Introduction to Data Science

- Random Variables & Probability Distribution Functions -

Mauricio Molina

Keio University, Faculty of Economics

May 28, 2025

Contents

Random Variables: Core Ideas

- **Random variable** = numeric outcome of a random process, together with its probability distribution.
- *Discrete* example: one coin flip $X \in \{0,1\}$, $P(X{=}1) = 0.5$, $P(X{=}0) = 0.5$.
- **Expected value** $E[X] = \sum_x x\, P(X{=}x)$. For the coin: $E[X] = 0 \cdot 0.5 + 1 \cdot 0.5 = 0.5$.
- **Conditioning** works on random variables, too. Two-child example: number of girls X has $P(X{=}0) = \frac{1}{4}$, $P(X{=}1) = \frac{1}{2}$, $P(X{=}2) = \frac{1}{4}$.
  - Given "at least one girl", Y has $P(Y{=}1) = \frac{2}{3}$, $P(Y{=}2) = \frac{1}{3}$.
  - Given "older child is a girl", Z has $P(Z{=}1) = \frac{1}{2}$, $P(Z{=}2) = \frac{1}{2}$.

Probability Mass Function (pmf)

For a **discrete** random variable X that takes values $x_1, x_2, \ldots$, the **probability mass function** is

$$p(x) = P(X = x).$$

- $p(x) \geq 0$ for every x.
- $\sum_i p(x_i) = 1$.
- All probabilities of interest are point masses, e.g. $P(X \in \{2, 3\}) = p(2) + p(3)$.

The pmf is the complete description of a discrete distribution—once you know $p(x)$, you can answer any probability question about X.

CDF of a Discrete Random Variable

Definition If X is discrete with probability mass function $p(x) = P(X = x)$, its cumulative distribution function is

$$F(x) = P(X \leq x) = \sum_{t \leq x} p(t).$$

Key Properties

- **Step function.** $F(x)$ is constant between successive support points and jumps only where $p(t) > 0$.
- **Jump size = pmf.** At any support value x_i, $F(x_i) - F(x_i^-) = p(x_i)$.
- **Boundary limits.** $\lim_{x \to -\infty} F(x) = 0$ and $\lim_{x \to \infty} F(x) = 1$.

Takeaway: Knowing the discrete CDF is equivalent to knowing the pmf; the jumps reveal the exact point probabilities.

CDF Table for a Fair Die

For a fair six-sided die, the cumulative distribution function $F(x) = P(X \leq x)$ is shown below:

x	$x < 1$	1	2	3	4	5	6
$F(x)$	0	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{3}{6}$	$\frac{4}{6}$	$\frac{5}{6}$	1

Reading the table:

- $F(3) = \frac{3}{6} = 0.5$ means half the time a roll is ≤ 3.
- The CDF jumps by $\frac{1}{6}$ at each face value because the pmf assigns $\frac{1}{6}$ probability to every outcome.

Probability Density Function (pdf)

Definition A **probability density function** $f(x)$ describes a continuous random variable X such that for any interval $[a, b]$,

$$P(a \le X \le b) = \int_a^b f(x)\,dx.$$

Fundamental Properties

- $f(x) \ge 0$ for all real x.
- $\displaystyle\int_{-\infty}^{\infty} f(x)\,dx = 1 \quad$ (total probability equals 1).
- The pdf itself is *not* a probability; it is "probability mass per unit length." Single points have zero probability: $P(X = c) = 0$.
- Connection to CDF: $F(x) = \displaystyle\int_{-\infty}^{x} f(t)\,dt$ and $f(x) = F'(x)$ wherever the derivative exists.

Intuition: For a tiny width h, $P(x \le X \le x + h) \approx h\,f(x)$; the pdf is the height of the probability landscape at x.

cdf for a Continuous Random Variable

Definition For a continuous random variable X with density $f(t)$, the **cumulative distribution function** is given by:

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(t)\, dt.$$

Key Properties

- **Smoothness.** $F(x)$ is continuous and non-decreasing; $\lim_{x \to -\infty} F(x) = 0$, $\lim_{x \to \infty} F(x) = 1$.
- **Derivative links to pdf.** Wherever the derivative exists,

$$\boxed{F'(x) = f(x)}.$$

- **Interval probabilities.** $P(a \leq X \leq b) = F(b) - F(a)$ for any $a < b$.

