Applications - Lecture 1

What You Must Know in Probability Theory

Ingénierie Mathématique & Informatique

25/09/2026

Two applications, five sections

Section	Notion	Used in
1.2	Discrete random variables	App. 2 (Poisson counts)
1.3	Random variables with density	App. 1, App. 2
1.4	Convergence & limit theorems (LLN, CLT)	App. 1, App. 2
1.5	Random variables with SciPy	App. 2

1. **Simulate** a continuous-time model: asset prices in finance.
2. **Fit & check** a discrete/continuous pair on real-looking data: server traffic with SciPy.

App. 1 - Geometric Brownian Motion in Finance

Modelling an asset price

Problem. Model the price S_t of a stock so that it (i) stays positive, (ii) has stationary independent returns, (iii) is analytically tractable.

Model (Black-Scholes-Merton).

$$dS_t = \mu S_t \, dt + \sigma S_t \, dW_t,$$

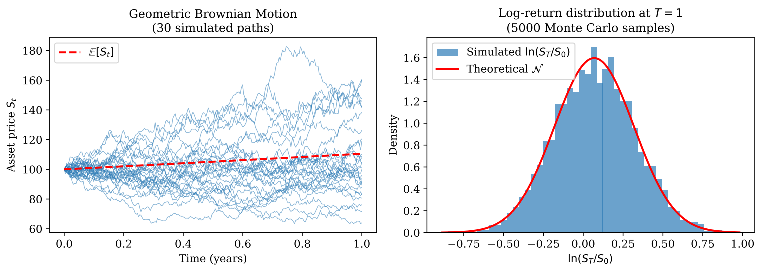
where W_t is a standard Brownian motion: $W_0 = 0$, independent increments $W_t - W_s \sim \mathcal{N}(0, t-s)$ for $t > s$.

Itô's formula applied to $\ln S_t$ gives the explicit solution

$$S_t = S_0 \exp\Big((\mu - \tfrac{1}{2}\sigma^2)t + \sigma W_t\Big) \implies \ln \frac{S_t}{S_0} \sim \mathcal{N}\Big((\mu - \tfrac{\sigma^2}{2})t, \ \sigma^2 t\Big).$$

So S_t is log-normal (Section 1.3): a smooth transform of a Gaussian.

Simulation and take-aways



Left: 30 simulated paths ($S_0 = 100$, $\mu = 0.10$, $\sigma = 0.25$), red dashed $= \mathbb{E}[S_t]$. Right: histogram of $\ln(S_T/S_0)$ vs. the theoretical Gaussian.

- Only the *law* of S_t is predictable; individual paths are erratic.

App. 2 - Server Traffic with SciPy

From a continuous clock to discrete counts

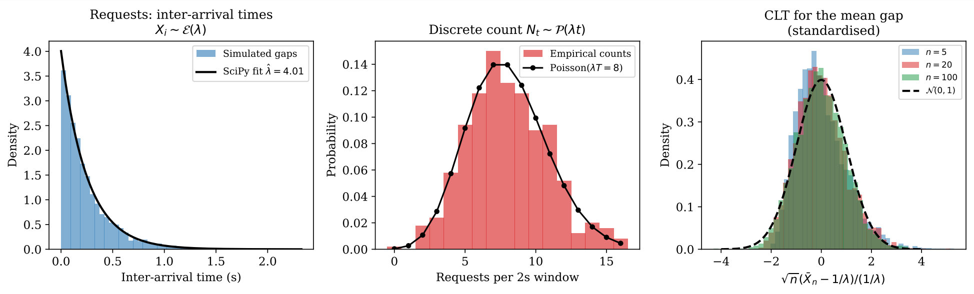
Problem. A web server receives requests at random times. We want a model for the number of requests in a window, calibrated to data with `scipy.stats`.

Model. Inter-arrival times $X_1, X_2, \ldots \overset{\text{iid}}{\sim} \mathcal{E}(\lambda)$ (Section 1.3, density $\lambda e^{-\lambda x} \mathbf{1}_{x>0}$). A classical computation shows the count N_t of requests up to time t is discrete (Section 1.2):

$$N_t \sim \mathcal{P}(\lambda t), \qquad \mathbb{P}(N_t = k) = e^{-\lambda t} \frac{(\lambda t)^k}{k!}.$$

This **Poisson process** links a density (1.3) to a discrete law (1.2).

Fitting, checking, and the SciPy toolbox



Left: gaps fitted by `expon.fit`. Middle: counts vs. `poisson.pmf`. Right: CLT for $\bar{X}_n$ (estimator of $1/\lambda$), standardised against $\mathcal{N}(0,1)$.

- The CLT (Section 1.4) that validated $\bar{X}_n$ here is the same tool behind every confidence interval in the rest of the course.