Data Transformations

4 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、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。

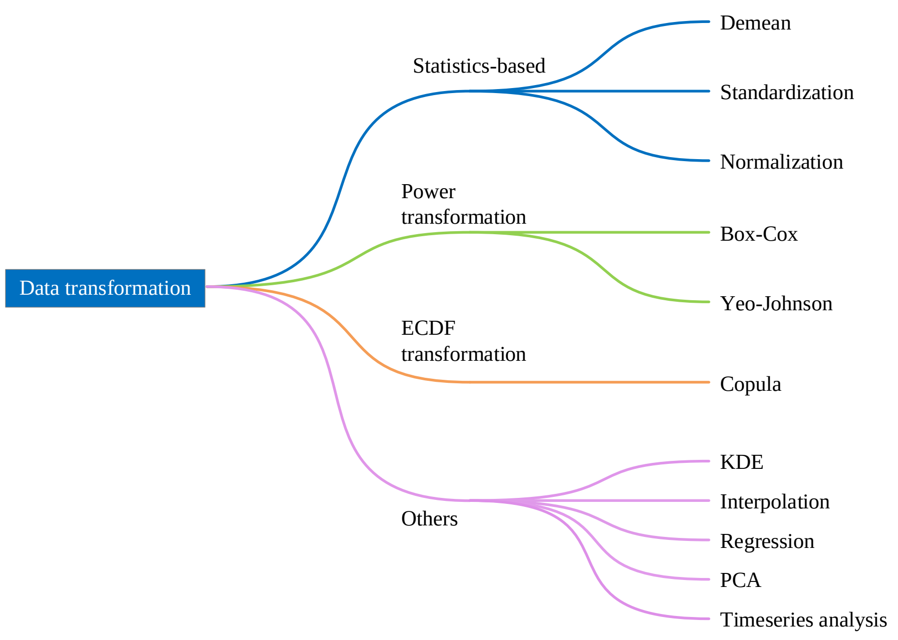

图 1. 常见数据转换方法

4.2 中心化：去均值

数据中心化 (centralize, demean)，也叫去均值，是最基本的基于统计的数据转换。

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，去均值处理效果并不明显。

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。

