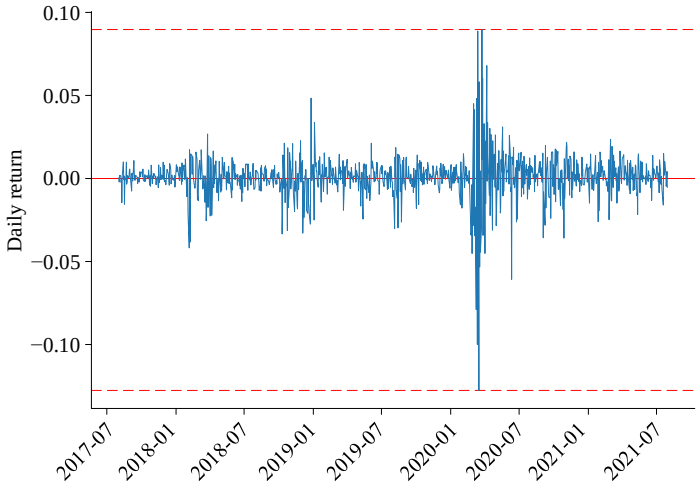


图 26. S&P 500 日对数回报率

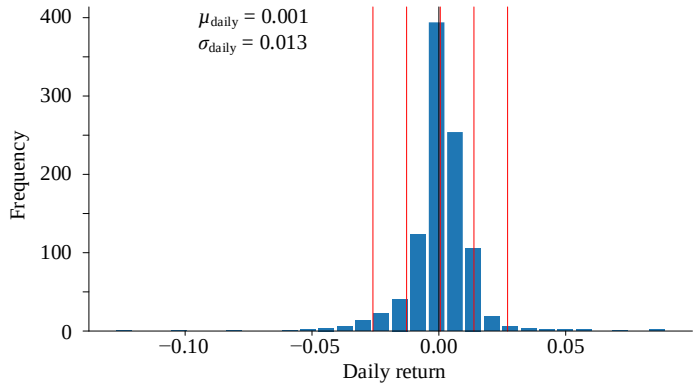


图 27. S&P 500 日对数回报率分布

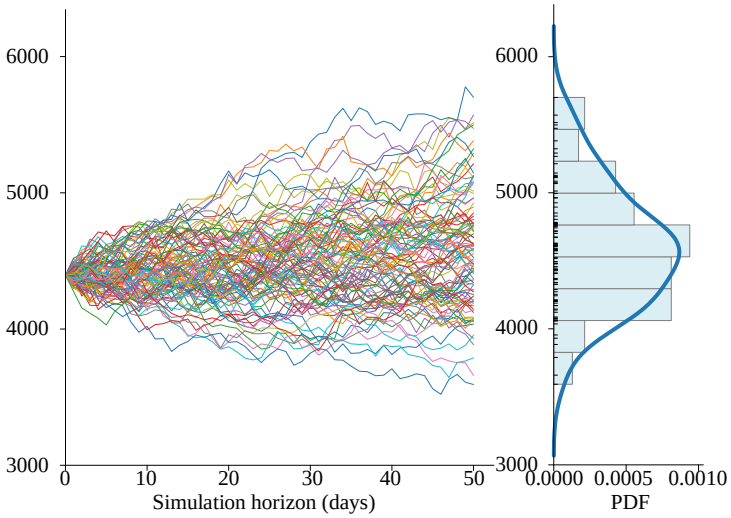


图 28. S&P 500 蒙特卡洛模拟

此外，图 29 所示的二叉树也可以用来模拟股票股价，本书不做展开。

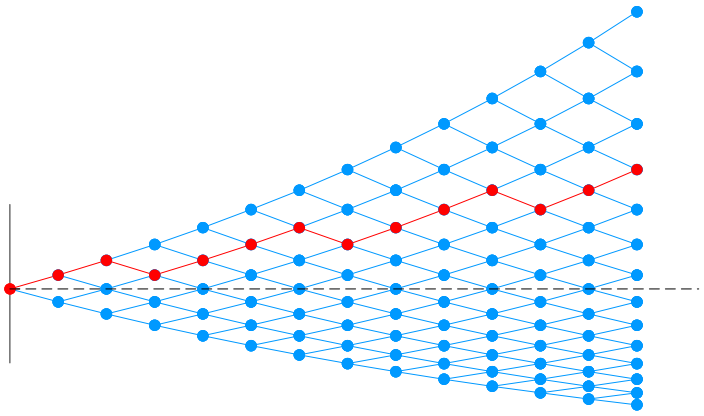


图 29. 二叉树随机路径模拟股票股价

Bk6_Ch08_07.py 绘制图 25 ~ 图 28。

8.7 相关股价模拟

当时间戳为列方向时，下式为几何布朗过程计算对数回报率矩阵 $\boldsymbol{X}$ 矩阵运算式：

$$\boldsymbol{X} = \left(\boldsymbol{\mu} - \frac{\left(\text{diag}\left(\boldsymbol{\Sigma}\right)\right)^{\mathrm{T}}}{2} \right) \Delta t + \boldsymbol{Z}\boldsymbol{R}\sqrt{\Delta t}$$

图 30 所示为上式矩阵运算过程。$\boldsymbol{\mu}$ 为股价年化期望收益率行向量。$\boldsymbol{\Sigma}$ 为化方差协方差矩阵。$\boldsymbol{Z}$ 是由随机数发生器产生的服从标准正态分布的线性无关随机数，$\boldsymbol{Z}$ 为列方向数据矩阵，每列代表一个变量；上三角矩阵 $\boldsymbol{R}$ 来自 Cholesky 分解 $\boldsymbol{\Sigma}$ 得到。Δt 设定为 1/252。

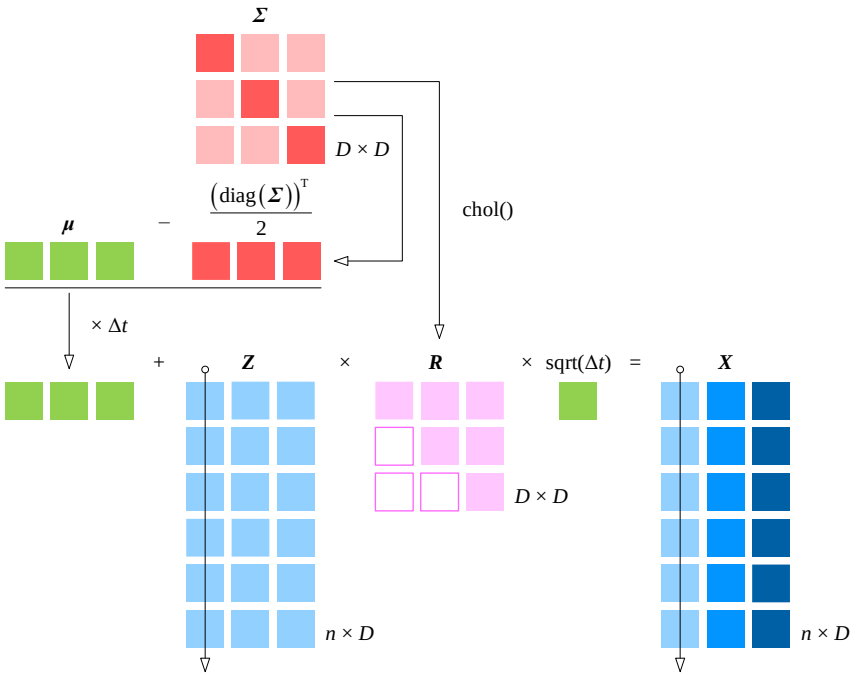


图 30. 几何布朗过程离散式的矩阵运算过程，列方向矩阵

模拟多路径相关股价走势具体矩阵运算过程如图 31 所示，其中矩阵 $\boldsymbol{Z}$ 和矩阵 $\boldsymbol{X}$ 的形状为 $n \times D \times n_{paths}$。$n_{paths}$ 为蒙特卡罗模拟轨迹的数量。

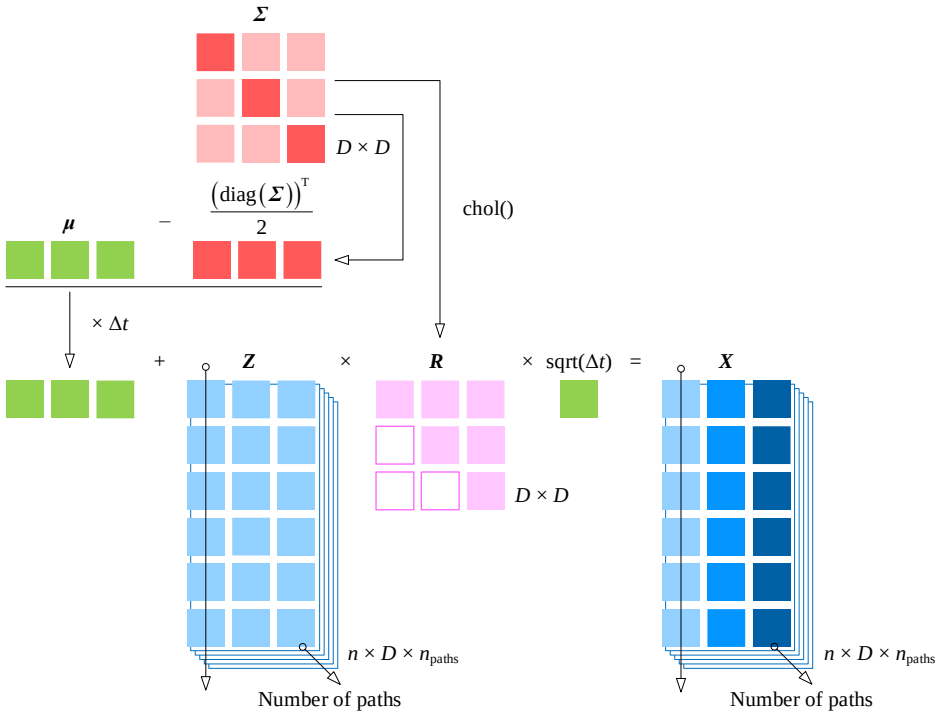


图 31. 几何布朗过程离散式的矩阵运算过程，多路径

图 32 所示为几只股票真实股价和归一化股价走势图。图 33 所示为日收益率的协方差矩阵、相关性系数矩阵。图 34 所示为协方差矩阵的 Cholesky 分解。图 35 所示为一组相关性股价的模拟。这种模拟方法的显著缺点是 Cholesky 分解，当协方差矩阵过大 Cholesky 分解可能会不稳定。此外，只有正定矩阵才能 Cholesky 分解。大家如果感兴趣可以搜索 Benson-Zangari 蒙特卡洛模拟，这种方法避免 Cholesky 分解，本书不展开讲解。

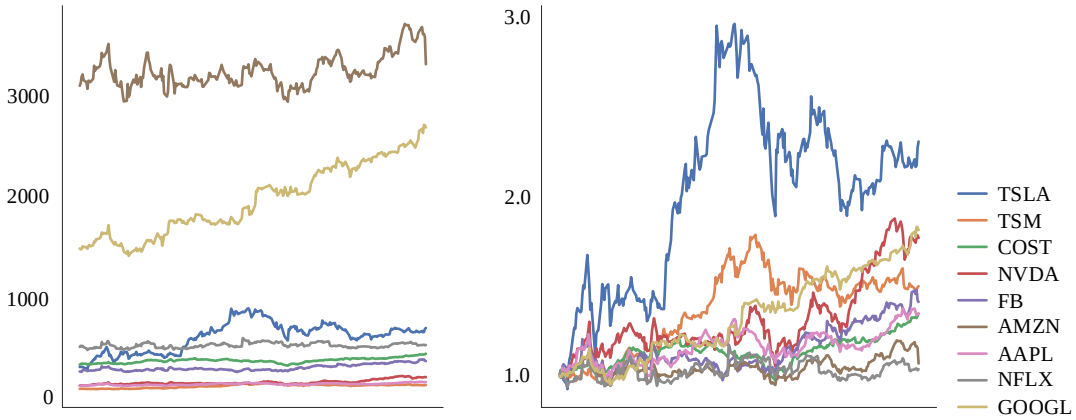


图 32. 几只股票走势和初值归一化股价

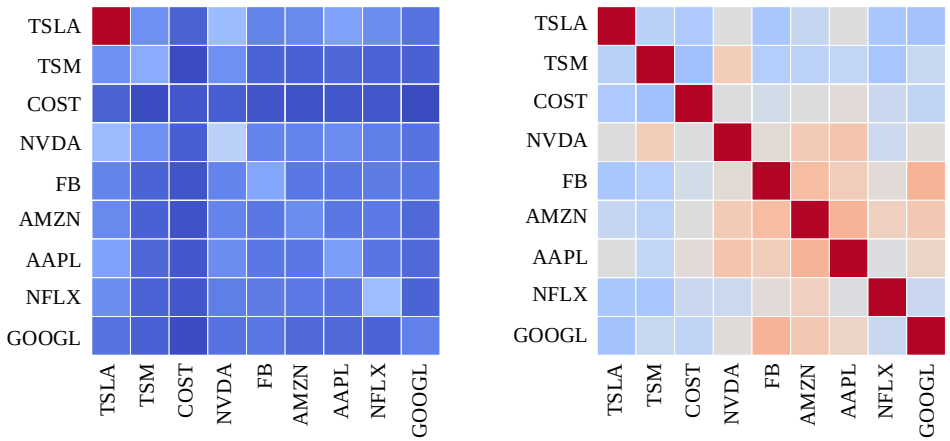


图 33. 协方差矩阵和相关性系数矩阵热图

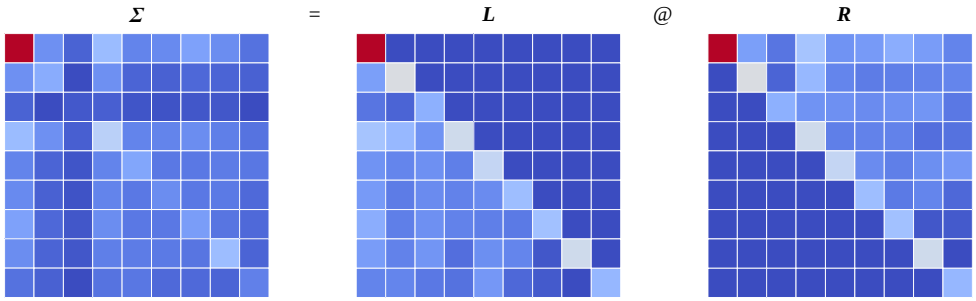


图 34. 对协方差矩阵进行 Cholesky 分解

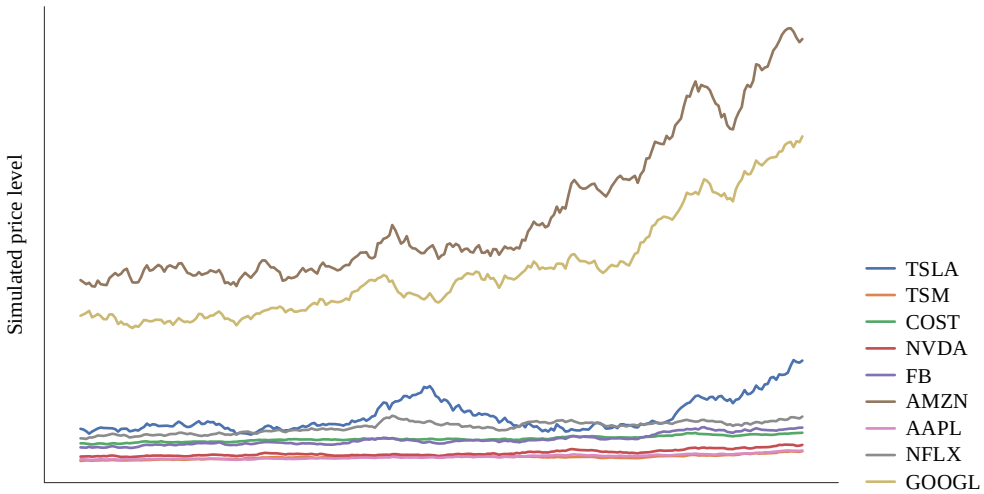
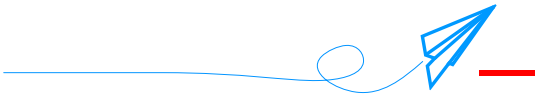


图 35. 一组蒙特卡罗模拟相关性股价结果



随机是指在一定的概率分布下，不确定的事件或过程。随机过程是指随机变量随时间变化的过程。布朗运动是一种最基本的连续时间随机过程，它是随机微积分的基础，因其随机性而具有广泛应用，如金融领域的股价预测、自然界中颗粒的扩散行为等。维纳过程也称标准布朗运动过程。几何布朗运动中，随机变量的对数服从布朗运动。在金融学中，股价往往被认为是一种几何布朗运动。本章介绍如何利用几何布朗运动单一模拟股价走势，以及具有特定相关性股价走势。

9 回归分析

Regression Analysis

线性回归结果不能拿来就用

真理太复杂了，除了近似，我们别无他法。

Truth is much too complicated to allow anything but approximations.

—— 约翰·冯·诺伊曼 (John von Neumann) | 美国籍数学家 | 1903 ~ 1957

- `scipy.stats.kurtosis()` 计算峰度
- `scipy.stats.normaltest()` Omnibus 正态检验
- `scipy.stats.skew()` 计算偏度
- `scipy.stats.t.ppf()` 求解 t 分布的逆累积分布函数
- `scipy.stats.t.sf()` 求解 t 分布的互补累积分布函数 CCDF = 1 - CDF
- `seaborn.distplot()` 绘制直方图，叠合 KDE 曲线
- `seaborn.pairplot()` 绘制成对分析图
- `seaborn.regplot()` 绘制回归图像
- `statsmodels.api.add_constant()` 线性回归增加一列常数 1
- `statsmodels.api.OLS()` 最小二乘法函数
- `statsmodels.graphics.tsaplots.plot_acf()` 绘制自相关结果
- `statsmodels.stats.anova.anova_lm` 获得 ANOVA 表格

9.1 线性回归：一个表格、一条直线

一个表格

大家是否还记得我们在《统计力量》第 24 章结尾给出过图 1 这个表格。

图 1 这个表格汇总某个线性回归分析的结果。本章的主要目的就是和大家理解这个表格各项数值的含义。下面首先介绍这个表格具体来自哪个线性回归。

OLS Regression Results

Dep. Variable:	AAPL	R-squared:	0.687
Model:	OLS	Adj. R-squared:	0.686
Method:	Least Squares	F-statistic:	549.7
Date:	XXXXXXXXXXX	Prob (F-statistic):	4.55e-65
Time:	XXXXXXXXXXX	Log-Likelihood:	678.03
No. Observations:	252	AIC:	-1352.
Df Residuals:	250	BIC:	-1345.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.0018	0.001	1.759	0.080	-0.000	0.004
SP500	1.1225	0.048	23.446	0.000	1.028	1.217

Omnibus:	52.424	Durbin-Watson:	1.864
Prob(Omnibus):	0.000	Jarque-Bera (JB):	210.803
Skew:	0.777	Prob(JB):	1.68e-46
Kurtosis:	7.203	Cond. No.	46.1

图 1. 一元线性回归结果

一条直线

图 2 所示为这个一元 OLS 线性回归的自变量、因变量散点数据以及分布特征。自变量为一段时间内标普 500 股票指数日收益率，因变量为某只特定股票的同期日收益率。观察散点图，我们可以发现明显的“线性”关系。

从金融角度，股指可以“解释”同一个市场上股票的涨跌。图 1 是利用 statsmodels.api.OLS() 函数构造的线性模型结果。

再次强调，线性回归不代表“因果关系”。

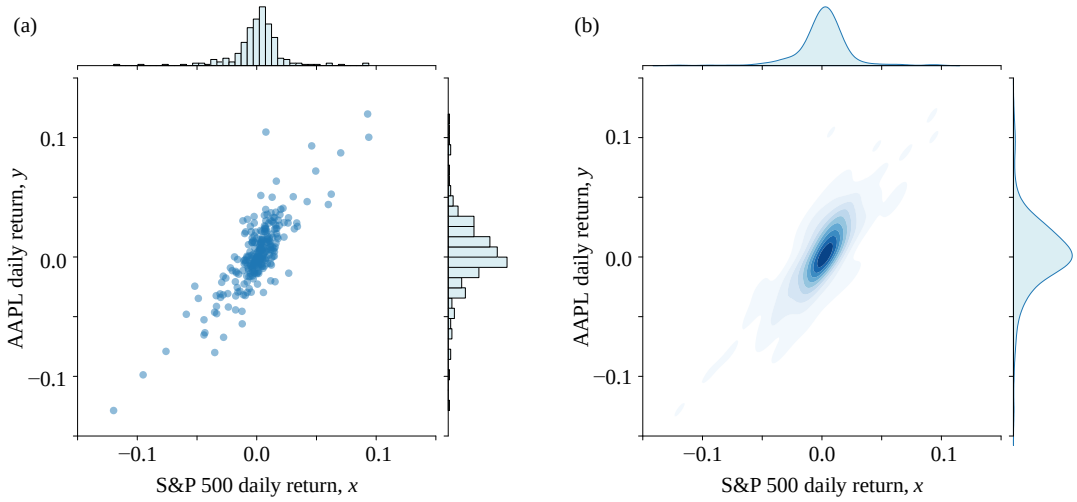


图 2. 日收益率数据关系

图 3 所示为用用 seaborn.jointplot() 绘制回归图，并且绘制边际分布。

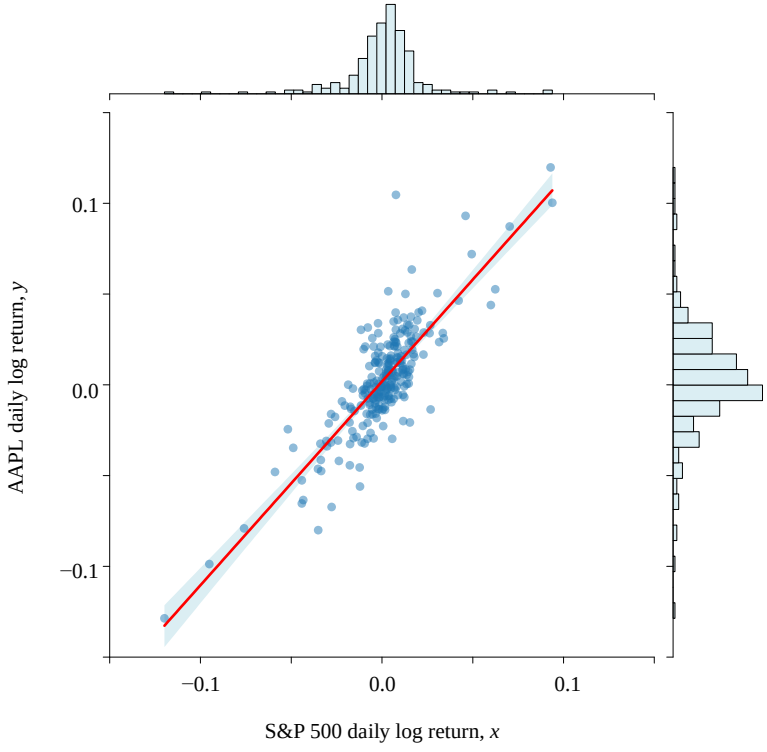


图 3. 用 seaborn.jointplot() 绘制回归直线

统计特征

图 4 (a) 所示为数据的协方差矩阵。

《统计至简》第 12、24 章介绍过如何从条件概率角度理解线性回归。

假设 X 和 Y 的均值为 0，请大家根据这个协方差矩阵写出线性回归解析式。

图 4 (b) 所示为相关性系数矩阵热图。

《矩阵力量》第 23 章介绍过相关性系数可以看成是“标准差向量”之间夹角，具体如图 4 (c) 所示。

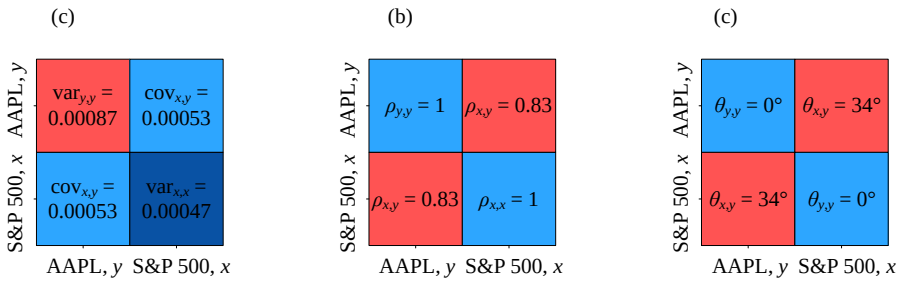


图 4. [y, x] 数据的协方差矩阵、相关性和夹角热图

图 5 所示为两个标准差向量的箭头图。夹角越小，说明因变量向量 $\boldsymbol{y}$ 和自变量向量 $\boldsymbol{x}$ 越相近。也就是说，夹角越小，自变量向量 $\boldsymbol{x}$ 能更充分解释因变量向量 $\boldsymbol{y}$。本章后文还会利用这个几何视角解释回归分析结果。

本章内容相对比较枯燥，建议大家主要理解 ANOVA。大家有实际需要时再回头查阅本章其余内容。

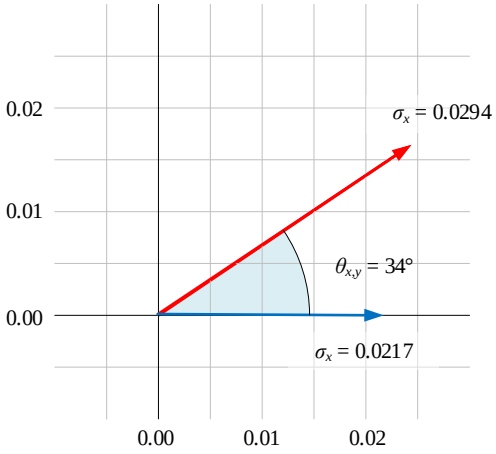
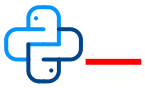


图 5. 标准差向量空间角度解释夹角



Bk6_Ch09_01.py 绘制本节图像。

9.2 方差分析 ANOVA

本节开始先介绍如何理解图 6 所示的 ANOVA 表格结果。ANOVA 的含义是**方差分析** (Analysis of Variance)。方差分析是一种用于确定线性回归模型中不同变量对目标变量解释程度的统计技术。方差分析通过比较模型中不同的变量的平均方差，来确定哪些变量对目标变量的解释程度更高。

ANOVA 是图 1 的重要组成部分之一。

	df	sum_sq	mean_sq	F	PR(>F)
x	1.0	0.149314	0.149314	549.729877	4.547141e-65
Residual	250.0	0.067903	0.000272	NaN	NaN

图 6. 一元线性回归 ANOVA 表格，来自本书第 6 章

表 1 所示为标准 ANOVA 表格对应的统计量。标准 ANOVA 表格比图 6 多一行。表 1 有五列：

第 1 列为计算方差的三个来源；

第 2 列 df 代表**自由度** (degrees of freedom)；自由度是指在计算统计量时可以随意变化的独立数据点的数量。

第 3 列 SS 代表**平方和** (Sum of Squares)；平方和通常用于描述数据的变异程度，即它们偏离平均值的程度。

第 4 列 MS 代表**均方和** (Mean Sum of Squares)；在统计学中，均方和是一种平均值的度量，其计算方法是将平方和除以自由度。

第 5 列 F 代表 F-test 统计量。F 检验是一种基于方差比较的统计检验方法，用于确定两个或多个样本之间是否存在显著性差异。

表中 n 代表参与回归的非 NaN 样本数量。k 代表回归模型参数数量，包括截距项。D 代表因变量的数量，因此 $k = D + 1$ (+1 代表常数项参数)。下面将逐个解密表 1 中的每一个值的含义，以及它们和线性回归的关系。

表 1. ANOVA 表格

Source	df	SS	MS	F	Significance

Regressor	DFR = $D = k - 1$	SSR	MSR = SSR/DFR	F = MSR/MSE	p-value of F-test
Residuals	DFE = $n - D - 1 = n - k$	SSE	MSE = SSE/DFE		
Total	DFT = $n - 1$	SST			

三个平方和

为了理解 ANOVA 表格，我们首先要了解三个平方和：

- **总离差平方和** (Sum of Squares for Total, SST)，也称 TSS (total sum of squares)。总离差平方和 SST 描述所有观测值与总体均值之间差异的平方和，用来评整个数据集的离散程度。
- **残差平方和** (Sum of Squares for Error, SSE)，也称 RSS (residual sum of squares)。残差平方和 SSE 反映了因变量中无法通过自变量预测的部分，也称为误差项，可以用于检查回归模型的拟合程度和判断是否存在异常值。在回归分析中，常用通过最小化残差平方和来确定最佳的回归系数。
- **回归平方和** (Sum of Squares for Regression, SSR)，也称 ESS (explained sum of squares)。回归平方和 SSR 反映了回归模型所解释的数据变异量的大小，用于评估回归模型的拟合程度以及自变量对因变量的影响程度。

图 7 给出计算三个平方和所需的数值。表 2 总结了三个平方和的定义。

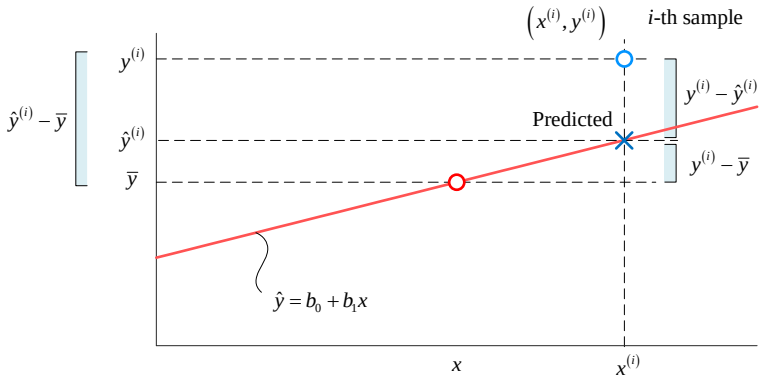
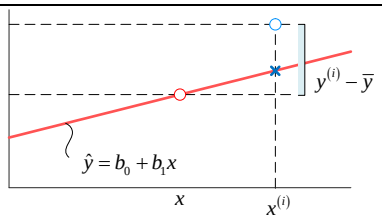
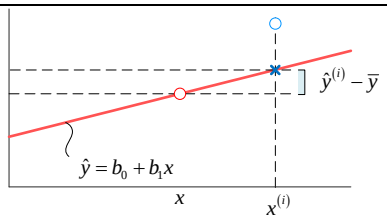
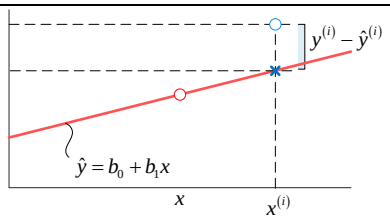


图 7. 通过一元线性回归模型分解因变量的变化

表 2. 三个平方和的定义

平方和	定义	图像

总离差平方和 (Sum of Squares for Total, SST)	$\text{SST}=\sum_{i=1}^{n}\left(y^{(i)}-\bar{y}\right)^2$	
回归平方和 (Sum of Squares for Regression, SSR)	$\text{SSR}=\sum_{i=1}^{n}\left(\hat{y}^{(i)}-\bar{y}\right)^2$	
残差平方和 (Sum of Squares for Error, SSE)	$\text{SSE}=\sum_{i=1}^{n}\left(y^{(i)}-\hat{y}^{(i)}\right)^2$	

等式关系

对于线性回归来说，方差分析实际上就是把 SST 分解成残差平方和 SSE、回归平方和 SSR：

$$\text{SST}=\text{SSR}+\text{SSE} \tag{1}$$

即：

$$\underbrace{\sum_{i=1}^{n}\left(y^{(i)}-\bar{y}\right)^2}_{\text{SST}}=\underbrace{\sum_{i=1}^{n}\left(\hat{y}^{(i)}-\bar{y}\right)^2}_{\text{SSR}}+\underbrace{\sum_{i=1}^{n}\left(y^{(i)}-\hat{y}^{(i)}\right)^2}_{\text{SSE}} \tag{2}$$

上式的证明并不难，本节不做展开讲解，本章后续会用向量几何视角解释以上等式关系。本章后续将介绍由这三个平方和引出的一些列有关回归的统计量，特别是 R-squared 和 Adj. R-squared。

9.3 总离差平方和 SST

总离差平方和 (Sum of Squares for Total, SST) 代表因变量 y 所有样本点与期望值 $\bar{y}$ 的差异：

$$\text{SST}=\sum_{i=1}^{n}\left(y^{(i)}-\bar{y}\right)^2 \tag{3}$$

其中，期望值 $\bar{y}$ 为：

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y^{(i)} \tag{4}$$

如图 8 所示，SST 可以看做一系列正方形面积之和。这些正方形的边长为 $\left|y^{(i)} - \bar{y}\right|$。图 8 中这些正方形的一条边都在期望值 $\bar{y}$ 这个高度上。

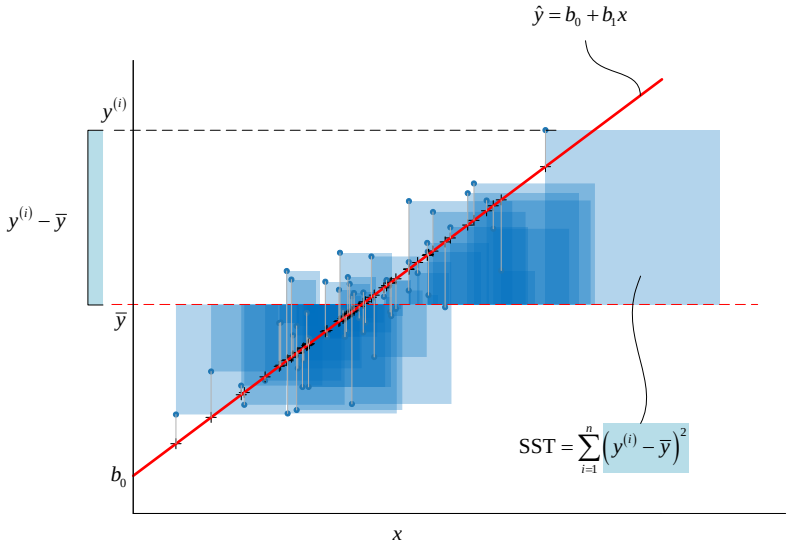


图 8. 总离差平方和 SST

总离差自由度 DFT

总离差自由度 (degree of freedom total, DFT) 的定义为：

$$\text{DFT} = n - 1 \tag{5}$$

n 是样本数据的数量 (NaN 除外)。

三个自由度关系

总离差自由度 DFT、回归自由度 DFR、残差自由度 DFE 三者关系为：

$$\text{DFT} = n - 1 = \text{DFR} + \text{DFE} = \underbrace{(k-1)}_{\text{DFR}} + \underbrace{(n-k)}_{\text{DFE}} = \underbrace{(D)}_{\text{DFR}} + \underbrace{(n-D-1)}_{\text{DFE}} \tag{6}$$

k 是回归模型的参数，其中包括截距项。因此，

$$k = D + 1 \tag{7}$$

D 为参与回归模型的特征数，也就是因变量的数量。

举个例子，对于一元线性回归，$D = 1$，$k = 2$。如果参与建模的样本数据为 $n = 252$，几个自由度分别为：

$$\begin{cases} \text{DFT} = 252 - 1 = 251 \\ k = D + 1 = 2 \\ \text{DFR} = k - 1 = D = 1 \\ \text{DFE} = n - k = n - D - 1 = 252 - 2 = 250 \end{cases} \tag{8}$$

平均总离差 MST

平均总离差 (mean square total, MST) 的定义为：

$$\text{MST} = \text{var}(Y) = \frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{n-1} = \frac{\text{SST}}{\text{DFT}} \tag{9}$$

实际上，总离差 MST 便是因变量 Y 样本数据方差。看到这里，大家应该理解为什么本章的内容叫“方差分析”了。

9.4 回归平方和 SSR

回归平方和 (Sum of Squares for Regression, SSR) 代表回归方程计算得到的预测值 $\hat{y}^{(i)}$ 和期望值 $\bar{y}$ 之间的差异：

$$\text{SSR} = \sum_{i=1}^{n}\left(\hat{y}^{(i)} - \bar{y}\right)^2 \tag{10}$$

图 9 所示为回归平方和 SSR 的几何意义。图 9 中的每个正方形边长为 $\left|\hat{y}^{(i)} - \bar{y}\right|$。

⚠ 注意，图中所有正方形的一个顶点都在回归直线上。

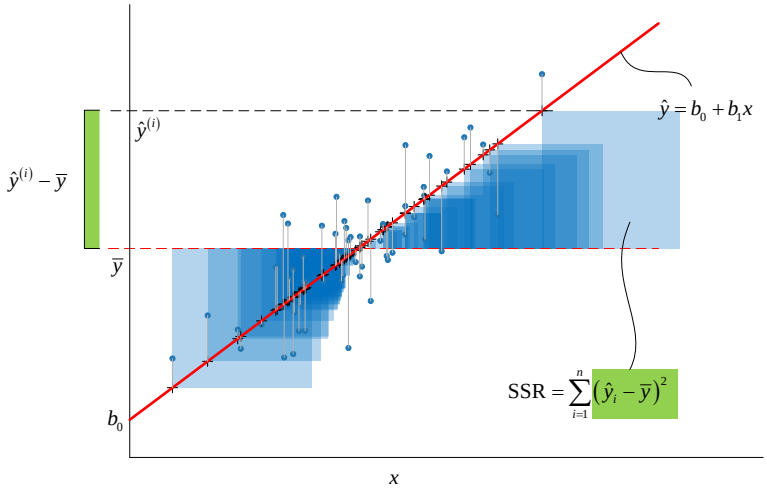


图 9. 回归平方和

回归自由度 DFR

回归自由度 (degrees of freedom for regression model, DFR) 为：

$$\text{DFR} = k - 1 = D \tag{11}$$

平均回归平方 MSR

平均回归平方 (mean square regression, MSR) 为：

$$\text{MSR} = \frac{\text{SSR}}{\text{DFR}} = \frac{\text{SSR}}{k-1} = \frac{\text{SSR}}{D} \tag{12}$$

9.5 残差平方和 SSE

残差平方和 (Sum of Squares for Error, SSE) 定义如下：

$$\text{SSE} = \sum_{i=1}^{n}\left(\varepsilon^{(i)}\right)^2 = \sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2 \tag{13}$$

相信大家对残差平方和 SSE 已经很熟悉。比如，在最小二乘法中，我们通过最小化残差平方和 SSE 优化回归参数。

图 10 所示为残差平方和 SSE 的示意图。图中每个正方形的边长为 $\left|y^{(i)} - \hat{y}^{(i)}\right|$。对于 OLS 一元线性回归，我们期待图中蓝色正方形面积之和最小。

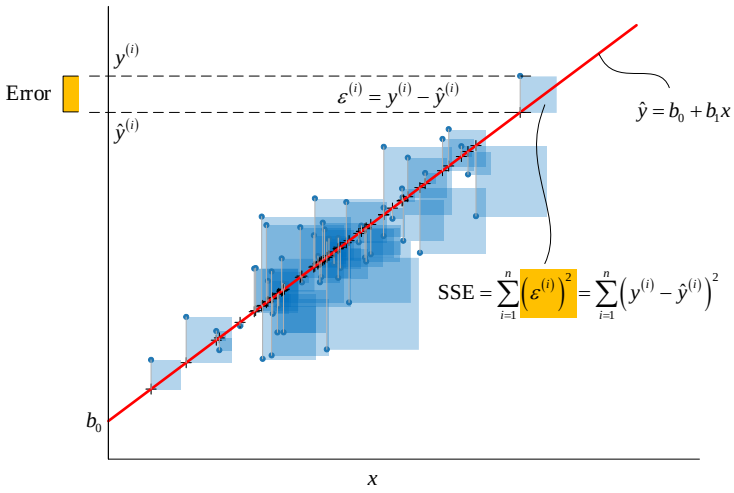


图 10. 残差平方和 SSE

残差自由度 DFE

残差自由度 (degrees of freedom for error, DFE) 为：

$$\text{DFE} = n - k = n - D - 1 \tag{14}$$

残差平均值 MSE

残差平均值 (mean squared error, MSE) 为：

$$\text{MSE} = \frac{\text{SSE}}{\text{DFE}} = \frac{\text{SSE}}{n-k} = \frac{\text{SSE}}{n-D-1} \tag{15}$$

均方根残差 RMSE

均方根残差 (Root mean square error, RMSE) 为 MSE 的平方根：

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{\text{SSE}}{\text{DFE}}} = \sqrt{\frac{\text{SSE}}{n-p}} = \sqrt{\frac{\text{SSE}}{n-D-1}} \tag{16}$$

9.6 几何视角：勾股定理

大家别忘了《矩阵力量》反复提到的线性回归几何视角！

一个直角三角形

看到 (2) 中三个求和，我们下面用向量范数算式完成三个求和运算：

$$\begin{aligned} \text{SST} &= \sum_{i=1}^{n}\left(y^{(i)} - \bar{y}\right)^2 = \left\|\boldsymbol{y} - \bar{y}\boldsymbol{1}\right\|_2^2 \\ \text{SSR} &= \sum_{i=1}^{n}\left(\hat{y}^{(i)} - \bar{y}\right)^2 = \left\|\hat{\boldsymbol{y}} - \bar{y}\boldsymbol{1}\right\|_2^2 \\ \text{SSE} &= \sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2 = \left\|\boldsymbol{y} - \hat{\boldsymbol{y}}\right\|_2^2 \end{aligned} \tag{17}$$

根据 (2)，我们可以得到如下等式：

$$\underbrace{\left\|\boldsymbol{y} - \bar{y}\boldsymbol{1}\right\|_2^2}_{\text{SST}} = \underbrace{\left\|\hat{\boldsymbol{y}} - \bar{y}\boldsymbol{1}\right\|_2^2}_{\text{SSR}} + \underbrace{\left\|\boldsymbol{y} - \hat{\boldsymbol{y}}\right\|_2^2}_{\text{SSE}} \tag{18}$$

相信大家一眼就会看出来，(18) 代表着直角三角形勾股定理！

如图 11 (a) 所示，$\boldsymbol{y}-\bar{y}\boldsymbol{1}$ 就是斜边对应的向量，斜边长度为 $\left\|\boldsymbol{y}-\bar{y}\boldsymbol{1}\right\|$。$\hat{\boldsymbol{y}}-\bar{y}\boldsymbol{1}$ 为第一条直角边，$\hat{\boldsymbol{y}}-\bar{y}\boldsymbol{1}$ 代表回归模型解释的部分。$\boldsymbol{y}-\hat{\boldsymbol{y}}$ 为第二条直角边，代表残差项，也就是回归模型不能解释的部分。

⚠ 注意，图 11 中 $\boldsymbol{y}-\bar{y}\boldsymbol{1}$ 和 $\hat{\boldsymbol{y}}-\bar{y}\boldsymbol{1}$ 的起点为 $\bar{y}\boldsymbol{1}$ 的终点，这相当于去均值。

如图 11 (b) 所示，这个勾股定理还可以写成：

$$\left(\sqrt{\text{SST}}\right)^2=\left(\sqrt{\text{SSR}}\right)^2+\left(\sqrt{\text{SSE}}\right)^2 \tag{19}$$

此外，请大家注意图中 θ，θ 是向量 $\boldsymbol{y}-\bar{y}\boldsymbol{1}$ 和向量 $\hat{\boldsymbol{y}}-\bar{y}\boldsymbol{1}$ 的夹角，下一节会用到它。

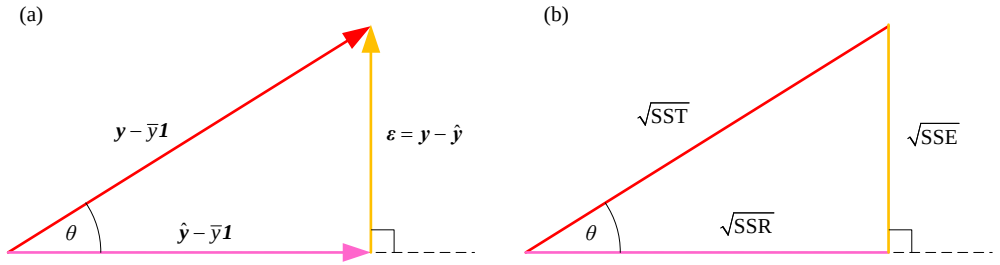


图 11. 几何角度看三个平方和

四个直角三角形

图 11 的直角三角形是图 12 这个四面体的一个面 (灰色底色)。而图 12 这个四面体的四个面都是直角三角形！

现在请大家自己试着理解这个四面体和四个直角三角形的含义，下一章会深入分析。

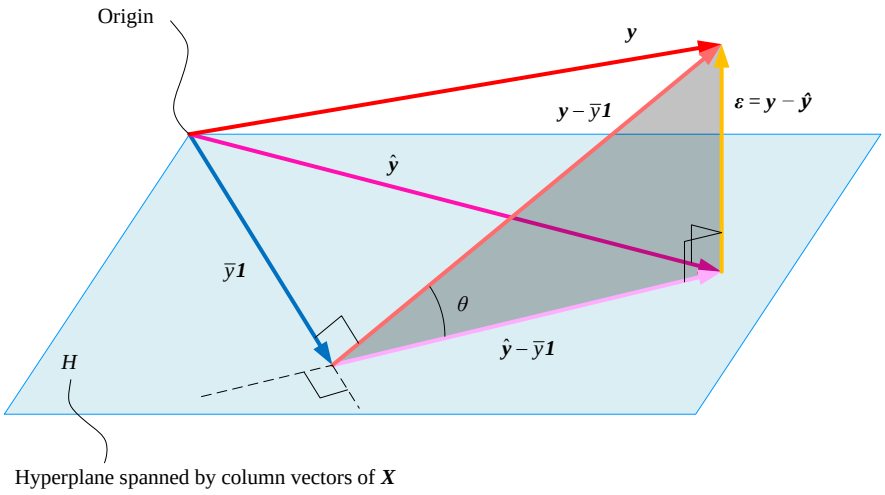


图 12. 四面体的四个面都是直角三角形

9.7 拟合优度：评价拟合程度

如图 13 所示，向量 $\boldsymbol{y}-\bar{y}\boldsymbol{1}$ 和向量 $\hat{\boldsymbol{y}}-\bar{y}\boldsymbol{1}$ 之间夹角 θ 越小，说明误差越小，代表拟合效果越好。

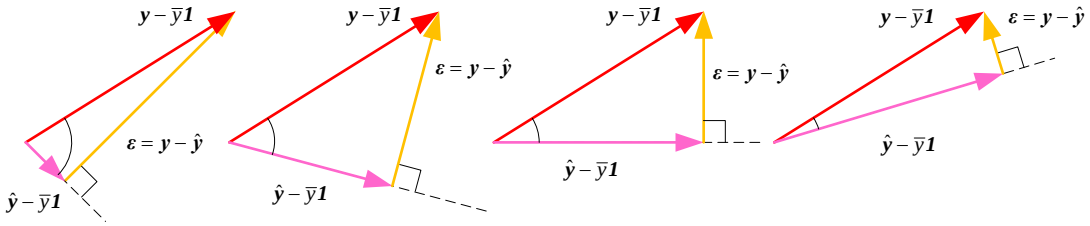


图 13. 因变量向量和预测值向量夹角从大到小

在回归模型创建之后，很自然就要考虑这个模型是否能够很好地解释数据，即考察这条回归线对观察值的拟合程度，也就是所谓的**拟合优度** (goodness of fit)。拟合优度是指一个统计模型与观测数据之间的拟合程度，即模型能够多好地解释数据。简单地说，拟合优度是回归分析中考察样本数据点对于回归线的贴合程度。

决定系数 (coefficient of determination，R^2) 是定量化反映模型拟合优度的统计量。从几何角度来看，R^2 是图 12 中 θ 余弦值 $\cos\theta$ 的平方：

$$R^2 = \cos(\theta)^2 \tag{20}$$

利用图 11 (b) 直角三角形三边之间的关系，R^2 可以整理为：

$$R^2 = \frac{\text{SSR}}{\text{SST}} = 1 - \frac{\text{SSE}}{\text{SST}} \tag{21}$$

当预测值越接近样本值，R^2 越接近 1；相反，若拟合效果越差，R^2 越接近 0。拟合优度可以帮助评估回归模型的可靠性和预测能力，并对模型进行改进和优化。

一元线性回归

特别地，对于一元线性回归，决定系数是因变量与自变量的相关系数的平方，与模型系数 b_1 也有直接关系。

$$R^2 = \rho_{X,Y}^2 = \left(b_1 \frac{\sigma_X}{\sigma_Y} \right)^2 \tag{22}$$

其中，

$$b_1 = \rho_{X,Y} \frac{\sigma_Y}{\sigma_X} \tag{23}$$

也就是说，在一元线性回归中，R^2 的平方根等于线性相关系数的绝对值。也就是说，当 ρ 等于 1 或−1 时，R^2 为 1，表示因变量完全由自变量解释；当 ρ 等于 0 时，R^2 为 0，表示自变量对因变量没有任何解释能力。因此，R^2 越接近 1，表示自变量对因变量的解释能力越强，线性相关系数 ρ 的绝对值也越大，反之亦然。

因此，线性相关系数 ρ 和决定系数 R^2 都是衡量变量之间线性关系强弱的重要指标，它们可以帮助我们理解自变量对因变量的解释能力，评估模型的拟合优度，以及选择最佳的回归模型。

修正决定系数

但是，仅仅使用 R^2 是不够的。对于多元线性模型，不断增加解释变量个数 D 时，R^2 将不断增大。我们可以利用**修正决定系数** (adjusted R squared)。简单来说，修正决定系数考虑到自变量的数目对决定系数的影响，避免了当自变量数量增加时决定系数的人为提高。修正决定系数的具体定义为：

$$\begin{aligned} R_{\text{adj}}^2 &= 1 - \frac{\text{MSE}}{\text{MST}} \\ &= 1 - \frac{\text{SSE}/(n-k)}{\text{SST}/(n-1)} \\ &= 1 - \left(\frac{n-1}{n-k} \right) \frac{\text{SSE}}{\text{SST}} \\ &= 1 - \left(\frac{n-1}{n-k} \right) \left(1 - R^2 \right) \\ &= 1 - \left(\frac{n-1}{n-D-1} \right) \frac{\text{SSE}}{\text{SST}} \end{aligned} \tag{24}$$

修正决定系数的作用在于，当模型中自变量的数量增加时，它能够惩罚**过拟合** (overfitting)，并避免了决定系数因为自变量个数增加而提高的问题。因此，在比较不同模型的拟合优度时，使用修正决定系数会更加准确，能够更好地刻画模型的解释能力。

过拟合是指一个模型在训练数据上表现良好，但在测试数据上表现较差的现象。在过拟合的情况下，模型过度地学习了训练数据的特征和噪声，导致其在测试数据上的预测能力下降。

过拟合通常发生在模型复杂度过高或者训练数据太少的情况下。例如，在一元线性回归中，如果使用高次多项式来拟合数据，就容易出现过拟合的情况。在这种情况下，模型会过度拟合训练数据，导致其在新数据上的预测能力下降。

为了避免过拟合，可以采取以下方法：增加训练数据量、降低模型复杂度、采用**正则化** (regularization) 技术等。

 本书第 11 章将讲解正则化回归。

9.8 F检验：模型参数不全为 0

在线性回归中，F 检验用于检验线性回归模型是否显著。它通过比较回归平方和和残差平方和的大小来判断模型是否具有显著的解释能力。

统计量

F 检验的统计量为：

$$F=\frac{\mathrm{MSR}}{\mathrm{MSE}}=\frac{\frac{\mathrm{SSR}}{k-1}}{\frac{\mathrm{SSE}}{n-k}}=\frac{\mathrm{SSR}(n-k)}{\mathrm{SSE}(k-1)}=\frac{\frac{\mathrm{SSR}}{D}}{\frac{\mathrm{SSE}}{n-D-1}}=\frac{\mathrm{SSR}\cdot(n-D-1)}{\mathrm{SSE}\cdot(D)}\sim F(k-1,n-k) \tag{25}$$

原假设、备择假设

假设检验 (hypothesis testing) 是统计学中常用的一种方法，用于根据样本数据推断总体参数是否符合某种假设。假设检验通常包括两个假设：原假设和备择假设。

原假设 (null hypothesis) 是指在实验或调查中假设成立的一个假设，通常认为其成立。

备择假设 (alternative hypothesis) 是指当原假设不成立时，我们希望成立的另一个假设。

通过收集样本数据，并根据统计学原理计算出样本统计量的概率分布，我们可以计算出拒绝原假设的概率。如果这个概率小于预设的显著性水平 (比如 0.05)，就可以拒绝原假设，认为备择假设成立。反之，如果这个概率大于预设的显著性水平，就不能拒绝原假设。

F 检验是单尾检验，原假设 H_0、备择假设 H_1 分别为：

$$\begin{aligned} H_0&: \ b_1 = b_2 = \cdots = b_D = 0 \\ H_1&: \ b_j \neq 0 \ \text{ for at least one } j \end{aligned} \tag{26}$$

具体来说，F 检验的零假设是模型的所有回归系数都等于零，即自变量对因变量没有显著的影响。如果 F 检验的 p 值小于设定的显著性水平，就可以拒绝零假设，认为模型是显著的，即自变量对因变量有显著的影响。

临界值

(25) 得到的 F 值和临界值 F_α 进行比较。临界值 F_α 可根据两个自由度 ($k-1$ 和 $n-k$) 以及置信水平 α 查表获得。$1-\alpha$ 为置信度或置信水平，通常取 $\alpha = 0.05$ 或 $\alpha = 0.01$。这表明，当作出接受原假设的决定时，其正确的可能性为 95%或 99%。

如果，

$$F > F_{1-\alpha}\left(k-1, n-k\right) \tag{27}$$

在该置信水平上拒绝零假设 H_0，不认为自变量系数同时具备非显著性，即所有系数不太可能同时为零。

否则，接受 H_0，自变量系数同时具有非显著性，即所有系数很可能同时为零。

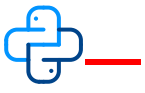
举个例子

给定条件 $\alpha = 0.01$，$F_{1-\alpha}(1, 250) = 6.7373$。图 6 结果告诉我们，$F = 549.7 > 6.7373$，表明可以显著地拒绝 H_0。

也可以用图 6 中 p 值，

$$p\text{-value} = \mathrm{P}\left(F < F_\alpha\left(k-1, n-k\right)\right) \tag{28}$$

如果 p 值小于 α，则可以拒绝零假设 H_0。



Bk6_Ch09_02.py 计算图 6 所示方差分析表格中统计量。

9.9 t检验：某个回归系数是否为 0

在线性回归中，t 检验主要用于检验线性回归模型中某个特定自变量的系数是否显著。具体地，t 检验的零假设是特定回归系数等于零，即自变量对因变量没有显著的影响。如果 t 检验的 p 值小于设定的显著性水平，就可以拒绝零假设，认为该自变量的系数是显著不为零的，即自变量对因变量有显著的影响。

需要注意的是，t 检验一般用来检验一个特定自变量的系数是否显著，而不能判断模型整体是否显著。如果需要判断模型整体的显著性，可以使用前文介绍的 F 检验。

原假设、备择假设

对于一元线性回归，t 检验原假设和备择假设分别为：

$$\begin{cases} H_0: b_1 = b_{1,0} \\ H_1: b_1 \neq b_{1,0} \end{cases} \tag{29}$$

一般 $b_{1,0}$ 取 0，也就是检验回归系数是否为 0。当然，$b_{1,0}$ 也可以取其他值。

统计量

b_1 的 t 检验统计量：

$$t_{b1} = \frac{\hat{b}_1 - b_{1,0}}{\mathrm{SE}\left(\hat{b}_1\right)} \tag{30}$$

$\hat{b}_1$ 为最小二乘法 OLS 线性回归估算得到的系数，$\mathrm{SE}\left(\hat{b}_1\right)$ 为其标准误：

$$\mathrm{SE}\left(\hat{b}_1\right) = \sqrt{\frac{\mathrm{MSE}}{\sum_{i=1}^{n}\left(x^{(i)} - \bar{x}\right)^2}} = \sqrt{\frac{\frac{\sum_{i=1}^{n}\left(\varepsilon^{(i)}\right)^2}{n-2}}{\sum_{i=1}^{n}\left(x^{(i)} - \bar{x}\right)^2}} \tag{31}$$

上式中，MSE 为本章前文介绍的**残差平均值** (mean squared error)，n 是样本数据的数量 (除 NaN)。标准误越大，回归系数的估计值越不可靠。

临界值

如果下式成立，接受零假设 H_0：

$$-t_{1-\alpha/2,\, n-2} < T < t_{1-\alpha/2,\, n-2} \tag{32}$$

否则，则拒绝零假设 H_0。

特别地，如果原假设和备择假设为：

$$\begin{cases} H_0: b_1 = 0 \\ H_1: b_1 \neq 0 \end{cases} \tag{33}$$

如果 (32) 成立，接受零假设 H_0，即回归系数不具有显著统计性；白话说，也就是 $b_1 = 0$，意味着自变量和因变量不存在线性关系。否则，则拒绝零假设 H_0，即回归系数具有显著统计性。

截距项系数

对于一元线性回归，对截距项系数 b_0 的假设检验程序和上述类似。b_0 的 t 检验统计值：

$$t_{b0} = \frac{\hat{b}_0 - b_{0,0}}{\mathrm{SE}\left(\hat{b}_0\right)} \tag{34}$$

$\hat{b}_0$ 为最小二乘法 OLS 线性回归估算得到的系数，$\mathrm{SE}\left(\hat{b}_0\right)$ 为其标准误：

$$\mathrm{SE}\left(\hat{b}_0\right) = \sqrt{\frac{\sum_{i=1}^{n}\left(\varepsilon^{(i)}\right)^2}{n-2}\left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}\left(x^{(i)} - \bar{x}\right)^2}\right]} \tag{35}$$

举个例子

t 检验统计值 T 服从自由度为 $n - 2$ 的 t 分布。本节采用的 t 检验是双尾检测。在统计学中，双尾假设检验是指在假设检验过程中，假设被拒绝的区域位于一个统计量分布的两个尾端，即研究者对于一个参数或者统计量是否等于某一特定值，不确定其比该值大或小，而是存在两种可能性，因此需要在两个尾端进行检验。

比如给定显著性水平 $\alpha = 0.05$ 和自由度 $n - 2 = 252 - 2 = 250$，可以查表得到 t 值，即：

$$t_{1-\alpha/2,\, n-2} = t_{0.975,\, 250} = 1.969498 \tag{36}$$

Python 中，可以用 stats.t.ppf(1 - alpha/2, DFE) 计算上式两值。

由于学生 t-分布对称，所以：

$$t_{\alpha/2,\, n-2} = t_{0.025,\, 250} = -1.969498 \tag{37}$$

如图 1 所示，$t_{b1} = 23.446$，因此：

$$t_{b1} > t_{0.975,\, 250} \tag{38}$$

表明参数 b_1 的 t 检验在 $\alpha = 0.05$ 水平下是显著的，也就是可以显著地拒绝 H_0: $b_1 = 0$，从而接受 H_1: $b_1 \neq 0$。回归系数的标准误差越大，回归系数的估计值越不可靠。

而 $t_{b0} = 1.759$，因此：

$$t_{b0} < t_{0.975,\ 250} \tag{39}$$

则表明参数 b_0 的 t 检验在 $\alpha = 0.05$ 水平下是不显著的，也就是不能显著地拒绝 H_0: $b_0 = 0$。尽管模型含有截距项，但若该项的出现是统计上不显著的 (即统计上等于零)，则从任何实际方面考虑，都可认为这个结果是一个过原点回归模型。

因此，系数 b_1 的 $1 - \alpha$ 置信区间为：

$$\hat{b}_1 \pm t_{1-\alpha/2,\ n-2} \cdot \mathrm{SE}\left(\hat{b}_1\right) \tag{40}$$

这个置信区间的的含义是，真实 b_1 在以上区间的概率为 $1 - \alpha$。

系数 b_0 的 $1 - \alpha$ 置信区间为：

$$\hat{b}_0 \pm t_{1-\alpha/2,\ n-2} \cdot \mathrm{SE}\left(\hat{b}_0\right) \tag{41}$$

同理，真实 b_0 在以上区间的概率为 $1 - \alpha$。

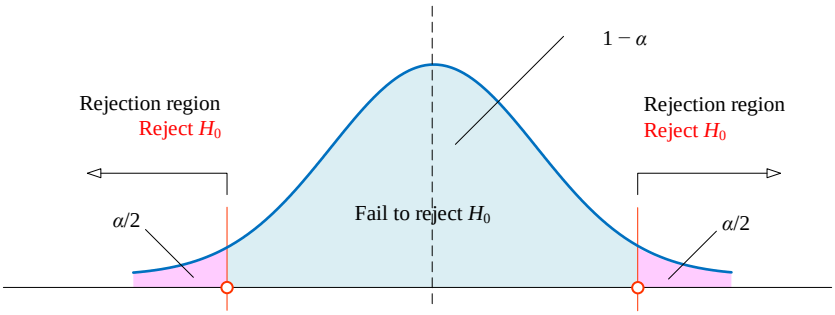


图 14. 双尾检验

9.10 置信区间：因变量均值的区间

本书前文在介绍一元线性回归中，大家都应该见过类似图 15 的图像。图中的带宽代表预测值的置信区间。

预测值 $\hat{y}^{(i)}$，的 $1 - \alpha$ 置信区间：

$$\hat{y}^{(i)} \pm t_{1-\alpha/2,\ n-2} \cdot \sqrt{\mathrm{MSE}} \cdot \sqrt{\frac{1}{n} + \frac{\left(x^{(i)} - \bar{x}\right)^2}{\sum_{k=1}^{n}\left(x^{(k)} - \bar{x}\right)^2}} \tag{42}$$

置信区间的宽度为：

$$2\times\left\{t_{1-\alpha/2,\,n-2}\cdot\sqrt{\mathrm{MSE}}\cdot\sqrt{\frac{1}{n}+\frac{\left(x^{(i)}-\bar{x}\right)^2}{\sum_{k=1}^{n}\left(x^{(k)}-\bar{x}\right)^2}}\right\} \tag{43}$$

随着 $\left|x^{(i)}-\bar{x}\right|$ 不断增大，置信区间宽度不断增大。当 $x^{(i)}=\bar{x}$ 时，置信区间宽度最窄。随着 MSE (mean square error) 减小，置信区间宽度减小。在回归分析中，预测值置信区间用于评估回归模型的预测能力。通常，预测值的置信区间越窄，说明模型预测的精度越高。

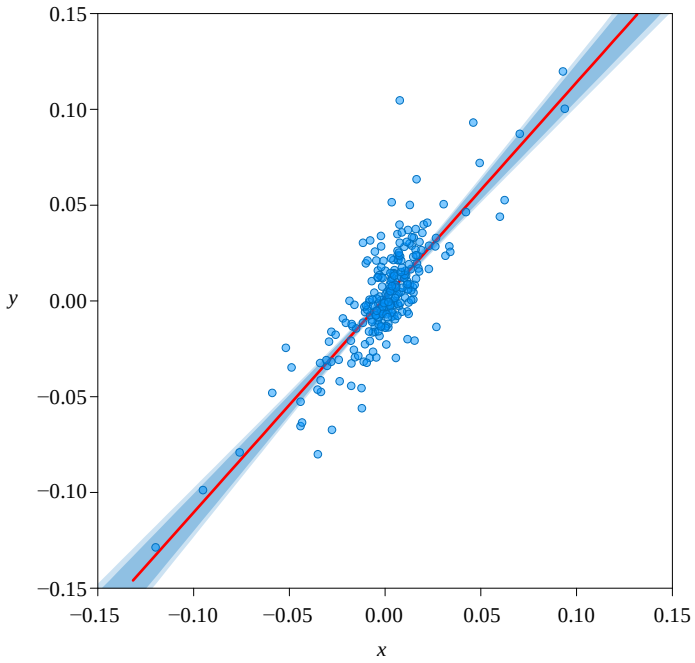


图 15. 一元线性回归线置信区间

9.11 预测区间：因变量特定值的区间

预测区间 (prediction interval) 是指回归模型估计时，对于自变量给定的某个值 x_p，求出因变量 y_p 的个别值的估计区间：

$$\hat{y}_p\pm t_{1-\alpha/2,\,n-2}\cdot\sqrt{\mathrm{MSE}}\cdot\sqrt{1+\frac{1}{n}+\frac{\left(x_p-\bar{x}\right)^2}{\sum_{k=1}^{n}\left(x^{(k)}-\bar{x}\right)^2}} \tag{44}$$

与预测值的置信区间不同，预测区间同时考虑了预测的误差和未来观测值的随机性。

预测区间包含两个方面的误差：回归方程中的估计误差和对未来观测值的随机误差。与预测值的置信区间不同，预测区间考虑了未来观测值的随机性，因此通常比置信区间更宽。

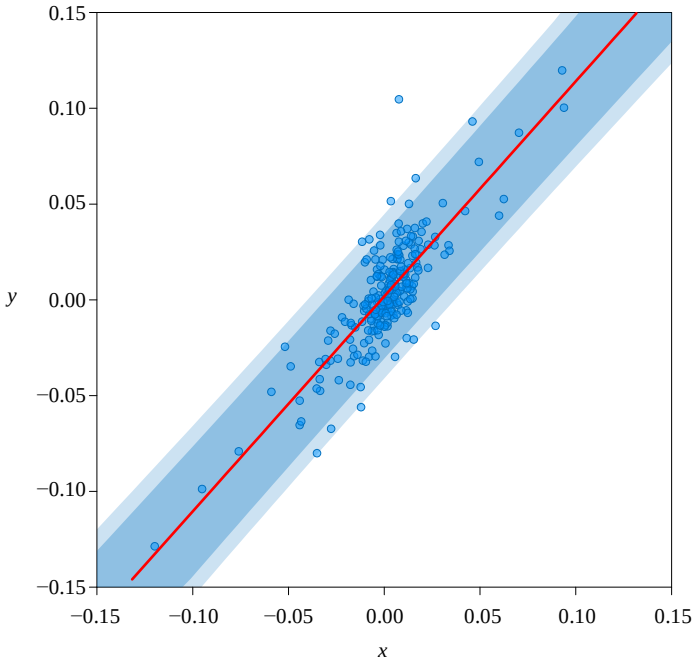


图 16. 一元线性回归线预测区间

9.12 对数似然函数：用在最大似然估计 MLE

似然函数是一种关于统计模型中的参数的函数，表示模型参数中的似然性。

残差的定义为：

$$\varepsilon^{(i)} = y^{(i)} - \hat{y}^{(i)} \tag{45}$$

在 OLS 线性回归中，假设残差服从正态分布 $N(0, \sigma^2)$，因此：

$$\Pr\left(\varepsilon^{(i)}\right) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(y^{(i)} - \hat{y}^{(i)}\right)^2}{2\sigma^2}\right) \tag{46}$$

似然函数为：

$$L = \prod_{i=1}^{n} \mathrm{P}\left(\varepsilon^{(i)}\right) = \prod_{i=1}^{n} \left\{\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(y^{(i)} - \hat{y}^{(i)}\right)^2}{2\sigma^2}\right)\right\} \tag{47}$$

常用对数似然 $\ln(L)$：

$$\ln(L)=\prod_{i=1}^{n}\mathrm{P}\left(\varepsilon^{(i)}\right)=-\frac{n}{2}\cdot\ln\left(2\pi\sigma^2\right)-\frac{\mathrm{SSE}}{2\sigma^2} \tag{48}$$

注意，MLE 中的 σ 为：

$$\sigma^2=\frac{\mathrm{SSE}}{n} \tag{49}$$

这样 $\ln(L)$ 可以写成：

$$\ln(L)=\prod_{i=1}^{n}\mathrm{P}\left(\varepsilon^{(i)}\right)=-\frac{n}{2}\cdot\ln\left(2\pi\sigma^2\right)-\frac{n}{2} \tag{50}$$

有似然函数和对数似然函数，请大家回顾《统计至简》第 16、24 章。

9.13 信息准则：选择模型的标准

AIC 和 BIC 是线性回归模型选择中常用的信息准则，用于在多个模型中选择最优模型。

AIC 为**赤池信息量准则** (Akaike information criterion, AIC)，定义如下：

$$\mathrm{AIC}=\underbrace{2k}_{\text{Penalty}}-2\ln(L) \tag{51}$$

其中，$k=D+1$；L 是似然函数。

AIC 鼓励数据拟合的优良性；但是，尽量避免出现过度拟合。(51) 中 $2k$ 项为**惩罚项** (penalty)。

贝叶斯信息准则 (Bayesian Information Criterion, BIC) 也称**施瓦茨信息准则** (Schwarz information criterion, SIC)，定义如下。

$$\mathrm{BIC}=\underbrace{k\cdot\ln(n)}_{\text{Penalty}}-2\ln(L) \tag{52}$$

其中，n 为样本数据数量。BIC 的惩罚项比 AIC 大。

在使用 AIC 和 BIC 进行模型选择时，应该选择具有最小 AIC 或 BIC 值的模型。这意味着，较小的 AIC 或 BIC 值表示更好的模型拟合和更小的模型复杂度。

需要注意的是，AIC 和 BIC 都是用来选择模型的工具，但并不保证选择的模型就是最优模型。在实际应用中，应该将 AIC 和 BIC 作为指导，结合领域知识和经验来选择最优模型。同时，还需要对模型的假设和限制进行检验，以确保模型的可靠性和实用性。

9.14 残差分析：假设残差服从均值为 0 正态分布

残差分析 (residual analysis) 通过残差所提供的信息，对回归模型进行评估，分析数据是否存在可能的干扰。残差分析的基本思想是，如果回归模型能够很好地拟合数据，那么残差应该是随机分布的，没有明显的模式或趋势。因此，对残差的分布进行检查可以提供关于模型拟合优度的信息。

残差分析通常包括以下步骤：

- 绘制残差图。残差图是观测值的残差与预测值之间的散点图。如果残差呈现出随机分布、没有明显的模式或趋势，那么模型可能具有较好的拟合优度。
- 检查残差分布。通过绘制残差直方图或核密度图来检查残差分布是否呈现出正态分布或近似正态分布。如果残差分布不是正态分布，那么可能需要采取转换或其他措施来改善模型的拟合。
- 检查残差对自变量的函数形式。通过绘制残差与自变量之间的散点图或回归曲线，来检查残差是否随自变量的变化而呈现出系统性变化。如果存在这种关系，那么可能需要考虑增加自变量、采取变量转换等方法来改善模型的拟合。

图 17 所示为残差的散点图。图 18 所示为残差分布的直方图。理想情况下，我们希望残差为均值为 0 的正态分布。为了检测残差的正态性，本节利用 Omnibus 正态检验。

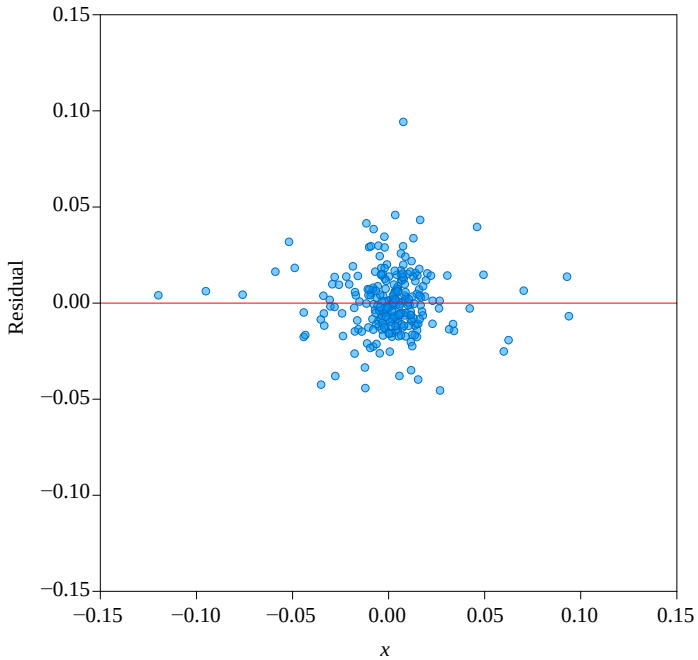


图 17. 残差散点图

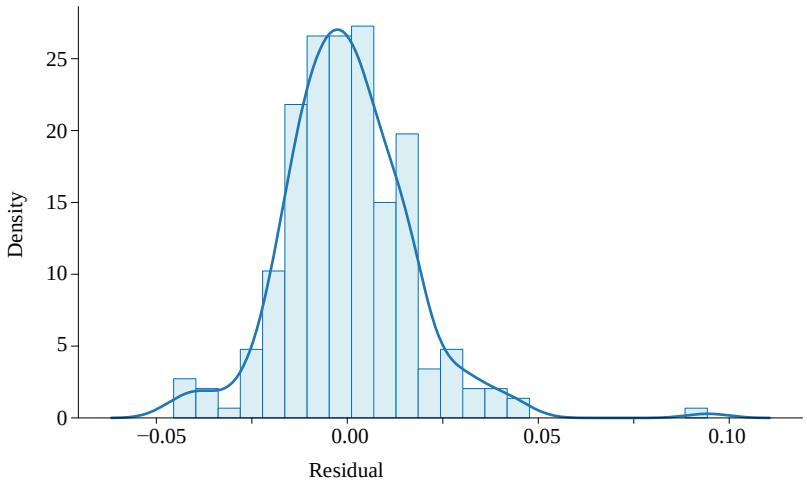


图 18. 残差分布直方图

Omnibus 正态检验 (Omnibus test for normality) 用于检验线性回归中残差是否服从正态分布。Omnibus 正态检验利用残差的偏度 S 和峰度 K，检验残差分布为正态分布的原假设。Omnibus 正态检验的统计值为偏度平方、超值峰度平方两者之和。Omnibus 正态检验利用 $\chi 2$ 检验 (Chi-squared test)。

代码中我们利用 scipy.stats.normaltest() 复现了本章前文的 Omnibus 正态检验统计量值。

《统计至简》第 2 章讲过偏度、峰度，请大家回顾。

9.15 自相关检测：Durbin–Watson

Durbin-Watson 用于检验序列的自相关。在线性回归中，**自相关** (autocorrelation) 用来分析模型中的残差与其在时间上的延迟版本之间的相关性。当模型中存在自相关时，它可能表明模型中遗漏了某些重要的变量，或者模型中的时间序列数据未被正确处理。

自相关可以通过检查残差图来诊断。如果残差图表现出明显的模式，例如残差值之间存在周期性关系或呈现出聚集在某个区域的情况，那么就可能存在自相关。在这种情况下，可以通过引入更多的自变量或使用时间序列分析方法来修正模型。图 19 所示为残差的自相关图。

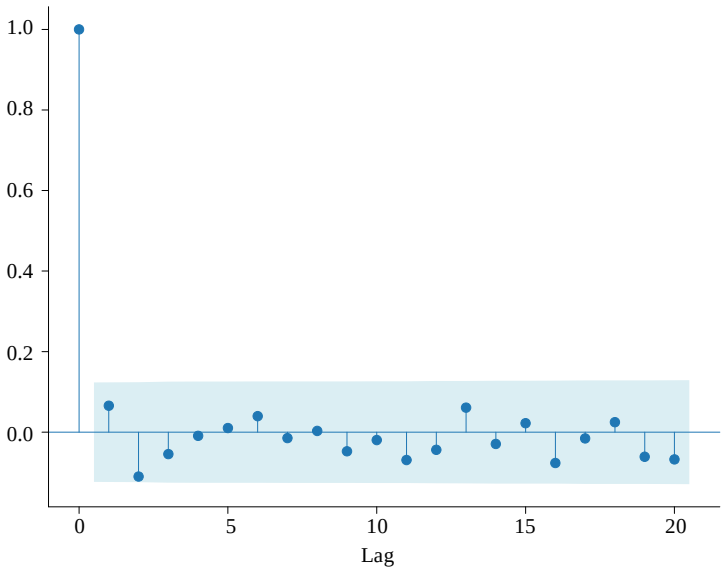


图 19. 残差自相关

Durbin-Watson 检测的统计量为：

$$DW = \frac{\sum_{i=2}^{n}\left(\left(y^{(i)}-\hat{y}^{(i)}\right)-\left(y^{(i-1)}-\hat{y}^{(i-1)}\right)\right)^2}{\sum_{i=1}^{n}\left(y^{(i)}-\hat{y}^{(i)}\right)^2} \tag{53}$$

上式本质上检测残差序列与残差的滞后一期序列之间的差异大小。DW 值的取值区间为 0 ~ 4。当 DW 值很小时 ($DW < 1$)，表明序列可能存在正自相关。当 DW 值很大时 ($DW > 3$) 表明序列可能存在负自相关。当 DW 值在 2 附近时 ($1.5 < DW < 2.5$)，表明序列无自相关。其余的取值区间表明无法确定序列是否存在自相关。

有关，请大家参考：

https://www.statsmodels.org/devel/generated/statsmodels.stats.stattools.durbin_watson.html

9.16 条件数：多重共线性

在线性回归中，**条件数** (condition number) 常用来检验设计矩阵 $\boldsymbol{X}_{k\times k}$ 是否存在**多重共线性** (multicollinearity)。

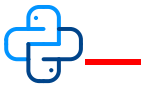
多重共线性是指在多元回归模型中，独立变量之间存在高度相关或线性关系的情况。多重共线性会导致回归系数的估计不稳定，使得模型的解释能力降低，甚至导致模型的预测精度下降。

对 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 进行特征值分解，得到最大特征值 $\lambda_{\max}$ 和最小特征值 $\lambda_{\min}$。条件数的定义为两者的比值的平方根：

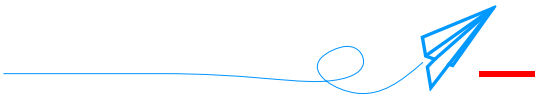
$$\text{condition number} = \sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}} \tag{54}$$

一般来说，条件数小于 30，可以不必担心多重共线性。

下一章讲到多元回归分析时，条件数的作用更明显。



Bk6_Ch09_03.py 代码复现图 1 中除 ANOVA 以外的其他统计量值。



线性回归是一种用于研究自变量与因变量之间关系的统计模型。方差分析可以评估模型的整体拟合优度，其中的 F 检验可以用来线性模型整体显著性，t 检验可以评估单个系数的显著性。拟合优度指模型能够解释数据变异的比例，常用 R^2 来度量。AIC 和 BIC 用于模型选择，可以在模型拟合度相似的情况下，选出最简单和最有解释力的模型。自相关指误差项之间的相关性，可以使用 Durbin-Watson 检验进行检测。条件数是用于评估多重共线性的指标，如果条件数过大，可能存在严重的多重共线性问题。

综上，这些概念是线性回归分析中非常重要的指标，可以帮助我们评估模型的拟合程度、系数显著性、预测能力和多重共线性等问题。这一章的内容很有难度，现在不要求大家掌握所有的知识点。

Scikit-learn 也提供线性回归分析工具，请大家参考如下网页：

https://scikit-learn.org/stable/auto_examples/inspection/plot_linear_model_coefficient_interpretation.html

10 Multivariate Linear Regression
多元线性回归

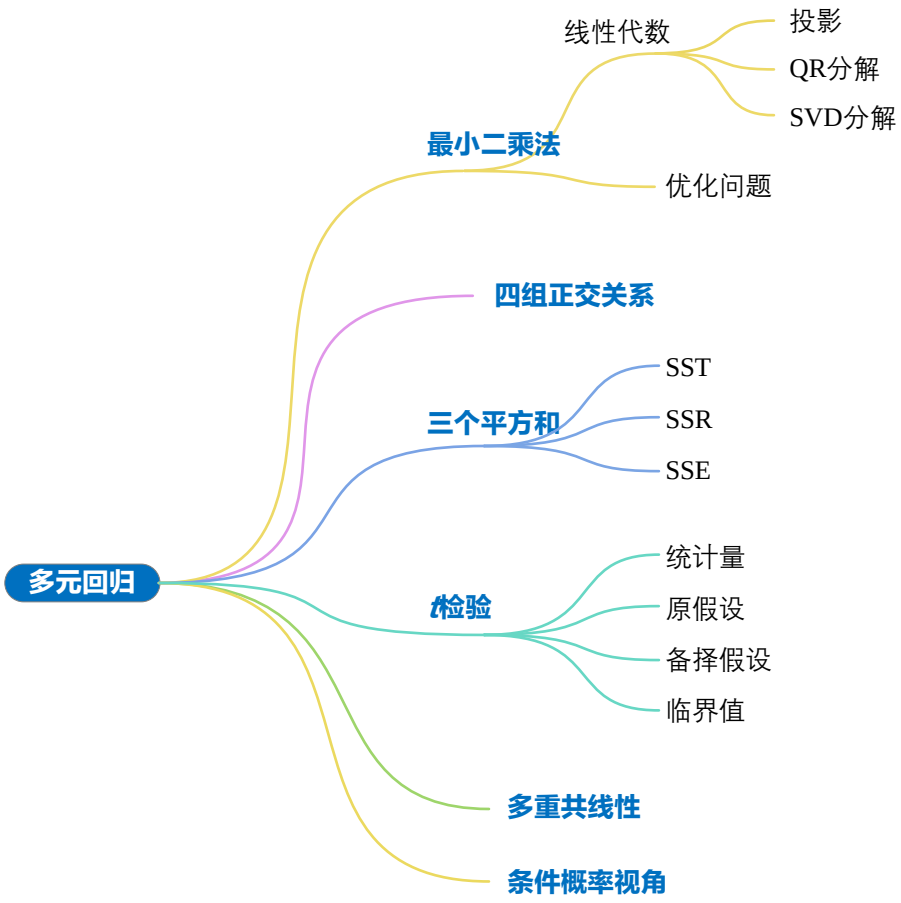
用多个解释变量来预测响应变量结果

科学不知道它对想象力的依赖。

Science does not know its debt to imagination.

