

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

2 Discrete Random Variables

离散随机变量

随机变量取值能够一一列举，概率质量函数本身为概率值

2.1 离散随机变量



本节你将掌握的核心技能：

- ▶ 随机变量是将随机试验结果映射为实数的函数；
- ▶ 离散随机变量的取值能够一一列举；
- ▶ 连续随机变量的取值不能一一列举或在区间上连续；
- ▶ 概率质量函数 PMF 描述离散随机变量在每个取值处的概率；
- ▶ 离散随机变量的 PMF 所有概率之和为 1；
- ▶ PDF 描述连续随机变量的概率分布，PDF 本身不是概率，概率由区间积分得到，PDF 取值非负但可大于 1；
- ▶ 累积分布函数 CDF 描述随机变量取值小于或等于某值的概率。

两种随机变量：离散、连续

随机变量 (random variable) 是连接“随机试验结果”和“数值世界”的桥梁。

具体来说，随机变量是一种函数，用来将每一次随机试验中可能出现的结果映射为一个实数。

随机变量分为两种——**离散** (discrete)、**连续** (continuous)。

如果随机变量的所有取值能够一一列举出来，可以是**有限个**或**可数无穷个**，这种随机变量被称作**离散随机变量** (discrete random variable)。

例如，抛一枚硬币时，样本空间为{正面，反面}。为了方便研究，我们定义一个随机变量 X ，把正面映射为 1，反面映射为 0。当然，我们也可以选择其他数值。比如，另一种常见的方式是把正面映射为 1，反面映射为 -1。

再如，掷一颗色子得到的点数为 1、2、3、4、5、6 中的一个值。鸢尾花的标签有三种——setosa (C_1)、versicolour (C_2)、virginica (C_3)；我们可以把这三个标签分别定义为 0、1、2 这三个整数。

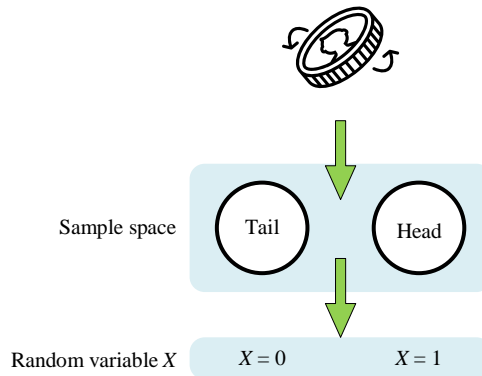


图 1. 随机试验、样本空间、随机变量

与离散随机变量相对的是，**连续随机变量** (continuous random variable)。连续随机变量取值可能对应全部实数，或者数轴上某一区间。比如，温度、人的身高体重都是连续随机变量。再比如，鸢尾花花萼长度、花萼宽度、花瓣长度、花瓣宽度也都可以视作连续随机变量。¹

本书用大写斜体字母表达随机变量，比如 X 、 Y 、 Z 、 X_1 、 X_2 、 Y_1 、 Y_2 等。用小写字母表达随机变量取值，比如 x 、 y 、 x_1 、 x_2 、 y_1 、 y_2 、 i 、 j 、 k 等。其中， x 、 y 、 x_1 、 x_2 、 y_1 、 y_2 等通用于离散、连续随机变量，而序号 i 、 j 、 k 一般用于离散随机变量。

简单来说， X 、 Y 、 Z 、 X_1 、 X_2 、 Y_1 、 Y_2 等替代描述随机试验结果的描述性文字。而 x 、 y 、 x_1 、 x_2 、 x_1 、 x_2 等相当于函数的输入变量，它们主要用在**概率密度函数** (probability density function, PDF)、**概率质量函数** (probability mass function, PMF) 中。

如图 2 所示，掷一颗色子试验中，令随机变量 X 为色子点数， $X = x$ ， x 代表取值。也就是说， X 的取值为变量 x 。举个例子， $\Pr(X = x)$ 为事件 $\{X = x\}$ 的概率， x 表示随机变量 X 的取值。当然我们可以把数值直接赋值给随机变量，比如 $\Pr(X = 5)$ 。