相信从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。

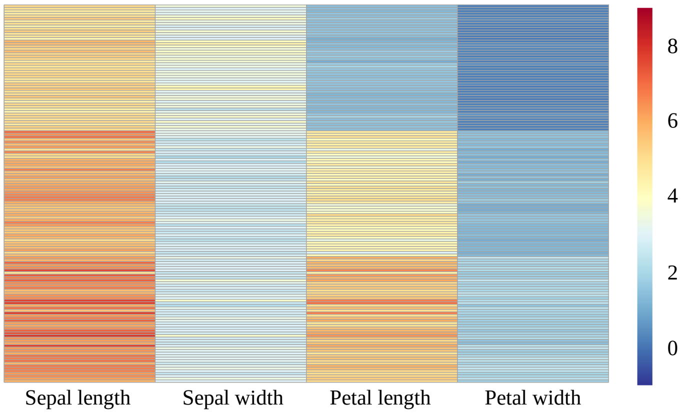

图 2. 鸢尾花数据，原始数据矩阵 $\boldsymbol{X}$

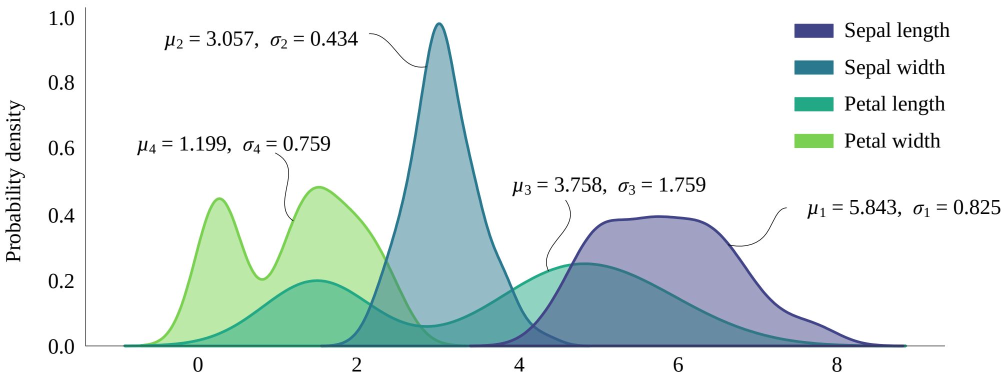

图 3. 鸢尾花数据四个特征上分布，KDE 估计

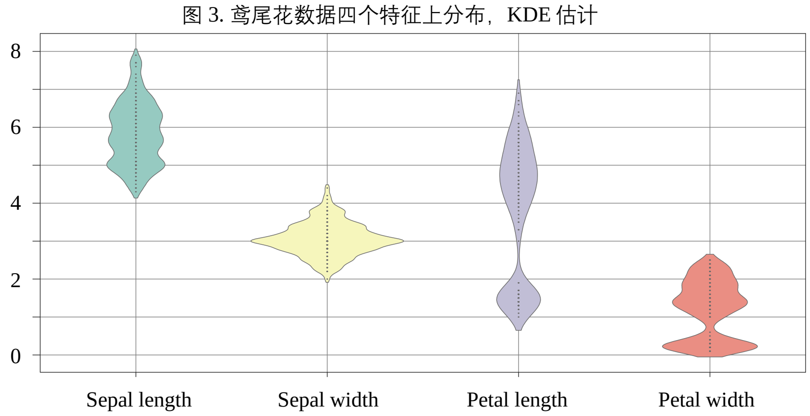

图 4. 鸢尾花原始数据，小提琴图

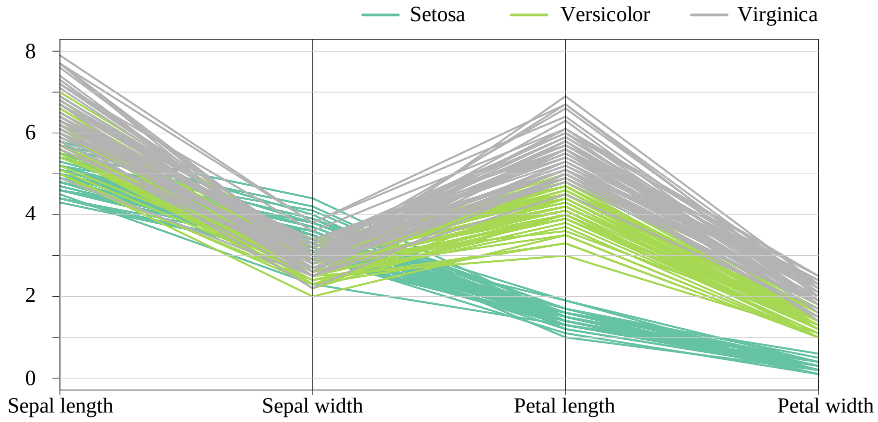

图 5. 鸢尾花数据，平行坐标图

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，去均值相当于将数据质心移动到 0，但是对数据分布的离散度没有任何影响。