—— 拉尔夫·沃尔多·爱默生 (Ralph Waldo Emerson) | 美国思想家、文学家 | 1942 ~ 2018

- `matplotlib.pyplot.quiver()` 绘制箭头图
- `numpy.arccos()` 反余弦函数
- `numpy.cov()` 计算协方差矩阵
- `numpy.identity()` 构造单位矩阵
- `numpy.linalg.det()` 计算矩阵的行列式值
- `numpy.linalg.inv()` 求矩阵逆
- `numpy.linalg.matrix_rank()` 计算矩阵的秩
- `numpy.matrix()` 构造矩阵
- `numpy.ones()` 构造全 1 矩阵或向量
- `numpy.ones_like()` 按照给定矩阵或向量形状构造全 1 矩阵或向量
- `plot_wireframe()` 绘制线框图
- `scipy.stats.f.cdf()` F 分布累积分布函数
- `seaborn.heatmap()` 绘制热图
- `seaborn.jointplot()` 绘制联合分布/散点图和边际分布
- `seaborn.kdeplot()` 绘制 KDE 核概率密度估计曲线
- `seaborn.pairplot()` 绘制成对分析图
- `statsmodels.api.add_constant()` 线性回归增加一列常数 1
- `statsmodels.api.OLS()` 最小二乘法函数
- `statsmodels.stats.outliers_influence.variance_inflation_factor()` 计算方差膨胀因子



10.1 多元线性回归

这一章将探讨多元线性回归。多元线性回归是一种统计分析方法，用于研究两个或多个自变量与一个因变量之间的关系。它通过拟合一个包含多个自变量的线性模型来预测因变量的值。

多元线性回归的表达式如下：

$$y = b_0 + b_1 x_1 + b_2 x_2 + ... + b_D x_D + \varepsilon \tag{1}$$

其中，b_0 为截距项，$b_1, b_2, \ldots, b_D$ 代表自变量系数，ε 为残差项，D 为自变量个数。几何角度来看，多元线性回归得到一个**超平面** (hyperplane)。

用矩阵运算表达 (1)：

$$\boldsymbol{y} = \underbrace{b_0 \boldsymbol{1} + b_1 \boldsymbol{x}_1 + b_2 \boldsymbol{x}_2 + \cdots + b_D \boldsymbol{x}_D}_{\hat{\boldsymbol{y}}} + \boldsymbol{\varepsilon} \tag{2}$$

其中，$\boldsymbol{1}$ 为全 1 列向量。

换一种方式来写 (2)：

$$\boldsymbol{y} = \underset{\hat{\boldsymbol{y}}}{\boldsymbol{X}\boldsymbol{b}} + \boldsymbol{\varepsilon} \tag{3}$$

其中，

$$\boldsymbol{X}_{n\times(D+1)} = \begin{bmatrix} \boldsymbol{1} & \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_D \end{bmatrix} = \begin{bmatrix} 1 & x_{1,1} & \cdots & x_{1,D} \\ 1 & x_{2,1} & \cdots & x_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & \cdots & x_{n,D} \end{bmatrix}_{n\times(D+1)}, \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_D \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon^{(1)} \\ \varepsilon^{(2)} \\ \vdots \\ \varepsilon^{(n)} \end{bmatrix} \tag{4}$$

矩阵 $\boldsymbol{X}$ 常被称作**设计矩阵** (design matrix)。图 1 所示矩阵运算对应 (3)。

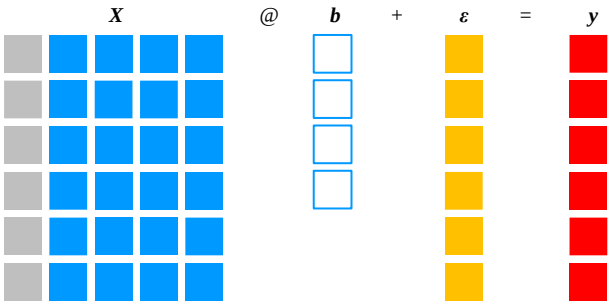


图 1. 多元线性回归模型矩阵运算

预测值构成的列向量 $\hat{\boldsymbol{y}}$，通过下式计算得到：

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{b} \tag{5}$$

残差向量的算式为：

$$\boldsymbol{\varepsilon} = \boldsymbol{y} - \hat{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{X}\boldsymbol{b} \tag{6}$$

如图 2 所示，第 i 个观测点的残差项，可以通过下式计算得到：

$$\varepsilon^{(i)} = y^{(i)} - \hat{y}^{(i)} = y^{(i)} - \boldsymbol{x}^{(i)}\boldsymbol{b}$$

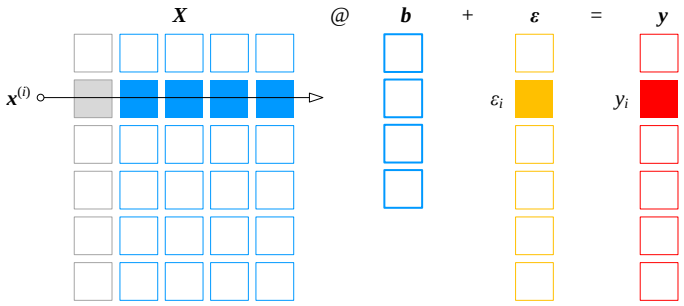


图 2. 计算第 i 个观测点的残差项

图 3 所示为多元 OLS 线性回归数据关系。也就是说，$\hat{\boldsymbol{y}}$ 可以看成设计矩阵 $\boldsymbol{X}$ 的列向量线性组合。

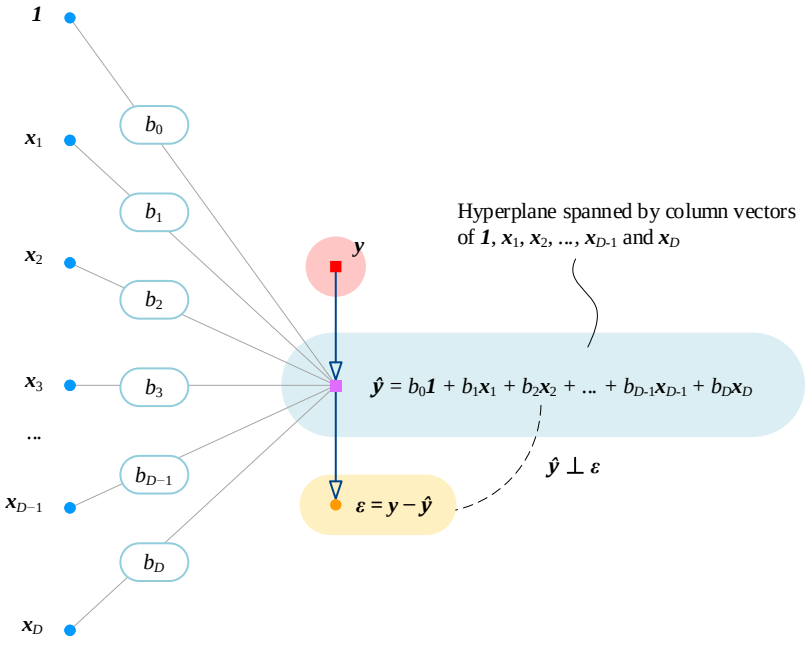


图 3. 多元 OLS 线性回归数据关系

⚠ 注意，矩阵 $\boldsymbol{X}$ 为 n 行，$D+1$ 列，第一列为全 1 列向量；增加一列全 1 列向量目的是为了引入常数项。

如图 4 所示，如果数据都已经中心化 (去均值)，则可以不必考虑常数项。

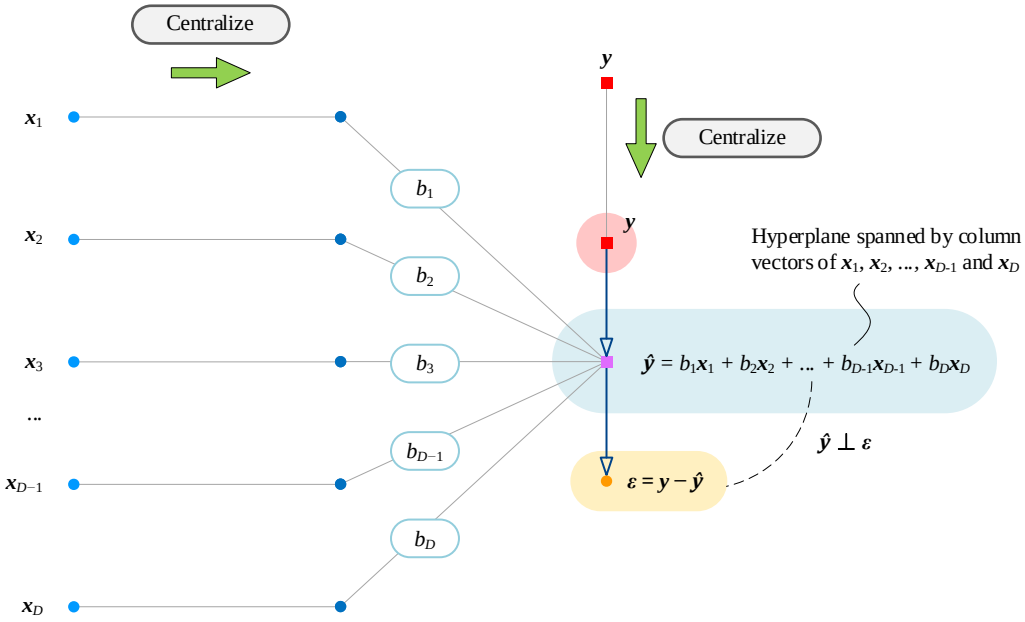


图 4. 多元 OLS 线性回归数据关系，中心化数据

10.2 优化问题：OLS

一般通过如下两种方式求得线性回归参数：

- **最小二乘法** (Ordinary Least Square, OLS)，因变量和拟合值之间的欧氏距离最小化；
- **最大似然概率估计** (Maximum Likelihood Estimation, MLE)，用样本数据反推最可能的模型参数值。

OLS 线性最小二乘法通过最小化残差值平方和 SSE 来计算得到最佳的拟合回归线参数：

$$\arg\min_{\boldsymbol{b}} \ \text{SSE} \tag{7}$$

对于多元线性回归，残差平方和 SSE 为：

$$\text{SSE} = \sum_{i=1}^{n}\left(\varepsilon^{(i)}\right)^2 = \boldsymbol{\varepsilon}\cdot\boldsymbol{\varepsilon} = \|\boldsymbol{\varepsilon}\|_2^2 = \boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\varepsilon} = (\boldsymbol{y}-\boldsymbol{X}\boldsymbol{b})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{b}) = \|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{b}\|_2^2 \tag{8}$$

OLS 多元线性优化问题的目标函数可以写成：

$$f(\boldsymbol{b}) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b})^{\mathrm{T}} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}) \tag{9}$$

$f(\boldsymbol{b})$ 可以整理为：

$$\begin{aligned} f(\boldsymbol{b}) &= (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b})^{\mathrm{T}} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}) \\ &= (\boldsymbol{y}^{\mathrm{T}} - \boldsymbol{b}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}})(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}) \\ &= \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{y}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{b} - \boldsymbol{b}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{b} \\ &= \underbrace{\boldsymbol{b}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{b}}_{\text{Quadratic term}} \underbrace{-2\boldsymbol{b}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}}_{\text{Linear term}} + \underbrace{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}}_{\text{Constant}} \end{aligned} \tag{10}$$

观察上式，发现 $f(\boldsymbol{b})$ 可以看成一个多元二次函数，含有二次项、一次项和常数项。

因此，对于二元回归，不考虑常数项系数 b_0 的话，b_1 和 b_2 构成的曲面 $f(b_1, b_2)$ 为椭圆抛物面，如图 5 所示。

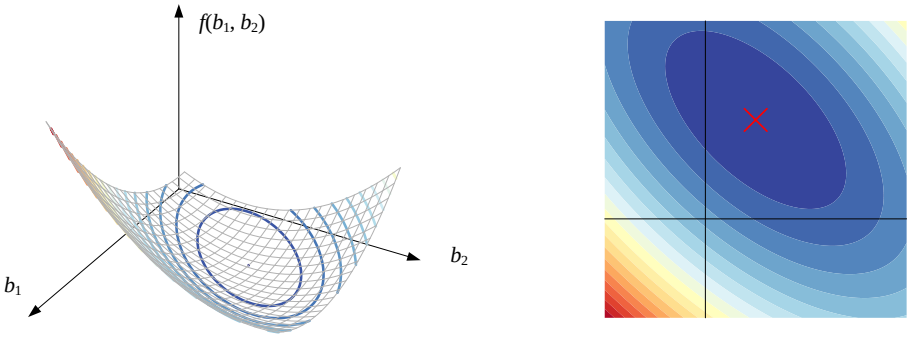


图 5. $f(b_1, b_2)$ 函数曲面

$f(\boldsymbol{b})$ 梯度向量如下：

$$\nabla f(\boldsymbol{b}) = \frac{\partial f(\boldsymbol{b})}{\partial \boldsymbol{b}} \tag{11}$$

$f(\boldsymbol{b})$ 为连续函数，取得极值时，梯度向量为零向量：

$$\nabla f(\boldsymbol{b}) = \boldsymbol{0} \quad \Rightarrow \quad \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{b} - \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{0} \tag{12}$$

如果 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 可逆，$\boldsymbol{b}$ 的解为：

$$\boldsymbol{b} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{13}$$

《矩阵力量》介绍过，如果 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 不可逆，可以用奇异值分解求伪逆。

$f(\boldsymbol{b})$ 的黑塞矩阵为：

$$\nabla^2 f(\boldsymbol{b}) = \frac{\partial^2 f(\boldsymbol{b})}{\partial \boldsymbol{b} \partial \boldsymbol{b}^{\mathrm{T}}} = 2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} \tag{14}$$

下面，判断 $f(\boldsymbol{b})$ 黑塞矩阵为正定矩阵，从而判定极值点为最小值点。

对于任意非零向量 $\boldsymbol{a}$，下式恒大于等于 0：

$$\boldsymbol{a}^{\mathrm{T}}\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)\boldsymbol{a}=\left(\boldsymbol{X}\boldsymbol{a}\right)^{\mathrm{T}}\left(\boldsymbol{X}\boldsymbol{a}\right)=\left\|\boldsymbol{X}\boldsymbol{a}\right\|^{2}\geq 0 \tag{15}$$

等号成立时，即 $\boldsymbol{X}\boldsymbol{a}=0$，即当 $\boldsymbol{X}$ 列向量线性相关，我们暂时不考虑这种情况。因此，对于 $\boldsymbol{X}$ 为列满秩，$f(\boldsymbol{b})$ 黑塞矩阵为正定矩阵，$f(\boldsymbol{b})$ 在极值点处取得最小值。

模型拟合值向量 $\hat{\boldsymbol{y}}$ 为：

$$\hat{\boldsymbol{y}}=\boldsymbol{X}\boldsymbol{b}=\boldsymbol{X}\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{16}$$

残差向量 $\boldsymbol{\varepsilon}$ 为：

$$\boldsymbol{\varepsilon}=\boldsymbol{y}-\boldsymbol{X}\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{17}$$

$\boldsymbol{X}\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}$ 为《矩阵力量》第 9 章介绍的**帽子矩阵** (hat matrix) $\boldsymbol{H}$，它常出现在矩阵投影运算中。

令，

$$\boldsymbol{H}=\boldsymbol{X}\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}} \tag{18}$$

帽子矩阵 $\boldsymbol{H}$ 为**幂等矩阵** (idempotent matrix)，幂等矩阵是指一个矩阵与自身相乘后仍等于它本身的矩阵，即满足 $\boldsymbol{H}^2=\boldsymbol{H}$。幂等矩阵在线性代数中有广泛的应用，特别是在投影、几何变换等领域。在投影中，幂等矩阵可以用来描述一个向量在一个子空间上的投影；在几何变换中，幂等矩阵可以用来描述一个对象在进行相应变换后仍等于它本身。最简单的幂等矩阵就是单位矩阵 $\boldsymbol{I}$，满足 $\boldsymbol{I}^2=\boldsymbol{I}$。

利用帽子矩阵 $\boldsymbol{H}$，

$$\begin{cases}\hat{\boldsymbol{y}}=\boldsymbol{H}\boldsymbol{y}\\ \boldsymbol{\varepsilon}=\left(\boldsymbol{I}-\boldsymbol{H}\right)\boldsymbol{y}\end{cases} \tag{19}$$

10.3 几何解释：投影

图 6 所示为多维空间视角下的数据矩阵；矩阵 $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_D]$ 每一列代表一个特征，每一列可以看做一个向量。

鸢尾花书《矩阵力量》一书中，我们反复探讨过这一点。

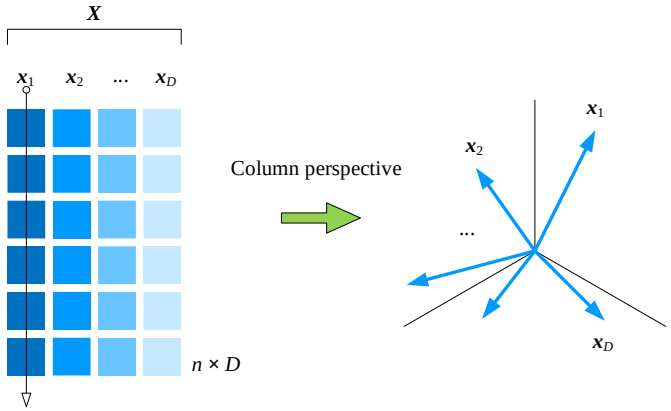


图 6. 多维空间视角下的矩阵 $\boldsymbol{X}$

不考虑常数项，预测值向量 $\hat{\boldsymbol{y}}$ 可以通过下式计算得到：

$$\hat{\boldsymbol{y}} = b_1\boldsymbol{x}_1 + b_2\boldsymbol{x}_2 + \cdots + b_D\boldsymbol{x}_D \tag{20}$$

(20) 说明，预测值向量 $\hat{\boldsymbol{y}}$ 是自变量向量 $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_D$ 的线性组合。如果 $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_D$ 构成一个超平面 H，$\hat{\boldsymbol{y}}$ 在 H 这个平面内。

有了这一思想，构造因变量向量 $\boldsymbol{y}$ 和自变量向量 $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_D$ 的线性回归模型，相当于 $\boldsymbol{y}$ 向 $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_D$ 构成的超平面 H 投影。如图 7 所示，预测值向量 $\hat{\boldsymbol{y}}$ 是因变量向量 $\boldsymbol{y}$ 在 H 的投影结果：

$$\boldsymbol{y} = \hat{\boldsymbol{y}} + \boldsymbol{\varepsilon} \tag{21}$$

简单来说，从向量投影的角度来理解多元线性回归，可以将回归问题看作是将因变量向量在自变量向量所张成的子空间上的投影。

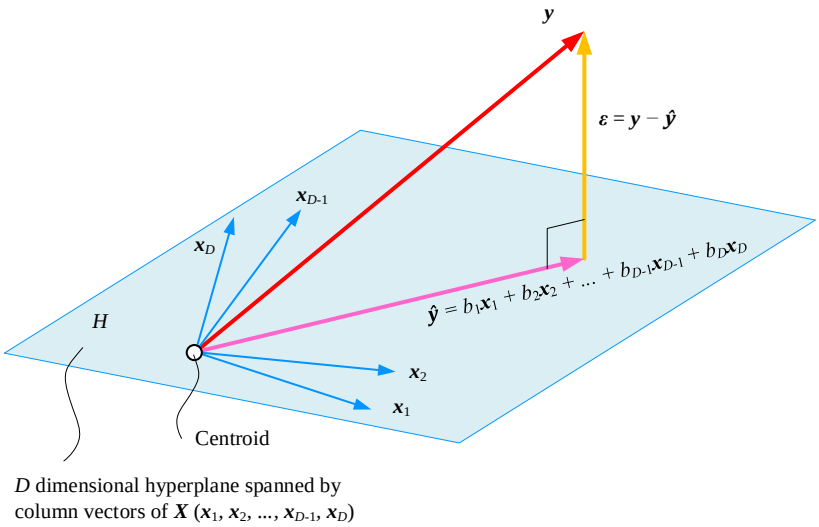


图 7. 几何角度解释多元最小二乘法线性回归

而残差项向量 $\boldsymbol{\varepsilon}$ 是预测值向量 $\hat{\boldsymbol{y}}$ 是因变量向量 $\boldsymbol{y}$ 两者之差：

$$\boldsymbol{\varepsilon} = \boldsymbol{y} - \hat{\boldsymbol{y}} \tag{22}$$

残差项向量 $\boldsymbol{\varepsilon}$ 垂直于 $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_D$ 构成的超平面 H。

由上所述，残差 $\boldsymbol{\varepsilon}$ ($\boldsymbol{\varepsilon} = \boldsymbol{y} - \hat{\boldsymbol{y}}$) 是无法通过 ($\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{D-1}, \boldsymbol{x}_D$) 解释部分向量，垂直于超平面：

$$\boldsymbol{\varepsilon} \perp \boldsymbol{X} \quad \Rightarrow \quad \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\varepsilon} = 0 \tag{23}$$

得到

$$\boldsymbol{X}^{\mathrm{T}}\left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}\right) = 0 \quad \Rightarrow \quad \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{b} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{24}$$

这和上一节得到的结果完全一致，但是从几何视角看 OLS，让求解过程变得非常简洁。

请大家再次注意，只有 $\boldsymbol{X}$ 为列满秩时，$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 才存在逆。

此外，我们可以很容易在 $\boldsymbol{X}$ 最左侧加入一列全 1 向量 $\boldsymbol{1}$，残差项向量 $\boldsymbol{\varepsilon}$ 则垂直于 $\boldsymbol{1}, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_D$ 构成的超平面 H。

《统计至简》介绍过 OLS 线性回归假设条件。OLS 线性回归的假设条件是用来保证模型的有效性和可靠性。简单来说，这些假设条件主要包括线性关系、正态分布、同方差性、独立性和残差之和为零。

首先，线性关系假设要求因变量和自变量之间的关系是线性的，即在自变量变化时，因变量的变化量是按照线性关系变化的。这个假设是 OLS 回归分析的前提条件，否则回归结果将会失真。

其次，正态分布假设要求模型的残差应该满足正态分布。正态分布是概率论和统计学中最为重要的分布之一，如果残差不满足正态分布，可能会导致回归结果失真。

同方差性假设要求残差的方差在各个自变量取值下都相等。如果残差的方差不相等，会导致回归结果的可靠性下降。

独立性假设要求各个观测值之间是独立的，即一个观测值的取值不受其他观测值的影响。如果存在相关性，回归结果可能会失真。

最后，残差之和为零要求模型的残差的总和为零，这是保证回归分析的正确性的必要条件。

总之，这些假设条件对于 OLS 线性回归的结果具有重要影响，需要在回归分析中进行检验和确认。

表 1 所示为用矩阵方式表达 OLS 线性回归假设。

表 1. 用矩阵运算表达 OLS 线性回归假设

假设	矩阵表达
线性模型	$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{b} + \boldsymbol{\varepsilon}$

残差服从正态分布	$\boldsymbol{\varepsilon}\|\boldsymbol{X} \sim N\left(\boldsymbol{0}, \hat{\sigma}^2 \boldsymbol{I}\right)$
残差期望值为 0	$\mathrm{E}\left(\boldsymbol{\varepsilon}\|\boldsymbol{X}\right)=\boldsymbol{0}$
残差同方差性	$\operatorname{var}\left(\boldsymbol{\varepsilon}\|\boldsymbol{X}\right)=\begin{bmatrix} \operatorname{var}\left(\varepsilon^{(1)}\right) & \operatorname{cov}\left(\varepsilon^{(1)},\varepsilon^{(2)}\right) & \cdots & \operatorname{cov}\left(\varepsilon^{(1)},\varepsilon^{(n)}\right) \\ \operatorname{cov}\left(\varepsilon^{(2)},\varepsilon^{(1)}\right) & \operatorname{var}\left(\varepsilon^{(2)}\right) & \cdots & \operatorname{cov}\left(\varepsilon^{(2)},\varepsilon^{(n)}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \operatorname{cov}\left(\varepsilon^{(n)},\varepsilon^{(1)}\right) & \operatorname{cov}\left(\varepsilon^{(n)},\varepsilon^{(2)}\right) & \cdots & \operatorname{var}\left(\varepsilon^{(n)}\right) \end{bmatrix}=\hat{\sigma}^2 \boldsymbol{I}$
矩阵 $\boldsymbol{X}$ 不存在多重共线性	$\operatorname{rank}\left(\boldsymbol{X}\right)=D+1$ $\det\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)\neq 0$

10.4 二元线性回归

为了方便大家理解，本节用实例讲解二元线性回归。

二元线性回归解析式为：

$$\hat{\boldsymbol{y}}=b_0\boldsymbol{1}+b_1\boldsymbol{x}_1+b_2\boldsymbol{x}_2 \tag{25}$$

图 8 所示为二元 OLS 线性回归数据关系。

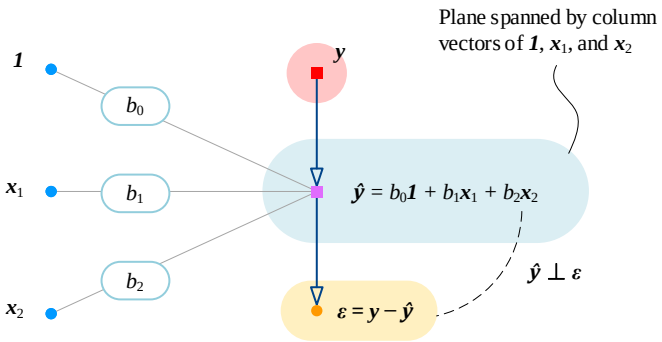


图 8. 二元 OLS 线性回归数据关系

本节介绍利用两个股票日收益率解释 S&P 500 日收益率。图 9 所示为参与回归数据 [$\boldsymbol{y}$, $\boldsymbol{x}_1$, $\boldsymbol{x}_2$] 的散点图。

图 10 所示为 [$\boldsymbol{y}$, $\boldsymbol{x}_1$, $\boldsymbol{x}_2$] 数据的成对特征分析图。

图 11 所示为 [$\boldsymbol{y}$, $\boldsymbol{x}_1$, $\boldsymbol{x}_2$] 数据的协方差矩阵、相关性和夹角热图。

图 12 所示为二元 OLS 线性回归结果。图 13 所示为三维数据散点图和回归平面。

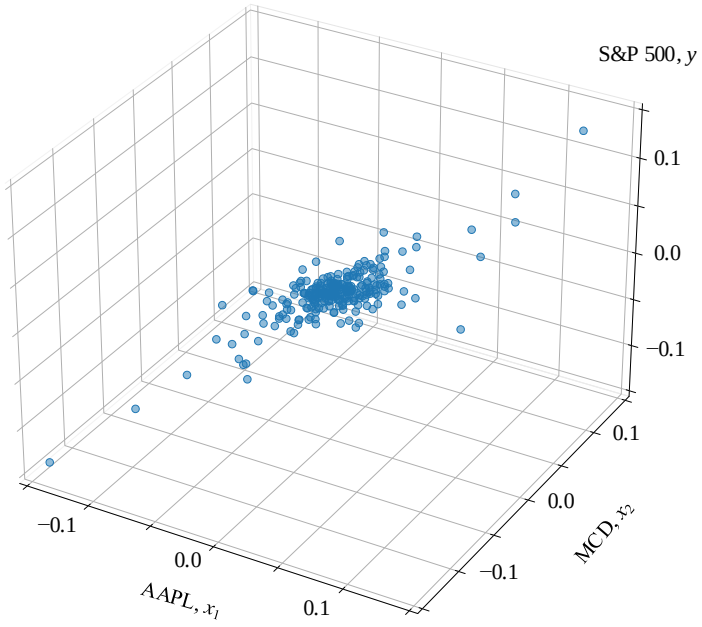


图 9. 二元线性回归数据

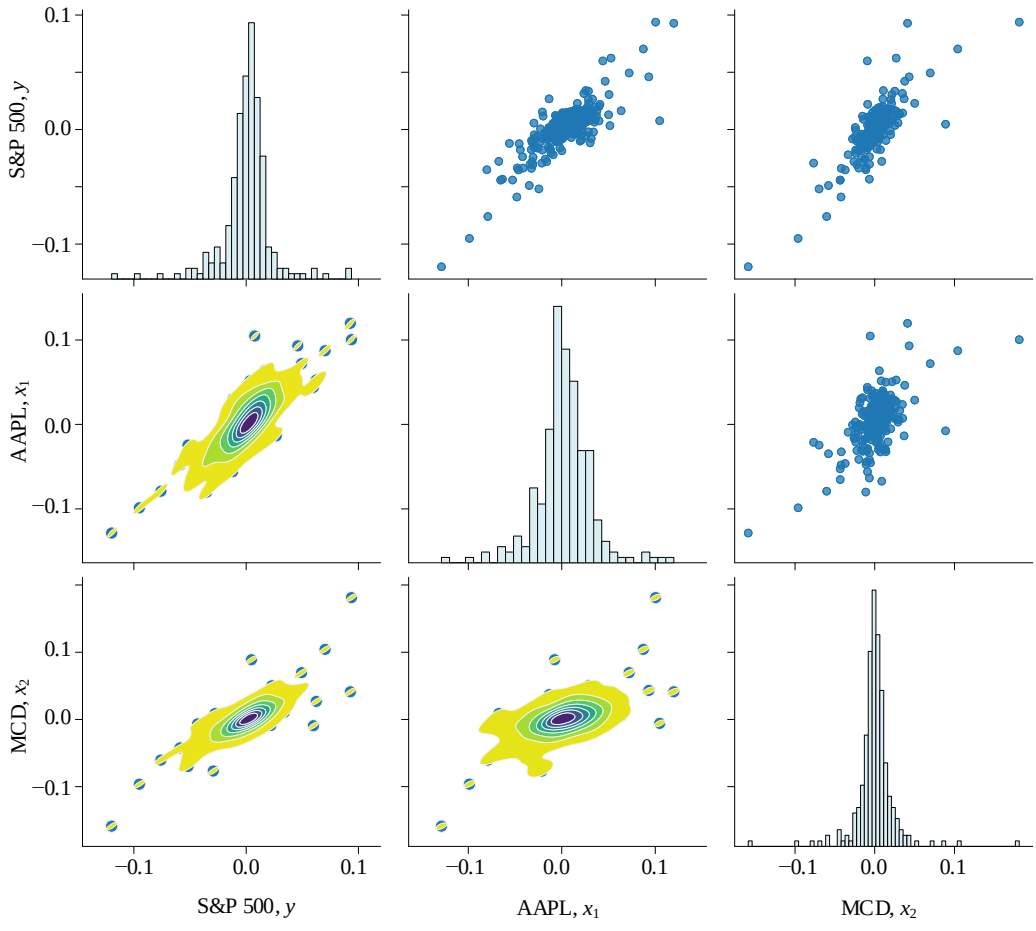


图 10. 二元线性回归数据 $[y, x_1, x_2]$ 成对特征分析图

图 11. $[y, x_1, x_2]$ 数据的协方差矩阵、相关性和夹角热图

OLS Regression Results

Dep. Variable:	SP500	R-squared:	0.830
Model:	OLS	Adj. R-squared:	0.829
Method:	Least Squares	F-statistic:	607.4
Date:	XXXXXXXXXXXXXXXX	Prob (F-statistic):	1.69e-96
Time:	XXXXXXXXXXXXXXXX	Log-Likelihood:	831.06
No. Observations:	252	AIC:	-1656.
Df Residuals:	249	BIC:	-1646.
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.0006	0.001	-0.984	0.326	-0.002	0.001
AAPL	0.3977	0.024	16.326	0.000	0.350	0.446
MCD	0.4096	0.028	14.442	0.000	0.354	0.465

Omnibus:	37.744	Durbin-Watson:	1.991
Prob(Omnibus):	0.000	Jarque-Bera (JB):	157.711
Skew:	0.492	Prob(JB):	5.67e-35
Kurtosis:	6.749	Cond. No.	59.4

图 12. 二元 OLS 线性回归分析结果

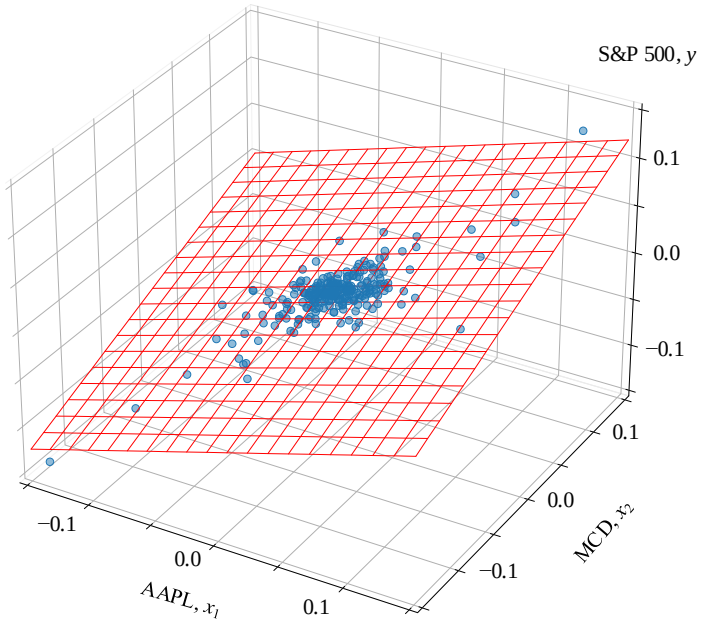


图 13. 三维空间，回归平面

Bk6_Ch10_01.py 完成本节二元线性回归。

10.5 多元回归

本节介绍一个多元回归问题，构造多元 OLS 线性回归模型用 12 只股票日收益率预测 S&P 500 日收益率。图 14 所示股价数据。

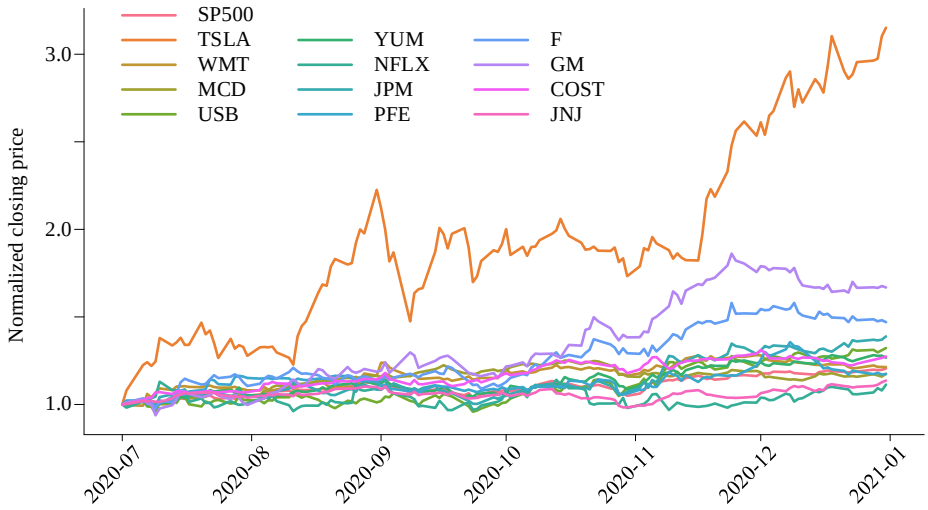


图 14. 股价数据，起始值归一化

根据股价水平计算得到的日收益率。图 15 所示为日收益率热图。图 16 所示为[$\boldsymbol{y}$, $\boldsymbol{X}$] 数据协方差矩阵。图 17 所示为均方差 (即波动率) 直方图。

图 18 所示为 [$\boldsymbol{y}$, $\boldsymbol{X}$] 数据相关性系数矩阵热图。图 19 所示为几只不同股票股价收益率和 S&P 500 收益率相关性系数柱状图。利用余弦相似性，根据相关性系数矩阵，可以计算得到[$\boldsymbol{y}$, $\boldsymbol{X}$] 标准差向量夹角，矩阵热图如图 20 所示。图 21 所示为多元 OLS 线性回归解。

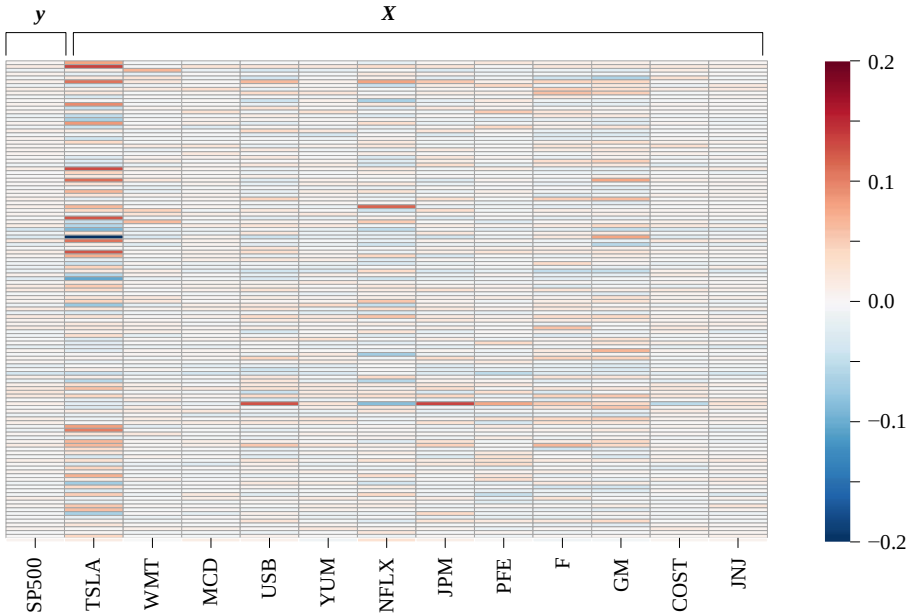


图 15. [$\boldsymbol{y}$, $\boldsymbol{X}$] 日收益率热图

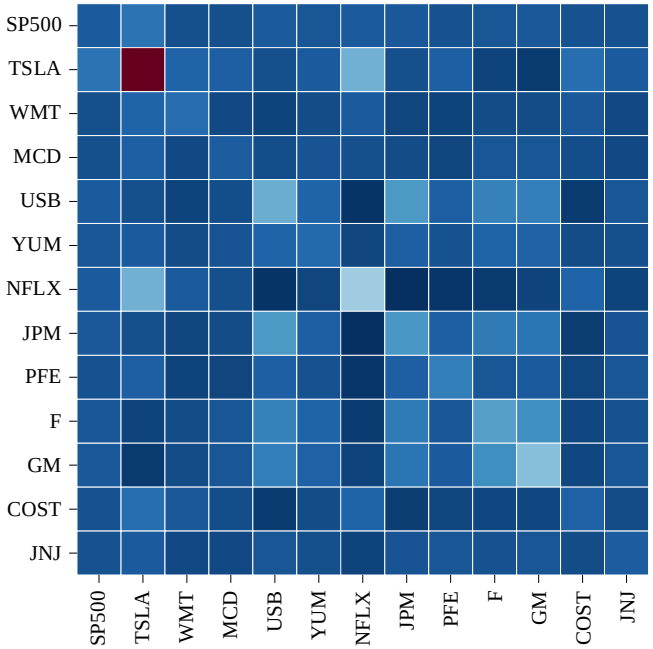


图 16. [$\boldsymbol{y}$, $\boldsymbol{X}$] 数据协方差矩阵

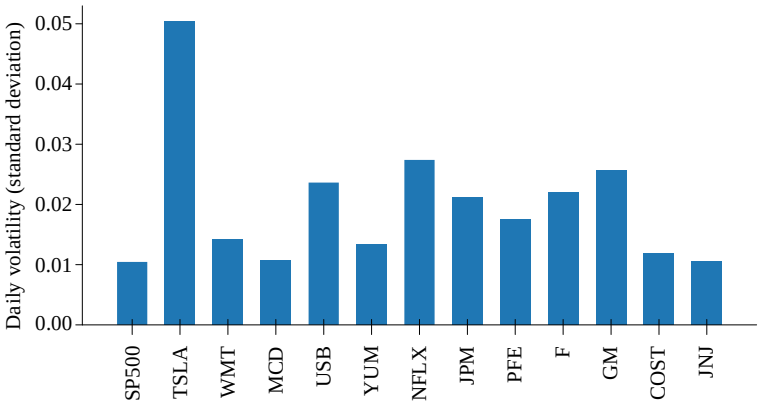


图 17. 日波动率柱状图

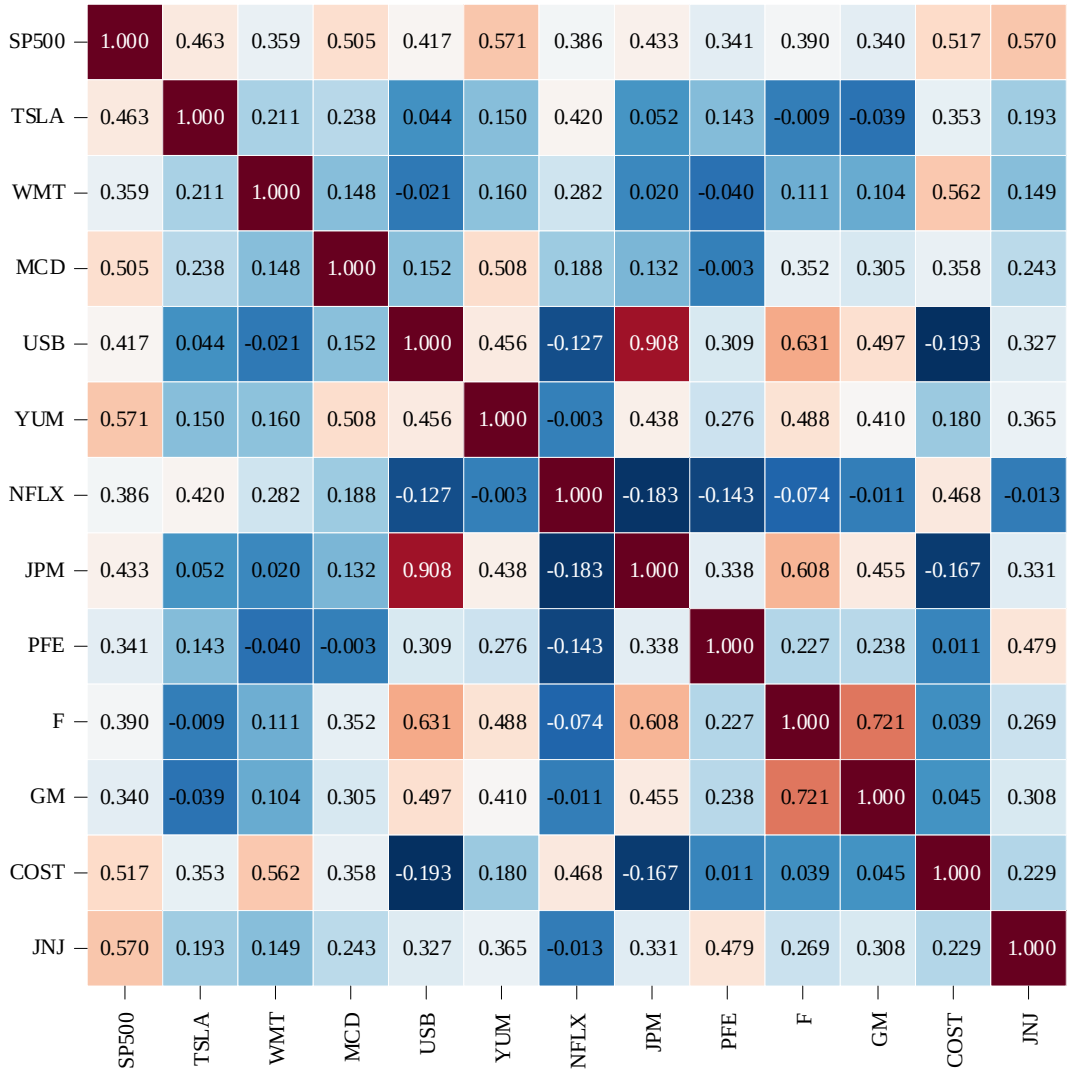


图 18. $[\boldsymbol{y}, \boldsymbol{X}]$ 数据相关性系数矩阵热图

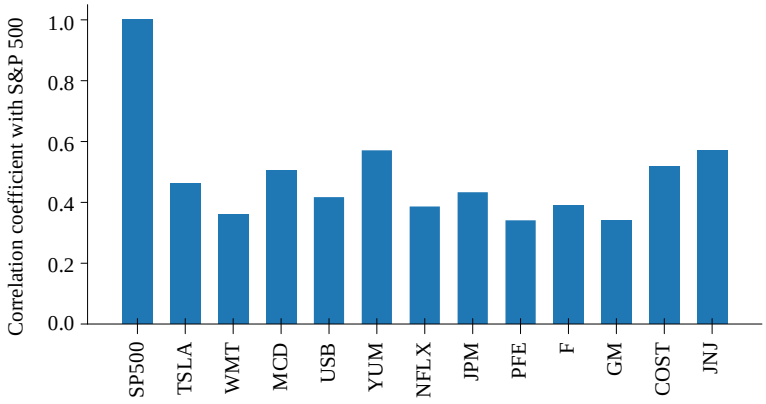


图 19. 股价收益率和 S&P 500 收益率相关性系数柱状图

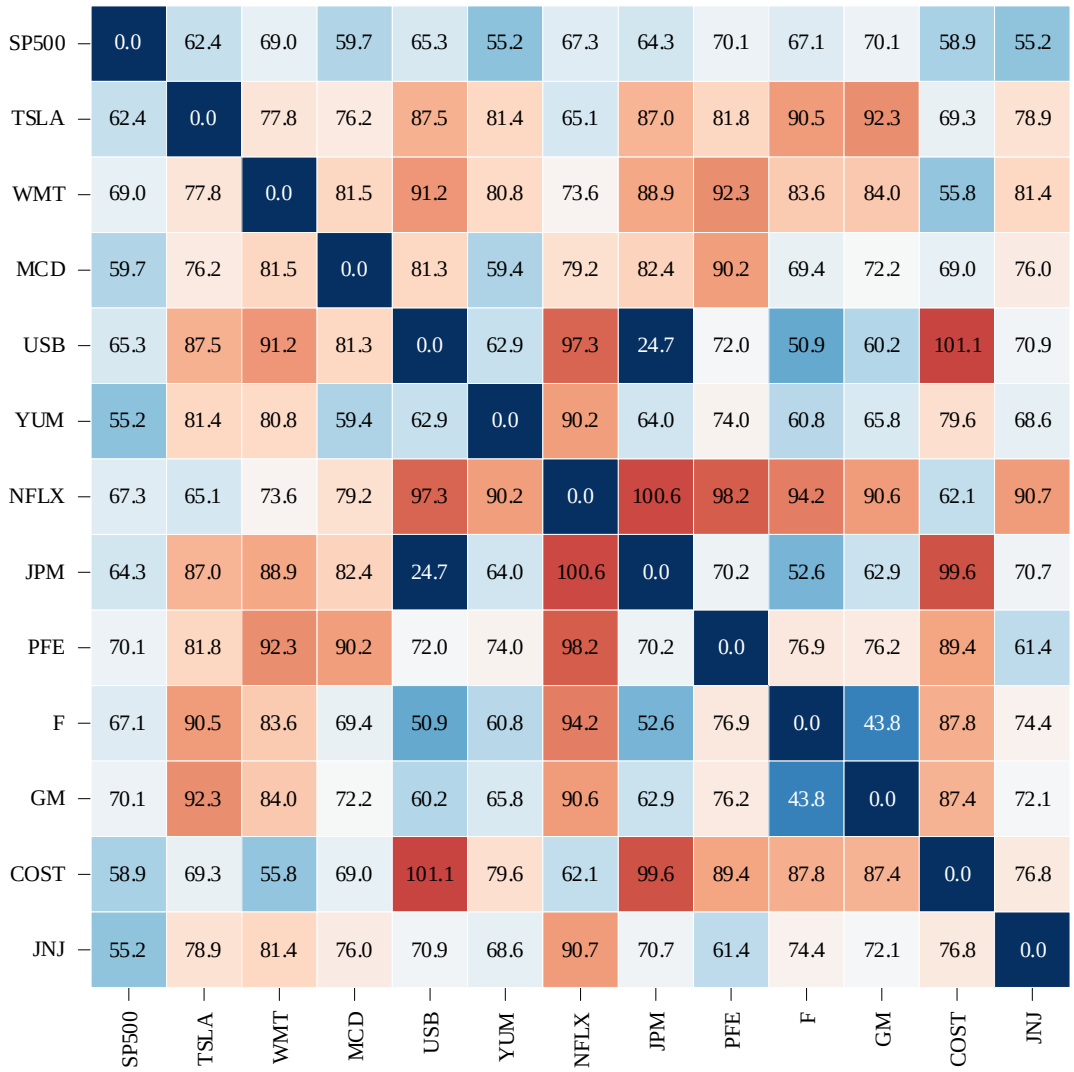


图 20. $[\boldsymbol{y}, \boldsymbol{X}]$ 标准差向量夹角矩阵热图，余弦相似性

OLS Regression Results

Dep. Variable:	SP500	R-squared:	0.774
Model:	OLS	Adj. R-squared:	0.750
Method:	Least Squares	F-statistic:	32.48
Date:	XXXXXXXXXXXXXXX	Prob (F-statistic):	3.03e-31
Time:	XXXXXXXXXXXXXXX	Log-Likelihood:	493.88
No. Observations:	127	AIC:	-961.8
Df Residuals:	114	BIC:	-924.8
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.0005	0.000	-1.038	0.302	-0.001	0.000
TSLA	0.0248	0.011	2.248	0.027	0.003	0.047
WMT	0.0272	0.041	0.667	0.506	-0.054	0.108
MCD	0.1435	0.057	2.536	0.013	0.031	0.256
USB	0.0164	0.051	0.322	0.748	-0.084	0.117
YUM	0.1469	0.047	3.114	0.002	0.053	0.240
NFLX	0.0972	0.021	4.539	0.000	0.055	0.140
JPM	0.1415	0.055	2.583	0.011	0.033	0.250
PFE	0.0546	0.033	1.662	0.099	-0.010	0.120
F	-0.0068	0.036	-0.187	0.852	-0.078	0.065
GM	-0.0105	0.027	-0.388	0.699	-0.064	0.043
COST	0.2176	0.059	3.713	0.000	0.101	0.334
JNJ	0.2414	0.056	4.350	0.000	0.131	0.351

Omnibus:	7.561	Durbin-Watson:	1.862
Prob(Omnibus):	0.023	Jarque-Bera (JB):	8.445
Skew:	0.400	Prob(JB):	0.0147
Kurtosis:	3.978	Cond. No.	156.

图 21. 多元 OLS 线性回归分析结果

Bk6_Ch10_02.py 完成本节多元线性回归。

10.6 正交关系

第一个直角三角形

通过上一章学习，大家都很清楚第一个勾股关系：

$$\underbrace{\left\|\boldsymbol{y}-\bar{y}\boldsymbol{1}\right\|_2^2}_{\text{SST}}=\underbrace{\left\|\hat{\boldsymbol{y}}-\bar{y}\boldsymbol{1}\right\|_2^2}_{\text{SSR}}+\underbrace{\left\|\boldsymbol{y}-\hat{\boldsymbol{y}}\right\|_2^2}_{\text{SSE}} \tag{26}$$

具体如图 22 所示。上一章提到这一个直角三角形可以帮助我们解释 R^2。

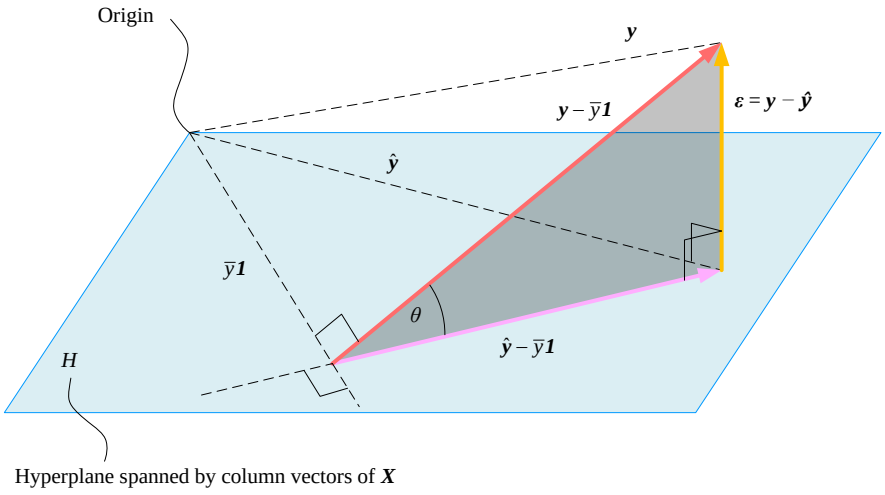


图 22. 第一个直角三角形

第二个直角三角形

除了 **(26)** 这个重要的直角三角形的勾股定理之外，还有另外一个重要的直角三角形勾股定理关系。

$$\|\boldsymbol{y}\|_2^2 = \|\hat{\boldsymbol{y}}\|_2^2 + \|\boldsymbol{y}-\hat{\boldsymbol{y}}\|_2^2 = \|\hat{\boldsymbol{y}}\|_2^2 + \|\boldsymbol{\varepsilon}\|_2^2 \tag{27}$$

具体如图 23 所示。图 23 这个直角很容易理解。残差向量 $\boldsymbol{\varepsilon}$ 垂直于超平面 H 内的一切向量，显然 $\boldsymbol{\varepsilon}$ 垂直 $\hat{\boldsymbol{y}}$ 。

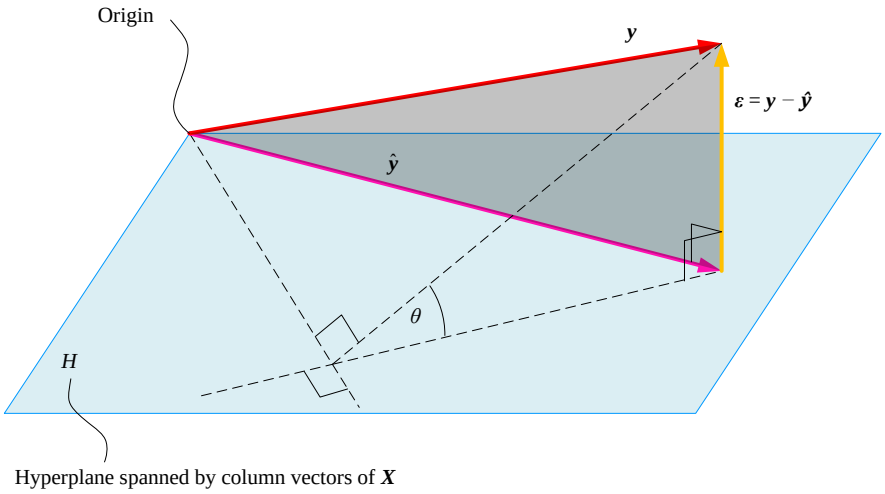


图 23. 第二个直角三角形

第三个直角三角形

此外，《矩阵力量》第 22 章介绍过，向量 $\boldsymbol{y}-\bar{y}\boldsymbol{1}$ 垂直于向量 $\bar{y}\boldsymbol{1}$：

$$(\bar{y}\boldsymbol{1})^{\mathrm{T}}(\boldsymbol{y}-\bar{y}\boldsymbol{1})=0 \tag{28}$$

具体如图 24 所示。上式体现的核心思想就是 $\boldsymbol{y}$ 中可以被均值解释的部分为 $\bar{y}\boldsymbol{1}$ 。

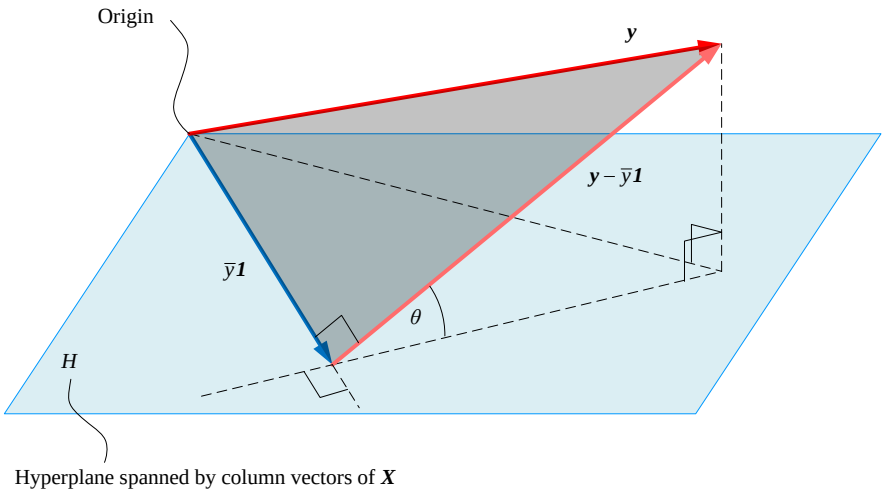


图 24. 第三个直角三角形

第四个直角三角形

OLS 假设残差之和为 0：

$$\sum_{i=1}^{n}\varepsilon^{(i)}=0 \tag{29}$$

注意，如果总残差不为 0，就说明预测值的总和与实际观测值的总和不相等，这意味着模型存在偏差，不能很好地解释数据。

对应向量运算：

$$\boldsymbol{1}^{\mathrm{T}}\boldsymbol{\varepsilon}=\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{1}=0 \tag{30}$$

残差向量可以写成：

$$\boldsymbol{\varepsilon}=\boldsymbol{y}-\hat{\boldsymbol{y}}=\boldsymbol{y}-\bar{y}\boldsymbol{1}-(\hat{\boldsymbol{y}}-\bar{y}\boldsymbol{1}) \tag{31}$$

上式左乘 $\boldsymbol{1}^{\mathrm{T}}$，得到：

$$\underbrace{\boldsymbol{1}^{\mathrm{T}}\boldsymbol{\varepsilon}}_{0}=\underbrace{\boldsymbol{1}^{\mathrm{T}}\left(\boldsymbol{y}-\bar{y}\boldsymbol{1}\right)}_{0}-\boldsymbol{1}^{\mathrm{T}}\left(\hat{\boldsymbol{y}}-\bar{y}\boldsymbol{1}\right) \tag{32}$$

即

$$\boldsymbol{1}^{\mathrm{T}}\left(\hat{\boldsymbol{y}}-\bar{y}\boldsymbol{1}\right)=0 \tag{33}$$

也就是说，如图 25 所示，$\hat{\boldsymbol{y}}-\bar{y}\boldsymbol{1}$ 垂直于向量 $\bar{y}\boldsymbol{1}$：

$$\bar{y}\boldsymbol{1}^{\mathrm{T}}\left(\hat{\boldsymbol{y}}-\bar{y}\boldsymbol{1}\right)=0 \tag{34}$$

上式体现的核心思想就是 $\hat{\boldsymbol{y}}$ 的均值也是 $\bar{y}$。

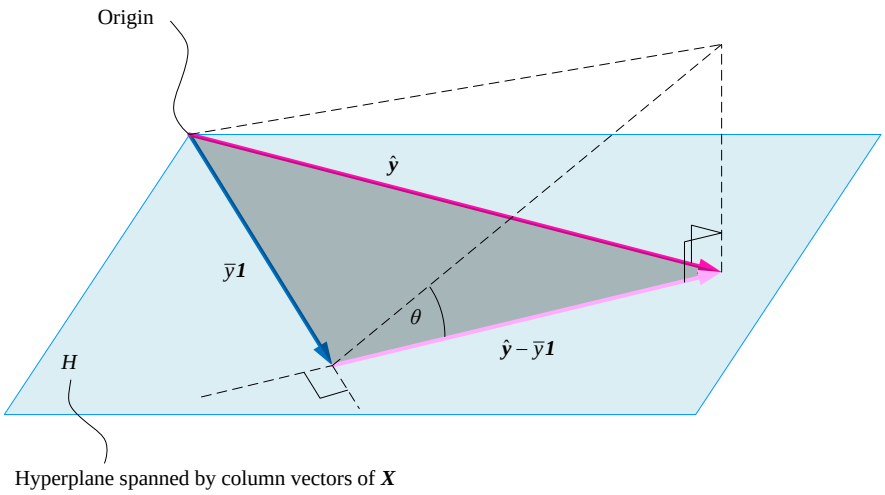


图 25. 第四个直角三角形

10.7 三个平方和

这一节介绍对于多元 OLS 线性回归，如何求解 SST、SSR 和 SSE 这三个平方和。

对于多元 OLS 线性回归模型，SST 可以通过矩阵运算求得：

$$\mathrm{SST}=\boldsymbol{y}^{\mathrm{T}}\left(\boldsymbol{I}-\frac{\boldsymbol{J}}{n}\right)\boldsymbol{y} \tag{35}$$

其中矩阵 $\boldsymbol{J}$ 为全 1 方阵，形状为 $n\times n$：

$$\boldsymbol{J}_{n\times n}=\boldsymbol{1}\boldsymbol{1}^{\mathrm{T}}=\begin{bmatrix}1 & 1 & \cdots & 1\\ 1 & 1 & \cdots & 1\\ \vdots & \vdots & \ddots & \vdots\\ 1 & 1 & \cdots & 1\end{bmatrix} \tag{36}$$

SSR 可以通过矩阵运算求得：

$$\text{SSR} = \boldsymbol{y}^{\mathrm{T}}\left(\boldsymbol{H} - \frac{\boldsymbol{J}}{n}\right)\boldsymbol{y} \tag{37}$$

其中矩阵 $\boldsymbol{H}$ 为本书前文所讲的帽子矩阵，形状为 $n \times n$：

$$\boldsymbol{H} = \boldsymbol{X}\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}} \tag{38}$$

同样，对于多元 OLS 线性回归模型，SSE 可以通过矩阵运算求得：

$$\text{SSE} = \boldsymbol{y}^{\mathrm{T}}\left(\boldsymbol{I} - \boldsymbol{H}\right)\boldsymbol{y} \tag{39}$$

对于多元 OLS 线性回归模型，MSE 的矩阵运算为：

$$\begin{aligned}\text{MSE} &= \frac{\left\|\left(\boldsymbol{I} - \boldsymbol{H}\right)\boldsymbol{y}\right\|_2^2}{n-k} \\ &= \frac{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - 2\boldsymbol{y}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{y} + \boldsymbol{y}^{\mathrm{T}}\boldsymbol{H}^2\boldsymbol{y}}{n-k} \\ &= \frac{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{y}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{y}}{n-k} \\ &= \frac{\boldsymbol{y}^{\mathrm{T}}\left(\boldsymbol{I} - \boldsymbol{H}\right)\boldsymbol{y}}{n-k}\end{aligned} \tag{40}$$

上式推导过程采用帽子矩阵重要的性质。

10.8 t检验

对于多元 OLS 线性回归模型，模型系数 b_0、b_1、b_2 … b_D 的协方差矩阵 $\boldsymbol{C}$ 可以通过下式计算得到：

$$\boldsymbol{C} = \hat{\sigma}^2\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1} \tag{41}$$

其中，

$$\hat{\sigma}^2 = \text{MSE} = \frac{\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\varepsilon}}{n-k} \tag{42}$$

矩阵 $\boldsymbol{C}$ 的对角线元素 $\boldsymbol{C}_{j+1,\,j+1}$ 为 $\hat{b}_j$ 的方差，非对角线元素为 $\hat{b}_j$ 和 $\hat{b}_k$ 的协方差。

$\hat{b}_j$ 的标准误 $\text{SE}\left(\hat{b}_j\right)$ 为：

$$\text{SE}\left(\hat{b}_j\right) = \sqrt{\boldsymbol{C}_{j+1,\,j+1}} \tag{43}$$

对于多元线性回归，假设检验原假设和备择假设分别为：

$$\begin{cases} H_0: b_j = b_{j,0} \\ H_1: b_j \neq b_{j,0} \end{cases} \tag{44}$$

b_j 的 t 检验统计值：

$$T_j = \frac{\hat{b}_j - b_{j,0}}{\mathrm{SE}\left(\hat{b}_j\right)} \tag{45}$$

类似地，如果下式成立，接受零假设 H_0：

$$-t_{1-\alpha/2,\, n-k} < T_j < t_{1-\alpha/2,\, n-k} \tag{46}$$

否则，则拒绝零假设 H_0。

系数 b_j 的 $1-\alpha$ 置信区间为：

$$\hat{b}_j \pm t_{1-\alpha/2,\, n-k} \cdot \mathrm{SE}\left(\hat{b}_j\right) \tag{47}$$

对于多元 OLS 线性模型，预测值 $\hat{y}^{(i)}$，的 $1-\alpha$ 置信区间：

$$\hat{y}^{(i)} \pm t_{1-\alpha/2,\, n-2} \cdot \sqrt{\mathrm{MSE}} \cdot \sqrt{\boldsymbol{x}^{(i)} \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1} \left(\boldsymbol{x}^{(i)}\right)^{\mathrm{T}}} \tag{48}$$

$\boldsymbol{x}^{(i)}$ 为矩阵 $\boldsymbol{X}$ 的第 i 行：

$$\boldsymbol{x}^{(i)} = \begin{bmatrix} 1 & x_{i,1} & x_{i,2} & \cdots & x_{i,D} \end{bmatrix} \tag{49}$$

类似地，对于多元 OLS 线性回归模型，y_p 的预测区间估计为：

$$\hat{y}^{(i)} \pm t_{1-\alpha/2,\, n-2} \cdot \sqrt{\mathrm{MSE}} \cdot \sqrt{1 + \boldsymbol{x}^{(i)} \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1} \left(\boldsymbol{x}^{(i)}\right)^{\mathrm{T}}} \tag{50}$$

10.9 多重共线性

线性回归模型的解释变量不满足相互独立的基本假设前提下，如果模型的解释变量存在多重共线性，将导致最小二乘法得到的模型参数估计量非有效且方差变大，参数估计量经济含义不合理等。

上一章介绍过采用**条件数** (Condition number) 来判定多重共线性。对 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 进行特征值分解，得到最大特征值 $\lambda_{\max}$ 和最小特征值 $\lambda_{\min}$。条件数的定义为两者的比值的平方根。条件数小于 30，可以不必担心多重共线性。

如果 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 可逆，$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 的行列式值不为 0：

$$\det\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right) \neq 0 \tag{51}$$

这里再介绍一个评价共线性的度量指标，**方差膨胀因子** (variance inflation factor, VIF)，也称为**方差扩大因子**。

一个还有 n 个解释变量的矩阵 $\hat{\boldsymbol{X}}_t$，对于其中的任意解释变量 $\{X_{i,t}\}$，其对应的方差膨胀因子 VIF_i 可由下式计算：

$$\mathrm{VIF}_i = \frac{1}{1-R_i^2} \tag{52}$$

其中 R_i^2 是解释变量 $\{X_{i,t}\}$ 与其解释变量 $\{X_{j,t}\}, j \neq i$ 回归模型的决定系数：

$$X_{i,t} = \alpha_0 + \sum_{j=1, j\neq i}^{n} \alpha_j X_{j,t} + \varepsilon_t \tag{53}$$

当某个变量 $\{X_{i,t}\}$ 能被其他变量完全线性解释时，R_i^2 的值趋近于 1，VIF_i 的值将趋近于无穷大；所以，各个变量的 VIF 值越小，说明共线性越弱。最常用的 VIF 阈值是 10，即解释变量的 VIF 值都不大于 10 时，认为共线性在可接受范围内；此外，VIF ≤ 5 也是比较常见的、但相对而言更为严格的判断标准。

10.10 条件概率视角看多元线性回归

《统计至简》第 12 章介绍过，多元线性回归本质上就是条件概率中的条件期望值。

如果随机变量向量 $\boldsymbol{\chi}$ 和 $\boldsymbol{\gamma}$ 服从多维高斯分布：

$$\begin{bmatrix} \boldsymbol{\chi} \\ \boldsymbol{\gamma} \end{bmatrix} \sim N\left(\begin{bmatrix} \boldsymbol{\mu}_{\chi} \\ \boldsymbol{\mu}_{\gamma} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{\chi\chi} & \boldsymbol{\Sigma}_{\chi\gamma} \\ \boldsymbol{\Sigma}_{\gamma\chi} & \boldsymbol{\Sigma}_{\gamma\gamma} \end{bmatrix} \right) \tag{54}$$

其中，$\boldsymbol{\chi}$ 为随机变量 X_i 构成的列向量，$\boldsymbol{\gamma}$ 为随机变量 Y_j 构成的列向量：

$$\boldsymbol{\chi} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_D \end{bmatrix}, \quad \boldsymbol{\gamma} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_M \end{bmatrix} \tag{55}$$

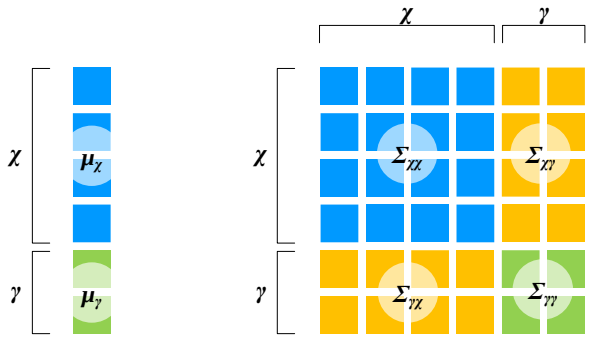


图 26. 均值向量、协方差矩阵形状，图片来自鸢尾花书《统计至简》第 12 章

如图 27 所示，给定 $\boldsymbol{\chi} = \boldsymbol{x}$ 的条件下 $\boldsymbol{\gamma}$ 的条件期望为：

$$\mathrm{E}\left(\boldsymbol{\gamma} \middle| \boldsymbol{\chi} = \boldsymbol{x}\right) = \boldsymbol{\mu}_{\boldsymbol{\gamma}|\boldsymbol{\chi}=\boldsymbol{x}} = \boldsymbol{\Sigma}_{\boldsymbol{\gamma}\boldsymbol{\chi}} \boldsymbol{\Sigma}_{\boldsymbol{\chi}\boldsymbol{\chi}}^{-1} \left(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{\chi}}\right) + \boldsymbol{\mu}_{\boldsymbol{\gamma}} \tag{56}$$

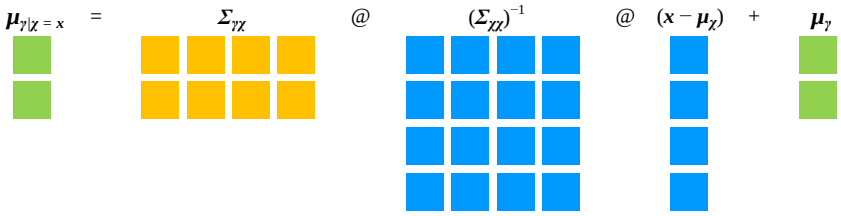


图 27. 给定 $\boldsymbol{\chi} = \boldsymbol{x}$ 的条件下 $\boldsymbol{\gamma}$ 的期望值的矩阵运算，图片来自鸢尾花书《统计至简》第 12 章

对于本例，我们对 (56) 进行转置得到：

$$\mu_{y|x} = \mathrm{E}\left(\boldsymbol{y}\right) + \left(\boldsymbol{x} - \mathrm{E}\left(\boldsymbol{X}\right)\right) \underbrace{\left(\boldsymbol{\Sigma}_{\boldsymbol{X}\boldsymbol{X}}\right)^{-1} \boldsymbol{\Sigma}_{\boldsymbol{X}\boldsymbol{y}}}_{\boldsymbol{b}} \tag{57}$$

$[\boldsymbol{y}, \boldsymbol{X}]$ 对应的协方差矩阵如图 28 所示。图 29 为对 $\boldsymbol{\Sigma}_{\boldsymbol{X}\boldsymbol{X}}$ 求逆。

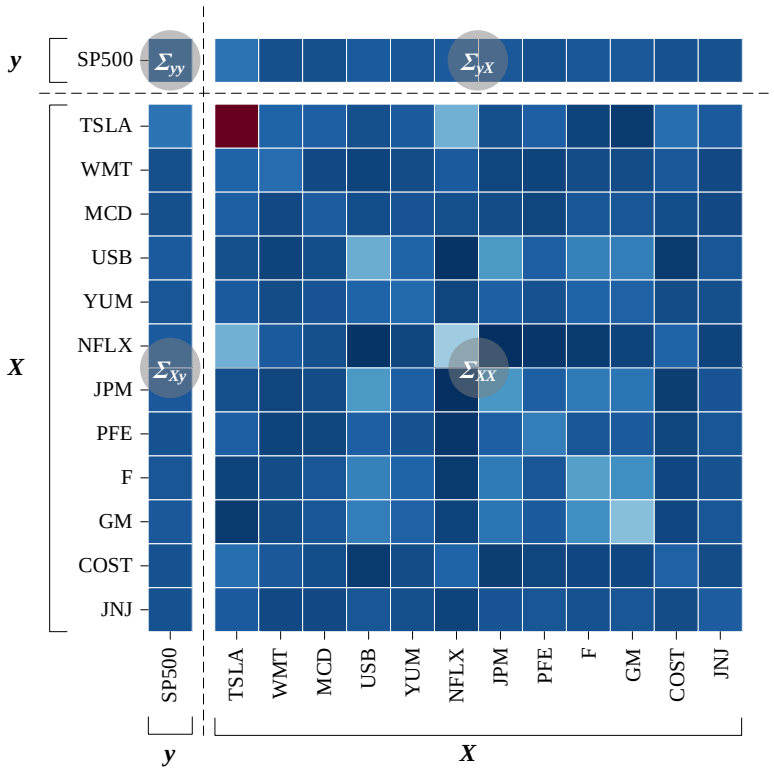


图 28. $[\boldsymbol{y}, \boldsymbol{X}]$ 协方差矩阵

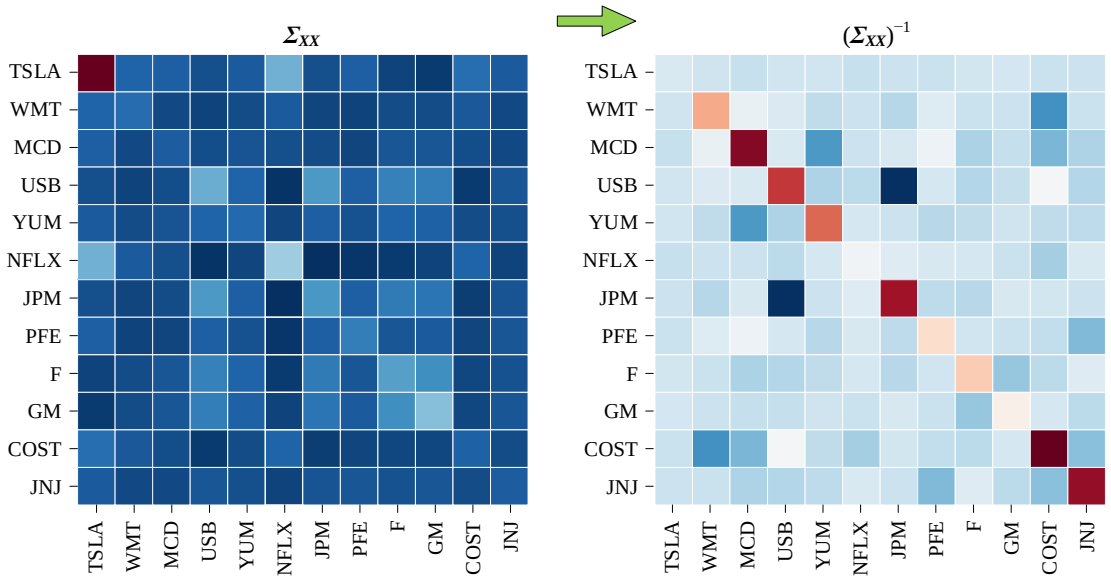


图 29. 分块协方差矩阵求逆

如图 30 所示，截距系数之外的多元线性回归系数向量为：

$$\boldsymbol{b}_{1\sim D} = \left(\boldsymbol{\Sigma}_{\boldsymbol{XX}}\right)^{-1} \boldsymbol{\Sigma}_{\boldsymbol{Xy}} \tag{58}$$

如图 31 所示，b_0 为：

$$b_0 = \mathrm{E}(\boldsymbol{y}) - \mathrm{E}(\boldsymbol{X})\boldsymbol{b}_{1\sim D} \tag{59}$$

其中，E($\boldsymbol{X}$) 为行向量。

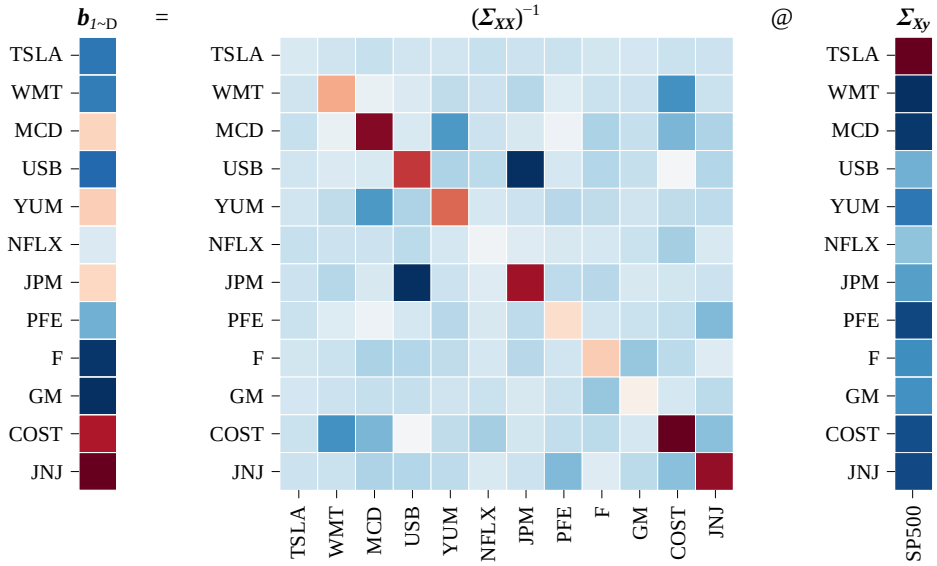


图 30. 求线性回归参数，除截距以外

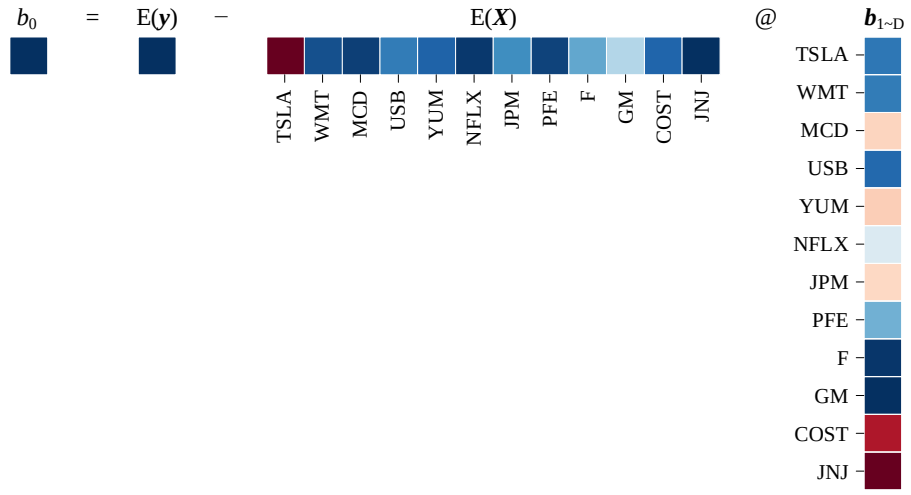
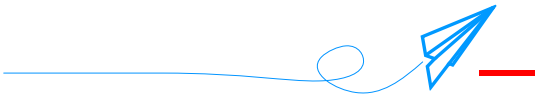


图 31. 求截距系数

Bk6_Ch10_03.py 完成本节运算。



OLS 线性回归是一种在机器学习中常用的算法，它可以通过最小化残差平方和来建立线性模型，从而用于预测和分析因变量与自变量之间的关系。OLS 线性回归适用于数据分析、预测模型、异常检测、特征工程等多种机器学习任务。通过使用 OLS 线性回归，可以得出自变量对因变量的影响程度、探索自变量之间的关系、预测因变量的取值，以及识别异常值等。OLS 线性回归是一种简单但可靠的机器学习算法，为数据分析和预测建模提供了强大的工具和方法。

鸢尾花书从不同视角介绍过 OLS 线性回归。《数学要素》从代数、几何、优化角度讲过线性回归，《矩阵力量》从线性代数、正交投影、矩阵分解视角分析线性回归。《统计至简》又增加了条件概率、MLE 这两个视角。鸢尾花书有关 OLS 线性回归的讲解至此告一段落，本书后续将介绍回归中的正则化、贝叶斯回归、非线性回归等话题。

11 Regularized Regression
正则化回归

利用正则项，缩减挑选特征，构造简洁模型

遇到数学难题，别犯愁；困扰我的难题比你的大得多。

Do not worry too much about your difficulties in mathematics, I can assure you that mine are still greater.