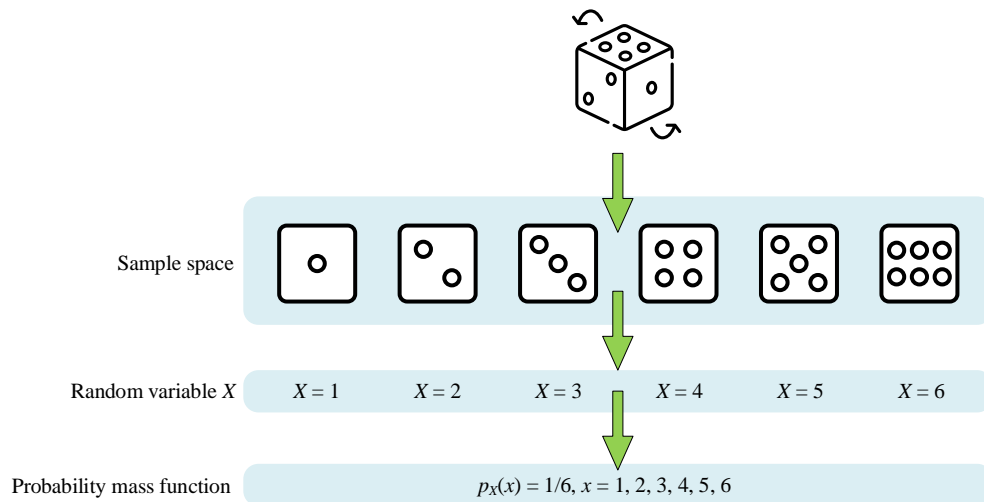


图 2. 随机试验、样本空间、随机变量、概率质量函数四者关系

以抛一枚六面色子为例，色子可能出现的所有点数组成了样本空间，即 $\{1, 2, 3, 4, 5, 6\}$ 。

为了便于数学处理，我们可以定义一个随机变量 X ，它的取值就是色子朝上的点数。随机变量 X 与样本空间中的结果一一对应。

接下来，如果我们假设色子是均匀的，那么每个点数出现的概率都是 $1/6$ ，这些概率值与 X 的各个可能取值之间的对应关系就构成了随机变量 X 的概率质量函数 PMF。换句话说，PMF 完整描述了 X 的取值以及相应的概率分布情况。

两种概率分布函数

研究随机变量取值的统计规律是概率论重要目的之一。概率分布函数是对统计规律的简化和抽象。图 3 比较两种概率分布函数——概率质量函数 PMF、概率密度函数 PDF。

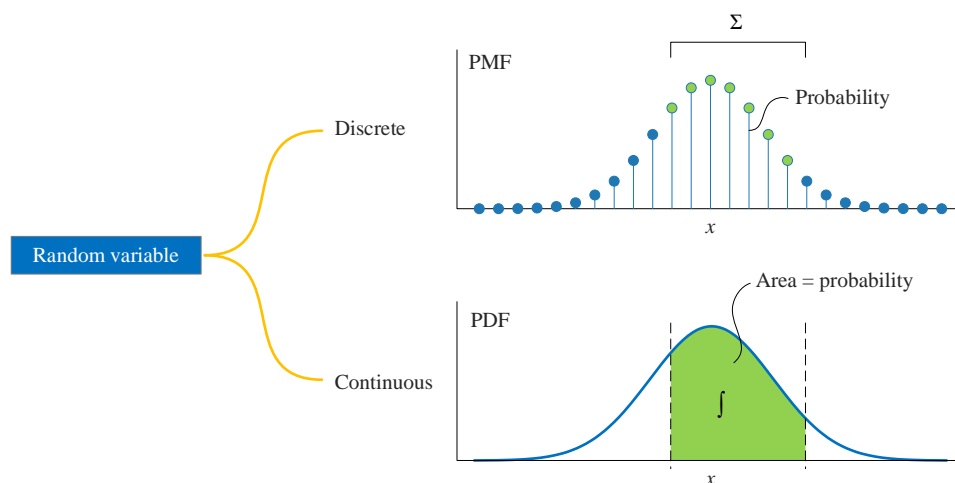


图 3. 比较概率质量函数、概率密度函数，图片来自《统计至简》

概率质量函数 PMF

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

版权归清华大学出版社所有，请勿商用，引用请注明出处。

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

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

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

当我们研究的是离散随机变量时，就使用**概率质量函数** (probability mass function, PMF) 来描述随机变量的分布规律 (如图 3 上图)。PMF 表示随机变量在某个具体取值处的概率大小。换句话说，它告诉我们“这个随机变量恰好等于某个值”的概率是多少。

⚠ 注意，很多教材翻译把 PMF 翻译做“分布列”，本书则将其直译为概率质量函数。

概率质量函数本质上就是概率，因此本书很多时候也直接称之为概率。此外，本书大多时候将概率质量函数直接简写为 PMF。

本书用小字斜体字母 p 表达 PMF，比如随机变量 X 的概率质量函数记做 $p_X(x)$ 。下角标 x 代表描述随机试验的随机变量，概率质量函数的输入为变量 x 。而概率质量函数 $p_X(x)$ 的输出则为“概率值”，即

$$p_X(x) = \Pr(X = x) \quad (1)$$

和函数一样，概率质量函数的输入也可以不止一个。如果我们研究多个随机变量的联合分布， $p_{X,Y}(x, y)$ 代表 (X, Y) 的联合概率质量函数。 $p_{X,Y}(x, y)$ 的输入为 (x, y) ，函数的输出为“ X 取 x 且 Y 取 y ”的概率。