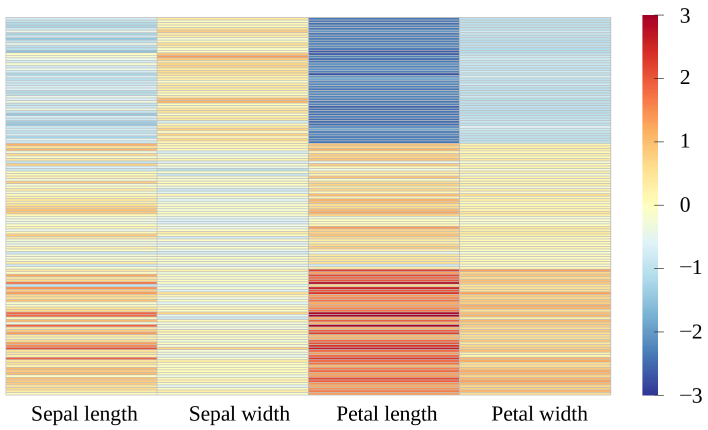

图 6. 数据热图，去均值

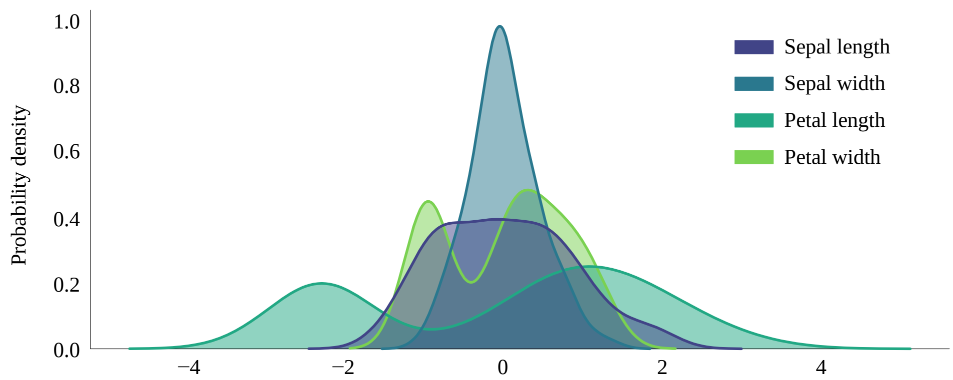

图 7. 数据 KDE 分布估计，去均值

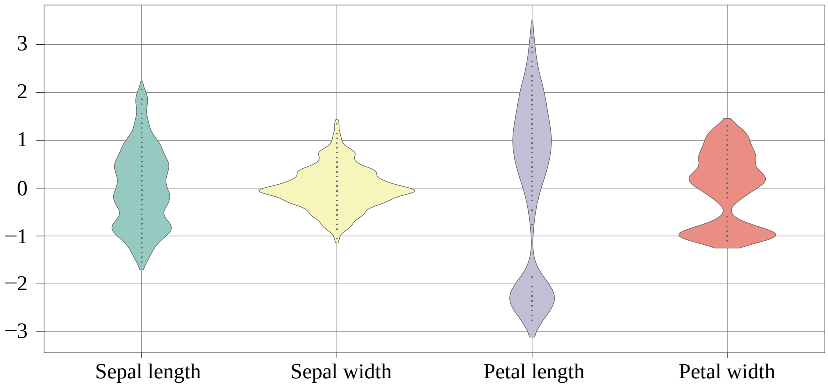

图 8. 小提琴图，去均值

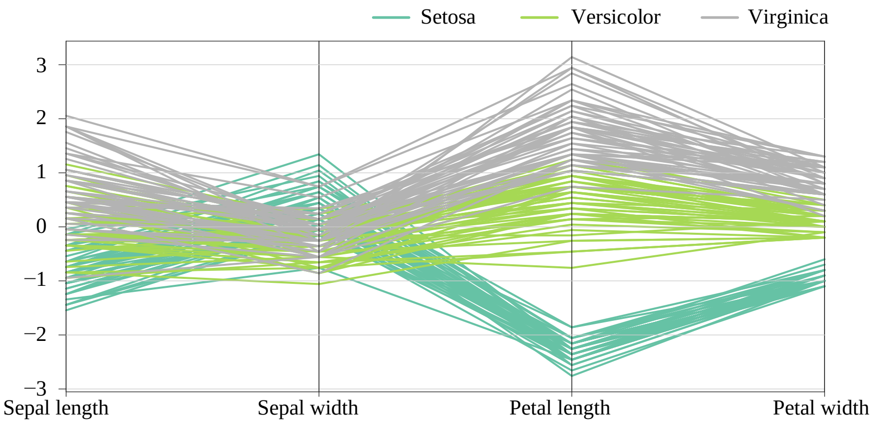

图 9. 平行坐标图，去均值

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 (z score)，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。