—— 阿尔伯特·爱因斯坦 (Albert Einstein) | 理论物理学家 | 1879 ~ 1955

- `seaborn.lineplot()` 绘制线图
- `sklearn.linear_model.ElasticNet()` 求解弹性网络回归问题
- `sklearn.linear_model.lars_path()` 生成 Lasso 回归参数轨迹图
- `sklearn.linear_model.Lasso()` 求解套索回归问题
- `sklearn.linear_model.Ridge()` 求解岭回归问题
- `sklearn.metrics.mean_squared_error()` 计算均方误差 MSE
- `statsmodels.api.add_constant()` 线性回归增加一列常数 1
- `statsmodels.api.OLS()` 最小二乘法函数

11.1 正则化：抑制过度拟合

正则化 (regularization) 可以用来抑制过度拟合。本书前文提过，所谓过度拟合，是指模型参数过多或者结构过于复杂。

正则项 (regularizer, regularization term, penalty term) 通常被加在**目标函数** (objective function) 当中。正则项可以让估计参数变小甚至为 0，这一现象也叫**特征缩减** (shrinkage)。本章将采用图形方式来讲解如何在多元线性回归目标函数中引入正则项。

本章将 L1 正则项、L2 正则项以及 L1 和 L2 混合正则项利用在多变量线性回归中。L1 正则化为回归参数的 L^1 范数，L2 正则化为回归参数的 L^2 范数。

⚠ 鸢尾花书中在谈及 L^p 范数时，会采用相对严格的数学记号 L^p。

OLS 优化问题

对于多元线性 OLS 回归，优化问题为：

$$\arg\min_{\boldsymbol{b}} \left\| \boldsymbol{y} - \boldsymbol{X}\boldsymbol{b} \right\|_2^2 \tag{1}$$

对于二元线性 OLS 回归，不考虑常数项系数，b_1 和 b_2 两个回归参数形成如图 1 所示曲面。容易发现曲面为二次椭圆曲面。

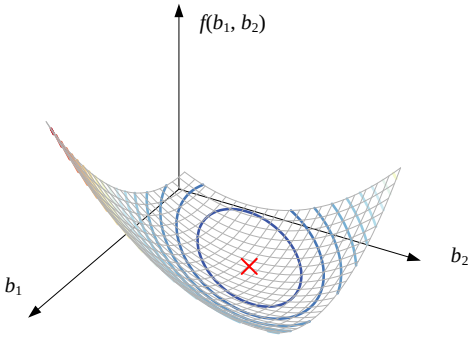


图 1. 二元线性 OLS 回归参数曲面

L2 正则化

线性 OLS 中，引入 L2 正则项，可以得到**岭回归** (ridge regression)：

$$\arg\min_{\boldsymbol{b}} \left\| \boldsymbol{y} - \boldsymbol{X}\boldsymbol{b} \right\|_2^2 + \alpha \underbrace{\left\| \boldsymbol{b} \right\|_2^2}_{\text{regularizer}} \tag{2}$$

白话说，L2 正则化是回归参数各个元素平方之和。α 这个惩罚系数是用户决定的。

⚠ 注意，一般文献中上式惩罚系数用 λ，本章和 Scikit-learn 保持一致采用 α。

(2) 相当于图 1 曲面叠加了 L2 正则项曲面，具体如图 2。L2 正则项曲面等高线为正圆面，对应的最小值点为原点。叠加得到的岭回归参数曲面最小值位置朝原点发生明显偏移。当 (2) 中参数 α 越大，正则项影响越大，求解优化问题得到的回归参数越靠近原点。

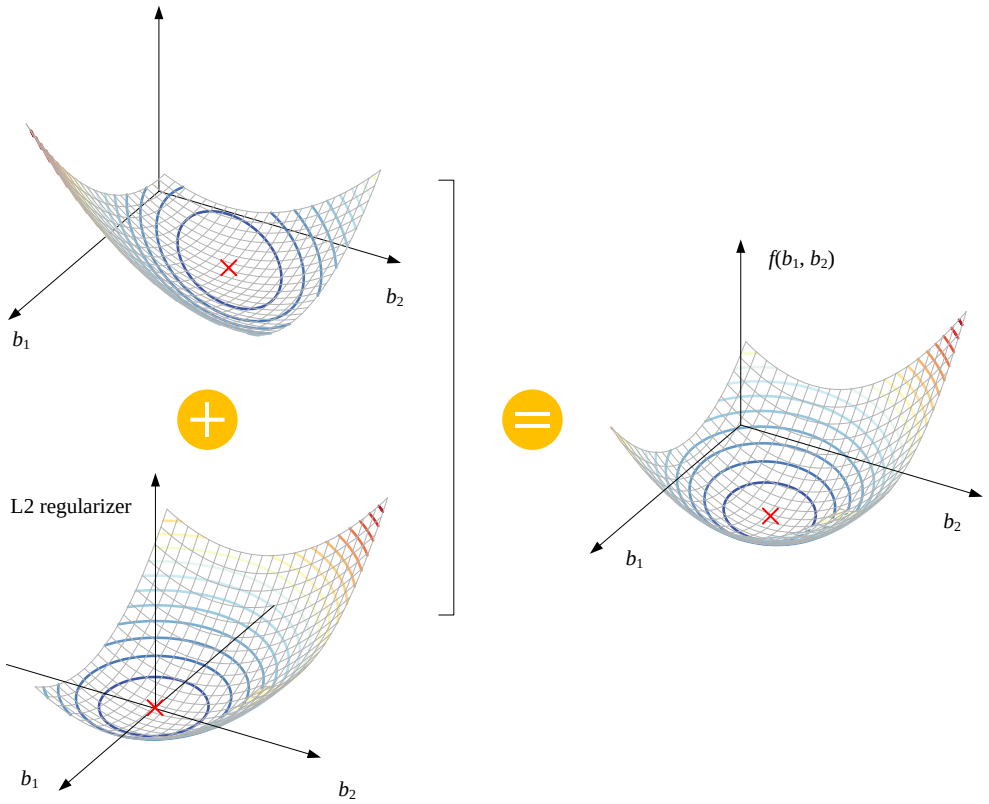


图 2. 岭回归参数曲面

L1 正则化

线性 OLS 中，引入 L1 正则项，可以得到**套索回归** (LASSO regression)：

$$\arg\min_{\boldsymbol{b}} \frac{1}{2n}\left\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{b}\right\|_2^2 + \underbrace{\alpha\left\|\boldsymbol{b}\right\|_1}_{\text{regularizer}} \tag{3}$$

⚠ 注意，(3) 中多元线性 OLS 回归优化项除以 $2n$，n 为样本数据数量。此外，不同文献套索回归的目标函数稍有不同，本章和 Scikit-learn 保持一致。

白话说，L1 正则化是回归参数各个元素绝对值之和。

鸢尾花书《矩阵力量》介绍过 L1 正则项曲面等高线为旋转正方形。

(3) 相当于在图 1 二次椭圆抛物面上叠加图 1 曲面叠加。图 3 所示为这一过程。套索回归可以进行特征选择，从而有效减少回归模型所依赖的特征数量，本章后文将从不同角度详细讲解这一点。

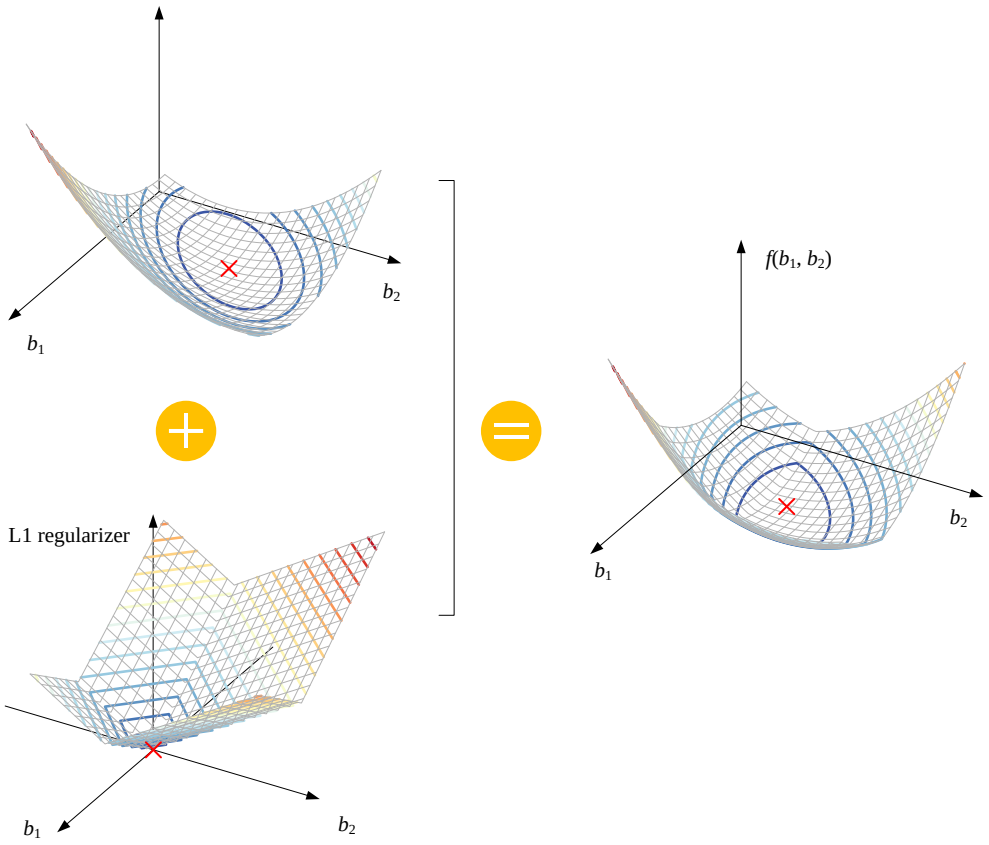


图 3. 套索回归参数曲面

L1 + L2 正则化

线性 OLS 中，以不同比例同时引入 L1 和 L2 正则项，可以得到**弹性网络回归** (elastic net regression)：

$$\arg\min_{\boldsymbol{b}} \frac{1}{2n}\left\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{b}\right\|_2^2+\alpha\left(\rho\left\|\boldsymbol{b}\right\|_1+\frac{(1-\rho)}{2}\left\|\boldsymbol{b}\right\|_2^2\right) \tag{4}$$

其中，参数 ρ 用来调和 L1 和 L2 正则项的比例。图 4 所示如何构造得到弹性网络回归系数曲面。弹性网络回归相当于岭回归和套索回归的合体。

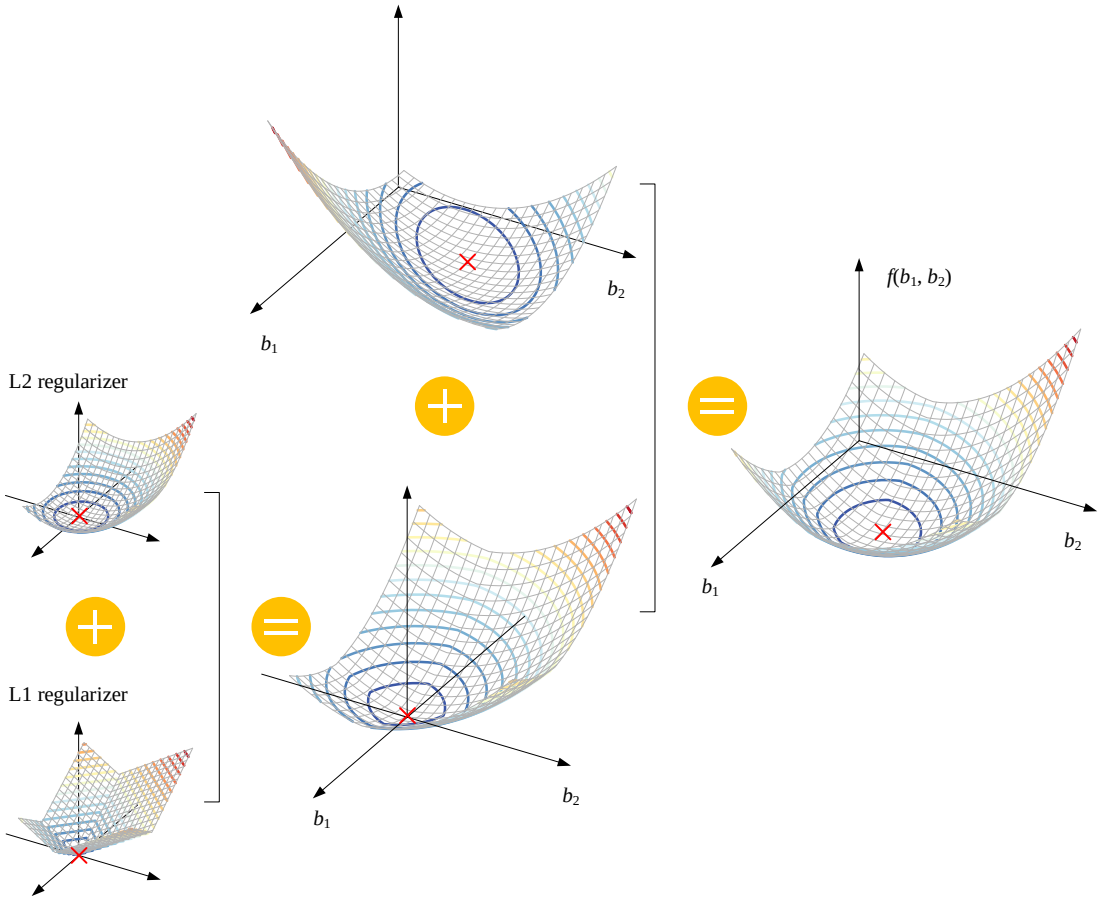


图 4. 弹性网络回归参数曲面

11.2 岭回归

如前文所述，岭回归引入 L2 正则项来缩减模型参数，岭回归的优化目标函数为：

$$f(\boldsymbol{b}) = \underbrace{\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}\|_2^2}_{\text{OLS}} + \underbrace{\alpha \|\boldsymbol{b}\|_2^2}_{\text{L2 regularizer}} \tag{5}$$

图 5 所示为给定 α 条件下，(5) 如何构造得到岭回归目标函数参数曲面等高线图。

⚠ 注意，本节假设回归问题为二元，只有 b_1 和 b_2 两个回归参数，并且不考虑常数项系数。

如前文所述，(5) 目标函数中 OLS 部分对应椭圆抛物面，最小值点为红色 ×；红色 ×为二元 OLS 线性回归参数解的位置。

(5) 中 L2 正则项则对应正圆抛物面，最小值点为蓝色 ×，位于原点。原点处，参数系数为全 0。

根据《数学要素》一书中介绍的二次曲面内容，两个二次曲面叠加得到的一般还是一个二次曲面。

(5) 对应的曲面 $f(b_1, b_2)$ 仍然是一个椭圆抛物面，最小值点为黄色 ×；黄色 × 为给定 α 条件下岭回归参数的优化解。

容易发现，黄色 × 位于红色 × 和蓝色 × 之间；相对 OLS 线性回归参数红色 ×，岭回归参数黄色 ×，更靠近原点。

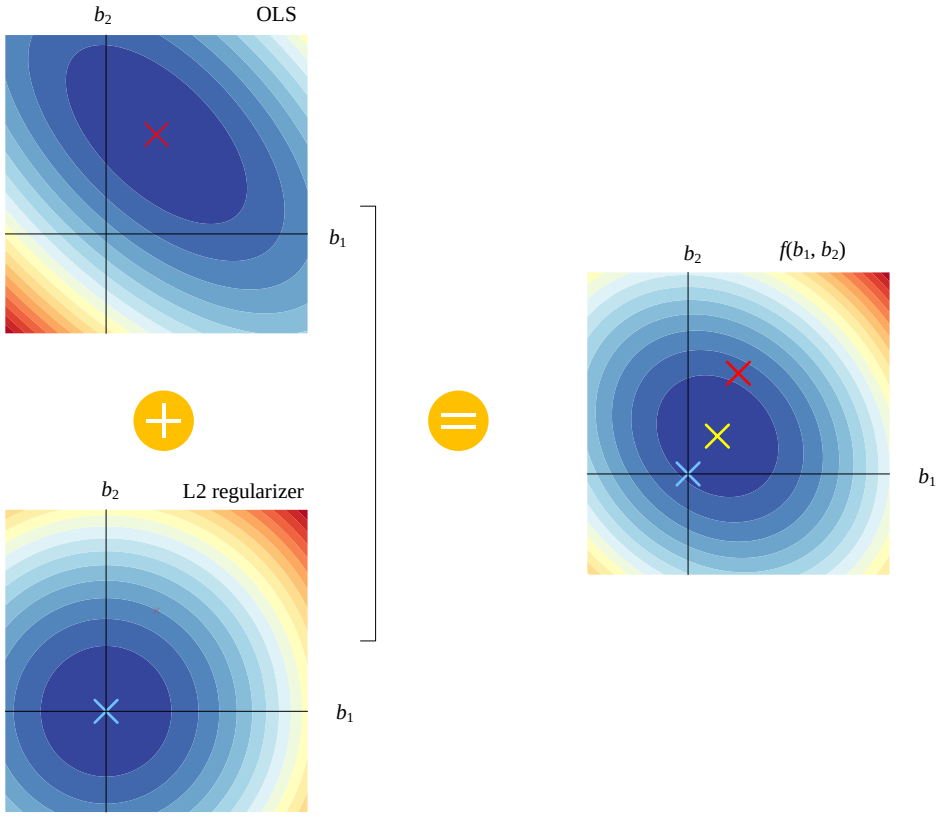


图 5. 构造岭回归优化问题参数曲面

不断增大 L2 约束项参数 α，可以发现岭回归参数优化解不断靠近原点，如图 6 所示。注意，图 6 分图中的等高线为岭回归曲面 $f(b_1, b_2)$。当约束项参数 α 不断增大，$f(b_1, b_2)$ 曲面中 L2 正则项 (正圆曲面) 影响力不断增强。参数 α 不断增大，$f(b_1, b_2)$ 曲面等高线也从旋转椭圆渐渐变成正圆，最小值点也渐渐靠近 (收缩到) 原点。

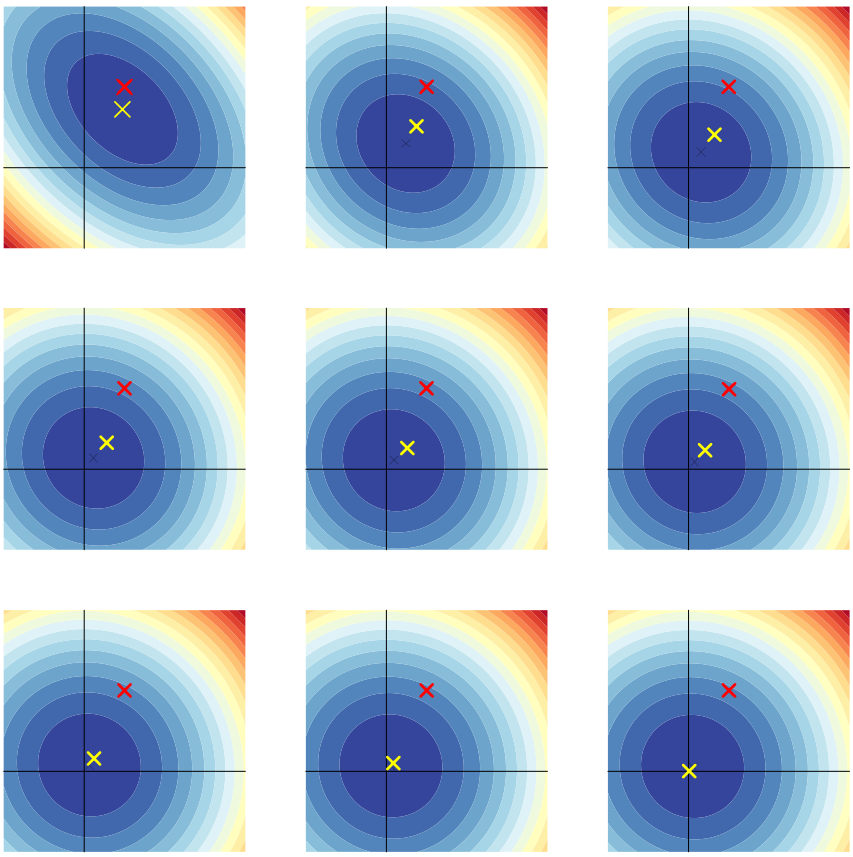


图 6. 不断增大 α，岭回归参数位置变化

构造一个线性回归问题，利用 12 只股票的日收益率解释标普 500 涨跌。图 7 所示为利用 OLS 多元线性回归得到的这个回归问题的参数。

OLS Regression Results

Dep. Variable:	SP500	R-squared:	0.774
Model:	OLS	Adj. R-squared:	0.750
Method:	Least Squares	F-statistic:	32.48
Date:	XXXXXXXXXXXXXXX	Prob (F-statistic):	3.03e-31
Time:	XXXXXXXXXXXXXXX	Log-Likelihood:	493.88
No. Observations:	127	AIC:	-961.8
Df Residuals:	114	BIC:	-924.8
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.0005	0.000	-1.038	0.302	-0.001	0.000
TSLA	0.0248	0.011	2.248	0.027	0.003	0.047
WMT	0.0272	0.041	0.667	0.506	-0.054	0.108
MCD	0.1435	0.057	2.536	0.013	0.031	0.256
USB	0.0164	0.051	0.322	0.748	-0.084	0.117
YUM	0.1469	0.047	3.114	0.002	0.053	0.240
NFLX	0.0972	0.021	4.539	0.000	0.055	0.140
JPM	0.1415	0.055	2.583	0.011	0.033	0.250
PFE	0.0546	0.033	1.662	0.099	-0.010	0.120
F	-0.0068	0.036	-0.187	0.852	-0.078	0.065
GM	-0.0105	0.027	-0.388	0.699	-0.064	0.043
COST	0.2176	0.059	3.713	0.000	0.101	0.334
JNJ	0.2414	0.056	4.350	0.000	0.131	0.351

Omnibus:	7.561	Durbin-Watson:	1.862
Prob(Omnibus):	0.023	Jarque-Bera (JB):	8.445
Skew:	0.400	Prob(JB):	0.0147
Kurtosis:	3.978	Cond. No.	156.

图 7. 多元 OLS 线性回归解

利用 sklearn.linear_model.Ridge() 函数，我们可以求解上述问题的岭回归参数。设定不同的 α 值，可以获得一系列岭回归参数。图 8 所示为随着 α 增大，岭回归参数变化。可以发现，α 增大时，参数逐步最大限度接近 0，但是不等于 0。这一点和本章后文将介绍的套索回归和弹性网络回归截然不同。

用残差平均值 MSE 来量化岭回归参数和 OLS 参数的差距：

$$\mathrm{MSE}\left(\boldsymbol{b}_{\text{ridge}}, \boldsymbol{b}_{\text{OLS}}\right) = \frac{1}{D+1}\left\|\boldsymbol{b}_{\text{ridge}} - \boldsymbol{b}_{\text{OLS}}\right\|_2^2 \tag{6}$$

图 9 所示为随着 α 增大，岭回归参数和 OLS 参数的差距不断增大。

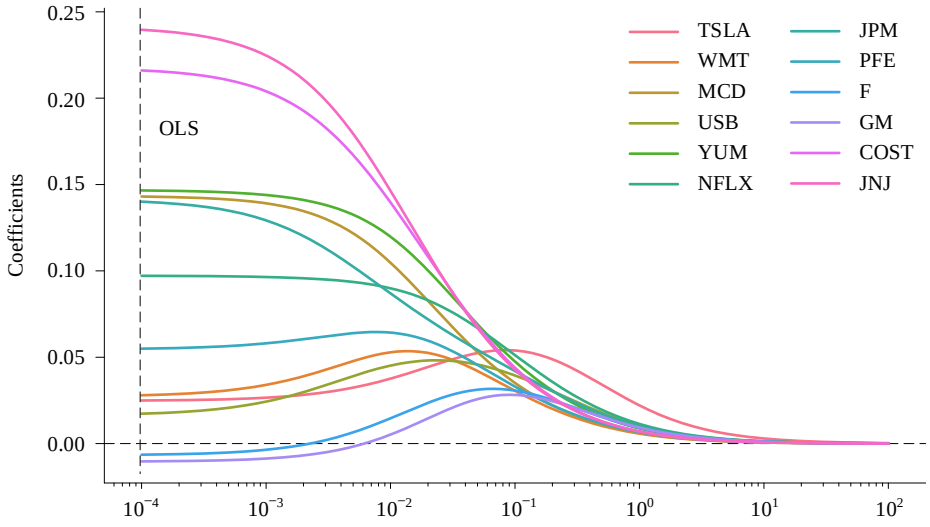


图 8. 随着 α 增大，岭回归参数变化

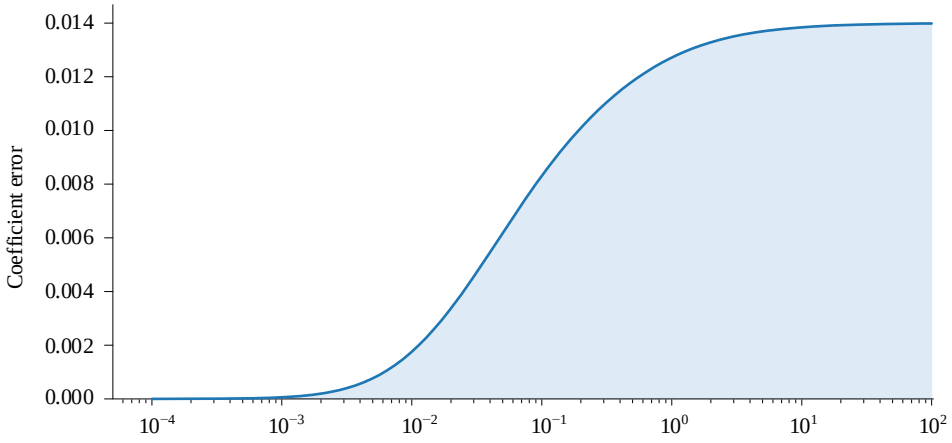


图 9. 和 OLS 相比，岭回归参数误差

Bk6_Ch11_01.py 绘制本节图像。

11.3 几何角度看岭回归

从另外一个角度看岭回归，岭回归可以看做是 OLS 线性回归问题，加一个约束条件。

$$\begin{aligned} &\arg\min_{\boldsymbol{b}} \left\| \boldsymbol{y} - \boldsymbol{X}\boldsymbol{b} \right\|_2^2 \\ &\text{subject to: } \left\| \boldsymbol{b} \right\|_2^2 - c \le 0 \end{aligned} \tag{7}$$

(7) 中的约束条件中 c 是一个阈值，就是把回归参数限制在一定范围之内，即：

$$b_0^2 + b_1^2 + b_2^2 + \cdots + b_D^2 \le c \tag{8}$$

注意，(7) 中阈值 c 越小，对应惩罚系数 α 越大。

不考虑常数系数，$D = 2$ 时，

$$b_1^2 + b_2^2 \le c \tag{9}$$

上式为一个正圆面，圆心位于原点，半径为 $\sqrt{c}$ 。OLS 对应的是旋转椭圆等高线和 (9) 正圆相切就是约束条件下优化解，也就是岭回归系数。

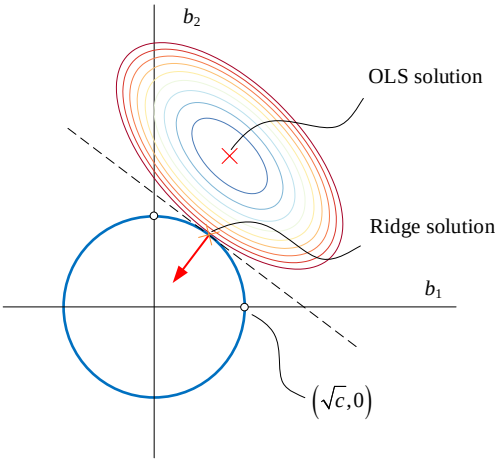


图 10. 约束角度看岭回归

图 11 所示为正圆面半径 $\sqrt{c}$ 取不同值时，岭回归回归系数的优化解位置变化。

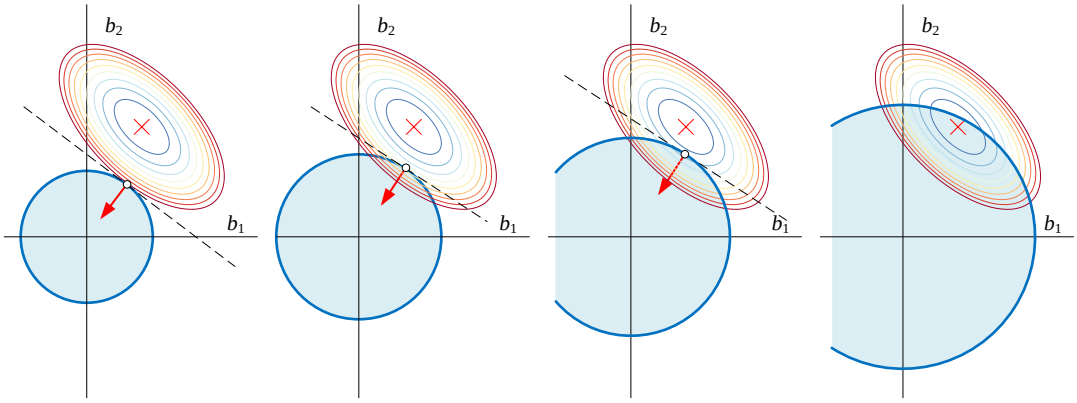


图 11. c 取不同值时，岭回归优化系数位置

多元 OLS 线性回归系数 $\boldsymbol{b}$ 的解：

$$\boldsymbol{b}=\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{10}$$

根据本书前文介绍的内容，OLS 线性回归的优化问题解存在且唯一的条件是 $\boldsymbol{X}$ 列满秩。

如果，不满足 $\boldsymbol{X}$ 列满秩这个条件，则表明 $\boldsymbol{X}$ 列向量存在线性相关，即多重共线性。当 $\boldsymbol{X}$ 列与列之间线性相关或者线性相关较大时，$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 的行列式等于或接近于 0，无法求解(10) 中 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 一项的逆，会使得 OLS 解不稳定，

而岭回归线性回归系数 $\boldsymbol{b}$ 的解为：

$$\boldsymbol{b}=\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\alpha\boldsymbol{I}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{11}$$

比较 (10)，可以发现 (11) 中变为求解 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\alpha\boldsymbol{I}$ 的逆；将 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 加上矩阵 $\alpha\boldsymbol{I}$ 变成非奇异矩阵并可以进行求逆运算。而 $\alpha\boldsymbol{I}$ 为对角矩阵，对角线上元素为 α，其余为 0，形状酷似“山岭”，这也就是“岭回归”名称的由来。

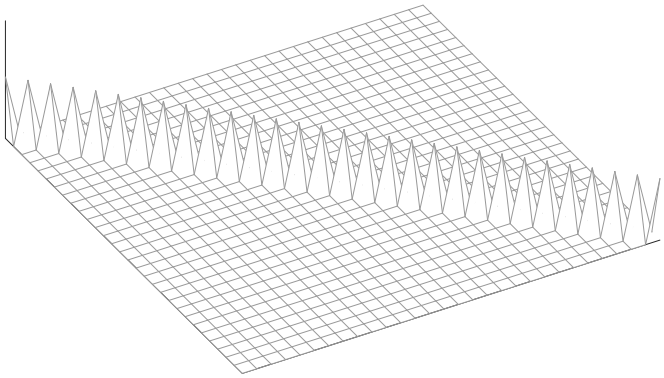


图 12. $\alpha\boldsymbol{I}$ 对角矩阵引入的“山岭”

11.4 套索回归

斯坦福大学教授 Robert Tibshirani 在 1996 年首次提出将 L1 范数作为 OLS 正则项，得到 Lasso 模型。Lasso 是 least absolute shrinkage and selection operator 的缩写。

套索的优化目标函数为：

$$f\left(\boldsymbol{b}\right)=\underbrace{\frac{1}{2n}\left\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{b}\right\|_2^2}_{\text{OLS}}+\underbrace{\alpha\left\|\boldsymbol{b}\right\|_1}_{\text{L1 regularizer}} \tag{12}$$

图 13 所示为给定 α 条件下，(12) 如何构造得到套索回归目标函数参数曲面等高线图。如前文所述，(12) 目标函数中 OLS 部分对应椭圆抛物面，最小值点为红色 ×；红色 ×为二元 OLS 线性回归参数解的位置。(12) 中 L1 正则项曲面等高线对应旋转正方形，最小值点为蓝色 ×，位于原点。

容易发现，黄色 × 位于红色 × 和蓝色 × 之间；相对 OLS 线性回归参数红色 ×，岭回归参数黄色 ×，更靠近原点。

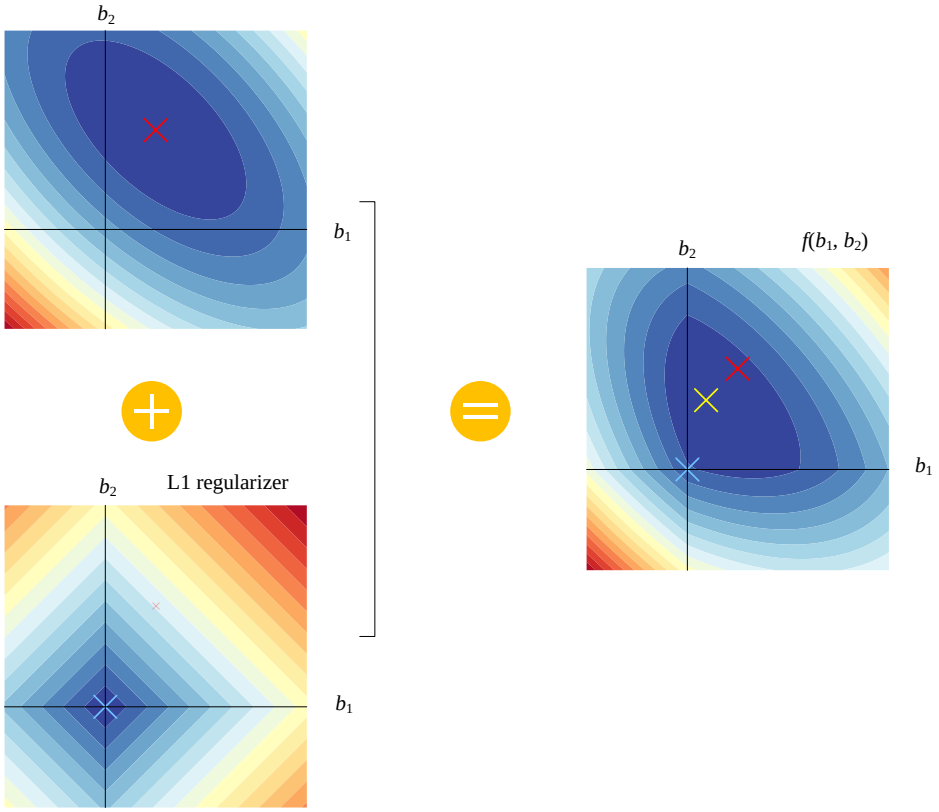


图 13. 构造套索回归优化问题参数曲面

图 14 所示为不断增大 α，套索回归参数位置变化；可以发现套索回归采用 L1 正则化，可以导致参数估计结果为 0。

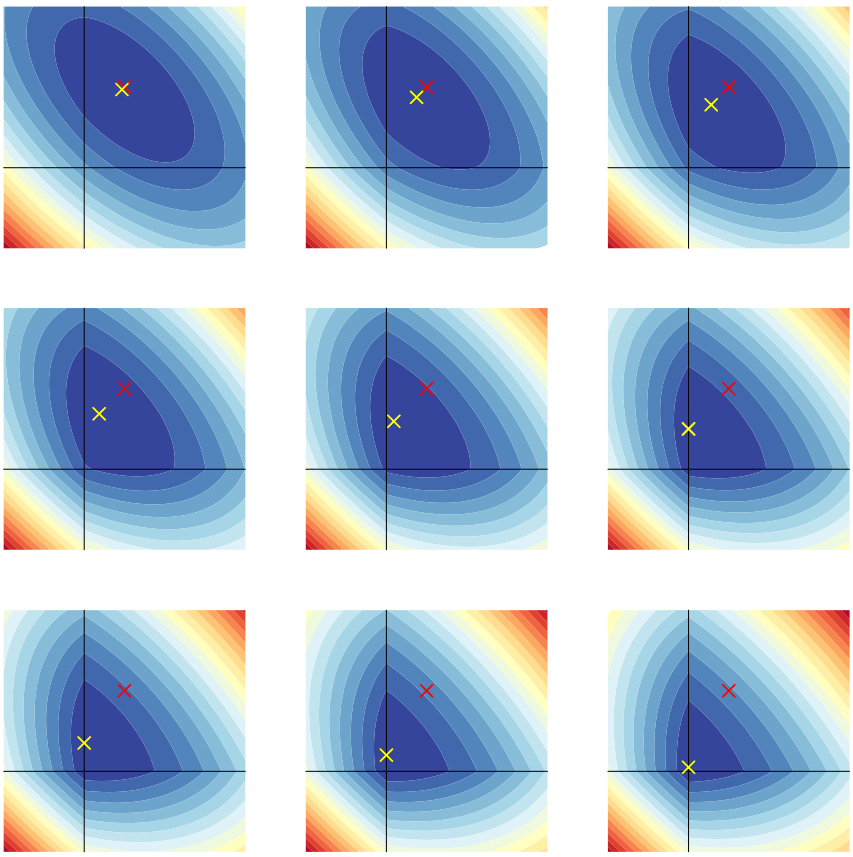


图 14. 不断增大 α，套索回归参数位置变化

利用 sklearn.linear_model.Lasso() 可以获得套索回归的结果，利用本章前文的代码，将岭回归函数，换成套索回归函数，对于同一个问题，可以得到图 15。该图所示为随着 α 增大，套索回归参数变化。

观察图 15，可以发现在回归模型中，α 增大，一些特征快速收缩为 0，这个过程也是一个特征选择的过程。在套索回归中，系数越小表示对结果的影响越小，系数为 0 表示该特征没有对结果的影响，因此套索回归可以用于特征选择和降维。因此套索回归可以删除没有必要的特征，产生更为简洁的回归模型。特别地，sklearn.linear_model.lars_path()函数可以用来生成套索回归参数轨迹图。图 16 所示为和 OLS 相比，套索回归参数误差。

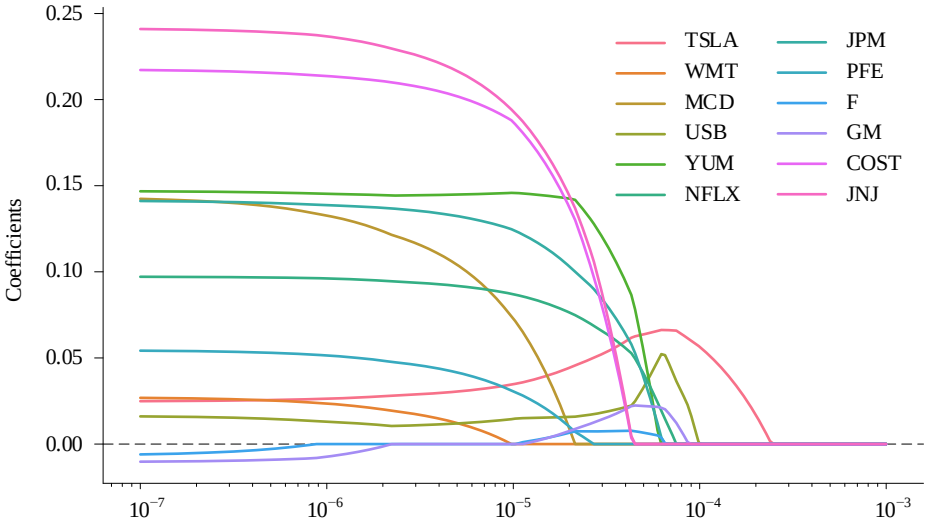


图 15. 随着 α 增大，套索回归参数变化

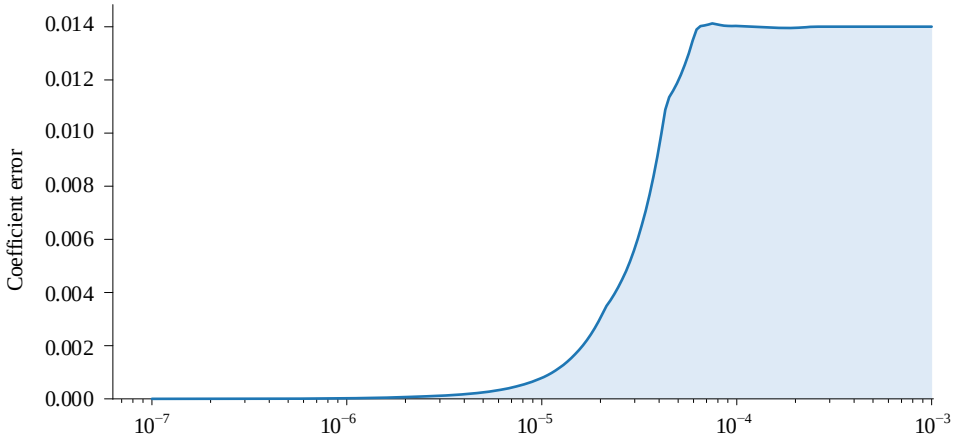


图 16. 和 OLS 相比，套索回归参数误差

11.5 几何角度看套索回归

类似地，本节从几何角度看套索回归。套索回归，可以看做是 OLS 线性回归问题，加一个约束条件：

$$\begin{aligned} &\arg\min_{\boldsymbol{b}} \left\| \boldsymbol{y} - \boldsymbol{X}\boldsymbol{b} \right\|_2^2 \\ &\text{subject to: } \left\| \boldsymbol{b} \right\|_1 - c \le 0 \end{aligned} \tag{13}$$

(7) 中的约束条件中 c 也是一个阈值，即：

$$|b_0|+|b_1|+|b_2|+\cdots+|b_D|\leq c \tag{14}$$

不考虑常数系数，$D=2$ 时，

$$|b_1|+|b_2|\leq c \tag{15}$$

上式为一个旋转正方形，中心位于原点。OLS 对应的是旋转椭圆等高线可以和 (15) 旋转正方形相切，或在顶点处相交，如图 17 所示。

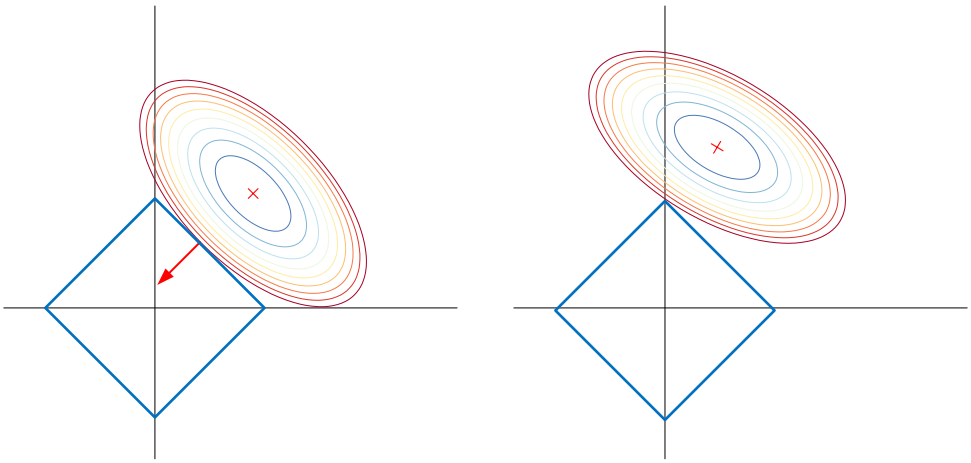


图 17. 套索回归的 L1 正则项

如图 18 所示，对于同一个 OLS 优化问题，不同的 c 阈值大小，会在不同位置得到套索回归系数解。前文说过，岭回归系数可以无限接近于 0，但是不等于 0；不同于岭回归，套索回归的参数可以直接为 0。套索回归参数的这种特点叫做**稀疏性** (sparsity)。稀疏性是指在套索回归中，某些特征系数被稀疏化为 0，使得模型参数更加简化和易于解释，同时也减少了数据维度，提高了模型的泛化能力。

当样本数据矩阵特征过多，但是只有少数特征对回归模型有贡献，去掉剩下的特征对模型没有什么影响。也就是说，回归模型只关注系数向量中非零项特征就足够了。因此，区别于岭回归，套索回归可以进行特征选择。

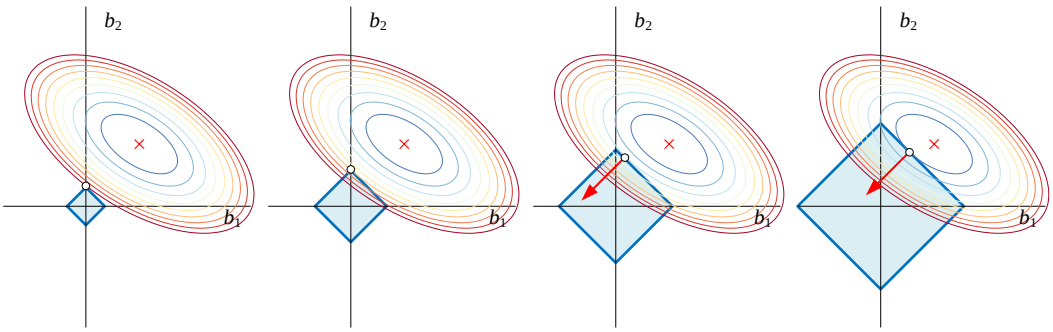


图 18. c 取不同值时，套索回归优化系数位置

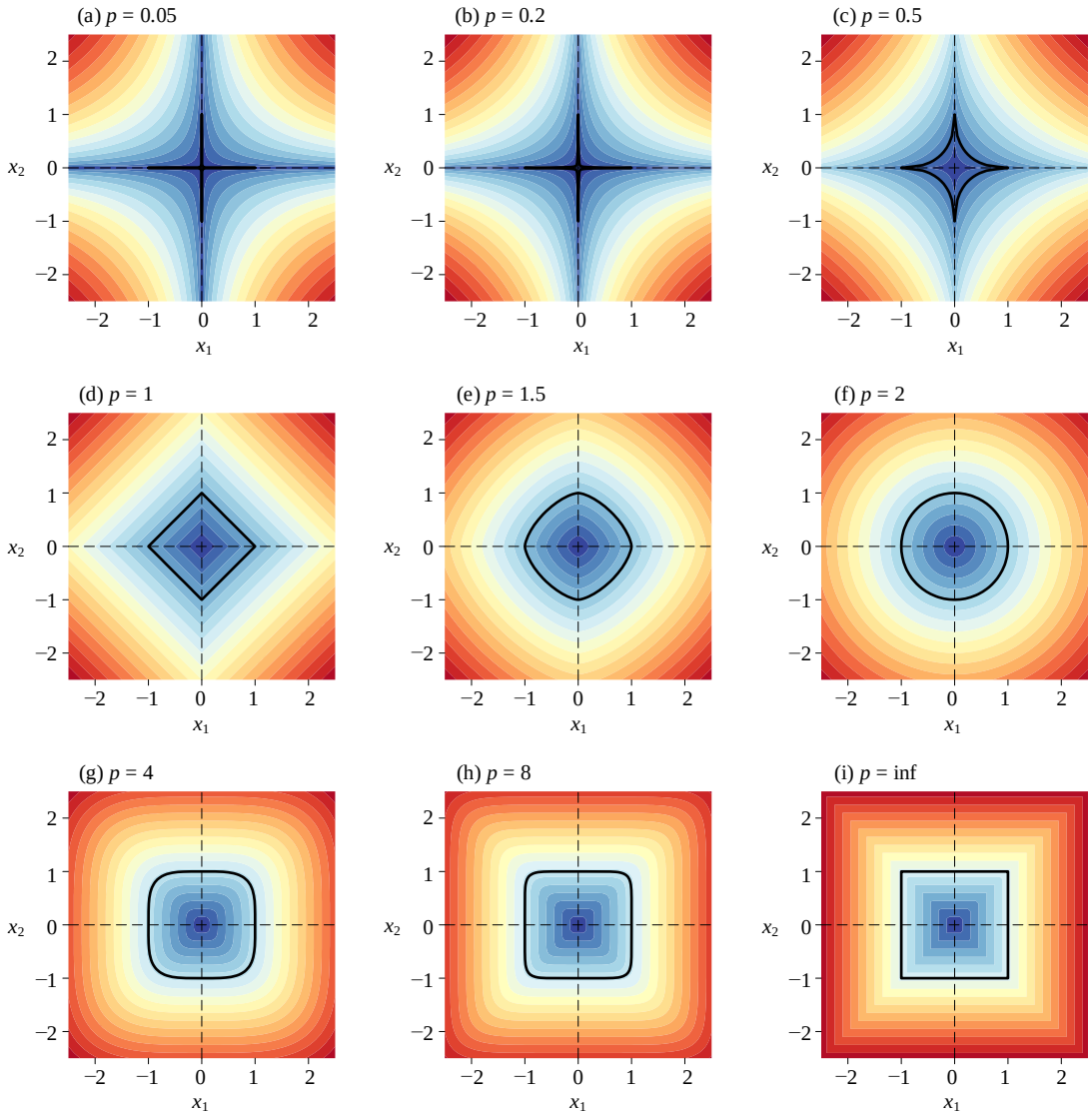


图 19. p 取不同值时，Lp 范数等高线形状变化；注意，严格来讲只有 $p \geqslant 1$ 才是范数

有大家可能会问，为什么 L1 正则项会有这种稀疏性效果？回顾丛书《矩阵力量》一书中给出的图 19。图 19 中给出，p 取不同值时，L^p 范数等高线形状变化。可以发现，$p > 1$ 时，L^p 范数等高线形状连续光滑，没有尖点。只有 $p \leqslant 1$ 时，等高线图出现顶点尖点；但是当 $p < 1$ 时，目标函数为非凸函数，优化问题求解困难。正是这个突出尖点的存在，且满足凸优化问题，让套索回归产生稀疏的向量解。

再次强调，数学上严格来讲，只有 $p \geqslant 1$ 才是 L^p 范数。

相信大家现在理解为什么，L2 范数作为正则项，无法产生稀疏性效果。二维平面下 L2 正则项的等高线是正圆；与正方形相比，正圆根本没有棱角。因此 OLS 等高线和这个正圆相切时，得到任意系数为 0 的机会几乎为零。这也就是为什么 L2 正则化不具备稀疏性的原因。

以上结论不仅仅适用于二维，三维甚至更多维度同样适用。图 20 比较三维空间的 L1 和 L2 正则项等高线曲面。

《数学要素》一本在超椭圆相关内容中介绍过图 20 图像。

图 20 (a) 中，L1 正则项存在大量突出尖点；这些尖点都对应着部分系数为 0。图 20 (b) 给出的正球体 (L2 正则)，任意一丁点扰动，比如计算误差、收敛等等，都会让回归系数不能恰好为 0。

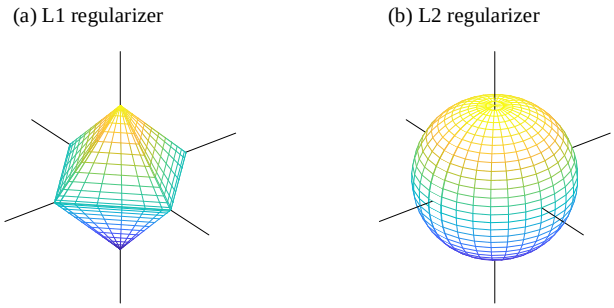


图 20. 三维空间的 L1 和 L2 正则项

此外，有些问题希望一些特征参数同时为 0，或者同时不为 0。这时可以设计，组 lasso (group lasso) 惩罚项来实线这一目标。与传统的 lasso 回归不同之处在于，组 lasso 回归在 L1 正则化项中增加了对特征分组的惩罚项。这个惩罚项是对组内系数的 L1 范数进行惩罚，从而鼓励组内特征系数共享相同的值或者趋近于零。因此，组 lasso 可以同时选择重要的特征和重要的特征组。这个方法在处理高维数据时特别有效，因为它可以减少特征的数量，避免过拟合，而且还可以保留组内特征之间的相关性。

图 21 所示为三维空间中两种 lasso 惩罚项结构。

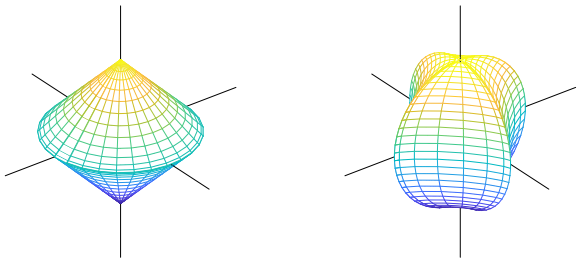


图 21. 三维空间中组 lasso 惩罚项

混合 L1 和 L2 正则项的弹性网络回归方法，可以克服 L2 正则项的不具备稀疏性这一缺点；这是我们下一节要介绍的内容。

11.6 弹性网络回归

弹性网络回归 (elastic net regression) 以不同比例同时引入 L1 和 L2 正则项，对应的目标函数为：

$$f(\boldsymbol{b}) = \underbrace{\frac{1}{2n}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{b}\|_2^2}_{\text{OLS}} + \underbrace{\alpha\left(\rho\|\boldsymbol{b}\|_1 + \frac{(1-\rho)}{2}\|\boldsymbol{b}\|_2^2\right)}_{\text{Elastic net regularizer}} \tag{16}$$

注意，α 为正则项惩罚系数，参数 ρ 用来调和 L1 和 L2 正则项的比例。

α 和 ρ 都是用户输入的数值。图 22 所示为构造弹性网络回归优化问题参数曲面等高线的过程。

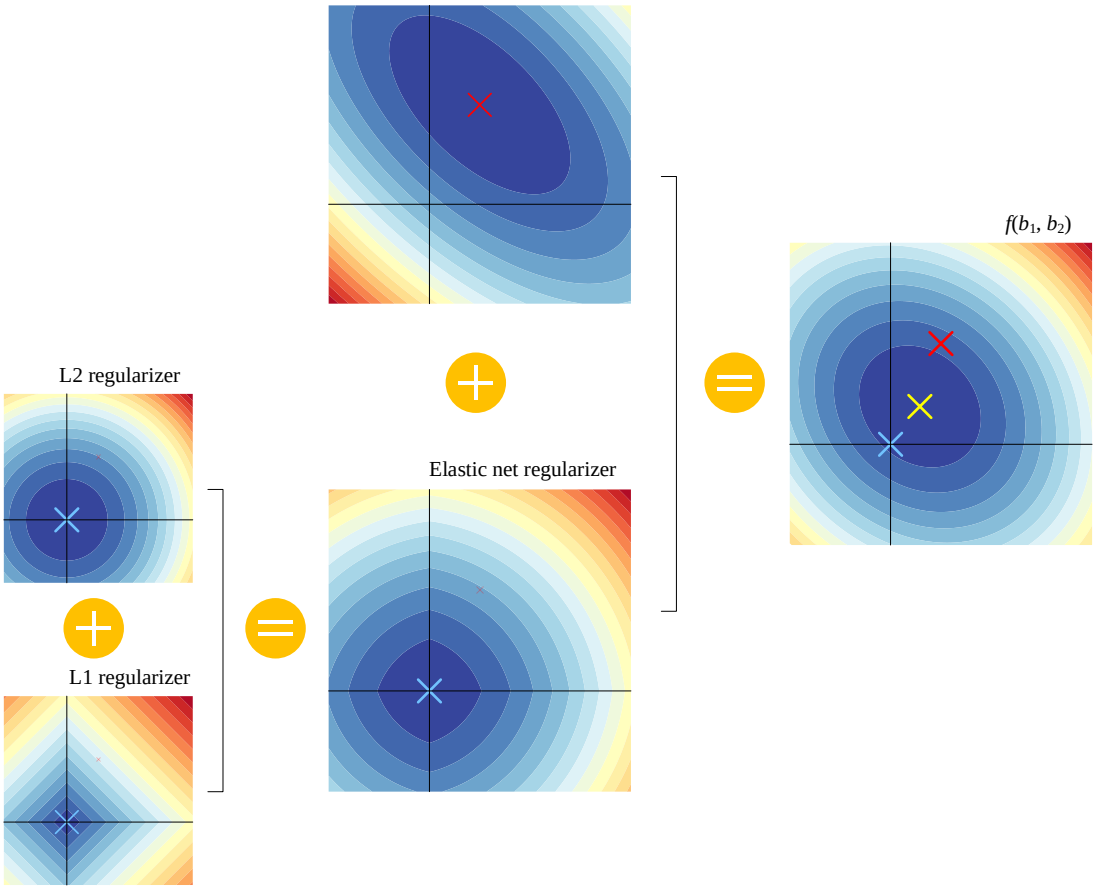


图 22. 构造弹性网络回归优化问题参数曲面等高线

图 23 所示为不断增大 α，弹性网络回归参数位置变化。可以发现 α 增大，回归系数 b_1 不断靠近 0，甚至为 0。图 24 所示为回归系数运动轨迹，弹性网络回归系数靠近 0 的“速度”慢于套索回归。

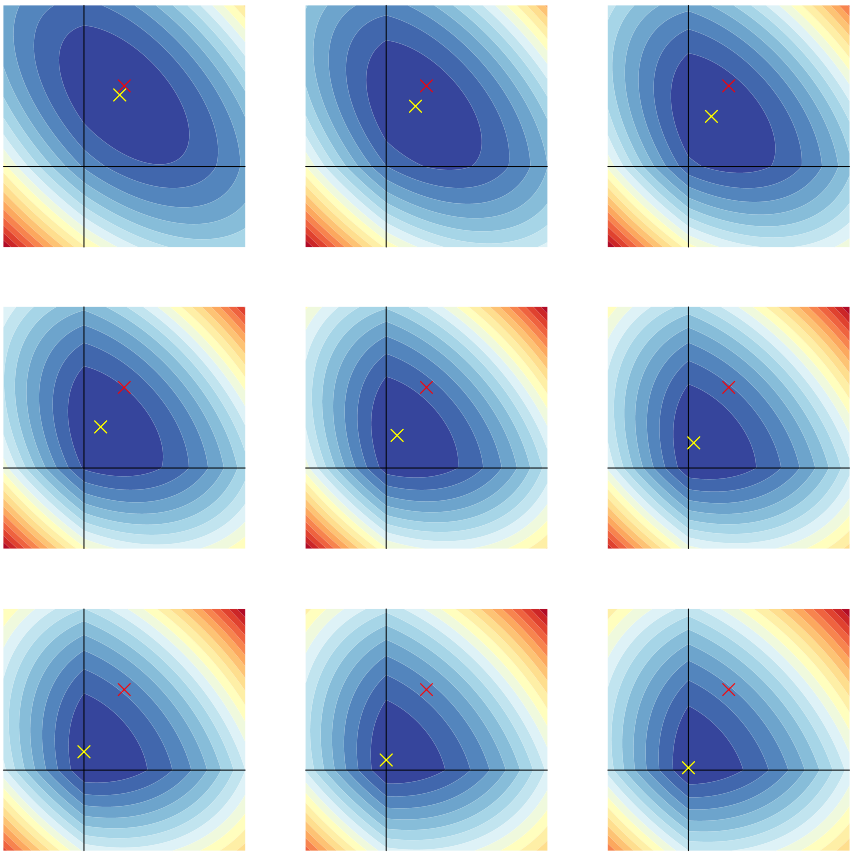


图 23. 不断增大 α，弹性网络回归参数位置变化

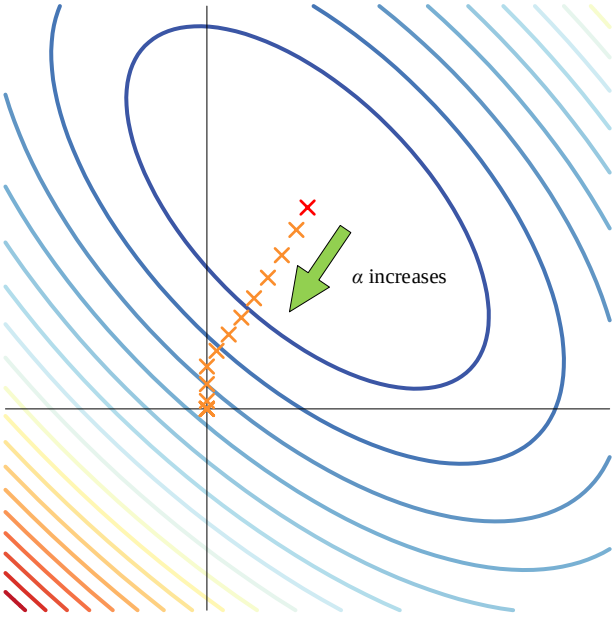


图 24. 不断增大 α，弹性网络回归参数变化轨迹

本节前文介绍，参数 ρ 用来调和 L1 和 L2 正则项的比例；下面看一下参数 ρ 对弹性网络正则项形状的影响。图 25 和图 26 分别展示二维平面和三维空间中弹性网络正则项形状随 ρ 变化。ρ 越大，弹性网络正则项越接近 L1，稀疏性越强；ρ 越小，弹性网络正则项越接近 L2，稀疏性越弱。

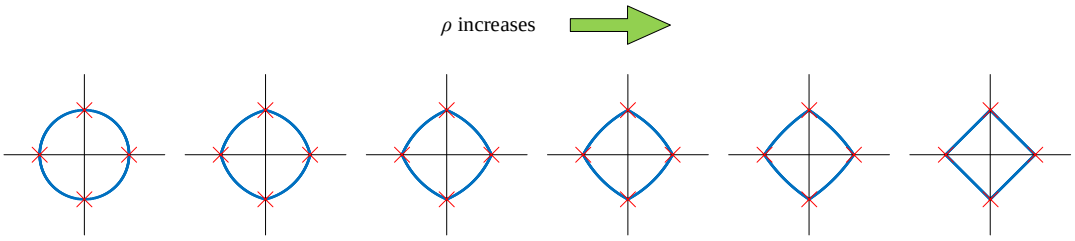


图 25. 不断增大 ρ，二维平面弹性网络正则项等高线形状

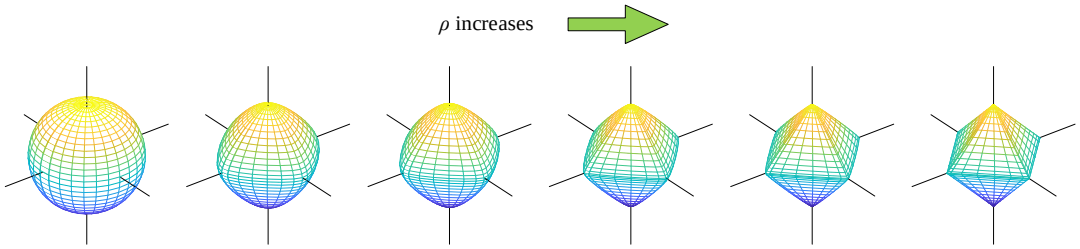


图 26. 不断增大 ρ，三维空间弹性网络正则项等高线形状

图 27 所示为随着 α 增大，弹性网络回归参数变化，也就是套索回归参数轨迹图。

注意，在这一过程中，参数 ρ 不变。

sklearn.linear_model.ElasticNet() 函数可以用来求解弹性网络回归问题。

此外，sklearn.linear_model.enet_path() 可以专门绘制套索回归参数轨迹图。

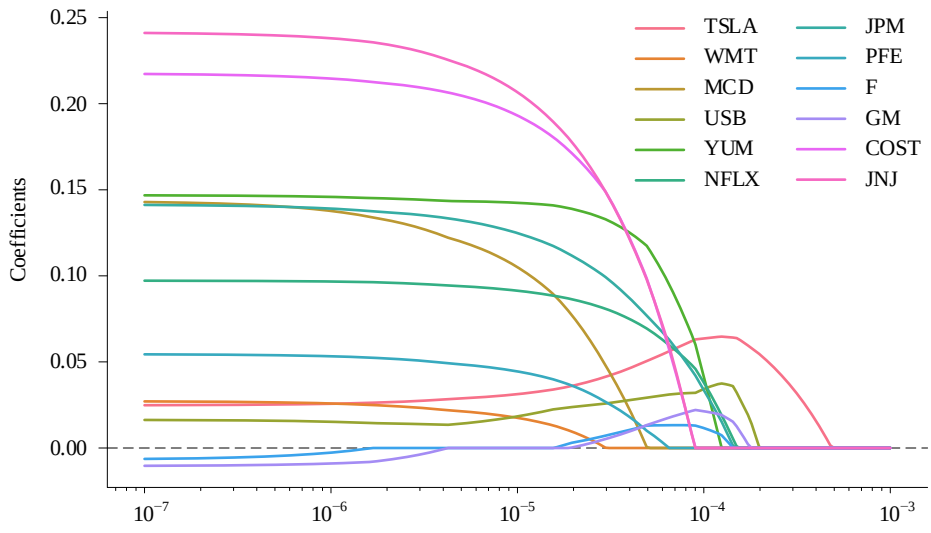


图 27. 随着 α 增大，弹性网络回归参数变化

图 28 比较套索回归和弹性网络回归参数随 α 变化；同样颜色的实线是套索回归参数，划线是弹性网络回归参数。容易发现，套索回归参数更快收缩到 0。弹性网络回归是套索回归和岭回归的结合体，它继承了套索回归的稀疏性，可以用来筛选特征，缩减无关参数。但是，由于引入岭回归 L2 正则项，弹性网络回归在淘汰特征的过程要慢于套索回归。

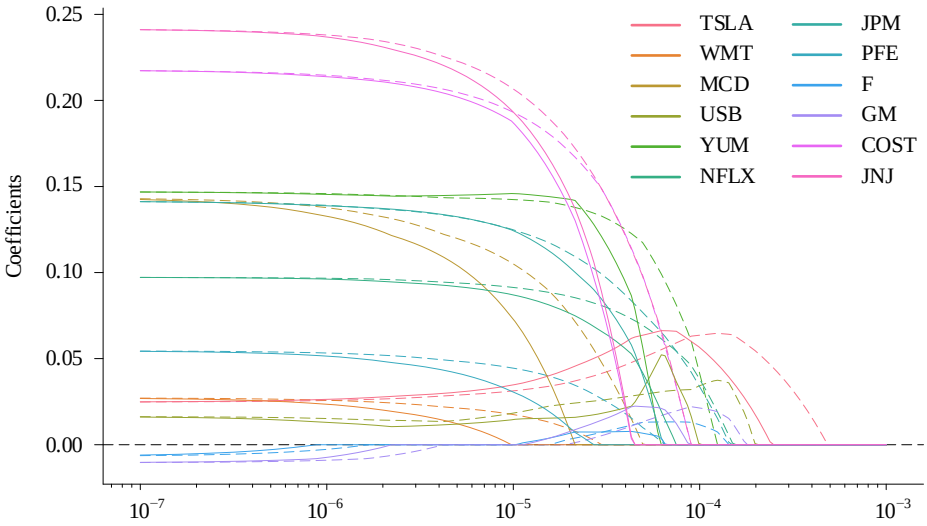


图 28. 比较套索回归和弹性网络回归参数随 α 变化

图 29 所示为和 OLS 相比，弹性网络回归参数误差。

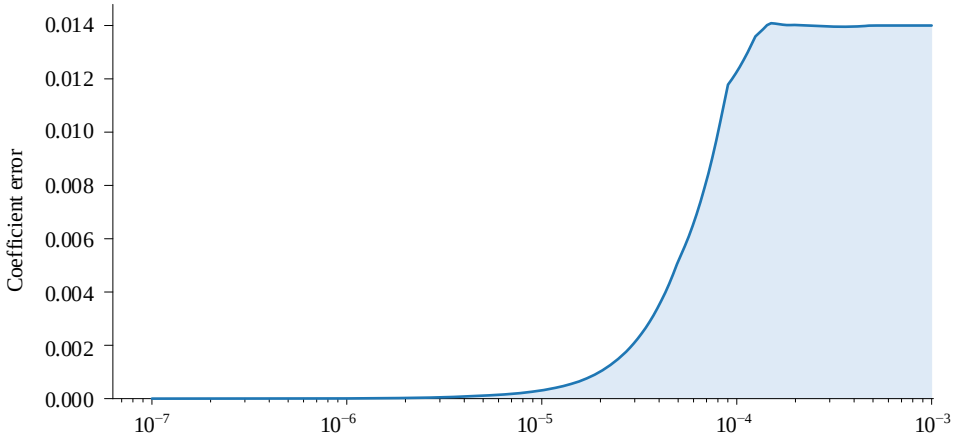
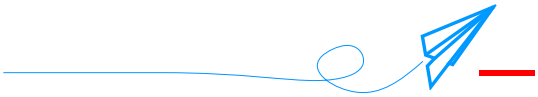


图 29. 和 OLS 相比，弹性网络回归参数误差



正则化是一种常用的机器学习技术，用于减小模型的复杂度和提高泛化能力。它通过在损失函数中添加一个正则项，强制模型参数的取值不要过大，从而避免模型过拟合。正则化技术包括 L1 正则化和 L2 正则化两种，L1 正则化将模型参数向 0 稀疏化，L2 正则化将模型参数平滑化，对于不同的数据集和模型结构可以选择不同的正则化方法。正则化技术在实际应用中被广泛使用，可以提高模型的预测能力和稳定性，避免过拟合和欠拟合等问题。

推荐大家阅读 *Statistical Learning with Sparsity: The Lasso and Generalizations*。本书是稀疏统计学习专著。图书 PDF 文件可以免费从如下网址下载。

https://web.stanford.edu/~hastie/StatLearnSparsity/

有关岭回归，建议大家阅读 *Lecture notes on ridge regression*。下载地址如下：

https://arxiv.org/abs/1509.09169

12 Bayesian Regression
贝叶斯回归

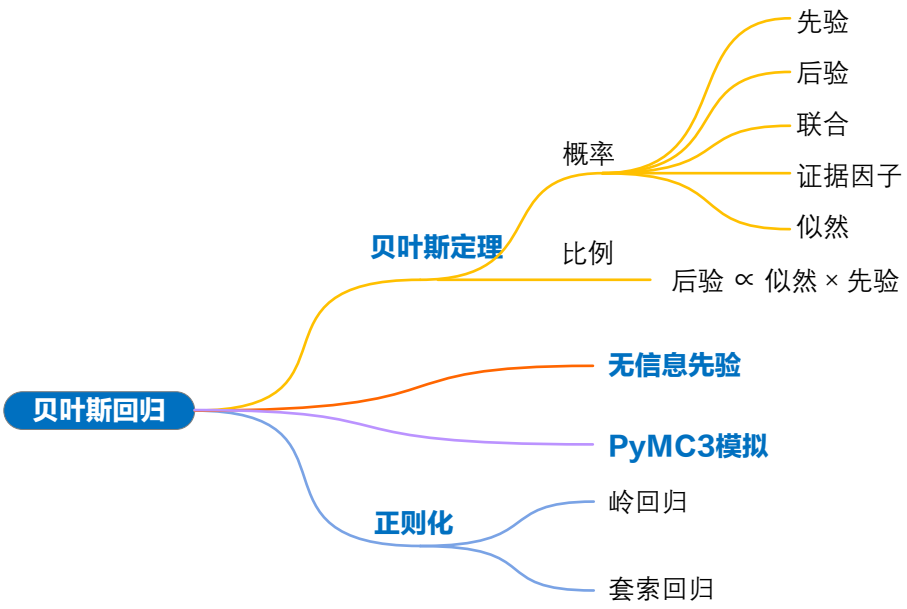
用贝叶斯推断求解回归模型参数

审视数学，你会发现，它不仅是颠扑不破的真理，而且是至高无上的美丽——那种冷峻而朴素的美，不需要唤起人们任何的怜惜，没有绘画和音乐的浮华装饰，纯粹，只有伟大艺术才能展现出来的严格完美。

Mathematics, rightly viewed, possesses not only truth, but supreme beauty — a beauty cold and austere, like that of sculpture, without appeal to any part of our weaker nature, without the gorgeous trappings of painting or music, yet sublimely pure, and capable of a stern perfection such as only the greatest art can show.

—— 伯特兰·罗素 (Bertrand Russell) | 英国哲学家、数学家 | 1872 ~ 1970

- `pymc3.Normal()` 定义正态先验分布
- `pymc3.HalfNormal()` 定义半正态先验分布
- `pymc3.plot_posterior()` 绘制后验分布
- `pymc3.sample()` 产生随机数
- `pymc3.traceplot()` 绘制后验分布随机数轨迹图



12.1 回顾贝叶斯推断

简单来说，**贝叶斯推断** (Bayesian inference) 就是结合“经验 (先验)”和“实践 (样本)”，得出“结论 (后验)”。贝叶斯推断把模型参数看作随机变量。在得到样本之前，根据主观经验和既有知识给出未知参数的概率分布叫做**先验分布** (prior)。获得样本数据后，根据贝叶斯定理，基于给定的样本数据先计算**似然分布** (likelihood)，然后模型参数的**后验分布** (prior)。

上面这段文字对应如下这个公式：

$$\overbrace{f_{\Theta|X}(\theta|x)}^{\text{Posterior}} = \frac{\overbrace{f_{X|\Theta}(x|\theta)}^{\text{Likelihood}}\overbrace{f_{\Theta}(\theta)}^{\text{Prior}}}{\int_{\vartheta} f_{X|\Theta}(x|\vartheta) f_{\Theta}(\vartheta)\mathrm{d}\vartheta} \tag{1}$$

最后根据参数的后验分布进行统计推断。贝叶斯推断对应的优化问题为**最大化后验概率** (Maximum A Posteriori, MAP)。本章介绍如何利用贝叶斯推断完成线性回归。

大家如果对 (1) 感到陌生的话，请回顾《统计至简》第 20、21 两章。

线性回归模型

为了配合贝叶斯推断，把多元线性回归模型写成：

$$\hat{y}^{(i)} = \theta_0 + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \cdots + \theta_D x_D^{(i)} \tag{2}$$

其中，i 为样本序号，D 为特征数。

当 $D = 1$ 时，一元线性回归模型为：

$$\hat{y}^{(i)} = \theta_0 + \theta_1 x_1^{(i)} \tag{3}$$

似然

似然函数可以写成：

$$f_{\gamma|\Theta}(\boldsymbol{y}|\boldsymbol{\theta}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(y^{(i)} - \left(\theta_0 + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \cdots + \theta_D x_D^{(i)} \right) \right)^2}{2\sigma^2} \right) \tag{4}$$

这意味着假设残差 ε 服从 $N(0, \sigma^2)$。

贝叶斯定理

利用贝叶斯定理，我们可以得到后验分布：

$$f_{\Theta|\gamma}(\boldsymbol{\theta}\,|\,\boldsymbol{y}) = \frac{f_{\gamma|\Theta}(\boldsymbol{y}\,|\,\boldsymbol{\theta})\cdot f_{\Theta}(\boldsymbol{\theta})}{f_{\gamma}(\boldsymbol{y})} \tag{5}$$

最大后验优化：

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} f_{\Theta|\gamma}(\boldsymbol{\theta}\,|\,\boldsymbol{y}) \tag{6}$$

如图 1 所示，随着样本不断引入，MAP 优化结果不断接近真实参数。

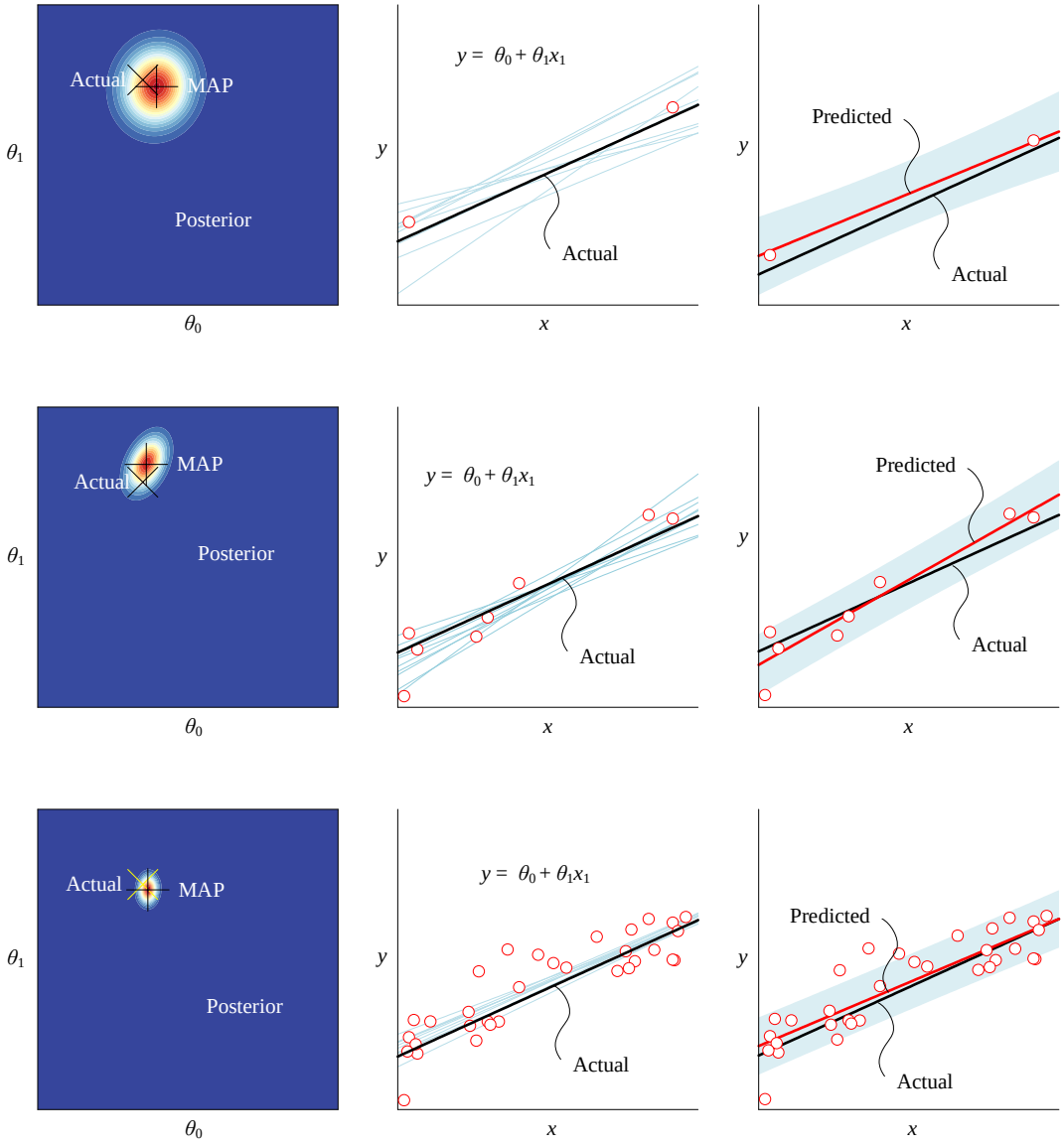


图 1. 贝叶斯回归后验概率随样本变化

由于后验 ∝ 似然 × 先验，最大后验优化等价于：

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} f_{\boldsymbol{\gamma}|\Theta}\left(\boldsymbol{y}\,|\,\boldsymbol{\theta}\right)\cdot f_{\Theta}\left(\boldsymbol{\theta}\right) \tag{7}$$

为了避免算数下溢，取对数后，优化问题可以写成：

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} \ln\left(f_{\boldsymbol{\gamma}|\Theta}\left(\boldsymbol{y}\,|\,\boldsymbol{\theta}\right)\cdot f_{\Theta}\left(\boldsymbol{\theta}\right)\right) \tag{8}$$

鸢尾花书之前介绍过，**算术下溢** (arithmetic underflow) 也称为**浮点数下溢** (floating point underflow)，是指计算机浮点数计算的结果小于可以表示的最小数。

(8) 进一步整理为：

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} \ln f_{\boldsymbol{\gamma}|\Theta}\left(\boldsymbol{y}\,|\,\boldsymbol{\theta}\right) + \ln f_{\Theta}\left(\boldsymbol{\theta}\right) \tag{9}$$

12.2 贝叶斯回归：无信息先验

《统计至简》第 20 章介绍过**无信息先验** (uninformative prior)。

没有先验信息，或者先验分布不清楚，我们可以用常数或均匀分布作为先验分布，比如 $f(\boldsymbol{\theta}) = 1$。最大后验优化就可以写成：

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} \ln f_{\boldsymbol{\gamma}|\Theta}\left(\boldsymbol{y}\,|\,\boldsymbol{\theta}\right) \tag{10}$$

这和 MLE 的目标函数一致：

$$\hat{\boldsymbol{\theta}}_{\text{MLE}} = \arg\max_{\boldsymbol{\theta}} \ln f\left(\boldsymbol{y};\boldsymbol{\theta}\right) \tag{11}$$

将 (4) 代入 $\ln f\left(\boldsymbol{y}\,|\,\boldsymbol{\theta}\right)$ 得到：

$$\begin{aligned} \ln f_{\boldsymbol{\gamma}|\Theta}\left(\boldsymbol{y}\,|\,\boldsymbol{\theta}\right) &= -\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y^{(i)} - \left(\theta_0 + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \cdots + \theta_D x_D^{(i)}\right)\right)^2 + \underbrace{n\ln\frac{1}{\sqrt{2\pi\sigma^2}}}_{\text{Constant}} \\ &= -\frac{\left\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\right\|_2^2}{2\sigma^2} + \underbrace{n\ln\frac{1}{\sqrt{2\pi\sigma^2}}}_{\text{Constant}} \end{aligned} \tag{12}$$

忽略常数，最大化后验 MAP 优化问题等价于如下最小化问题：

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\min_{\boldsymbol{\theta}} \left\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\right\|_2^2 \tag{13}$$

这和前文的 OLS 线性回归优化问题一致。

12.3 使用 PyMC3 完成贝叶斯回归

本节利用 PyMC3 完成模型为 $y = \theta_0 + \theta_1 x_1$ 贝叶斯回归。如图 2 所示，黑色线为真实模型，参数为截距 $\theta_0 = 1$、斜率 $\theta_1 = 2$。图 2 中蓝色散点为样本点。

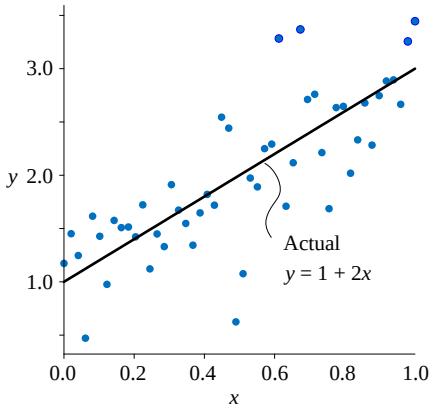


图 2. 真实模型和样本点

图 3 所示为三个参数的后验分布随机数轨迹图。随机数轨迹由 PyMC3 中**马尔科夫链蒙特卡洛** (Markov Chain Monte Carlo, MCMC) 生成。图中只绘制达到平稳状态后的轨迹。每个参数模拟两条轨迹。

前文提过残差 ε 服从 $N(0, \sigma^2)$，所以残差也是一个模型参数。本章配套代码中，残差的先验分布为**半正态分布** (half normal distribution)，如图 4 所示。有关半正态分布，大家可以参考：

https://www.pymc.io/projects/docs/en/latest/api/distributions/generated/pymc.HalfNormal.html

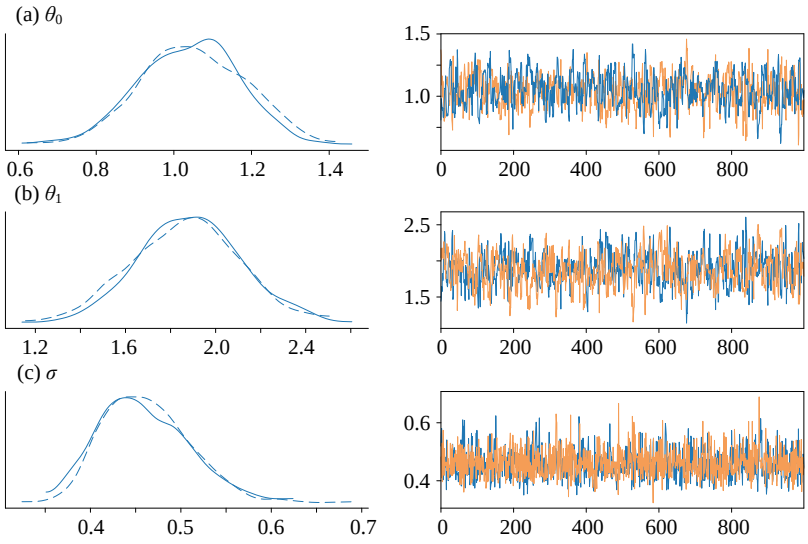


图 3. 后验分布随机数轨迹图

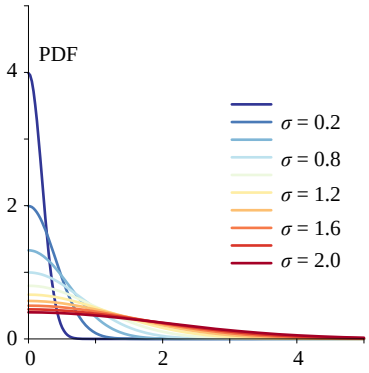


图 4. 半正态分布概率密度曲线

图 5 所示为后验分布随机数的直方图。直方图合并两条 MCMC 轨迹。图中均值可以视作 MAP 的优化解。HDI 代表**最大密度区间** (highest density interval)，即后验分布的可信区间。可信区间越窄，后验分布的确信度越高。图 6 所示为参数 θ_0 和 θ_1 后验分布随机生数构成的分布。

图 7 所示为贝叶斯线性回归的结果，图中红色线为预测模型。图中的浅蓝色线为 50 条后验分布的采样函数，它们对应图 6 中的 50 个散点。红色线相当于这些浅蓝色线“取平均”。

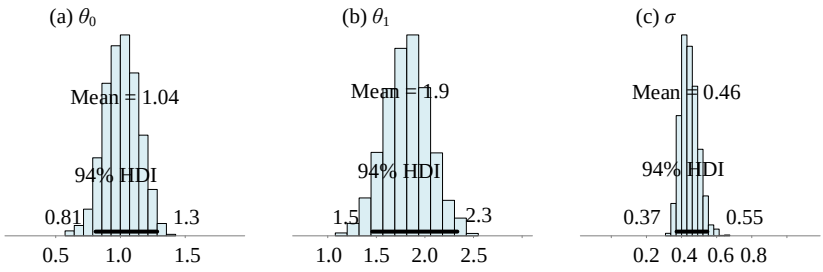


图 5. 后验分布直方图

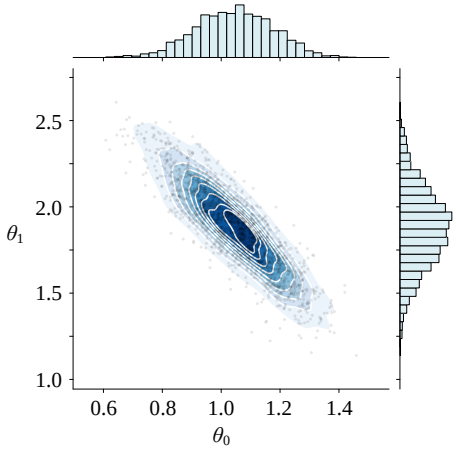


图 6. 参数 θ_0 和 θ_1 后验分布随机生数构成的分布

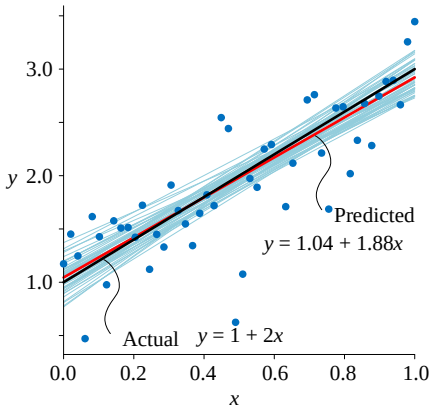


图 7. 贝叶斯线性回归结果

Bk6_Ch12_01.ipynb 绘制本节图像。

12.4 贝叶斯视角理解 Ridge 正则化

上一章的岭回归可以从贝叶斯推断角度理解。

本章中假设线性回归参数服从正态分布：

$$f_{\Theta_j}\left(\theta_j\right)=\frac{1}{\sqrt{2\pi\tau^2}}\exp\left(-\frac{\theta_j^2}{2\tau^2}\right) \tag{14}$$

图 8 所示为先验分布随 τ 变化。τ 越大代表越不确信，τ 越小代表确信程度越高。

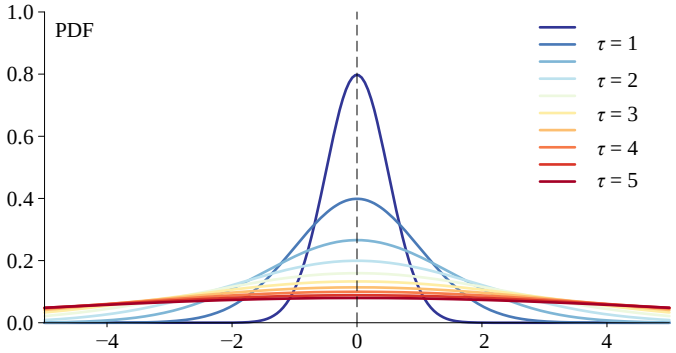


图 8. 先验分布随 τ 变化

(8) 所示的优化问题等价于：

$$\arg\max_{\theta}\left[\ln\prod_{i=1}^{n}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{\left(y^{(i)}-\left(\theta_0+\theta_1x_1^{(i)}+\theta_2x_2^{(i)}+\cdots+\theta_Dx_D^{(i)}\right)\right)^2}{2\sigma^2}\right)+\ln\prod_{j=1}^{D}\frac{1}{\sqrt{2\pi\tau^2}}\exp\left(-\frac{\theta_j^2}{2\tau^2}\right)\right] \tag{15}$$

上式目标函数可以分为两部分整理。大家已经清楚，第一部分为：

$$-\frac{\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\theta}\|_2^2}{2\sigma^2}+\underbrace{n\ln\frac{1}{\sqrt{2\pi\sigma^2}}}_{\text{Constant}} \tag{16}$$

第二部分为：

$$-\frac{\|\boldsymbol{\theta}\|_2^2}{2\tau^2}+\underbrace{D\ln\frac{1}{\sqrt{2\pi\tau^2}}}_{\text{Constant}} \tag{17}$$

忽略常数后，(15) 优化问题进一步整理为：

$$\arg\max_{\theta}\left[-\frac{\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\theta}\|_2^2}{2\sigma^2}-\frac{\|\boldsymbol{\theta}\|_2^2}{2\tau^2}\right] \tag{18}$$

将上式最大化问题调整为最小化问题：

$$\arg\min_{\theta}\frac{1}{2\sigma^2}\left(\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\theta}\|_2^2+\frac{\sigma^2}{\tau^2}\|\boldsymbol{\theta}\|_2^2\right) \tag{19}$$

令

$$\lambda=\frac{\sigma^2}{\tau^2} \tag{20}$$

(19) 等价于：

$$\arg\min_{\theta}\underbrace{\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\theta}\|_2^2}_{\text{OLS}}+\underbrace{\lambda\|\boldsymbol{\theta}\|_2^2}_{\text{L2 regularizer}} \tag{21}$$

这和上一章的岭回归优化问题完全一致。

《统计至简》第 20 章介绍过，先验的影响力很大，MAP 的结果向先验均值“收缩”。这种效果常被称作**贝叶斯收缩** (Bayes shrinkage)。

根据 (20)，σ 保持不变条件下，τ 越小代表确信度越高，λ 越大，通过 MAP 得到的优化解向原点 $\boldsymbol{0}$ (先验均值) 收缩。图 9 上可以看到，优化解随着约束项参数 λ 不断增大运动轨迹，“收缩”的这种现象显而易见。

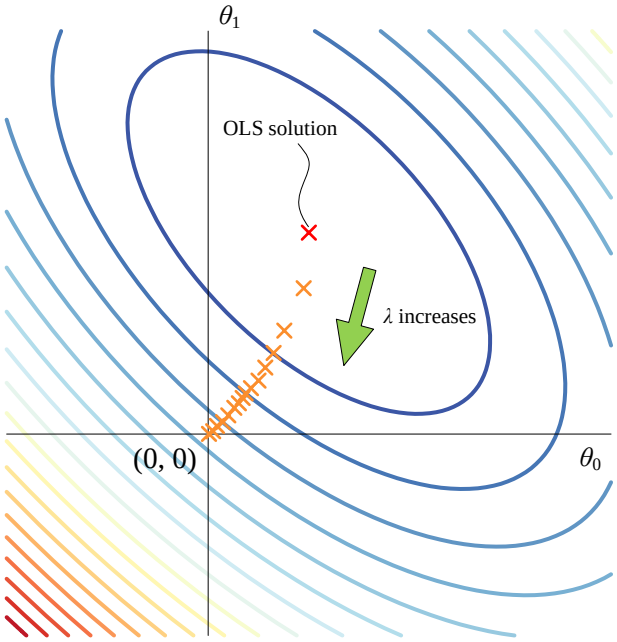


图 9. 不断增大 λ，岭回归优化解变化路径

12.5 贝叶斯视角理解套索正则化

如果先验分布为拉普拉斯分布：

$$f_{\Theta_j}\left(\theta_j\right)=\frac{1}{2b}\exp\left(-\frac{\left|\theta_j\right|}{b}\right) \tag{22}$$

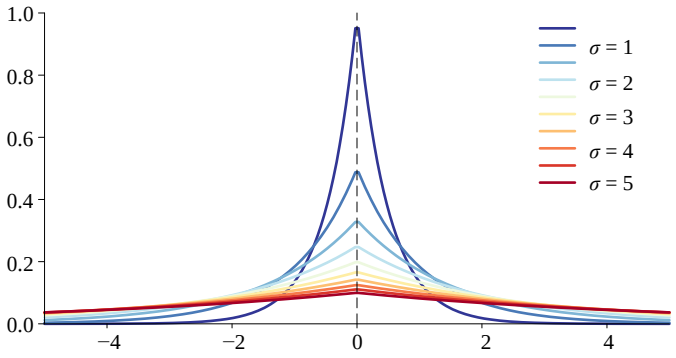


图 10. 先验分布随 b 变化

(8) 所示的优化问题等价于：

$$\arg\max_{\theta}\left[\ln\prod_{i=1}^{n}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{\left(y^{(i)}-\left(\theta_0+\theta_1x_1^{(i)}+\theta_2x_2^{(i)}+\cdots+\theta_Dx_D^{(i)}\right)\right)^2}{2\sigma^2}\right)+\ln\prod_{j=1}^{D}\frac{1}{2b}\exp\left(-\frac{|\theta_j|}{b}\right)\right] \tag{23}$$

也是分两部分来看上式。第一部分和上一节完全相同：

$$-\frac{\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\theta}\|_2^2}{2\sigma^2}+\underbrace{n\ln\frac{1}{\sqrt{2\pi\sigma^2}}}_{\text{Constant}} \tag{24}$$

第二部分为：

$$-\frac{1}{b}\sum_{j=1}^{D}|\theta_j|+\underbrace{D\ln\frac{1}{2b}}_{\text{Constant}}=-\frac{1}{b}\|\boldsymbol{\theta}\|_1+\underbrace{D\ln\frac{1}{2b}}_{\text{Constant}} \tag{25}$$

忽略常数后，优化问题为：

$$\arg\max_{\theta}-\frac{\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\theta}\|_2^2}{2\sigma^2}-\frac{1}{b}\|\boldsymbol{\theta}\|_1 \tag{26}$$

最大化问题调整为最小化问题得到：

$$\arg\min_{\theta}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\theta}\|_2^2+\frac{2\sigma^2}{b}\|\boldsymbol{\theta}\|_1 \tag{27}$$

令

$$\lambda=\frac{2\sigma^2}{b} \tag{28}$$

(27) 等价于

$$\arg\min_{\theta}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\theta}\|_2^2+\lambda\|\boldsymbol{\theta}\|_1 \tag{29}$$

这和上一章套索回归的优化问题的目标函数本质上一致。

图 11 所示为不断增大 λ，套索回归参数变化轨迹；可以发现参数变化轨迹有两段，第一段从 OLS 结果为起始点，几乎沿着斜线靠近 y 轴 ($\theta_0 = 0$)，直至到达 y 轴。到达 y 轴时，回归系数 θ_0 为 0。第二段，沿着 y 轴朝着原点运动。

请大家自己思考从贝叶斯推断视角来看，套索回归的先验概率分布应该是什么？

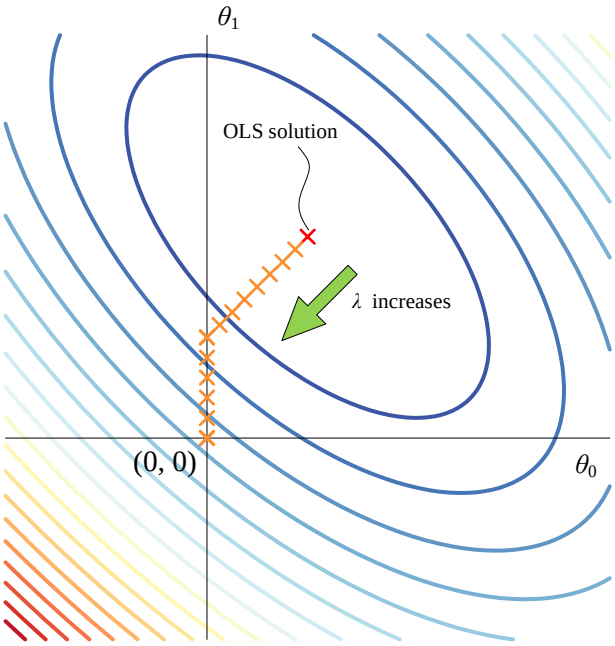


图 11. 不断增大 λ，套索回归优化解变化轨迹

贝叶斯回归是一种基于贝叶斯理论的回归分析方法，它不仅考虑了自变量与因变量之间的线性关系，还考虑了模型的不确定性和误差。在贝叶斯回归中，模型的参数被视为概率变量，因此可以通过贝叶斯定理来计算模型参数的后验分布，从而得到对未来数据的预测结果。贝叶斯回归不仅可以有效地避免过拟合和欠拟合等问题，还可以处理噪声和缺失数据等复杂情况，具有广泛的应用前景。

从贝叶斯回归角度理解正则化回归，可以将正则化项视为参数的先验分布。正则化回归通过在损失函数中加入先验分布，来约束模型参数的取值范围，从而避免过拟合和提高泛化能力。在贝叶斯回归中，先验分布可以通过经验知识或者领域知识来确定，这种方法可以更好地适应实际问题的复杂性和不确定性。因此，正则化回归可以看作是贝叶斯回归在参数估计中的一种特殊情况。

想深入学习贝叶斯推断和贝叶斯回归的读者可以参考开源图书 *Bayesian Modeling and Computation in Python*：

https://bayesiancomputationbook.com/welcome.html

13 非线性回归

Moving Beyond Linearity

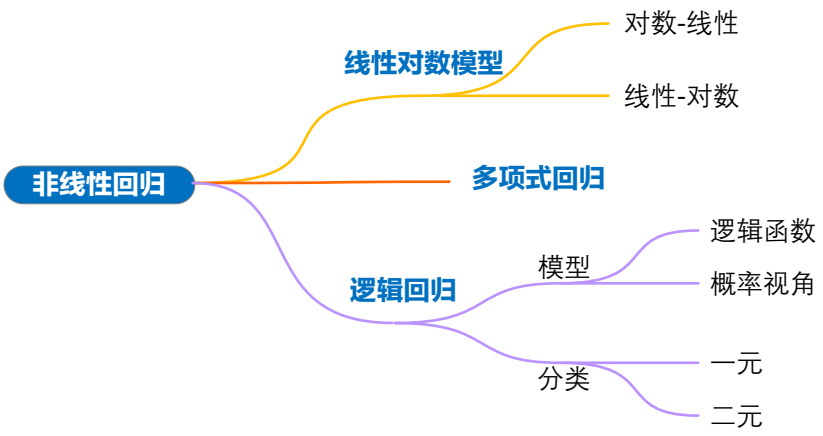
寻找因变量和自变量之间关系的非线性模型

科学不去尝试辩解，甚至几乎从来不解读；科学主要工作就是数学建模。模型是一种数学构造；基于少量语言说明，每个数学构造描述观察到的现象。数学模型合理之处是它具有一定的普适性；此外，数学模型一般具有优美的形式——也就是不管它能解释多少现象，它必须相当简洁。

The sciences do not try to explain, they hardly even try to interpret, they mainly make models. By a model is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work.