$p_X(x)$ 本身就是“概率值”，因此计算离散随机变量 X 取不同值时的概率，我们使用求和运算。因此， $p_X(x)$ 对应的数学运算符是 Σ 。

⚠ 注意，有些资料为了方便，将 $p_X(x)$ 简写为 $p(x)$ ， $p_{X,Y}(x, y)$ 简做 $p(x, y)$ 。

抛一枚硬币

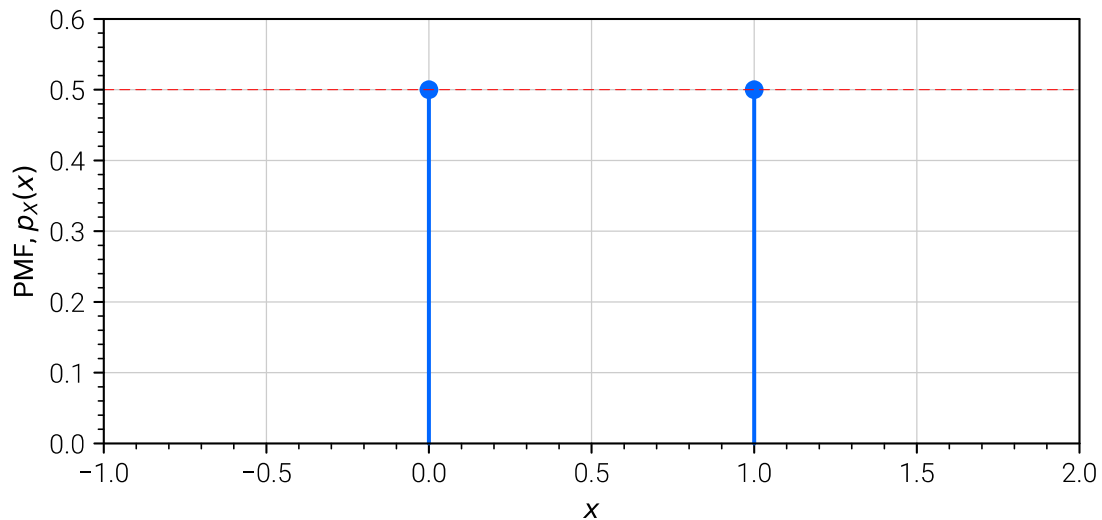
举一个例子，抛一枚 (质地均匀) 硬币试验中，令 X 为正面朝上数量， X 的样本空间为 $\{0, 1\}$ 。 $X = 1$ 代表硬币正面朝上， $X = 0$ 代表硬币反面朝上。

随机变量 X 的 PMF 为：

$$p_X(x) = \begin{cases} 1/2 & x = 0 \\ 1/2 & x = 1 \end{cases} \quad (2)$$

相信读者已经对图 4 不陌生，我们在图像上增加标注，水平轴代表 PMF 输入 x ，纵轴改为 PMF， $p_X(x)$ 代表概率质量函数。由于离散随机变量的取值是一个个独立的点，因此火柴梗图呈现为两根垂直的线段，分别位于 $x = 0$ 和 $x = 1$ 处，每根线的高度代表该结果出现的概率。

通过这个例子，我们可以更直观地理解 PMF 的含义：它就像是一个“概率柱状图”，告诉我们每个结果出现的可能性有多大。当我们在后续章节研究更复杂的离散随机变量时，都可以用同样的方式，用 PMF 来描述每种结果的概率分布情况。