Takeaway: The CDF translates the area under the density curve into direct probabilities; knowing F fully characterises a continuous distribution.

Example: Uniform(0,1) CDF

Density:

$$f(x) = \begin{cases} 1, & 0 \le x < 1, \\ 0, & \text{otherwise.} \end{cases}$$

CDF:

$$F(x) = \begin{cases} 0, & x < 0, \\ x, & 0 \le x < 1, \\ 1, & x \ge 1. \end{cases}$$

- Linear growth between 0 and 1 reflects equal weight everywhere.
- Slope $F'(x) = 1 = f(x)$ on $(0, 1)$ confirms the derivative–density link.
- Example probability: $P(0.2 \le X \le 0.3) = F(0.3) - F(0.2) = 0.1$.

Discrete Uniform Distribution on $\{1, \dots, n\}$

Definition A random variable X is *discrete uniform* on the first n positive integers if every value from 1 to n is equally likely:

$$P(X = k) = \frac{1}{n}, \quad k = 1, 2, \dots, n.$$

Properties

- **Support:** $\{1, 2, \dots, n\}$.
- **PMF:** constant $1/n \Rightarrow \sum_{k=1}^{n} P(X = k) = 1$.
- **Expected value:**

$$E[X] = \frac{1 + n}{2}.$$

- **Variance:**

$$\mathrm{Var}[X] = \frac{n^2 - 1}{12}.$$

Example Fair six-sided die ($n = 6$): $E[X] = 3.5$, $\mathrm{Var}[X] = 35/12 \approx 2.92$.
Key: Use the discrete uniform when you have no reason to prefer one integer outcome over another within a finite range.

Discrete Uniform

```
def roll_dice(n):
 """
 Roll a fair six-sided die n times, plot relative frequencies,
 and return a list [f1,...,f6] with those frequencies.
 """
 faces  = [1, 2, 3, 4, 5, 6]
 counts = [0] * 6

 for _ in range(n):
 r = random.randint(1, 6)
 counts[r - 1] += 1

 rel_freq = [c / n for c in counts]

 # --- bar chart ---
 plt.bar(faces, rel_freq, tick_label=faces)
 plt.xlabel("Die face")
 plt.ylabel("Relative frequency")
 plt.ylim(0, 1)
 plt.title(f"Relative Frequencies for {n} Rolls")
 plt.show()

 return rel_freq
```

With this function, you can roll one dice n-times and plot the results in a bar chart.

Bernoulli Distribution

Definition A random variable X is *Bernoulli* with parameter p $(0 \leq p \leq 1)$ if it takes value

$$X = \begin{cases} 1 & \text{with probability } p, \\ 0 & \text{with probability } 1 - p. \end{cases}$$

We write $X \sim \text{Bernoulli}(p)$.

Properties

- **PMF:** $P(X = k) = p^k(1-p)^{1-k}$ for $k \in \{0, 1\}$.
- **Expected value:** $E[X] = p$.
- **Variance:** $\mathrm{Var}[X] = p(1-p)$.

Example

- One coin flip with "success = heads" has $p = 0.75$. Then $E[X] = 0.75$, $\mathrm{Var}[X] = 0.1875$.
- Indicator variables in probability proofs often follow Bernoulli distributions.

The Bernoulli is the building block for many models, e.g, sums of independent Bernoulli variables form the Binomial distribution.

