

作者	生姜 DrGinger
脚本	生姜 DrGinger
视频	崔崔 CuiCui
开源学习资源	<a href="https://github.com/Visualize-ML">https://github.com/Visualize-ML</a>
平台	<a href="https://www.youtube.com/@DrGinger_Jiang">https://www.youtube.com/@DrGinger_Jiang</a> <a href="https://space.bilibili.com/3546865719052873">https://space.bilibili.com/3546865719052873</a> <a href="https://space.bilibili.com/513194466">https://space.bilibili.com/513194466</a>

## 2.3 矩阵的形状



### 本节你将掌握的核心技能：

- ▶ 列向量、行向量在数据分析、机器学习中的用途。
- ▶ 细高矩阵表示样本多、特征少，扁平矩阵则反之，二者可通过转置互换。
- ▶ 方阵：很多运算的前提，如行列式、逆矩阵、特征值分解等。
- ▶ 对称矩阵：对称矩阵转置等于自身。
- ▶ 对角矩阵：非主对角元素为零，可为方阵或长方阵。
- ▶ 单位矩阵：对角方阵，主对角线元素为 1，矩阵乘法中特别重要。
- ▶ 三角矩阵：上三角矩阵与下三角矩阵互为转置，简化方程求解。
- ▶ 分块矩阵：将大矩阵拆为子矩阵，提升复杂运算可读性与效率。

本章第一节提过，矩阵的形状丰富多样，本节详细介绍各种矩阵形状，这些特殊形状的矩阵都有自己独特的用途。矩阵的形状是矩阵运算的基石，它不仅决定了矩阵之间能否进行加减、矩阵乘法等基本运算，还深刻影响着运算的结果及其背后的几何意义。此外，很多矩阵运算都对矩阵形状有要求。比如，行列式、LU 分解、Cholesky 分解、特征值分解、迹等，都以方阵为前提条件。

此外，在数据分析、机器学习和科学计算中，不同形状的矩阵——无论是行向量、列向量、细高矩阵、扁平矩阵，还是方阵——都扮演着独特的角色。如果我们对矩阵的形状不满意，还可以把矩阵切成满意的形状；这里用到的工具是分块矩阵，这也是本节要介绍的内容。