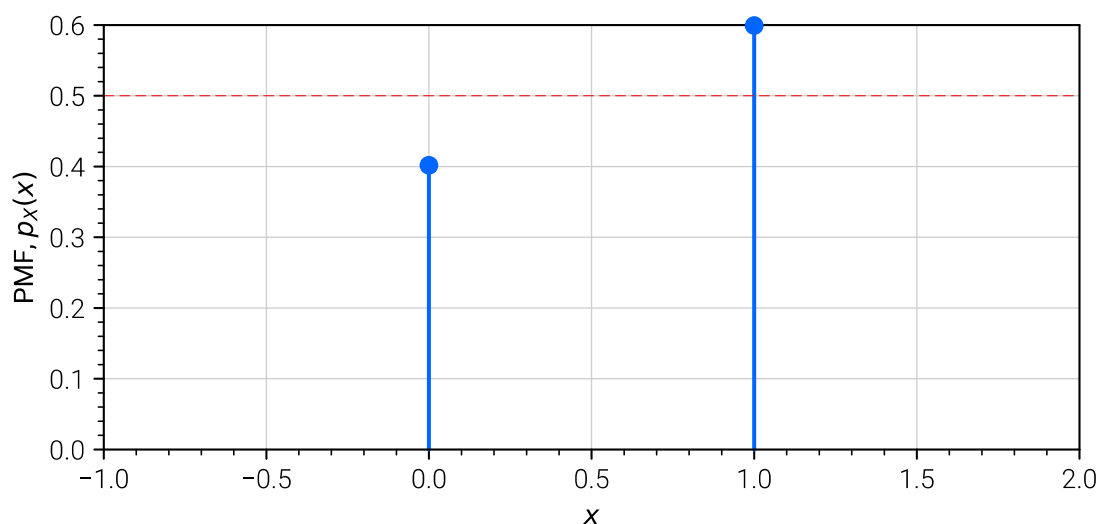
图 4. 离散随机变量 X 的 PMF，抛一枚质量均匀硬币

试想，抛一枚“头重脚轻”的硬币对应的 PMF 可能如图 5 所示。

试想，如果抛一枚“头重脚轻”的硬币，它并不是均匀的。设正面朝上的概率为 0.6，反面朝上的概率为 0.4，则它的概率质量函数可以表示为：

$$p_X(x) = \begin{cases} 0.4 & x=0 \\ 0.6 & x=1 \end{cases} \quad (3)$$

对应的火柴梗图如图 5 所示。然而，无论偏向程度如何，这两个概率仍然满足概率值总和为 1。换句话说，不论硬币如何偏向，每次试验必然得到正面或反面（互斥事件），没有第三种可能性，这保证了总和恒为 1。

图 5. 离散随机变量 X 的 PMF，抛一枚“头重脚轻”硬币

这实际上就是本册后续要介绍的**伯努利分布** (Bernoulli distribution) 的直观例子。伯努利分布描述的就是只有两种可能结果的随机试验，比如“正面或反面”、“成功或失败”、“有或无”。每一次试验中，某

一结果发生的概率记作 p ，另一结果发生的概率就是 $1 - p$ 。抛一枚硬币就是经典的伯努利分布，不论 p 是 0.5 还是 0.6。

掷一颗色子

我们再来看一个稍微复杂一点的例子——掷一颗色子。这同样是一个典型的离散随机实验。设随机变量 X 表示“色子朝上的点数”。因为色子共有六个面，编号从 1 到 6，所以随机变量 X 的样本空间为 $\{1, 2, 3, 4, 5, 6\}$ 。

这意味着，实验的所有可能结果就是色子落地后显示出的点数 1 到 6 之一。

如果色子质地均匀，那么每个面出现的概率都相同。因此，随机变量 X 的概率质量函数 PMF 可以写作：

$$p_X(x) = \begin{cases} 1/6 & x = 1, 2, 3, 4, 5, 6 \\ 0 & \text{Otherwise} \end{cases} \quad (4)$$

图 6 展示了这个概率质量函数的图像。图中，横轴表示离散随机变量 X 的所有可能取值，即 1 到 6；纵轴 $p_X(x)$ 表示每个取值对应的概率值。

这种形状的 PMF 通常被称为**离散均匀分布** (discrete uniform distribution)，离散均匀分布是指一个随机变量在有限个可能取值上，每个取值出现的概率完全相等。

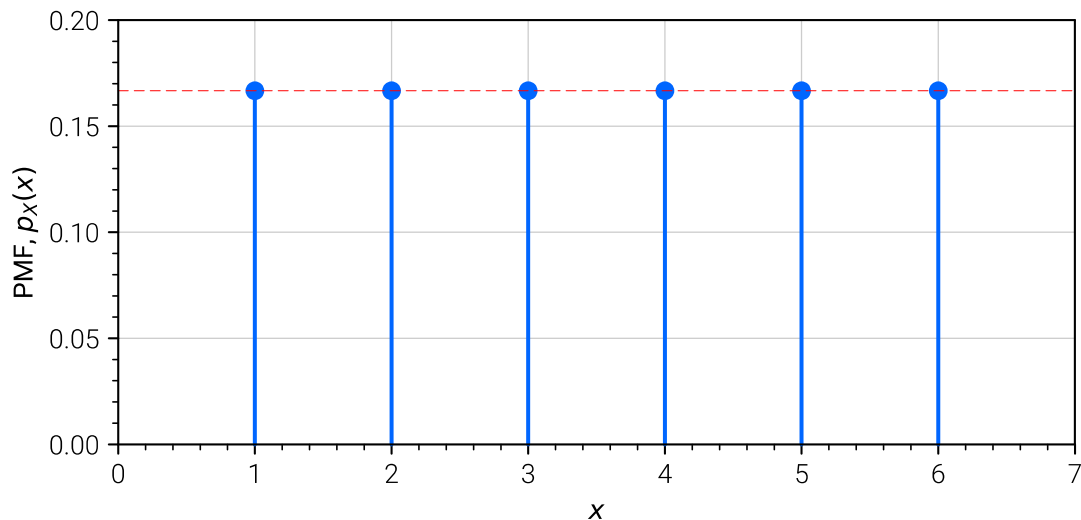


图 6. 离散随机变量 X 的 PMF，掷一颗色子

代码 1 绘制图 6 的离散随机分布 PMF，下面聊聊其中关键语句。

a 加载必要的 Python 包。NumPy 是 Python 中用于做数组、矩阵和数值计算的工具库。写成 `np` 是为了后面用它的函数时更简短，比如 `np.arange()` 就是 NumPy 的函数。加载 Matplotlib 的绘图库，简称为 `plt`。这是 Python 里最常用的画图工具。