—— 约翰·冯·诺伊曼 (John von Neumann) | 美国籍数学家 | 1903 ~ 1957

- `matplotlib.pyplot.contour()` 绘制等高线线图
- `matplotlib.pyplot.contourf()` 绘制填充等高线图
- `matplotlib.pyplot.getp()` 获绘图对象的属性
- `matplotlib.pyplot.plot_wireframe()` 绘制线框图
- `matplotlib.pyplot.scatter()` 绘制散点图
- `matplotlib.pyplot.setp()` 设置绘图对象的一个或者多个属性
- `numpy.random.normal()` 产生服从高斯分布的随机数
- `numpy.random.rand()` 产生服从均匀分布的随机数
- `numpy.random.randn()` 产生服从标准正态分布的随机数
- `scipy.special.expit()`
- `seaborn.jointplot()` 绘制联合分布/散点图和边际分布
- `seaborn.kdeplot()` 绘制概率密度估计曲线
- `seaborn.scatterplot()` 绘制散点图
- `sklearn.linear_model.LinearRegression()` 最小二乘法回归
- `sklearn.linear_model.LogisticRegression()` 逻辑回归函数，也可以用来分类
- `sklearn.pipeline.Pipeline()` 将许多算法模型串联起来形成一个典型的机器学习问题工作流
- `sklearn.preprocessing.FunctionTransformer()` 根据函数对象或者自定义函数处理样本数据
- `sklearn.preprocessing.PolynomialFeatures()` 建模过程中构造多项式特征



13.1 线性回归

本书前文介绍过线性回归，白话说，线性回归使用直线、平面或超平面来预测。多元线性回归的数学表达式如下：

$$y = b_0 + b_1 x_1 + b_2 x_2 + ... + b_D x_D + \varepsilon \tag{1}$$

可以发现 $x_1, x_2, ..., x_D$ 这几个变量的次数都是一次，这也就是“线性”一词的来由。图 1 所示为最小二乘法多元线性回归数据关系。

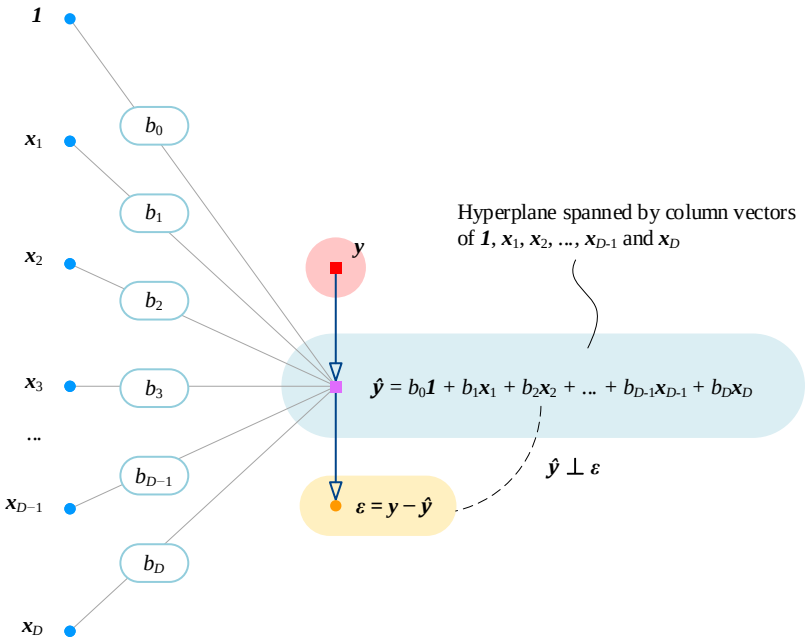


图 1. 最小二乘法多元线性回归数据关系

此外，特征还可以进行线性组合得到一系列新特征：

$$\boldsymbol{z}_k = v_{1,k}\boldsymbol{x}_1 + v_{2,k}\boldsymbol{x}_2 + \cdots + v_{D,k}\boldsymbol{x}_D = \phi_k\left(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_D\right) \tag{2}$$

即

$$\begin{aligned} \boldsymbol{Z} &= \begin{bmatrix} \boldsymbol{z}_1 & \cdots & \boldsymbol{z}_p \end{bmatrix} = \begin{bmatrix} \phi_1(\boldsymbol{X}) & \cdots & \phi_p(\boldsymbol{X}) \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_D \end{bmatrix} \begin{bmatrix} v_{1,1} & \cdots & v_{1,p} \\ v_{2,1} & \cdots & v_{2,p} \\ \vdots & \ddots & \vdots \\ v_{D,1} & \cdots & v_{D,p} \end{bmatrix} \end{aligned} \tag{3}$$

然后可以用最小二乘求解回归系数：

$$\hat{\boldsymbol{y}} = \boldsymbol{Z}\left(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{Z}\right)^{-1}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{y} \tag{4}$$

图 2 所示为线性组合的数据关系，得到的模型可以通过 (3) 反推得到基于 $x_1, x_2,\ldots, x_D$ 这几个变量的线性模型。本书后续介绍的基于主成分分析的回归方法采用的就是这一思路。

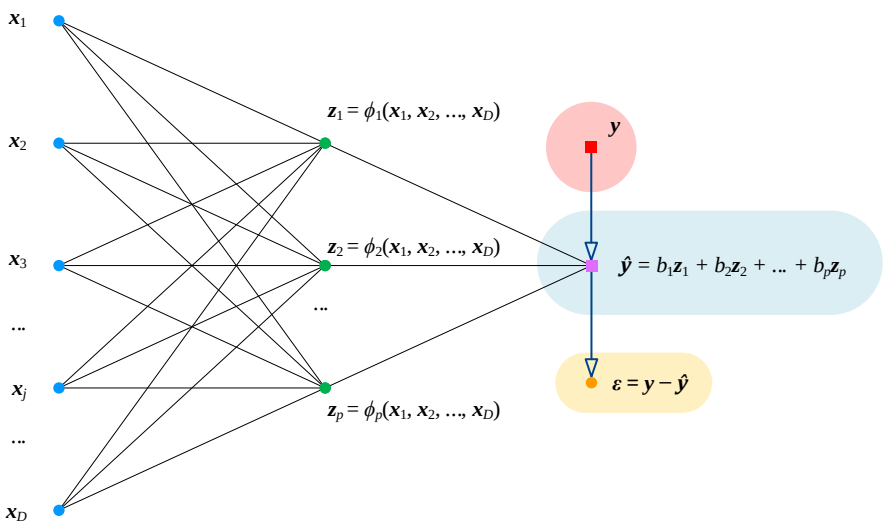


图 2. 特征线性组合

线性回归虽然简单，但是并非万能。图 3 给出的三组数据都不适合用线性回归来描述。本章就介绍如何采用几种非线性回归方法来解决线性回归不能解决的问题。

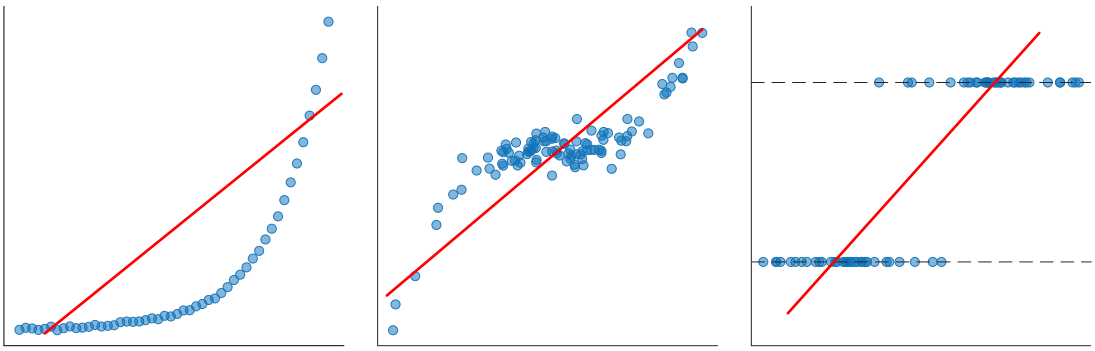


图 3. 线性回归失效的三个例子

13.2 线性对数模型

本书前文介绍过数据转换，一些回归问题可以对输入或输出进行数据转换，甚至对两者同时进行数据转换，之后再来构造线性模型。本节介绍几个例子。

观察图 4 (a)，容易发下样本数据呈现出“指数”形状，而且输出值 y 大于 0；容易想到对输出值 y 取对数，得到图 4 (b)。而图 4 (b) 展现出明显的线性回归特征，便于进行线性回归建模。

利用以上思路便可以得到所谓对数-线性模型：

$$\ln y = b_0 + b_1 x + \varepsilon \tag{5}$$

图 5 所示为通过拟合得到的对数-线性模型。

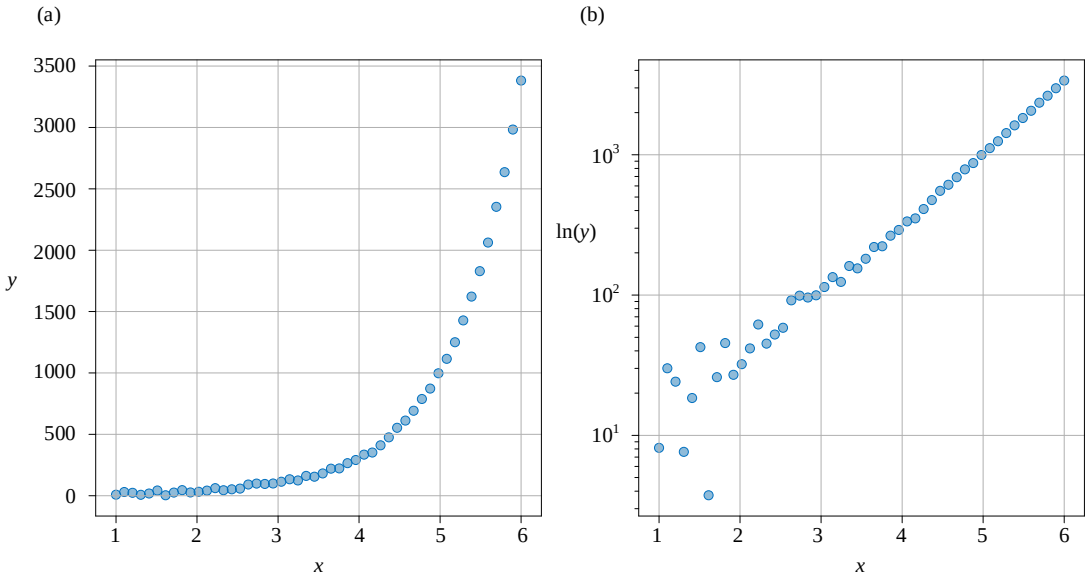


图 4. 类似“指数”形状的样本数据

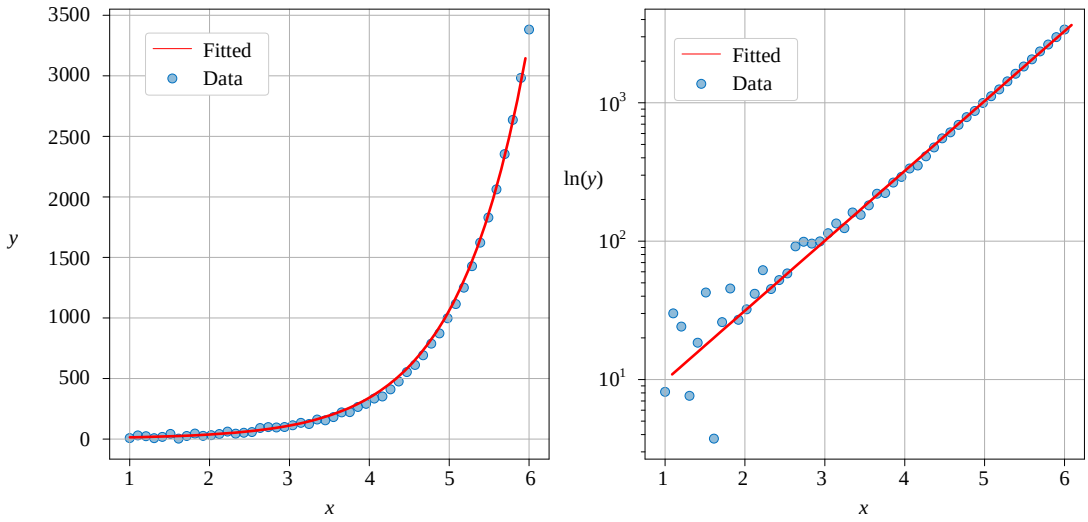


图 5. 对数-线性模型

反过来，当数据呈现类似“对数”形状时 (见图 6 (a))，可以对输入 x 去对数，得到图 6 (b)。观察图 6 (b)，可以发现数据展现出一定的线性关系。这样我们就可以使用线性-对数模型：

$$y = b_0 + b_1 \ln x + \varepsilon \tag{6}$$

图 7 所示为得到的线性-对数模型。

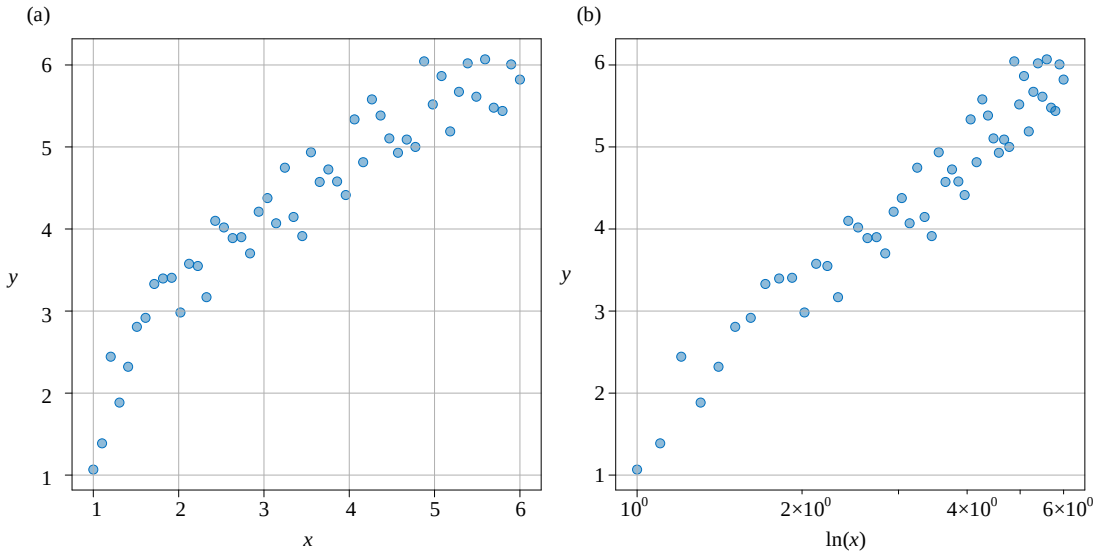


图 6. 类似"对数"形状的样本数据

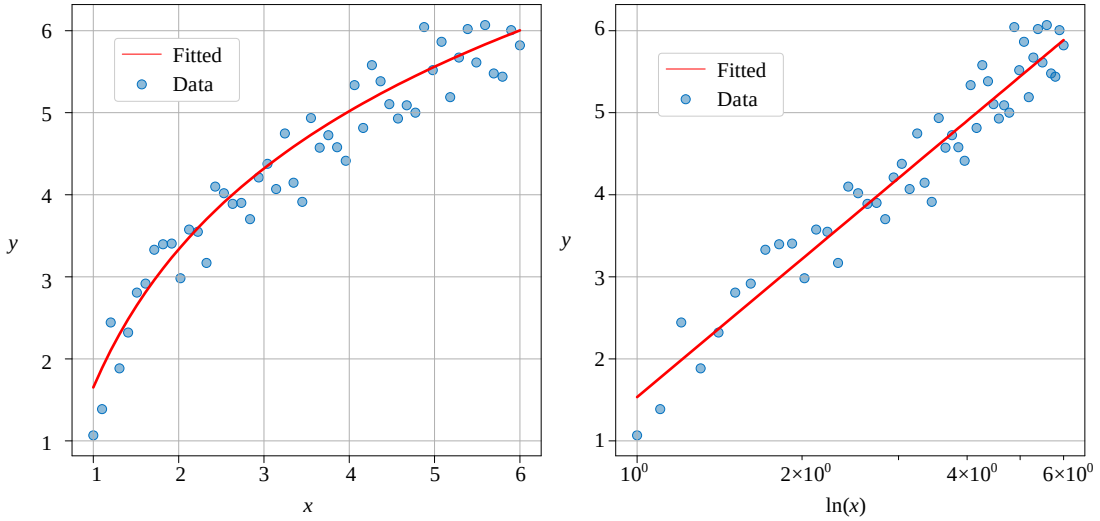


图 7. 线性-对数模型

此外，我们可以理解同时对输入和输出数据取对数，然后再构造线性回归模型；这种模型叫做双对数模型：

$$\ln y = b_0 + b_1 \ln x + \varepsilon \tag{7}$$

需要注意的是，进行对数变换的前提是，所有的观测值都必须大于 0。当观测值中存在 0 或者小于 0 的数值，可以对所有的观测值加 $-\min(x)+1$，然后再进行对数变换。

Bk6_Ch13_01.py 绘制本节图像。

13.3 非线性回归

非线性回归是一种回归分析方法，建立自变量与因变量之间的非线性关系模型，用于预测连续变量的值。非线性回归需要应对线性回归无法解决的复杂问题。

有些情况下，简单的将数据做对数处理是不够的，需要对数据做进一步处理。模型如下所示：

$$y = f(x) + \varepsilon \tag{8}$$

$f(x)$ 可以是任意函数，比如多项式函数，逻辑函数，甚至是分段函数。

(8) 中 $f(x)$ 可以是多项式，得到**多项式回归** (polynomial regression)。比如，一元三次多项式回归：

$$y = b_0 + b_1 x + b_2 x^2 + b_3 x^3 \tag{9}$$

图 8 所示为一元三次多项式回归模型数据关系。

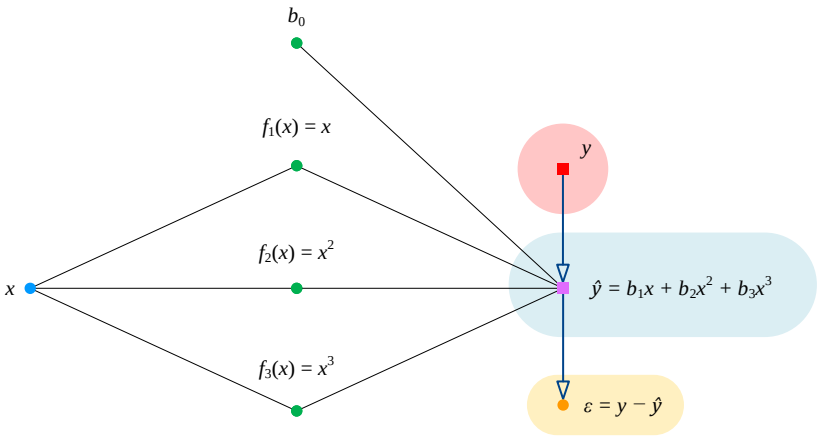


图 8. 一元三次多项式回归

图 9 所示为利用一元三次多项式回归模型来拟合样本数据。下一节，我们将仔细讲解多项式回归。

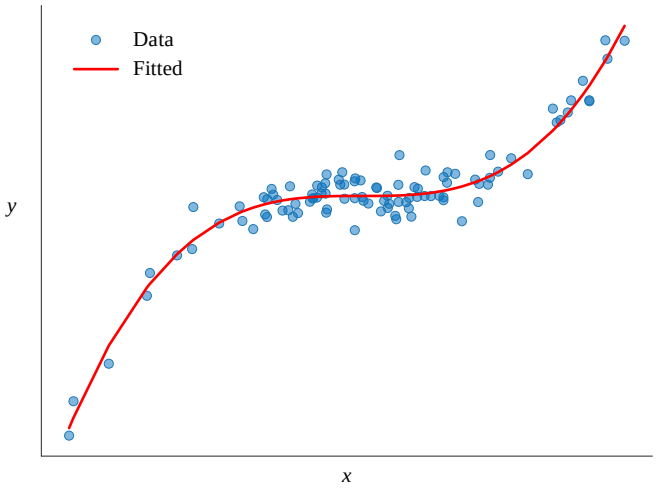


图 9. 一元三次多项式回归模型

逻辑回归 (logistic regression) 也是一种重要的非线性回归模型。一元逻辑回归模型如下：

$$y = \frac{1}{1+\exp\left(-\left(\underbrace{b_0 + b_1 x}_{\text{linear model}}\right)\right)} \tag{10}$$

图 10 所示为拟合数据得到的逻辑回归模型。图 11 所示为逻辑回归模型数据关系，逻辑回归模型可以看做时线性模型通过逻辑函数转换得到。

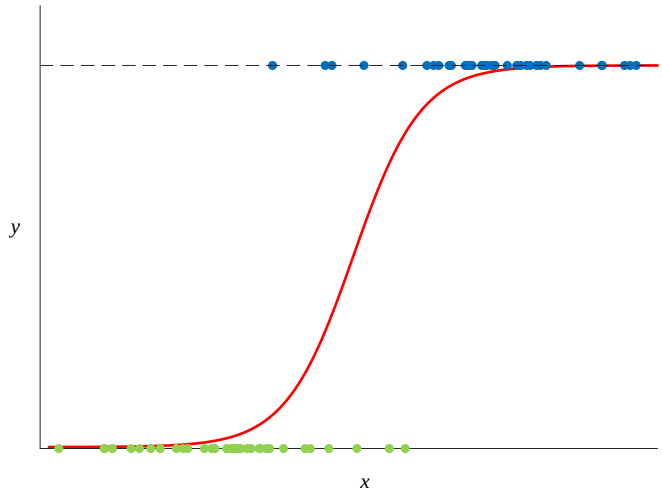


图 10. 逻辑回归模型

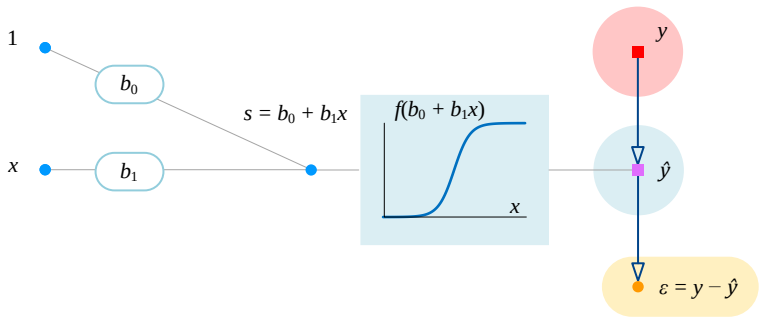


图 11. 逻辑回归数据关系

逻辑回归虽然是个回归模型，但是常被用作分类模型，用于二分类。

下一章将讲解逻辑回归。

此外，我们还可以用分段函数来拟合数据。如图 12 所示，两段线性函数用来拟合样本数据，效果也是不错的。

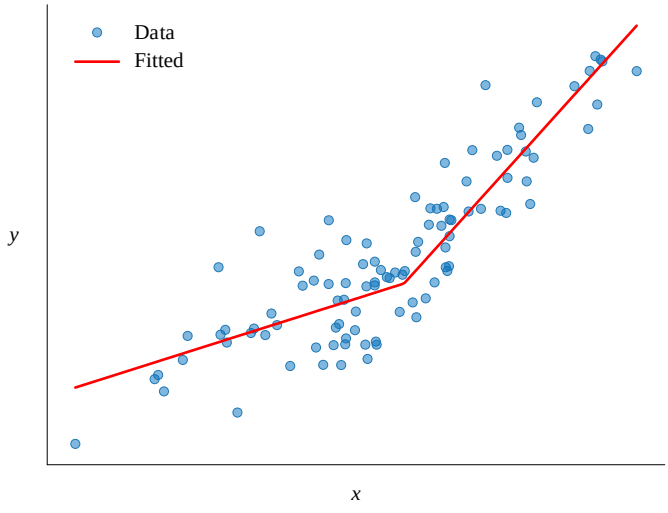


图 12. 分段函数模型

非参数回归 (non-parametric regression) 也是一种非常重要的非线性拟合方法。本章前面介绍的回归模型都有自身的“参数”，但是非参数回归模型并不假设回归函数的具体形式。参数回归分析时假定变量之间某种关系，然后估计参数；而非参数回归，则让数据本身说话。

比如，图 13 所示为采用**最邻近回归** (k-nearest neighbor regression)。最邻近可以用来分类，也可以用来构造回归模型。

鸢尾花书《机器学习》一书讲介绍最邻近方法。

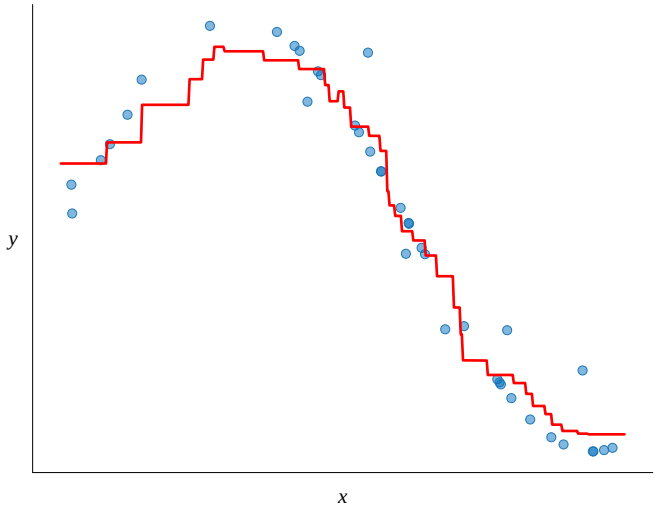


图 13. 最邻近回归

13.4 多项式回归

多项式回归是回归分析的一种形式，多项式回归是指回归函数的自变量的指数大于 1。在多项式回归中，一元回归模型最佳拟合线不是直线，而是一条拟合了数据点的多项式曲线。

图 14 所示为第一到五次一元函数的形状。

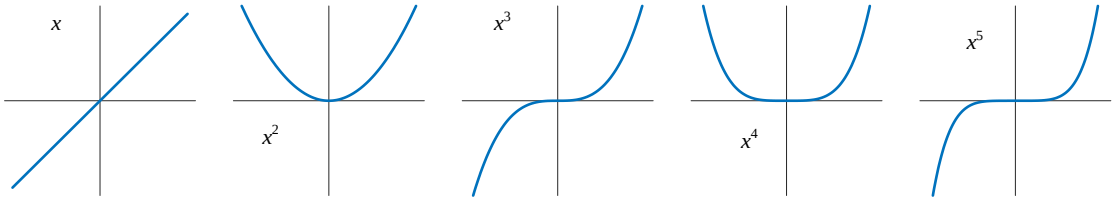


图 14. 一次到五次一元函数

自变量 x 和因变量 y 之间的关系被建模为关于 x 的 m 次多项式：

$$\hat{y} = b_0 + b_1 x + b_2 x^2 + \cdots + b_m x^m \tag{11}$$

其中，m 为多项式函数最高次项系数。

图 15 所示为一元多项式回归数据关系。

《矩阵力量》第 9 章介绍过采用矩阵运算得到多项式回归系数，请大家回顾。

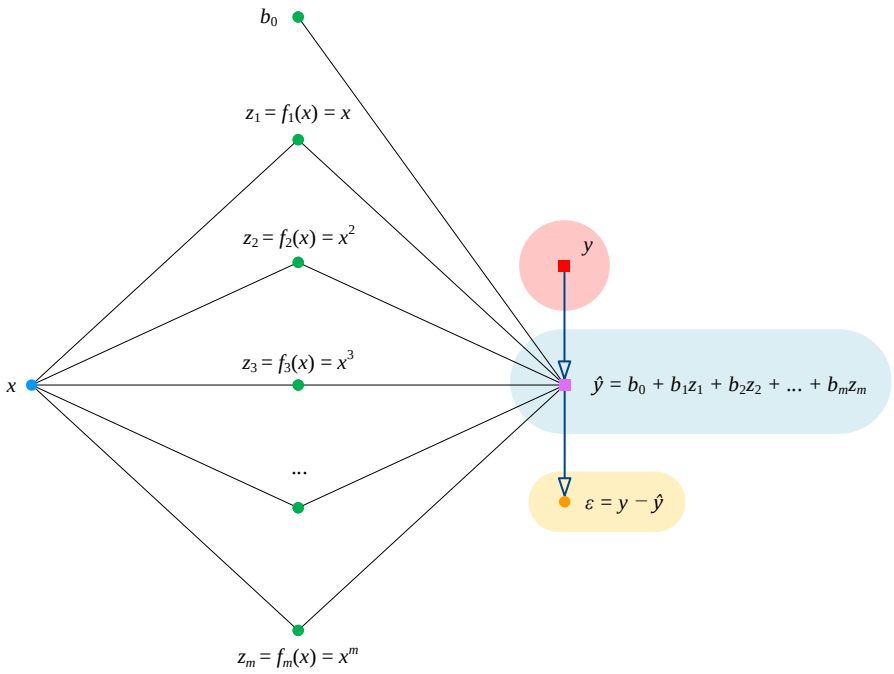


图 15. 一元多项式回归数据关系

图 16 所示为采用一次到四次一元多项式回归模型拟合样本数据。多项式回归的最大优点就是可以通过增加自变量的高次项对数据进行逼近。

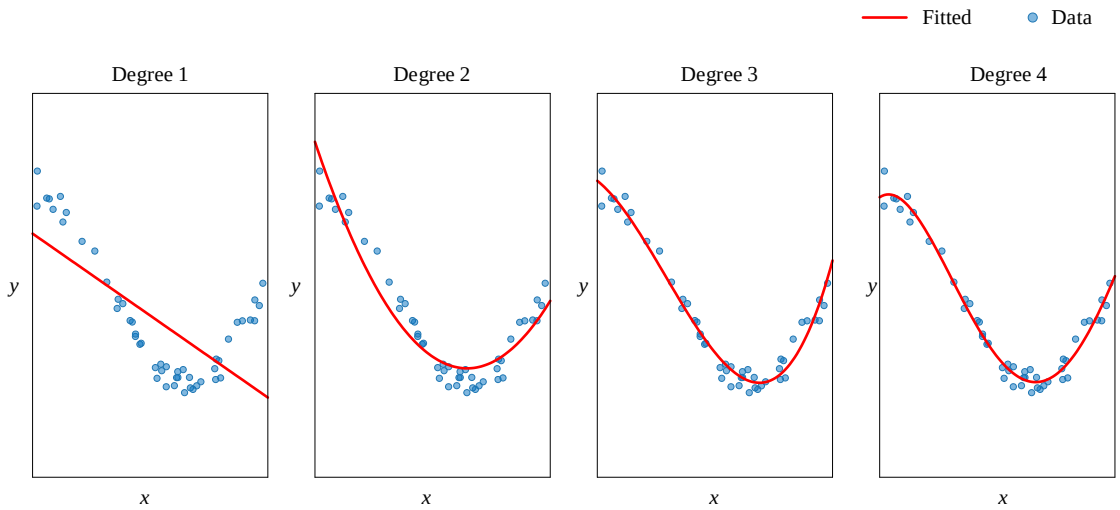


图 16. 一元多项式回归，一次到四次

但是，对于多项式回归，次数越高，越容易产生过度**拟合** (overfitting) 问题。过拟合发生的原因是，使用过于复杂的模型，导致模型过于精确地描述训练数据。如图 17 所示，采用过高次数的多项式回归模型，模型过于复杂，过度捕捉训练数据中的细节信息，甚至是噪音。但是，使用该模型预测其他样本数据时，会无法良好地预测未来观察结果。丛书后续还要深入探讨过拟合问题。

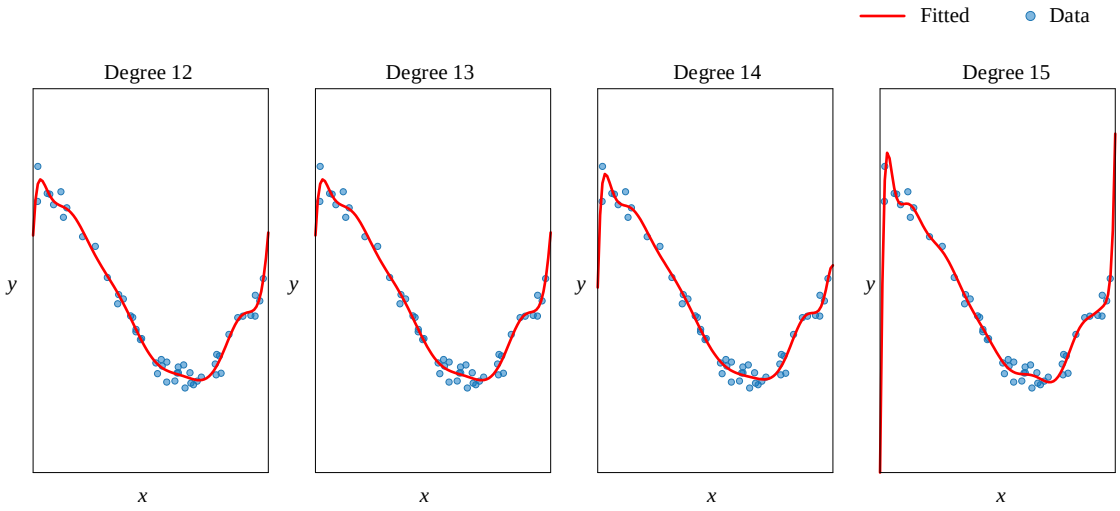


图 17. 一元多项式回归过度拟合，12 次到 15 次

此外，多项式回归可以有多个特征，而特征和特征之间可以形成较为复杂的多项式关系。比如，下式给出的是二元二次多项式回归：

$$f(x_1, x_2) = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_1 x_2 + b_4 x_1^2 + b_5 x_2^2 \tag{12}$$

(12) 相当于以一定比例组合图 18 所示的六个平面。提高多项式项次数，可以获得更加复杂的曲线或曲面，这样可以描述更加复杂的数据关系。因此不论因变量与其它自变量的关系如何，一般都可以尝试用多项式回归来进行分析。

图 19 所示为 (12) 所示的数据关系。

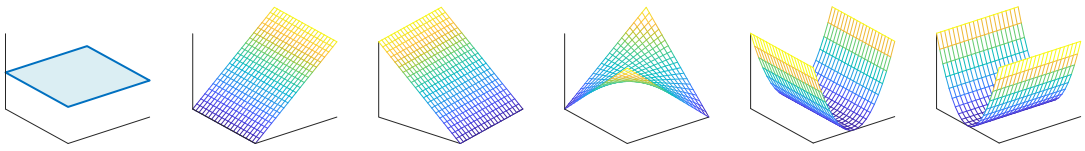


图 18. 六个二元平面/曲面

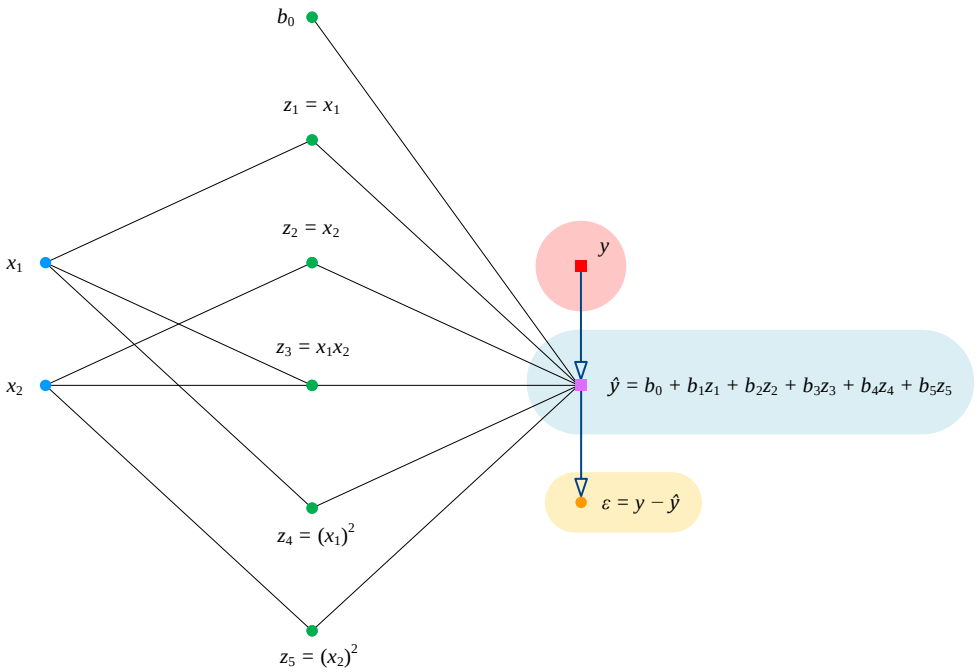


图 19. 二元二次多项式回归数据关系

Bk6_Ch13_02.py 绘制本节图像。

13.5 逻辑回归

图 20 给出一组数据的散点图，取值为 1 的数据点被标记为蓝色，取值为 0 的数据点被标记为红色。图 21 给出三种可以描述红蓝散点数据的函数。线性函数显然不适合这一问题。阶跃函数虽然可以捕捉函数从 0 到 1 的跳变，但是函数本身不光滑。

逻辑函数似乎能够胜任描述红蓝三点数据的任务。线性函数的因变量一般为连读数据；而逻辑函数的因变量为离散数值，即分类数据。

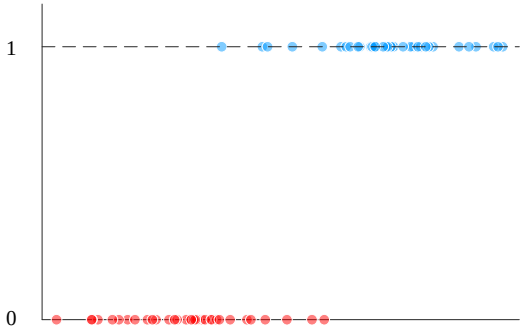


图 20. 红蓝数据的散点图

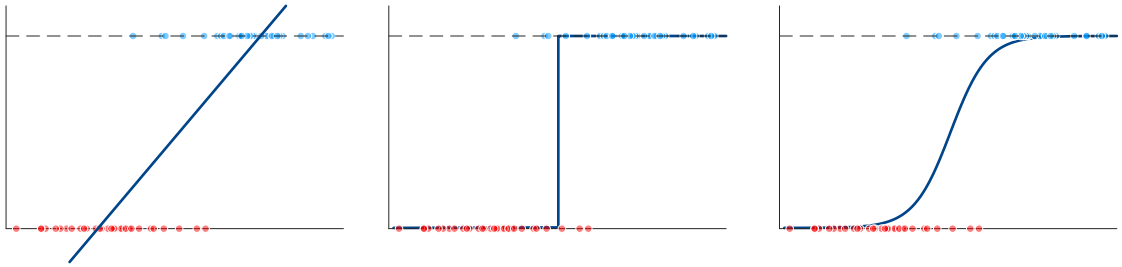


图 21. 可以描述红蓝数据的函数

逻辑函数

回顾《数学要素》12 章讲过的逻辑函数。

最简单的逻辑函数：

$$f(x)=\frac{1}{1+e^{-x}}=\frac{e^{x}}{1+e^{x}} \tag{13}$$

更一般的一元逻辑函数：

$$f(x)=\frac{1}{1+\exp\left(-\left(b_0+b_1x\right)\right)} \tag{14}$$

图 22 所示为 b_1 影响一元逻辑函数图像的陡峭程度。图中，$b_0=0$。可以发现函数呈现 S 形，取值范围在 [0, 1] 之间；函数在左右两端无限接近 0 或 1。函数的这一性质，方便从概率角度解释，这是下一节要介绍的内容。

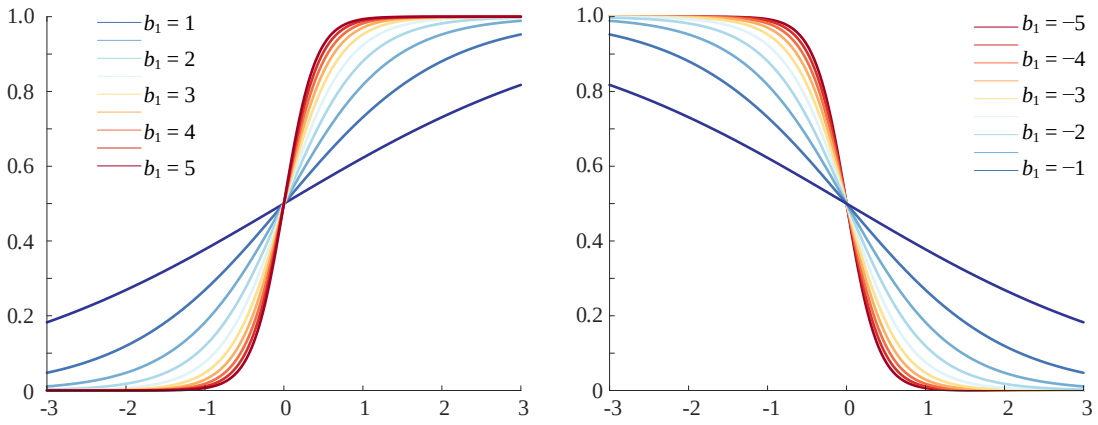


图 22. b_1影响一元逻辑函数图像的陡峭程度

找到 $f(x) = 1/2$ 位置：

$$f(x) = \frac{1}{1+\exp\left(-\left(b_0 + b_1 x\right)\right)} = \frac{1}{2} \tag{15}$$

整理得到 $f(x) = 1/2$ 对应的 x 值：

$$x = -\frac{b_0}{b_1} \tag{16}$$

也就是当 b_1 确定时，b_0 决定逻辑函数位置。注意，图 23 中，$b_1 = 0$。

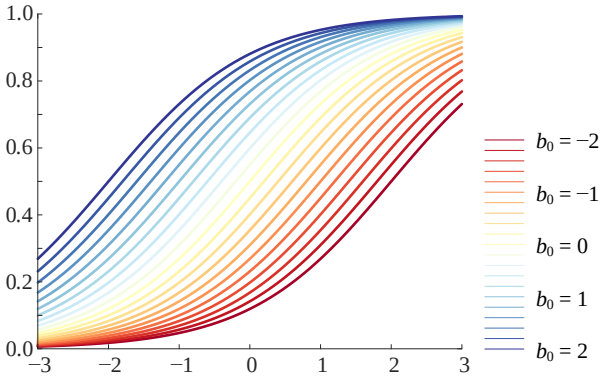


图 23. b_0决定逻辑函数位置，$b_1 = 0$

图 24 所示为根据数据的分布，选取不同的逻辑函数参数。

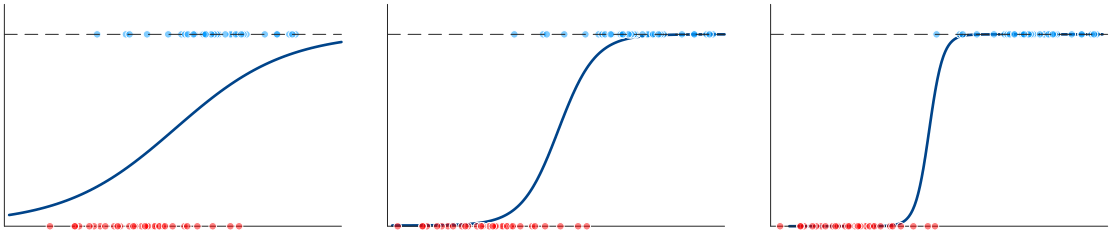


图 24. 根据数据的分布，选取不同的逻辑函数参数

Bk6_Ch13_03.py 绘制逻辑函数图像。

多元

对于多元情况，逻辑函数的一般式如下：

$$f\left(x_1, x_2, ..., x_D\right) = \frac{1}{1+\exp\left(-\left(b_0 + b_1 x_1 + b_2 x_2 + \cdots b_D x_D\right)\right)} \tag{17}$$

利用矩阵运算表达多元逻辑函数：

$$f\left(\boldsymbol{x}\right) = \frac{1}{1+\exp\left(-\boldsymbol{b}^{\mathrm{T}}\boldsymbol{x}\right)} \tag{18}$$

其中

$$\begin{aligned} \boldsymbol{x} &= \begin{bmatrix} 1 & x_1 & x_2 & \cdots & x_D \end{bmatrix}^{\mathrm{T}} \\ \boldsymbol{b} &= \begin{bmatrix} b_0 & b_1 & b_2 & \cdots & b_D \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{19}$$

令

$$s\left(\boldsymbol{x}\right) = \boldsymbol{b}^{\mathrm{T}}\boldsymbol{x} = b_0 + b_1 x_1 + b_2 x_2 + \cdots b_D x_D \tag{20}$$

(18) 可以记做：

$$f\left(s\right) = \frac{1}{1+\exp\left(-s\right)} \tag{21}$$

(20) 相当于是线性回归，经过如 (21) 逻辑函数映射，得到逻辑回归。图 25 所示为逻辑回归和线性回归之间关系。图 25 这幅图已经让我们看到**神经网络** (neural network) 的一点影子，逻辑函数 $f(s)$ 类似**激活函数** (activation function)。

特别地，对于二元逻辑函数：

$$f\left(x_1,x_2\right)=\frac{1}{1+\exp\left(-\left(b_0+b_1x_1+b_2x_2\right)\right)} \tag{22}$$

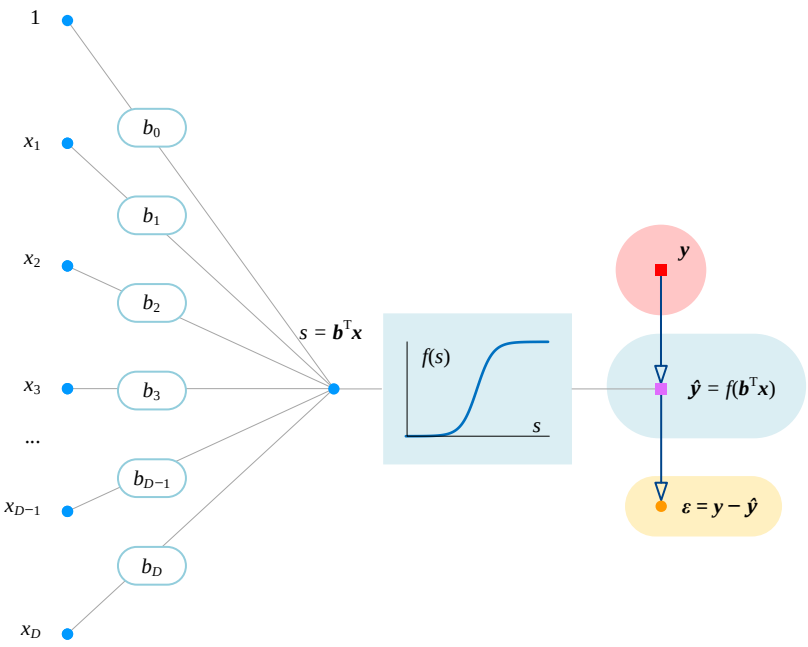


图 25. 逻辑回归和线性回归之间关系

概率视角

形似 (14) 是逻辑分布的 CDF 曲线，对应的表达式：

$$F\left(x|\mu,s\right)=\frac{1}{1+\exp\left(\frac{-\left(x-\mu\right)}{s}\right)}=\frac{1}{2}+\frac{1}{2}\tanh\left(\frac{x-\mu}{2s}\right) \tag{23}$$

其中，μ 为位置参数，s 为形状参数。注意，对于逻辑分布，$s>0$。

逻辑回归可以用来解决二分类，标签为 0 或 1；这是因为逻辑回归可以用来估计事件发生的可能性。

标签为 1 对应的概率为：

$$\Pr\left(y=1|x\right)=\frac{1}{1+\exp\left(-\left(b_0+b_1x\right)\right)} \tag{24}$$

标签为 0 对应的概率为：

$$\Pr\left(y=0|x\right)=1-\Pr\left(y=1|x\right)=\frac{\exp\left(-\left(b_0+b_1x\right)\right)}{1+\exp\left(-\left(b_0+b_1x\right)\right)} \tag{25}$$

图 26 所示为标签为 1 和为 0 的概率关系。

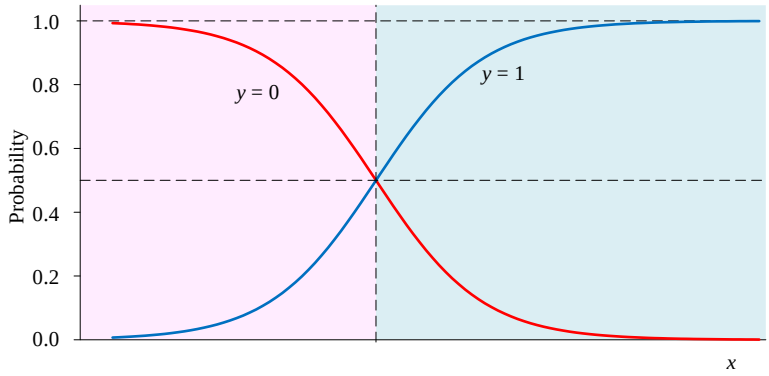


图 26. 标签为 1 和为 0 的概率关系

显然，对于二分类问题，对于任意一点 x，标签为 1 的概率和标签为 0 的概率相加为 1：

$$\mathrm{P}\left(y=0|x\right)+\mathrm{P}\left(y=1|x\right)=1 \tag{26}$$

白话说，某一点要么标签为 1，要么标签为 0，如图 27 所示。

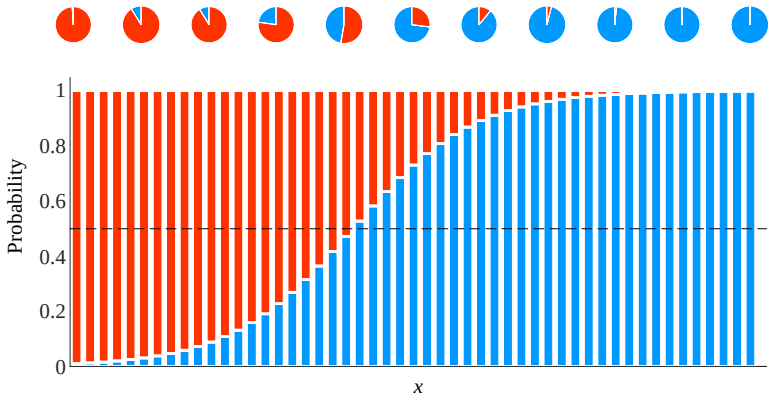


图 27. 逻辑回归模型用于二分类问题

优势率 (odds ratio, OR)，比值比；缩写词为 OR 的对数值：

$$\mathrm{OR}=\text{odds ratio}=\frac{\Pr\left(y=1|x\right)}{\Pr\left(y=0|x\right)}=\frac{1}{\exp\left(-\left(b_0+b_1x\right)\right)} \tag{27}$$

分界 OR = 1，两者概率相同：

$$\frac{1}{\exp\left(-\left(b_0+b_1x\right)\right)}=1 \tag{28}$$

整理得到：

$$b_0+b_1x=0 \tag{29}$$

即

$$x=-\frac{b_0}{b_1} \tag{30}$$

本章后文介绍如何用 sklearn 中逻辑回归函数解决三分类问题。

13.6 逻辑函数完成分类问题

单特征

本节介绍用 sklearn.linear_model.LogisticRegression() 逻辑回归模型，根据鸢尾花花萼长度这一单一特征数据进行分类。

图 28 所示为鸢尾花花萼长度数据和真实三分类 y 之间关系。

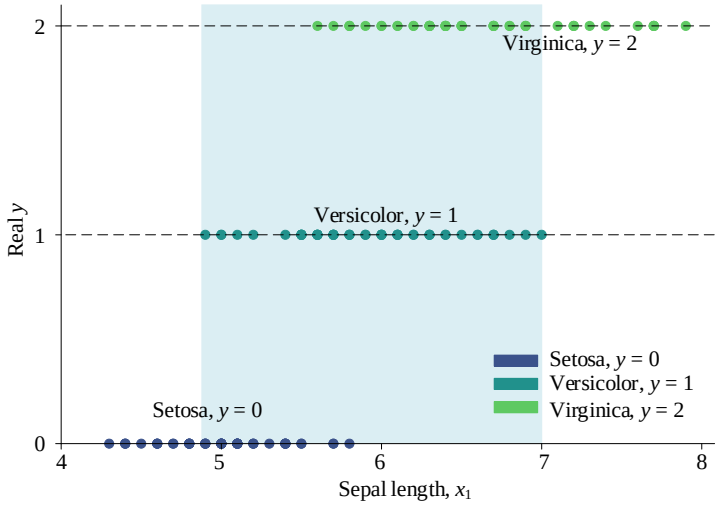


图 28. 鸢尾花花萼长度和真实分类之间关系

图 29 所示为鸢尾花花萼长度数据分类概率密度估计。这幅图实际上已经能够透露出比较合适的分类区间。

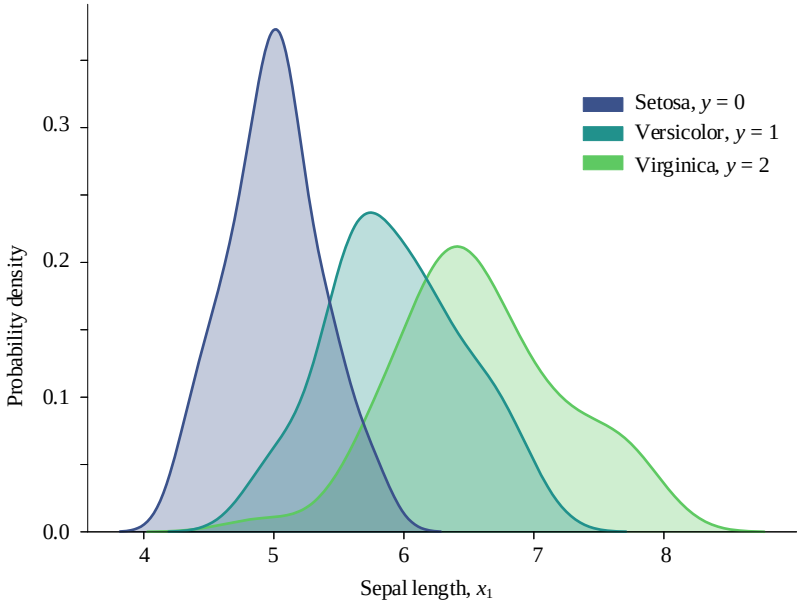


图 29. 鸢尾花花萼长度数据分类概率密度估计

sklearn.linear_model.LogisticRegression() 模型结果可以输出各个分类的概率，得到的图像如图 30 所示。比较三个类别的概率，可以进行分类预测。

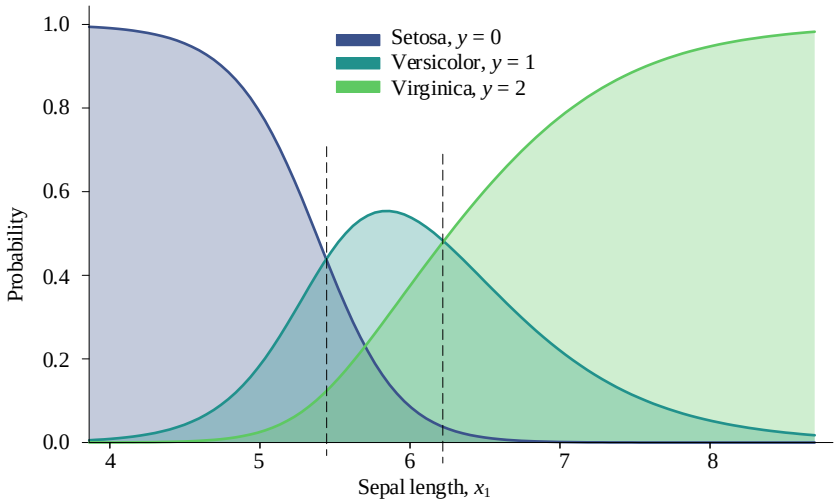


图 30. 逻辑回归估算得到的分类概率

图 31 所示为鸢尾花分类预测结果。

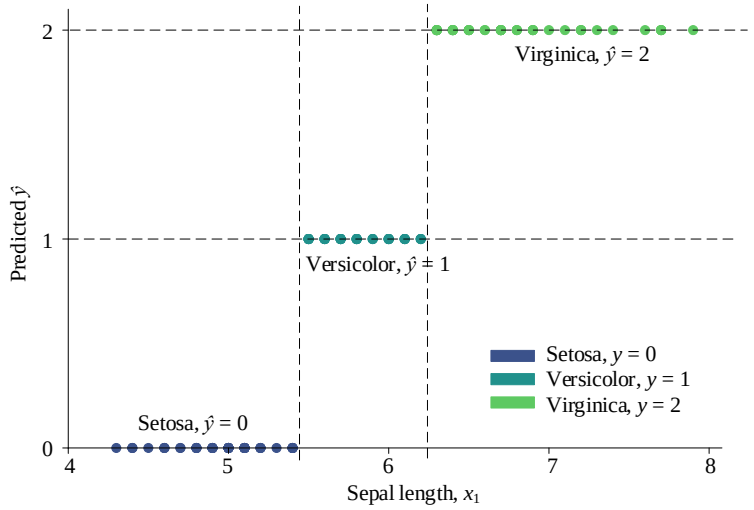


图 31. 鸢尾花花萼长度和预测分类之间关系

Bk6_Ch13_04.py 绘制本节图像。

双特征

本节介绍用 sklearn.linear_model.LogisticRegression() 逻辑回归模型，根据鸢尾花花萼长度和花萼宽度这两个特征数据进行分类。

图 32 所示为鸢尾花花萼长度和花萼宽度两个特征数据散点图，和分类边际分布概率密度估计曲线。

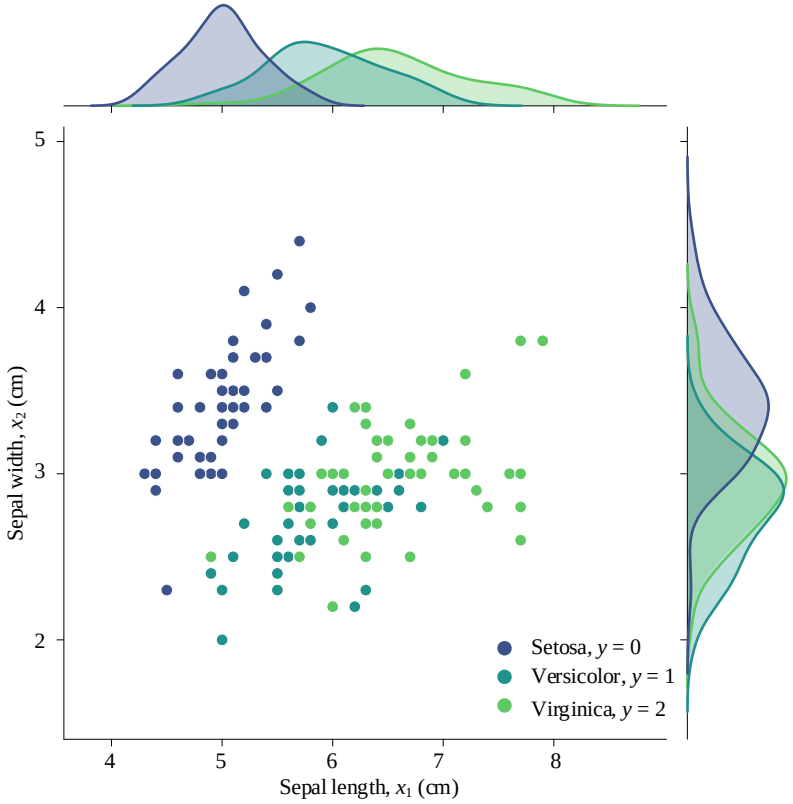


图 32. 鸢尾花双特征数据和分类边际分布

图 33 ~ 图 35 三幅图分别给出鸢尾花双特征分类概率预测曲面。比较三个曲面高度可以得到分类决策边界。在分类问题中，决策边界 (decision boundary) 指的是将不同类别样本分开的平面或曲面。

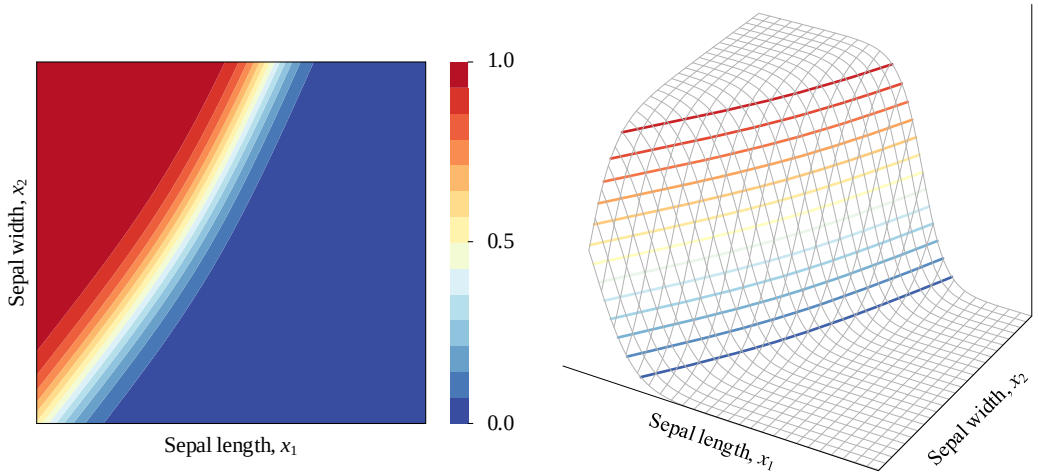


图 33. 鸢尾花双特征分类预测，$\hat{y} = 0$

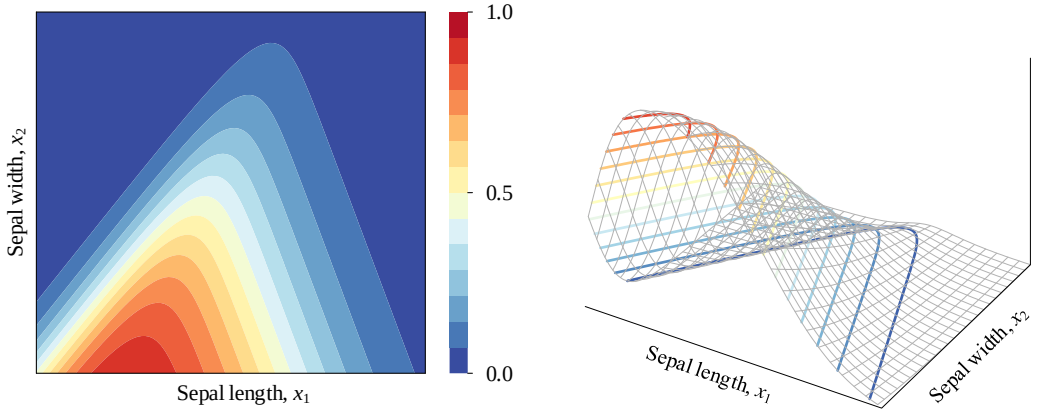


图 34. 鸢尾花双特征分类预测，$\hat{y} = 1$

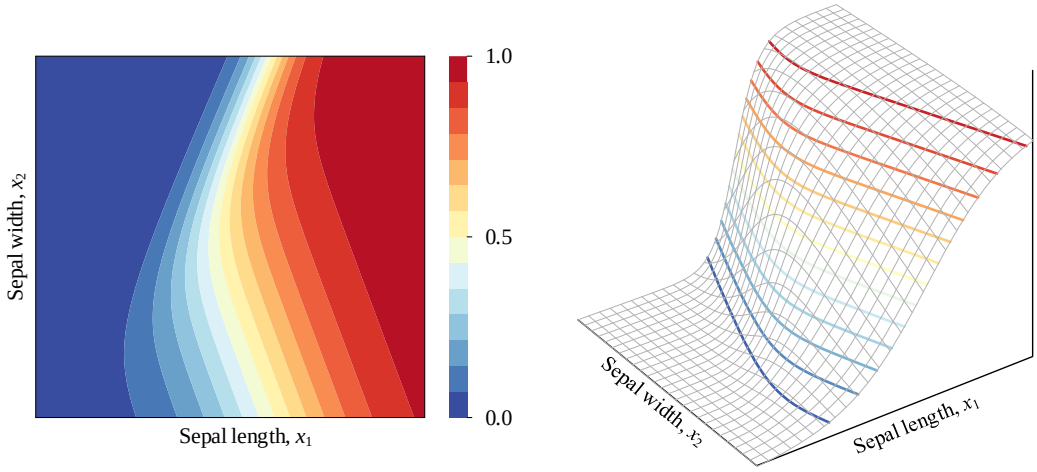


图 35. 鸢尾花双特征分类预测，$\hat{y} = 2$

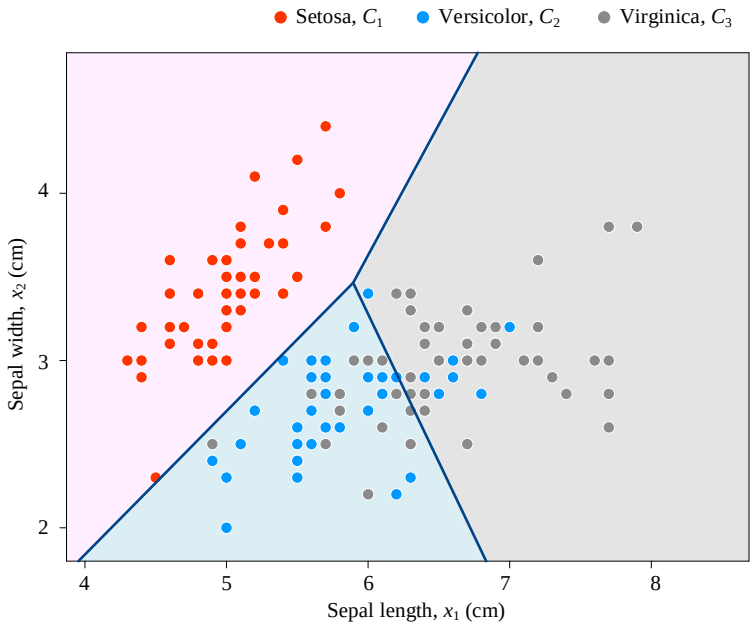
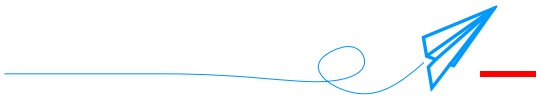


图 36. 利用逻辑回归得到的分类决策边界

Bk6_Ch13_05.py 绘制本节图像。



非线性回归是一种用于建模非线性关系的统计方法。在非线性回归中，因变量和自变量之间的关系不是线性的，而是可以通过非线性函数来描述。

需要非线性回归的原因是许多自然现象和实际问题都不是线性的，例如，随着时间的推移，人口增长率和经济增长率并不是线性的，这就需要非线性回归模型。

常见的非线性回归方法包括多项式回归、指数回归、对数回归、幂函数回归、逻辑回归、等等。每种方法都有其优缺点，例如多项式回归可以拟合大部分的非线性关系，但容易出现过拟合。

逻辑回归将自变量和因变量之间的关系建模为一种逻辑函数，如 sigmoid 函数。从概率视角来看，逻辑回归可以将输出解释为给定输入的条件下，观察到给定类别的概率。它将自变量映射到一个概率值，该值介于 0 和 1 之间，并使用这个概率来预测分类结果。

欢迎读者阅读 *An Introduction to Statistical Learning: With Applications in R* 一书第七章，图书下载地址。

https://www.statlearning.com/

15 Principal Component Analysis
主成分分析

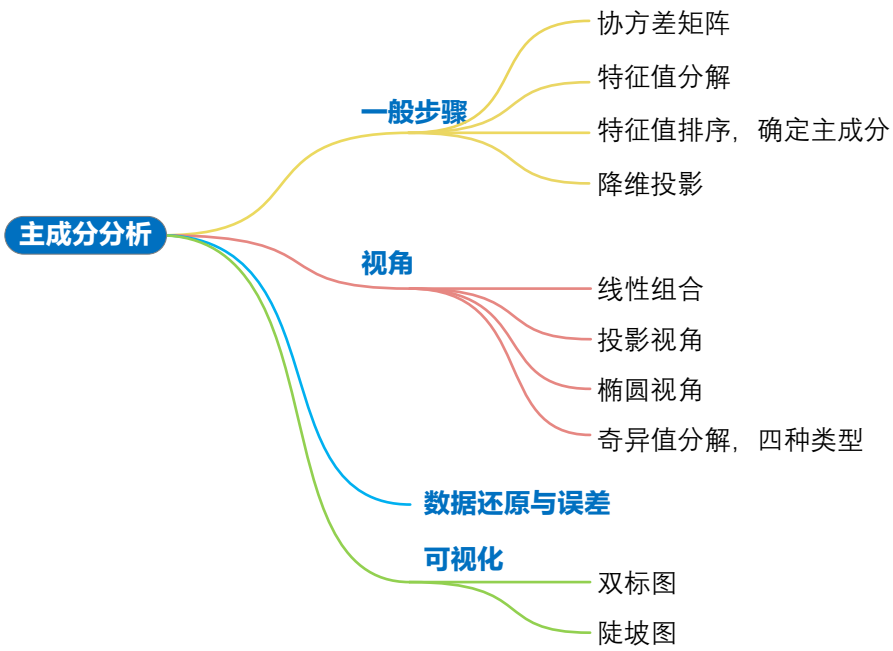
处理多维数据，通过降维发现数据隐藏规律

忽视数学会损害所有知识，因为不了解数学的人无法了解世界上的其他科学或事物。更糟糕的是，那些无知的人无法感知自己的无知，因此不寻求补救。

Neglect of mathematics work injury to all knowledge, since he who is ignorant of it cannot know the other sciences or things of this world. And what is worst, those who are thus ignorant are unable to perceive their own ignorance, and so do not seek a remedy.