图 1 所示为各种常见的矩阵形状。本节将详细介绍各种常见形状矩阵以及它们用途。

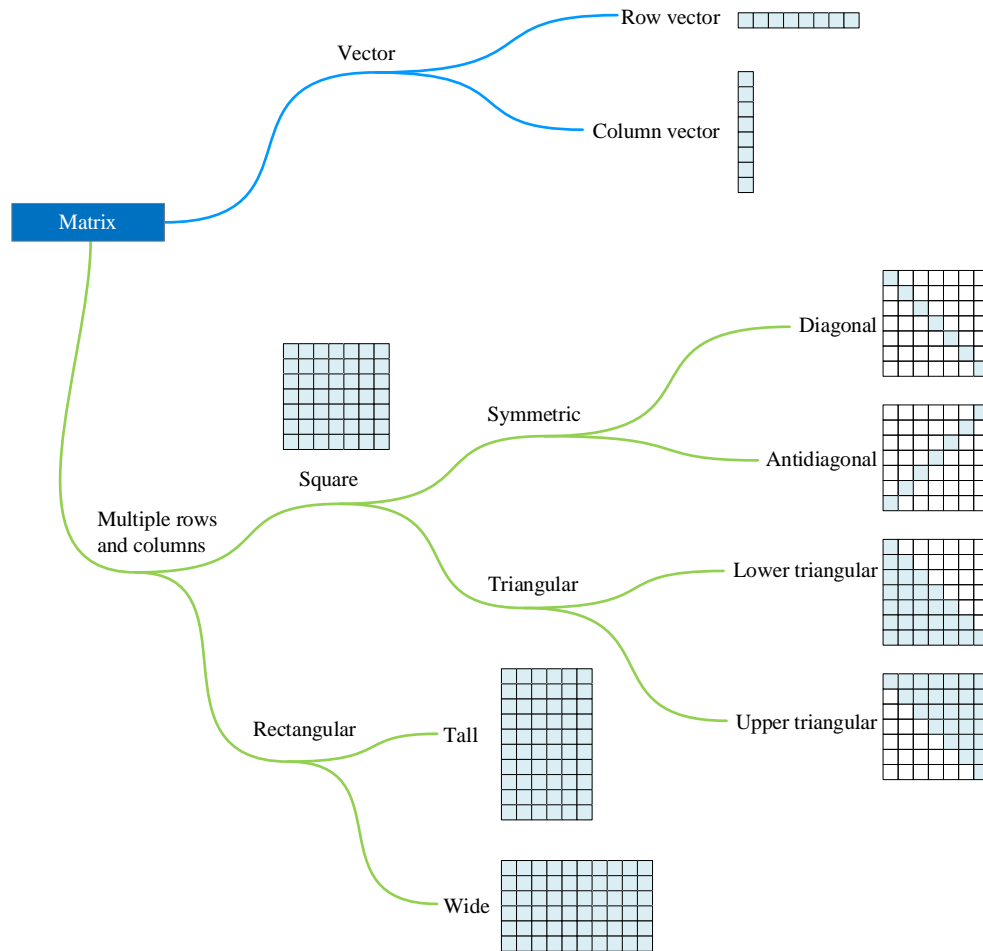


图 1. 几种常见矩阵形状

## 列向量

本书之前介绍的**行向量** (row vector)、**列向量** (column vector) 也是特殊形状的矩阵。

行向量可以看作一行多列的矩阵，而列向量则是一列多行的矩阵。两者之间通过转置操作相互转换：行向量转置得到列向量，列向量转置得到行向量。

行向量、列向量虽然形式简单，但它们在数据分析、线性代数和机器学习中扮演着不可或缺的角色。让我们先聊聊列向量。

平面上任意一点可以写成一个二维列向量，比如

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \quad (1)$$

三维空间中的一点可以写成一个三维列向量，比如

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad (2)$$

大家已经很熟悉的，在 RGB 空间中，颜色向量都可以写成列向量的形式。比如，红色、绿色、蓝色向量可以写成

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3)$$

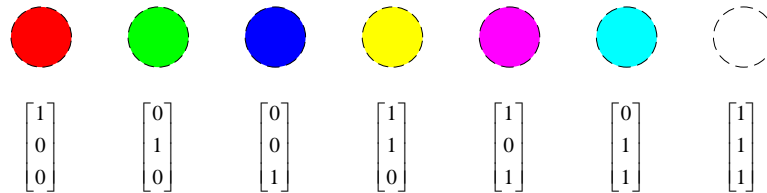


图 2. 用列向量表示颜色

在数据矩阵中，列向量常用来存储单一特征的数据。

例如，在鸢尾花数据集中，每一列向量代表一个特征，如花萼长度、花萼宽度、花瓣长度或花瓣宽度。以花萼长度为例，其列向量为

$$\begin{bmatrix} 5.1 \\ 4.9 \\ 4.7 \\ \vdots \end{bmatrix} \quad (4)$$

这个列向量记录了数据集中所有样本的花萼长度。从数据的角度来看，列向量表示数据集中一个特征的分布，基于此可以进行各种统计运算，如计算均值、方差或绘制分布直方图。

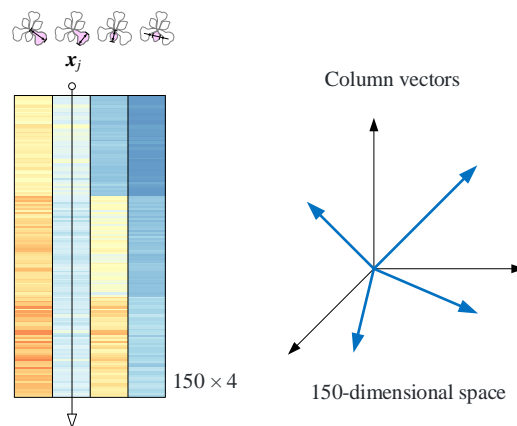


图 3. 鸢尾花数据列向量，不考虑标签列

从特征工程的角度来看，列向量的形式便于进行标准化、归一化或特征提取等操作，为后续的机器学习建模奠定基础。

没有特别说明，本书提到的向量默认为列向量。列向量则常常是**线性变换** (linear transformation) 作用的对象。简单来说，线性变换就是保持向量加法和数乘运算规则不变的变换，比如旋转、缩放、反射等。