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。

《概率统计》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。

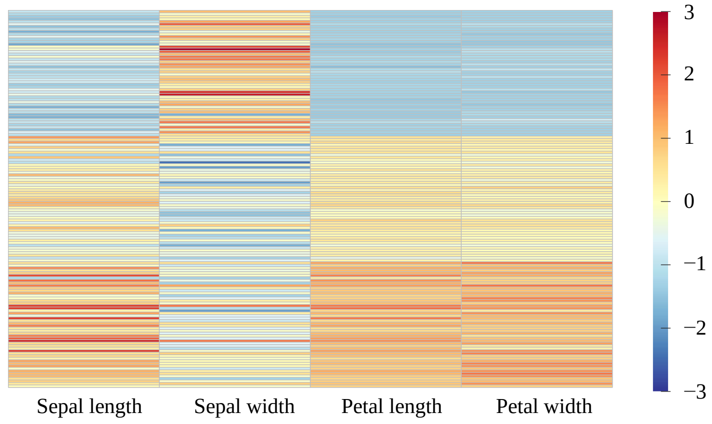

图 10. 热图，标准化

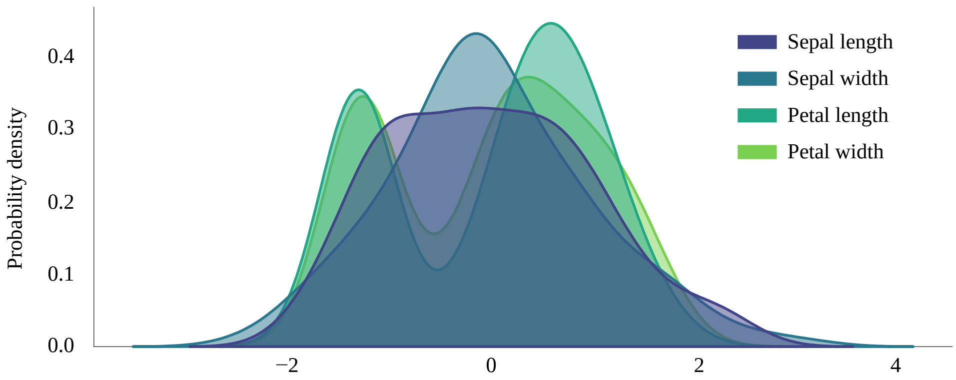

图 11. KDE 分布估计，标准化

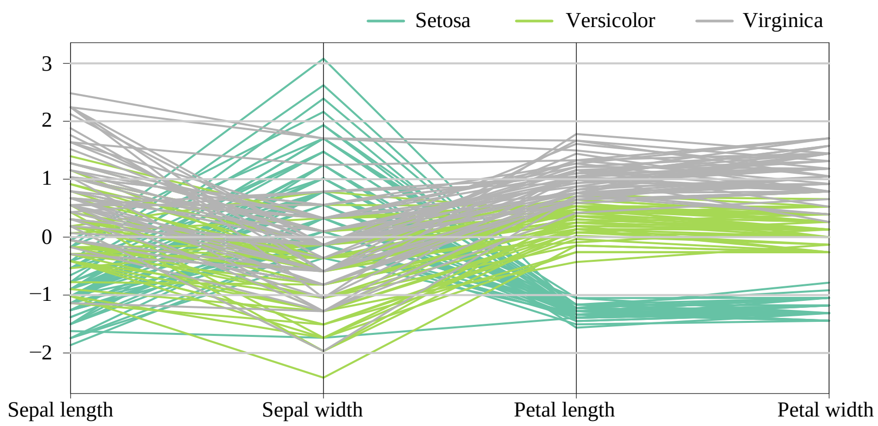

图 12. 平行坐标图，标准化

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。注意，很多时候 normalization 和 standardization 两个词混用，大家注意区分。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。