—— 罗吉尔·培根 (Roger Bacon) | 英国哲学家 | 1214 ~ 1294

- `numpy.corrcoef()` 计算相关性系数矩阵
- `numpy.cov()` 计算协方差矩阵
- `numpy.linalg.eig()` 特征值分解
- `numpy.linalg.svd()` 奇异值分解
- `numpy.mean()` 计算均值
- `numpy.random.multivariate_normal()` 产生多元正态分布随机数
- `numpy.std()` 计算均方差
- `numpy.var()` 计算方差
- `numpy.zeros_like()` 产生形如输入矩阵的全 0 矩阵
- `seaborn.heatmap()` 绘制热图
- `seaborn.jointplot()` 绘制联合分布和边际分布
- `seaborn.kdeplot()` 绘制 KDE 核概率密度估计曲线
- `seaborn.lineplot()` 绘制线图
- `seaborn.pairplot()` 绘制成对分析图
- `sklearn.decomposition.PCA()` 主成分分析函数
- `yellowbrick.features.PCA()` 绘制 PCA 双标图



15.1 原始数据

主成分分析

主成分分析 (principal component analysis, PCA) 最初由**卡尔·皮尔逊** (Karl Pearson) 在 1901 提出。主成分分析是数据降维的重要方法之一。通过线性变换，主成分分析将原始多维数据投影到一个新的正交坐标系，将原始数据中的最大方差成分提取出来。

卡尔·皮尔逊 (Karl Pearson)
英国数学家 | 1857 ~ 1936
常被誉为现代统计科学的创立者；丛书关键词：● 相关性系数 ● 线性回归 ● 主成分分析

举个例子，主成分分析实际上寻找数据在主元空间内投影。图 1 所示杯子，它是一个 3D 物体，在一张图展示杯子，而且尽可能多地展示杯子细节，就需要从空间多个角度观察杯子并找到合适角度。这个过程实际上是将三维数据投影到二维平面过程。这也是一个降维过程，即从三维变成二维。图 2 展示杯子六个平面上投影结果。

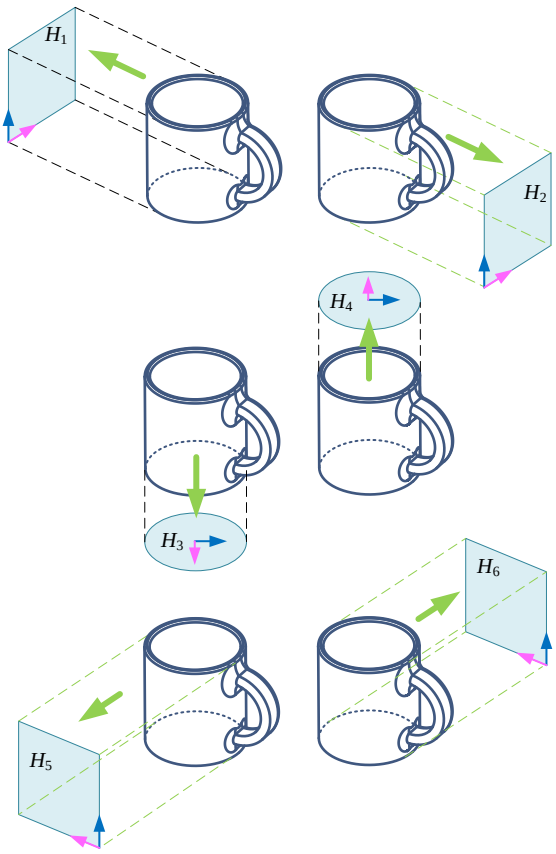


图 1. 咖啡杯六个投影方向

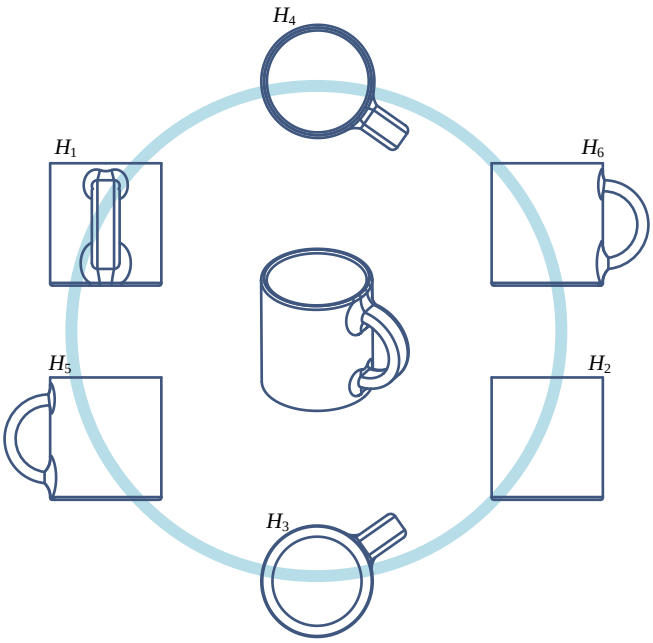


图 2. 咖啡杯在六个方向投影图像

以鸢尾花数据为例

本章以鸢尾花数据为例介绍如何利用主成分分析处理数据。图 3 所示为鸢尾花原始数据矩阵 $\boldsymbol{X}$ 构成的热图。数据矩阵 $\boldsymbol{X}$ 有 150 个数据点，即 150 行；矩阵 $\boldsymbol{X}$ 有 4 个特征，即 4 列。

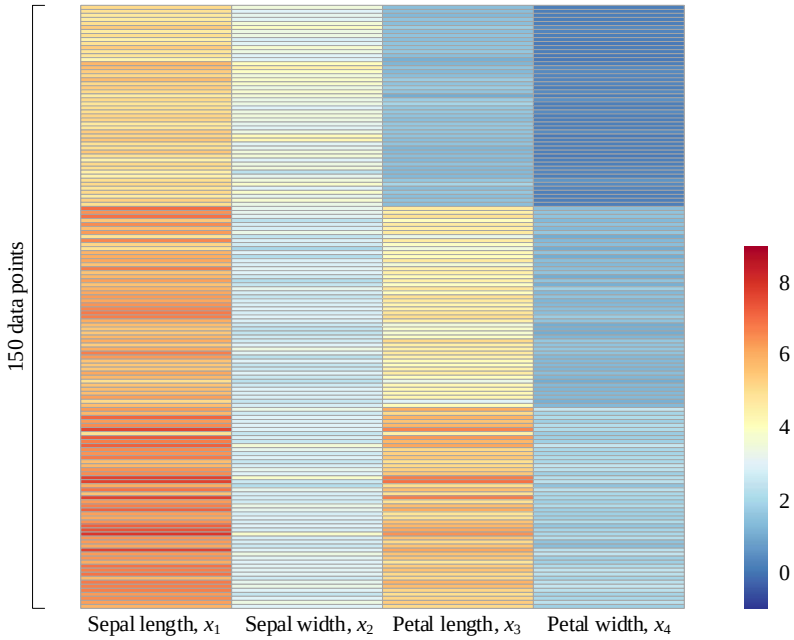


图 3. 鸢尾花数据，原始数据矩阵 $\boldsymbol{X}$

对原始数据进行统计分析。首先以行向量表达数据矩阵 $\boldsymbol{X}$ 质心：

$$\boldsymbol{\mu}_{\boldsymbol{X}} = \begin{bmatrix} \underset{\text{Sepal length, } x_1}{5.843} & \underset{\text{Sepal width, } x_2}{3.057} & \underset{\text{Petal length, } x_3}{3.758} & \underset{\text{Petal width, } x_4}{1.199} \end{bmatrix} \tag{1}$$

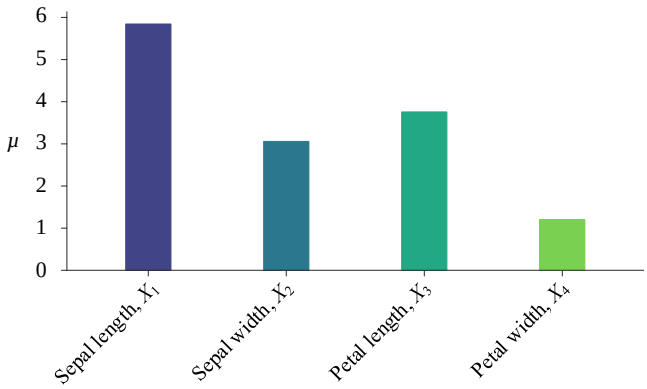


图 4. 鸢尾花数据四个特征上均值

然后，计算 $\boldsymbol{X}$ 每一列均方差，以行向量表达：

$$\boldsymbol{\sigma}_{\boldsymbol{X}} = \begin{bmatrix} \underset{\text{Sepal length, } x_1}{0.825} & \underset{\text{Sepal width, } x_2}{0.434} & \underset{\text{Petal length, } x_3}{1.759} & \underset{\text{Petal width, } x_4}{0.759} \end{bmatrix} \tag{2}$$

$\boldsymbol{X}$ 第三个特征，也就是花瓣长度 x_3 对应的均方差最大。图 5 所示为 KDE 估计得到的鸢尾花四个特征分布图。

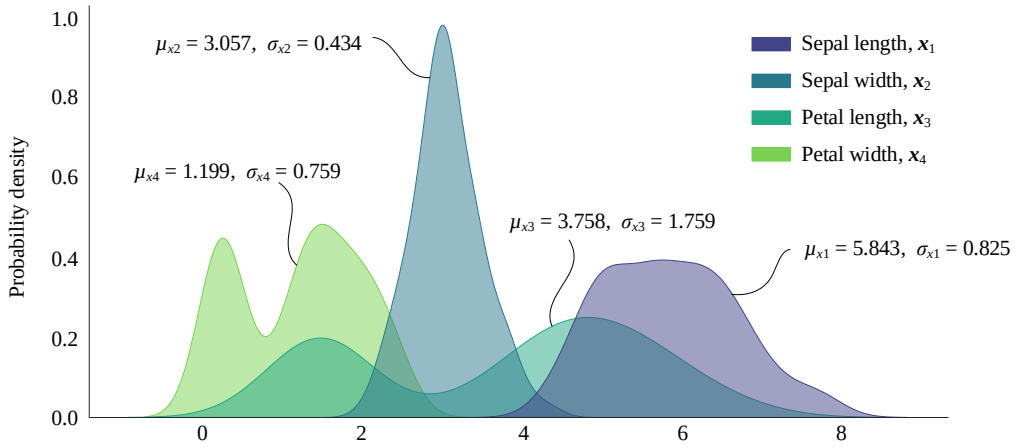


图 5. 鸢尾花数据四个特征上分布，KDE 估计

利用 seaborn.pairplot() 函数可以绘制如图 6 所示成对特征分析图；成对特征分析图方便展示每一对数据特征之间的关系，而对角线图像则展示每一个特征单独的统计规律。

由于鸢尾花数据存在三个分类，所以可以利用 seaborn.pairplot() 函数展示具有分类特征的成对分析图，具体如图 7 所示。图 7 这幅图让我们看到了每一类别数据特征之间和自身的分布规律。

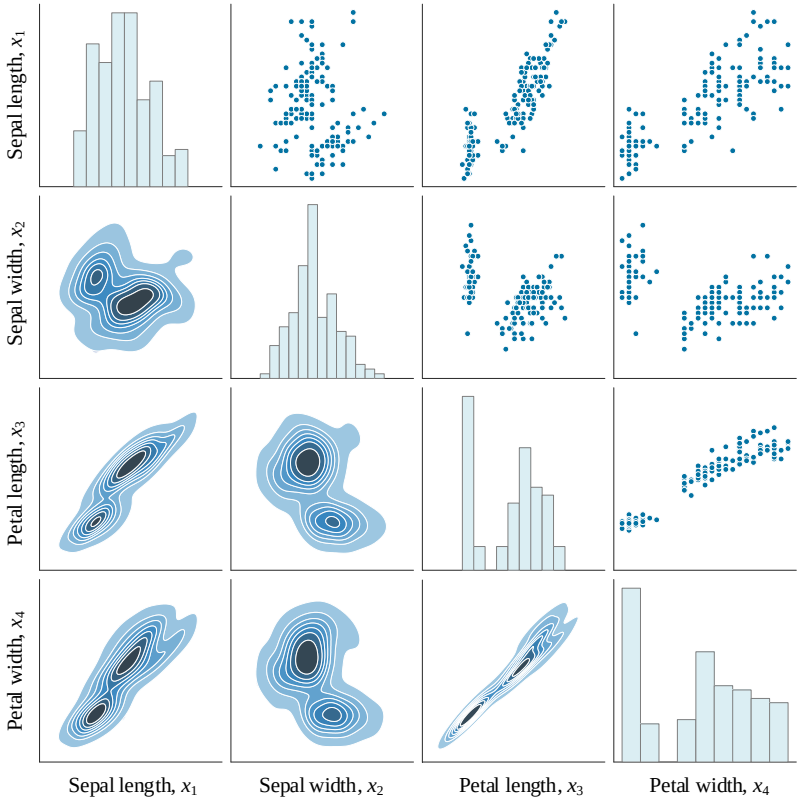


图 6. 鸢尾花数据成对特征分析图，不分类

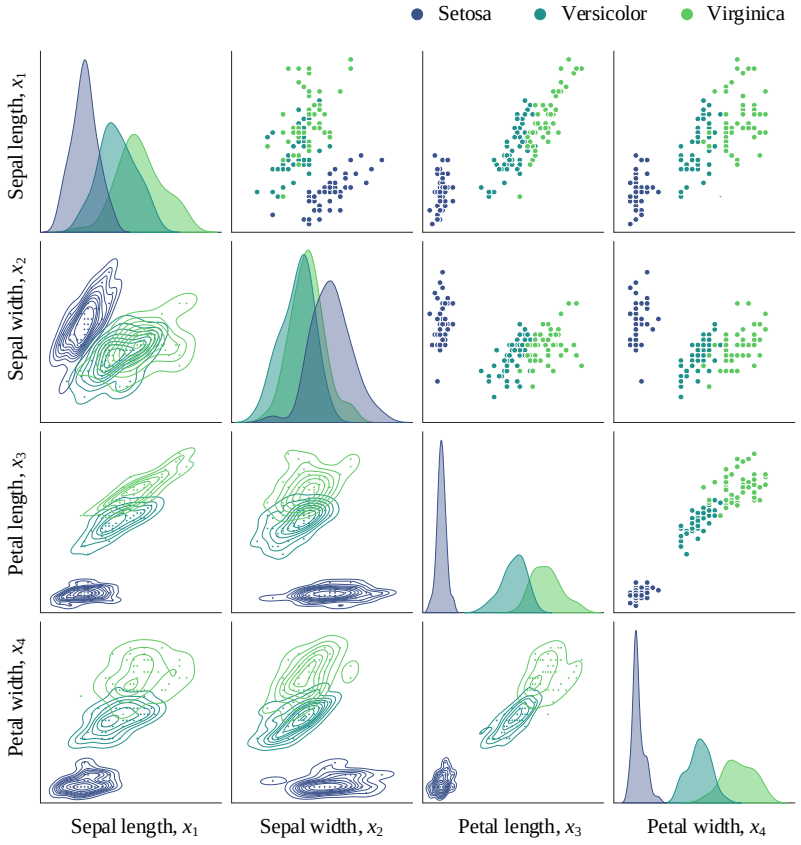


图 7. 鸢尾花数据成对特征分析图，分类

计算数据矩阵 $\boldsymbol{X}$ 协方差矩阵 $\boldsymbol{\Sigma}$:

$$\boldsymbol{\Sigma} = \begin{bmatrix} 0.686 & -0.042 & 1.274 & 0.516 \\ -0.042 & 0.190 & -0.330 & -0.122 \\ 1.274 & -0.330 & 3.116 & 1.296 \\ 0.516 & \underbrace{-0.122}_{\text{Sepal width, } x_2} & 1.296 & 0.581 \end{bmatrix} \begin{matrix} \leftarrow \text{Sepal length, } x_1 \\ \leftarrow \text{Sepal width, } x_2 \\ \leftarrow \text{Petal length, } x_3 \\ \leftarrow \text{Petal width, } x_4 \end{matrix} \tag{3}$$

接下来，协方差矩阵 $\boldsymbol{\Sigma}$ 将用于特征值分解。

在 PCA 中，有时候会对数据进行标准化是因为不同特征的单位和尺度不同，可能会对 PCA 的结果产生影响。如果不进行标准化处理，那么在协方差矩阵的计算过程中，某些特征的方差较大，将会对 PCA 的结果产生更大的影响，而这些特征不一定是最重要的。因此，为了消除这种影响，我们需要对数据进行标准化处理。

标准化的目的是将不同特征的值域缩放到相同的范围，使得所有特征的平均值为 0，标准差为 1，从而消除不同特征间的单位和尺度差异，使得所有特征具有相同的重要性。原始数据标准化的结果是 Z 分数。Z 分数的协方差矩阵实际上是原始数据的相关性系数矩阵。

总结来说，在进行 PCA 之前，如果数据中的特征具有不同的度量单位，或者特征值的范围变化很大，那么就应该考虑进行标准化。标准化可以使得 PCA 的结果更加准确和可靠，避免某些特征在主成分分析中被过度强调或者忽略。但是需要注意的是，有些情况下，标准化并不适用于所有数据集，例如当数据中的特征已经被精心设计或处理过时，标准化可能会使得信息损失或降低 PCA 的效果。

计算数据矩阵 $\boldsymbol{X}$ 相关性系数矩阵 $\boldsymbol{P}$:

$$\boldsymbol{P} = \begin{bmatrix} 1.000 & -0.118 & 0.872 & 0.818 \\ -0.118 & 1.000 & -0.428 & -0.366 \\ 0.872 & -0.428 & 1.000 & 0.963 \\ 0.818 & \underbrace{-0.366}_{\text{Sepal width, } x_2} & 0.963 & 1.000 \end{bmatrix} \begin{matrix} \leftarrow \text{Sepal length, } x_1 \\ \leftarrow \text{Sepal width, } x_2 \\ \leftarrow \text{Petal length, } x_3 \\ \leftarrow \text{Petal width, } x_4 \end{matrix} \tag{4}$$

观察相关性系数矩阵 $\boldsymbol{P}$，可以发现花萼长度 x_1 和花萼宽度 x_2 线性负相关，花瓣长度 x_3 和花萼宽度 x_2 线性负相关，花瓣宽度 x_4 和花萼宽度 x_2 线性负相关。

15.2 特征值分解

对 $\boldsymbol{\Sigma}$ 特征值分解得到：

$$\boldsymbol{\Sigma} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1} \tag{5}$$

其中，$\boldsymbol{V}$ 是正交矩阵，满足 $\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{I}$。实际上 $\boldsymbol{\Sigma}$ 为对称矩阵，因此上式为谱分解，即 $\boldsymbol{\Sigma} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}}$。

特征值矩阵 $\boldsymbol{\Lambda}$ 为：

$$\boldsymbol{\Lambda} = \begin{bmatrix} 4.228 & & & \\ & 0.242 & & \\ & & 0.078 & \\ & & & 0.023 \end{bmatrix} \tag{6}$$

特征向量构成的矩阵 $\boldsymbol{V}$ 为：

$$\begin{aligned} \boldsymbol{V} &= \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \boldsymbol{v}_3 & \boldsymbol{v}_4 \end{bmatrix} \\ &= \begin{bmatrix} v_{1,1} & v_{1,2} & v_{1,3} & v_{1,4} \\ v_{2,1} & v_{2,2} & v_{2,3} & v_{2,4} \\ v_{3,1} & v_{3,2} & v_{3,3} & v_{3,4} \\ v_{4,1} & v_{4,2} & v_{4,3} & v_{4,4} \end{bmatrix} \begin{matrix} \leftarrow \text{Sepal length, } x_1 \\ \leftarrow \text{Sepal width, } x_2 \\ \leftarrow \text{Petal length, } x_3 \\ \leftarrow \text{Petal width, } x_4 \end{matrix} = \begin{bmatrix} 0.361 & 0.656 & -0.582 & -0.315 \\ -0.084 & 0.730 & 0.597 & 0.319 \\ 0.856 & -0.173 & 0.076 & 0.479 \\ \underbrace{0.358}_{\text{PC1, } \boldsymbol{v}_1} & \underbrace{-0.075}_{\text{PC2, } \boldsymbol{v}_2} & \underbrace{0.545}_{\text{PC3, } \boldsymbol{v}_3} & \underbrace{-0.753}_{\text{PC4, } \boldsymbol{v}_4} \end{bmatrix} \end{aligned} \tag{7}$$

矩阵 $\boldsymbol{V}$ 每一列代表一个主成分，该主成分中每一个元素相当于原始数据特征的系数。图 8 所示为不同主成分的系数线图。

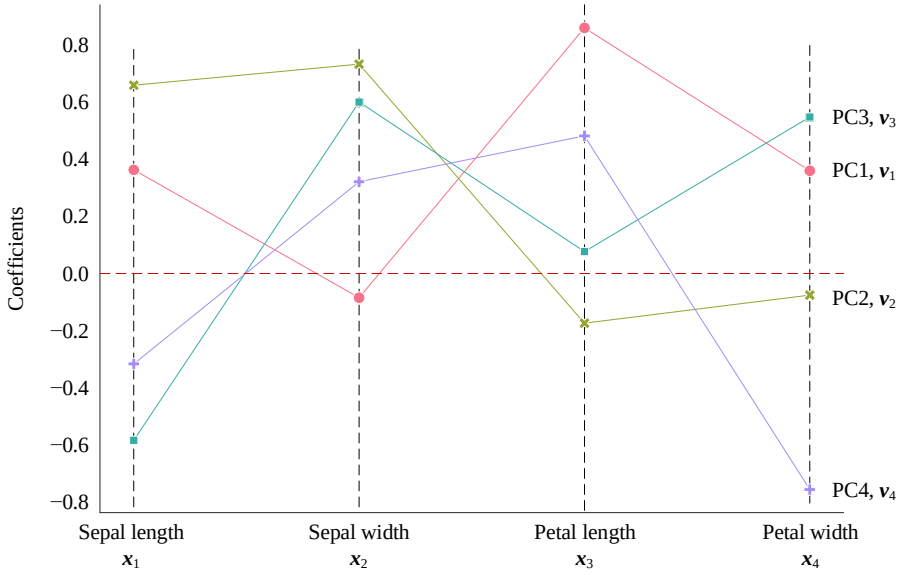


图 8. $\boldsymbol{V}$ 系数线图

如图 9 所示，$\boldsymbol{V}$ 和自己转置 $\boldsymbol{V}^{\mathrm{T}}$ 乘积为单位阵 $\boldsymbol{I}$，即：

$$\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V} = \boldsymbol{I} \tag{8}$$

展开上式得到：

$$
\begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \boldsymbol{v}_3 & \boldsymbol{v}_4 \end{bmatrix}^{T} \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \boldsymbol{v}_3 & \boldsymbol{v}_4 \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_1^{T} \\ \boldsymbol{v}_2^{T} \\ \boldsymbol{v}_3^{T} \\ \boldsymbol{v}_4^{T} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \boldsymbol{v}_3 & \boldsymbol{v}_4 \end{bmatrix}
$$

$$
= \begin{bmatrix} \boldsymbol{v}_1^{T}\boldsymbol{v}_1 & \boldsymbol{v}_1^{T}\boldsymbol{v}_2 & \boldsymbol{v}_1^{T}\boldsymbol{v}_3 & \boldsymbol{v}_1^{T}\boldsymbol{v}_4 \\ \boldsymbol{v}_2^{T}\boldsymbol{v}_1 & \boldsymbol{v}_2^{T}\boldsymbol{v}_2 & \boldsymbol{v}_2^{T}\boldsymbol{v}_3 & \boldsymbol{v}_2^{T}\boldsymbol{v}_4 \\ \boldsymbol{v}_3^{T}\boldsymbol{v}_1 & \boldsymbol{v}_3^{T}\boldsymbol{v}_2 & \boldsymbol{v}_3^{T}\boldsymbol{v}_3 & \boldsymbol{v}_3^{T}\boldsymbol{v}_4 \\ \boldsymbol{v}_4^{T}\boldsymbol{v}_1 & \boldsymbol{v}_4^{T}\boldsymbol{v}_2 & \boldsymbol{v}_4^{T}\boldsymbol{v}_3 & \boldsymbol{v}_4^{T}\boldsymbol{v}_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \boldsymbol{I} \tag{9}
$$

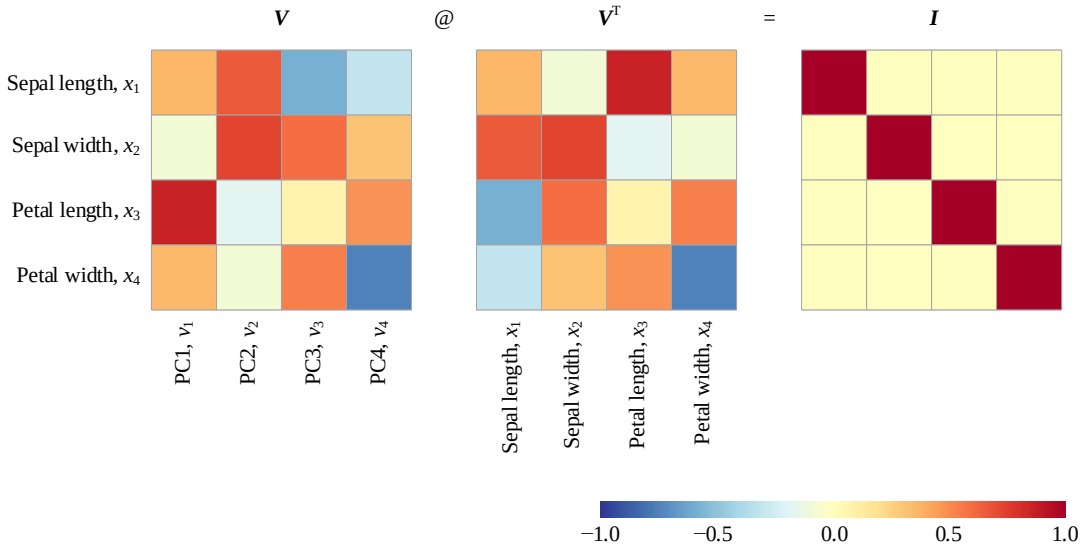


图 9. 特征矩阵 $\boldsymbol{V}$ 和自身转置的乘积为单位矩阵 $\boldsymbol{I}$

如果对鸢尾花数据先进行标准化处理，即使用每一列变成 Z 分数；再计算得到的矩阵 $\boldsymbol{V}$ 则为：

$$
\boldsymbol{V} = \begin{bmatrix} 0.521 & 0.377 & 0.720 & -0.261 \\ -0.269 & 0.923 & -0.244 & 0.124 \\ 0.580 & 0.024 & -0.142 & 0.801 \\ 0.565 & 0.067 & \underbrace{-0.634}_{\text{PC3, } \boldsymbol{v}_3} & \underbrace{-0.524}_{\text{PC4, } \boldsymbol{v}_4} \\ \text{PC1, } \boldsymbol{v}_1 & \text{PC2, } \boldsymbol{v}_2 & & \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \end{matrix} \begin{matrix} \leftarrow \text{Sepal length, } x_1 \\ \leftarrow \text{Sepal width, } x_2 \\ \leftarrow \text{Petal length, } x_3 \\ \leftarrow \text{Petal width, } x_4 \end{matrix} \tag{10}
$$

可以发现 (7) 和 (10) 明显不同，下一章将对比这两种技术路线。

15.3 正交空间

矩阵 $\boldsymbol{V}$ 有 D 个列向量，对应 D 个正交基，如下：

$$\boldsymbol{V}=\begin{bmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,D-1} & v_{1,D} \\ v_{2,1} & v_{2,2} & \cdots & v_{2,D-1} & v_{2,D} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ v_{D-1,1} & v_{D-1,2} & \cdots & v_{D-1,D-1} & v_{D-1,D} \\ v_{D,1} & v_{D,2} & \cdots & v_{D,D-1} & v_{D,D} \end{bmatrix}=\begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \ldots & \boldsymbol{v}_{D-1} & \boldsymbol{v}_D \end{bmatrix} \tag{11}$$

任意列向量 $\boldsymbol{v}_i$ 每一个元素都包含 $\boldsymbol{X}$ 列向量 $[\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_D]$ 成分，即列向量 $\boldsymbol{v}_i$ 为 $[\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_D]$ 线性组合。

$$\begin{aligned} \boldsymbol{v}_1 &= v_{1,1}\boldsymbol{x}_1 + v_{2,1}\boldsymbol{x}_2 + \ldots + v_{D-1,1}\boldsymbol{x}_{D-1} + v_{D,1}\boldsymbol{x}_D \\ \boldsymbol{v}_2 &= v_{1,2}\boldsymbol{x}_1 + v_{2,2}\boldsymbol{x}_2 + \ldots + v_{D-1,2}\boldsymbol{x}_{D-1} + v_{D,2}\boldsymbol{x}_D \\ &\ldots \\ \boldsymbol{v}_D &= v_{1,D}\boldsymbol{x}_1 + v_{2,D}\boldsymbol{x}_2 + \ldots + v_{D-1,D}\boldsymbol{x}_{D-1} + v_{D,D}\boldsymbol{x}_D \end{aligned} \tag{12}$$

图 10 所示为线性组合构造正交空间 $[\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_D]$。注意，$[\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_D]$ 类似于 $[\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_D]$，它们代表方向向量，而不是具体的数据。

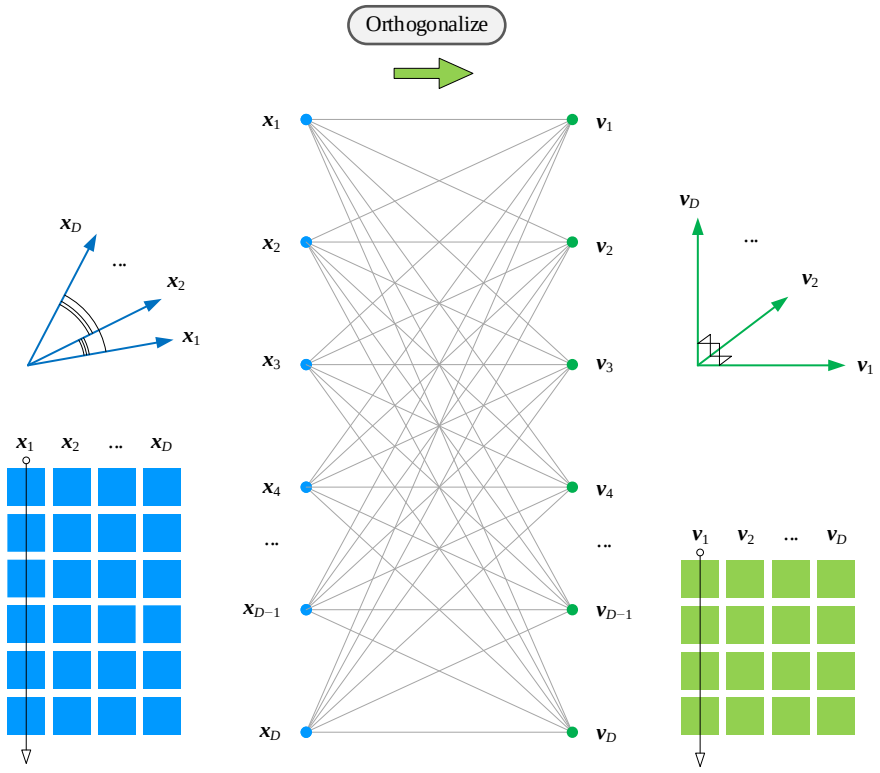


图 10. 线性组合构造正交空间 $[\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_D]$

如图 11 所示，以 $\boldsymbol{v}_1$ 为例，第一主成分方向上，$\boldsymbol{v}_1$ 等价于由 $v_{1,1}$ 比例 $\boldsymbol{x}_1$，$v_{2,1}$ 比例 $\boldsymbol{x}_2$，$v_{3,1}$ 比例 $\boldsymbol{x}_3$...以及 $v_{D,1}$ 比例 $\boldsymbol{x}_D$ 线性组合构造。从另外一个角度，$[\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_D]$ 在向量 $\boldsymbol{v}_1$ 上标量投影值分别为 $v_{1,1}, v_{2,1}, ..., v_{D,1}$。图 12 所示为鸢尾花数据主成分分析第一主成分 $\boldsymbol{v}_1$ 的构造情况。

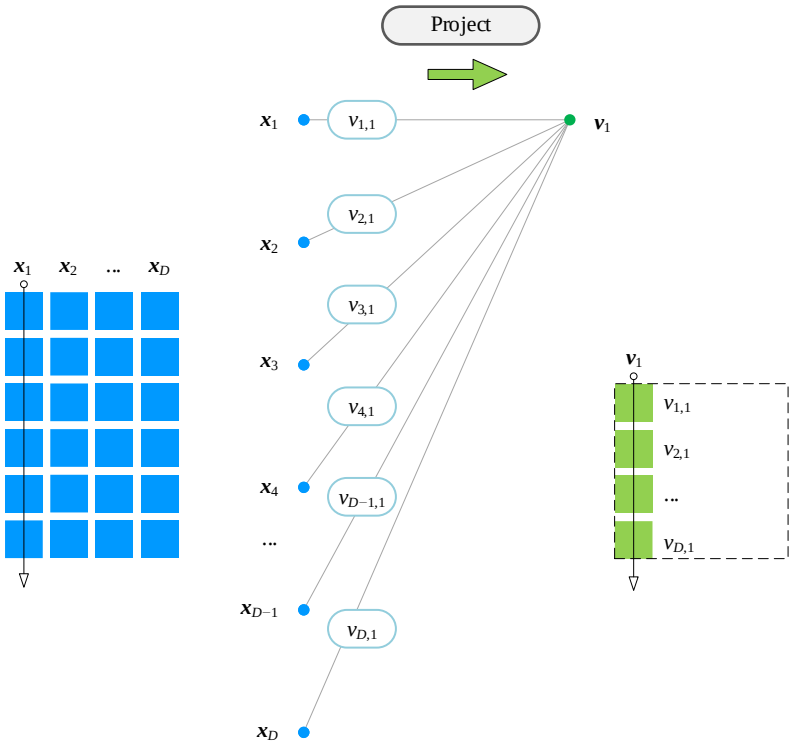


图 11. 构造第一主成分 $\boldsymbol{v}_1$

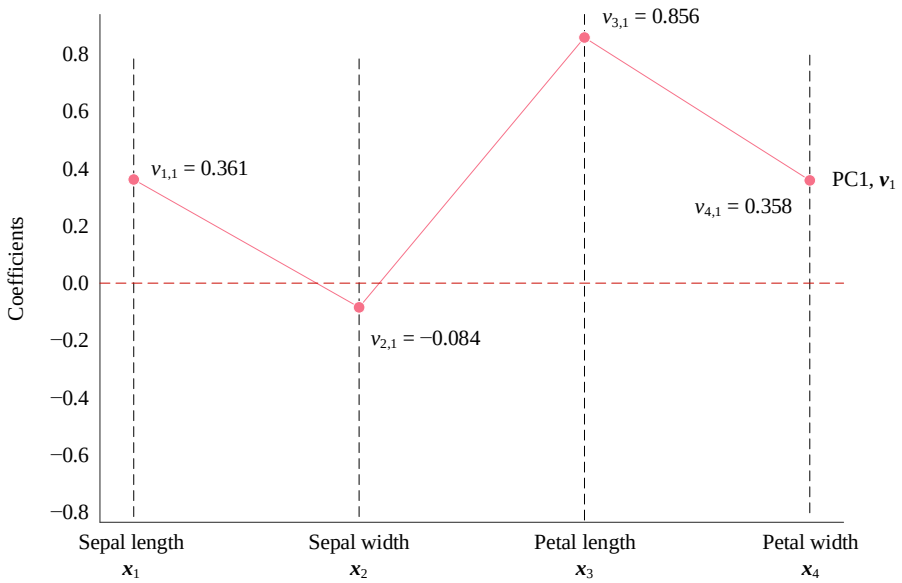


图 12. 构造第一主成分 $\boldsymbol{v}_1$，鸢尾花数据

如图 13 所示，第二主成分 $\boldsymbol{v}_2$ 方向上，$\boldsymbol{v}_2$ 等价于由 $v_{1,2}$ 比例 $\boldsymbol{x}_1$，$v_{2,2}$ 比例 $\boldsymbol{x}_2$，$v_{3,2}$ 比例 $\boldsymbol{x}_3$...以及 $v_{D,2}$ 比例 $\boldsymbol{x}_D$ 线性构造。图 14 所示为鸢尾花数据主成分分析第二主成分 $\boldsymbol{v}_2$ 的构造情况。

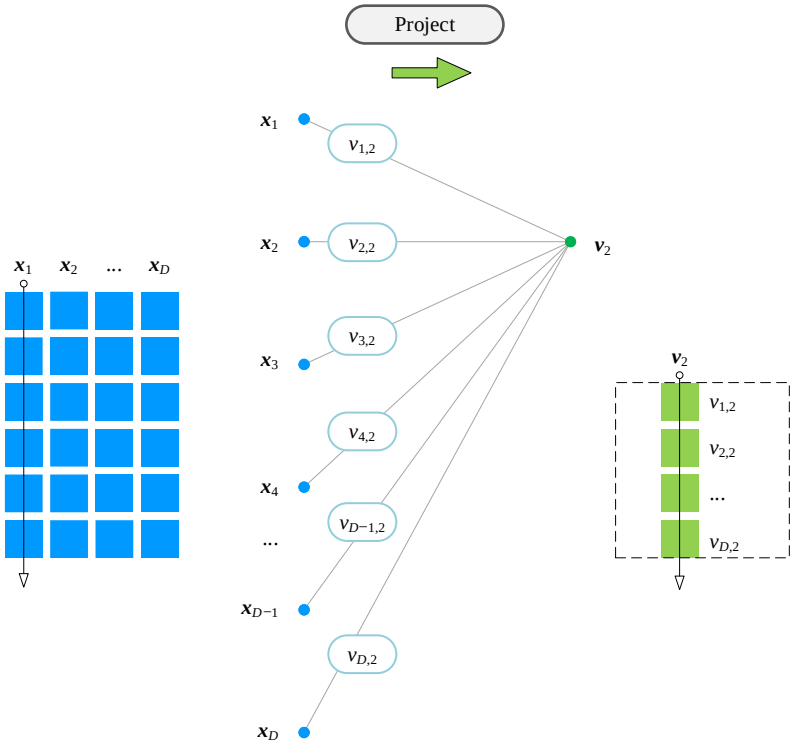


图 13. 构造第二主成分 $\boldsymbol{v}_2$

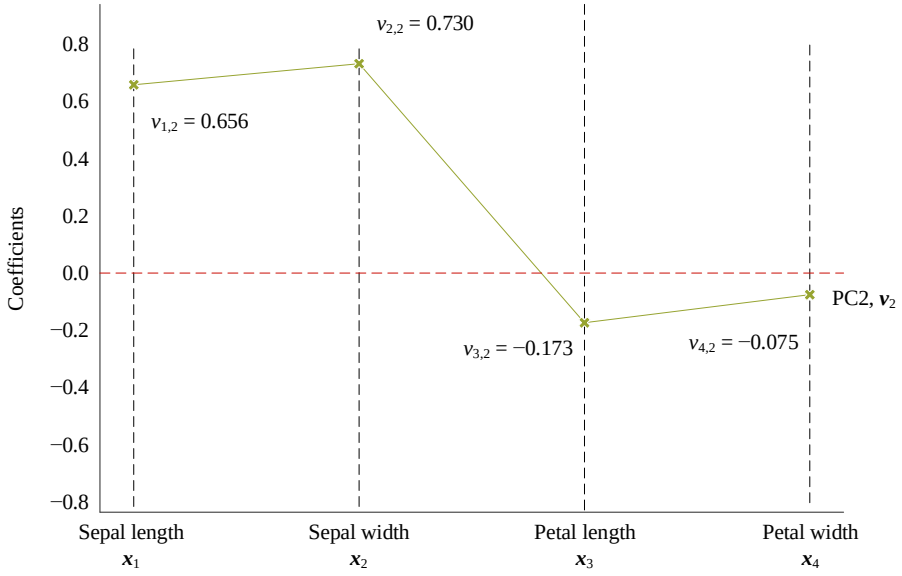


图 14. 构造第二主成分 $\boldsymbol{v}_2$，鸢尾花数据

如图 15 所示，第三主成分 $\boldsymbol{v}_3$ 方向上，$\boldsymbol{v}_3$ 等价于由 $v_{1,3}$ 比例 $\boldsymbol{x}_1$，$v_{2,3}$ 比例 $\boldsymbol{x}_2$，$v_{3,3}$ 比例 $\boldsymbol{x}_3$...以及 $v_{D,3}$ 比例 $\boldsymbol{x}_D$ 线性构造。图 16 所示为鸢尾花数据主成分分析第三主成分 $\boldsymbol{v}_3$ 的构造情况。

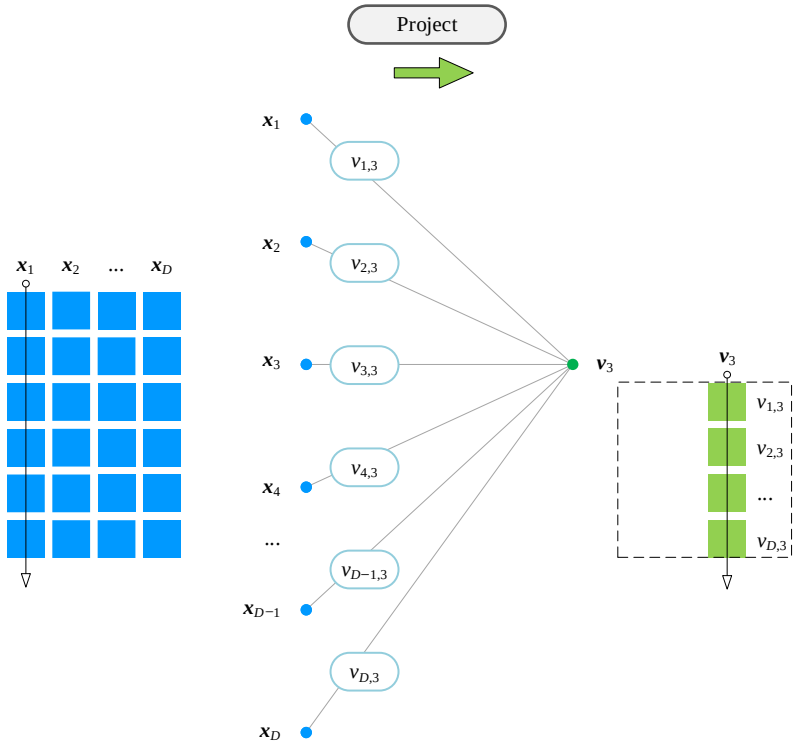


图 15. 构造第三主成分 $\boldsymbol{v}_3$

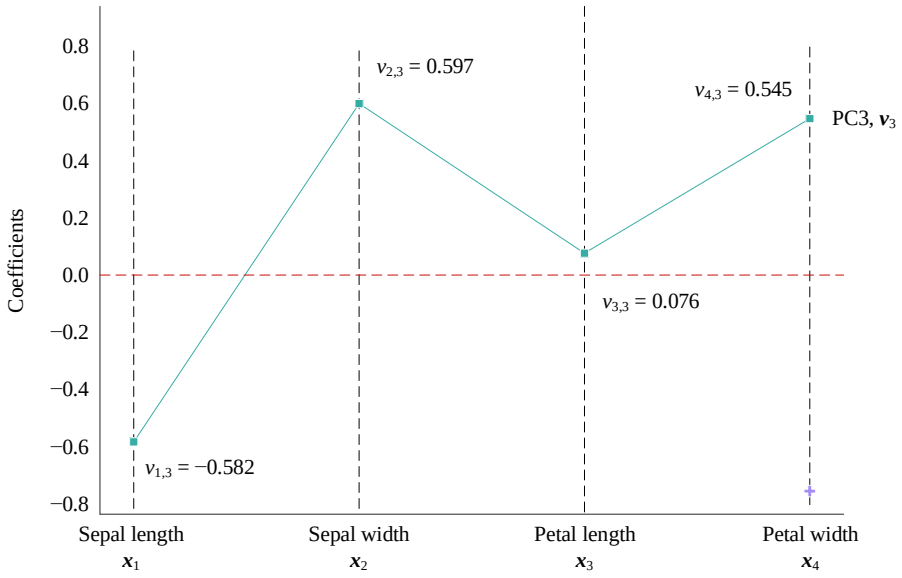


图 16. 构造第三主成分 $\boldsymbol{v}_3$，鸢尾花数据

15.4 投影结果

图 17 所示为投影后得到的新特征数据矩阵 $\boldsymbol{Z}$。这幅热图，蓝色色系数据接近 0，红色色系数据接近 8；可以发现矩阵 $\boldsymbol{Z}$ 四个新特征 (z_1, z_2, z_3 和 z_4) 从左到右颜色差异逐渐减小，即方差不断减小。

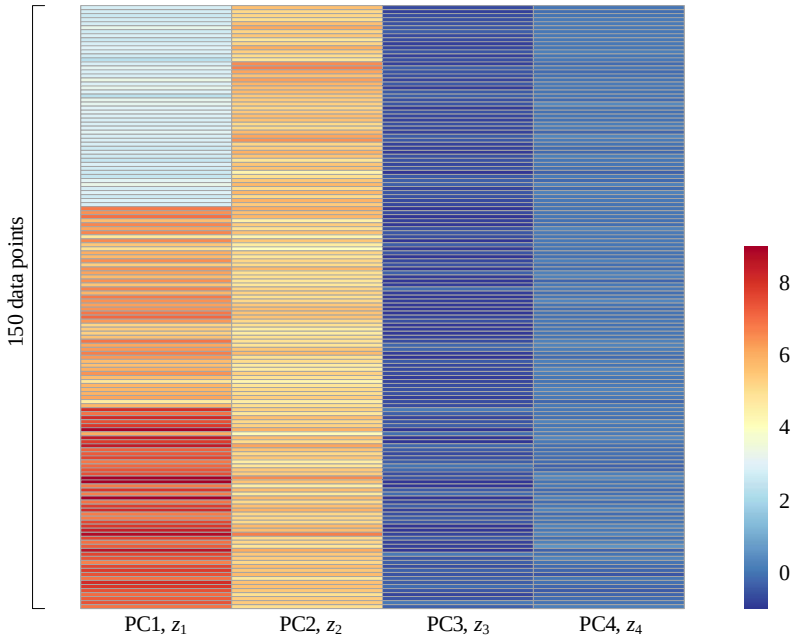


图 17. 新特征数据矩阵 $\boldsymbol{Z}$

对转换数据 $\boldsymbol{Z}$ 进行统计分析，以行向量表达数据矩阵 $\boldsymbol{Z}$ 质心：

$$\boldsymbol{\mu}_{\boldsymbol{Z}} = \begin{bmatrix} \underset{\text{PC1, } z_1}{5.502} & \underset{\text{PC2, } z_2}{5.326} & \underbrace{-0.631}_{\text{PC3, } z_3} & \underset{\text{PC4, } z_4}{0.033} \end{bmatrix} \tag{13}$$

数据矩阵 $\boldsymbol{Z}$ 质心和原始数据矩阵 $\boldsymbol{X}$ 质心之间的关系如下所示：

$$\begin{aligned} \boldsymbol{\mu}_{\boldsymbol{Z}} &= \boldsymbol{\mu}_{\boldsymbol{X}} \boldsymbol{V} \\ &= \begin{bmatrix} \underset{\text{Sepal length, } x_1}{5.843} & \underset{\text{Sepal width, } x_2}{3.057} & \underset{\text{Petal length, } x_3}{3.758} & \underset{\text{Petal width, } x_4}{1.199} \end{bmatrix} \begin{bmatrix} 0.521 & 0.377 & 0.720 & -0.261 \\ -0.269 & 0.923 & -0.244 & 0.124 \\ 0.580 & 0.024 & -0.142 & 0.801 \\ \underset{\text{PC1, } \boldsymbol{v}_1}{0.565} & \underset{\text{PC2, } \boldsymbol{v}_2}{0.067} & \underbrace{-0.634}_{\text{PC3, } \boldsymbol{v}_3} & \underbrace{-0.524}_{\text{PC4, } \boldsymbol{v}_4} \end{bmatrix} \\ &= \begin{bmatrix} \underset{\text{PC1, } z_1}{5.502} & \underset{\text{PC2, } z_2}{5.326} & \underbrace{-0.631}_{\text{PC3, } z_3} & \underset{\text{PC4, } z_4}{0.033} \end{bmatrix} \end{aligned} \tag{14}$$

⚠ 注意，若使用 sklearn.decomposition.PCA() 函数进行主成分分析，则会发现数据矩阵 $\boldsymbol{Z}$ 质心均为 0；这是因为数据已经标准化。

$\boldsymbol{Z}$ 每一列均方差，以行向量表达：

$$\boldsymbol{\sigma}_{\boldsymbol{Z}} = \begin{bmatrix} \underset{\text{PC1, } z_1}{2.056} & \underset{\text{PC2, } z_2}{0.492} & \underset{\text{PC3, } z_3}{0.279} & \underset{\text{PC4, } z_4}{0.154} \end{bmatrix} \tag{15}$$

$\boldsymbol{Z}$ 每一列方差，以行向量表达：

$$\boldsymbol{\sigma}_{\boldsymbol{Z}}^2 = \begin{bmatrix} \underset{\text{PC1, } z_1}{4.228} & \underset{\text{PC2, } z_2}{0.242} & \underset{\text{PC3, } z_3}{0.078} & \underset{\text{PC4, } z_4}{0.023} \end{bmatrix} \tag{16}$$

图 18 所示为 KDE 估计得到的转换数据 $\boldsymbol{Z}$ 四个特征分布图。

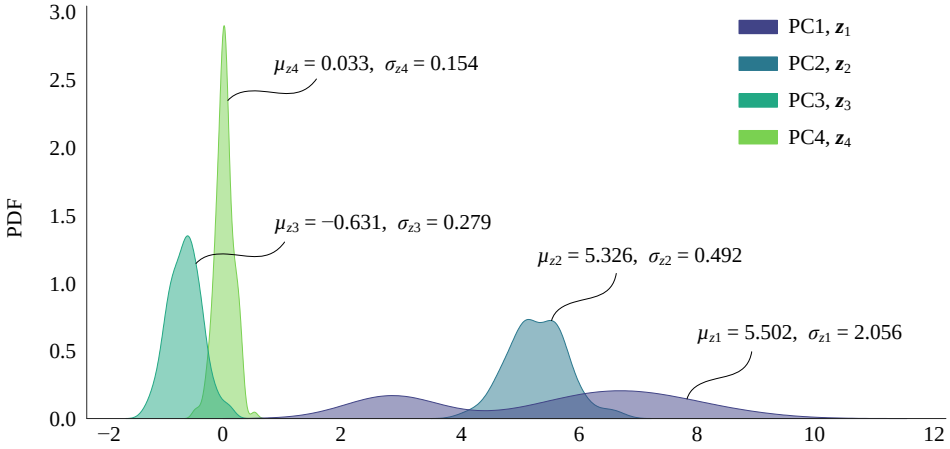


图 18. 转换数据 $\boldsymbol{Z}$ 四个特征上分布，KDE 估计

作为对比，图 19 所示为已经中心化的数据 $\boldsymbol{X}_c$ 朝 $\boldsymbol{V}$ 投影的结果。对比图 18 和图 19，我们可以发现方差没有变化。唯一的区别是，图 19 中所有特征的均值均为 0。

⚠ 注意，$\boldsymbol{V}$ 是通过对协方差矩阵特征值分解得到的。

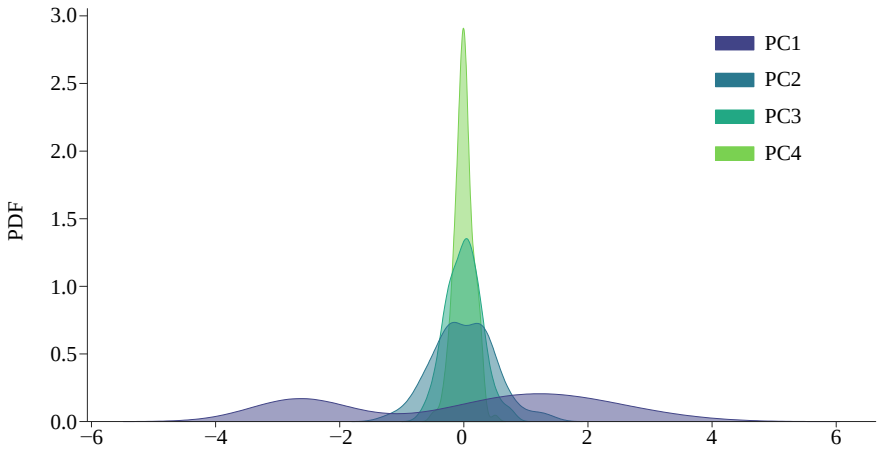


图 19. 转换数据 $\boldsymbol{Z}$ 四个特征上分布，KDE 估计；数据已经中心化

图 20 所示为转换数据 $\boldsymbol{Z}$ 协方差矩阵和相关性系数矩阵热图。

图 21 所示为不分类条件下，转换数据 $\boldsymbol{Z}$ 成对特征分析图；根据本节计算结果，可以知道转换数据 $\boldsymbol{Z}$ 任意两列数据之间的线性相关性系数为 0，也就是正交。图 22 所示为分类条件下，转换数据 $\boldsymbol{Z}$ 成对特征分析图。

$\boldsymbol{Z}$ 的协方差矩阵 $\boldsymbol{\Sigma_Z}$ 和 $\boldsymbol{X}$ 的协方差矩阵 $\boldsymbol{\Sigma_X}$ 之间关系如下：

$$\text{var}(\boldsymbol{X}) = \boldsymbol{\Sigma}_x = \boldsymbol{V}^{\mathrm{T}} \boldsymbol{\Sigma}_x \boldsymbol{V} \tag{17}$$

图 20 所示为转换数据 $\boldsymbol{Z}$ 协方差矩阵和相关性系数矩阵热图。

有关协方差运算，请大家回顾《统计至简》第 14 章。

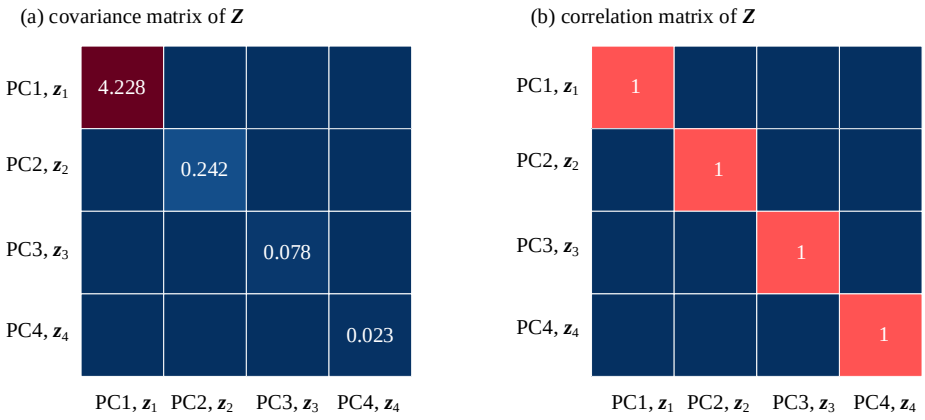


图 20. 转换数据 $\boldsymbol{Z}$ 协方差矩阵和相关性系数矩阵热图

图 21 所示为不分类条件下，转换数据 $\boldsymbol{Z}$ 成对特征分析图；根据本节计算结果，可以知道转换数据 $\boldsymbol{Z}$ 任意两列数据之间的线性相关性系数为 0，也就是正交。图 22 所示为分类条件下，转换数据 $\boldsymbol{Z}$ 成对特征分析图。

下一章还会用椭圆代表散点的分布情况。

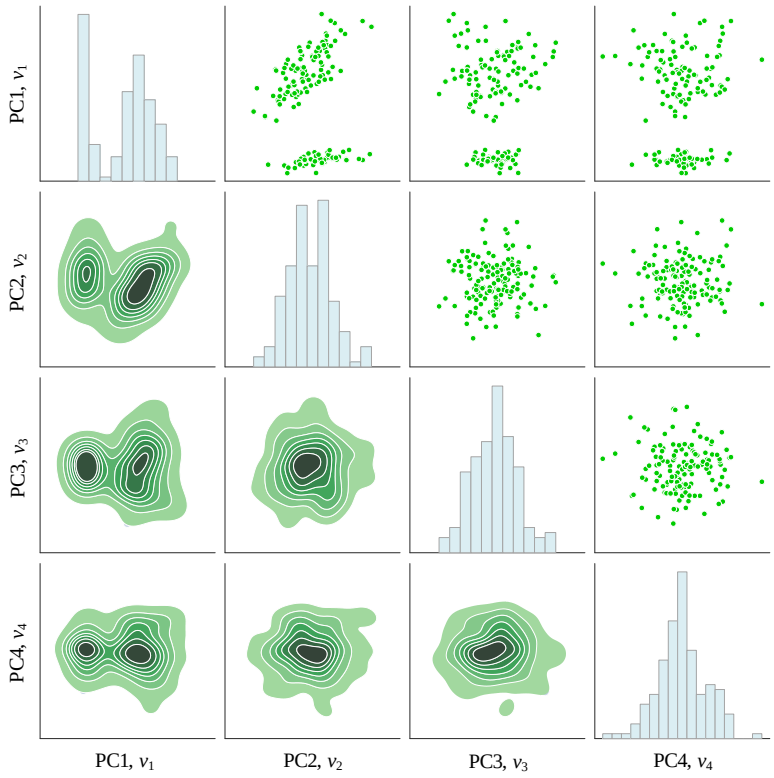


图 21. 转换数据 $\boldsymbol{Z}$ 成对特征分析图，不分类

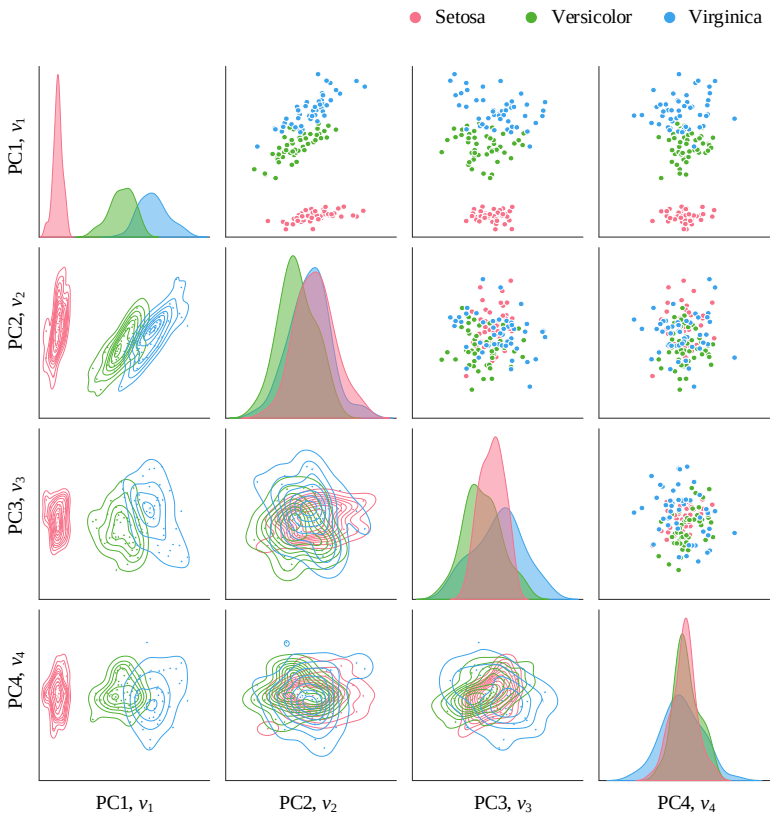


图 22. 转换数据 $\boldsymbol{Z}$ 成对特征分析图，分类

15.5 还原

主成分 $\boldsymbol{v}_1$ 和 $\boldsymbol{v}_2$ 上的投影结果可以用来还原部分原始数据。残差数据矩阵 $\boldsymbol{E}$，即原始热图和还原热图色差，利用下式计算获得：

$$\boldsymbol{E} = \boldsymbol{X} - \hat{\boldsymbol{X}} \tag{18}$$

图 23 所示为 $\boldsymbol{z}_1$ 还原 $\boldsymbol{X}$ 部分数据。图 24 所示为 $\boldsymbol{z}_1$ 还原 $\boldsymbol{X}$ 部分数据。图 25 所示为 $[\boldsymbol{z}_1, \boldsymbol{z}_2]$ 还原 $\boldsymbol{X}$ 部分数据。比较原始数据和图 25 所示 $[\boldsymbol{z}_1, \boldsymbol{z}_2]$ 还原 $\boldsymbol{X}$ 部分数据，可以得到误差热图，如图 26 所示。

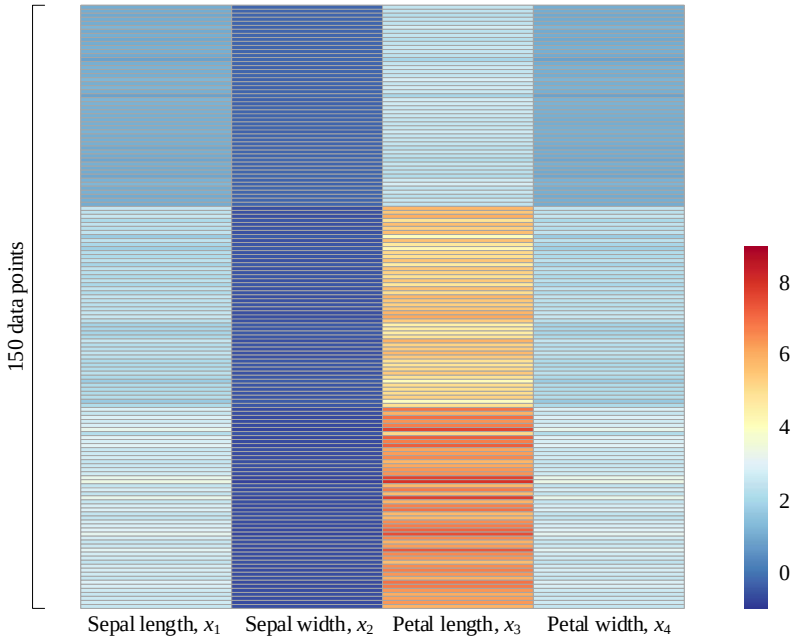


图 23. z_1 还原 $\boldsymbol{X}$ 部分数据

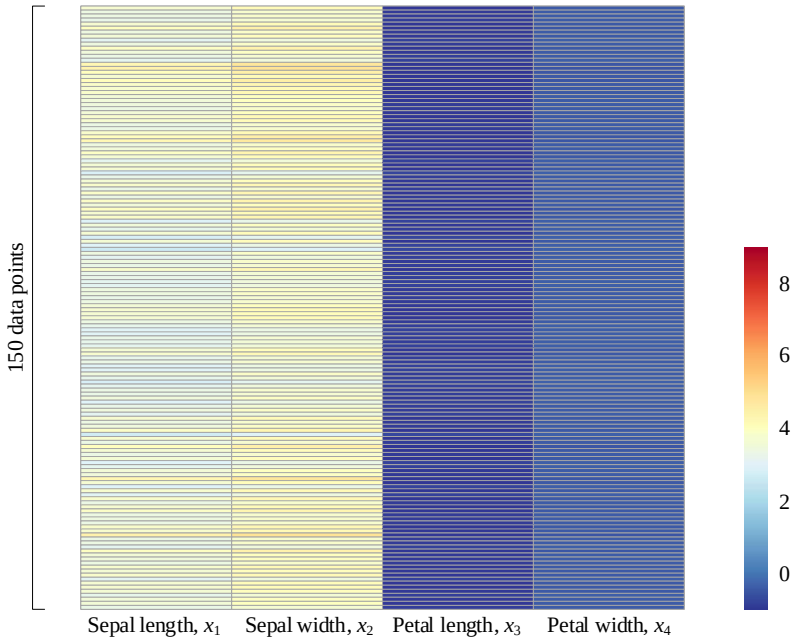


图 24. z_2 还原 $\boldsymbol{X}$ 部分数据

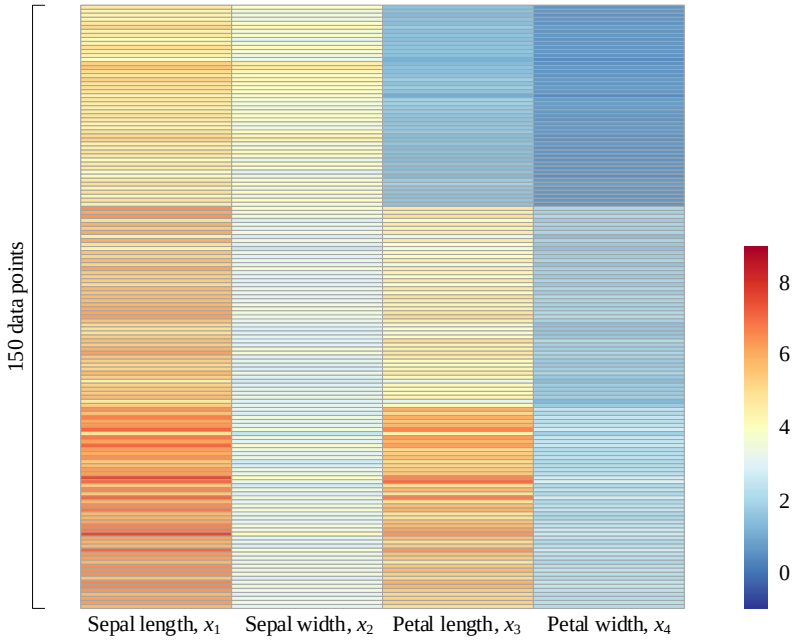


图 25.[$\boldsymbol{z}_1$, $\boldsymbol{z}_2$] 还原 $\boldsymbol{X}$ 部分数据

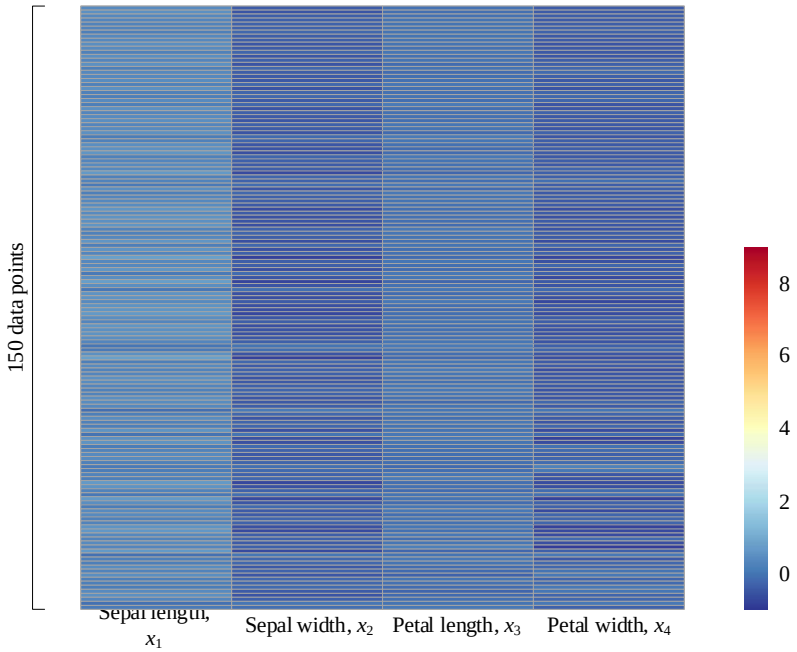


图 26. 误差 $\boldsymbol{E}$

15.6 双标图

双标图 (biplot) 是主成分分析中常用的可视化方案。它能够将高维数据投影到二维或三维空间中，并用散点图的形式展示出来，同时还能够显示原始数据和主成分的信息。一般情况，平面双标图的横坐标和纵坐标分别表示 PCA 的前两个主成分，每个点代表一个样本数据。

通过观察双标图，可以发现不同样本之间的相似性和差异性。如果两个点在双标图上非常接近，那么它们在原始数据中的特征值也可能非常接近，反之亦然。同时，双标图还能够帮助我们找出数据中的异常值和离群点，这些点在双标图上往往会距离其他点较远。

除了用于可视化，双标图还能够用来评估 PCA 的效果。如果双标图中的数据点分布较为均匀且没有聚集在一起，那么说明 PCA 的效果较好，主成分能够较好地解释数据的方差；如果双标图中的数据点呈现出聚集或者明显的分块现象，那么说明 PCA 的效果可能不太理想，主成分并不能完全解释数据的方差。

如图 27 所示，双标图相当于原始数据特征向量向主成分构造的平面投影结果。比如，$\boldsymbol{x}_1$ 向量向 $\boldsymbol{v}_1$-$\boldsymbol{v}_2$ 平面投影，$\boldsymbol{x}_1$ 在 $\boldsymbol{v}_1$ 方向投影得到的标量值为 $v_{1,1}$，$\boldsymbol{x}_1$ 在 $\boldsymbol{v}_2$ 方向投影得到的标量值为 $v_{1,2}$。这两个值对应 $\boldsymbol{V}$ 矩阵第一行前两列数值。

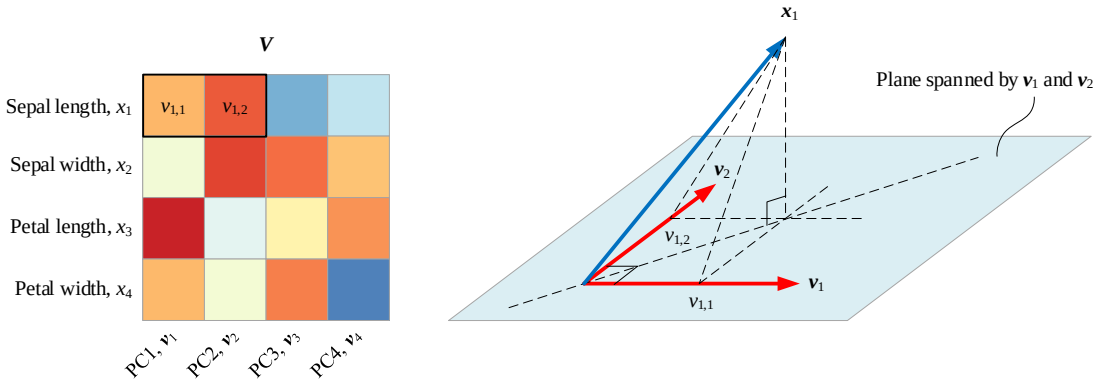


图 27. 双标图原理

图 28 所示为鸢尾花原始数据 PCA 分解后得到的双标图。该图横纵坐标分别是第一主成分 $\boldsymbol{v}_1$ 和第二主成分 $\boldsymbol{v}_2$。如图 28 所示，在双标图上，如果两个特征向量夹角越小，说明两个特征相似度越高，也就是相关性系数越高。比如图中，花萼长度 x_3 和花萼宽度 x_4，在双标图上几乎重合，说明两者相关性极高，(4) 中给出的两者相关性高达 0.963，这也印证了这一点。

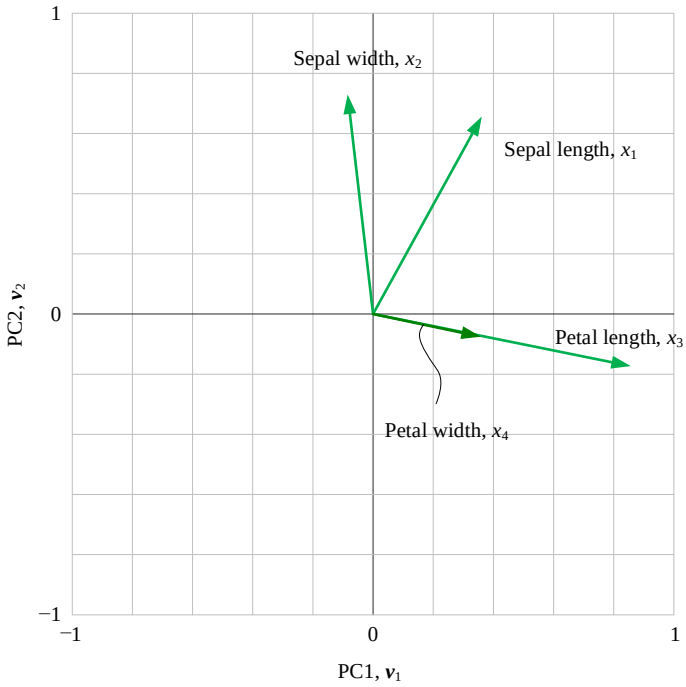


图 28. v_1-v_2 平面双标图，基于鸢尾花原始数据

图 29 所示为向量 x_1、x_2、x_3和 x_4向 v_1-v_2平面投影结果和矩阵 V 之间的数值关系。

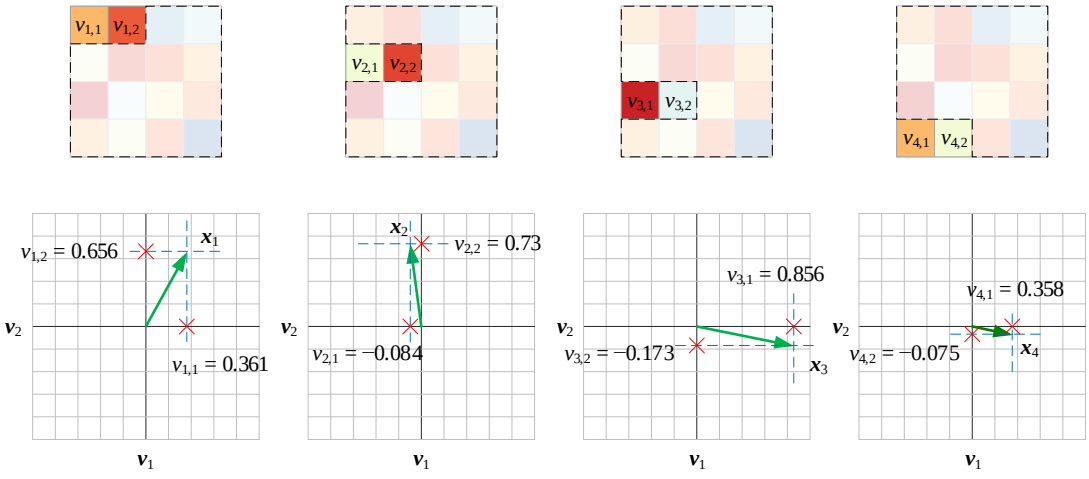


图 29. 向量 x_1、x_2、x_3和 x_4向 v_1-v_2平面投影结果

图 30 所示为向量 x_1、x_2、x_3和 x_4向 v_3-v_4平面投影结果。

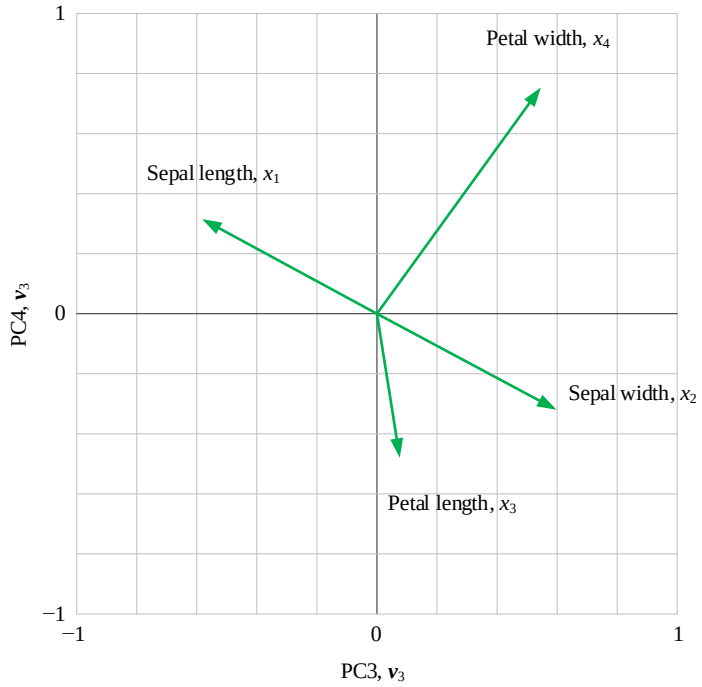


图 30. v_3-v_4 平面双标图，基于鸢尾花原始数据

双标图还可以基于标准化后数据；图 31 所示为基于鸢尾花标准化数据后的双标图，投影值对应 (10)。

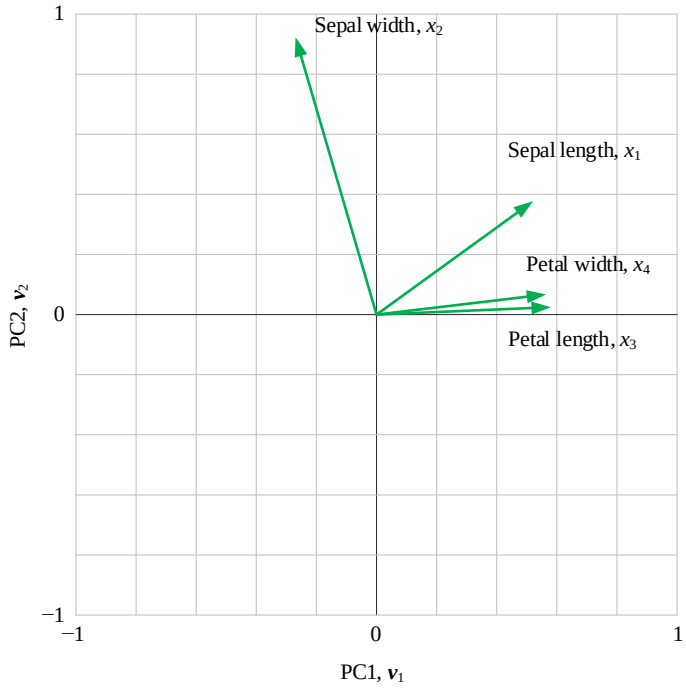


图 31. 平面双标图，基于鸢尾花标准化数据

此外，除了特征向量之外，双标图还会绘制数据点投影，如图 32 所示。图 32 采用 yellowbrick.features.PCA() 绘制。该函数绘制的双标图基于标准化鸢尾花数据。双标图中，点与点之间的距离，反映它们对应的样本之间的差异大小，两点相距较远，对应样本差异大；两点相距较近，对应样本差异小，存在相似性。

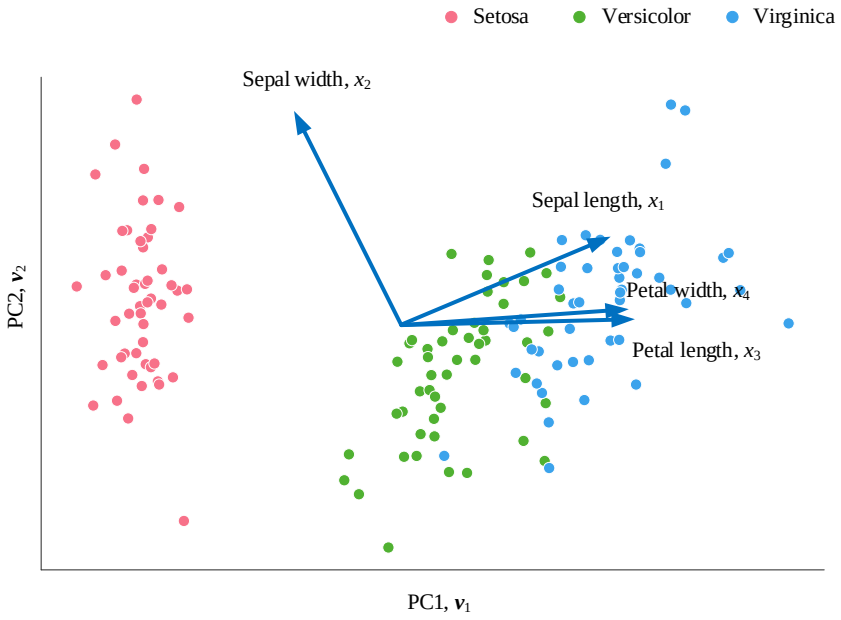


图 32. 平面双标图，标准化数据

图 33 给出的是由前三个主成分构造的空间，也就是将原始数据和它的四个特征向量投影到这个三维正交空间。该图也是采用 yellowbrick.features.PCA() 绘制。

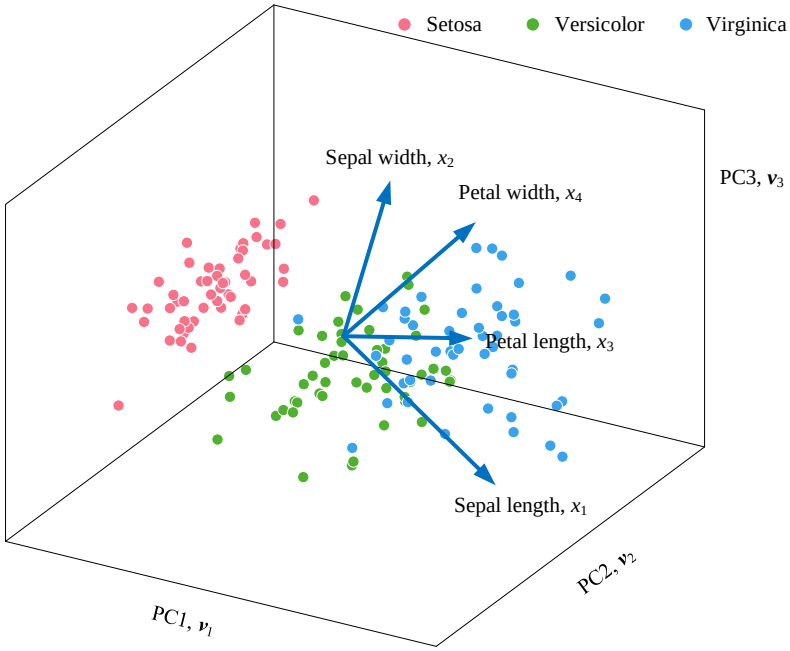


图 33. 三维双标图

15.7 陡坡图

《统计至简》第 25 章介绍过，第 j 个特征值 λ_j 对方差总和的贡献百分比为：

$$\frac{\lambda_j}{\sum_{i=1}^{D}\lambda_i}\times 100\% \tag{19}$$

上式分母是数据总方差。

协方差矩阵 $\boldsymbol{\Sigma}$ 的迹——方阵对角线元素之和——等于特征值之和，请大家回顾《统计至简》第 13 章。

(19) 这个比值可以用来衡量第 j 个主成分对数据的解释能力。如果已释方差较大，那么说明第 j 个主成分能够较好地解释数据的方差，即它包含了较多的信息。如果已释方差较小，那么说明第 k 个主成分对数据的解释能力较弱，不足以对数据进行有效的降维和特征提取。

前 p 个特征值累积解释总方差的百分比为：

$$\frac{\sum_{j=1}^{p}\lambda_j}{\sum_{i=1}^{D}\lambda_i}\times 100\% \tag{20}$$

这个比值代表前 p 个主成分所能解释的已释方差之和占所有主成分已释方差之和的比例。累计已释方差和百分比能够用来评估 PCA 的降维效果，它衡量了前 p 个主成分能够解释数据方差的比例。

通常来说，我们希望通过选择适当的主成分数 p，使累计已释方差和百分比达到预设的阈值 (比如 80%或 90%)，以保留尽可能多的原始数据信息。通过观察累计已释方差和百分比的变化趋势，我们可以得出选择适当主成分数的建议，以及对 PCA 的降维效果进行评估和比较。

图 34 给出图像可视化 (19) 和 (20) 。鸢尾花数据的主成分分析特征值如下：

$$\lambda_1=4.228,\quad \lambda_2=0.242,\quad \lambda_3=0.078,\quad \lambda_4=0.023 \tag{21}$$

PCA 主成分顺序根据各个主成分维度方向方差贡献大小排序。第一主成分方向上的方差最大，也就是这个方向最有力地解释了数据的分布。当第一主成分的方差贡献不足 (比如小于 50%)，我们就要依次引入其它主成分。如图 34 所示，第一和第二主成分两者已释方差之和为 72.5%。

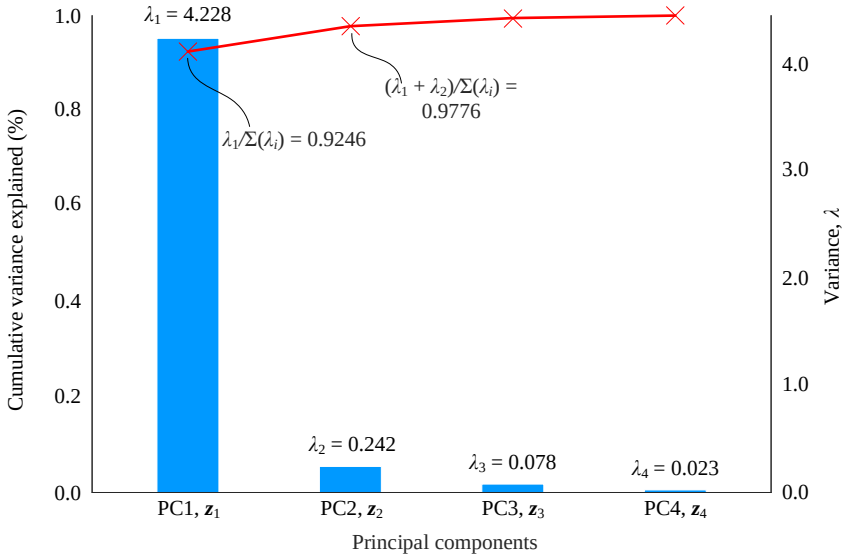


图 34. 陡坡图

Bk6_Ch15_01.py 绘制本章前文大部分图片。

15.8 分析鸢尾花照片

本节用 PCA 分析一章鸢尾花照片。图 35 所示为作者拍的一章鸢尾花照片，经过黑白化处理后的每个像素都是 [0, 1] 范围内的数字。所以整幅图片可以看成一个数据矩阵。

《可视之美》一册专门介绍过彩色和黑白图像之间转换。

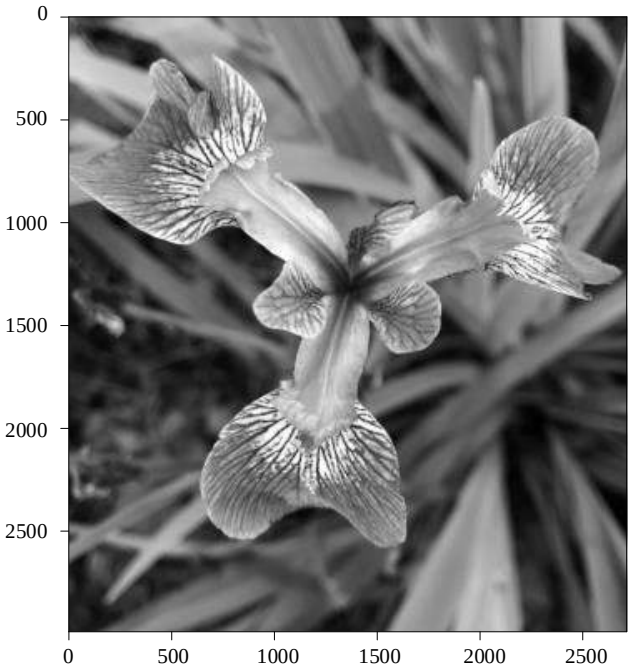


图 35. 鸢尾花图片，经过黑白处理

图 36 所示为利用 SVD 分解得到的奇异值随主成分变化。图 37 所示为特征值随主成分变化。图 38 所示为累积解释方差百分比随主成分变化。我们可以发现前 10 个主成分已经解释超过 90%的方差。

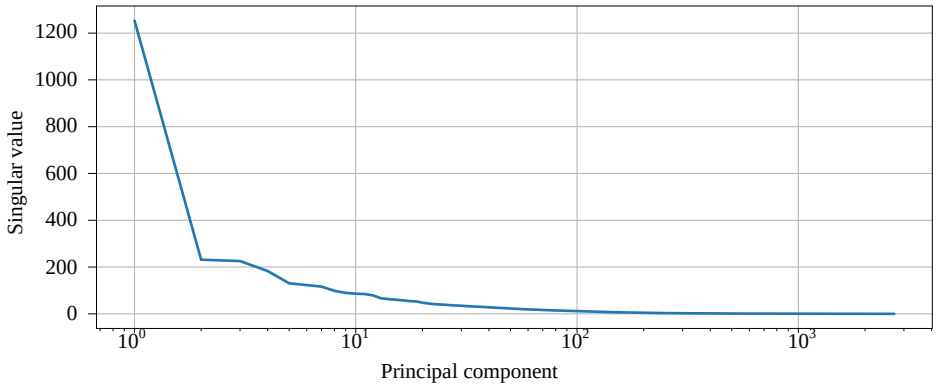


图 36. 奇异值随主成分变化

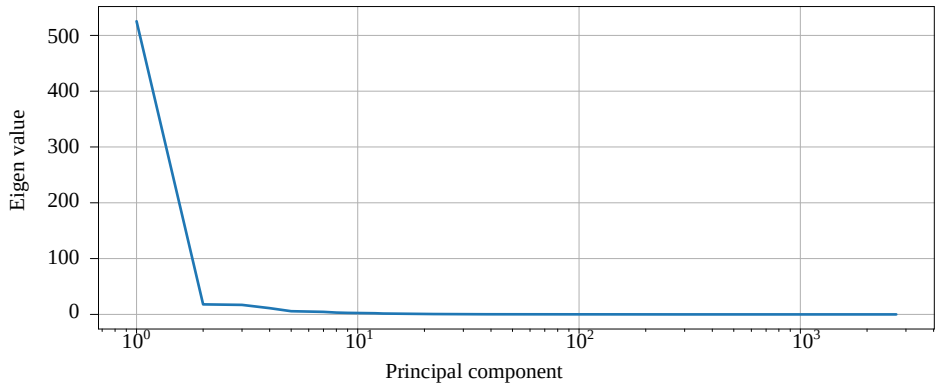


图 37. 特征值随主成分变化

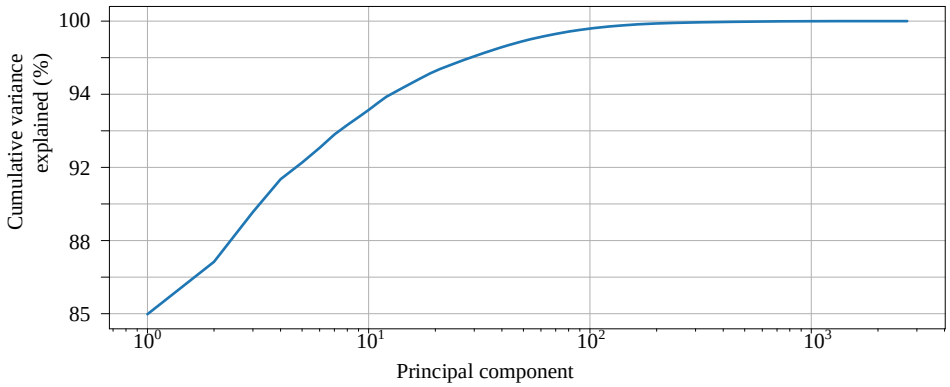


图 38. 累积解释方差百分比随主成分变化

图 39 所示为利用第 1 主元还原鸢尾花图片，左图为还原结果，右图为误差。左图中，鸢尾花还难觅踪影。图 40 所示为利用第 1、2 主元还原鸢尾花照片，图 41 所示为利用前 4 个主元还原鸢尾花照片，在两幅图的左图中我们仅仅能够看到“格子”。图 42 的左图利用前 16 个主元还原照片，我们已经能够看到鸢尾花的样子，注意这幅图的秩为 16。图 43 所示为利用前 64 个主元还原鸢尾花图片，图形已经很清晰。相比原图片，图 43 的数据发生大幅压缩。

这种利用 PCA 进行图像降维方法用途很广泛。比如，在人脸识别中，**特征脸** (eigenface) 是一种基于 PCA 的特征提取方法，用于将人脸图像转换成低维特征向量进行分类或识别。特征脸是指由 PCA 分解出来的主成分图像，它们是一组基于训练数据集的线性组合，每个特征脸表示了一个数据集中的特定方向，可以看作是数据集的主要特征或重要性征。

特征脸的提取过程可以分为以下几步：1) 对人脸图像进行预处理，比如灰度化、尺度归一化、去除噪声等。2) 将预处理后的图像转换成向量形式。3) 将向量集合进行 PCA 降维，得到一组主成分向量，也就是特征脸。4） 将人脸图像向量投影到主成分向量上，得到每个人脸的特征向量表示。

特征脸在人脸识别中的作用是对人脸图像进行有效的特征提取和降维，使得原始图像数据被压缩到一个低维空间中，并且保留了原始数据中的大部分信息。通过比较人脸图像的特征向量之间的相似度，可以进行人脸识别、验证等应用。

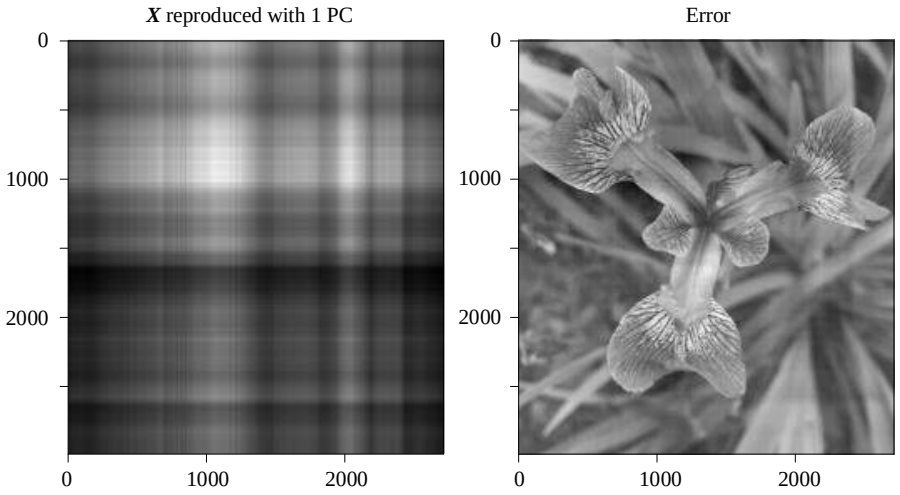


图 39. 利用第 1 主元还原鸢尾花照片

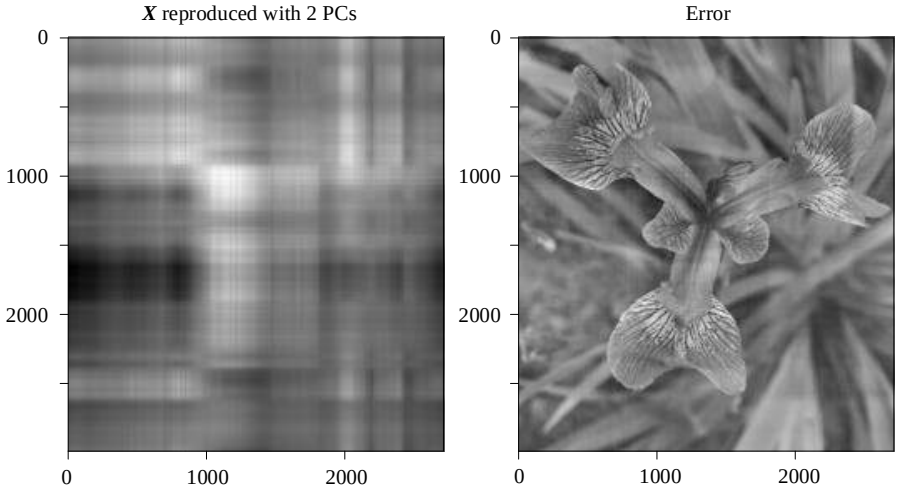


图 40. 利用第 1、2 主元还原鸢尾花照片

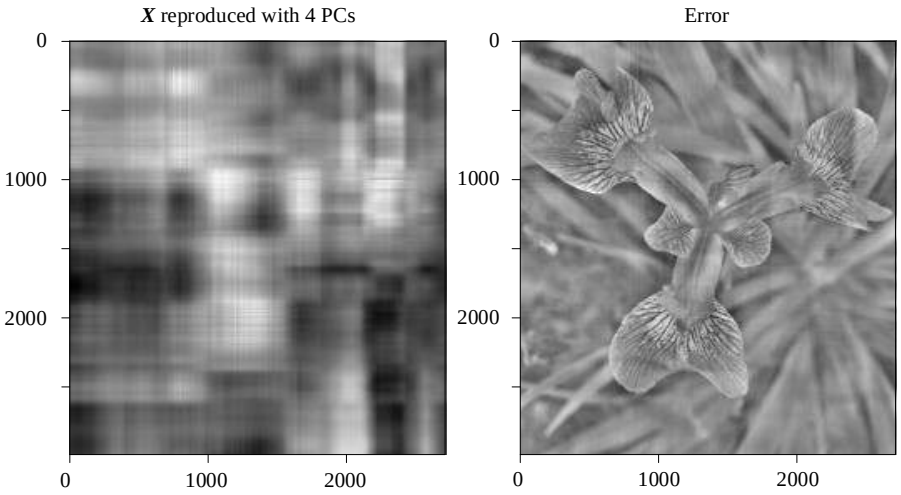


图 41. 利用第 1、2、3、4 主元还原鸢尾花照片

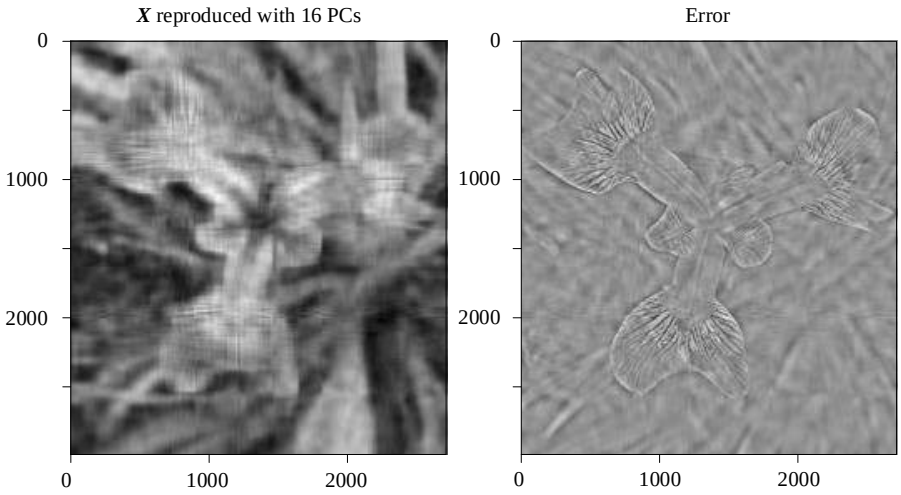


图 42. 利用前 16 个主元还原鸢尾花照片

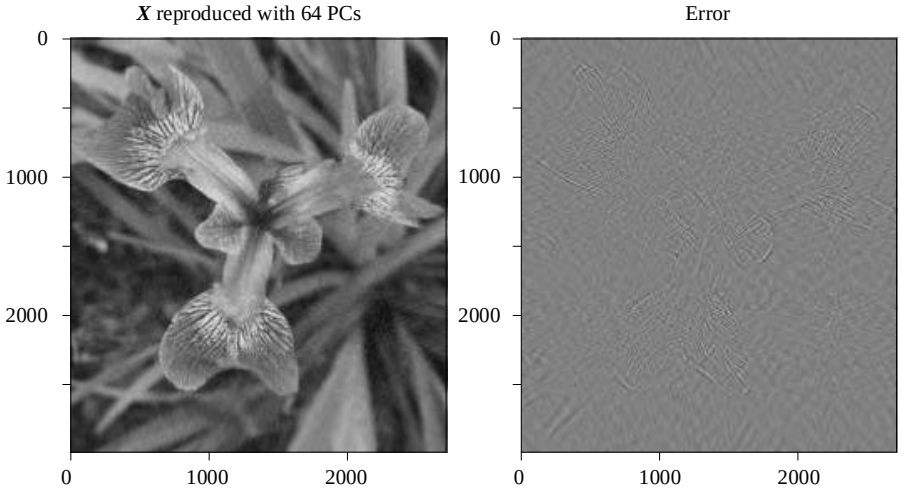


图 43. 利用前 64 个主元还原鸢尾花照片

Bk6_Ch15_02.py 绘制本节图片。鸢尾花照片也在文件夹中。

主成分分析是一种广泛使用的数据降维和特征提取技术，它可以将高维数据降至低维，同时保留数据的主要特征和结构。PCA 通过寻找一组最能解释数据变异性的线性组合，即主成分，来实现数据降维和特征提取。主成分是原始特征的线性组合，它们的排序代表了它们的重要性。通常，我们只需要保留前几个主成分，因为它们可以解释大部分数据的变异性。