Python Function: toss_coins

Code

```
import random
import matplotlib.pyplot as plt

def toss_coins(n, p=0.5):
 """
 Simulate n Bernoulli trials (coin tosses) with success-probability p.
 Plot a bar chart of relative frequencies and return [p_tail, p_head].
 """
 counts = [0, 0]

 for _ in range(n):
 outcome = 1 if random.random() < p else 0
 counts[outcome] += 1

 rel_freq = [c / n for c in counts]

 # --- bar chart ---
 labels = ["Tails (0)", "Heads (1)"]
 plt.bar(labels, rel_freq)
 plt.ylabel("Relative frequency")
 plt.ylim(0, 1)
 plt.title(f"{n} Coin Tosses (p = {p})")
 plt.show()

 return rel_freq

# example:
# toss_coins(10_000)
```

Binomial Distribution

Scenario Perform n independent Bernoulli trials, each with success probability p.
Definition Let X be the number of successes. Then

$$X \sim \text{Binomial}(n, p), \quad \text{with pmf} \quad P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}, \quad k = 0, 1, \ldots, n.$$

Key Properties

- **Support:** integers 0 through n.
- **Expected value:** $E[X] = np$.
- **Variance:** $\mathrm{Var}[X] = np(1-p)$.
- **Additivity:** Sum of independent Bernoulli(p)s.

Example Number of heads in $n = 10$ fair coin flips: $X \sim \text{Binomial}(10, 0.5)$.

$$P(X = 5) = \binom{10}{5} 0.5^{10} \approx 0.246.$$

Comment: For large n and moderate p, the Binomial's shape approaches a normal curve with the same mean and variance (*De Moivre–Laplace* approximation).

Python Simulation: Binomial `np.random.binomial`

Code

```
def simulate_binom(n_trials=10_000, n=10, p=0.5):
    """
    Draw n_trials samples from Binomial(n, p),
    plot a normalized histogram, and return the sample array.
    """
    samples = np.random.binomial(n, p, size=n_trials)

    plt.hist(samples, bins=bins, density=True, rwidth=0.8)
    plt.xlabel("Number of successes (k)")
    plt.ylabel("Relative frequency")
    plt.title(f"Histogram of Binomial({n}, {p}) | {n_trials:,} samples")
    plt.xticks(range(n + 1))
    plt.show()

    return samples

# example usage:
# simulate_binom(n_trials=1000, n=10, p=0.5)
```

Normal (Gaussian) Distribution

Definition A continuous random variable X is *Normal* with mean μ and standard deviation $\sigma > 0$ (notation $X \sim \mathcal{N}(\mu, \sigma^2)$) if it has pdf

$$f(x) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

Key Properties

- **Shape:** symmetric "bell curve" centred at μ; spread controlled by σ.
- **Moments:** $E[X] = \mu, \ \mathrm{Var}[X] = \sigma^2$.
- **Standard normal:** $Z \sim \mathcal{N}(0, 1)$. Any normal can be standardised via $Z = (X - \mu)/\sigma$.
- **Empirical Rule rule:** About 68 % of mass within $\pm 1\sigma$, 95 % within $\pm 2\sigma$, 99.7% within $\pm 3\sigma$.

Comment: Thanks to the Central Limit Theorem, sums and averages of many independent variables tend toward the normal, making it the workhorse of statistical modelling and inference.

Python: Plot Normal pdf and cdf

Code

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

# mean, std. dev
mu, sigma = 0, 1
xs = np.linspace(mu - 4*sigma, mu + 4*sigma, 400)

plt.figure(figsize=(6,4))
plt.plot(xs, norm.pdf(xs, mu, sigma),  label="pdf  $\mathcal N(0,1)$")
plt.plot(xs, norm.cdf(xs, mu, sigma),  label="cdf  $\mathcal N(0,1)$")
plt.xlabel("x")
plt.title("Normal pdf and cdf")
plt.legend()
plt.grid(alpha=0.3)
plt.show()
```

Assume adult people heights follow $X \sim \mathcal{N}(\mu = 175 \text{ cm}, \ \sigma = 7 \text{ cm})$.
Question A – What's the probability an adult man is taller than 185 cm?

$$z = \frac{185 - \mu}{\sigma} = \frac{185 - 175}{7} = 1.43, \quad P(X > 185) = 1 - \Phi(1.43) \approx 1 - 0.9236 = \boxed{0.0764}.$$

Question B – Probability that height lies between 160 cm and 190 cm?

$$z_1 = \frac{160 - 175}{7} = -2.14, \qquad z_2 = \frac{190 - 175}{7} = 2.14,$$

$$P(160 \leq X \leq 190) = \Phi(2.14) - \Phi(-2.14) \approx 0.9834 - 0.0166 = \boxed{0.9678}.$$