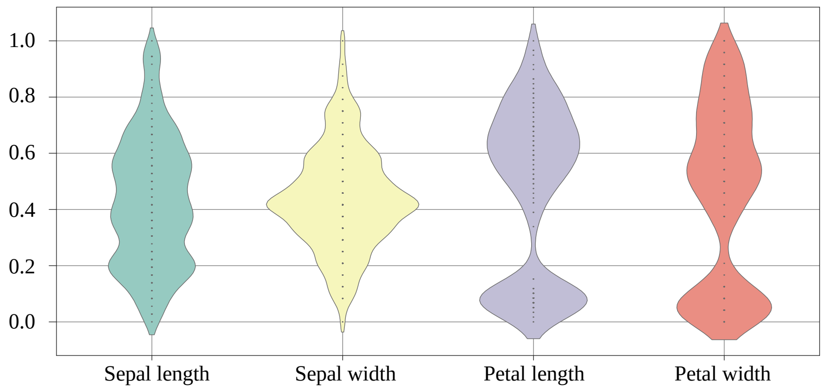

图 13. 小提琴图，归一化

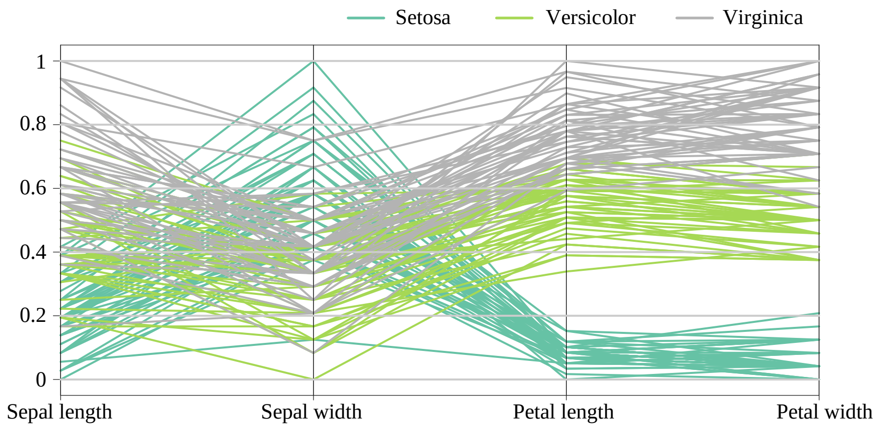

图 14. 平行坐标图，归一化

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，

$$IQR = Q_3 - Q_1 \tag{6}$$

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 转换：

$$x^{(\lambda)} = \begin{cases} \dfrac{x^{\lambda} - 1}{\lambda} & \lambda \neq 0 \\ \ln x & \lambda = 0 \end{cases} \tag{7}$$

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 X 为正数。

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)。

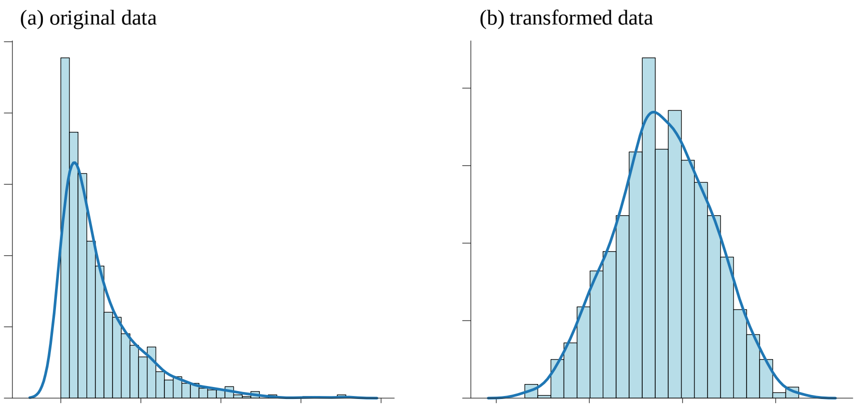

图 15. 原始数据和转换数据的直方图

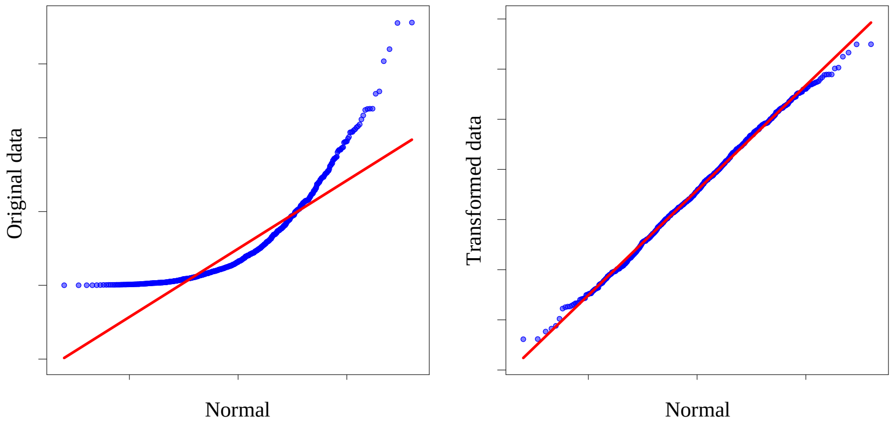

图 16. 原始数据和转换数据的 QQ 图

Yeo-Johnson 转换

另外一种转换为，Yeo-Johnson 转换，这种转换不要求 X 大于 0：

$$x^{(\lambda)} = \begin{cases} \dfrac{(x+1)^{\lambda}-1}{\lambda} & \lambda \neq 0, x \geq 0 \\ \ln(x+1) & \lambda = 0, x \geq 0 \\ \dfrac{-\left((-x+1)^{2-\lambda}-1\right)}{2-\lambda} & \lambda \neq 2, x < 0 \\ -\ln(-x+1) & \lambda = 2, x < 0 \end{cases} \tag{8}$$