一般的 PCA 步骤包括：中心化 (标准化) 数据、计算协方差矩阵、计算特征值和特征向量、排序特征值和对应的特征向量、选择前 p 个主成分、计算投影矩阵并对数据进行降维。在计算特征值和特征向量时，我们通常使用特征值分解，当然也可以使用奇异值分解，这是下一章要介绍的内容。

PCA 的投影可以帮助我们理解数据的结构和关系。投影到第一二主成分方向上的投影数据通常成椭圆形状，其中椭圆的长轴方向表示最大的方差方向，短轴方向表示最小的方差方向。通过线性组合，我们可以将主成分重新组合成原始数据，并通过双标图和陡坡图来分析 PCA 的效果。双标图可以帮助我们了解主成分之间的相关性，陡坡图可以帮助我们了解主成分的贡献程度。

在 PCA 中，理解数据和分析结果的视角非常重要。这涉及到如何选择主成分和如何解释它们，以及如何应用 PCA 的结果。选择主成分时，我们通常考虑主成分的贡献程度和解释能力，以及降维后的数据能否保留足够的信息。解释主成分时，我们需要考虑主成分的物理意义和应用背景。应用 PCA 的结果时，我们可以利用降维后的数据进行可视化、聚类、分类等分析。

总之，主成分分析是一种强大的数据降维和特征提取技术，它可以帮助我们更好地理解和分析数据。在应用 PCA 时，需要注意数据预处理、主成分选择和解释、以及降维后的数据应用等问题。下一章将比较六种不同的 PCA 技术路线。

15

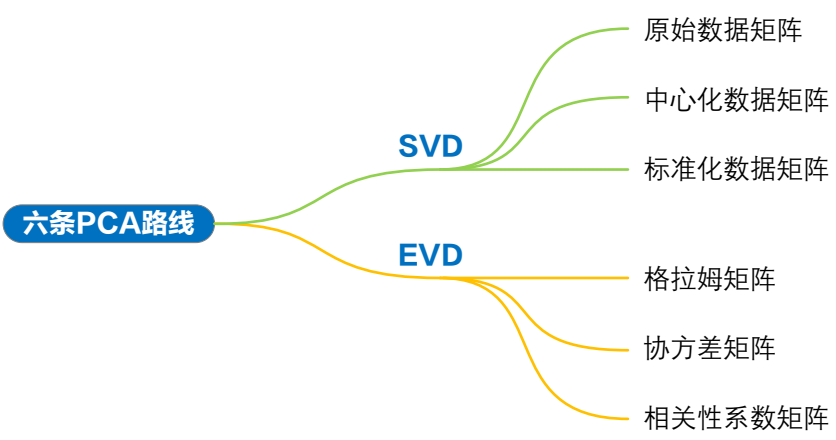
Dive into Principal Component Analysis

主成分分析进阶

区分六条基本 PCA 技术路线

> 我发现了！
>
> ***Eureka!***
>
> —— 阿基米德 (Archimedes) | 数学家、发明家、物理学家 | 287 ~ 212 BC

- `numpy.cov()` 计算协方差矩阵
- `numpy.linalg.eig()` 特征值分解
- `numpy.linalg.svd()` 奇异值分解
- `seaborn.heatmap()` 绘制热图
- `seaborn.kdeplot()` 绘制 KDE 核概率密度估计曲线
- `seaborn.pairplot()` 绘制成对分析图
- `sklearn.decomposition.PCA()` 主成分分析函数



15.1 从“六条技术路线”说起

来自《矩阵力量》的表格

表 1 来自《矩阵力量》第 25 章，本章将讲解表 1 中六条 PCA 技术路线的细节，并比较它们的差异。

表 1. 六条 PCA 技术路线，来自《矩阵分解》第 25 章

对象	方法	结果
原始数据矩阵 $\boldsymbol{X}$	奇异值分解	$\boldsymbol{X} = \boldsymbol{U_X S_X V_X}^{\mathrm{T}}$
格拉姆矩阵 $\boldsymbol{G} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 本章中用“修正”的格拉姆矩阵 $\boldsymbol{G} = \frac{\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}}{n-1}$	特征值分解	$\boldsymbol{G} = \boldsymbol{V_X \Lambda_X V_X}^{\mathrm{T}}$
中心化数据矩阵 $\boldsymbol{X}_c = \boldsymbol{X} - \mathrm{E}(\boldsymbol{X})$	奇异值分解	$\boldsymbol{X}_c = \boldsymbol{U_c S_c V_c}^{\mathrm{T}}$
协方差矩阵 $\boldsymbol{\Sigma} = \frac{(\boldsymbol{X} - \mathrm{E}(\boldsymbol{X}))^{\mathrm{T}}(\boldsymbol{X} - \mathrm{E}(\boldsymbol{X}))}{n-1}$	特征值分解	$\boldsymbol{\Sigma} = \boldsymbol{V_c \Lambda_c V_c}^{\mathrm{T}}$
标准化数据 (z 分数) $\boldsymbol{Z}_{\boldsymbol{X}} = (\boldsymbol{X} - \mathrm{E}(\boldsymbol{X}))\boldsymbol{D}^{-1}$ $\boldsymbol{D} = \mathrm{diag}(\mathrm{diag}(\boldsymbol{\Sigma}))^{\frac{1}{2}}$	奇异值分解	$\boldsymbol{Z_X} = \boldsymbol{U_Z S_Z V_Z}^{\mathrm{T}}$
相关性系数矩阵 $\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{\Sigma}\boldsymbol{D}^{-1}$ $\boldsymbol{D} = \mathrm{diag}(\mathrm{diag}(\boldsymbol{\Sigma}))^{\frac{1}{2}}$	特征值分解	$\boldsymbol{P} = \boldsymbol{V_Z \Lambda_Z V_Z}^{\mathrm{T}}$

比较六个输入矩阵

表 1 中有六个输入矩阵，它们都衍生自原始数据矩阵 $\boldsymbol{X}$。如图 1 所示，原始数据矩阵 $\boldsymbol{X}$ 的形状为 $n \times D$。

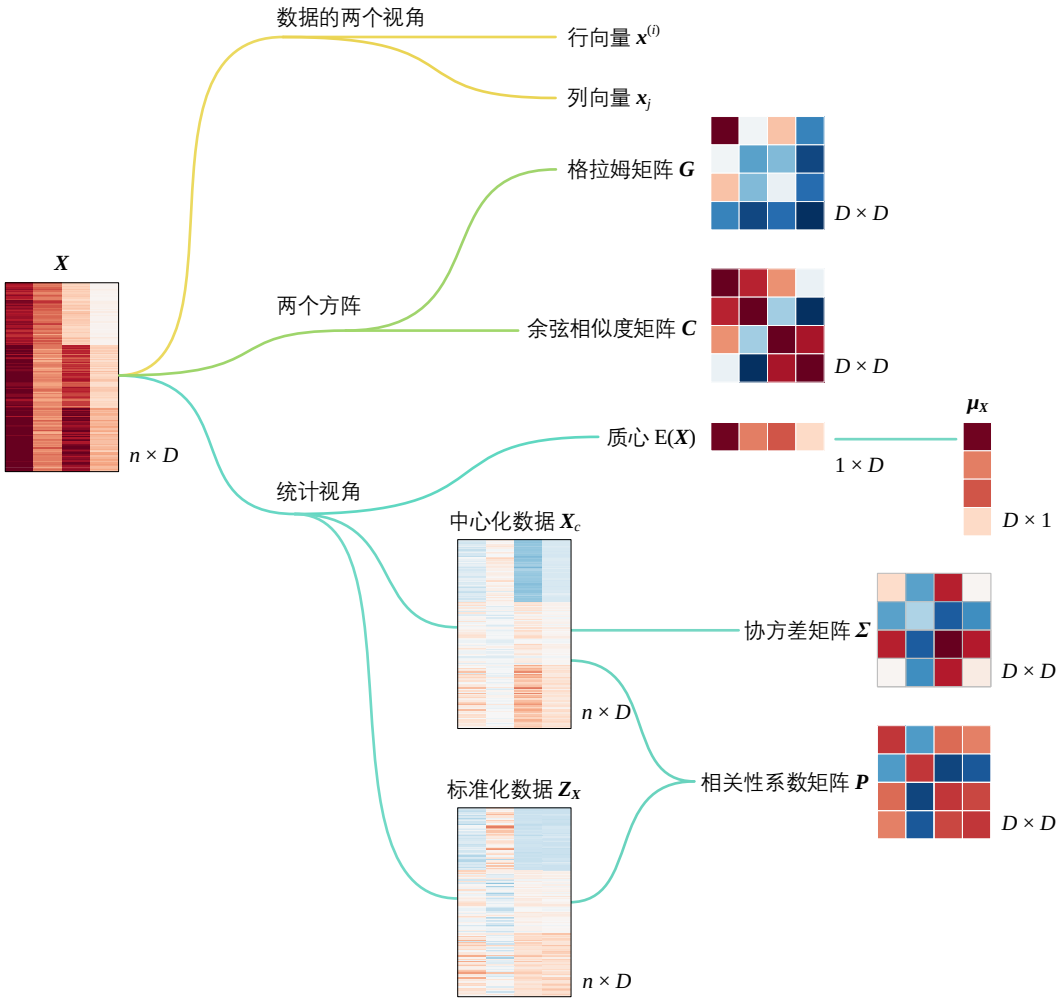


图 1. X 衍生得到的几个矩阵，来自《矩阵力量》

X 的格拉姆矩阵 G 为：

$$G = X^{\mathrm{T}} X \tag{1}$$

格拉姆矩阵 G 形状为 $D \times D$。G 的主对角线元素是 X 的每一列向量 L^2 模的平方。

中心化 (去均值) 矩阵 X_c 为：

$$X_c = X - \mathrm{E}(X) \tag{2}$$

即 X 的每一列分别减去各自的均值得到 X_c。几何角度，X 的质心位于 E(X)，X_c 的质心则位于原点 0。

样本数据矩阵 X 的协方差矩阵 Σ 为：

$$\Sigma = \frac{X_c^{\mathrm{T}} X_c}{n-1} = \frac{\left(X - \mathrm{E}(X)\right)^{\mathrm{T}} \left(X - \mathrm{E}(X)\right)}{n-1} \tag{3}$$

容易发现，协方差相当于特殊的格拉姆矩阵。

请大家特别注意，为了方便和协方差比较，本章中 $\boldsymbol{G}$ 特别定义为：

$$\boldsymbol{G} = \frac{\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}}{n-1} \tag{4}$$

标准化 (standardization 或 z-score normalization) 数据矩阵 $\boldsymbol{Z_X}$ 为：

$$\boldsymbol{Z_X} = \left(\boldsymbol{X} - \mathrm{E}\left(\boldsymbol{X}\right)\right)\boldsymbol{D}^{-1} \tag{5}$$

其中 $\boldsymbol{D}$ 为：

$$\boldsymbol{D} = \mathrm{diag}\left(\mathrm{diag}\left(\boldsymbol{\Sigma}\right)\right)^{\frac{1}{2}} = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_D \end{bmatrix} \tag{6}$$

(5) 中的每一列都是每个特征的 Z 分数。$\boldsymbol{Z_X}$ 的质心也位于原点，不同的是 $\boldsymbol{Z_X}$ 每个特征的标准差都是 1。

线性相关性系数矩阵 $\boldsymbol{P}$ 为：

$$\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{\Sigma}\boldsymbol{D}^{-1} \tag{7}$$

$\boldsymbol{P}$ 实际上是 $\boldsymbol{Z_X}$ 的协方差，即：

$$\boldsymbol{P} = \frac{\boldsymbol{Z_X}^{\mathrm{T}}\boldsymbol{Z_X}}{n-1} \tag{8}$$

比较 SVD 和 EVD

主成分分析的核心数学工具为**奇异值分解** (Singular Value Decomposition, SVD) 和**特征值分解** (Eigen Decomposition, EVD)。

《矩阵力量》强调过 SVD 和 EVD 在主成分分析中具有等价性，这也就是为什么表 1 看上去是六种技术路线，实际上可以归纳为三大类技术路线。下面简单说明一下。

对原始矩阵 $\boldsymbol{X}$ 进行经济型 SVD 分解：

$$\boldsymbol{X} = \boldsymbol{U_X}\boldsymbol{S_X}\boldsymbol{V_X}^{\mathrm{T}} \tag{9}$$

其中，$\boldsymbol{S_X}$ 为对角方阵。

将 (9) 代入 (1)：

$$\boldsymbol{G} = \boldsymbol{V_X}\boldsymbol{S_X}^2\boldsymbol{V_X}^{\mathrm{T}} \tag{10}$$

上式便是格拉姆 $\boldsymbol{G}$ 的特征值分解。

对中心化数据矩阵 $\boldsymbol{X}_c$ 经济型 SVD 分解：

$$\boldsymbol{X}_c = \boldsymbol{U}_c \boldsymbol{S}_c \boldsymbol{V}_c^{\mathrm{T}} \tag{11}$$

而协方差矩阵 $\boldsymbol{\Sigma}$ 则可以写成：

$$\boldsymbol{\Sigma} = \boldsymbol{V}_c \frac{\boldsymbol{S}_c^2}{n-1} \boldsymbol{V}_c^{\mathrm{T}} \tag{12}$$

相信大家在上式中能够看到协方差矩阵 $\boldsymbol{\Sigma}$ 的特征值分解。请大家注意 (11) 中奇异值和 (12) 中特征值关系：

$$\lambda_{c\_j} = \frac{s_{c\_j}^2}{n-1} \tag{13}$$

同样，对标准化数据矩阵 $\boldsymbol{Z}_X$ 进行经济型 SVD 分解：

$$\boldsymbol{Z}_X = \boldsymbol{U}_Z \boldsymbol{S}_Z \boldsymbol{V}_Z^{\mathrm{T}} \tag{14}$$

相关性系数矩阵 $\boldsymbol{P}$ 则可以写成：

$$\boldsymbol{P} = \boldsymbol{V}_Z \frac{\boldsymbol{S}_Z^2}{n-1} \boldsymbol{V}_Z^{\mathrm{T}} \tag{15}$$

上式相当于对 $\boldsymbol{P}$ 特征值分解。

本章下面将分别讲解特征值分解 1) 协方差矩阵、2）格拉姆矩阵、3) 相关性系数矩阵，来完成主成分分析。并利用诸如热图、饼图、直方图、陡坡图、双标图、椭圆等可视化工具分析三种路线。本章以下三节将采用完全相似的结构，方便大家比较三大类不同 PCA 技术路线的异同。

15.2 协方差矩阵

本节讲解利用特征值分解协方差矩阵 $\boldsymbol{\Sigma}$ 完成主成分分析。

特征值分解

图 2 所示为特征值分解协方差矩阵 $\boldsymbol{\Sigma}$。$\boldsymbol{\Sigma}$ 的对角线元素为方差，其他元素为协方差。$\boldsymbol{\Sigma}$ 的迹代表方差之和：

$$\mathrm{trace}\left(\boldsymbol{\Sigma}\right) = \sigma_1^2 + \sigma_2^2 + \cdots + \sigma_D^2 = \sum_{j=1}^{D} \sigma_j^2 \tag{16}$$

图 2 中 $\boldsymbol{\Sigma}$ 为对称矩阵，因此对 $\boldsymbol{\Sigma}$ 的特征值分解实际上是谱分解。

Λ_c为对角矩阵，对角线元素为特征值，特征值从大到小排列。X_c投影到规范正交基V_c中得到Y_c，即$Y_c = X_c V_c$。Λ_c主对角线上的特征值实际上是Y_c的方差，也就是说Λ_c是Y_c的协方差矩阵。因此，在主成分分析中，特征值也叫主成分方差。

Λ_c的方差，即特征值，之和为：

$$\text{trace}(\Lambda_c) = \lambda_1 + \lambda_2 + \cdots \lambda_D = \sum_{j=1}^{D} \lambda_j \tag{17}$$

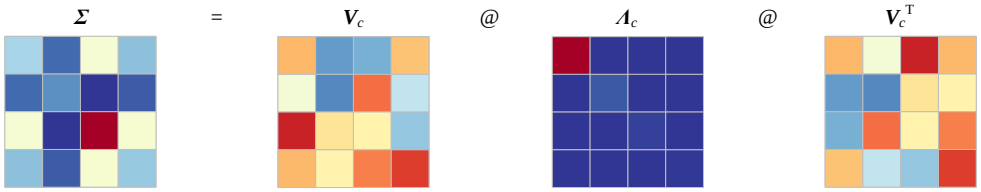


图 2. 特征值分解协方差矩阵Σ

图 16 对比格拉姆矩阵G和Λ_X。

下面，我们进一步分析这两个矩阵。

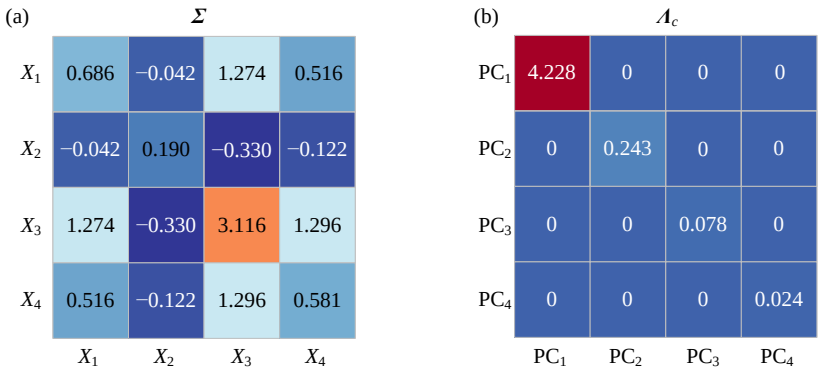


图 3. 对比协方差矩阵Σ和Λ_c热图

分解前后

大家在本书第 12 章已经见过图 4 和图 5。

如图 4 所示，数据矩阵X中第三列，即X_3，的方差最大，X_3对方差和 trace(Σ) 贡献超过 68%。

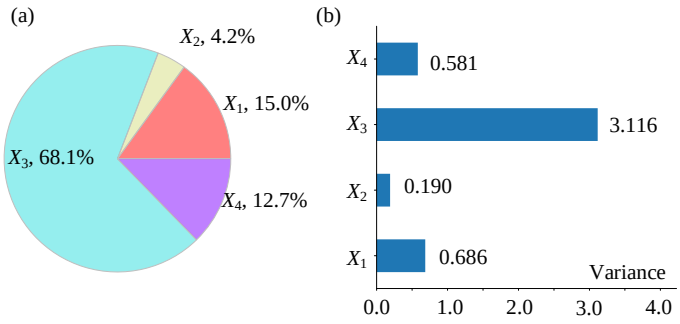


图 4. 协方差矩阵 $\boldsymbol{\Sigma}$ 的主对角线成分，即方差

我们在《矩阵力量》第 13 章提过，特征值分解前后矩阵的迹不变，也就是说协方差矩阵 $\boldsymbol{\Sigma}$ 的迹 trace($\boldsymbol{\Sigma}$) 等于的特征值方阵 $\boldsymbol{\Lambda}_c$ 迹 trace($\boldsymbol{\Lambda}_c$)：

$$\text{trace}\left(\boldsymbol{\Sigma}\right) = \text{trace}\left(\boldsymbol{\Lambda}_c\right) \tag{18}$$

即：

$$\sum_{j=1}^{D}\sigma_j^2 = \sum_{j=1}^{D}\lambda_j \tag{19}$$

也就是说，PCA 不改变数据各个特征方差总和。

而第 j 个特征值 λ_j 对 trace($\boldsymbol{\Lambda}_c$) 的贡献百分比为：

$$\frac{\lambda_j}{\sum_{i=1}^{D}\lambda_i}\times 100\% \tag{20}$$

如图 5 所示，第一主成分的贡献超过 92%，解释了数据中大部分“方差”。数据分析中，如果原始数据特征很多，彼此之间又具有复杂的相关性，那么我们就可以考虑利用主成分分析对数据进行“降维”，减少特征的数量。而这个过程又保留了原始数据主要的信息。

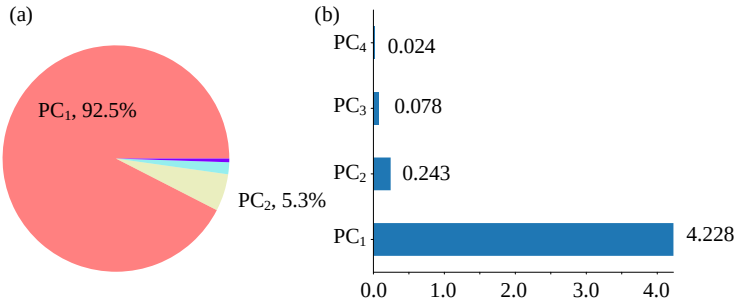


图 5. $\boldsymbol{\Lambda}_c$ 的主对角线成分，协方差矩阵 $\boldsymbol{\Sigma}$ 的特征值

陡坡图

上一章介绍过我们经常用陡坡图可视化前 p 个主成分解释总方差的百分比，即累积贡献率：

$$\frac{\sum_{j=1}^{p}\lambda_j}{\sum_{i=1}^{D}\lambda_i}\times 100\% \tag{21}$$

图 6 所示为特征值分解协方差矩阵 $\boldsymbol{\Sigma}$ 获得的陡坡图。观察陡坡图，可以帮助我们确定选取多少个主成分。

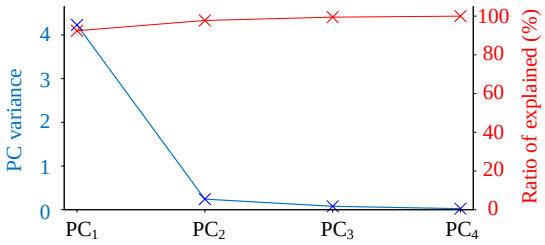


图 6. 陡坡图，特征值分解协方差矩阵 $\boldsymbol{\Sigma}$

特征向量矩阵

图 7 所示为特征向量矩阵 $\boldsymbol{V}_c$ 热图。$\boldsymbol{V}_c$ 的每一列便代表一个主成分的方向，即 $\boldsymbol{V}_c = [\boldsymbol{v}_{c\_1}, \boldsymbol{v}_{c\_2}, \boldsymbol{v}_{c\_3}, \boldsymbol{v}_{c\_4}]$ 从左到右分别是第一、二、三、四主成分。这些主成分方向两两正交。

在主成分分析中，$\boldsymbol{V}_c$ 叫主成分系数，也称为载荷 (loading)。注意，有一些参考文献中，载荷还要乘上特征值的平方根，即 $\boldsymbol{v}_j\sqrt{\lambda_j}$ 。

$\boldsymbol{V}_c$ 也可以通过经济型 SVD 分解中心化矩阵 $\boldsymbol{X}_c$ 得到。

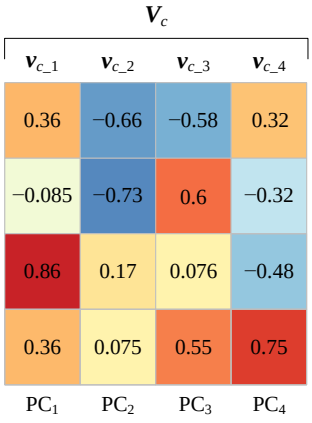


图 7. 特征向量矩阵 $\boldsymbol{V}_c$ 热图

投影

由于 $\boldsymbol{V}_c$ 为正交矩阵，满足 $\boldsymbol{V}_c^{\mathrm{T}}\boldsymbol{V}_c = \boldsymbol{V}_c\boldsymbol{V}_c^{\mathrm{T}} = \boldsymbol{I}$，因此 $\boldsymbol{V}_c$ 本身也是规范正交基。如图 8 所示，将中心化矩阵 $\boldsymbol{X}_c$ 投影到 $\boldsymbol{V}_c$ 这个规范正交基中得到数据矩阵 $\boldsymbol{Y}_c$，即 $\boldsymbol{Y}_c = \boldsymbol{X}_c\boldsymbol{V}_c$。通过图 8 中的 $\boldsymbol{Y}_c$ 每一列的色差，我们就可以看出来不同的次序主成分对数据总体方差的解释力度。

《矩阵力量》第 18 章介绍过 SVD 分解的优化视角。

利用 L^2 范数，$\boldsymbol{V}_c$ 的第一列列向量实际上是如下优化问题的解：

$$\begin{aligned} \boldsymbol{v}_{c\_1} &= \arg\max_{\boldsymbol{v}} \ \left\| \boldsymbol{X}_c \boldsymbol{v} \right\| \\ &\text{subject to: } \left\| \boldsymbol{v} \right\| = 1 \end{aligned} \tag{22}$$

前文提过，$\boldsymbol{\Lambda_X}$ 本身是 $\boldsymbol{Y}_c$ 的协方差矩阵。$\boldsymbol{\Lambda_X}$ 为对角方阵，因此 $\boldsymbol{Y}_c$ 的任意两列之间线性相关系数为 0。也就是说，$\boldsymbol{V}_c$ 完成了 $\boldsymbol{X}_c$ 的正交化，注意不是原始数据矩阵 $\boldsymbol{X}$ 的正交化。

请大家思考 $\boldsymbol{Y}_c$ 的每一列的均值是多少？$\boldsymbol{Y}_c$ 的质心位置是什么？为什么？

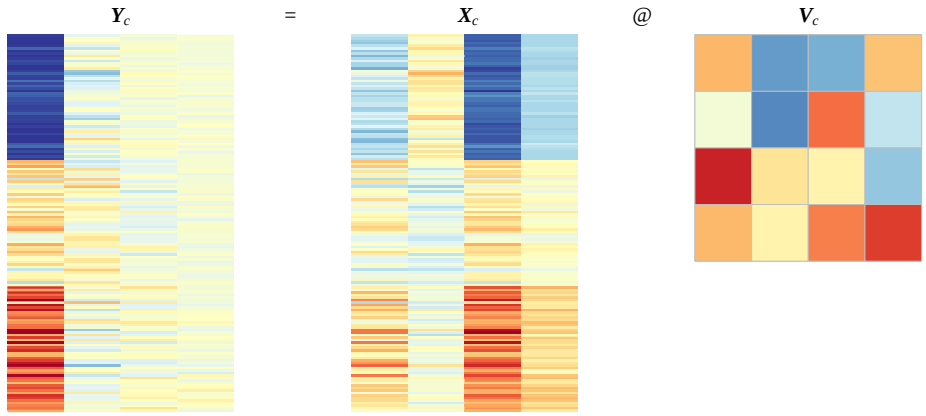


图 8. 将中心化数据 $\boldsymbol{X}_c$ 投影到 $\boldsymbol{V}_c$

双标图

如图 9 所示，双标图是可视化特征向量矩阵 $\boldsymbol{V}_c$ 的重要方法。

以图 9 中蓝色背景的双标图为例，中心化数据 $\boldsymbol{X}_c$ 投影到第一、二主成分平面内的结果如四个箭头所示。比如，X_1、X_2、X_3、X_4 在 PC1 上贡献的分量分别为 0.36、−0.085、0.86、0.36，这正是如图 7 所示的 $\boldsymbol{V}_c$ 第一列 $\boldsymbol{v}_{c\_1}$。

我们还可以把投影数据的散点图也画在双标图上，大家已经在上一章看到很多例子，本章不再重复。

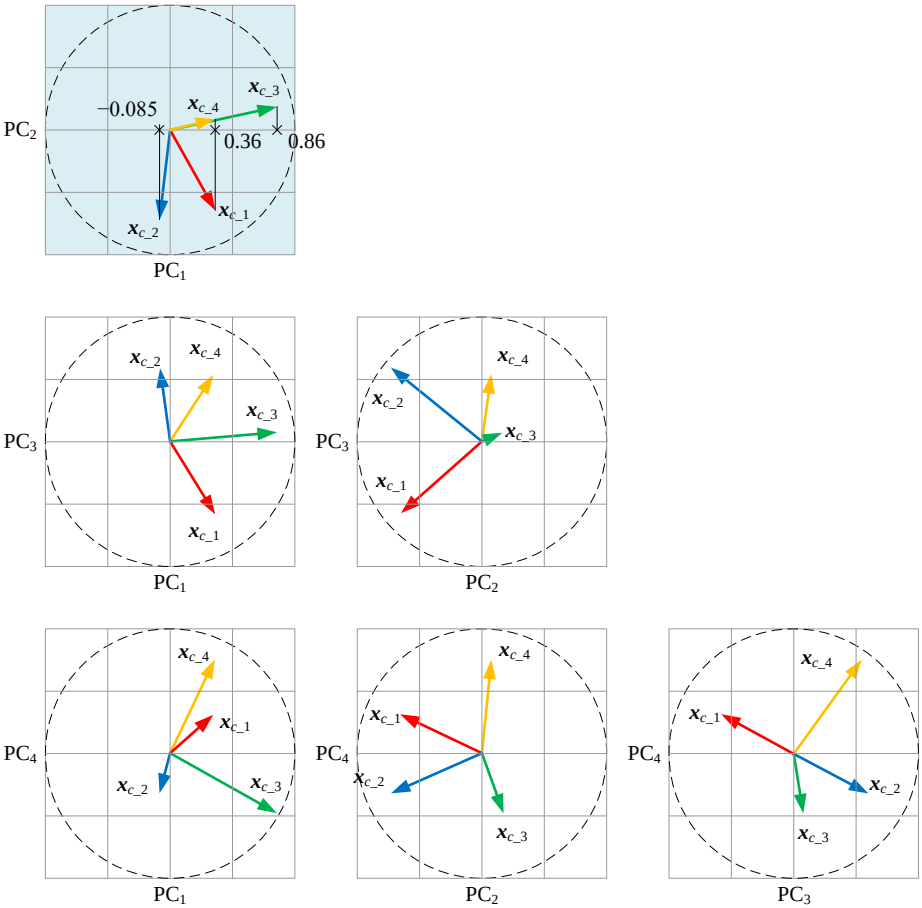


图 9. $\boldsymbol{V}_c$ 双标图，特征值分解协方差矩阵 $\boldsymbol{\Sigma}$

数据还原、误差

将 (11) 展开写成：

$$\boldsymbol{X}_c = \underbrace{\begin{bmatrix} \boldsymbol{u}_{c\_1} & \boldsymbol{u}_{c\_2} & \cdots & \boldsymbol{u}_{c\_D} \end{bmatrix}}_{\boldsymbol{U}_c} \underbrace{\begin{bmatrix} s_{c\_1} & & & \\ & s_{c\_2} & & \\ & & \ddots & \\ & & & s_{c\_D} \end{bmatrix}}_{\boldsymbol{S}_c} \underbrace{\begin{bmatrix} \boldsymbol{v}_{c\_1}^{\mathrm{T}} \\ \boldsymbol{v}_{c\_2}^{\mathrm{T}} \\ \vdots \\ \boldsymbol{v}_{c\_D}^{\mathrm{T}} \end{bmatrix}}_{\boldsymbol{V}_c^{\mathrm{T}}} \tag{23}$$

$$= s_{c\_1}\boldsymbol{u}_{c\_1}\boldsymbol{v}_{c\_1}^{\mathrm{T}} + s_{c\_2}\boldsymbol{u}_{c\_2}\boldsymbol{v}_{c\_2}^{\mathrm{T}} + \cdots + s_{c\_D}\boldsymbol{u}_{c\_D}\boldsymbol{v}_{c\_D}^{\mathrm{T}} = \sum_{j=1}^{D} s_{c\_j}\boldsymbol{u}_{c\_j}\boldsymbol{v}_{c\_j}^{\mathrm{T}}$$

图 10 所示为用第一主成分逼近估计 $\boldsymbol{X}_c$，即：

$$\hat{\boldsymbol{X}}_c = \underbrace{s_{c\_1}\boldsymbol{u}_{c\_1}\boldsymbol{v}_{c\_1}^{\mathrm{T}}}_{\text{First principal}} \tag{24}$$

图中可以看到，$\hat{\boldsymbol{X}}_c$ 和 $\boldsymbol{X}_c$ 非常相似；虽然 $\hat{\boldsymbol{X}}_c$ 是个 150×4 矩阵，$\hat{\boldsymbol{X}}_c$ 的秩还是 1。请大家回顾如何用张量积计算 $\hat{\boldsymbol{X}}_c$。图 10 中的 $\boldsymbol{E}$ 为误差，即 $\boldsymbol{E} = \boldsymbol{X}_c - \hat{\boldsymbol{X}}_c$。

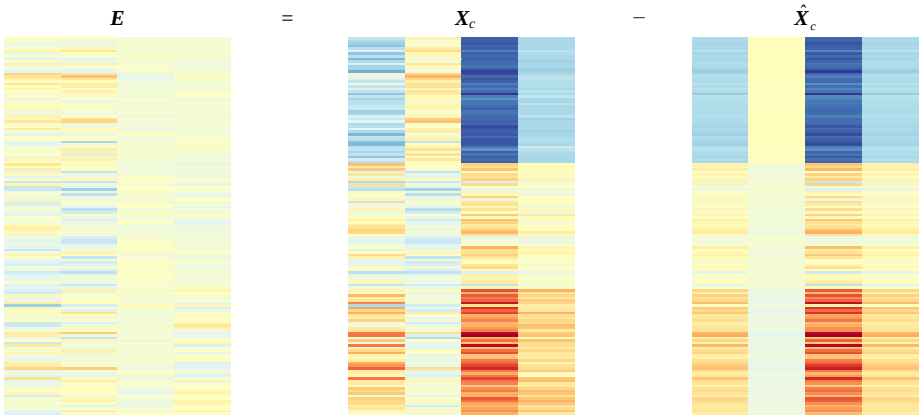


图 10. 第一主成分估计 $\boldsymbol{X}_c$

要想还原原始数据 $\boldsymbol{X}$，我们还需要考虑 (2) 这个等式关系，即：

$$\boldsymbol{X} = \boldsymbol{X}_c + \mathrm{E}(\boldsymbol{X}) = \sum_{j=1}^{D} s_{c\_j} \boldsymbol{u}_{c\_j} \boldsymbol{v}_{c\_j}^{\mathrm{T}} + \mathrm{E}(\boldsymbol{X}) \tag{25}$$

如果利用第一主成分估计原始数据矩阵 $\boldsymbol{X}$ 的话，可以利用：

$$\boldsymbol{X} \approx s_{c\_1} \boldsymbol{u}_{c\_1} \boldsymbol{v}_{c\_1}^{\mathrm{T}} + \mathrm{E}(\boldsymbol{X}) \tag{26}$$

上式中，$\mathrm{E}(\boldsymbol{X})$ 为行向量，计算用到了广播原则。

大家可能会问，图 2 中特征值分解仅仅获得了 $\boldsymbol{V}_c$，没有 $\boldsymbol{U}_c$。难道我们还需要再对 $\boldsymbol{X}_c$ 做 SVD 分解？答案是不需要。

《矩阵力量》第 10 章介绍过“二次投影”，也就是说 $\boldsymbol{X}_c$ 可以写成：

$$\boldsymbol{X}_c = \boldsymbol{X}_c \boldsymbol{I} = \boldsymbol{X}_c \boldsymbol{V}_c \boldsymbol{V}_c^{\mathrm{T}} \tag{27}$$

将 $\boldsymbol{V}_c$ 展开，上式可以写成：

$$\begin{aligned} \boldsymbol{X}_c &= \boldsymbol{X}_c \underbrace{\begin{bmatrix} \boldsymbol{v}_{c\_1} & \boldsymbol{v}_{c\_2} & \cdots & \boldsymbol{v}_{c\_D} \end{bmatrix}}_{\boldsymbol{V}_c} \underbrace{\begin{bmatrix} \boldsymbol{v}_{c\_1}^{\mathrm{T}} \\ \boldsymbol{v}_{c\_2}^{\mathrm{T}} \\ \vdots \\ \boldsymbol{v}_{c\_D}^{\mathrm{T}} \end{bmatrix}}_{\boldsymbol{V}_c^{\mathrm{T}}} \\ &= \boldsymbol{X}_c \boldsymbol{v}_{c\_1} \boldsymbol{v}_{c\_1}^{\mathrm{T}} + \boldsymbol{X}_c \boldsymbol{v}_{c\_2} \boldsymbol{v}_{c\_2}^{\mathrm{T}} + \cdots + \boldsymbol{X}_c \boldsymbol{v}_{c\_D} \boldsymbol{v}_{c\_D}^{\mathrm{T}} = \boldsymbol{X}_c \sum_{j=1}^{D} \boldsymbol{v}_{c\_j} \boldsymbol{v}_{c\_j}^{\mathrm{T}} \end{aligned} \tag{28}$$

所以，(24) 可以写成：

$$\hat{\boldsymbol{X}}_c = \boldsymbol{X}_c \boldsymbol{v}_{c\_1} \boldsymbol{v}_{c\_1}^{\mathrm{T}} = \boldsymbol{X}_c \boldsymbol{v}_{c\_1} \otimes \boldsymbol{v}_{c\_1} \tag{29}$$

(26) 则可以写成：

$$\boldsymbol{X} \approx \boldsymbol{X}_c \boldsymbol{v}_{c\_1} \otimes \boldsymbol{v}_{c\_1} + \mathrm{E}(\boldsymbol{X}) \tag{30}$$

如果用第一、二主成分还原 $\boldsymbol{X}$，上式需要再加一项：

$$\boldsymbol{X} \approx \underbrace{\boldsymbol{X}_c \boldsymbol{v}_{c\_1} \otimes \boldsymbol{v}_{c\_1}}_{\text{First principal}} + \underbrace{\boldsymbol{X}_c \boldsymbol{v}_{c\_2} \otimes \boldsymbol{v}_{c\_2}}_{\text{Second principal}} + \underbrace{\mathrm{E}(\boldsymbol{X})}_{\text{Centroid}} \tag{31}$$

鸢尾花书在不同位置反复强调数据单位，也就是量纲。如果原始数据的每列数据的量纲不一致，比如高度、质量、时间、温度、密度、百分比、股价、收益率、GDP 等等。利用特征值分解协方差矩阵完成 PCA 就会有麻烦，因为大家通过图 9 可以看到每一个主成分是若干特征的“线性融合”。哪怕每一列数据的量纲一致，比如鸢尾花前四列的单位都是厘米 cm，这种 PCA 技术路线还会受到不同特征方差大小影响。解决这些问题的方法是特征值分解线性相关系数矩阵，这是本章后文要讨论的话题。

椭圆：投影之前

如图 11 所示，协方差矩阵 $\boldsymbol{\Sigma}$ 椭球 (马氏距离为 1) 在六个平面上的投影。

通过旋转椭圆的形状、位置、旋转角度，我们可以读出标准差、相关性系数等重要信息。

图 12 比较数据 $\boldsymbol{X}$ 的分类和合并协方差矩阵对应的椭圆。

对椭圆、合并方差这些概念感到陌生的话，请回顾《统计至简》第 13 章。

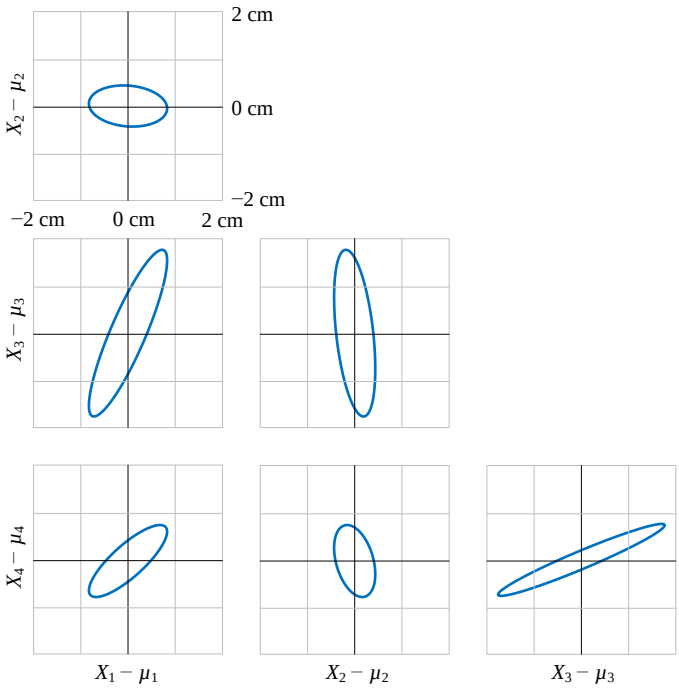


图 11. 马氏距离 1 椭圆，协方差矩阵 $\boldsymbol{\Sigma}$

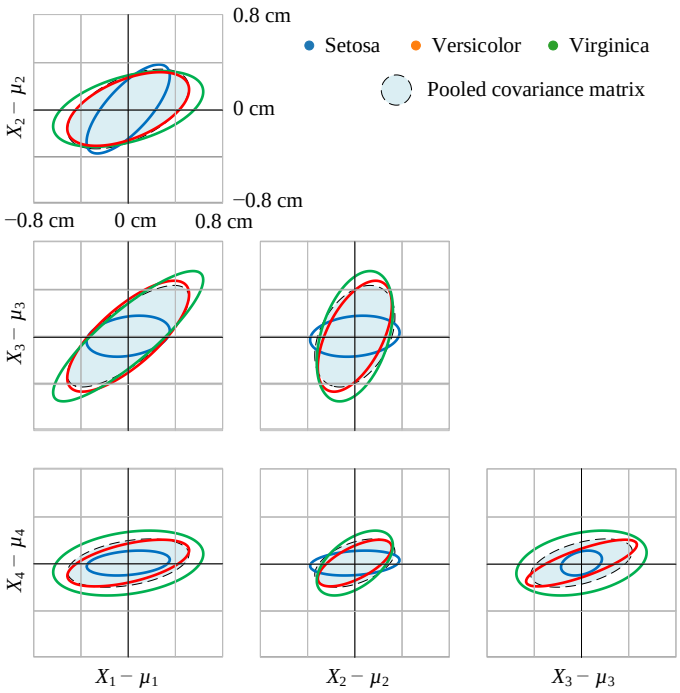


图 12. 马氏距离 1 椭圆，数据 $\boldsymbol{X}$ 的分类、合并协方差矩阵 $\boldsymbol{\Sigma}$

椭圆：投影之后

将中心化数据 $\boldsymbol{X}_c$ 投影到 $\boldsymbol{V}_c$ 得到的结果为 $\boldsymbol{Y}_c$：

$$\boldsymbol{Y}_c = \boldsymbol{X}_c \boldsymbol{V}_c \tag{32}$$

$\boldsymbol{Y}_c$ 的协方差矩阵就是 $\boldsymbol{X}$ 的协方差矩阵的特征值矩阵。

图 13 所示为 $\boldsymbol{Y}_c$ 的协方差矩阵在六个平面上的投影，这些椭圆都是正椭圆。$\boldsymbol{Y}_c$ 的协方差矩阵实际上就是 $\boldsymbol{\Sigma}$ 的特征值矩阵。

图 14 比较数据 $\boldsymbol{Y}_c$ 的分类和合并协方差矩阵对应的椭圆。

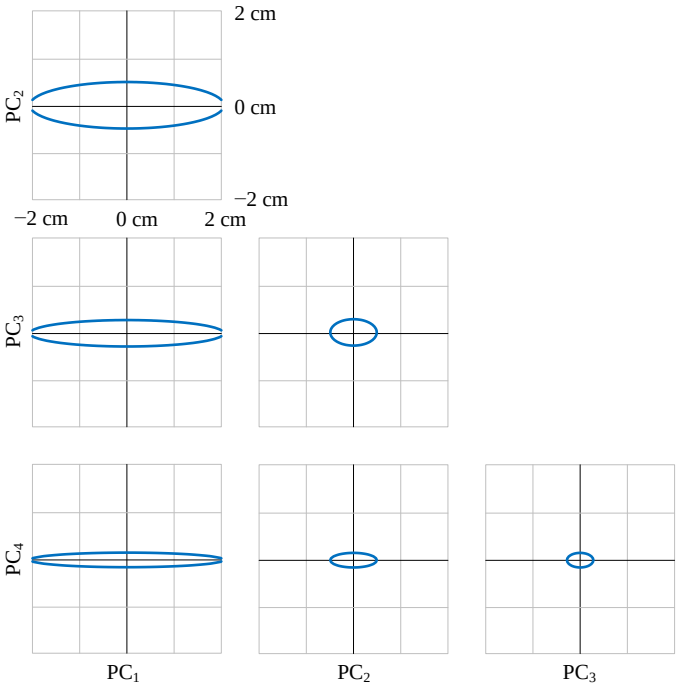


图 13. 马氏距离 1 椭圆，$\boldsymbol{Y}_c$ 的协方差矩阵

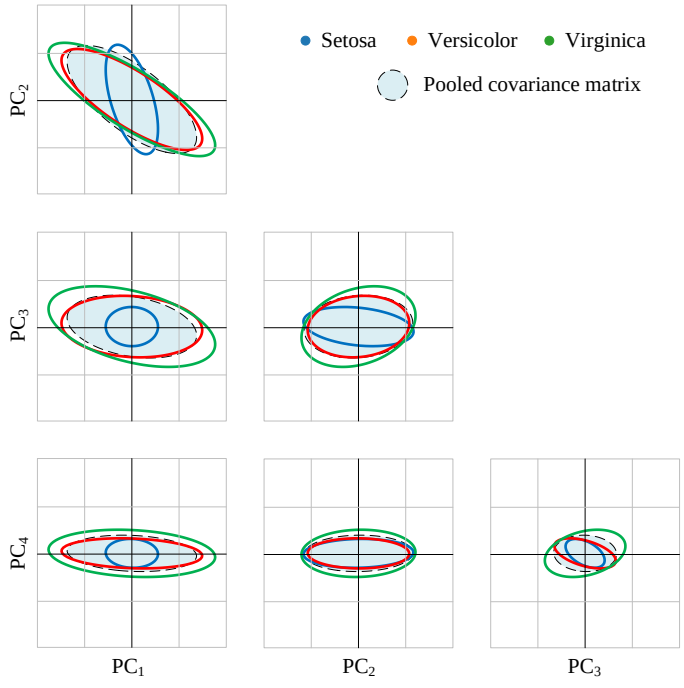


图 14. 马氏距离 1 椭圆，数据 $\boldsymbol{Y}_c$ 的分类、合并协方差矩阵 $\boldsymbol{\Sigma}$

15.3 格拉姆矩阵

特征值分解

图 15 所示为特征值分解格拉姆矩阵 $\boldsymbol{G}$。

注意，前文提过为了便于和协方差矩阵比较，本章中用的格拉姆矩阵 $\boldsymbol{G}$ 实际上是 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}/(n-1)$。

图 15 中的格拉姆矩阵 $\boldsymbol{G}$ 为对称矩阵，因此这个特征值分解同样是谱分解。

$\boldsymbol{V_X}$ 为正交矩阵，满足 $\boldsymbol{V_X}^{\mathrm{T}}\boldsymbol{V_X} = \boldsymbol{V_X}\boldsymbol{V_X}^{\mathrm{T}} = \boldsymbol{I}$。$\boldsymbol{\Lambda_X}$ 为对角矩阵，对角线元素为特征值，特征值从大到小排列。图 16 对比格拉姆矩阵 $\boldsymbol{G}$ 和 $\boldsymbol{\Lambda_X}$。下面，我们进一步分析这两个矩阵。

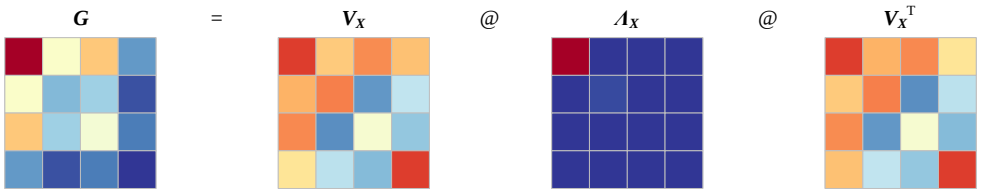


图 15. 特征值分解格拉姆矩阵 $\boldsymbol{G}$

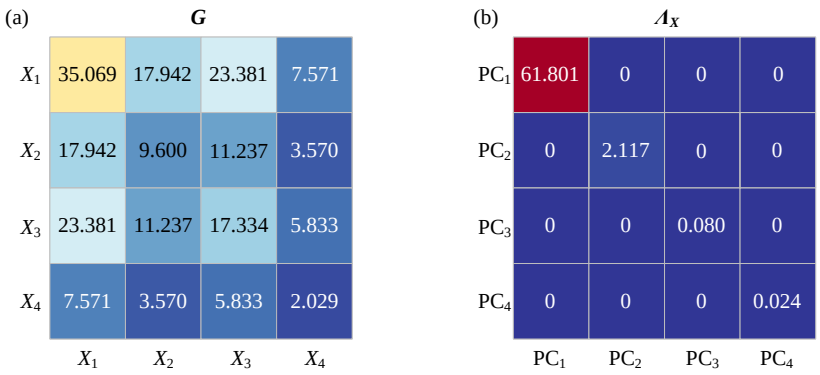


图 16. 对比 $\boldsymbol{G}$ 和 $\boldsymbol{\Lambda_X}$ 热图

分解前后

$\boldsymbol{G}$ 和 $\boldsymbol{\Lambda_X}$ 的主对角线之和相同，即 $\mathrm{trace}(\boldsymbol{G}) = \mathrm{trace}(\boldsymbol{\Lambda_X})$。如图 17 所示，矩阵 $\boldsymbol{G}$ 的主对角成分为矩阵 $\boldsymbol{X}$ 的每一列向量的模除以 $n-1$，代表某个特征相对于原点的分散情况，即“不去均值”的方差。

而 trace($\boldsymbol{G}$) 相当于数据整体相对于原点的分散度量。如图 17 所示，矩阵 $\boldsymbol{X}$ 的第一列和第二列贡献最大。经过特征值分解之后，如图 18 所示，第一主成分解释了大部分数据分散情况，占比高达 96.3%。

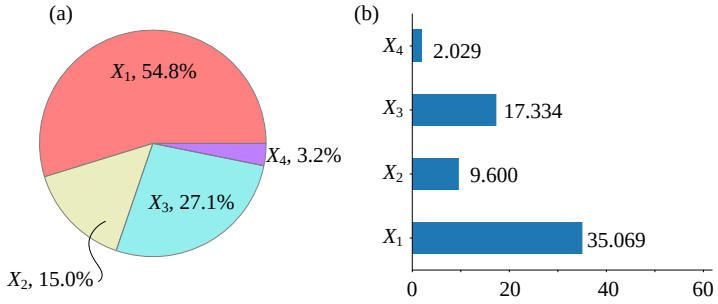


图 17. $\boldsymbol{G}$ 的主对角线成分

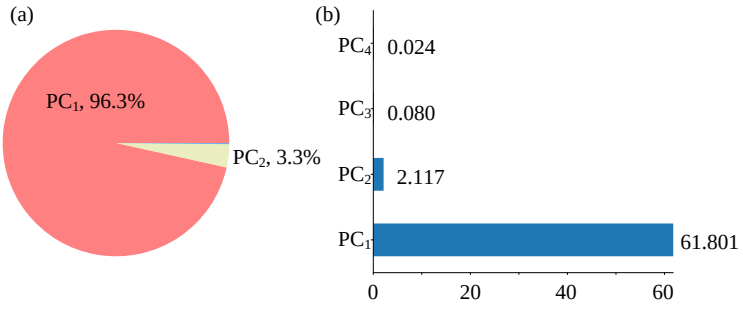


图 18. $\boldsymbol{\Lambda_X}$ 的主对角线成分，格拉姆矩阵 $\boldsymbol{G}$ 的特征值

陡坡图

图 19 所示为在特征值分解格拉姆矩阵 $\boldsymbol{G}$ 主成分分析的陡坡图。

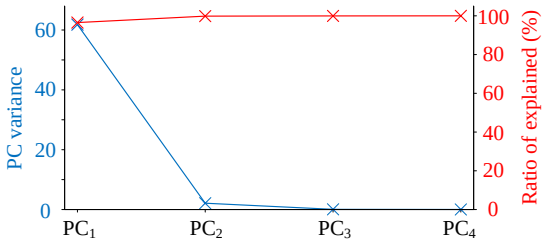


图 19. 陡坡图，特征值分解格拉姆矩阵 $\boldsymbol{G}$

特征向量矩阵

图 20 所示为特征向量矩阵 $\boldsymbol{V_X}$ 热图。显然，图 20 不同于图 7。

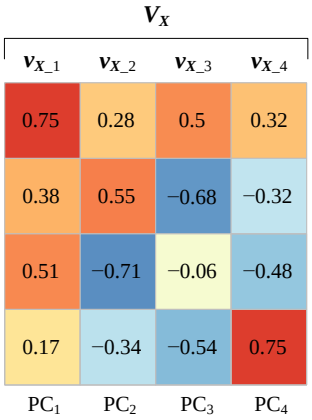


图 20. 特征向量矩阵 V_X 热图

投影

图 21 是将原始数据 X 投影到 V_X，即 $Y_X = XV_X$。Y_X 的特点是其格拉姆矩阵为对角方阵，也就是说 Y_X 的列向量两两正交。

注意，两两正交不代表线性无关。

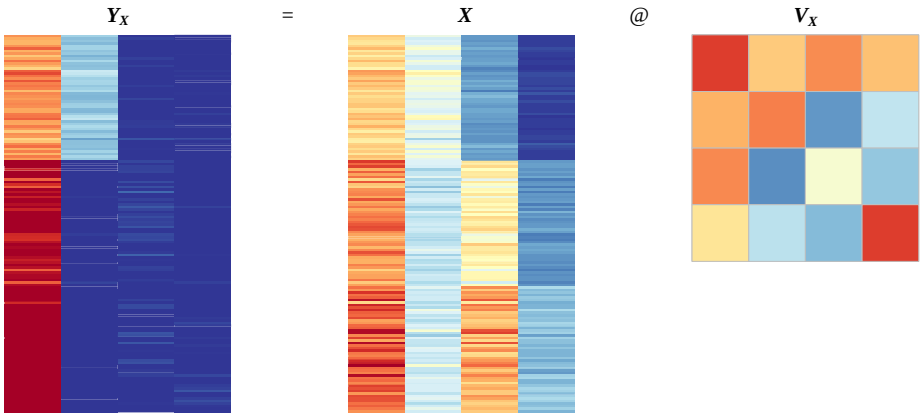


图 21. 将原始数据 X 投影到 V_X

正交矩阵 V_X 也是一个规范正交基，V_X 是因原始数据 X 而生。前文提到，V_c 同样是一个规范正交基，但是 V_c 是因中心化数据矩阵 X_c 而生。

我们当然可以将 X 投影到 V_c 这个规范正交基中，大家可以自行验证 XV_c 的协方差和 X_cV_c 相同，都是对角方阵。也就是说，XV_c 的列向量也是线性无关。但是，XV_c 的质心不再是原点。

双标图

图 22 所示为 V_X 的双标图。请大家自行比较图 9 和图 22。

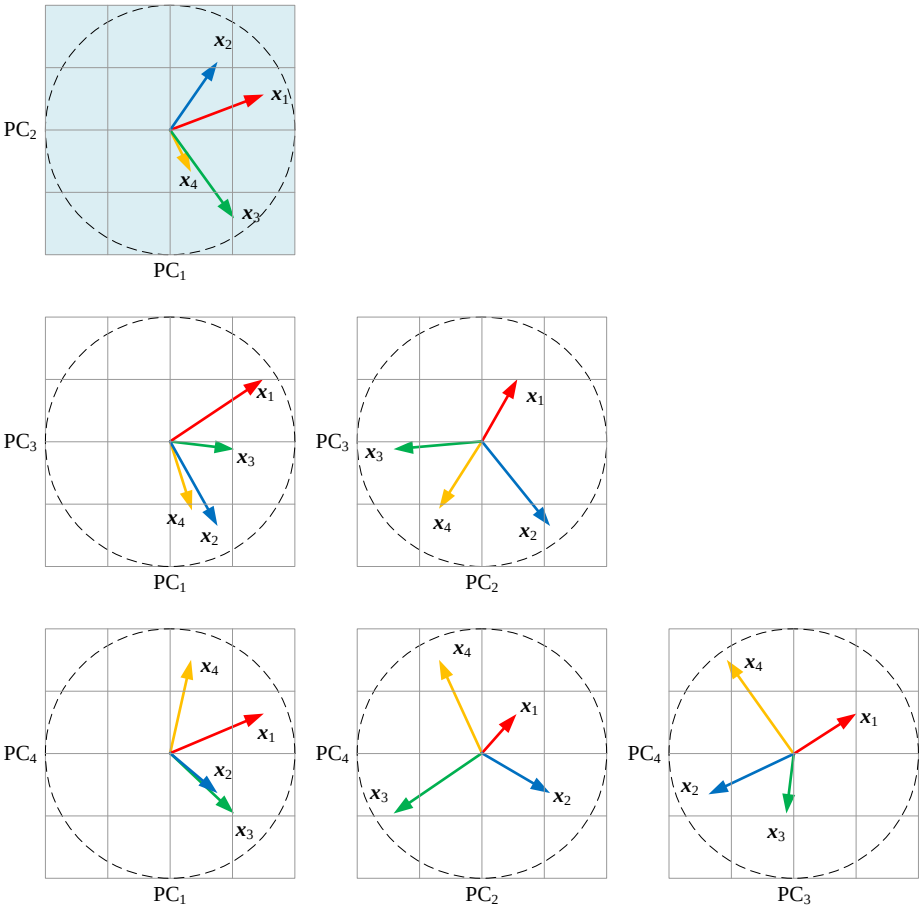


图 22. V_X 双标图，特征值分解格拉姆矩阵 G

数据还原、误差

由于本节中 PCA 分析直接采用特征值分解格拉姆矩阵 G，根据 (1)，利用第一主成分还原原始数据 X 时我们不需要加入质心成分：

$$X \approx X v_{X\_1} \otimes v_{X\_1} \tag{33}$$

如果用第一、二主成分还原 X，上式也需要再加一项：

$$X \approx \underbrace{X v_{X\_1} \otimes v_{X\_1}}_{\text{First principal}} + \underbrace{X v_{X\_2} \otimes v_{X\_2}}_{\text{Second principal}} \tag{34}$$

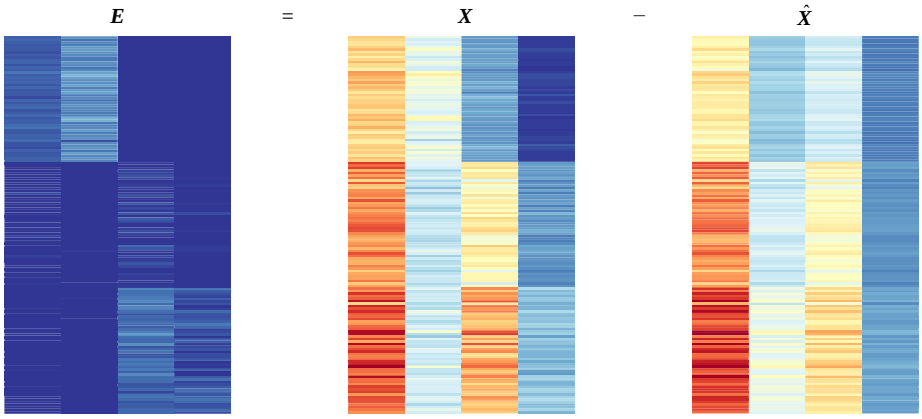


图 23. 第一主成分估计 $\boldsymbol{X}$

椭圆：投影之前

图 24 所示为格拉姆矩阵 $\boldsymbol{G}$ 对应的旋转椭圆。$\boldsymbol{G}$ 相当于“不去均值”的协方差矩阵。观察图 24，我们发现椭圆的朝向都是一三象限，而且椭圆都细长。比较图 11 和图 24，大家应该理解为什么需要去均值。

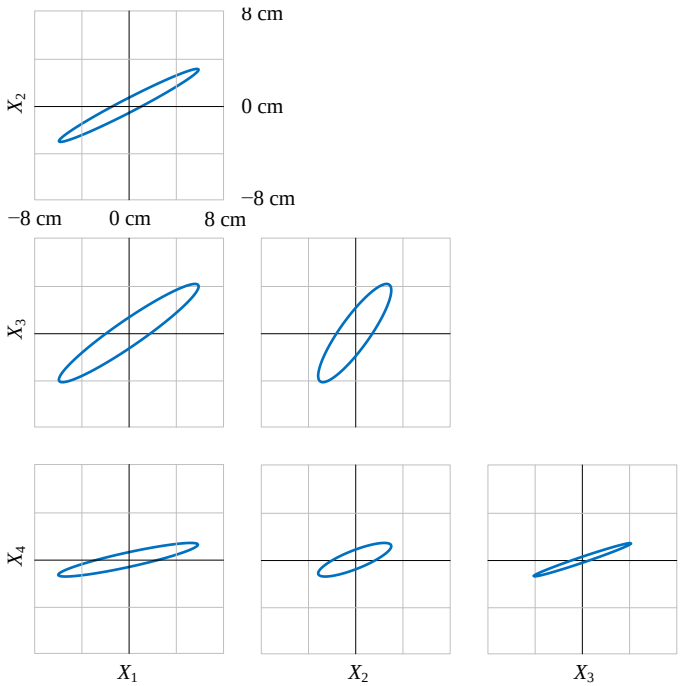


图 24. 马氏距离 1 椭圆，“不去均值”的协方差矩阵 $\boldsymbol{\Sigma}$

椭圆：投影之后

经过 $\boldsymbol{Y_X = XV_X}$ 投影之后，图 25 所示 $\boldsymbol{Y_X}$ 协方差矩阵对应的椭圆。

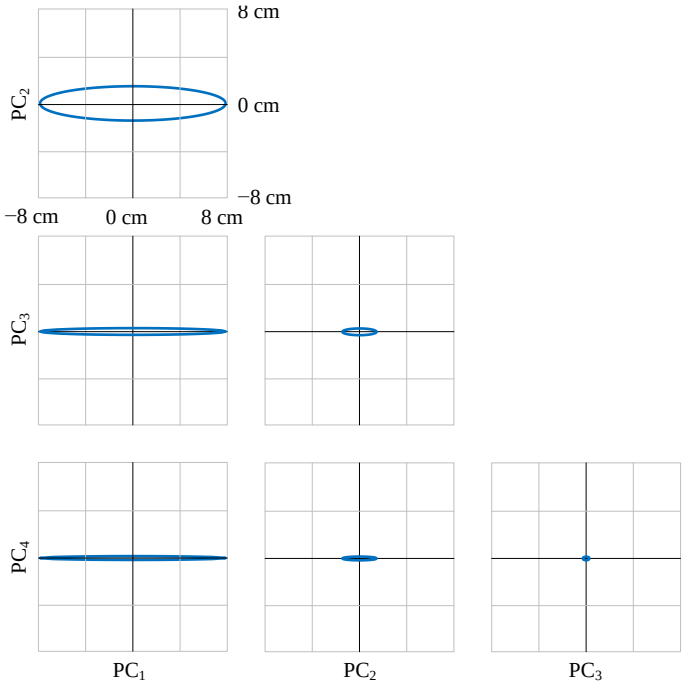


图 25. 马氏距离 1 椭圆，$\boldsymbol{Y_X}$ 的协方差矩阵

15.4 相关性系数矩阵

标准化数据 $\boldsymbol{Z_X}$ 相当于是 Z 分数，因此消除了特征量纲影响。因此，特征值分解相关系数矩阵不再受量纲影响。此外，标准化数据每一列特征数据均值均为 0，方差为 1。这也消除了较大方差特征的影响。

特征值分解

图 26 所示为特征值分解相关性系数矩阵 $\boldsymbol{P}$，$\boldsymbol{P}$ 的主对角线都是 1，$\boldsymbol{P}$ 对角线之外的元素都是线性相关系数。图 27 对比相关性系数矩阵 $\boldsymbol{P}$ 和 $\boldsymbol{\Lambda_Z}$ 热图。同样地，$\boldsymbol{P}$ 和 $\boldsymbol{\Lambda_Z}$ 主对角线之和相同，即 $\text{trace}(\boldsymbol{P}) = \text{trace}(\boldsymbol{\Lambda_Z})$。

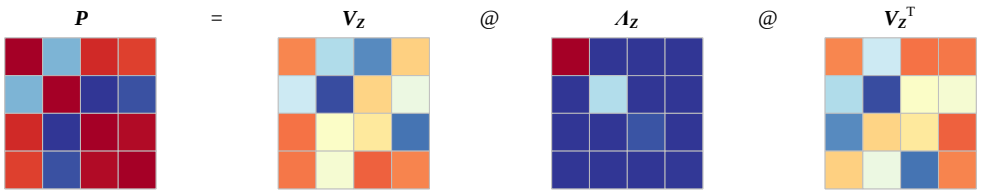


图 26. 特征值分解相关性系数矩阵 $\boldsymbol{P}$

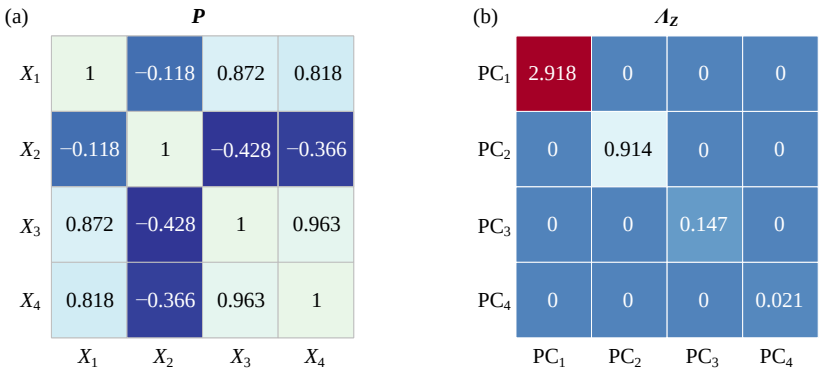


图 27. 对比相关性系数矩阵 $\boldsymbol{P}$ 和 $\boldsymbol{\Lambda_Z}$ 热图

分解前后

图 4 中，X_3 对方差和 trace($\boldsymbol{\Sigma}$) 贡献超过 68%，而 X_3 的贡献小于 5%。而图 28 中每个特征经过标准化之后，贡献率完全相同。方差小特征也可能含有重要的信息，利用特征值分解相关性系数完成 PCA，可以消除这种顾虑。

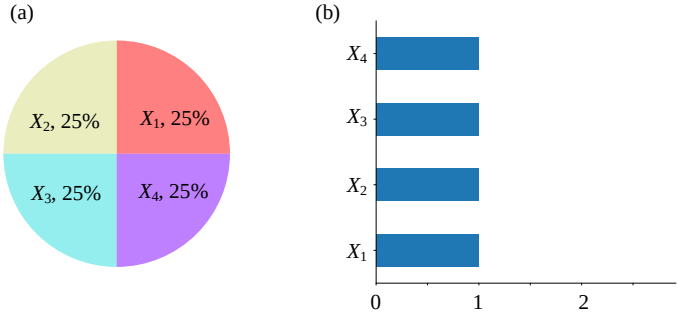


图 28. 相关性系数矩阵 $\boldsymbol{P}$ 主对角线成分

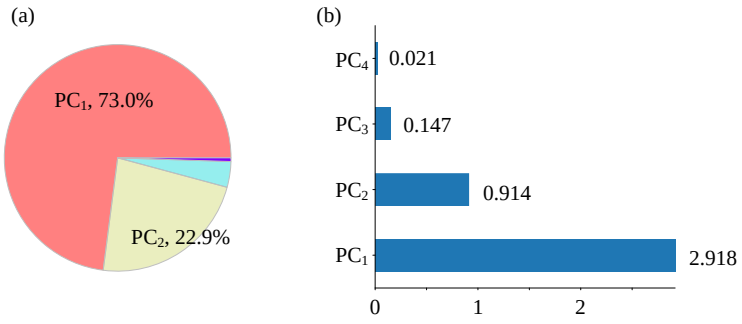


图 29. $\boldsymbol{\Lambda_Z}$ 的主对角线成分，相关性系数矩阵 $\boldsymbol{P}$ 特征值

陡坡图

图 30 所示为特征值分解相关性系数矩阵 $\boldsymbol{P}$ 主成分分析结果陡坡图。第一主成分贡献小于 80%。

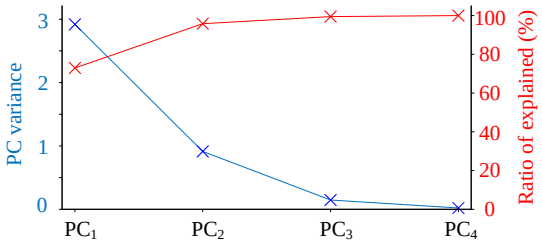


图 30. 陡坡图，特征值分解相关性系数矩阵 $\boldsymbol{P}$

特征向量矩阵

图 31 所示为特征向量矩阵 $\boldsymbol{V_Z}$ 热图。这幅图和图 7、图 20 均不同。

图 31. 特征向量矩阵 $\boldsymbol{V_Z}$ 热图

投影

图 32 所示为标准化数据 $\boldsymbol{Z}$ 投影到 $\boldsymbol{V_Z}$ 得到数据矩阵 $\boldsymbol{Y_Z}$。同样地，正交矩阵 $\boldsymbol{V_Z}$ 也是一个规范正交基，而 $\boldsymbol{V_Z}$ 是因中心化数据 $\boldsymbol{Z_X}$ 而生。

请大家将原数据 $\boldsymbol{X}$、中心化 $\boldsymbol{X_C}$ 也投影到 $\boldsymbol{V_Z}$ 中，并检验结果的协方差矩阵和质心。

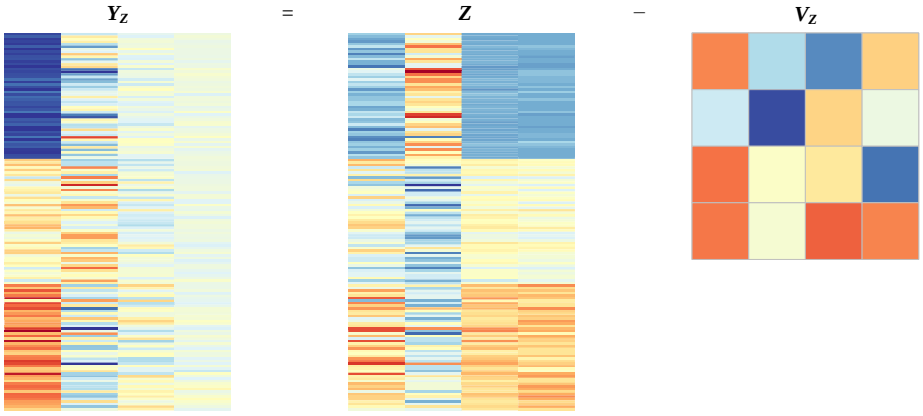


图 32. 中心化数据 Z 投影到 V_Z

双标图

图 33 所示为 V_Z 双标图，请大家比较本章三幅双标图。

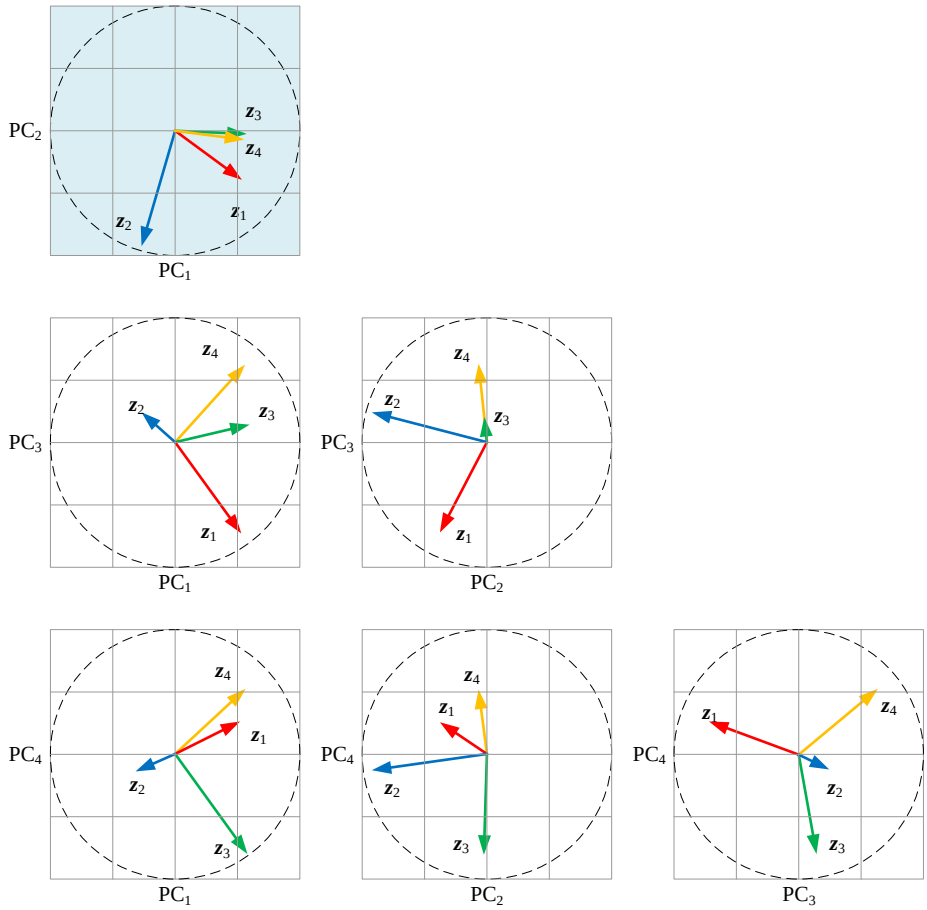


图 33. V_Z 双标图，特征值分解格相关性系数矩阵 P

数据还原、误差

图 34 所示为第一主成分估计 $\boldsymbol{Z}_{\boldsymbol{X}}$：

$$\boldsymbol{Z}_{\boldsymbol{X}} \approx \boldsymbol{Z}_{\boldsymbol{X}} \boldsymbol{v}_{\boldsymbol{X}\_1} \otimes \boldsymbol{v}_{\boldsymbol{X}\_1} \tag{35}$$

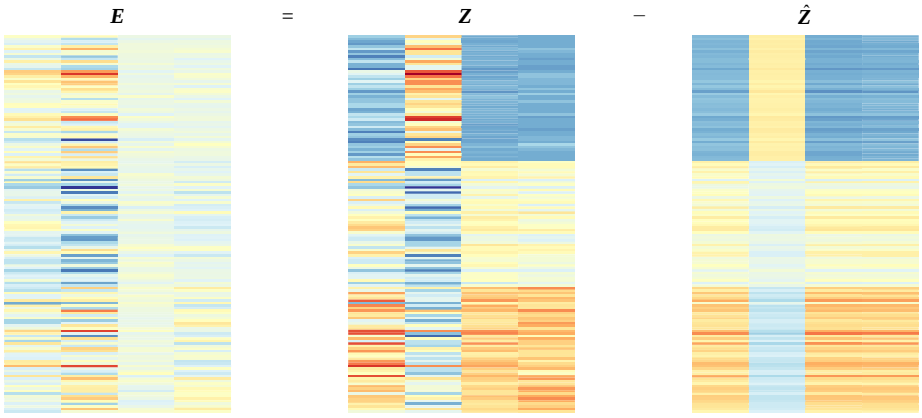


图 34. 第一主成分还原 $\boldsymbol{Z}_{\boldsymbol{X}}$

$\boldsymbol{Z}_{\boldsymbol{X}}$ 可以写成：

$$\boldsymbol{Z}_{\boldsymbol{X}} = \left(\boldsymbol{X} - \mathrm{E}\left(\boldsymbol{X}\right)\right)\boldsymbol{D}^{-1} = \sum_{j=1}^{D} \boldsymbol{Z}_{\boldsymbol{X}} \boldsymbol{v}_{\boldsymbol{X}\_j} \otimes \boldsymbol{v}_{\boldsymbol{X}\_j} \tag{36}$$

用 $\boldsymbol{V}_{\boldsymbol{Z}}$ 还原 $\boldsymbol{X}$：

$$\boldsymbol{X} = \left(\sum_{j=1}^{D} \boldsymbol{Z}_{\boldsymbol{X}} \boldsymbol{v}_{\boldsymbol{X}\_j} \otimes \boldsymbol{v}_{\boldsymbol{X}\_j}\right)\boldsymbol{D} + \mathrm{E}\left(\boldsymbol{X}\right) \tag{37}$$

用 $\boldsymbol{V}_{\boldsymbol{Z}}$ 第一主成分估计 $\boldsymbol{X}$：

$$\boldsymbol{X} \approx \underbrace{\left(\boldsymbol{Z}_{\boldsymbol{X}} \boldsymbol{v}_{\boldsymbol{X}\_1} \otimes \boldsymbol{v}_{\boldsymbol{X}\_1}\right)}_{\text{First principal}}\boldsymbol{D} + \mathrm{E}\left(\boldsymbol{X}\right) \tag{38}$$

其中，$\boldsymbol{D}$ 起到缩放的作用，$\mathrm{E}(\boldsymbol{X})$ 是平移的作用。

椭圆：投影之前

图 35 所示为投影之前相关性系数矩阵 $\boldsymbol{P}$ 对应的椭圆。请大家特别和前文协方差矩阵对应椭圆进行比较。

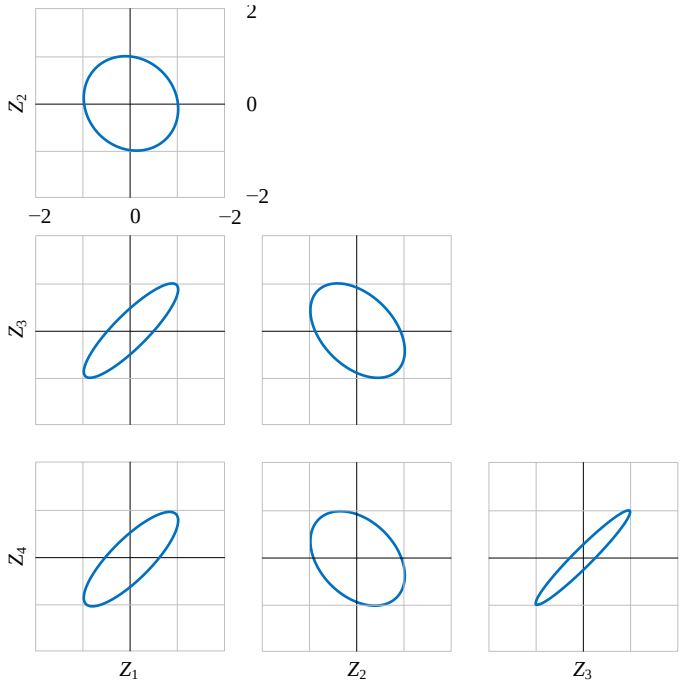


图 35. 马氏距离 1 椭圆，相关性系数矩阵 $\boldsymbol{P}$

椭圆：投影之后

图 36 所示为投影之后正椭圆的位置和形状。

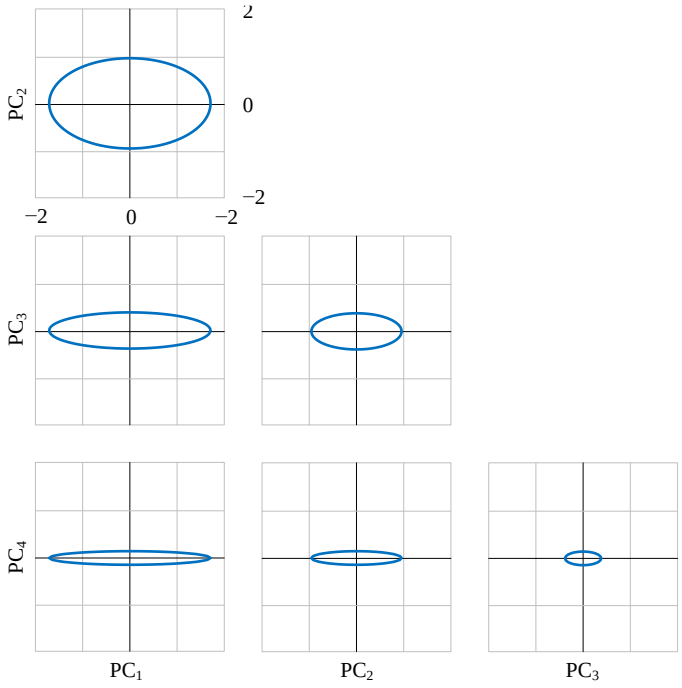
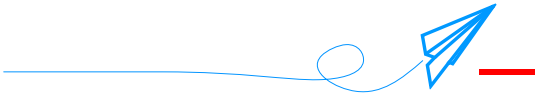


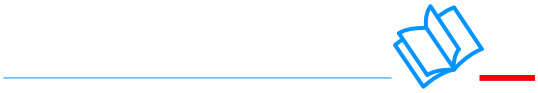
图 36. 马氏距离 1 椭圆，$\boldsymbol{Y_Z}$ 的协方差矩阵

Bk6_Ch15_01.py 绘制本章大部分图片。



主成分分析是鸢尾花书的“常客”，我们用椭圆、数据、格拉姆矩阵、协方差矩阵、特征值分解、奇异值分解、线性组合、优化、随机变量的线性函数等等视角探讨过主成分分析。换句话来说，机器学习常用的数学工具在主成分分析处达到了一种融合，大家也看到了数学板块实际上不是一个个孤立的个体，它们有其内在联系和网络。

鸢尾花书有关主成分分析专题内容到此为止，下两章我们将主要介绍和主成分分析相关的回归算法。此外，本书还会在最后一章比较奇异值分解和因子分析的异同。《机器学习》一册还要综述常见降维算法，其中还包括核主成分分析 KPCA，KPCA 相当于 PCA 的升级版。



在用椭圆理解数据、解释主成分分析方面，以下论文给本章很多启发，欢迎大家阅读：

https://arxiv.org/pdf/1302.4881.pdf

16 正交回归

Orthogonal Distance Regression

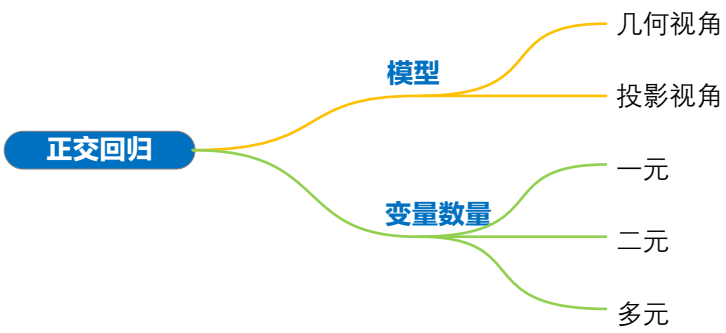
输入和输出数据都参与主成分分析，构造正交空间

数学展现出秩序、对称和有限——这些都是美的极致形态。

The mathematical sciences particularly exhibit order, symmetry, and limitations; and these are the greatest forms of the beautiful.