```
from scipy.stats import norm
mu, sigma = 175, 7
print("P(X>185)     =", 1 - norm.cdf(185, mu, sigma))
print("P(160<=X<=190)=", norm.cdf(190, mu, sigma) -
norm.cdf(160, mu, sigma))
```

Converting to a z-score lets us read probabilities from the standard normal CDF Φ; software libraries automate this step.

Student–t Distribution

Definition If $Z \sim \mathcal{N}(0,1)$ and $V \sim \chi^2_\nu$ are independent, the random variable

$$T = \frac{Z}{\sqrt{V/\nu}}$$

follows a **Student–t** distribution with ν degrees of freedom, written $T \sim t_\nu$.

PDF

$$f(t) = \frac{\Gamma\big((\nu+1)/2\big)}{\sqrt{\nu\pi}\,\Gamma(\nu/2)}\left(1+\frac{t^2}{\nu}\right)^{-(\nu+1)/2}, \qquad -\infty < t < \infty.$$

- **Mean**: 0 for $\nu > 1$.
- **Variance**: $\nu/(\nu-2)$ for $\nu > 2$ (infinite when $1 < \nu \leq 2$).
- **Behaviour**: heavy tails; approaches $\mathcal{N}(0,1)$ as $\nu \to \infty$.
- **Use**: small-sample inference for unknown variance (e.g. t-tests, confidence intervals).

Python: Simulate and Plot a t_ν Distribution

Code

```
from scipy.stats import t

df = 5
samples = np.random.standard_t(df, size=20_000)

xs = np.linspace(-6, 6, 400)
plt.hist(samples, bins=60, density=True, alpha=0.4, label="histogram")
plt.plot(xs, t.pdf(xs, df), "k-", lw=2, label=f"t pdf (df={df})")
plt.title("Student-t distribution")
plt.xlim(-5,5)
plt.xlabel("x")
plt.ylabel("density")
plt.legend()
```

Joint Distribution of Two Variables

Let (X, Y) be a pair of random variables.

Discrete case Joint pmf: $p_{X,Y}(x,y) = P(X = x,\ Y = y)$ with $\sum_x \sum_y p_{X,Y}(x,y) = 1$.

Continuous case Joint pdf: $f_{X,Y}(x,y) \geq 0$ such that $\iint_{\mathbb{R}^2} f_{X,Y}(x,y)\,dx\,dy = 1$.

Marginals

$$p_X(x) = \sum_y p_{X,Y}(x,y), \qquad f_Y(y) = \int_{-\infty}^{\infty} f_{X,Y}(x,y)\,dx.$$

The joint distribution encodes every probabilistic relationship between X and Y; marginals and conditionals are obtained via summation or integration.

Conditional Distribution & Conditional Expectation

Conditional pmf / pdf

$$p_{X|Y}(x \mid y) = \frac{p_{X,Y}(x,y)}{p_Y(y)}, \qquad f_{X|Y}(x \mid y) = \frac{f_{X,Y}(x,y)}{f_Y(y)} \quad (p_Y(y), f_Y(y) > 0).$$

Conditional expectation

$$E[X \mid Y = y] = \begin{cases} \sum_x x\, p_{X|Y}(x \mid y), & \text{discrete,} \\ \int_{-\infty}^{\infty} x\, f_{X|Y}(x \mid y)\, dx, & \text{continuous.} \end{cases}$$

Law of Total Expectation $E[X] = E\big[E[X \mid Y]\big]$.
Use: conditioning simplifies problems by "freezing" one variable and averaging later.

Independence of Two Random Variables

X and Y are **independent** if their joint distribution factorises:

- **Discrete:** $p_{X,Y}(x,y) = p_X(x)\,p_Y(y)$ for all x, y.
- **Continuous:** $f_{X,Y}(x,y) = f_X(x)\,f_Y(y)$ for all x, y.

Equivalent statements:

- $P(X \le x,\ Y \le y) = F_X(x)F_Y(y)$ (product of CDFs).
- $E[g(X)\,h(Y)] = E[g(X)]\,E[h(Y)]$ for suitable functions g, h.

Implication: If independent, knowing $Y = y$ gives no information about X: $p_{X|Y}(x \mid y) = p_X(x)$.

Bivariate Normal Distribution

Definition A random vector $\mathbf{X} = (X, Y)^\top$ follows a *bivariate normal* distribution with mean $\boldsymbol{\mu} = (\mu_X, \mu_Y)^\top$ and covariance matrix $\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_X^2 & \rho\sigma_X\sigma_Y \\ \rho\sigma_X\sigma_Y & \sigma_Y^2 \end{pmatrix}$ if its pdf is

$$f(x, y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} \exp\Big[-\tfrac{1}{2(1-\rho^2)}\Big(\frac{(x-\mu_X)^2}{\sigma_X^2} - 2\rho\frac{(x-\mu_X)(y-\mu_Y)}{\sigma_X\sigma_Y} + \frac{(y-\mu_Y)^2}{\sigma_Y^2}\Big)\Big].$$

Key Properties

- **Marginals**: $X \sim \mathcal{N}(\mu_X, \sigma_X^2)$, $Y \sim \mathcal{N}(\mu_Y, \sigma_Y^2)$.
- **Correlation**: $\rho \in (-1, 1)$ determines tilt of elliptical contours.
- **Conditionals**: $X \mid Y = y \sim \mathcal{N}\big(\mu_X + \rho\frac{\sigma_X}{\sigma_Y}(y - \mu_Y), (1-\rho^2)\sigma_X^2\big)$ (and symmetrically for $Y \mid X = x$).
- **Independence**: X and Y are independent iff $\rho = 0$.

Visualization: Contours of constant density are ellipses centred at (μ_X, μ_Y); their orientation depends on ρ.

Python: Bivariate Normal pdf (Contour Plot)

Code