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

《概率统计》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。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。

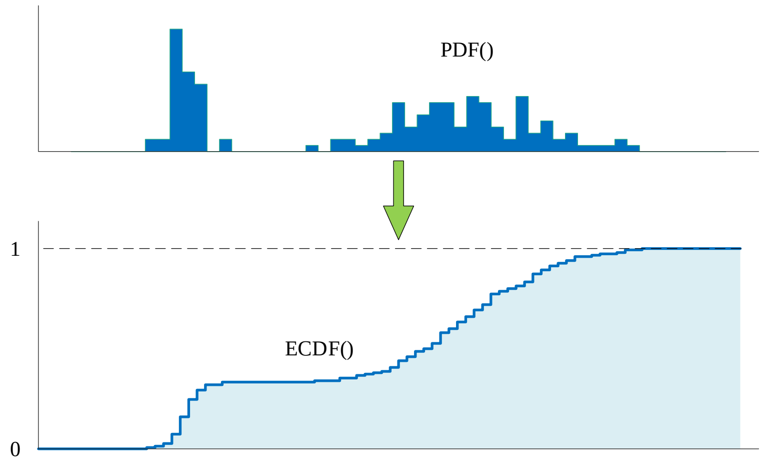

图 17. ECDF 函数转换样本数据

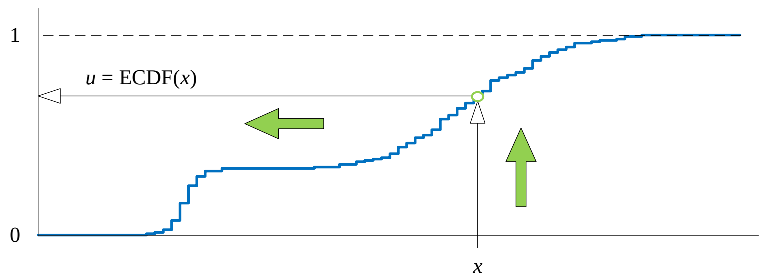

图 18. ECDF 函数原理

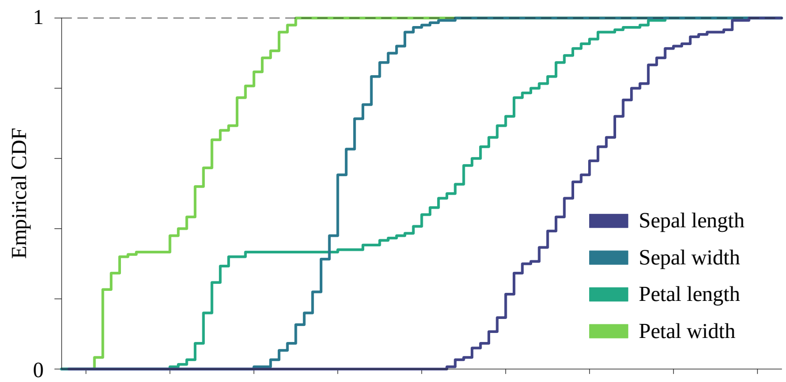

图 19. 鸢尾花数据四个特征的 ECDF

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。

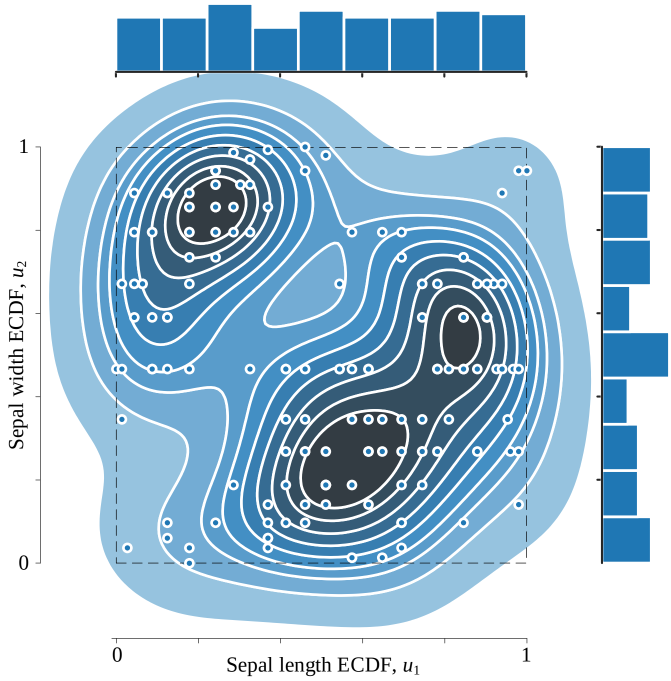