—— 亚里士多德 (Aristotle) | 古希腊哲学家 | 384 ~ 322 BC

- `numpy.linalg.eig()` 特征值分解
- `numpy.linalg.svd()` 奇异值分解
- `numpy.mean()` 计算均值
- `numpy.std()` 计算均方差
- `numpy.var()` 计算方差
- `pandas_datareader.get_data_yahoo()` 下载股价数据
- `scipy.odr` 正交回归
- `scipy.odr.Model()` 构造正交回归模型
- `scipy.odr.ODR()` 设置正交回归数据、模型和初始自
- `scipy.odr.RealData()` 加载正交回归数据
- `statsmodels.api.add_constant()` 增加 OLS 常数项
- `statsmodels.api.OLS` 最小二乘法线性回归



16.1 主成分与回归

本章主要介绍一种和主成分分析息息相关的回归方法——**正交回归** (orthogonal regression)。

正交回归，也叫做**正交距离回归** (Orthogonal Distance Regression, ODR)，又叫**全线性回归** (total linear regression)。正交回归通过将自变量通过主成分分析转换成互相正交的新变量，来消除自变量之间的多重共线性问题，从而提高回归分析的准确性和稳定性。

具体来说，正交回归通过以下步骤实现：1) 对自变量进行主成分分析，得到主成分变量，使它们互相正交。2) 对因变量和主成分变量进行回归分析，得到每个主成分变量的回归系数。3) 根据主成分变量的回归系数和主成分分析的结果，计算出每个自变量的回归系数和截距项。

正交回归的优点之一是消除自变量之间的多重共线性，提高回归分析的准确性和稳定性。正交回归可以在保证预测准确性的前提下，降低自变量的维度，提高回归模型的可解释性。

正交回归的缺点是计算复杂度较高，需要进行主成分分析和回归分析等多个步骤。此外，由于正交回归是基于主成分分析的，因此它可能会失去一些原始自变量的信息，因此需要在可接受的误差范围内进行权衡。

举个例子，平面上，最小二乘法线性回归 OLS 仅考虑纵坐标方向上误差，如图 1 (a) 所示；而正交回归 TLS 同时考虑横纵两个方向误差，如图 1 (b) 所示。

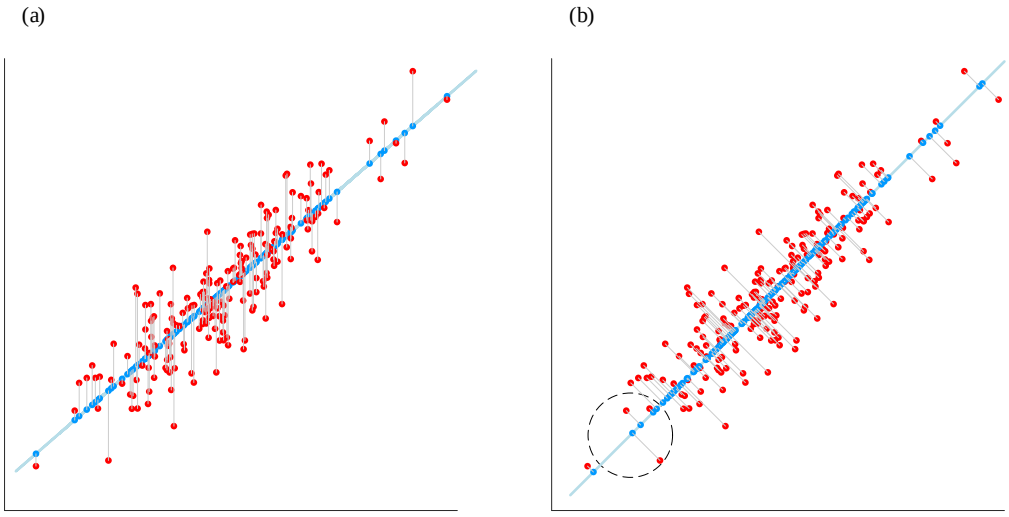


图 1. 对比 OLS 和 TLS 线性回归

从主成分分析角度，正交回归特点是输入数据 $\boldsymbol{X}$ 和输出数据 $\boldsymbol{y}$ 都参与主成分分析。按照特征值从小到大顺序排列特征向量 $[\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_D, \boldsymbol{v}_{D+1}]$，用其中前 D 个向量 $[\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_D]$ 构造一个全新超平面 H。利用 $\boldsymbol{v}_{D+1}$ 垂直于超平面 H 便可以求解出回归系数。

下面用两特征 $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2]$ 数据作例子，聊一下主成分回归的思想。如图 2 所示，$\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$ 为输入数据，$\boldsymbol{y}$ 为输出数据；通过主成分分析，$\boldsymbol{x}_1$、$\boldsymbol{x}_2$ 和 $\boldsymbol{y}$ 正交化之后得到 $\boldsymbol{v}_1$、$\boldsymbol{v}_2$ 和 $\boldsymbol{v}_3$ (根据特征值从小到大排列)；$\boldsymbol{v}_1$、$\boldsymbol{v}_2$ 和 $\boldsymbol{v}_3$ 两两正交。第一主成分 $\boldsymbol{v}_1$ 和第二主成分 $\boldsymbol{v}_2$ 构造平面 H。$\boldsymbol{v}_3$ 垂直于平面 H，通过这层关系求解出正交回归系数。

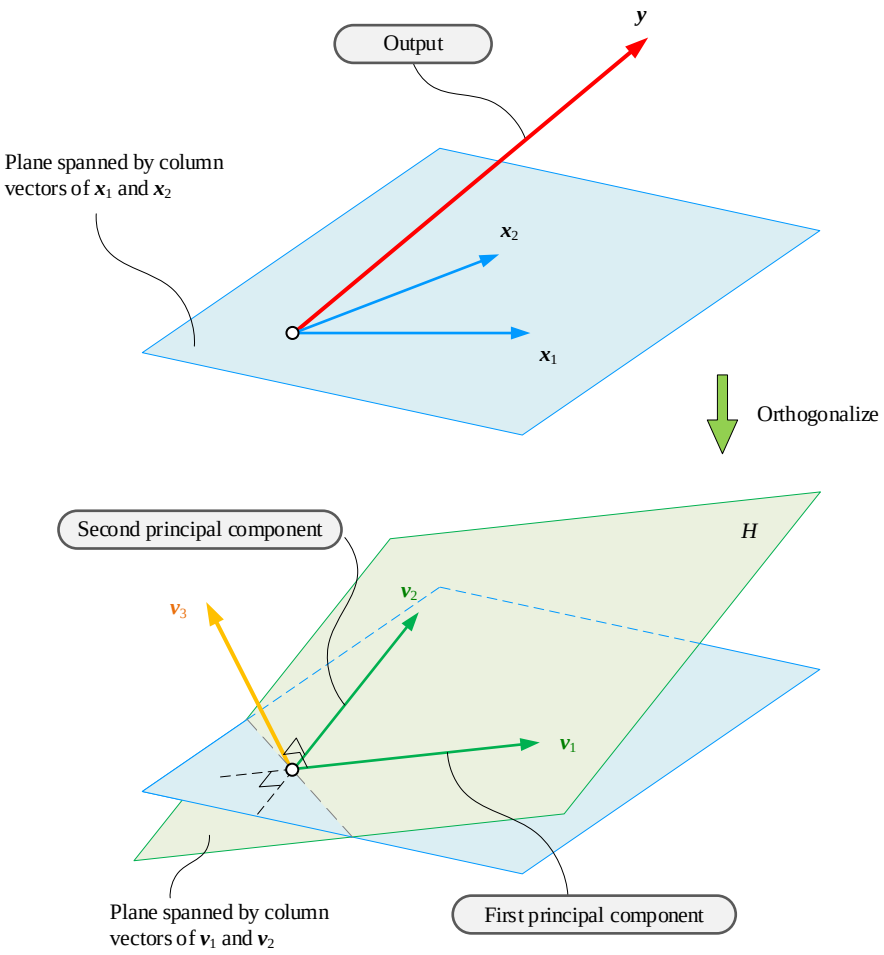


图 2. 通过主成分分析构造正交空间

前文介绍的线性回归采用算法叫做**普通最小二乘法** (Ordinary Least squares, OLS)；而正交回归采用的算法叫做**完全最小二乘法** (Total Least Squares, TLS)。

如图 3 所示，最小二乘回归，将 $\boldsymbol{y}$ 投影到 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$ 构造的平面上。而对于正交回归，将 $\boldsymbol{y}$ 投影到 H，得到 $\hat{\boldsymbol{y}}$。而残差，$\boldsymbol{\varepsilon} = \boldsymbol{y} - \hat{\boldsymbol{y}}$，平行于 $\boldsymbol{v}_3$。再次强调，平面 H 是由第一主成分 $\boldsymbol{v}_1$ 和第二主成分 $\boldsymbol{v}_2$ 构造。

此外，建议读者完成本章学习之后，回过头来再比较图 3 和图 4。这样，相信大家会更清楚 OLS 和 TLS 之间的区别。

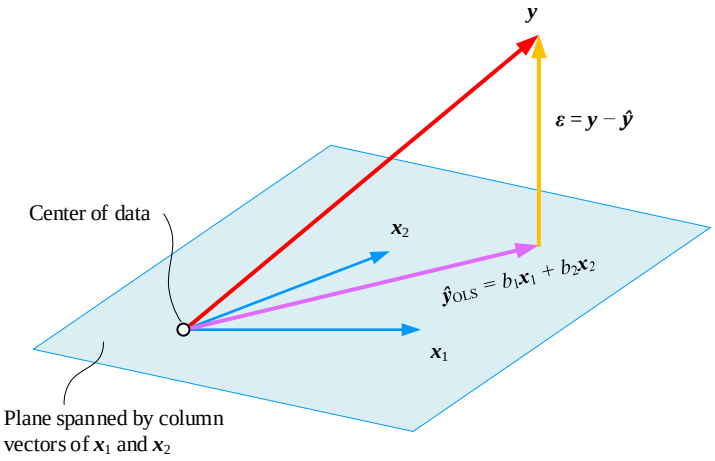


图 3. 最小二乘回归，将 $\boldsymbol{y}$ 投影到 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$ 构造的平面上

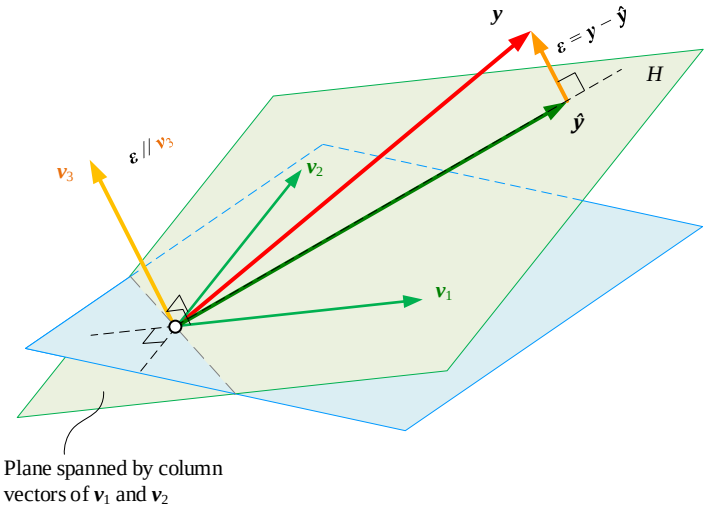


图 4. 正交回归，将输出数据 $\boldsymbol{y}$ 投影到 H

下一节首先用一元正交回归给大家建立正交回归的直观印象，本章后续将逐步扩展到二元回归和多元回归。

16.2 一元正交回归

设定一元正交回归解析式如下：

$$y = b_0 + b_1 x \tag{1}$$

其中，b_0 为截距项，b_1 为斜率。

如图 5 所示，x-y 平面上任意一点 $(x^{(i)}, y^{(i)})$ 和正交回归直线距离可以利用下式获得：

$$d_i = \frac{y^{(i)} - \left(b_0 + b_1 x^{(i)}\right)}{\sqrt{1 + b_1^2}} \tag{2}$$

当 $i = 1 \sim n$ 时，d_i 构成列向量为 $\boldsymbol{d}$：

$$\boldsymbol{d} = \frac{\boldsymbol{y} - \left(b_0 + b_1 \boldsymbol{x}\right)}{\sqrt{1 + b_1^2}} \tag{3}$$

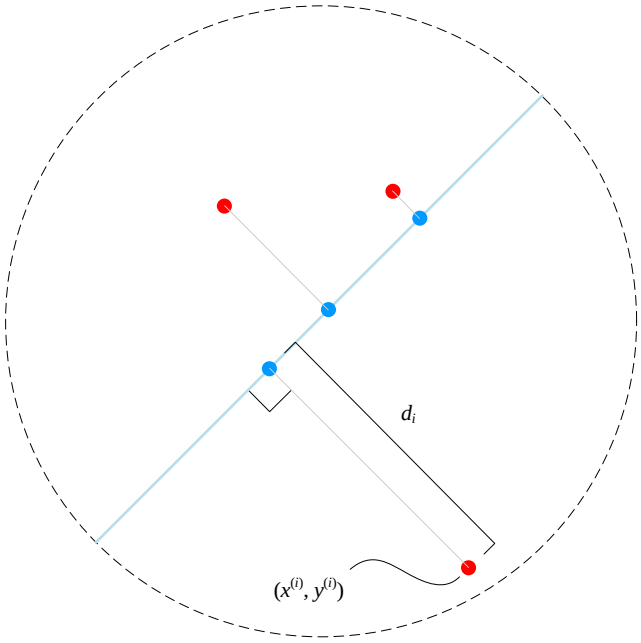


图 5. 正交投影几何关系

构造如下优化问题，b_0 和 b_1 为优化变量，优化目标为最小化欧氏距离平方和：

$$\underset{b0,b1}{\arg\min}\ f\left(b_0, b_1\right) = \left\|\boldsymbol{d}\right\|^2 = \boldsymbol{d}^{\mathrm{T}}\boldsymbol{d} \tag{4}$$

将 (3) 代入 $f(b_0, b_1)$ 得到：

$$f\left(b_0, b_1\right) = \frac{\left(\boldsymbol{y} - \left(b_0 + b_1 \boldsymbol{x}\right)\right)^{\mathrm{T}} \left(\boldsymbol{y} - \left(b_0 + b_1 \boldsymbol{x}\right)\right)}{1 + b_1^2} \tag{5}$$

为了方便计算，也引入全 1 向量 $\boldsymbol{1}$，它和 $\boldsymbol{x}$ 形状一样为 n 行 1 列向量；$f(b_0, b_1)$ 展开整理为下式：

$$f\left(b_0,b_1\right)=\frac{nb_0^2+2b_0b_1\boldsymbol{x}^{\mathrm{T}}\boldsymbol{1}+b_1^2\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}-2b_0\boldsymbol{y}^{\mathrm{T}}\boldsymbol{1}-2b_1\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}+\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}}{1+b_1^2} \tag{6}$$

$f(b_0, b_1)$ 对 b_0 偏导为 0，构造如下等式：

$$\frac{\partial f\left(b_0,b_1\right)}{\partial b_0}=\frac{2nb_0+2b_1\boldsymbol{x}^{\mathrm{T}}\boldsymbol{1}-2\boldsymbol{y}^{\mathrm{T}}\boldsymbol{1}}{1+b_1^2}=0 \tag{7}$$

$f(b_0, b_1)$ 对 b_1 偏导为 0，构造如下等式：

$$\frac{\partial f\left(b_0,b_1\right)}{\partial b_1}=\frac{2b_1\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}+2b_0\boldsymbol{x}^{\mathrm{T}}\boldsymbol{1}-2\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}}{1+b_1^2}-\frac{\left(nb_0^2+2b_0b_1\boldsymbol{x}^{\mathrm{T}}\boldsymbol{1}+b_1^2\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}-2b_0\boldsymbol{y}^{\mathrm{T}}\boldsymbol{1}-2b_1\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}+\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}\right)2b_1}{\left(1+b_1^2\right)^2}=0 \tag{8}$$

观察 (7)，容易用 b_1 表达 b_0：

$$b_0=\frac{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{1}-b_1\boldsymbol{x}^{\mathrm{T}}\boldsymbol{1}}{n}=\mathrm{E}\left(\boldsymbol{y}\right)-b_1\mathrm{E}\left(\boldsymbol{x}\right) \tag{9}$$

其中，

$$\begin{cases}\mathrm{E}\left(\boldsymbol{x}\right)=\dfrac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{1}}{n}=\dfrac{\sum_{i=1}^{n}x^{(i)}}{n}\\ \mathrm{E}\left(\boldsymbol{y}\right)=\dfrac{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{1}}{n}=\dfrac{\sum_{i=1}^{n}y^{(i)}}{n}\end{cases} \tag{10}$$

将 (9) 给出 b_0 解析式代入 (8) 获得仅含有 b_1 的一元二次方程：

$$b_1^2+kb_1-1=0 \tag{11}$$

其中，

$$\begin{aligned}k&=\frac{n\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{l}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{l}-n\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}+\boldsymbol{y}^{\mathrm{T}}\boldsymbol{l}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{l}}{n\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{l}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{l}}\\&=\frac{\left(\dfrac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}{n}-\dfrac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{l}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{l}}{n^2}\right)-\left(\dfrac{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}}{n}-\dfrac{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{l}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{l}}{n^2}\right)}{\dfrac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}}{n}-\dfrac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{l}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{l}}{n^2}}\\&=\frac{\mathrm{var}\left(\boldsymbol{x}\right)-\mathrm{var}\left(\boldsymbol{y}\right)}{\mathrm{cov}\left(\boldsymbol{x},\boldsymbol{y}\right)}=\frac{\sigma_x^2-\sigma_y^2}{\rho_{xy}\sigma_x\sigma_y}\end{aligned} \tag{12}$$

上式，不区分求解方差协方差时，$1/(n-1)$和 $1/n$ 之间差别。

求解 (11) 一元二次方程，得到 b_1 解如下：

$$b_1=\frac{-k\pm\sqrt{k^2+4}}{2} \tag{13}$$

将 (12) 给出的 k，代入 (13)，整理得到 b_1 解：

$$b_1 = \frac{\left(\sigma_y^2 - \sigma_x^2\right) \pm \sqrt{\left(\sigma_x^2 - \sigma_y^2\right)^2 + 4\left(\rho_{xy}\sigma_x\sigma_y\right)^2}}{2\rho_{xy}\sigma_x\sigma_y} \tag{14}$$

发现 b_1 两个解即**主成分分析** (principal component analysis, PCA) 主元方向。

构造 $[\boldsymbol{x}, \boldsymbol{y}]$ 数据矩阵，它的协方差矩阵 $\boldsymbol{\Sigma}$ 可以记做：

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_x^2 & \rho_{xy}\sigma_x\sigma_y \\ \rho_{xy}\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix} \tag{15}$$

对 $\boldsymbol{\Sigma}$ 进行特征值分解，得到两个特征向量：

$$\begin{aligned} \boldsymbol{v}_1 &= \begin{bmatrix} \dfrac{\left(\sigma_y^2 - \sigma_x^2\right) + \sqrt{\left(\sigma_x^2 - \sigma_y^2\right)^2 + 4\left(\rho_{xy}\sigma_x\sigma_y\right)^2}}{2\rho_{xy}\sigma_x\sigma_y} \\ 1 \end{bmatrix} \\ \boldsymbol{v}_2 &= \begin{bmatrix} \dfrac{\left(\sigma_y^2 - \sigma_x^2\right) - \sqrt{\left(\sigma_x^2 - \sigma_y^2\right)^2 + 4\left(\rho_{xy}\sigma_x\sigma_y\right)^2}}{2\rho_{xy}\sigma_x\sigma_y} \\ 1 \end{bmatrix} \end{aligned} \tag{16}$$

$\boldsymbol{\Sigma}$ 两个特征值，从大到小排列：

$$\begin{aligned} \lambda_1 &= \frac{\sigma_x^2 + \sigma_y^2}{2} + \sqrt{\left(\rho_{xy}\sigma_x\sigma_y\right)^2 + \left(\frac{\sigma_x^2 - \sigma_y^2}{2}\right)^2} \\ \lambda_2 &= \frac{\sigma_x^2 + \sigma_y^2}{2} - \sqrt{\left(\rho_{xy}\sigma_x\sigma_y\right)^2 + \left(\frac{\sigma_x^2 - \sigma_y^2}{2}\right)^2} \end{aligned} \tag{17}$$

特征值较大的特征向量为正交回归直线切线向量；特征值较小特征向量对应直线法线向量，这样求得 b_1 斜率。有了上述思路，便可以用 PCA 分解来获得正交回归系数，这是下一节要讲解的内容。

如下代码首先介绍如何利用 scipy.odr 可以求解得到正交回归系数。构造线性函数 linear_func(b, x)，利用 scipy.odr.Model(linear_func) 创建线性模型；然后，采用 scipy.odr.RealData() 加载数据，再用 scipy.odr.ODR() 整合数据、模型和初始值，输出为 odr。odr.run() 求解回归问题。然后，用 pprint()打印结果。

```
Beta: [0.00157414 1.43773257]
Beta Std Error: [0.00112548 0.05617699]
Beta Covariance: [[ 1.21904872e-02 -2.43641786e-02]
 [-2.43641786e-02  3.03712371e+01]]
Residual Variance: 0.00010390932459480641
Inverse Condition #: 0.22899877744275976
Reason(s) for Halting:
Sum of squares convergence
```

一元正交回归的解析式为：

$$y = 1.4377x + 0.00157 \tag{18}$$

下一节将介绍如下采用主成分分析来求解一元正交回归系数，并比较正交回归和最小二乘法线性回归。

16.3 几何角度看正交回归

图 6 所示为正交回归和 PCA 分解关系，发现主元回归直线通过数据中心 (E($\boldsymbol{x}$), E($\boldsymbol{y}$))，回归直线方向和主元方向 $\boldsymbol{v}_1$ 平行，垂直于次元 $\boldsymbol{v}_2$ 方向。即，次元方向 $\boldsymbol{v}_2$ 和直线法向量 $\boldsymbol{n}$ 平行。

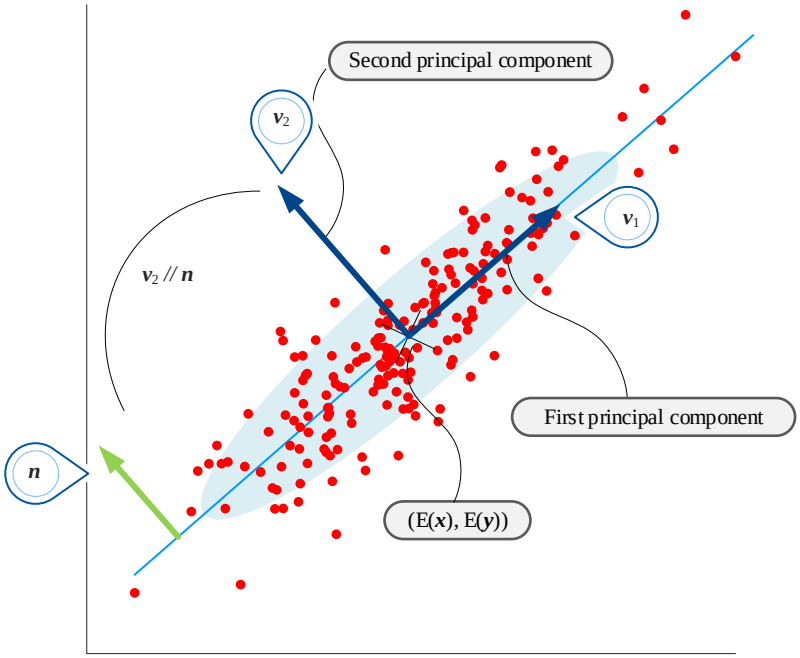


图 6. 正交回归和 PCA 分解关系

对于 (1) 所示一元一次函数，构造二元 $F(x, y)$ 函数如下：

$$F\left(x,y\right)=b_0+b_1x-y \tag{19}$$

$F(x, y)$ 法向量，即平面上形如 (1) 直线法向量 $\boldsymbol{n}$ 可以通过下式求解：

$$\boldsymbol{n}=\left(\frac{\partial F}{\partial x},\frac{\partial F}{\partial y}\right)^{\mathrm{T}}=\begin{bmatrix}b_1\\-1\end{bmatrix} \tag{20}$$

如前文所示，$\boldsymbol{n}$ 方向即 PCA 分解第二主元方向，即次元方向。

为了方便计算，假设数据已经经过中心化处理，即已经完成如下运算：

$$\boldsymbol{x}=\boldsymbol{x}-\mathrm{E}\left(\boldsymbol{x}\right),\quad \boldsymbol{y}=\boldsymbol{y}-\mathrm{E}\left(\boldsymbol{y}\right) \tag{21}$$

由于 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 已经是中心化向量，协方差矩阵可以通过下式运算得到：

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{x} & \boldsymbol{y} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{x} & \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}^{\mathrm{T}} \\ \boldsymbol{y}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} & \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} & \boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} \\ \boldsymbol{y}^{\mathrm{T}}\boldsymbol{x} & \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} \end{bmatrix} \tag{22}$$

为了方便计算，本节计算协方差矩阵不考虑系数 $1/(n-1)$。

由于 $\boldsymbol{n}$ 为 $\boldsymbol{\Sigma}$ 次元方向：

$$\boldsymbol{\Sigma}\boldsymbol{n} = \lambda_2 \boldsymbol{n} \quad \Rightarrow \quad \begin{bmatrix} \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} & \boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} \\ \boldsymbol{y}^{\mathrm{T}}\boldsymbol{x} & \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} \end{bmatrix} \boldsymbol{n} = \lambda_2 \boldsymbol{n} \tag{23}$$

将 (20) 代入 (23)，整理得到如下两个等式：

$$\begin{bmatrix} \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} & \boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} \\ \boldsymbol{y}^{\mathrm{T}}\boldsymbol{x} & \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} \end{bmatrix} \begin{bmatrix} b_1 \\ -1 \end{bmatrix} = \lambda_2 \begin{bmatrix} b_1 \\ -1 \end{bmatrix} \quad \Rightarrow \quad \begin{cases} \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}b_1 - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} = \lambda_2 b_1 \\ \boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}b_1 - \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} = -\lambda_2 \end{cases} \tag{24}$$

联立 (24) 两个等式，用 λ_2 表示 b_1：

$$b_{1\_\mathrm{TLS}} = \left(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} - \lambda_2\right)^{-1} \boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} \tag{25}$$

下式为本书前文获得的一元线性回归 OLS 中 b_1 解：

$$b_{1\_\mathrm{OLS}} = \left(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}\right)^{-1} \boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} \tag{26}$$

对比 OLS 和 TLS；当 (25) 中 λ_2 为 0 时，两种回归方法得到斜率完全一致。$\lambda_2 = 0$ 时，$\boldsymbol{y}$ 和 $\boldsymbol{x}$ 完全线性相关。

数据中心化前后，回归直线梯度向量不变；中心化之前的回归直线通过 $(\mathrm{E}(\boldsymbol{x}), \mathrm{E}(\boldsymbol{y}))$ 一点，即：

$$\mathrm{E}\left(\boldsymbol{y}\right) = b_0 + b_1 \mathrm{E}\left(\boldsymbol{x}\right) \tag{27}$$

获得回归式截距项 b_0 表达式：

$$b_0 = \mathrm{E}\left(\boldsymbol{y}\right) - b_1 \mathrm{E}\left(\boldsymbol{x}\right) \tag{28}$$

图 7 所示为一元正交回归数据之间关系。发现自变量 $\boldsymbol{x}$ 列向量和因变量 $\boldsymbol{y}$ 列向量数据都参与 PCA 分解得到正交化向量 $\boldsymbol{v}_1$ 和 $\boldsymbol{v}_2$，然后用特征值中较大值对应特征向量 $\boldsymbol{v}_1$ 作为一元正交回归直线切线向量。更为简单计算方法是，用特征值较小值对应特征向量 $\boldsymbol{v}_2$ 作为一元正交回归直线法向量。

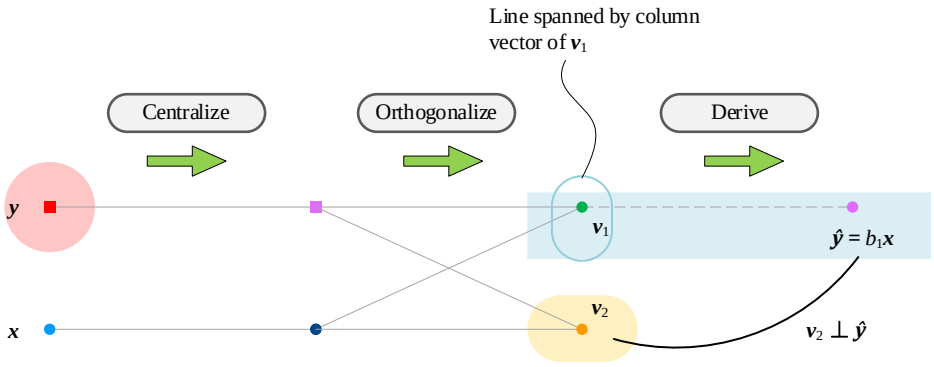


图 7. 一元正交回归 TLS 数据关系

图 8 所示为最小二乘法 OLS 一元线性回归系数，对应的一元 OLS 解析式为：

$$y = 1.1225x + 0.0018 \tag{29}$$

图 9 比较 OLS 和 TLS 结果。

OLS Regression Results

Dep. Variable:	AAPL	R-squared:	0.687
Model:	OLS	Adj. R-squared:	0.686
Method:	Least Squares	F-statistic:	549.7
Date:	Thu, 07 Oct 2021	Prob (F-statistic):	4.55e-65
Time:	07:08:46	Log-Likelihood:	678.03
No. Observations:	252	AIC:	-1352.
Df Residuals:	250	BIC:	-1345.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.0018	0.001	1.759	0.080	-0.000	0.004
SP500	1.1225	0.048	23.446	0.000	1.028	1.217

Omnibus:	52.424	Durbin-Watson:	1.864
Prob(Omnibus):	0.000	Jarque-Bera (JB):	210.804
Skew:	0.777	Prob(JB):	1.68e-46
Kurtosis:	7.203	Cond. No.	46.1

图 8. 最小二乘法 OLS 一元线性回归结果

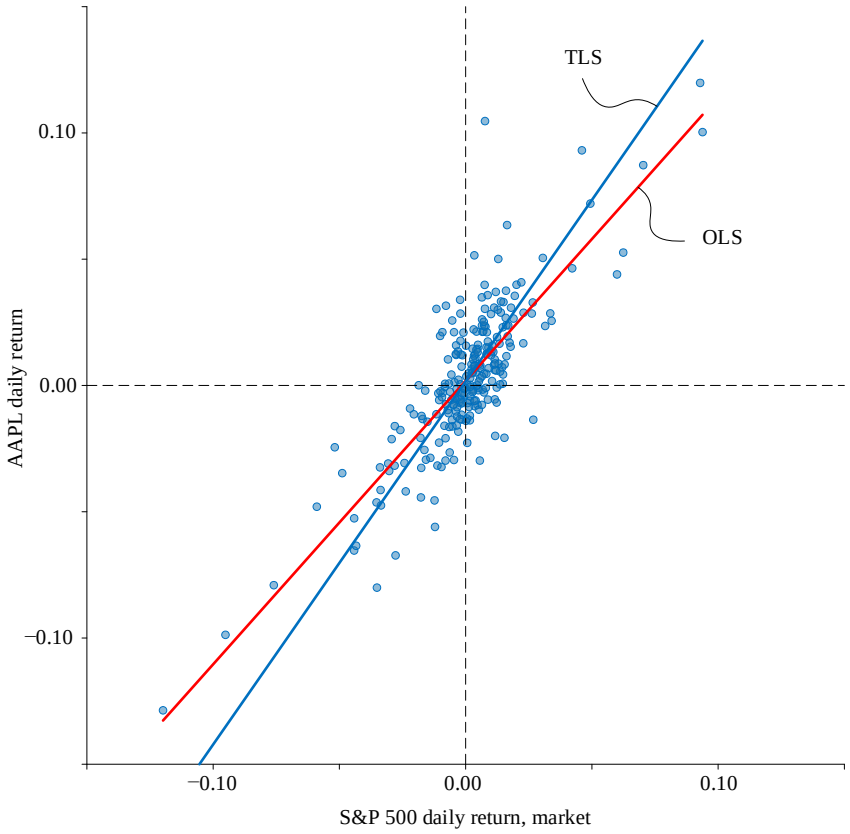


图 9. 比较 OLS 和 TLS 结果

Bk6_Ch16_01.py 绘制本节图像。

16.4 二元正交回归

这一节用主成分分析讨论二元正交回归。

首先也是对数据进行中心化处理：

$$\boldsymbol{x}_1 = \boldsymbol{x}_1 - \mathrm{E}(\boldsymbol{x}_1), \quad \boldsymbol{x}_2 = \boldsymbol{x}_2 - \mathrm{E}(\boldsymbol{x}_1), \quad \boldsymbol{y} = \boldsymbol{y} - \mathrm{E}(\boldsymbol{y}) \tag{30}$$

根据 PCA 计算法则，首先求解协方差矩阵。由于 $\boldsymbol{x}_1$、$\boldsymbol{x}_2$ 和 $\boldsymbol{y}$ 已经为中心化矩阵，因此协方差矩阵 $\boldsymbol{\Sigma}$ 通过下式计算获得。

$$\begin{aligned}\boldsymbol{\Sigma} &= \begin{bmatrix}\boldsymbol{x}_1 & \boldsymbol{x}_2 & \boldsymbol{y}\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\boldsymbol{x}_1 & \boldsymbol{x}_2 & \boldsymbol{y}\end{bmatrix} \\ &= \begin{bmatrix}\boldsymbol{x}_1^{\mathrm{T}} \\ \boldsymbol{x}_2^{\mathrm{T}} \\ \boldsymbol{y}^{\mathrm{T}}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_1 & \boldsymbol{x}_2 & \boldsymbol{y}\end{bmatrix} = \begin{bmatrix}\boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_1 & \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_2 & \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{y} \\ \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{x}_1 & \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{x}_2 & \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{y} \\ \boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}_1 & \boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}_2 & \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}\end{bmatrix}\end{aligned} \tag{31}$$

为了方便计算，本节也计算不考虑系数 $1/(n-1)$。

正交回归解析式表达：

$$y = b_0 + b_1 x_1 + b_2 x_2 \tag{32}$$

构造二元 $F(x_1, x_2, y)$ 函数如下：

$$F\left(x_1, x_2, y\right) = b_0 + b_1 x_1 + b_2 x_2 - y \tag{33}$$

$F(x_1, x_2, y)$ 法向量即平面 $f(x_1, x_2)$ 法向量 $\boldsymbol{n}$ 通过下式求解：

$$\boldsymbol{n} = \left(\frac{\partial F}{\partial x_1}, \frac{\partial F}{\partial x_2}, \frac{\partial F}{\partial y}\right)^{\mathrm{T}} = \begin{bmatrix}b_1 & b_2 & -1\end{bmatrix}^{\mathrm{T}} \tag{34}$$

$\boldsymbol{n}$ 平行于 $\boldsymbol{\Sigma}$ 矩阵 PCA 分解特征值最小特征向量，即：

$$\boldsymbol{\Sigma}\boldsymbol{v}_3 = \lambda_3 \boldsymbol{v}_3 \quad \Rightarrow \quad \begin{bmatrix}\boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_1 & \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_2 & \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{y} \\ \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{x}_1 & \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{x}_2 & \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{y} \\ \boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}_1 & \boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}_2 & \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}\end{bmatrix}\boldsymbol{n} = \lambda_3 \boldsymbol{n} \tag{35}$$

整理得到：

$$\begin{bmatrix}\boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_1 & \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_2 & \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{y} \\ \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{x}_1 & \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{x}_2 & \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{y} \\ \boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}_1 & \boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}_2 & \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}\end{bmatrix}\begin{bmatrix}b_1 \\ b_2 \\ -1\end{bmatrix} = \lambda_3\begin{bmatrix}b_1 \\ b_2 \\ -1\end{bmatrix} \quad \Rightarrow \quad \begin{cases}\left(\boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_1 - \lambda_3\right)b_1 + \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_2 b_2 = \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{y} \\ \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{x}_1 b_1 + \left(\boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{x}_2 - \lambda_3\right)b_2 = \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{y}\end{cases} \tag{36}$$

$\boldsymbol{n}$ 平行于 $\boldsymbol{\Sigma}$ 矩阵 PCA 分解特征值最小特征向量 $\boldsymbol{v}_3$，构造如下等式并求解 b_1 和 b_2：

$$\begin{bmatrix}b_1 \\ b_2 \\ -1\end{bmatrix} = k\boldsymbol{v}_3 \quad \Rightarrow \quad \begin{bmatrix}b_1 \\ b_2 \\ -1\end{bmatrix} = k\begin{bmatrix}v_{1,3} \\ v_{2,3} \\ v_{3,3}\end{bmatrix} \tag{37}$$

根据 (37) 最后一行，可以求得 k

$$k = \frac{-1}{v_{3,3}} \tag{38}$$

b_1 和 b_2 构成的列向量为：

$$\begin{bmatrix}b_1 \\ b_2\end{bmatrix} = \frac{-1}{v_{3,3}}\begin{bmatrix}v_{1,3} \\ v_{2,3}\end{bmatrix} \tag{39}$$

回归方程常数项通过下式获得：

$$b_0 = \mathrm{E}(\boldsymbol{y}) - \begin{bmatrix} \mathrm{E}(\boldsymbol{x}_1) & \mathrm{E}(\boldsymbol{x}_2) \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{40}$$

为了方便多元正交回归运算，令：

$$\begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{X} \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{X} & \boldsymbol{y} \end{bmatrix} \tag{41}$$

协方差矩阵 $\boldsymbol{\Sigma}$ 为：

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} & \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \\ \boldsymbol{y}^{\mathrm{T}}\boldsymbol{X} & \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \end{bmatrix} \tag{42}$$

上式 $\boldsymbol{\Sigma}$ 也不考虑系数 $1/(n-1)$：

$$\boldsymbol{\Sigma}\boldsymbol{v}_3 = \lambda_3 \boldsymbol{v}_3 \quad \Rightarrow \quad \begin{bmatrix} \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} & \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \\ \boldsymbol{y}^{\mathrm{T}}\boldsymbol{X} & \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} \end{bmatrix} \boldsymbol{n} = \lambda_3 \boldsymbol{n} \tag{43}$$

构造 $\boldsymbol{b} = [b_1, b_2]^{\mathrm{T}}$ 这样重新构造特征值和特征向量以及 $\boldsymbol{\Sigma}$ 之间关系：

$$\boldsymbol{n} = \begin{bmatrix} b_1 \\ b_2 \\ -1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} \\ -1 \end{bmatrix} \tag{44}$$

将 (44) 代入 (43)，整理得到 $\boldsymbol{b}$：

$$\begin{bmatrix} \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} & \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \\ \boldsymbol{y}^{\mathrm{T}}\boldsymbol{X} & \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} \end{bmatrix} \begin{bmatrix} \boldsymbol{b} \\ -1 \end{bmatrix} = \lambda_3 \begin{bmatrix} \boldsymbol{b} \\ -1 \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{b} = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} - \lambda_3 \right)^{-1} \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{45}$$

下一节将使用 (45) 这一解析式计算正交回归解析式系数。

图 10 回顾本章第一节介绍的二元正交回归坐标转换过程。

数据 $[\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{y}]$ 中心化后，用 PCA 正交化获得正交系 $[\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3]$。$\boldsymbol{v}_1, \boldsymbol{v}_2$ 和 $\boldsymbol{v}_3$ 对应特征值由大到小。前两个主元向量 $\boldsymbol{v}_1$ 和 $\boldsymbol{v}_2$ 相互垂直，构成了一个平面 H，特征值最小主元 $\boldsymbol{v}_3$ 垂直于该平面。$\boldsymbol{n}$ 为 H 平面法向量，$\boldsymbol{n}$ 和 $\boldsymbol{v}_3$ 两者平行。

图 10 还比较了 OLS 和 TLS 回归结果。值得大家注意的是，如图 10 上半部分所示，对于最小二乘回归 OLS，$\hat{\boldsymbol{y}}$ 在 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$ 构造的平面上；而如图 10 下半部分，正交回归 TLS 中，$\hat{\boldsymbol{y}}$ 在 $\boldsymbol{v}_1$ 和 $\boldsymbol{v}_2$ 构造平面 H 上。

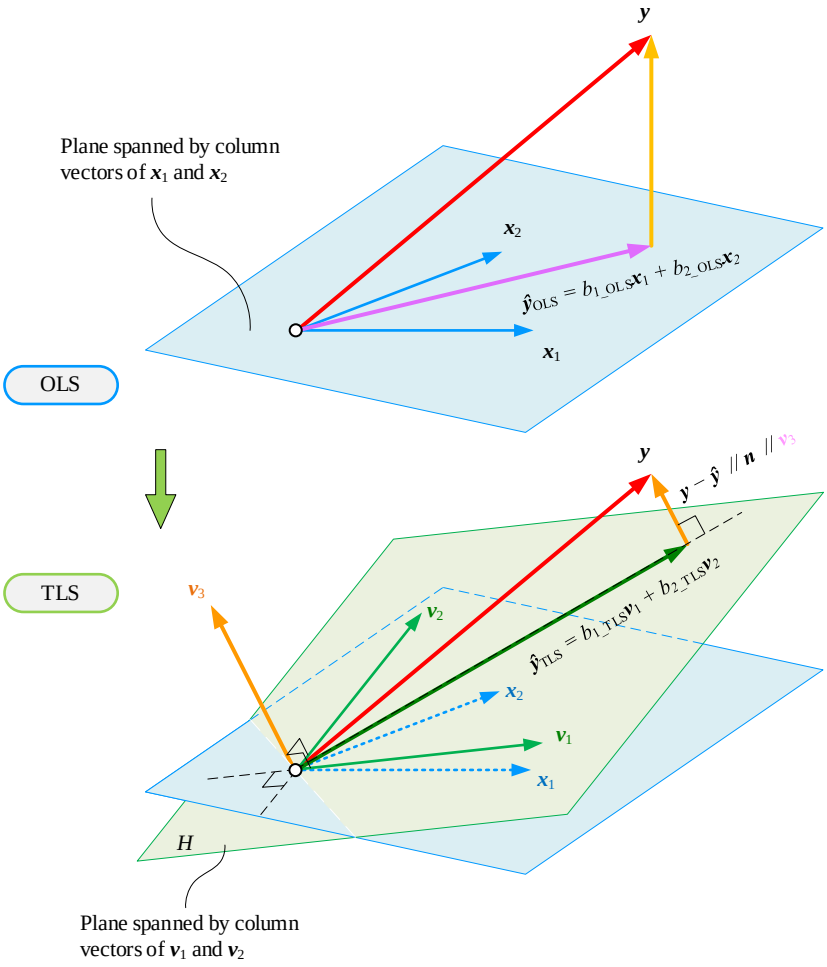


图 10. 几何角度解释二元正交回归坐标转换

图 11 解释二元正交回归数据关系。如前文反复强调，输入数据和输出数据都参与主成分分析，也就是正交化过程，因此特征向量既有“输入”成分，也有“输出”成分，呈现“你中有我，我中有你”。

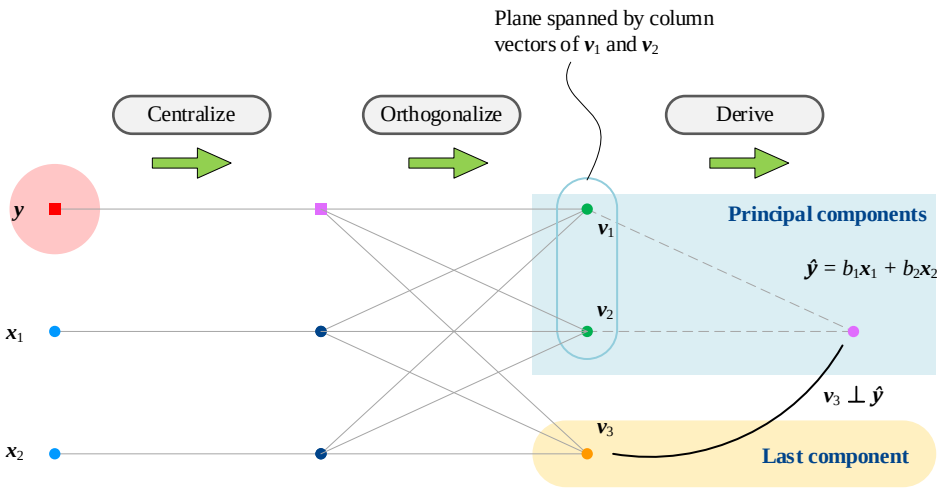


图 11. 二元正交回归数据关系

利用上一节介绍的 scipy.odr，可以求解一个二元正交回归的结果如下。利用主成分分析，我们可以获得相同正交回归的系数。

```
Beta: [-0.00061177  0.40795725  0.44382723]
Beta Std Error: [0.00057372 0.02454606 0.02864744]
Beta Covariance: [[ 5.46486647e-03 -2.24817813e-02  1.00466594e-02]
 [-2.24817813e-02  1.00032390e+01 -7.07446738e+00]
 [ 1.00466594e-02 -7.07446738e+00  1.36253753e+01]]
Residual Variance: 6.02314210079386e-05
Inverse Condition #: 0.16900716799896934
Reason(s) for Halting:
Sum of squares convergence
```

二元正交回归的平面解析式为：

$$y = 0.4079x_1 + 0.4438x_2 - 0.00061 \tag{46}$$

图 12 所示为最小二乘法 OLS 二元线性回归结果，对应的平面解析式如下：

$$y = 0.3977x_1 + 0.4096x_2 - 0.006 \tag{47}$$

OLS Regression Results

Dep. Variable:	SP500	R-squared:	0.830
Model:	OLS	Adj. R-squared:	0.829
Method:	Least Squares	F-statistic:	607.4
Date:	Thu, 07 Oct 2021	Prob (F-statistic):	1.69e-96
Time:	07:31:57	Log-Likelihood:	831.06
No. Observations:	252	AIC:	-1656.
Df Residuals:	249	BIC:	-1646.
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.0006	0.001	-0.984	0.326	-0.002	0.001
AAPL	0.3977	0.024	16.326	0.000	0.350	0.446
MCD	0.4096	0.028	14.442	0.000	0.354	0.465

Omnibus:	37.744	Durbin-Watson:	1.991
Prob(Omnibus):	0.000	Jarque-Bera (JB):	157.710
Skew:	0.492	Prob(JB):	5.67e-35
Kurtosis:	6.749	Cond. No.	59.4

图 12. 最小二乘法 OLS 二元线性回归结果

图 13 比较 OLS 和 TLS 二元回归结果。

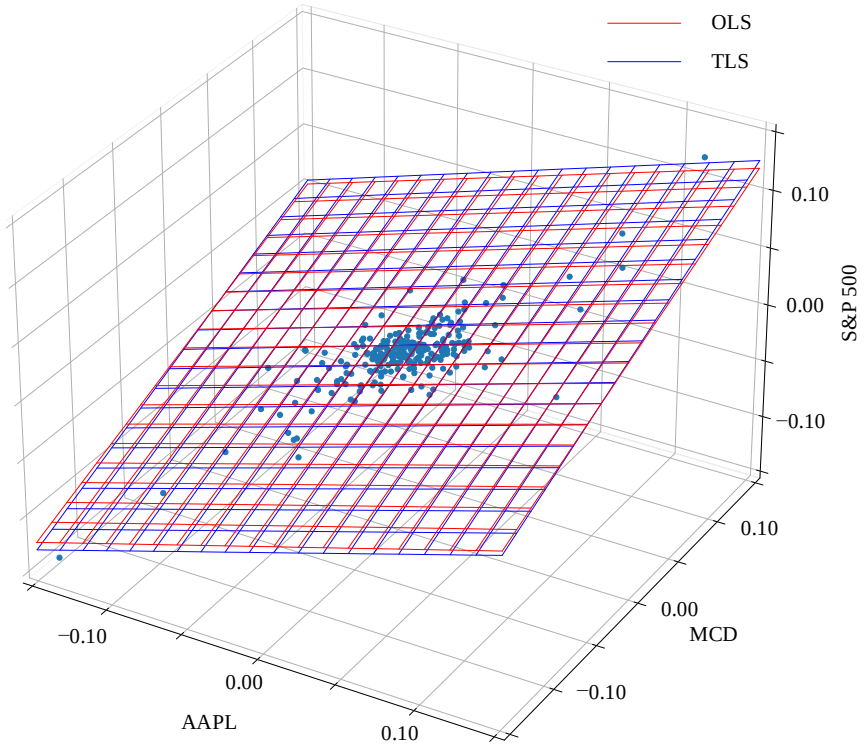


图 13. 比较 OLS 和 TLS 二元回归结果

Bk6_Ch16_02.py 完成本节回归运算。

16.5 多元正交回归

下面，把上述思路推广到 D 维度 $\boldsymbol{X}$ 矩阵。首先中心化数据，获得如下两个中心化 $\boldsymbol{X}, \boldsymbol{y}$ 向量：

$$\boldsymbol{X}_{n\times D} = \left(\boldsymbol{I} - \frac{1}{n}\boldsymbol{1}\boldsymbol{1}^T\right)\boldsymbol{X}, \quad \boldsymbol{y} = \boldsymbol{y} - \mathrm{E}(\boldsymbol{y}) \tag{48}$$

为了表达方便，假设 $\boldsymbol{X}$ 和 $\boldsymbol{y}$ 已经为中心化数据；这样，构造回归方程式时，不必考虑常数项 b_0，即回归方程中没有截距项：

$$y = b_1 x_1 + b_2 x_2 + \cdots + b_{D-1} x_{D-1} + b_D x_D \tag{49}$$

为了进行 PCA 分解，首先计算 $[\boldsymbol{X}, \boldsymbol{y}]$ 矩阵协方差矩阵。

$\boldsymbol{X}$ 和 $\boldsymbol{y}$ 均是中心化数据，不考虑系数 $1/(n-1)$，协方差矩阵通过下式简单运算获得：

$$\boldsymbol{\Sigma}_{(D+1)\times(D+1)} = [\boldsymbol{X},\boldsymbol{y}]^{\mathrm{T}}[\boldsymbol{X},\boldsymbol{y}] = \begin{bmatrix}\boldsymbol{X}^{\mathrm{T}}\\ \boldsymbol{y}^{\mathrm{T}}\end{bmatrix}[\boldsymbol{X},\boldsymbol{y}] = \begin{bmatrix}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} & \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}\\ \boldsymbol{y}^{\mathrm{T}}\boldsymbol{X} & \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}\end{bmatrix} \tag{50}$$

上述协方差矩阵行列宽度均为 $D+1$。对它进行特征值分解得到：

$$\boldsymbol{\Sigma} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1} \tag{51}$$

其中，

$$\boldsymbol{\Lambda} = \begin{bmatrix}\lambda_1 & & & & \\ & \lambda_2 & & & \\ & & \ddots & & \\ & & & \lambda_D & \\ & & & & \lambda_{D+1}\end{bmatrix}, \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D \geq \lambda_{D+1}$$
$$\boldsymbol{V} = \begin{bmatrix}\boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_D & \boldsymbol{v}_{D+1}\end{bmatrix} \tag{52}$$

特征值矩阵对角线特征值从左到右，由大到小。有了本章之前内容铺垫，相信读者已经清楚正交回归的矩阵运算过程，具体如图 14 所示。

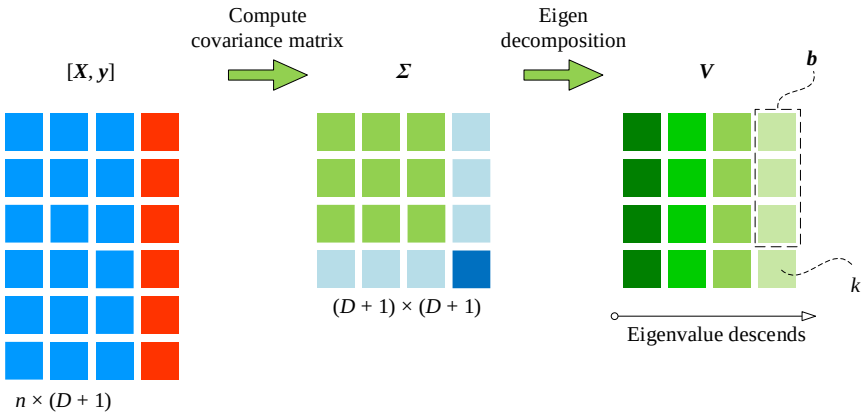


图 14. 多元正交回归矩阵运算过程

$\boldsymbol{V}$ 中第 1 到第 D 个行向量 $[\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_D]$ 构造超平面 H，而 $\boldsymbol{v}_{D+1}$ 垂直于该超平面。

构造 $F(x_1, x_2, ..., x_D, y)$ 函数：

$$F(x_1, x_2, ..., x_D, y) = b_1x_1 + b_2x_2 + \cdots + b_{D-1}x_{D-1} + b_Dx_D - y \tag{53}$$

$F(x_1, x_2, ..., x_D, y)$ 法向量即平面上 $f(x_1, x_2, ..., x_D)$ 法向量 $\boldsymbol{n}$ 通过下式求解：

$$\boldsymbol{n} = \left(\frac{\partial F}{\partial x_1}, ..., \frac{\partial F}{\partial x_D}, \frac{\partial F}{\partial y}\right)^{\mathrm{T}} = \begin{bmatrix}b_1 & b_2 & \cdots & b_D & -1\end{bmatrix}^{\mathrm{T}} = \begin{bmatrix}\boldsymbol{b}\\ -1\end{bmatrix} \tag{54}$$

这样重新构造特征值 λ_{D+1} 和特征向量 $\boldsymbol{v}_{D+1}$ 以及 $\boldsymbol{\Sigma}$ 之间关系。注意，$\boldsymbol{n}$ 平行 $\boldsymbol{v}_{D+1}$。$\boldsymbol{n}$ 对应 $\boldsymbol{\Sigma}$ 矩阵 PCA 分解特征值最小特征向量，即：

$$\boldsymbol{\Sigma} \boldsymbol{v}_{D+1} = \lambda_{D+1} \boldsymbol{v}_{D+1} \quad \Rightarrow \quad \begin{bmatrix} \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} & \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \\ \boldsymbol{y}^{\mathrm{T}}\boldsymbol{X} & \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} \end{bmatrix} \boldsymbol{n} = \lambda_{D+1} \boldsymbol{n} \tag{55}$$

求解获得多元正交回归系数列向量 $\boldsymbol{b}$ 解：

$$\begin{bmatrix} \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} & \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \\ \boldsymbol{y}^{\mathrm{T}}\boldsymbol{X} & \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} \end{bmatrix} \begin{bmatrix} \boldsymbol{b} \\ -1 \end{bmatrix} = \lambda_{D+1} \begin{bmatrix} \boldsymbol{b} \\ -1 \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{b}_{\mathrm{TLS}} = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} - \lambda_{D+1} \right)^{-1} \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{56}$$

对比多元线性最小二乘系数向量结果：

$$\boldsymbol{b}_{\mathrm{OLS}} = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} \right)^{-1} \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{57}$$

发现当 λ_{D+1} 等于 0 时，$\boldsymbol{y}$ 完全被 $\boldsymbol{X}$ 列向量解释，即两个共线性。

这里我们再次区分一下最小二乘法和正交回归。最小二乘法寻找因变量和自变量之间残差平方和最小超平面；几何角度上讲，将因变量投影在自变量构成超平面 H，使得残差向量垂直 H。正交回归则通过正交化自变量和因变量，构造一个新正交空间；这个新正交空间基底向量为分解得到主元向量，具体如图 15 所示。

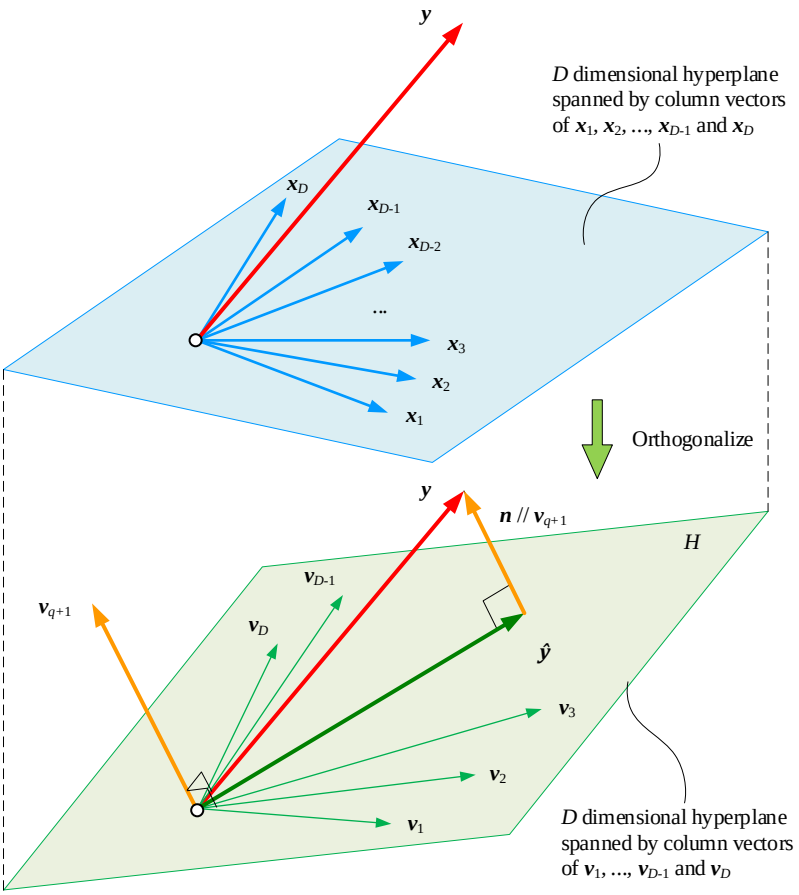


图 15. 几何角度解释多元正交回归

$\boldsymbol{n}$ 平行于数据 $[\boldsymbol{X}, \boldsymbol{y}]$ PCA 分解特征值最小特征向量 $\boldsymbol{v}_{D+1}$，构造如下等式并求解 $b_1, ..., b_D$：

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_D \\ -1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} \\ -1 \end{bmatrix} = k\boldsymbol{v}_{D+1} \quad \Rightarrow \quad \begin{bmatrix} \boldsymbol{b} \\ -1 \end{bmatrix} = k \begin{bmatrix} v_{1,D+1} \\ v_{2,D+1} \\ \vdots \\ v_{D,D+1} \\ v_{D+1,D+1} \end{bmatrix} \tag{58}$$

求解 k 得到：

$$k = \frac{-1}{v_{D+1,D+1}} \tag{59}$$

求解 $\boldsymbol{b}$ 得到：

$$\boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_D \end{bmatrix} = \frac{-1}{v_{D+1,D+1}} \begin{bmatrix} v_{1,D+1} \\ v_{2,D+1} \\ \vdots \\ v_{D,D+1} \end{bmatrix} \tag{60}$$

b_0 通过下式求得。

$$b_0 = \mathrm{E}(\boldsymbol{y}) - \begin{bmatrix} \mathrm{E}(\boldsymbol{x}_1) & \mathrm{E}(\boldsymbol{x}_2) & \cdots & \mathrm{E}(\boldsymbol{x}_D) \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_D \end{bmatrix} \tag{61}$$

图 16 展示多元正交回归运算数据关系。看到数据 $[\boldsymbol{X}, \boldsymbol{y}]$ 均参与到了正交化中；正交化结果为 $D + 1$ 个正交向量 $[\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_D, \boldsymbol{v}_{D+1}]$。通过向量 $\boldsymbol{v}_{D+1}$ 垂直 $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_D$ 构成超平面，推导出多元正交回归解析式。

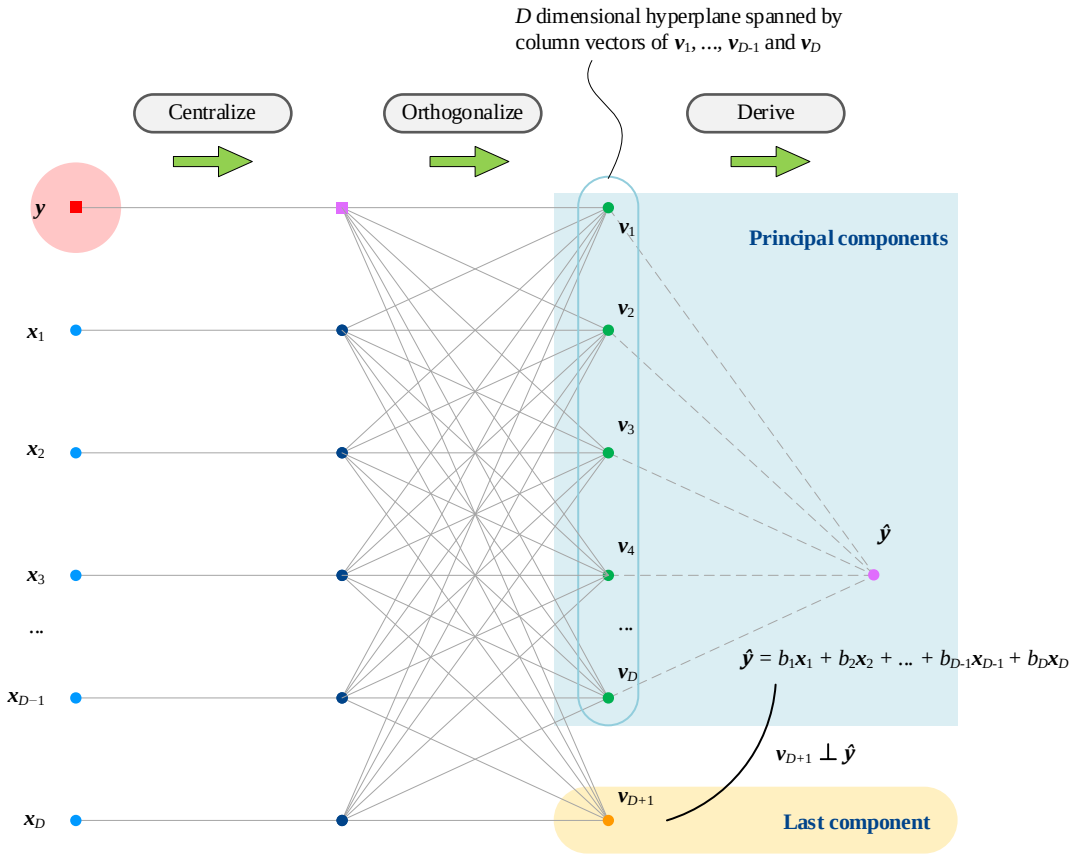


图 16. 多元正交回归运算数据关系

图 17 所示直方图，比较多元 TLS 回归和多元 OLS 回归系数。

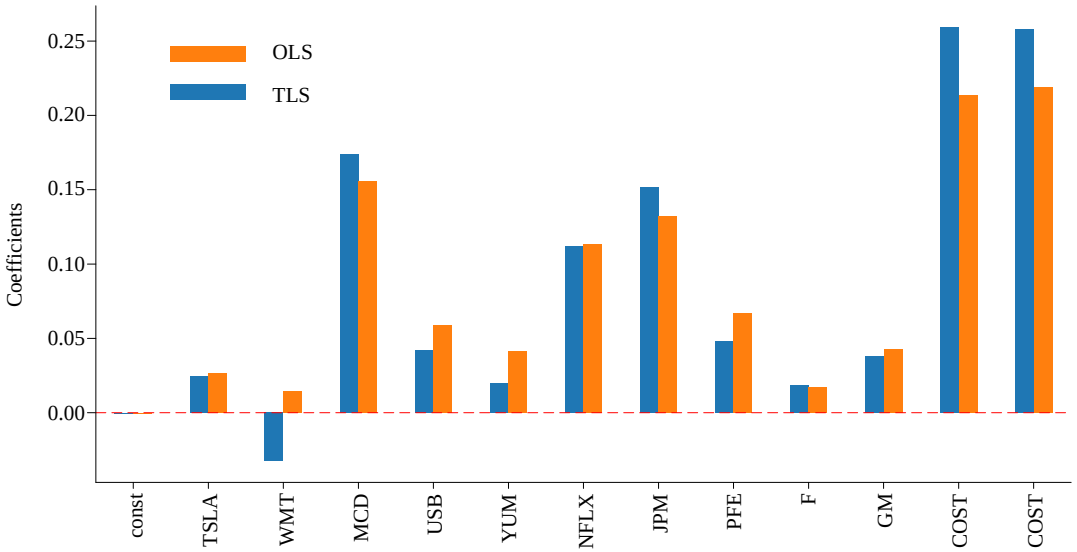
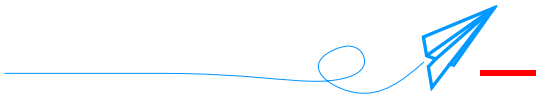


图 17. 比较多元 TLS 回归和多元 OLS 回归系数

Bk6_Ch16_03.py 完成本节回归运算。



正交回归和最小二乘法回归都是回归分析中的方法，但它们之间有很大的区别。

OLS 通过最小化实际观测值与预测值之间的误差平方和，来确定回归系数。这种方法非常直观且易于理解，但存在一些缺点，例如当数据存在多重共线性时，OLS 的估计结果可能会变得不稳定，且估计结果受到极端值的影响较大。

与 OLS 不同，正交回归是一种基于主成分分析的回归方法。它通过将自变量通过主成分分析转换成互相正交的新变量，来消除自变量之间的多重共线性问题，从而提高回归分析的准确性和稳定性。

因此，正交回归方法相对于 OLS 方法更加鲁棒，适用于多重共线性较强的数据集，同时也能够在保证预测准确性的前提下，降低自变量的维度，提高回归模型的可解释性。

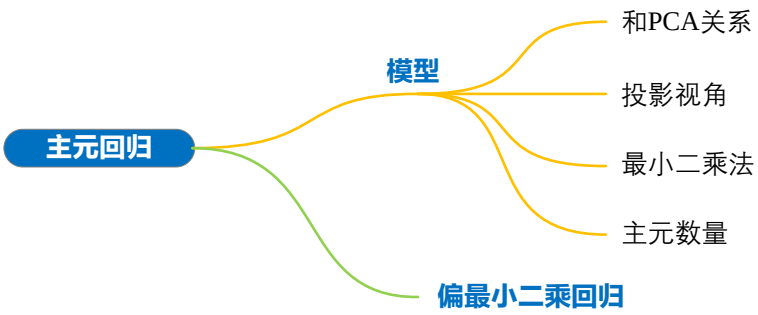
17 主元回归

Principal Components Regression

输入特征主成分分析，输出数据投影到选定主元超平面

> 大理石中我看到了天使，我拿起刻刀不停雕刻，直到还它自由。
>
> ***I saw the angel in the marble and carved until I set him free.***
>
> —— 米开朗琪罗 (Michelangelo) | 文艺复兴三杰之一 | 1475 ~ 1564

- `seaborn.heatmap()` 绘制数据热图
- `seaborn.jointplot()` 绘制联合分布和边际分布
- `seaborn.kdeplot()` 绘制 KDE 核概率密度估计曲线
- `seaborn.lineplot()` 绘制线图
- `seaborn.relplot()` 绘制散点图和曲线图
- `sklearn.decomposition.PCA()` 主成分分析函数
- `statsmodels.api.add_constant()` 线性回归增加一列常数 1
- `statsmodels.api.OLS()` 最小二乘法函数



17.1 主元回归

本节讲解主元回归 (Principal Components Regression, PCR)。主元回归类似本章前文介绍的正交回归。多元正交回归中，自变量和因变量数据 $[\boldsymbol{X}, \boldsymbol{y}]$ 利用正交化，按照特征值从大小排列特征向量，用 $[\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_D]$ 构造一个全新超平面，$\boldsymbol{v}_{D+1}$ 垂直于超平面关系求解出正交化回归系数。

而主元回归，因变量数据 $\boldsymbol{y}$ 完全不参与正交化，即仅仅 $\boldsymbol{X}$ 参与 PCA 分解，获得特征值由大到小排列 D 个主元 $\boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_D)$；这 D 个主元方向 $(\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_D)$ 两两正交。选取其中 k ($k < D$) 个特征值较大主元 $(\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_k)$，构造超平面；最后一步，用最小二乘法将因变量 $\boldsymbol{y}$ 投影在超平面上。

图 1 提供一个例子，$\boldsymbol{X}$ 有三个维度数据，$\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3]$。首先对 $\boldsymbol{X}$ 列向量 PCA 分解，获得正交化向量 $[\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3]$。然后，选取作为 $\boldsymbol{v}_1$ 和 $\boldsymbol{v}_2$ 主元，构造一个平面；用最小二乘法，将因变量 $\boldsymbol{y}$ 投影在平面上，获得回归方程。再次请大家注意，主元回归因变量 $\boldsymbol{y}$ 数据并不参与正交化；另外，主元回归选取前 P ($P < D$) 个特征值较大主元 $\boldsymbol{V}_{D\times P}$ $(\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_P)$，构造一个超平面。

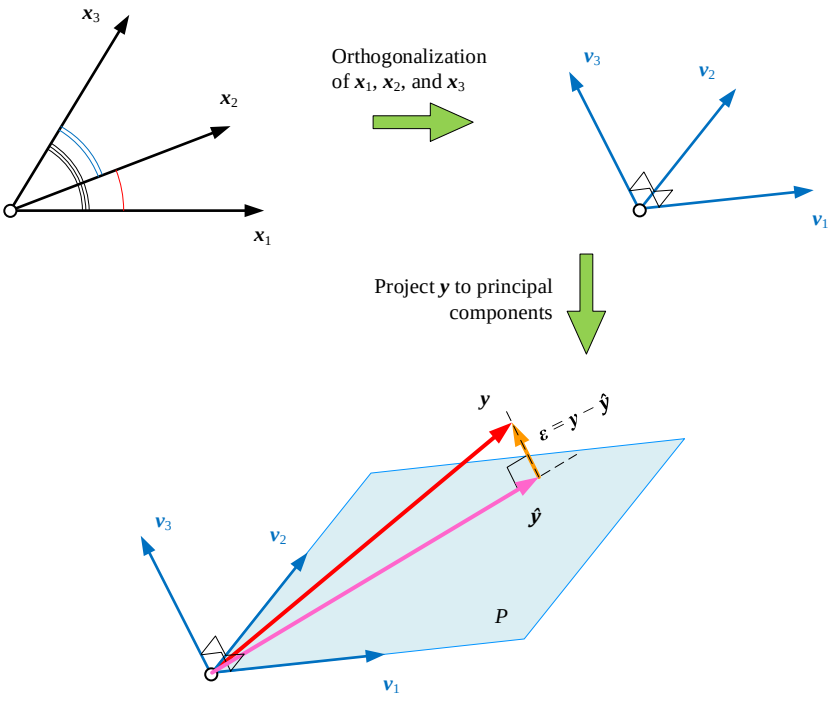


图 1. 主元回归原理

17.2 原始数据

下载如图 2 所示为归一化股价数据，将其转化为日收益率，作为数据 $\boldsymbol{X}$ 和 $\boldsymbol{y}$；其中 S&P 500 日收益率为数据 $\boldsymbol{y}$，其余股票日收益率作为数据 $\boldsymbol{X}$。图 3 所示为数据 $\boldsymbol{X}$ 和 $\boldsymbol{y}$ 的热图。

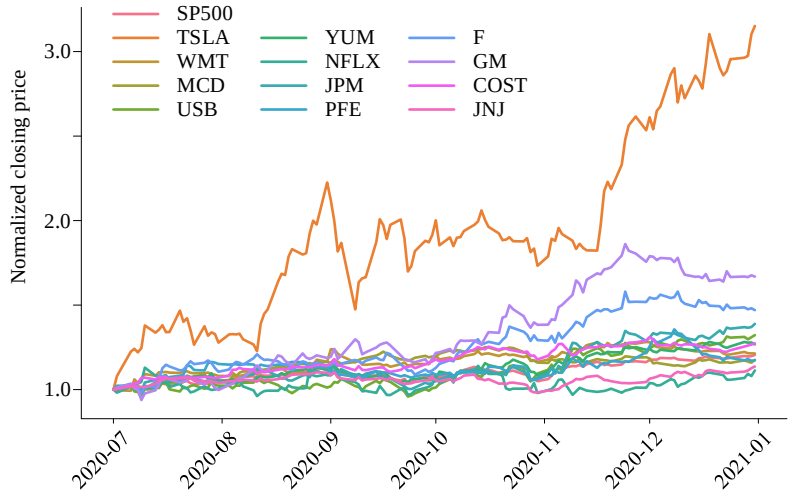


图 2. 股价走势，归一化数据

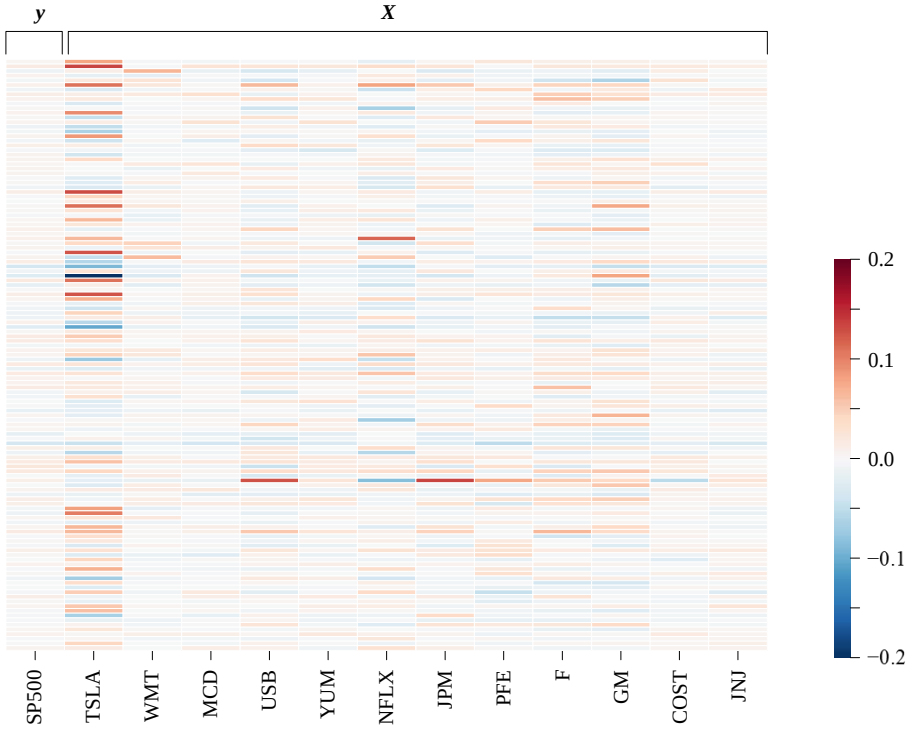


图 3. 数据 $\boldsymbol{X}$ 和 $\boldsymbol{y}$ 的热图

图 4 几个分图给出的是数据 $\boldsymbol{X}$ 和 $\boldsymbol{y}$ 的 KDE 分布。

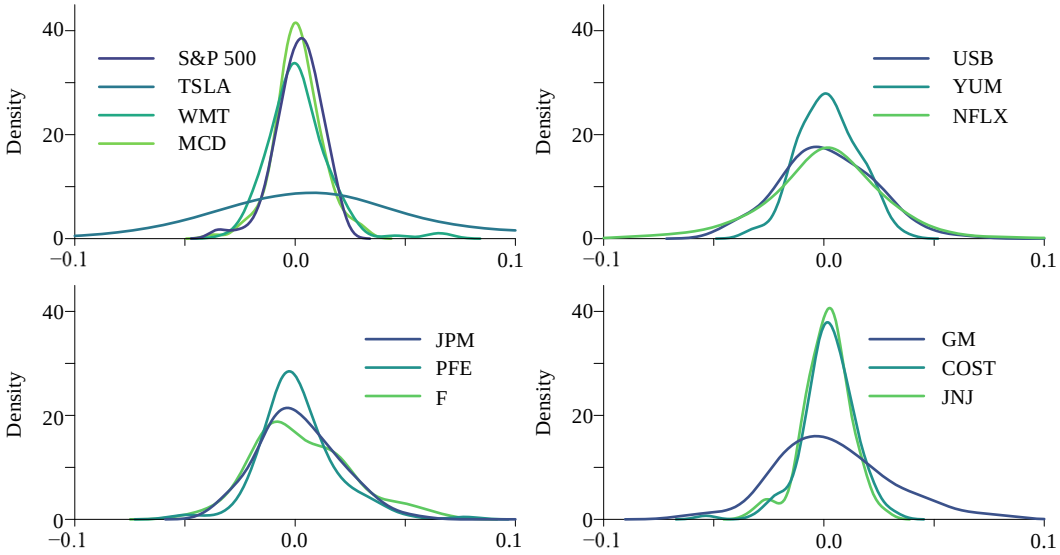


图 4. 数据 $\boldsymbol{X}$ 和 $\boldsymbol{y}$ 的 KDE 分布

17.3 主成分分析

对数据 $\boldsymbol{X}$ 进行主成分分析，可以获得如表 1 所示的前四个主成分 $\boldsymbol{V}_{D\times p}$ 参数。可以利用热图和线图对 $\boldsymbol{V}_{D\times p}$ 进行可视化，如图 5 所示。

表 1. 前四个主成分

	PC1	PC2	PC3	PC4
TSLA	-0.947	-0.004	0.256	0.121
WMT	-0.073	0.016	-0.193	0.066
MCD	-0.056	0.076	-0.111	0.115
USB	-0.021	0.503	0.122	-0.502
YUM	-0.044	0.188	-0.037	0.057
NFLX	-0.281	-0.133	-0.776	-0.448
JPM	-0.019	0.442	0.167	-0.425
PFE	-0.045	0.174	0.187	0.118
F	-0.004	0.457	-0.179	0.178
GM	0.007	0.491	-0.360	0.518
COST	-0.096	-0.027	-0.203	0.114
JNJ	-0.042	0.108	0.021	0.066

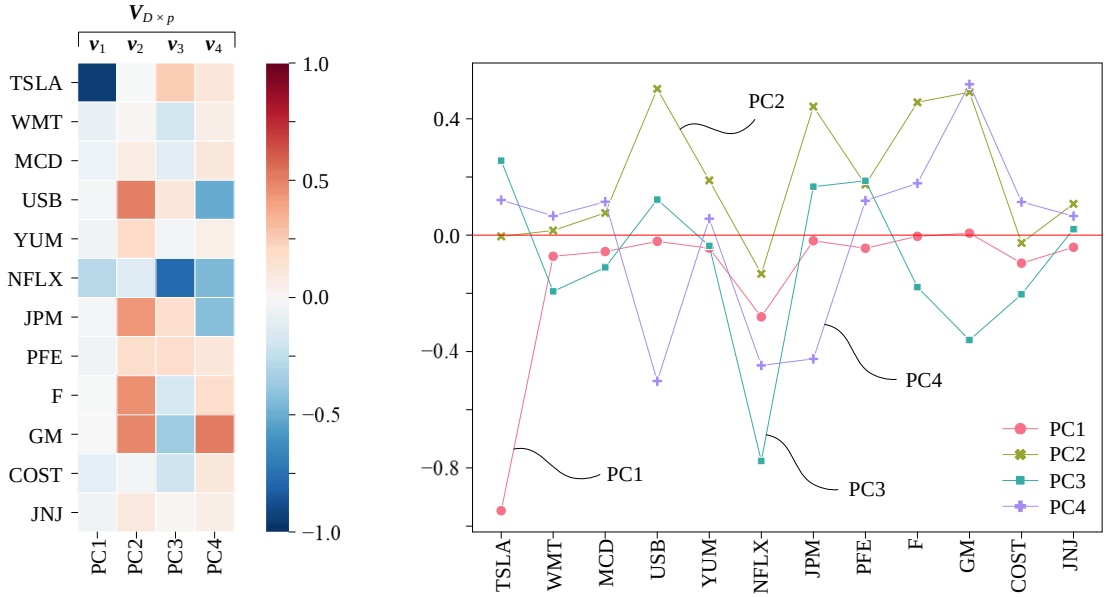


图 5. 前四个主成分可视化

图 5 所示 $\boldsymbol{V}_{D\times p}$ 两两正交，具有如下性质：

$$\boldsymbol{V}_{D\times p}{}^{\mathrm{T}}\boldsymbol{V}_{D\times p}=\boldsymbol{I}_{p\times p} \tag{1}$$

图 6 所示为 (1) 计算热图。

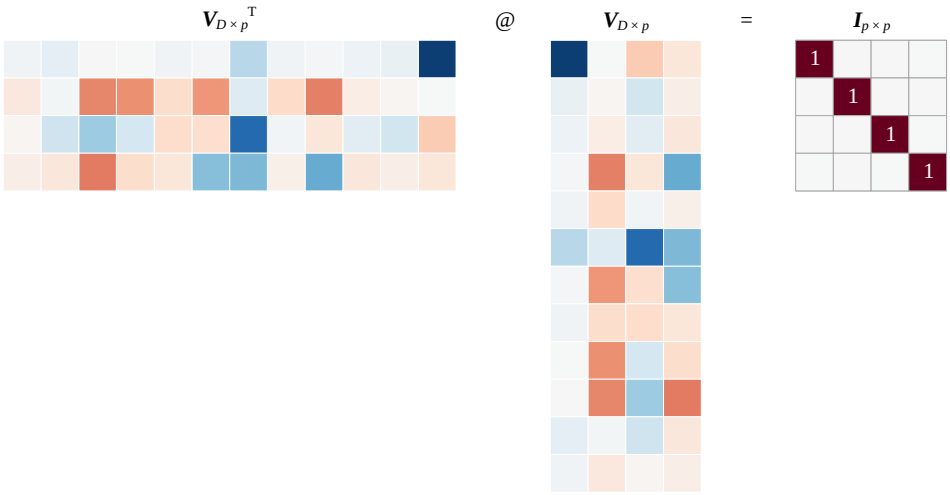


图 6. $\boldsymbol{V}_{D\times p}$ 两两正交

17.4 数据投影

如图 7 所示，原始数据 $\boldsymbol{X}$ 在 p 维正交空间 $(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_p)$ 投影得到数据 $\boldsymbol{Z}_{n\times p}$：

$$\boldsymbol{Z}_{n\times p} = \boldsymbol{X}_{n\times D}\boldsymbol{V}_{D\times p} \tag{2}$$

图 8 所示为 $\boldsymbol{Z}_{n\times p}$ 数据热图。

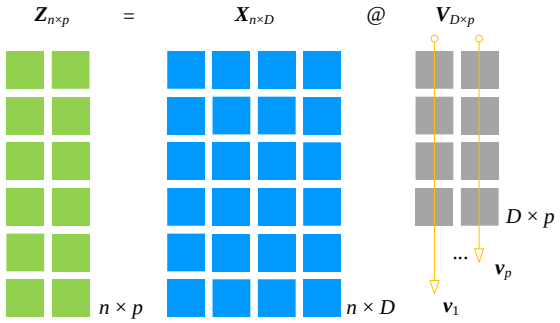


图 7. PCA 分解部分数据关系

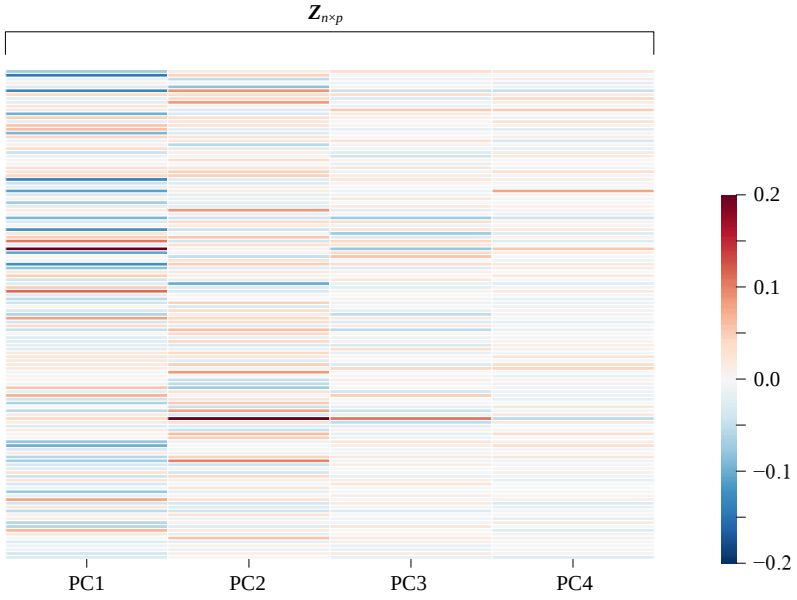


图 8. 前四个主成分数据

图 9 所示为 $\boldsymbol{Z}_{n\times p}$ 每列主成分数据的分布情况。容易注意到，第一主成分数据解释最大方差。

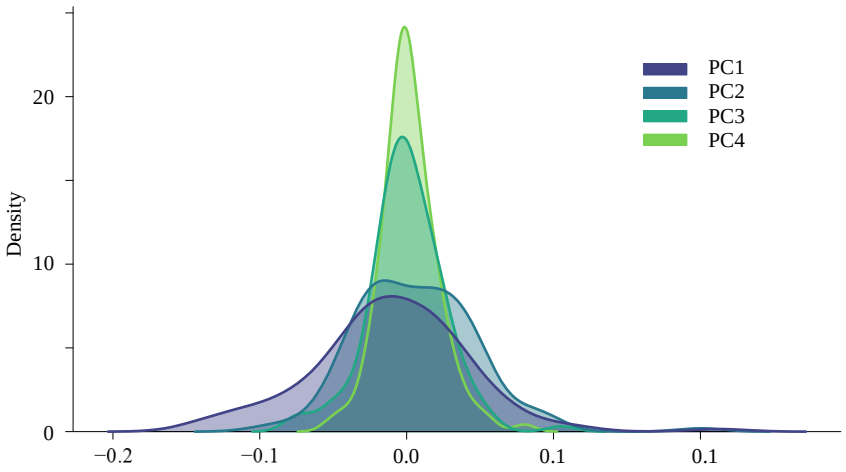


图 9. 前四个主成分数据分布

图 10 所示为 $\boldsymbol{Z}_{n\times p}$ 数协方差矩阵热图。

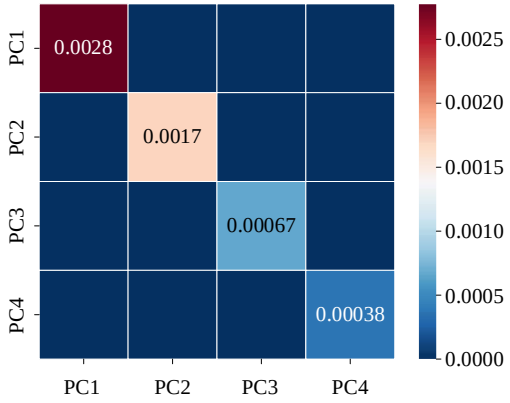


图 10. 前四个主元的协方差矩阵

前四个主成分对应的奇异值分别为：

$$s_1 = 0.5915, \quad s_2 = 0.4624, \quad s_3 = 0.2911, \quad s_4 = 0.2179 \tag{3}$$

所对应的特征值：

$$\begin{aligned}
\lambda_1 &= \frac{s_1^2}{n-1} = \frac{0.5915^2}{126} = 0.0028 \\
\lambda_2 &= \frac{s_2^2}{n-1} = \frac{0.4624^2}{126} = 0.0017 \\
\lambda_3 &= \frac{s_3^2}{n-1} = \frac{0.2911^2}{126} = 0.00067 \\
\lambda_4 &= \frac{s_4^2}{n-1} = \frac{0.2179^2}{126} = 0.00038
\end{aligned} \tag{4}$$