b 中 `np.arange()` 用来生成一个“等间隔的整数序列”。`np.arange(1, 7)` 会生成从 1 到 6 的所有整数，也就是 `[1, 2, 3, 4, 5, 6]`。这代表我们要绘图的横坐标，就是抛色子的六个点数。

c `np.full_like()` 的意思是“创建一个与 `x` 形状一样的数组，但内容全部填同一个数”。这里 `x` 有六个数字，所以它会生成六个一样的值。`fill_value=1/6` 表示每个元素都填成 $1/6$ ，`dtype=float` 指定生成浮点数。它的作用就是生成 $[1/6, 1/6, 1/6, 1/6, 1/6, 1/6]$ ，即色子点数的等概率值。

d 是可视化代码。其中，`plt.figure(figsize=(6,3))` 创建一张画布，准备开始画图。`figsize=(6,3)` 是设置图像的宽和高，以英寸为单位。6 表示宽度，3 表示高度。

e 用 `plt.stem()` 来画“火柴梗图”。它会在每个 `x` 位置画一根竖线，竖线的高度就是 pmf 对应的值。它还会在竖线顶部画一个小圆点。这里的 `linefmt` 和 `markerfmt` 是设置竖线和圆点的颜色，`'#0066FF'` 是一种蓝色。`basefmt` 是基线（底部横线）的颜色。这个函数返回三样东西：圆点对象、竖线对象和基线对象，我们把它分别存到三个变量里，方便后面修改样式。

f 这部分代码用来装饰图像。

其中，`plt.setp()` 是一个专门用来修改图形属性的函数。这里用来修改 `markerline` 这个对象，也就是那些圆点，把它们的形状设置成 `'o'`，也就是空心圆。

`plt.setp(baseline, visible=False)` 把基线去掉。`visible=False` 表示“不显示”。基线就是一条在 `x` 轴位置的水平线，去掉后图形看起来更干净。

`plt.axhline()` 用来画一条水平直线。`y=1/6` 表示这条线在高度 $1/6$ 的位置。`c='r'` 表示颜色是红色，`ls='-'` 表示线型是虚线。这条线作为参考线，用来显示所有竖线的高度是一样的。

`plt.xlabel()` 设置横坐标标题。这里内容是 `"x"`，但用了 `r""` 的写法，这是原始字符串格式，方便写数学符号。

`plt.ylabel()` 设置纵坐标标签。标签内容是 `"PMF, p_X(x)"`，表示这是概率质量函数。依然用了原始字符串，让显示更像数学排版。

`plt.grid()` 打开网格线。`True` 表示启用网格。`color=0.8` 代表使用浅灰色的网格。

`plt.xlim()` 设置横轴显示的范围。这里把范围设为从 0 到 7，所以虽然数据只在 1 到 6，但会留空白区域。

`plt.xticks()` 设置横轴刻度。`np.arange(0,8)` 生成 0 到 7 的整数作为刻度值。

`plt.ylim()` 设置纵轴的范围，从 0 到 0.2。

`plt.tight_layout()` 会自动调整图中的各种元素，让它们之间不会挤在一起。

代码 1. 离散随机分布 PMF |  PS_02_01_01.ipynb


```

## 初始化
a import numpy as np
import matplotlib.pyplot as plt

## 生成数据
b x = np.arange(1, 7)
c pmf = np.full_like(x, fill_value=1/6, dtype=float)

## 可视化
d plt.figure(figsize=(6,3))
e markerline, stemlines, baseline = plt.stem(x, pmf,
                                              linefmt='#0066FF',
                                              markerfmt='#0066FF',
                                              basefmt='#0066FF')
f plt.setp(markerline, marker='o')
plt.setp(baseline, visible=False)
plt.axhline(y = 1/6, c = 'r', ls = '--')
plt.xlabel(r"$x$")
plt.ylabel(r"PMF, $p_{\{X\}}(x)$")
plt.grid(True, color = '0.8')
plt.xlim(0, 7)
plt.xticks(np.arange(0,8))
plt.ylim(0, 0.2)
plt.tight_layout()
plt.grid(axis='y', linestyle='--', linewidth=0.5)

```

归一律

在概率论中，任何随机变量的概率分布都必须满足一个基本原则：**归一律** (Normalization Law)。这个原则体现了概率的最核心含义：所有可能事件的总概率之和必须等于 1。

以一元离散随机变量 X 为例，设它的概率质量函数为 $p_X(x)$ 。根据归一律，我们有：

$$\sum_x p_X(x) = 1, \quad 0 \leq p_X(x) \leq 1 \quad (5)$$

这个式子说明，如果我们把随机变量 X 所有可能的取值一一列出，并将它们对应的概率全部相加，那么结果一定是 1。

这种“穷举所有可能结果再求和”的方式，也被称为穷举法。在离散随机变量的分析中，穷举法是最直接、最基本的概率计算方式。它不仅能帮助我们验证概率分布是否合理，也能用于计算各种事件的概率、期望值或方差等统计量。