而想要了解线性变换，我们需要先熟练掌握**矩阵乘法** (matrix multiplication)，这是下一节要介绍的话题。

## 行向量

在数据分析中，行向量通常用来表示数据集中的单个样本点。

具体来说，每一行数据对应一个样本，其各列则表示该样本的不同特征。

例如，在经典的鸢尾花数据集中，每一行对应一朵鸢尾花的记录，可以表示为一个行向量。比如，第一行行向量

$$\mathbf{x}^{(1)} = [5.1 \quad 3.5 \quad 1.4 \quad 0.2 \quad \text{Setosa}] \quad (5)$$

上述数据分别代表一朵鸢尾花样本的花萼长度、花萼宽度、花瓣长度、花瓣宽度，以及分类标签。

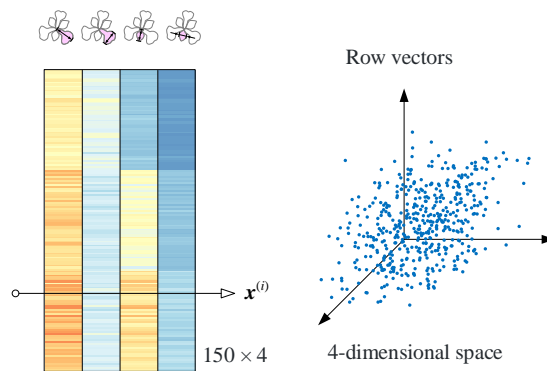


图 4. 鸢尾花数据行向量，不考虑标签列

通过行向量的形式，我们可以将复杂的多维数据简化为一个紧凑的数学对象，便于存储、处理和建模。

以颜色为例，图 5 中有一组  $n$  个颜色，这  $n$  个颜色构成了 RGB 颜色立方体的外框线。我们可以把这些颜色向量储存在矩阵中，这个矩阵的每一个行向量代表一个颜色。

而这个矩阵便是长方形矩阵。

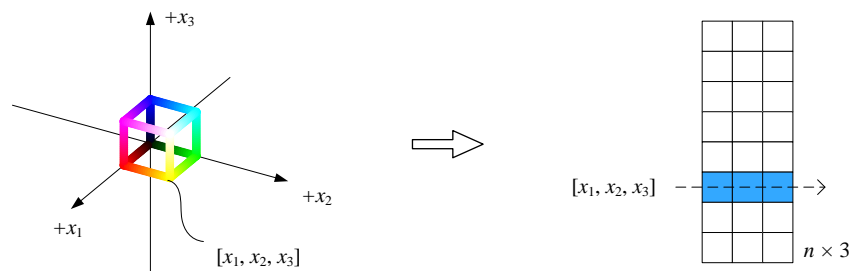


图 5. RGB 颜色立方体的外框线

## 长方形矩阵

**长方形矩阵** (rectangular matrix) 是指行数、列数不相等的矩阵；长方形矩阵有两种形式——**细高** (tall)、**扁平** (tail)。

细高矩阵是指行数远大于列数的矩阵。这种矩阵的形状像一根高耸的立柱，比如鸢尾花数据集。

细高矩阵常用于表示样本数远多于特征数的数据集。例如，在机器学习中，一个包含 1000 个样本、每个样本有 10 个特征的数据集可以表示为一个  $1000 \times 10$  的细高矩阵。

扁平矩阵是指列数远大于行数的矩阵。扁平矩阵常用于表示特征数远多于样本数的数据集。例如，在基因表达数据分析中，一个包含 100 个样本、每个样本有 10000 个基因特征的数据集可以表示为一个  $100 \times 10000$  的扁平矩阵。

