

# Stochastic Models and Simulation: Motivation & Poisson Recap

Why Poisson Processes, Queues, and Renewal Theory  
Matter

Sebastian Müller  
Lecture 1



# Framing the Gap

---

- ▶ Most data science programs focus on regression, classification, and ML.
- ▶ But what about systems that evolve **over time** and under **uncertainty?**
- ▶ Key questions:
  - ▶ How long do users wait when servers are busy?
  - ▶ When will a machine fail again?
  - ▶ What's the expected delay in a hospital triage?
- ▶ These are not supervised learning problems – they require **stochastic models**.

# Use Case: Cloud Infrastructure

---

Cloud  
Infra

Auto-scale compute to  
match bursty traffic.

- ▶ Servers handle requests arriving randomly over time.
- ▶ Resource allocation modeled via **Poisson arrivals + queuing models.**
- ▶ Objective: minimize latency, avoid overload.

*Example: AWS Auto Scaling uses modeled request rates to provision resources.*

AWS Auto Scaling monitors metrics (CPU, request count, latency) and automatically scales EC2 instances, containers, or serverless capacity up or down to match demand while controlling cost.

# Use Case: Call Centers

---



Right-size staffing,  
manage service levels.

- ▶ Incoming calls → arrival process (Poisson or empirical).
- ▶ Limited agents → queuing system ( $M/M/k$ ).
- ▶ Metrics: average wait time, service levels.

*Used by companies to staff efficiently and reduce wait times.*

# Use Case: Hospitals

---

Health  
Care

Triage limited resources  
across care stages.

- ▶ Patient arrivals: random (empirical or Poisson).
- ▶ Multi-stage service (triage, doctor, tests).
- ▶ Queueing + renewal theory to estimate wait times and allocate staff.

*Used during COVID-19 to plan emergency capacity.*

# Use Case: Predictive Maintenance

---

## Industry 4.0

Balance uptime, spare parts, and repair crews.

- ▶ Machines degrade over time → failure is random.
- ▶ Renewal processes model time-to-failure and repair.
- ▶ Used in manufacturing and IoT to schedule maintenance.

*Industrial example: Siemens, Bosch use similar models.*

# Use Case: Crypto and Finance

---

## Markets

Quantify latency,  
auctions, and  
microstructure.

- ▶ Order arrival in exchanges modeled as Poisson processes.
- ▶ Queueing logic in Ethereum gas auctions or Solana validator schedules.
- ▶ Stochastic models used to analyze MEV, fairness, and latency.

# Why Simulate?

---

- ▶ Closed-form expressions are rare or complicated.
- ▶ Simulation provides robust, interpretable insight.
- ▶ We'll use Python (with SimPy) to model real systems.

## Recap: Poisson Random Variable

---

- ▶ Parameter  $\lambda > 0$ ; support on  $\{0, 1, 2, \dots\}$ .
- ▶ Probability mass function:  $\mathbb{P}(X = k) = e^{-\lambda} \frac{\lambda^k}{k!}$ .
- ▶ Models the **count of events** in a fixed interval given independent events at rate  $\lambda$ .
- ▶ Examples: arrivals per minute, defects per meter, emails per hour.

# Key Properties

---

- ▶ Expectation and variance coincide:  $\mathbb{E}[X] = \text{Var}(X) = \lambda$ .
- ▶ **Aggregation:** sum of independent  $\text{Poi}(\lambda_i)$  is  $\text{Poi}(\sum \lambda_i)$ .
- ▶ **Thinning:** keep each event with prob.  $p \Rightarrow$  result is  $\text{Poi}(p\lambda)$ .
- ▶ Moment generating function:  $M_X(t) = \exp(\lambda(e^t - 1))$ .
- ▶ Probability of  $\geq 1$  event:  $1 - e^{-\lambda}$ .

# Poisson as a Binomial Limit

---

- ▶ Start with  $\text{Bin}(n, p)$  where  $n$  large,  $p$  small, and  $np \rightarrow \lambda$ .
- ▶ Then  $\text{Bin}(n, p)$  converges to  $\text{Poi}(\lambda)$ .
- ▶ Interpretation: many independent trials with tiny success probability.
- ▶ Use for rare-event approximation and sanity-checking rate estimates.

# From Counts to Arrival Times

---

- ▶ If counts per interval follow  $\text{Poi}(\lambda)$ , inter-arrival times are  $\text{Exp}(\lambda)$ .
- ▶ Memorylessness of the exponential  $\Rightarrow$  independent increments for counts.
- ▶ This coupling defines the homogeneous Poisson process.
- ▶ Queues and renewal models rely on this structure.

## Poisson in Simulation

---

- ▶ Python: ‘`numpy.random.poisson(lam, size)`’ samples counts directly.
- ▶ For arrival times, sample exponentials:  
`np.random.exponential(1/lambda_rate).`
- ▶ In SimPy, schedule arrivals via ‘`env.timeout(...)`’ using exponential draws.
- ▶ Validate simulated averages and variances against  $\lambda$ .

## Worked Example: Probability of an Arrival

---

- ▶ Help desk receives on average  $\lambda = 4$  tickets per hour.
- ▶ For a 30 minute window:  $\lambda_{0.5h} = 4 \times 0.5 = 2$ .
- ▶ Probability of at least one ticket:  $1 - e^{-2} \approx 0.865$ .
- ▶ Exact prob. of zero tickets in that window:  $e^{-2} \approx 0.135$ .
- ▶ Quick sanity check: expected tickets in 30 min is also 2.

## Checkpoint

---

- ▶ What data or diagnostics would tell you a Poisson model is a poor fit?
- ▶ If arrivals thin independently with prob. 0.3, what is the new rate?
- ▶ How would you simulate the time until the second arrival in Python?

# Exercise: Plain Python Warm-Up

---

1. Import numpy as np; set rng = np.random.default\_rng(42).
2. Let lam = 4 per hour; compute lambda\_window = lam \* 0.5.
3. Simulate n\_windows = 1000 half-hour counts via rng.poisson(lam=lambda\_window, size=n\_windows).
4. Report sample mean/variance and compare to the theoretical value  $\lambda = 2$ .
5. Empirically estimate  $\mathbb{P}(X \geq 1)$  by checking counts > 0.

Deliverables: short script + markdown cell summarising fit vs. theory.

## Exercise: SimPy Arrival Stream

---

1. Import numpy as np and simpy; create env = simpy.Environment() and define generator arrival(env).
2. Inside the generator, loop: yield env.timeout(np.random.exponential(1/lam)).
3. For each arrival, log the timestamp and append to a list for post-analysis.
4. Run with lam = 4 per hour for simulated duration env.run(until=8) (two hours).
5. Plot inter-arrival histogram and overlay the exponential density with rate  $\lambda$ .

Stretch goal: promote the logged arrivals into a simple M/M/1 queue and compare waiting times.

# Summary

---

- ▶ Stochastic models give tools to reason under uncertainty.
- ▶ Motivation: systems in cloud, healthcare, manufacturing, finance.
- ▶ Foundations: Poisson counts, thinning/aggregation, exponential inter-arrivals.
- ▶ Next: construct Poisson processes and simulate queues.

*"When data meets time, and time meets uncertainty — we need more than ML."*