⚠ 值得强调的是，概率质量函数 $p_X(x)$ 最小值为 0，最大取值为 1。这是因为概率本质上反映的是事件发生的可能性，它不可能为负，也不可能超过“必然发生”的上限，即 1。

概率密度函数 PDF

在离散随机变量中，我们用概率质量函数 PMF 来描述每个取值出现的概率。而对于连续随机变量，情况则完全不同——因为连续变量的取值是不可数的、无限多的，我们无法像枚举离散点那样给每

一个值分配一个具体的概率。此时，就需要引入另一种描述方式：**概率密度函数** (probability density function, PDF)。

概率密度函数是一种用来刻画连续随机变量分布规律的函数。本书中，连续随机变量 X 的概率密度函数记作 $f_X(x)$ ，其中小写斜体字母 f 用于区分它与离散情形下的 $p_X(x)$ 。

▲ 注意，在本书讲解贝叶斯推断时，为了方便，概率质量函数、概率密度函数都用 $f()$ 。

与 PMF 最大的不同在于：PDF 的值本身不是概率。对于连续随机变量，我们只能通过积分 \int 来计算一个区间内，比如 $[a, b]$ ，的概率。例如，

$$\Pr(a \leq X \leq b) = \int_a^b f_X(x) dx \quad (6)$$

这意味着， $f_X(x)$ 的“高度”代表概率密度，而区间下方的“面积”才代表实际的概率。也正因为如此，概率密度函数对应的数学运算符是积分符号 \int ，而非离散情形下的求和符号 Σ 。

为了帮助读者形成直观理解，我们来看一个最经典的例子。如图 7 所示，连续随机变量 X 服从标准正态分布 $N(0, 1)$ ，其 PDF 为：

$$f_X(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \quad (7)$$

$f_X(x)$ 函数在整个实数轴上都有定义，且随着 x 远离 0 而逐渐趋近于 0，但从不真正等于 0。图像呈现出对称的“钟形曲线”，表示数值越接近均值 0，出现的可能性越高；而数值越远离中心，发生的可能性就越小。



本书后文有专门章节讲解正态分布 (高斯分布)。

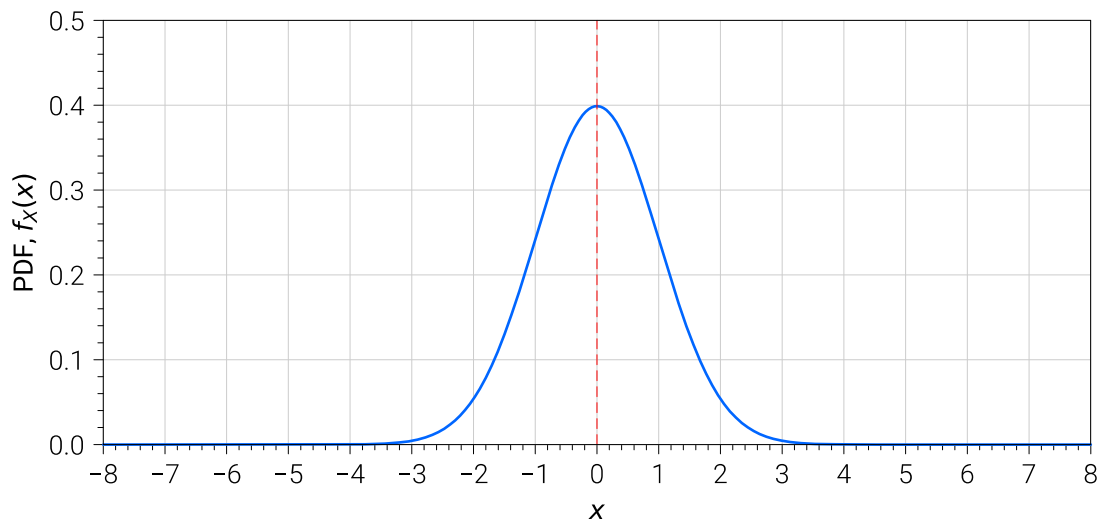


图 7. 标准正态分布 PDF

如图 7 所示，当 $x=0$ 时， $f_X(x)$ 约为 0.4，这个值是概率密度，不是概率。只有对连续随机变量 PDF 在指定区间，比如 $[-1, 1]$ ，内进行积分后结果才“可能”是概率。

例如，对于标准正态分布，有 $\Pr(-1 \leq X \leq 1) \approx 0.68$ ，即 68% 的样本落在距离“均值 ± 1 个标准差”以内。