图 20. 鸢尾花花萼长度、花萼宽度 ECDF 散点图

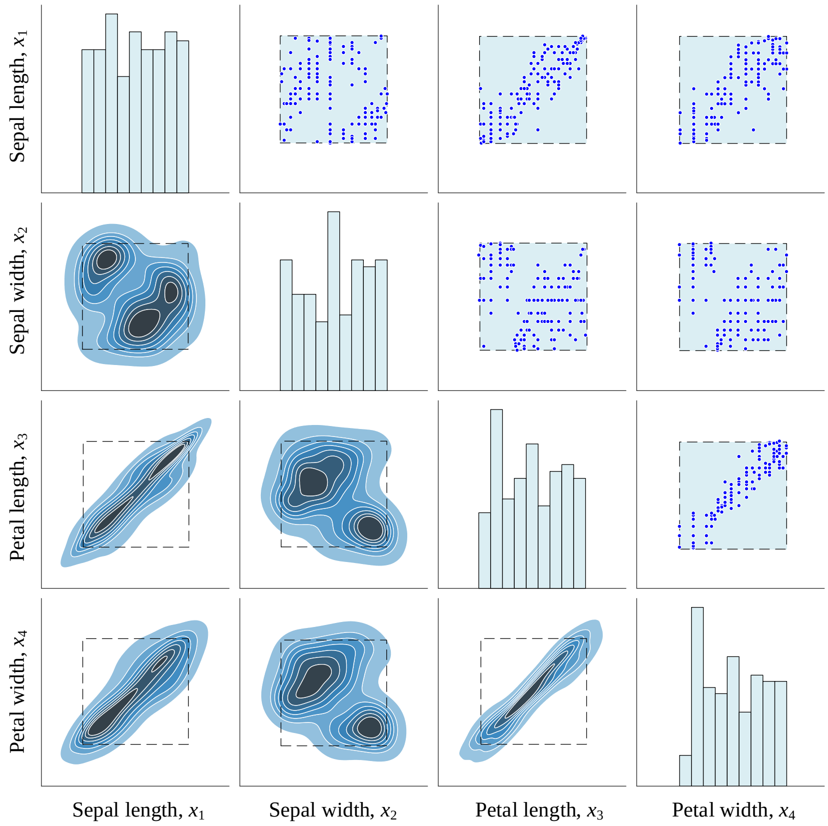

图 21. 鸢尾花数据 ECDF 的成对特征图

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 和图 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：

$$\begin{bmatrix} x_1 & x_2 & \cdots & x_D \end{bmatrix} \tag{9}$$

它们各自的边缘经验累积概率分布值可以构成如下向量：

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

其中 $u_j = \mathrm{ECDF}_j(x_j)$ 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。