当然，细高矩阵转置之后得到扁平矩阵。

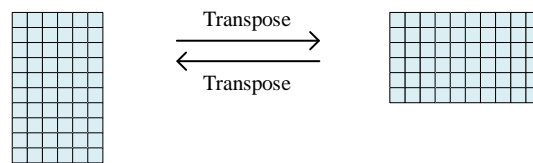


图 6. 细高矩阵、扁平矩阵通过转置相互转换

## 方阵

**方阵** (square matrix) 是矩阵家族中一种特殊且重要的形式，其行数、列数相等，即形状为  $n \times n$ 。

一个  $n \times n$  的方阵  $A$  可以表示为：

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix} \quad (6)$$

对于方阵来说，主对角线是左上角延伸到右下角元素的直线；这条对角线完美地将矩阵分为对角线，以及上下两个三角形区域。

把缺省的元素用 0 来补充，我们就可以看到不同的方阵类型，这是下文要介绍的内容。

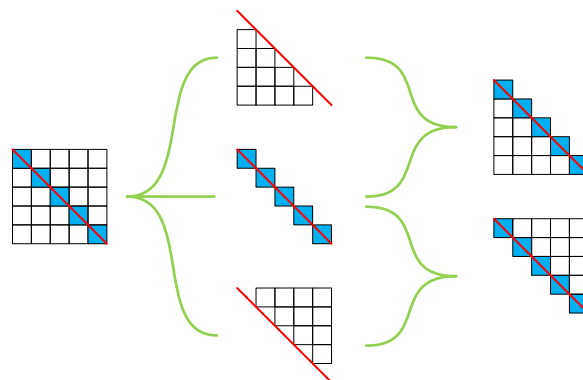


图 7. 方阵的主对角线将矩阵分割成不同区域

表 1 所示为以方阵为前提的矩阵运算。大家现在不必为这些线性代数概念费心劳神；建议大家学完本书后再回顾这个表格。

表 1. 以方阵为前提的矩阵运算

运算	说明
矩阵行列式	行列式仅对方阵定义，用于判断矩阵是否可逆（非奇异）等重要性质
矩阵的迹	矩阵的迹是方阵主对角线元素的总和，反映了矩阵线性变换整体的“缩放效应”或特征值的总和
矩阵的逆	仅当矩阵是方阵且行列式非零时，逆矩阵才存在。非方阵可以求其伪逆
矩阵的幂	矩阵的整数次幂，如 $A^n$ ，通常要求矩阵是方阵，否则矩阵乘法无法进行
特征值分解	特征值分解仅仅适用于方阵
谱分解	谱分解仅仅适用于对称方阵
对角化	只有方阵可以被对角化，即通过相似变换将矩阵转化为对角矩阵
LU 分解	将方阵分解为下三角矩阵和上三角矩阵
Cholesky 分解	正定矩阵（对称方阵为前提）可以进行 Cholesky 分解
二次型	二次型也是以方阵为前提
瑞利商	瑞利商的分子对应的矩阵乘法也离不开方阵

## 对称矩阵

上一节提过，一个  $n \times n$  方阵  $A$  转置结果为自身，即

$$A = A^T \quad (7)$$

则称  $A$  为**对称矩阵** (symmetric matrix)。

这种性质意味着方阵  $A$  的元素关于主对角线具有镜像对称性。

具体来说，对于矩阵  $A$  中的位于第  $i$  行、第  $j$  列的元素  $a_{i,j}$ ，与位于第  $j$  行、第  $i$  列的元素  $a_{j,i}$  相等。这种对称性使得对称矩阵只需要存储主对角线及其一侧的元素（上三角或下三角矩阵），就可以完全描述整个矩阵。

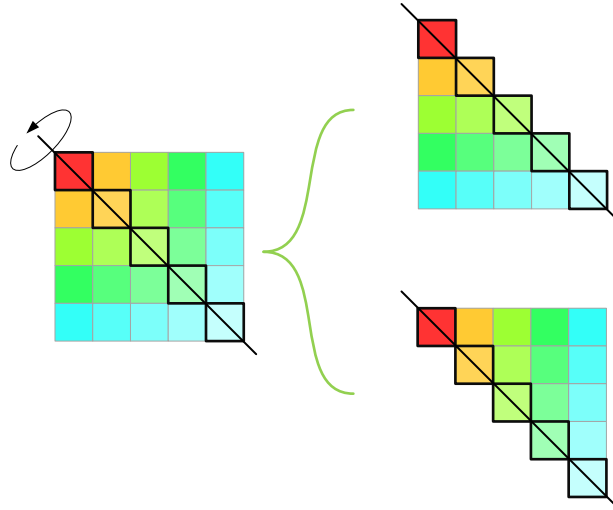


图 8. 对称矩阵拆分

## 对角矩阵

**对角矩阵** (diagonal matrix) 所有非零元素都位于主对角线上，而其他**非主对角线元素** (off-diagonal elements) 必须为 0。

⚠ 注意，对角矩阵的主对角线元素可以是 0 或非 0 元素。

对角矩阵有两种：**对角方阵** (square diagonal matrix)，**对角长方形** (rectangular diagonal matrix)。