▲ 值得反复强调的是，PMF 本身就是概率，对应的数学工具为 Σ 求和。PDF 积分后才可能是概率，对应的数学工具为 \int 积分。

与 PMF 一样，PDF 也必须满足归一化性质。比如，对于一元连续随机变量 X 的概率密度函数 $f_X(x)$ 必须满足：

$$\int_{-\infty}^{+\infty} f_X(x) dx = 1, \quad f_X(x) \geq 0 \quad (8)$$

上式也相当于“穷举法”。

▲ 注意，概率密度函数 $f_X(x)$ 取值非负，但是不要求小于 1。

概率质量函数 PMF、概率密度函数 PDF 是特殊的函数。特殊之处在于它们的输入为随机变量的取值，输出为概率质量、概率密度。但是，本质上，它们又都是函数。所以，我们可以把函数的分析工具用在概率质量函数 PMF、概率密度函数 PDF 上。



本书下一章将专门讲解连续随机变量。

代码 2 绘制图 7，下面聊聊其中关键语句。

- a 导入 Python 库，这和本节之前的代码 1 完全一致。
- b 定义了一个名为 `gaussian_pdf` 的函数。函数的作用是：给它传入横轴上的点 x_1 ，均值 μ ，标准差 σ ，它就会返回这些点对应的概率密度值。函数里面的内容是用户自己写的，并不是 Python 内置的。
- c 计算概率密度曲线的“前面的系数”，用到了 NumPy 的 `np.sqrt()` 函数来求平方根，`np.pi` 表示圆周率 π 。
- d 计算概率密度中“指数部分”的值。 $(x_1 - \mu)$ 是把横坐标点和均值比较， $/\sigma$ 是进行缩放， $**2$ 表示平方运算。 $-0.5 *$ 是在计算最终要放进指数函数里的值。
- e 把上面计算的系数和指数部分组合起来，并使用 NumPy 的 `np.exp()` 函数对指数进行“取 e 的幂”，最终得到曲线在每个点的高度值，也就是概率密度值。
- f 中， $\mu = 0$ 把分布的中心位置（均值）设为 0。之后画出来的曲线会以 0 为中心对称。
 $\sigma = 1$ 把曲线的标准差设为 1。标准差控制曲线的胖瘦，值越大曲线越宽，这里用的是标准正态分布的常用设置。
`np.linspace()` 是 NumPy 用来生成连续数字序列的函数。这里生成从 -8 到 8 的 1000 个均匀间隔的点。这些点就是绘图时横轴上的坐标。
- g 为可视化部分代码。
- h 调用自己写好的函数，把刚才生成的 1000 个点传进去，再把均值和标准差传进去，让函数计算这些点上的概率密度。返回的就是一条光滑曲线所需的纵坐标值。
- i 把横轴 x 和纵轴 pdf 画成折线图；点越密集，曲线看上去越光滑。`color='#0066FF'` 是指定线的颜色为蓝色。`plt.plot()` 是 Matplotlib 最基础、最常用的画线函数。
- j 为图像装饰部分代码，和代码 2 大部分一致。



大家也可以试着使用 `scipy.stats.norm` 中的 `pdf` 获得一元高斯分布的 PDF。

代码 2. 连续随机变量的 PDF | PS_02_01_02.ipynb

```

## 可视化
a import numpy as np
import matplotlib.pyplot as plt

## 自定义函数，高斯分布PDF
b def gaussian_pdf(x1, mu, sigma):
c     coeff = 1.0 / (np.sqrt(2.0 * np.pi) * sigma)
d     # 计算归一化系数
    exponent = -0.5 * ((x1 - mu) / sigma) ** 2
e     # 计算指数部分
    pdf = coeff * np.exp(exponent)
    # 返回概率密度值
    return pdf

## 设置参数
f mu = 0
sigma = 1
# 定义 x 轴取值范围
x = np.linspace(-8, 8, 1000)

## 可视化
g plt.figure(figsize=(6, 3))
h pdf = gaussian_pdf(x, mu, sigma) # 计算 PDF
i plt.plot(x, pdf, color='#0066FF') # 绘图
j plt.axvline(x = mu, color = 'r', ls = '--')
plt.xlabel(r"$x$")
plt.ylabel(r"PDF, $f_{\{X\}}(x)$")
plt.grid(True, color = '0.8')
plt.xlim(-8, 8)
plt.xticks(np.arange(-8, 9))
plt.ylim(0, 0.5)
plt.tight_layout()

```

累积分布函数

在理解了概率质量函数 PMF 之后，我们再来看另一种刻画随机变量分布规律的重要工具：**累积分布函数** (Cumulative Distribution Function, CDF)。

对于离散随机变量来说，CDF 可以理解为“概率的累计过程”。CDF 描述的是：随机变量 X 取值小于或等于某个特定值 x 的概率。换句话说，CDF 告诉我们从样本空间的最小值一直累加到 x 为止的总概率有多大。