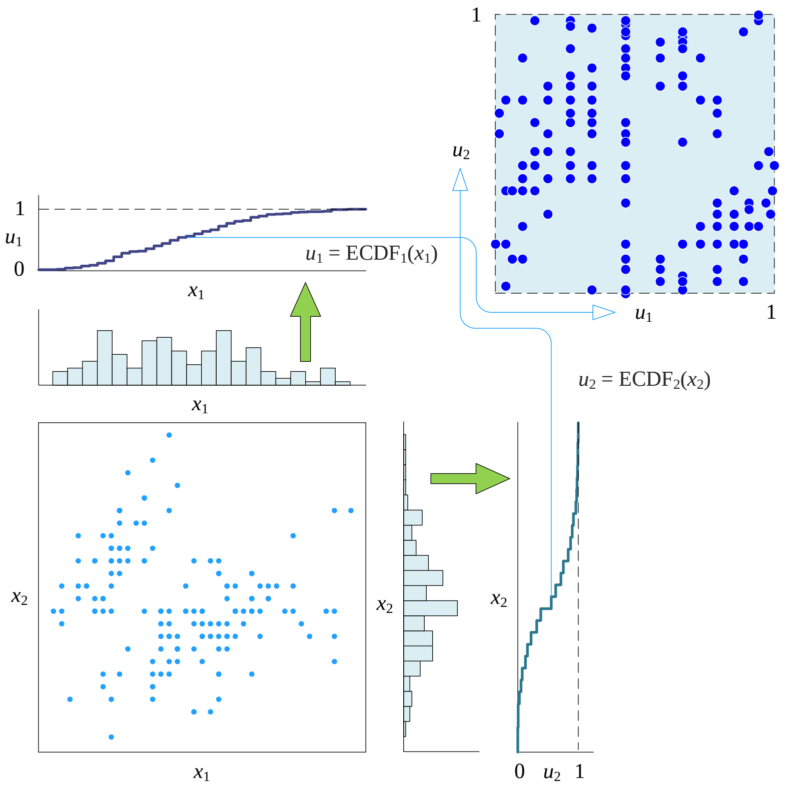

图 22. x_1 和 x_2，和 u_1 和 u_2 的关系

从反方向来看 (10)：

$$\begin{bmatrix} x_1 & x_2 & \cdots & x_D \end{bmatrix} = \begin{bmatrix} \mathrm{ECDF}_1^{-1}(u_1) & \mathrm{ECDF}_2^{-1}(u_2) & \cdots & \mathrm{ECDF}_D^{-1}(u_D) \end{bmatrix} \tag{11}$$

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：

$$C(u_1, u_2, \ldots u_D) = \mathrm{ECDF}\left(\mathrm{ECDF}_1^{-1}(u_1), \mathrm{ECDF}_2^{-1}(u_2), \ldots, \mathrm{ECDF}_D^{-1}(u_D)\right) \tag{12}$$

连接函数的概率密度函数，也就是 copula PDF 可以通过下式求得：

$$c(u_1, u_2, \ldots u_D) = \frac{\partial^D}{\partial u_1 \cdot \partial u_2 \cdot \ldots \cdot \partial u_D} C(u_1, u_2, \ldots u_D) \tag{13}$$

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。

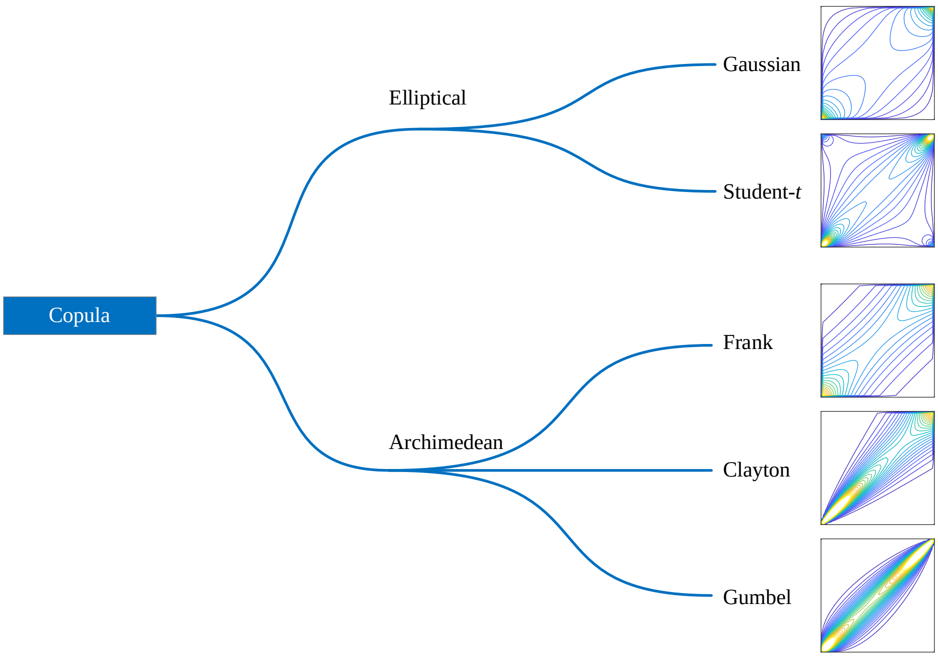

图 23. 常见连接函数

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