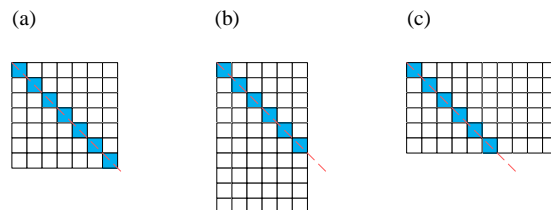


图 9. 不同形状的对角矩阵

对角方阵的形式如下所示

$$D = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{bmatrix} \quad (8)$$

如果行数大于列数 (细高)，对角长方形的形式为

$$D_{m \times p} = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_p \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad (m > p) \quad (9)$$

如果行数小于列数 (扁平), 对角长方阵的形式为

$$D_{m \times p} = \begin{bmatrix} d_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_m & 0 & \cdots & 0 \end{bmatrix} \quad (m < p) \quad (10)$$

## 单位矩阵

**单位矩阵** (identity matrix) 是一种特殊对角方阵。单位矩阵的主对角线上的元素均为 1, 其余位置的元素均为 0。

本书中, 单位矩阵用  $I$  来表达:

$$I_{n \times n} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \quad (11)$$

比如,

$$I_{2 \times 2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad I_{3 \times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (12)$$

也有很多文献用  $E$  代表单位矩阵。本书的  $E$  专门用来代表**标准正交基** (standard orthonormal basis)。

单位矩阵的每一列代表标准正交基向量, 比如

$$I_{2 \times 2} = \begin{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1^{(2)} & \mathbf{e}_2^{(2)} \end{bmatrix}, \quad I_{3 \times 3} = \begin{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1^{(3)} & \mathbf{e}_2^{(3)} & \mathbf{e}_3^{(3)} \end{bmatrix} \quad (13)$$

单位矩阵在线性代数扮演着“数字 1”的角色, 是矩阵乘法中的单位元。大家很快会在矩阵乘法、逆矩阵中看到单位矩阵的用途。

## 三角矩阵

**三角矩阵** (triangular matrix) 也是特殊的方阵。



如果方阵对角线以下元素均为零，这个方阵被称作**上三角矩阵** (upper triangular matrix):

$$U_{n \times n} = \begin{bmatrix} u_{1,1} & u_{1,2} & \dots & u_{1,n} \\ 0 & u_{2,2} & \dots & u_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u_{n,n} \end{bmatrix} \quad (14)$$

如果方阵对角线以上元素均为零，这个方阵被称作**下三角矩阵** (lower triangular matrix):

$$L_{n \times n} = \begin{bmatrix} l_{1,1} & 0 & \dots & 0 \\ l_{2,1} & l_{2,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n,1} & l_{n,2} & \dots & l_{n,n} \end{bmatrix} \quad (15)$$

显然，上三角矩阵转置得到下三角矩阵，反之亦然。

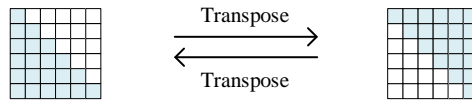


图 10. 上三角矩阵转置得到下三角矩阵，反之亦然

## 分块矩阵

大家如果对矩阵的形状“不满意”的话，我们还可以把矩阵“切割”成满意的形状！

本章前文提过，分块矩阵将一个大矩阵像“切豆腐”一样分割成多个小矩阵的形式，每个小矩阵称为**块** (block)，或**子矩阵** (submatrix)。

举个例子，给定如下  $5 \times 5$  矩阵  $A$

$$A = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (16)$$

我们可以把这个矩阵上下切一刀，矩阵  $A$  被分割成两个矩阵

$$A = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (17)$$

给这两个子矩阵分别取名字

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \quad (18)$$

可以发现  $A_1$  为对角方阵，形状为 3；而  $A_2$  为零矩阵。

再举个例子，如下矩阵  $A$  横竖各切一刀， $A$  被分割成四个子矩阵

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 4 & 5 & 5 \\ 0 & 0 & 7 & 8 & 9 \end{bmatrix} \quad (19)$$