形式上，离散随机变量 X 的累积分布函数 $F_X(x)$ 定义为：

$$F_X(x) = \Pr(X \leq x) = \sum_{t \leq x} p_X(t) \quad (9)$$

这里的求和符号表示从随机变量的最小取值开始，把所有不大于 x 的概率值都加起来。也就是说，CDF 就是 PMF 的“累加结果”。

离散随机变量 X 的取值范围为 $a < X \leq b$ 时，对应的概率可以利用 CDF 计算：

$$\Pr(a < X \leq b) = F_X(b) - F_X(a) \quad (10)$$

我们仍以“掷一颗公平的六面色子”为例来理解这一概念。计算其累积分布函数 $F_X(x)$ 可以按如下方式计算：

$$\begin{aligned} F_X(x) &= 0, & x < 1 \\ F_X(x) &= 1/6, & 1 \leq x < 2 \\ F_X(x) &= 2/6, & 2 \leq x < 3 \\ F_X(x) &= 3/6, & 3 \leq x < 4 \\ F_X(x) &= 4/6, & 4 \leq x < 5 \\ F_X(x) &= 5/6, & 5 \leq x < 6 \\ F_X(x) &= 1, & 6 \leq x \end{aligned} \quad (11)$$

图 6 对应的 CDF 图像为图 8。

观察图 8，我们可以发下在抛一颗色子的随机试验中，随机变量表示色子朝上的点数，其可能取值为 1 到 6。累计分布函数描述了随机变量小于或等于某个值的概率。在这种离散情况下，CDF 呈阶梯状，每个可能取值对应一次水平跳跃。

一元离散随机变量 X 的累积分布函数 CDF $F_X(x)$ 单调不减；如果 $a < b$

$$F_X(a) \leq F_X(b) \quad (12)$$

这意味着，随着点数增大，CDF 的取值要么增大，要么不变，但不会减小。

只有在随机变量可能取值的点上，CDF 才发生跳跃；每次跳跃的高度等于该点出现的概率。对于质量均匀的色子，每次跳跃的高度相等，即 $1/6$ 。

CDF 在最小点数之前 ($x < 1$) 为 0，在最大点数处以及之后 ($x \geq 6$) 达到 1。如图 8 所示，CDF 图像整体从 0 开始逐步跳升到 1，直观地反映了离散随机变量的累积概率分布特性。

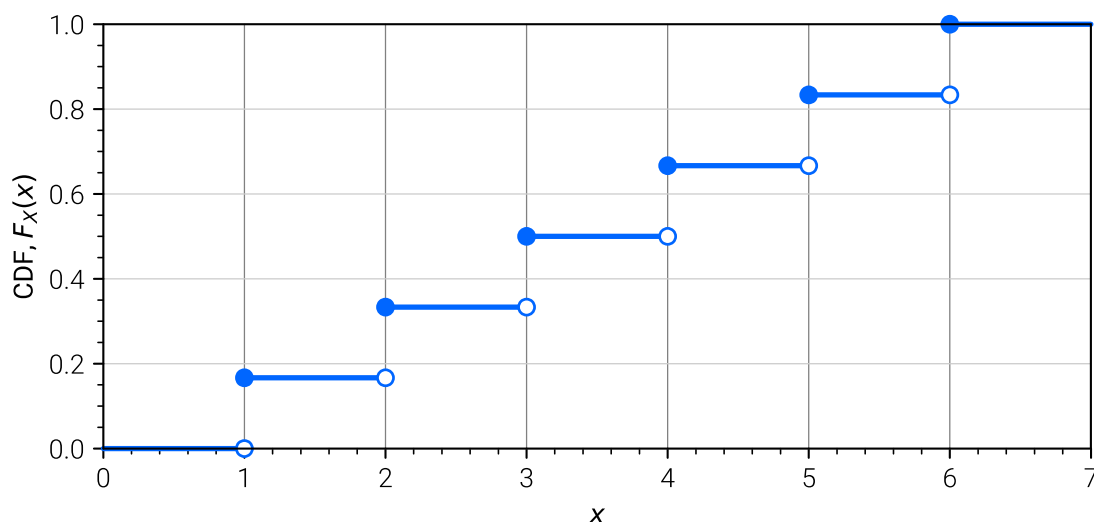


图 8. 离散随机变量 X 的 CDF，掷一颗色子



请大家用 DeepSeek/ChatGPT 等工具完成本节如下习题。

- Q1.** 在抛硬币实验中，为什么把“正面映射为 1、反面映射为 0”可以方便概率计算？这种映射体现了随机变量的什么性质？
- Q2.** 解释“随机变量是一个函数”这句话的含义，并结合掷色子的例子说明。
- Q3.** 什么时候一个随机变量被称为“离散随机变量”？请举两个生活中的例子。
- Q4.** 连续随机变量的取值有什么特点？为什么人的身高属于连续随机变量？
- Q5.** 什么是概率质量函数 (PMF)？它的输入是什么？输出又是什么？
- Q6.** 对于离散随机变量 X ，其 PMF $p_X(x)$ 需要满足哪两个重要条件？
- Q7.** 在掷一颗质地均匀色子实验中，为什么随机变量 X 的 PMF 形状呈现“六根等高的竖线”？
- Q8.** 请手绘图 5 对应的 CDF 图像。