这四个特征值对应图 10 热图对角线元素。如图 11 所示陡坡图，前四个主元解释了 84.87%方差。

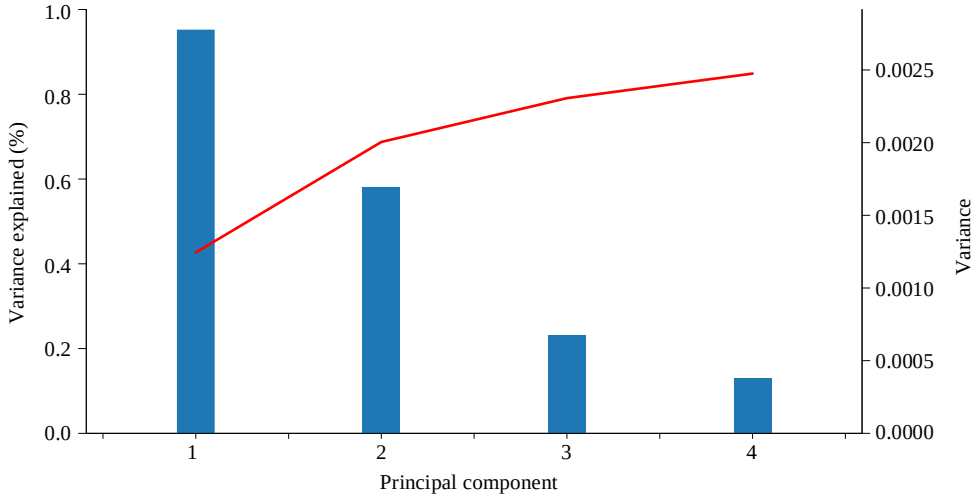


图 11. 陡坡图

转化矩阵 $\boldsymbol{Z}_{n\times P}$ 仅包含 $\boldsymbol{X}$ 部分信息，两者信息之间差距通过下式计算获得，如图 12：

$$\boldsymbol{X}_{n\times D} = \boldsymbol{Z}_{n\times P}\left(\boldsymbol{V}_{D\times P}\right)^{\mathrm{T}} + \boldsymbol{E}_{n\times D} \tag{5}$$

图 12. $\boldsymbol{Z}_{n\times P}$ 还原数据和 $\boldsymbol{X}$ 信息差距

17.5 最小二乘法

主元回归最后一步，用最小二乘法把因变量 $\boldsymbol{y}$ 投影在数据 $\boldsymbol{Z}_{n\times P}$ 构造空间中：

$$\hat{\boldsymbol{y}} = b_{Z,1}\boldsymbol{z}_1 + b_{Z,2}\boldsymbol{z}_2 + \ldots + b_{Z,P}\boldsymbol{z}_P \tag{6}$$

写成矩阵运算：

$$\hat{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{z}_1 & \boldsymbol{z}_2 & \cdots & \boldsymbol{z}_P \end{bmatrix} \begin{bmatrix} b_{Z,1} \\ b_{Z,2} \\ \vdots \\ b_{Z,P} \end{bmatrix} = \boldsymbol{Z}_{n\times P}\boldsymbol{b}_Z \tag{7}$$

图 13 所示为上述运算过程。

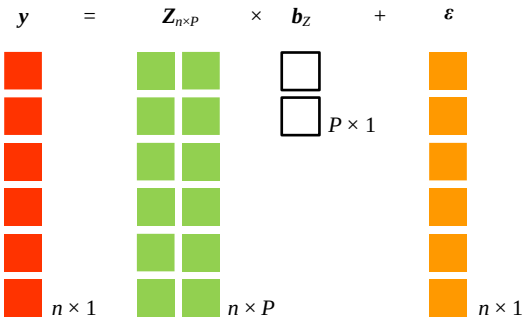


图 13. 最小二乘法回归获得 $\boldsymbol{y} = \boldsymbol{Z}_{n\times P}\boldsymbol{b}_Z + \boldsymbol{\varepsilon}$

根据本书前文讲解内容最小二乘法解，获得 $\boldsymbol{b}_Z$：

$$\begin{aligned} \boldsymbol{b}_Z &= \left(\boldsymbol{Z}_{n\times P}^{\mathrm{T}}\boldsymbol{Z}_{n\times P}\right)^{-1}\boldsymbol{Z}_{n\times P}^{\mathrm{T}}\boldsymbol{y} \\ &= \left(\left(\boldsymbol{X}_{n\times D}\boldsymbol{V}_{D\times P}\right)^{\mathrm{T}}\left(\boldsymbol{X}_{n\times D}\boldsymbol{V}_{D\times P}\right)\right)^{-1}\left(\boldsymbol{X}_{n\times D}\boldsymbol{V}_{D\times P}\right)^{\mathrm{T}}\boldsymbol{y} \end{aligned} \tag{8}$$

如图 13 所示，$\boldsymbol{y}$、拟合数据 $\hat{\boldsymbol{y}}$ 和数据 $\boldsymbol{Z}_{n\times P}$ 关系如下：

$$\begin{cases} \boldsymbol{y} = \boldsymbol{Z}_{n\times P}\boldsymbol{b}_Z + \boldsymbol{\varepsilon} \\ \hat{\boldsymbol{y}} = \boldsymbol{Z}_{n\times P}\boldsymbol{b}_Z \\ \boldsymbol{\varepsilon} = \boldsymbol{y} - \hat{\boldsymbol{y}} \end{cases} \tag{9}$$

图 14 所示为最小二乘法线性回归结果。

系数向量 $\boldsymbol{b}_Z$ 结果如下：

$$\boldsymbol{b}_Z = \begin{bmatrix} -0.1039 & 0.1182 & -0.0941 & -0.0418 \end{bmatrix}^{\mathrm{T}} \tag{10}$$

OLS Regression Results

Dep. Variable:	SP500	R-squared:	0.552
Model:	OLS	Adj. R-squared:	0.537
Method:	Least Squares	F-statistic:	37.60
Date:	XXXXXXXXXX	Prob (F-statistic):	1.82e-20
Time:	XXXXXXXXXX	Log-Likelihood:	450.53
No. Observations:	127	AIC:	-891.1
Df Residuals:	122	BIC:	-876.8
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.0003	0.001	-0.520	0.604	-0.002	0.001
PC1	-0.1039	0.012	-8.647	0.000	-0.128	-0.080
PC2	0.1182	0.015	7.689	0.000	0.088	0.149
PC3	-0.0941	0.024	-3.854	0.000	-0.142	-0.046
PC4	-0.0418	0.033	-1.283	0.202	-0.106	0.023

Omnibus:	9.631	Durbin-Watson:	2.087
Prob(Omnibus):	0.008	Jarque-Bera (JB):	21.795
Skew:	0.092	Prob(JB):	1.85e-05
Kurtosis:	5.021	Cond. No.	51.7

图 14. 最小二乘法线性回归结果

下面将系数向量 $\boldsymbol{b}_Z$ 利用 $(\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_P)$ 转换为 $\boldsymbol{b}_X$，具体过程图 15 所示：

$$\boldsymbol{b}_X = \boldsymbol{V}_{D\times P}\boldsymbol{b}_Z = \boldsymbol{V}_{D\times P}\left(\boldsymbol{Z}_{n\times P}^{\mathrm{T}}\boldsymbol{Z}_{n\times P}\right)^{-1}\boldsymbol{Z}_{n\times P}^{\mathrm{T}}\boldsymbol{y} \tag{11}$$

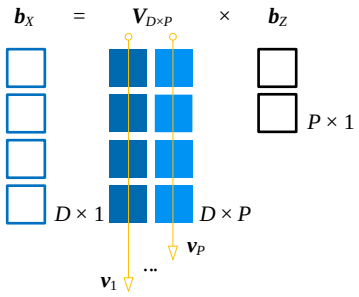


图 15. $\boldsymbol{b}_z$ 和 $\boldsymbol{b}_x$ 之间转换关系

系数 $\boldsymbol{b}_X$ 可以通过下式计算得到：

$$\boldsymbol{b}_X = \boldsymbol{V}_{D\times P}\boldsymbol{b}_Z = \boldsymbol{V}_{D\times P}\begin{bmatrix}-0.1039 & 0.1182 & -0.0941 & -0.0418\end{bmatrix}^{\mathrm{T}} \tag{12}$$

图 16 所示为系数 $\boldsymbol{b}_X$ 直方图。

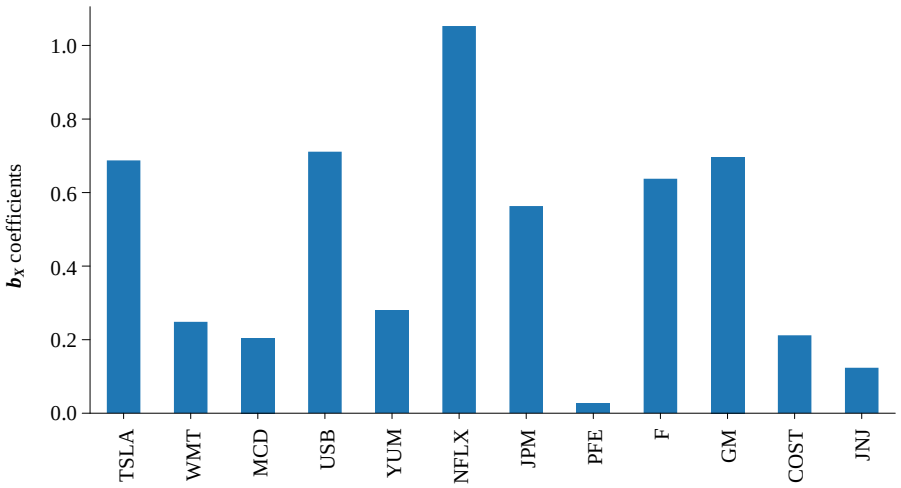


图 16. 系数 $\boldsymbol{b}_X$ 直方图

这样获得 $\boldsymbol{y}$、拟合数据 $\hat{\boldsymbol{y}}$ 和数据 $\boldsymbol{X}$ 之间关系，如图 17 所示：

$$\begin{cases} \boldsymbol{y} = \boldsymbol{X}\boldsymbol{b}_X + \boldsymbol{\varepsilon} \\ \hat{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{b}_X \\ \boldsymbol{\varepsilon} = \boldsymbol{y} - \hat{\boldsymbol{y}} \end{cases} \tag{13}$$

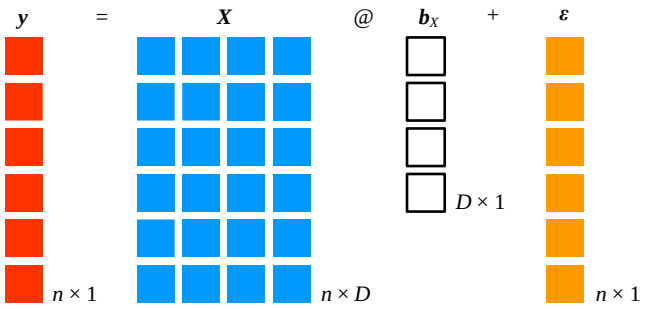


图 17. $\boldsymbol{y}$ 和数据 $\boldsymbol{X}$ 之间回归方程

计算截距项系数 b_0：

$$b_0 = \mathrm{E}(\boldsymbol{y}) - \begin{bmatrix} \mathrm{E}(\boldsymbol{x}_1) & \mathrm{E}(\boldsymbol{x}_2) & \cdots & \mathrm{E}(\boldsymbol{x}_D) \end{bmatrix} \boldsymbol{b}_X \tag{14}$$

计算截距项系数 b_0：

$$\begin{aligned} b_0 &= \mathrm{E}(\boldsymbol{y}) - \begin{bmatrix} \mathrm{E}(\boldsymbol{x}_1) & \mathrm{E}(\boldsymbol{x}_2) & \cdots & \mathrm{E}(\boldsymbol{x}_D) \end{bmatrix} \boldsymbol{b}_X \\ &= -0.00034057 \end{aligned} \tag{15}$$

最后主元回归函数可以通过下式计算得到：

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_D x_D = b_0 + \begin{bmatrix} x_1 & x_2 & \cdots & x_D \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_D \end{bmatrix} = b_0 + \begin{bmatrix} x_1 & x_2 & \cdots & x_D \end{bmatrix} \boldsymbol{b}_X$$

$$= b_0 + \begin{bmatrix} z_1 & z_2 & z_3 & z_4 \end{bmatrix} \boldsymbol{V}_{D\times P} \boldsymbol{b}_Z \tag{16}$$

$$= b_0 + \begin{bmatrix} z_1 & z_2 & z_3 & z_4 \end{bmatrix} \begin{bmatrix} b_{z1} \\ b_{z2} \\ b_{z3} \\ b_{z4} \end{bmatrix}$$

图 18 展示主元回归计算过程数据关系。

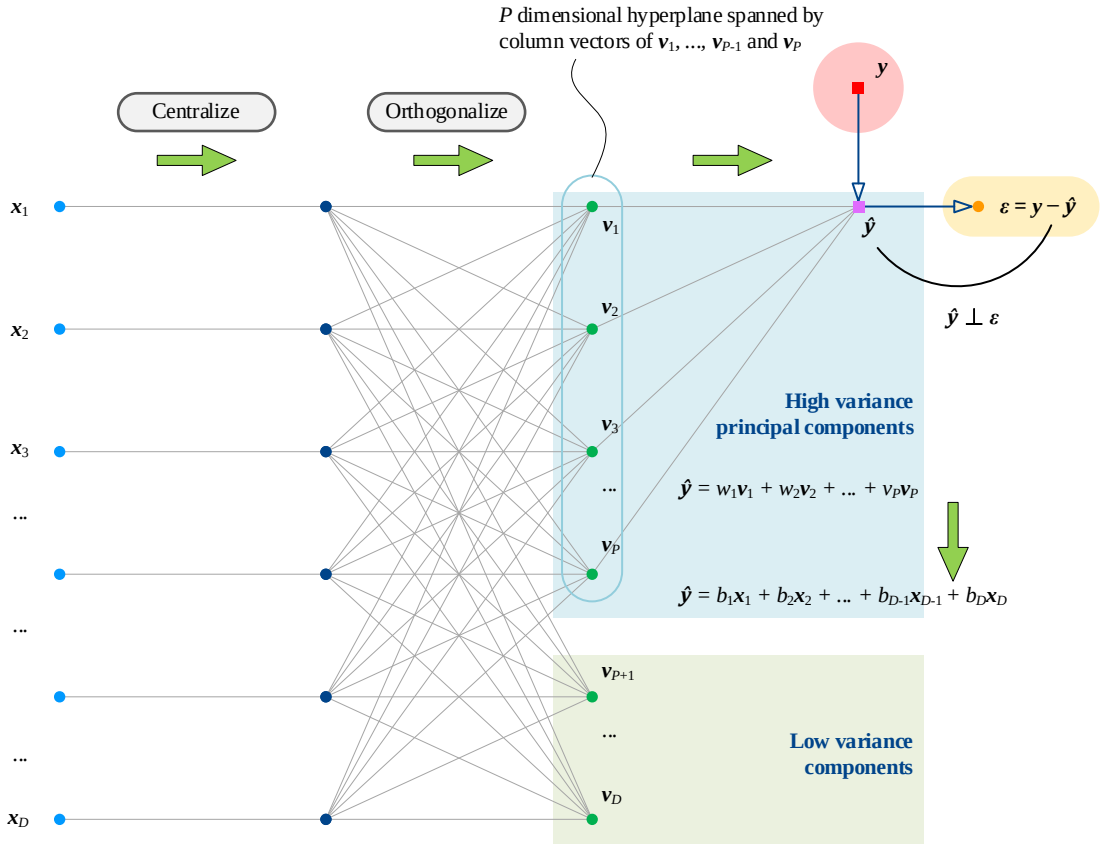


图 18. 主元回归数据关系

17.6 改变主元数量

对于主元回归，当改变参与最小二乘法线性回归的主元数量时，线性回归结果会有很大变化；本节将重点介绍主元数量对主元回归的影响。

图 19 所示为主元数量从 4 增加到 9 时，累计已释方差和百分比变化情况。图 20 和图 21 展示两个视角观察参与主元回归主元数量对于系数的影响。

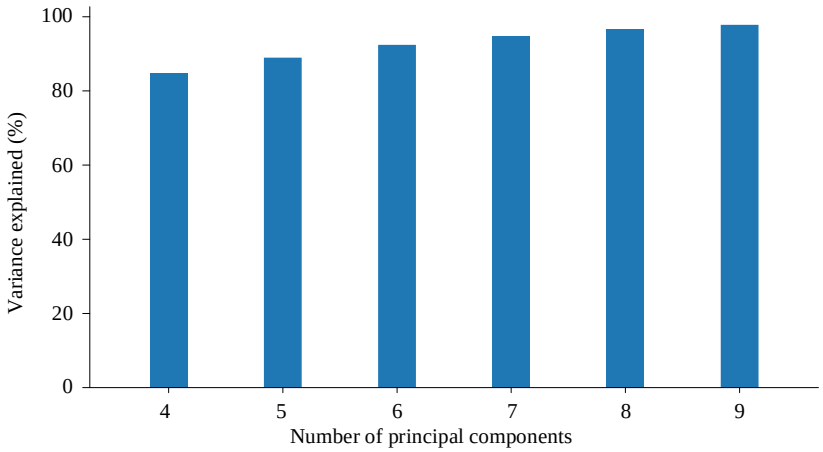


图 19. 主元数量对累计已释方差和百分比

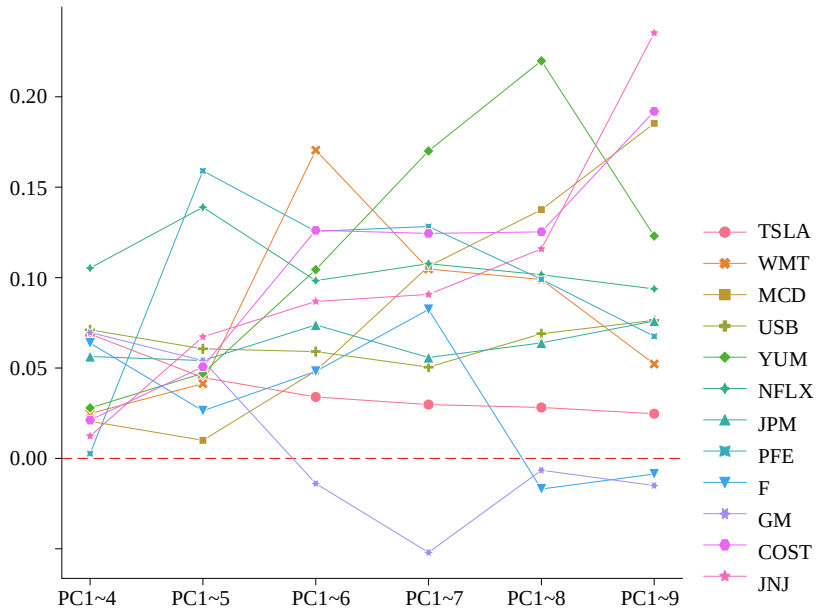


图 20. 参与主元回归主元数量对于系数的影响

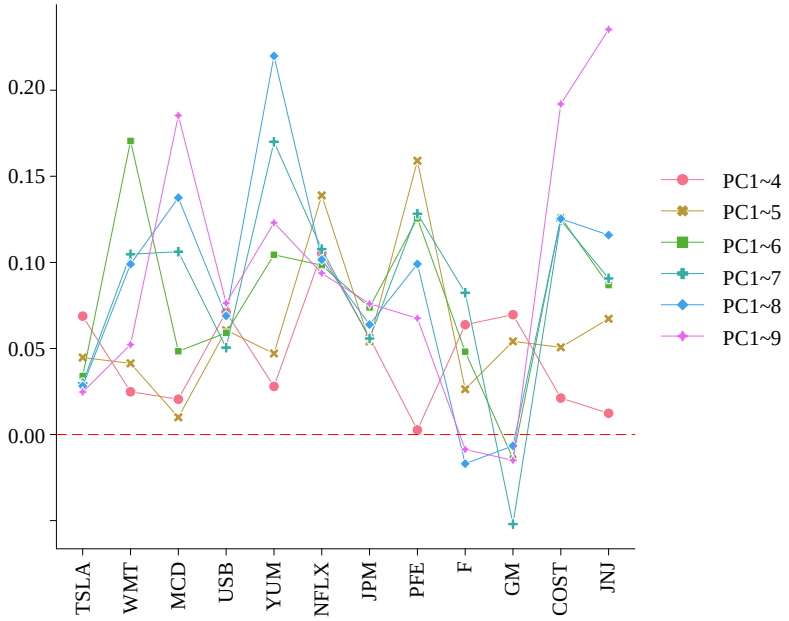


图 21. 参与主元回归主元数量对于系数的影响，第二视角

Bk6_Ch17_01.py 完成主元回归运算图像。

17.7 偏最小二乘回归

本章最后介绍偏最小二乘回归 (partial least squares regression, PLS)。类似主元回归，偏最小二乘回归也是一种降维回归方法。PLS 在降低自变量维度的同时，建立自变量和因变量之间的线性关系模型，因此常被用于处理高维数据分析和建立多元回归模型。

不同于主元回归，偏最小二乘回归利用因变量数据 $\boldsymbol{y}$ 和自变量数据 $\boldsymbol{X}$ (形状为 $n \times q$) 之间相关性构造一个全新空间。$\boldsymbol{y}$ 和 $\boldsymbol{X}$ 投影到新空间来确定一个线性回归模型。另外一个不同点，偏最小二乘回归采用迭代算法 (iterative algorithm)。

偏最小二乘法处理多元因变量，为方便区分，一元因变量被定义为 $\boldsymbol{y}$ (形状为 $n \times 1$)，多元因变量被定义为 $\boldsymbol{Y}$ (形状为 $n \times p$)。偏最小二乘回归迭代方法很多，本节介绍较为经典一元因变量对多元自变量迭代算法。迭代算法主要由七步构成；其中，第二步到第七步为循环。

第一步

获得中心化自变量数据矩阵 $\boldsymbol{X}^{(0)}$ 和因变量数据向量 $\boldsymbol{y}^{(0)}$：

$$\begin{aligned} \boldsymbol{X}^{(0)} &= \left(\boldsymbol{I} - \frac{1}{n}\boldsymbol{l}\boldsymbol{l}^{\mathrm{T}}\right)\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^{(0)} & \boldsymbol{x}_2^{(0)} & \cdots & \boldsymbol{x}_q^{(0)} \end{bmatrix} \\ \boldsymbol{y}^{(0)} &= \boldsymbol{y} - \mathrm{E}(\boldsymbol{y}) = \left(\boldsymbol{I} - \frac{1}{n}\boldsymbol{l}\boldsymbol{l}^{\mathrm{T}}\right)\boldsymbol{y} \end{aligned} \tag{17}$$

偏最小二乘回归是迭代运算，上标 (0) 代表迭代代次。

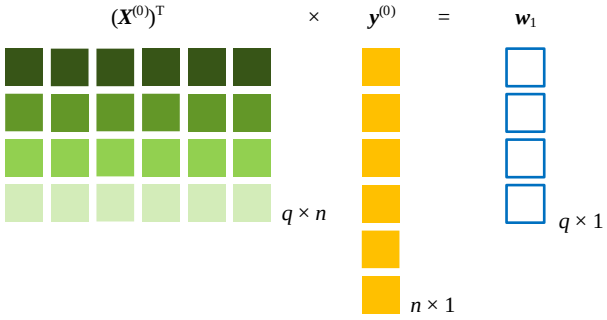


图 22. 计算权重系数列向量 $\boldsymbol{w}_1$

第二步

计算 $\boldsymbol{y}^{(0)}$ 和 $\boldsymbol{X}^{(0)}$ 列向量相关性，构建权重系数列向量 $\boldsymbol{w}_1$：

$$\boldsymbol{w}_1 = \begin{bmatrix} \mathrm{cov}\left(\boldsymbol{x}_1^{(0)}, \boldsymbol{y}^{(0)}\right) \\ \mathrm{cov}\left(\boldsymbol{x}_2^{(0)}, \boldsymbol{y}^{(0)}\right) \\ \vdots \\ \mathrm{cov}\left(\boldsymbol{x}_q^{(0)}, \boldsymbol{y}^{(0)}\right) \end{bmatrix} = \frac{1}{n}\begin{bmatrix} \left(\boldsymbol{x}_1^{(0)}\right)^{\mathrm{T}}\boldsymbol{y}^{(0)} \\ \left(\boldsymbol{x}_2^{(0)}\right)^{\mathrm{T}}\boldsymbol{y}^{(0)} \\ \vdots \\ \left(\boldsymbol{x}_q^{(0)}\right)^{\mathrm{T}}\boldsymbol{y}^{(0)} \end{bmatrix} = \left(\boldsymbol{X}^{(0)}\right)^{\mathrm{T}}\boldsymbol{y}^{(0)} \tag{18}$$

其中，列向量 $\boldsymbol{w}_1$ 行数为 q 行。

图 22 所示获得权重系数列向量计算过程；过程也可看做是一个投影运算，即将 $(\boldsymbol{X}^{(0)})^{\mathrm{T}}$ 投影到 $\boldsymbol{y}^{(0)}$。

为方便计算，将列向量 $\boldsymbol{w}_1$ 单位化：

$$\boldsymbol{w}_1 = \frac{\boldsymbol{w}_1}{\|\boldsymbol{w}_1\|} = \begin{bmatrix} w_{1,1} \\ w_{2,1} \\ \vdots \\ w_{q,1} \end{bmatrix} \tag{19}$$

列向量 $\boldsymbol{w}_1$ 每个元素大小代表着 $\boldsymbol{y}^{(0)}$ 和 $\boldsymbol{X}^{(0)}$ 列向量相关性。

第三步，利用上一步获得权重系数列向量 $\boldsymbol{w}_1$ 和 $\boldsymbol{X}^{(0)}$ 构造偏最小二乘回归主元向量，$\boldsymbol{z}_1$：

$$\boldsymbol{z}_1 = w_{1,1}\boldsymbol{x}_1 + w_{2,1}\boldsymbol{x}_2 + \cdots + w_{q,1}\boldsymbol{x}_q = \boldsymbol{X}^{(0)}\boldsymbol{w}_1 \tag{20}$$

图 23 所示为计算偏最小二程回归主元列向量 z_1。这样理解，主元列向量 z_1 为 $X^{(0)}$ 列向量通过加权构造；$y^{(0)}$ 和 $X^{(0)}$ 某一列向量相关性越高，这一列获得权重越高，在主元列向量 z_1 成分越高。同样，过程等价于投影过程，即 $X^{(0)}$ 投影到 w_1。

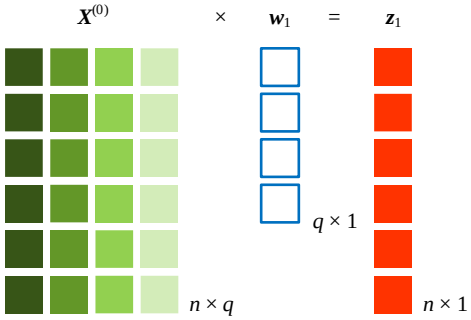


图 23. 计算偏最小二程回归主元列向量 z_1

将自变量数据矩阵 $X^{(0)}$ 和因变量数据向量 $y^{(0)}$投影到主元 z_1 方向上。

第四步

把自变量数据矩阵 $X^{(0)}$投影到主元列向量 z_1 上，获得系数向量 v_1。先以 $X^{(0)}$第一列解释投影过程。

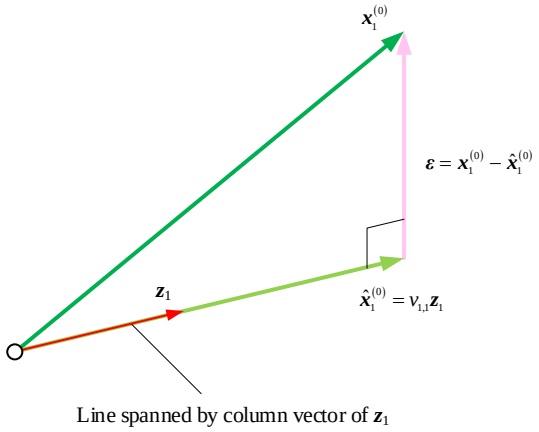


图 24. $X^{(0)}$第一列投影在主元列向量 z_1

如图 24 所示，将 $X^{(0)}$ 第一列投影到主元列向量 z_1，得到 $\hat{x}_1^{(0)}$：

$$\hat{x}_1^{(0)} = v_{1,1} z_1 \tag{21}$$

残差 ε 则垂直于主元列向量 z_1，计算获得系数 $v_{1,1}$：

$$\begin{aligned} &\boldsymbol{\varepsilon} \perp \boldsymbol{z}_1 \quad \Rightarrow \quad \boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{\varepsilon} = \boldsymbol{z}_1^{\mathrm{T}}\left(\boldsymbol{x}_1^{(0)} - \hat{\boldsymbol{x}}_1^{(0)}\right) = \boldsymbol{z}_1^{\mathrm{T}}\left(\boldsymbol{x}_1^{(0)} - v_{1,1}\boldsymbol{z}_1\right) = 0 \\ &\Rightarrow \quad v_{1,1} = \frac{\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{x}_1^{(0)}}{\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1} = \frac{\left(\boldsymbol{x}_1^{(0)}\right)^{\mathrm{T}}\boldsymbol{z}_1}{\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1} \end{aligned} \tag{22}$$

上式说明偏最小二乘法回归核心仍是 OLS。同样，把 $\boldsymbol{X}^{(0)}$ 第二列投影在主元列向量 $\boldsymbol{z}_2$，计算得到系数 $v_{2,1}$：

$$v_{2,1} = \frac{\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{x}_2^{(0)}}{\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1} = \frac{\left(\boldsymbol{x}_2^{(0)}\right)^{\mathrm{T}}\boldsymbol{z}_1}{\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1} \tag{23}$$

类似，获得 $\boldsymbol{X}^{(0)}$ 每列投影在主元列向量 $\boldsymbol{z}_2$ 系数，这些系数一个列向量 $\boldsymbol{v}_1$。下式计算列向量 $\boldsymbol{v}_1$：

$$\boldsymbol{v}_1 = \begin{bmatrix} v_{1,1} \\ v_{2,1} \\ \vdots \\ v_{q,1} \end{bmatrix} = \frac{\left(\boldsymbol{X}^{(0)}\right)^{\mathrm{T}}\boldsymbol{z}_1}{\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1} = \frac{\left(\boldsymbol{X}^{(0)}\right)^{\mathrm{T}}\boldsymbol{X}^{(0)}\boldsymbol{w}_1}{\boldsymbol{w}_1^{\mathrm{T}}\left(\boldsymbol{X}^{(0)}\right)^{\mathrm{T}}\boldsymbol{X}^{(0)}\boldsymbol{w}_1} = \frac{\boldsymbol{\Sigma}^{(0)}\boldsymbol{w}_1}{\boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{\Sigma}^{(0)}\boldsymbol{w}_1} \tag{24}$$

第五步

根据最小二乘回归原理，利用列向量 $\boldsymbol{v}_1$ 和 $\boldsymbol{z}_1$ 估算，并到拟合矩阵 $\hat{\boldsymbol{X}}^{(0)}$：

$$\hat{\boldsymbol{X}}^{(0)} = \boldsymbol{z}_1\boldsymbol{v}_1^{\mathrm{T}} = \boldsymbol{X}^{(0)}\boldsymbol{w}_1\boldsymbol{v}_1^{\mathrm{T}} \tag{25}$$

原始数据矩阵 $\boldsymbol{X}$ 和拟合数据矩阵 $\hat{\boldsymbol{X}}^{(0)}$ 之差便是残差矩阵 $\boldsymbol{E}^{(0)}$：

$$\boldsymbol{E}^{(0)} = \boldsymbol{X}^{(0)} - \hat{\boldsymbol{X}}^{(0)} = \boldsymbol{X}^{(0)} - \boldsymbol{X}^{(0)}\boldsymbol{w}_1\boldsymbol{v}_1^{\mathrm{T}} = \boldsymbol{X}^{(0)}\left(\boldsymbol{I} - \boldsymbol{w}_1\boldsymbol{v}_1^{\mathrm{T}}\right) \tag{26}$$

而残差矩阵 $\boldsymbol{E}^{(0)}$ 便是进入迭代过程第二步数据矩阵 $\boldsymbol{X}^{(1)}$：

$$\boldsymbol{X}^{(1)} = \boldsymbol{E}^{(0)} = \boldsymbol{X}^{(0)} - \hat{\boldsymbol{X}}^{(0)} = \boldsymbol{X}^{(0)}\left(\boldsymbol{I} - \boldsymbol{w}_1\boldsymbol{v}_1^{\mathrm{T}}\right) \tag{27}$$

数据矩阵 $\boldsymbol{X}^{(1)}$ 和原始数据 $\boldsymbol{X}^{(0)}$ 之间关系如图 25 所示。

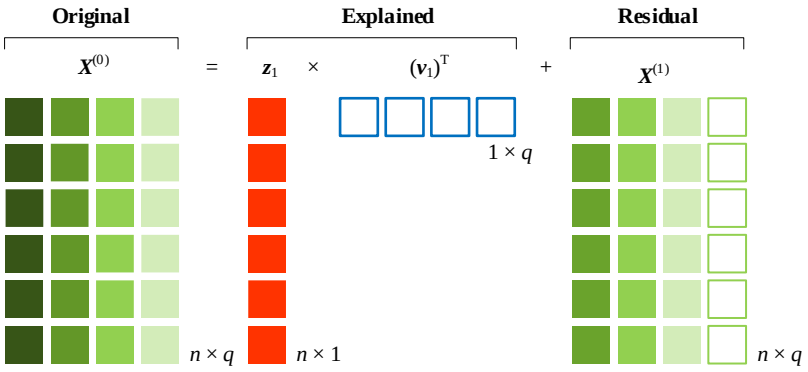


图 25. 计算得到数据矩阵 $\boldsymbol{X}^{(1)}$

第六步

把因变量数据列向量 $\boldsymbol{y}^{(0)}$ 投影于主元列向量 $\boldsymbol{z}_1$ 上，获得系数 b_1。类似第四步，如图 26 所示，用最小二乘法计算获得系数 b_1：

$$\begin{aligned} &\boldsymbol{\varepsilon} \perp \boldsymbol{z}_1 \quad \Rightarrow \quad \boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{\varepsilon} = \boldsymbol{z}_1^{\mathrm{T}}\left(\boldsymbol{y}^{(0)} - \hat{\boldsymbol{y}}^{(0)}\right) = \boldsymbol{z}_1^{\mathrm{T}}\left(\boldsymbol{y}^{(0)} - b_1\boldsymbol{z}_1\right) = 0 \\ &\Rightarrow \quad b_1 = \frac{\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{y}^{(0)}}{\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1} = \frac{\left(\boldsymbol{y}^{(0)}\right)^{\mathrm{T}}\boldsymbol{z}_1}{\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1} \end{aligned} \tag{28}$$

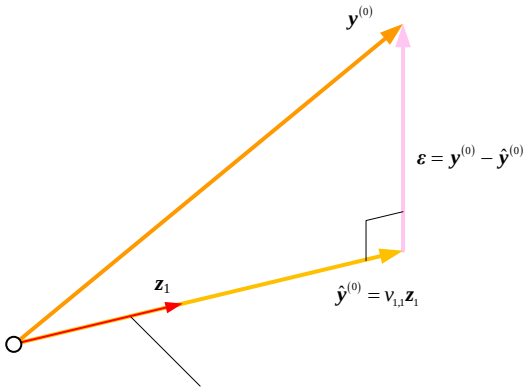


图 26. $\boldsymbol{y}^{(0)}$ 向量投影在主元列向量 $\boldsymbol{z}_1$

第七步

根据 OLS 原理，利用列向量 b_1 和 $\boldsymbol{z}_1$ 估算因变量列向量 $\boldsymbol{y}$，并到拟合 $\hat{\boldsymbol{y}}^{(0)}$：

$$\hat{\boldsymbol{y}}^{(0)} = b_1\boldsymbol{z}_1 = \frac{\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{y}^{(0)}\boldsymbol{z}_1}{\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1} = \frac{\left(\boldsymbol{y}^{(0)}\right)^{\mathrm{T}}\boldsymbol{z}_1\boldsymbol{z}_1}{\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1} \tag{29}$$

原始因变量列向量 $\boldsymbol{y}^{(0)}$ 和拟合列向量 $\hat{\boldsymbol{y}}^{(0)}$ 之差便是残差向量 $\boldsymbol{\varepsilon}^{(0)}$：

$$\boldsymbol{\varepsilon}^{(0)} = \boldsymbol{y}^{(1)} = \boldsymbol{y}^{(0)} - \hat{\boldsymbol{y}}^{(0)} = \boldsymbol{y}^{(0)} - \frac{\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{y}^{(0)}\boldsymbol{z}_1}{\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1} \tag{30}$$

而残差向量 $\boldsymbol{\varepsilon}^{(0)}$ 便是进入迭代循环第二步数据向量 $\boldsymbol{y}^{(1)}$。如图 27 所示，$\hat{\boldsymbol{y}}^{(0)}$ 解释部分 $\boldsymbol{y}^{(0)}$。

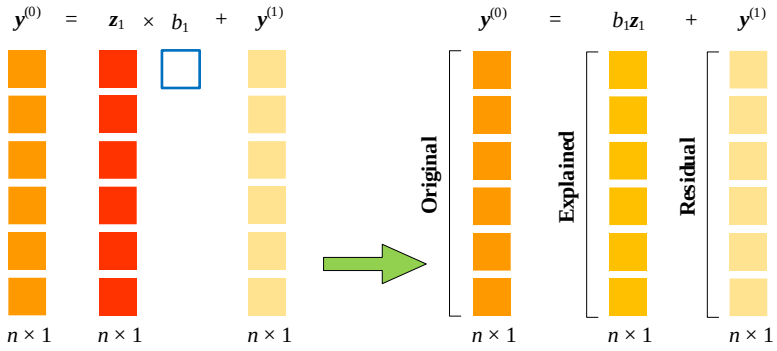


图 27. 估算 $\boldsymbol{y}^{(0)}$

重复迭代

将数据矩阵 $\boldsymbol{X}^{(1)}$ 和数据向量 $\boldsymbol{y}^{(1)}$ 带入如上迭代运算第二步到第七步。

重复第二步得到权重系数列向量 $\boldsymbol{w}_2$：

$$\boldsymbol{w}_2 = \frac{\left(\boldsymbol{X}^{(1)}\right)^{\mathrm{T}} \boldsymbol{y}^{(1)}}{\left\| \left(\boldsymbol{X}^{(1)}\right)^{\mathrm{T}} \boldsymbol{y}^{(1)} \right\|} \tag{31}$$

重复第三步，利用权重系数列向量 $\boldsymbol{w}_2$ 和 $\boldsymbol{X}^{(1)}$ 构造偏最小二乘回归第二主元向量，$\boldsymbol{z}_2$：

$$\boldsymbol{z}_2 = \boldsymbol{X}^{(1)} \boldsymbol{w}_2 \tag{32}$$

重复第四步，把自变量数据残差矩阵 $\boldsymbol{X}^{(1)}$ 投影于第二主元列向量 $\boldsymbol{z}_2$ 上，获得系数向量 $\boldsymbol{v}_2$：

$$\boldsymbol{v}_2 = \begin{bmatrix} v_{1,2} \\ v_{2,2} \\ \vdots \\ v_{q,2} \end{bmatrix} = \frac{\left(\boldsymbol{X}^{(1)}\right)^{\mathrm{T}} \boldsymbol{z}_2}{\boldsymbol{z}_2^{\mathrm{T}} \boldsymbol{z}_2} = \frac{\left(\boldsymbol{X}^{(1)}\right)^{\mathrm{T}} \boldsymbol{X}^{(1)} \boldsymbol{w}_2}{\boldsymbol{w}_2^{\mathrm{T}} \left(\boldsymbol{X}^{(1)}\right)^{\mathrm{T}} \boldsymbol{X}^{(1)} \boldsymbol{w}_2} = \frac{\boldsymbol{\Sigma}^{(1)} \boldsymbol{w}_2}{\boldsymbol{w}_2^{\mathrm{T}} \boldsymbol{\Sigma}^{(1)} \boldsymbol{w}_2} \tag{33}$$

重复第五步，用列向量 $\boldsymbol{v}_2$ 和 $\boldsymbol{z}_2$ 估算，并到拟合矩阵 $\hat{\boldsymbol{X}}^{(1)}$：

$$\hat{\boldsymbol{X}}^{(1)} = \boldsymbol{z}_2 \boldsymbol{v}_2^{\mathrm{T}} = \boldsymbol{X}^{(1)} \boldsymbol{w}_2 \boldsymbol{v}_2^{\mathrm{T}} \tag{34}$$

$\boldsymbol{X}^{(1)}$ 和拟合数据矩阵 $\hat{\boldsymbol{X}}^{(1)}$ 之差便是残差矩阵 $\boldsymbol{E}^{(1)}$，$\boldsymbol{E}^{(1)}$便是再次进入迭代过程第二步数据矩阵 $\boldsymbol{X}^{(2)}$：

$$\boldsymbol{X}^{(2)} = \boldsymbol{E}^{(1)} = \boldsymbol{X}^{(1)} - \hat{\boldsymbol{X}}^{(1)} = \boldsymbol{X}^{(1)} \left(\boldsymbol{I} - \boldsymbol{w}_2 \boldsymbol{v}_2^{\mathrm{T}}\right) \tag{35}$$

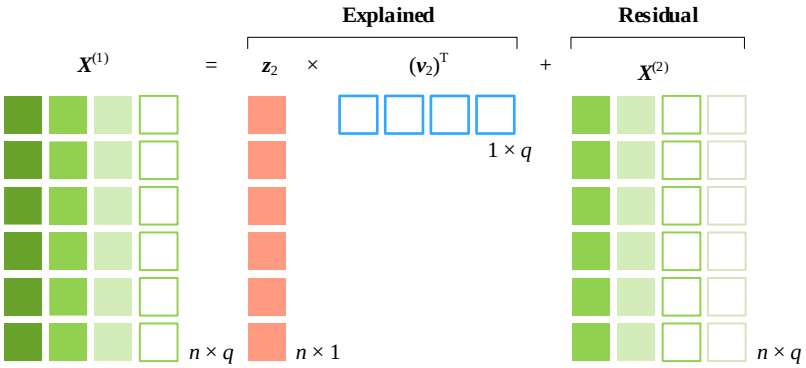


图 28. 计算得到数据矩阵 $X^{(2)}$

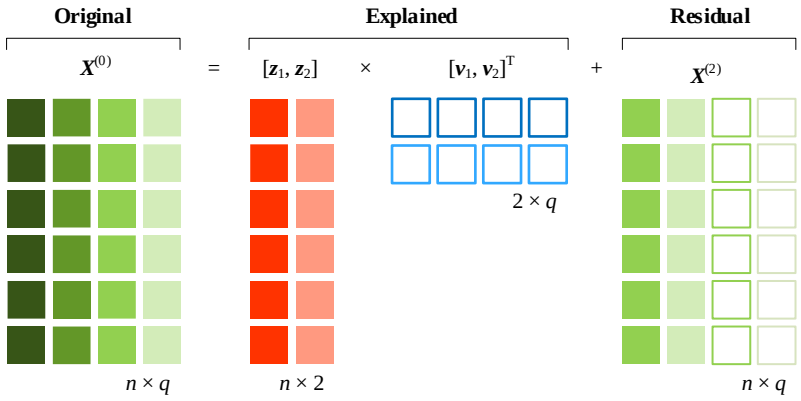


图 29. 前两个主元 z_1 和 z_1 还原数据矩阵 $X^{(0)}$

图 25 和图 28 相结合获得图 29，这即前两个主元 z_1 和 z_1 还原数据矩阵 $X^{(0)}$。随着主元数量不断增多，偏最小二乘回归更精确地还原原始数据 $X^{(0)}$；即说，对数据 $X^{(0)}$方差解释力度越强。

重复第六步，把因变量数据列向量 $y^{(1)}$ 投影在主元列向量 z_2 上，获得系数 b_2：

$$b_2 = \frac{z_2^T y^{(1)}}{z_2^T z_2} = \frac{\left(y^{(1)}\right)^T z_2}{z_2^T z_2} \tag{36}$$

重复第七步，利用 b_2 和 z_2 得到拟合列向量 $\hat{y}^{(1)}$：

$$\hat{y}^{(1)} = b_2 z_2 \tag{37}$$

列向量 $y^{(1)}$ 和拟合数据列向量 $\hat{y}^{(1)}$ 之差便是残差向量 $\varepsilon^{(1)}$：

$$\varepsilon^{(0)} = y^{(2)} = y^{(1)} - \hat{y}^{(1)} = y^{(1)} - b_2 z_2 \tag{38}$$

而残差向量 $\varepsilon^{(1)}$ 也是进入下一次迭代过程第二步数据向量 $y^{(2)}$。

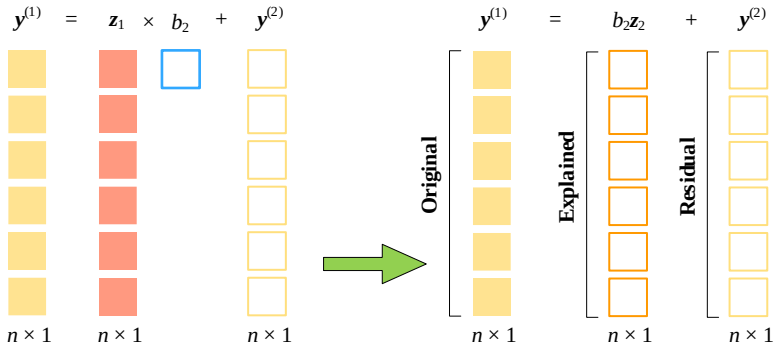


图 30. 估算 $y^{(1)}$

图 31 结合图 27 和图 30，这幅图中前两个主元 z_1 和 z_1 还原部分数据列向量 $y^{(0)}$。同理，随着主元数量不断增多，偏最小二乘回归更精确地还原原始因变量列向量 $y^{(0)}$；即，对 $y^{(0)}$方差解释力度越强。截止目前，迭代循环已经完成两次。

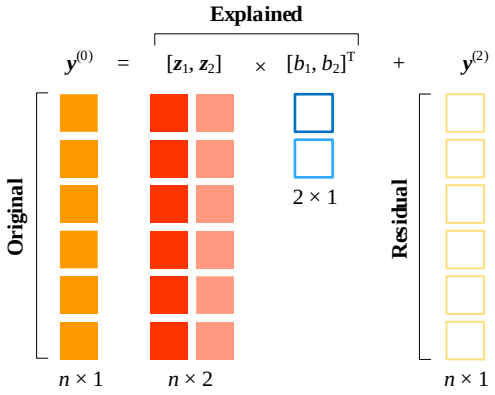


图 31. 前两个主元 z_1 和 z_1 还原部分数据列向量 $y^{(0)}$

Scikit-learn 中 PLS 回归的函数为 sklearn.cross_decomposition.PLSRegression()。

主元回归 PCR 是一种基于主成分分析的回归方法，它在回归建模之前，先对自变量进行主成分分析，将自变量降维成少量的主成分变量，然后再对这些主成分变量进行回归分析。

PCR 的基本思想是将自变量通过主成分分析转换成少数互相正交的主成分变量，从而消除自变量之间的多重共线性问题，提高回归分析的准确性和稳定性。在降维过程中，PCR 保留了自变量中最主要的信息，因此相比于直接使用全部自变量的回归分析，PCR 可以显著提高回归模型的准确性和可解释性。

偏最小二乘 PLS 也是一种基于主成分分析和回归分析的统计建模方法，它是对 PCR 的一种改进，主要用于解决多重共线性和高维数据分析问题。

与 PCR 不同的是，PLS 在主成分分析的过程中，不仅仅考虑了自变量之间的方差，还考虑了自变量和因变量之间的协方差，从而将主成分分析与回归分析相结合，得到了一组互相正交的主成分变量，每个主成分变量都包含了自变量和因变量的信息，可以用于回归分析。

下例展示如何使用偏最小二乘回归。这个例子还比较了本书最后一章要介绍的典型相关分析。请大家自行阅读学习：

https://scikit-learn.org/stable/auto_examples/cross_decomposition/plot_compare_cross_decomposition.html

18 典型相关分析

Canonical Correlation Analysis

找到两组数据的整体相关性的最大线性组合

人类生而好奇，这正是科学的火种。

Men love to wonder, and that is the seed of science.

—— 拉尔夫·爱默生 (Ralph Waldo Emerson) | 美国思想家、文学家 | 1803 ~ 1882

- `numpy.linalg.eig()` 特征值分解
- `numpy.linalg.inv()` 矩阵求逆
- `seaborn.heatmap()` 绘制热图
- `seaborn.jointplot()` 绘制散点图，含边缘分布
- `seaborn.pairplot()` 成对散点图
- `seaborn.scatterplot()` 绘制散点图
- `sklearn.cross_decomposition.CCA()` 典型相关分析

18.1 典型相关分析原理

典型相关分析 (Canonical Correlation Analysis, CCA) 是一种用于探究两组变量之间关系的多元统计分析方法。其核心思想是将两组变量分别投影到新的低维空间中，使得这两组变量在新空间中的投影尽可能相关。

CCA 常用于处理两组多元变量之间的关系。通过 CCA 可以发现这两组变量中的某些维度之间存在相关性，这种相关性可以帮助研究者更好地理解两组变量之间的关系。

在 CCA 中，研究者需要先对两组变量进行标准化处理，然后计算它们的相关系数矩阵。接着，CCA 会生成一组线性组合，使得两组变量在新的低维空间中的投影尽可能相关。这些线性组合称为典型变量，相关系数则称为典型相关系数。最终的结果是一组典型变量和对应的典型相关系数。

原理

下面以 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 为例介绍典型相关分析原理。

$n \times p$ 数据矩阵 $\boldsymbol{X}$ 可以写成：

$$\boldsymbol{X}_{n\times p} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_p \end{bmatrix} \tag{1}$$

$n \times q$ 数据矩阵 $\boldsymbol{Y}$ 可以写成：

$$\boldsymbol{Y}_{n\times q} = \begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_q \end{bmatrix} \tag{2}$$

⚠ 注意，$\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 的行数一致。

$\boldsymbol{X}$ 朝向量 $\boldsymbol{u}_1$ 投影结果为 $\boldsymbol{s}_1$：

$$\boldsymbol{s}_1 = \boldsymbol{X}_{n\times p}\boldsymbol{u}_1 \tag{3}$$

其中，$\boldsymbol{u}_1$ 的形状为 $p \times 1$，$\boldsymbol{s}_1$ 的形状为 $n \times 1$。

⚠ 注意，很多参考文献中，向量一般记做 $\boldsymbol{a}$ 和 $\boldsymbol{b}$，投影结果一般记做 $\boldsymbol{u}$ 和 $\boldsymbol{v}$；但是本书 $\boldsymbol{u}$ 和 $\boldsymbol{v}$ 特指代表投影方向的向量，所以本章依然沿用这种记法。

展开 (3) 得到如下线性组合形式：

$$\boldsymbol{s}_1 = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_p \end{bmatrix} \begin{bmatrix} u_{1,1} \\ u_{2,1} \\ \vdots \\ u_{p,1} \end{bmatrix} = u_{1,1}\boldsymbol{x}_1 + u_{2,1}\boldsymbol{x}_2 + \cdots u_{p,1}\boldsymbol{x}_p \tag{4}$$

$\boldsymbol{Y}$ 朝向量 $\boldsymbol{v}_1$ 投影结果为 $\boldsymbol{t}_1$：

$$\boldsymbol{t}_1 = \boldsymbol{Y}_{n\times q}\boldsymbol{v}_1 \tag{5}$$

其中，$\boldsymbol{v}_1$ 的形状为 $q \times 1$，$\boldsymbol{t}_1$ 的形状为 $n \times 1$。p 和 q 可以不相等，也就是说 $\boldsymbol{u}_1$、$\boldsymbol{v}_1$ 形状可能不同。但是 $\boldsymbol{s}_1$、$\boldsymbol{t}_1$ 形状相同。

展开 (5) 得到如下线性组合形式：

$$\boldsymbol{t}_1 = \begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_q \end{bmatrix} \begin{bmatrix} v_{1,1} \\ v_{2,1} \\ \vdots \\ v_{q,1} \end{bmatrix} = v_{1,1}\boldsymbol{y}_1 + v_{2,1}\boldsymbol{y}_2 + \cdots v_{q,1}\boldsymbol{y}_q \tag{6}$$

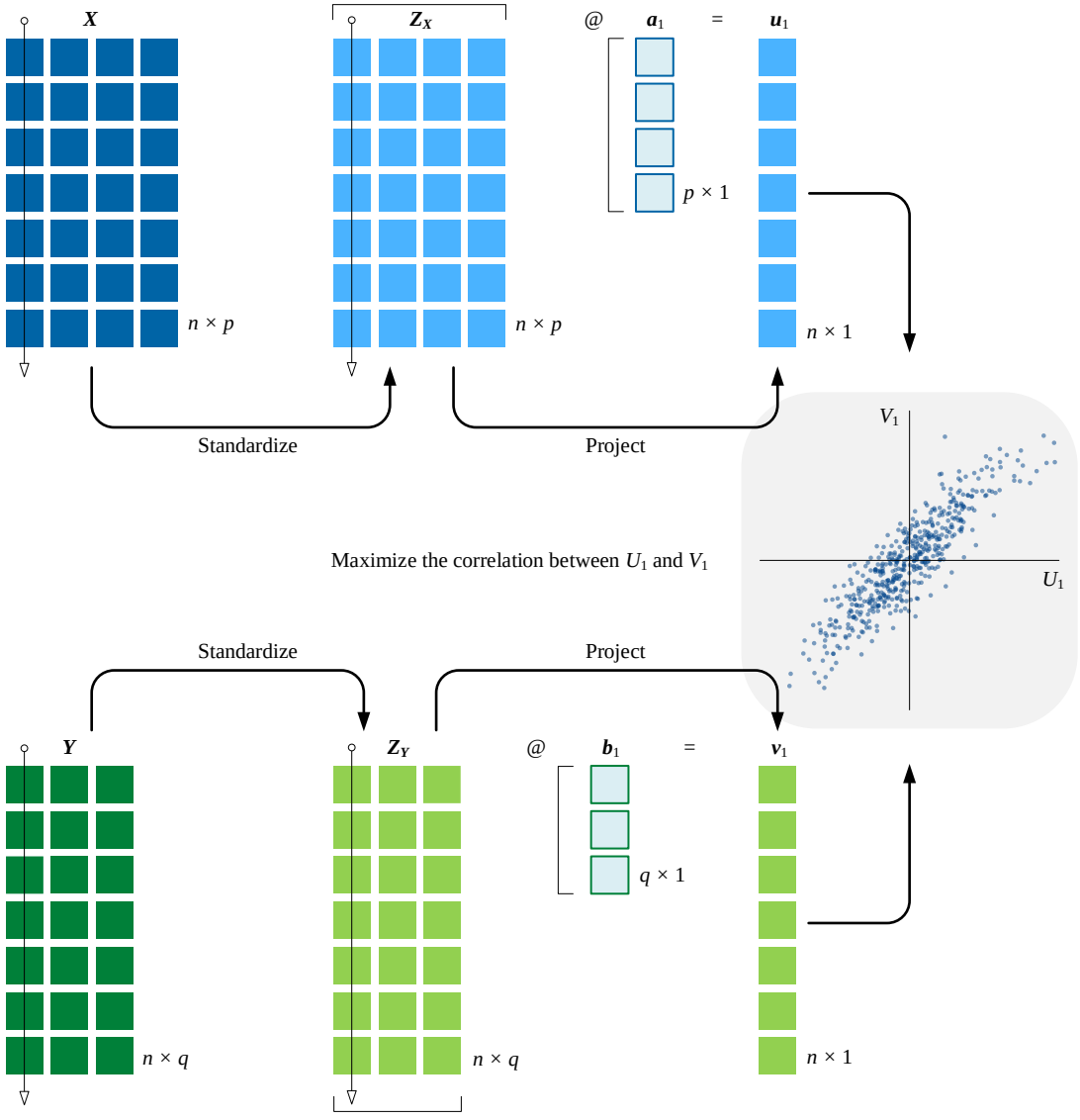


图 1. 典型相关分析原理

优化问题

如图 1 所示，典型相关分析 CCA 的问题便是找到 $\boldsymbol{u}_1$ 和 $\boldsymbol{v}_1$，使得 $\boldsymbol{s}_1$ 和 $\boldsymbol{t}_1$ 相关性最大。

⚠ 注意，如图 1 所示，从数据角度来看，一般情况 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 都先经过标准化处理。

随机变量

用随机变量来写的话，S_1 对应 $\boldsymbol{s}_1$，T_1 对应 $\boldsymbol{t}_1$。随机变量 S_1 可以写成如下线性变换：

$$S_1 = \boldsymbol{u}_1^{\mathrm{T}} \boldsymbol{\chi} = \begin{bmatrix} u_{1,1} & u_{2,1} & \cdots & u_{p,1} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix} = u_{1,1} X_1 + u_{2,1} X_2 + \cdots + u_{p,1} X_p \tag{7}$$

同理，随机变量 T_1 可以写成：

$$T_1 = \boldsymbol{v}_1^{\mathrm{T}} \boldsymbol{\gamma} = \begin{bmatrix} v_{1,1} & v_{2,1} & \cdots & v_{q,1} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_q \end{bmatrix} = v_{1,1} Y_1 + v_{2,1} Y_2 + \cdots + v_{q,1} Y_q \tag{8}$$

S_1 和 T_1 是**第一对典型变量** (first pair of canonical variables)。

S_1 和 T_1 的相关性系数为：

$$\operatorname{corr}(S_1, T_1) = \frac{\operatorname{cov}(S_1, T_1)}{\sqrt{\operatorname{var}(S_1, S_1)} \sqrt{\operatorname{var}(T_1, T_1)}} \tag{9}$$

这样寻找第一对典型变量的优化问题可以写成：

$$\underset{\boldsymbol{u}_1, \boldsymbol{v}_1}{\operatorname{argmax}} \operatorname{corr}(S_1, T_1) \tag{10}$$

有关随机变量的线性变换，请大家回顾《统计至简》第 14 章。

寻找更多典型变量

如图 2 所示，再找到第一对典型变量之后，依然最大化相关性系数可以找到**第二对典型变量** (second pair of canonical variables)。约束条件是第一、第二对典型变量不相关。

用向量来写，$\boldsymbol{s}_2$ 也是 $\begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_p \end{bmatrix}$ 的线性组合：

$$\boldsymbol{s}_2 = \boldsymbol{X}\boldsymbol{u}_2 = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_p \end{bmatrix} \begin{bmatrix} u_{1,2} \\ u_{2,2} \\ \vdots \\ u_{p,2} \end{bmatrix} = u_{1,1}\boldsymbol{x}_1 + u_{2,1}\boldsymbol{x}_2 + \cdots u_{p,1}\boldsymbol{x}_p \tag{11}$$

上式相当于 $\boldsymbol{X}$ 朝 $\boldsymbol{u}_2$ 投影。

$\boldsymbol{t}_2$ 为 $\begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_q \end{bmatrix}$ 的线性组合：

$$\boldsymbol{t}_2 = \begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_q \end{bmatrix} \begin{bmatrix} v_{1,2} \\ v_{2,2} \\ \vdots \\ v_{q,2} \end{bmatrix} = v_{1,2}\boldsymbol{y}_1 + v_{2,2}\boldsymbol{y}_2 + \cdots v_{q,2}\boldsymbol{y}_q \tag{12}$$

上式相当于 $\boldsymbol{Y}$ 朝 $\boldsymbol{v}_2$ 投影。

通过最大化的 $\boldsymbol{s}_2$ 和 $\boldsymbol{t}_2$ 相关性系数，可以找到第二对典型变量。这步优化问题的约束条件为：

$$\begin{aligned} \boldsymbol{u}_1^{\mathrm{T}}\boldsymbol{u}_2 &= 0 \\ \boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{v}_2 &= 0 \\ \boldsymbol{u}_1^{\mathrm{T}}\boldsymbol{v}_2 &= 0 \\ \boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{u}_2 &= 0 \end{aligned} \tag{13}$$

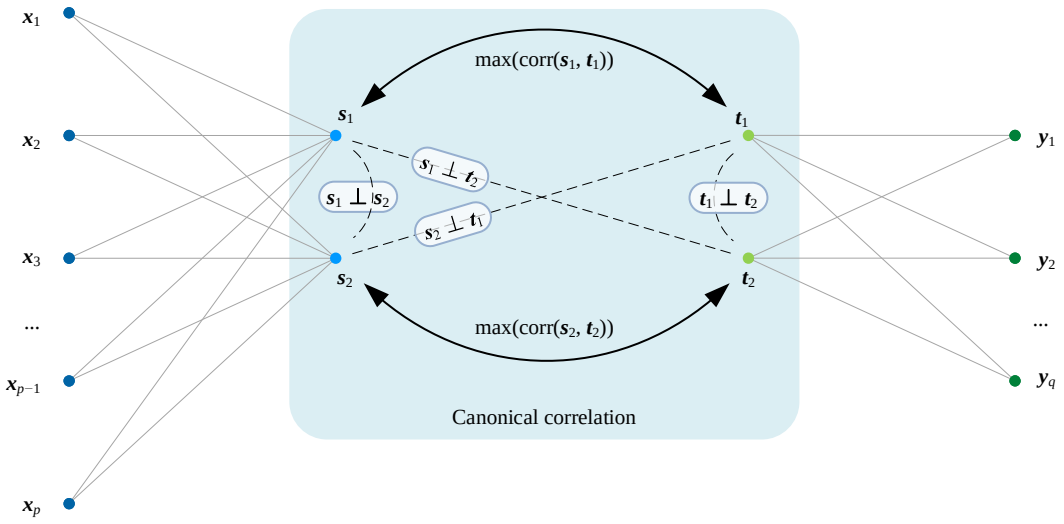


图 2. 线性组合角度看 CCA

随机变量 S_2 可以写成：

$$S_2 = \boldsymbol{u}_2^{\mathrm{T}} \boldsymbol{\chi} = \begin{bmatrix} u_{1,2} & u_{2,2} & \cdots & u_{p,2} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix} = u_{1,2} X_1 + u_{2,2} X_2 + \cdots + u_{p,2} X_p \tag{14}$$

随机变量 T_2 可以写成：

$$T_2 = \boldsymbol{v}_2^{\mathrm{T}} \boldsymbol{\gamma} = \begin{bmatrix} v_{1,2} & v_{2,2} & \cdots & v_{q,2} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_q \end{bmatrix} = v_{1,2} Y_1 + v_{2,2} Y_2 + \cdots + v_{p,2} Y_q \tag{15}$$

同理，为了求解 U_2 和 V_2，约束条件为：

$$\begin{aligned} \operatorname{cov}(U_1, U_2) &= 0 \\ \operatorname{cov}(V_1, V_2) &= 0 \\ \operatorname{cov}(U_1, V_2) &= 0 \\ \operatorname{cov}(V_1, U_2) &= 0 \end{aligned} \tag{16}$$

考虑到一般情况下 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 已经标准化，$E(\boldsymbol{X}) = \boldsymbol{0}$ 且 $E(\boldsymbol{Y}) = \boldsymbol{0}$。这样 $E(U_1) = 0$，$E(V_1) = 0$。

这个步骤最多重复 $\min(p, q)$ 次，可以最多找到 $\min(p, q)$ 对典型变量。$\min(p, q)$ 对应 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 的列数最小值。

18.2 从一个协方差矩阵考虑

《统计至简》第 13 章特别介绍过协方差矩阵分块。

$[\boldsymbol{X}, \boldsymbol{Y}]$ 的协方差矩阵可以按图 3 所示形式分成四个子块。$\boldsymbol{\Sigma_{XX}}$ 为 $\boldsymbol{X}$ 的协方差矩阵，$\boldsymbol{\Sigma_{YY}}$ 为 $\boldsymbol{Y}$ 的协方差矩阵，它俩都是方阵。$\boldsymbol{\Sigma_{XY}}$、$\boldsymbol{\Sigma_{YX}}$ 都是 $\boldsymbol{X}$、$\boldsymbol{Y}$ 的**互协方差矩阵** (cross-covariance matrix)，它俩互为转置。

S_1 和 T_1 各自的方差、协方差为：

$$\begin{aligned} \operatorname{var}(S_1, T_1) &= \boldsymbol{u}_1^{\mathrm{T}} \boldsymbol{\Sigma_{XX}} \boldsymbol{u}_1 \\ \operatorname{var}(S_1, T_1) &= \boldsymbol{v}_1^{\mathrm{T}} \boldsymbol{\Sigma_{YY}} \boldsymbol{v}_1 \\ \operatorname{cov}(S_1, T_1) &= \boldsymbol{u}_1^{\mathrm{T}} \boldsymbol{\Sigma_{XY}} \boldsymbol{v}_1 \end{aligned} \tag{17}$$

如果大家对上式概念模糊的话，请回顾《统计至简》第 14 章。

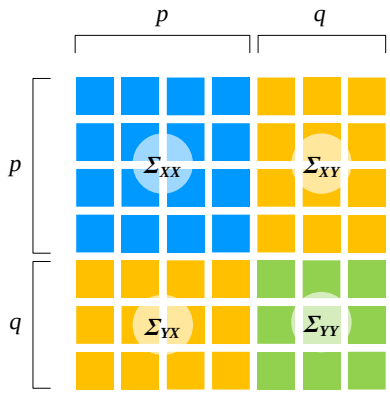


图 3. $[\boldsymbol{X}, \boldsymbol{Y}]$ 的协方差矩阵分块

这样，(9) 的相关性系数可以写成：

$$\operatorname{corr}(S_1, T_1) = \frac{\boldsymbol{u}_1^{\mathrm{T}} \boldsymbol{\Sigma}_{XY} \boldsymbol{v}_1}{\sqrt{\boldsymbol{u}_1^{\mathrm{T}} \boldsymbol{\Sigma}_{XX} \boldsymbol{u}_1} \sqrt{\boldsymbol{v}_1^{\mathrm{T}} \boldsymbol{\Sigma}_{YY} \boldsymbol{v}_1}} \tag{18}$$

观察上式，大家是否发现它实际上是个**瑞利商** (Rayleigh quotient)。

我们在《矩阵力量》第 14 章了解过瑞利商。

优化结果

利用拉格朗日乘子法，我们可以求得优化问题的解。此处，省略推导过程，直接给出结果。

向量 $\boldsymbol{u}$ 是 $\boldsymbol{P} = \boldsymbol{\Sigma}_{XX}^{-1} \boldsymbol{\Sigma}_{XY} \boldsymbol{\Sigma}_{YY}^{-1} \boldsymbol{\Sigma}_{YX}$ 的特征向量。如图 4 所示，$\boldsymbol{P}$ 为 $p \times p$ 方阵。

向量 $\boldsymbol{v}$ 是 $\boldsymbol{Q} = \boldsymbol{\Sigma}_{YY}^{-1} \boldsymbol{\Sigma}_{YX} \boldsymbol{\Sigma}_{XX}^{-1} \boldsymbol{\Sigma}_{XY}$ 的特征向量。如图 5 所示，$\boldsymbol{Q}$ 为 $q \times q$ 方阵。

值得大家注意的是，如图 1 所示，一般 CCA 算法中，数据先要经过标准化处理。也就是说图 3 中真正参与运算的是相关性系数矩阵，而非协方差矩阵。

本章下面要使用的 sklearn.cross_decomposition.CCA() 函数就是先对数据标准化，再进行 CCA 分析。

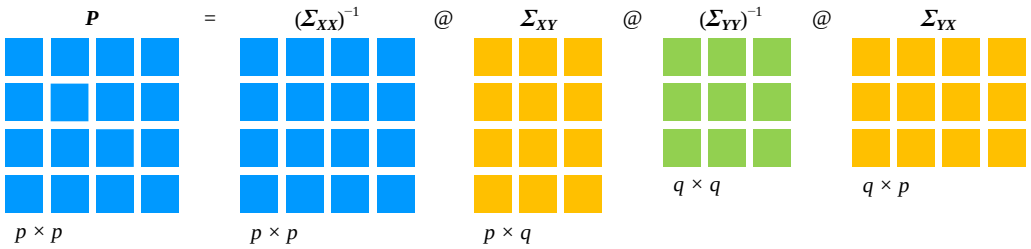


图 4. $\boldsymbol{\Sigma}_{XX}^{-1} \boldsymbol{\Sigma}_{XY} \boldsymbol{\Sigma}_{YY}^{-1} \boldsymbol{\Sigma}_{YX}$ 对应运算

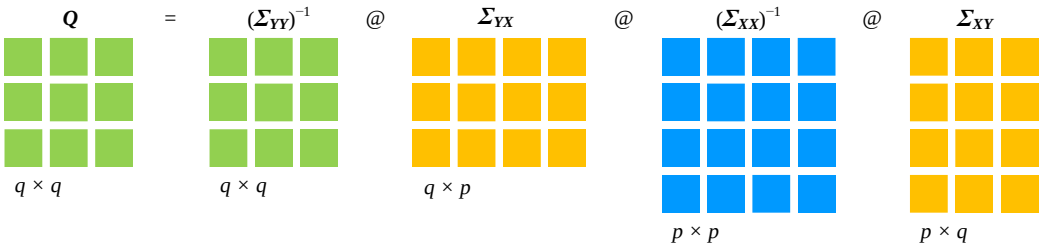


图 5. $\boldsymbol{\Sigma}_{YY}^{-1}\boldsymbol{\Sigma}_{YX}\boldsymbol{\Sigma}_{XX}^{-1}\boldsymbol{\Sigma}_{XY}$ 对应运算

18.3 以鸢尾花数据为例

本节以鸢尾花数据为例介绍如何完成典型相关分析。

如所示，我们把鸢尾花数据 4 列均分为 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 两个矩阵。$\boldsymbol{X}$ 代表花萼 (长度、宽度)，$\boldsymbol{Y}$ 代表花瓣 (长度、宽度)。

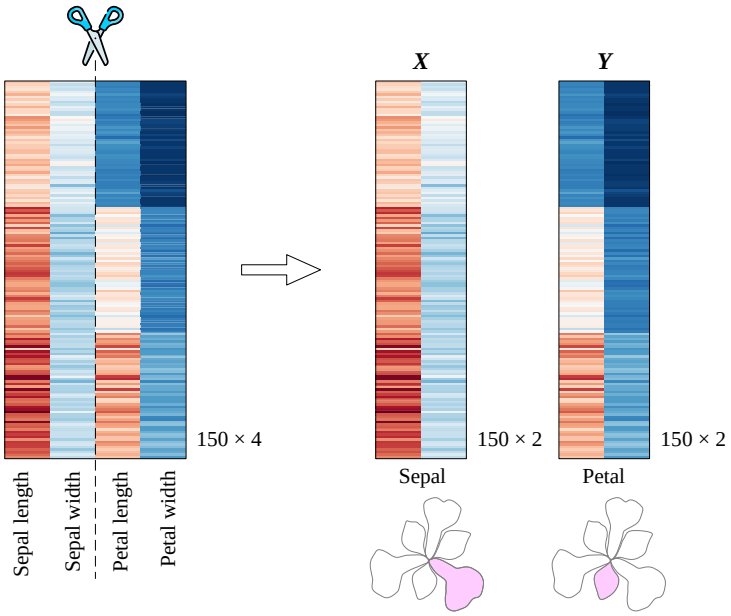


图 6. 把鸢尾花数据均分成两个子块

典型相关分析就是，将花萼数据 $\boldsymbol{X}$ 的两列合成一列 $\boldsymbol{s}_1$，将花瓣数据 $\boldsymbol{Y}$ 的两列合成一列 $\boldsymbol{t}_1$。通过合适的组合方式，让 $\boldsymbol{s}_1$ 和 $\boldsymbol{t}_1$ 的相关性最大。可以理解为找到花萼、花瓣之间“整体”关系。

图 7 所示为鸢尾花数据的相关性系数矩阵。请大家特别关注热图中黄色框高亮的两个子块，花萼和花瓣之间最大的相关性存在于花萼长度和花瓣长度 (0.87)。

比 0.87 更大的相关性系数是 0.96，这个相关性系数是花瓣长度、宽度之间的关系，而非花萼、花瓣之间的关系。

此外，CCA 分析中，图 7 的相关性系数矩阵就相当于图 3 的协方差矩阵。

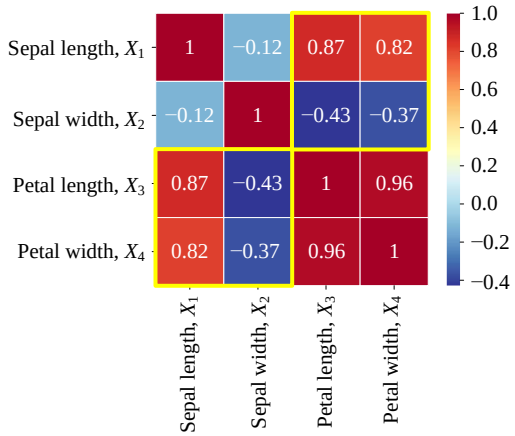


图 7. 鸢尾花数据的相关性系数矩阵

CCA 结果

通过 CCA 分析，我们得到的结果如图 8 (a) 所示。大家可以在本章代码中自行验算，可以发现图 8 (a) 中每一列均值均为 0。

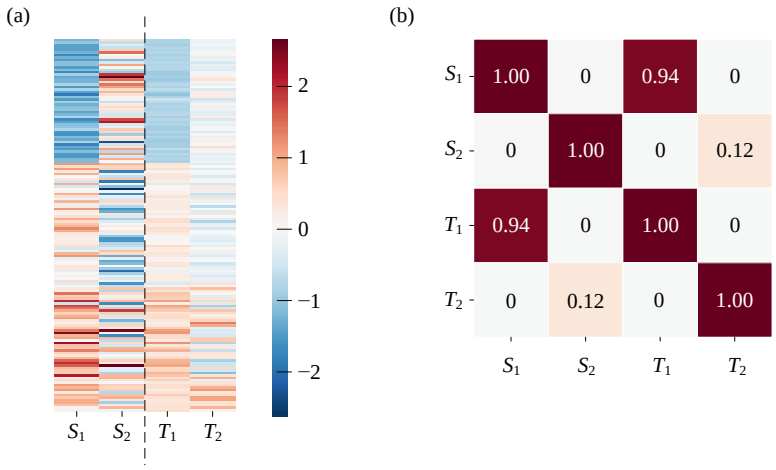


图 8. CCA 分析结果

图 8 (b) 所示为图 8 (a) 结果的相关性系数矩阵。S_1 和 T_1 的相关性系数达到 0.94。此外，大家发现图 8 (b) 中很多相关性系数为 0 的情况，这就是本章前文介绍的优化问题约束条件。

图 9 所示为用散点图可视化 S_1 和 T_1 的关系。图 9 (b) 还考虑了鸢尾花分类。观察图 9 (a)，大家可能已经发现 S_1 和 T_1 均方差明显不同。

图 10 所示为 CCA 结果成对特征散点图。

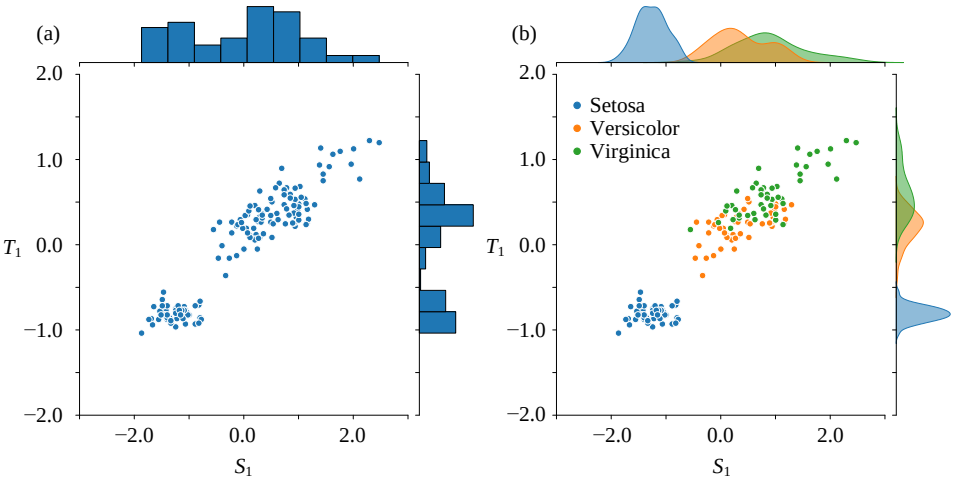


图 9. S_1 和 T_1 的散点图

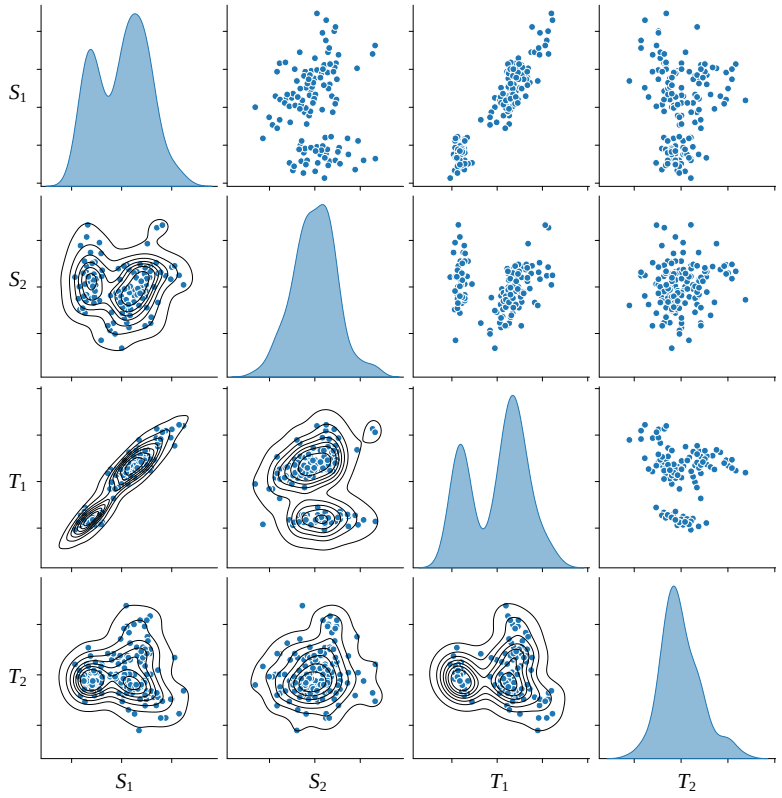


图 10. CCA 结果成对特征散点图

投影

大家可能会好奇到底怎样的 $\boldsymbol{u}_1$、$\boldsymbol{v}_1$ 让 S_1 和 T_1 的相关性系数如此之大？

sklearn.cross_decomposition.CCA() 函数同样返回 $\boldsymbol{u}_1$、$\boldsymbol{v}_1$，具体如图 11 所示。

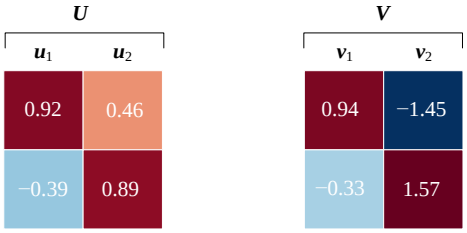


图 11. CCA 投影向量结果

假设 $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2]$ 已经标准化，$\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$ 按如下方式线性组合得到 $\boldsymbol{s}_1$：

$$\boldsymbol{s}_1 = \boldsymbol{X}_{150\times 2}\boldsymbol{u}_1 = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 \end{bmatrix} \begin{bmatrix} 0.92 \\ -0.39 \end{bmatrix} = 0.92\boldsymbol{x}_1 - 0.39\boldsymbol{x}_2 \tag{19}$$

大家可以自己验证 $\boldsymbol{u}_1$ 为单位向量。

同样，假设 $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2]$ 已经标准化，$\boldsymbol{y}_1$ 和 $\boldsymbol{y}_2$ 按如下方式线性组合得到 $\boldsymbol{t}_1$：

$$\boldsymbol{t}_1 = \boldsymbol{Y}_{150\times 2}\boldsymbol{v}_1 = \begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 \end{bmatrix} \begin{bmatrix} 0.94 \\ -0.33 \end{bmatrix} = 0.94\boldsymbol{x}_1 - 0.33\boldsymbol{x}_2 \tag{20}$$

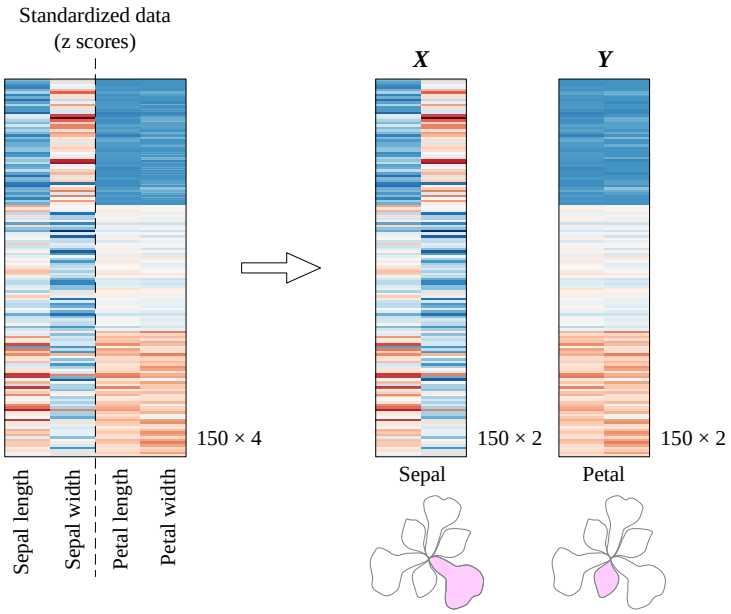


图 12. 标准化的鸢尾花数据

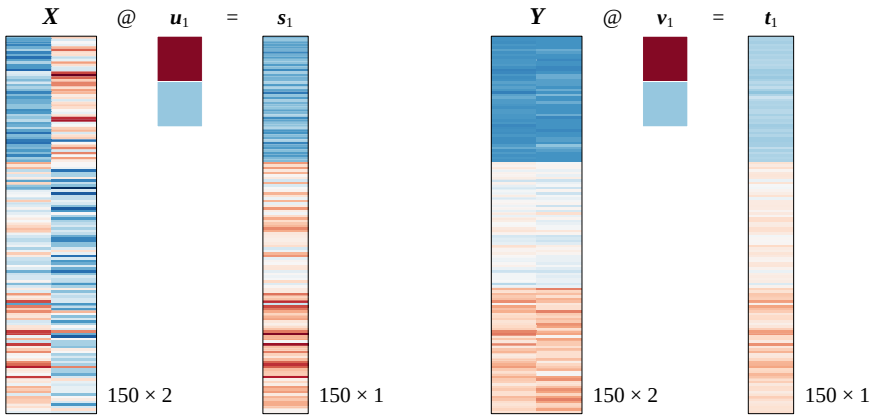


图 13. 通过投影计算 s_1 和 t_1

特征值分解

下面我们利用特征值分解自行求解 u_1、v_1。根据图 4 和图 5，我们先需要计算 P 和 Q 两个方阵。具体过程如图 14、图 15 所示。

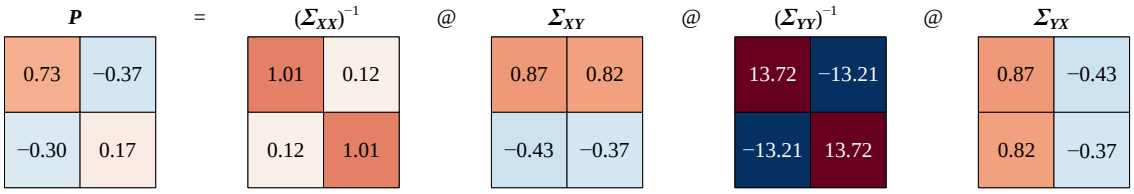


图 14. 计算矩阵 P

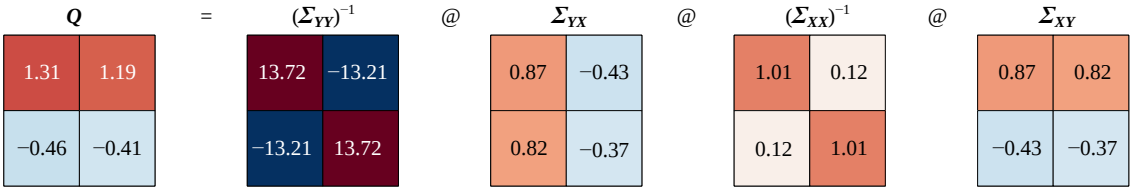


图 15. 计算矩阵 Q

然后对 P 和 Q 分别进行特征值分解，具体如图 16、图 17 所示。

注意，图 17 中矩阵 V 的第 2 列向量 v_2 和图 11 中不同，但是两者为倍数关系，即共线。

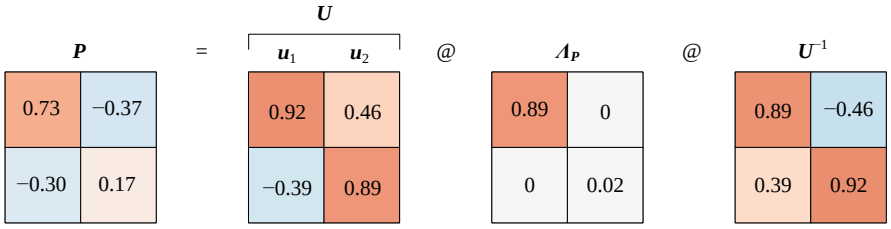


图 16. 矩阵 $\boldsymbol{P}$ 特征值分解

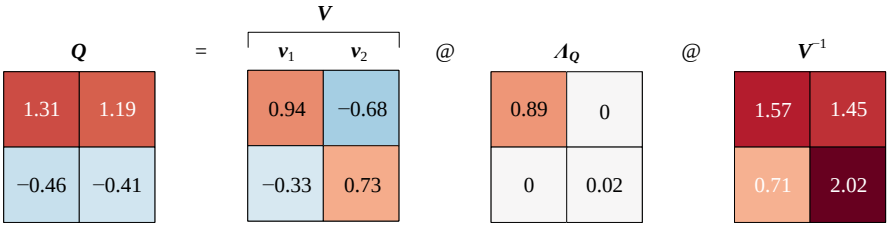


图 17. 矩阵 $\boldsymbol{Q}$ 特征值分解

Bk6_Ch18_01.py 完成本章 CCA 分析及可视化。

至此，我们完成了《数据有道》一册学习！恭喜大家，走完了鸢尾花书 6/7 的旅程！

本册两个核心话题是回归、降维。鸢尾花书中线性回归、主成分分析被反反复复提及，原因很简单，这两种算法实际上是各种数据工具的合体。我们可以从代数、几何、数据、概率统计、线性组合、向量空间、矩阵分解、优化各种角度理解线性回归、主成分分析。这也是鸢尾花书想给大家“灌输”的理念——见树又见林。

数据可以是各种各样的形式，比如数字、文本、图像等等。但是，这些数据并不是随意的，需要经过处理和清洗才能用于机器学习。Garbage in, garbage out! 我们不能让机器学习算法去学习一些无用的垃圾数据吧！而《数据有道》介绍的算法常被用于特征工程。

大家已经清楚，回归、降维、分类、聚类是机器学习的四大类问题。本册关注机器学习中的回归、降维这两类问题。鸢尾花书最后一册《机器学习》则关注经典分类、聚类算法。

让我们在《机器学习》一册再见！