给这四个子矩阵分别取名字

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \quad (20)$$

大家发现这个矩阵被分割成四个子矩阵；这四个子矩阵是矩阵的新“元素”，即

$$A_{1,1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad A_{1,2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad A_{2,1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad A_{2,2} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 8 \end{bmatrix} \quad (21)$$

大家可能已经发现这种记法的缺点是，我们并不能直接获得每个子矩阵的形状。我们需要进一步指出各个子矩阵各自的行列数。

图 11、图 12、图 13 这种分块矩阵记法参考了 NumPy 矩阵索引、切片。不同的是，这三幅都是基于 1 的计数。

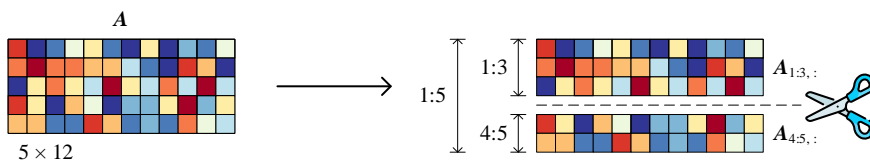


图 11. 上下切一刀

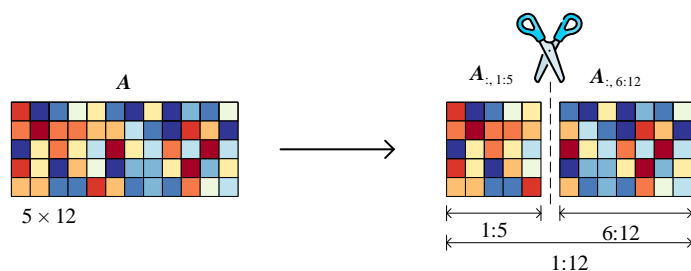


图 12. 左右切一刀

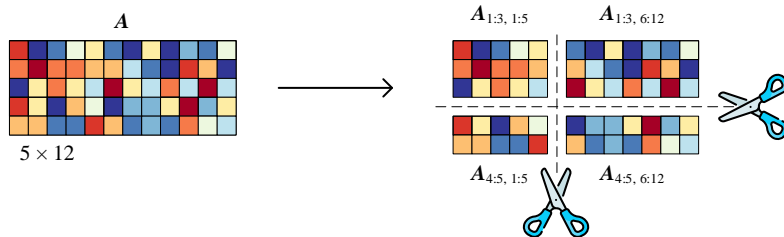


图 13. 上下、左右各切一刀

分块矩阵可以将复杂的大矩阵分解为多个小矩阵，使得计算更高效，尤其在处理大规模数据或进行矩阵运算时。本书后续将会用分块矩阵帮助我们理解**矩阵乘法** (matrix multiplication)。



请大家用 DeepSeek/ChatGPT 等工具完成本节如下习题。

- Q1.** 请用 `numpy.eye()` 生成  $3 \times 3$  单位矩阵，并判断任意两列正交 (内积为 0)。
- Q2.** 请自己写代码 (不用 `numpy.diag()`)，提取任意矩阵的主对角线元素。
- Q3.** 请自己写代码 (不用 `numpy.diag()`)，将给定主对角线元素序列转化成对角方阵。
- Q4.** 请学习使用 `numpy.triu()` 提取上三角方阵。
- Q5.** 请学习使用 `numpy.tril()` 提取下三角方阵。
- Q6.** 请根据矩阵转置是否为自原矩阵，写代码判断矩阵是否为对称矩阵。用 `numpy.array_equal()` 判断两个矩阵是否相同。
- Q7.** 用 `numpy.random.randint()` 生成  $5 \times 12$  矩阵，并按照图 11、图 12、图 13 切成子矩阵。