```
from scipy.stats import multivariate_normal

mean = [0, 0]
cov  = [[1, 0.6],
[0.6, 1]]

# grid over which to evaluate
x = np.linspace(-3, 3, 120)
y = np.linspace(-3, 3, 120)
X, Y = np.meshgrid(x, y)
pos = np.dstack((X, Y))

rv = multivariate_normal(mean, cov)
Z = rv.pdf(pos)

plt.contourf(X, Y, Z, levels=20, cmap="viridis")
plt.xlabel("x")
plt.ylabel("y")
plt.title("Bivariate Normal pdf")
plt.colorbar(label="density")
plt.show()
```

Assignment 7

Answer all three questions in a Jupyter Notebook. Show your Python code (when requested) and a short explanation for every result. Upload the completed `.ipynb` to K-LMS by next Tuesday at midnight.

Q1: Simulate 50 000 rolls of a fair die.

- Plot the *empirical* CDF and overlay the *theoretical* step-CDF.
- Compute the sample mean and variance; compare with the theoretical values $E[X] = 3.5$ and $\mathrm{Var}[X] = 35/12$.

Q2: Let $T \sim t_5$ and $Z \sim \mathcal{N}(0, 1)$.

- Compute $P(T > 2)$ and $P(Z > 2)$.
- Use a Python plot to overlay the pdfs of t_5 and $N(0, 1)$ on the same axes.
- In a short paragraph discuss why the probabilities differ and how the t-distribution changes as the degrees of freedom increase.

Q3: Generate 100 000 samples from a bivariate normal with mean $\boldsymbol{\mu} = (0, 0)$ and covariance $\Sigma = \begin{pmatrix} 1 & 0.7 \\ 0.7 & 1 \end{pmatrix}$.

- Estimate $E[X \mid Y > 1]$ empirically.
- The conditional expectation for a bivariate normal distribution is given by: $E[X \mid Y > y] = \mu_X + \rho\sigma_X \frac{\phi\left(\frac{y-\mu_Y}{\sigma_Y}\right)}{1-\Phi\left(\frac{y-\mu_Y}{\sigma_Y}\right)}$ where $\phi(\cdot)$ and $\Phi(\cdot)$ are the pdf and cdf of an standard normal variable. Use this formula to compute the theorical value of $E[X \mid Y > 1]$ of the previous point.
- Compare simulation and theorical values; comment on any discrepancy.