Stochastic Models and Simulation: Motivation & Poisson Recap

Why Poisson Processes, Queues, and Renewal Theory Matter

Sebastian Müller

Lecture 1

Course Roadmap

- **Lecture 1**: Motivation, Poisson recap, simulation mindset.
- **Lecture 2**: Queueing theory basics, Markovian queues, simulation patterns.
- **Lectures 3–4**: Renewal theory, regenerative analysis, heavy tails.
- **Lectures 5–6**: Networks, resource allocation, optimisation under uncertainty.
- **Lectures 7–8**: Project clinics, validation strategies, presentations.

We move from first principles to deployable stochastic simulations in eight sessions.

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

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

Used by companies to staff efficiently and reduce wait times.

What Stakeholders Care About

- **Reliability**: likelihood the system breaches SLA thresholds.
- **Efficiency**: server utilisation vs. customer experience.
- **Elasticity**: ability to scale up/down during seasonal peaks.
- **Fairness**: which users incur delays; equity across customer classes.
- **Resilience**: recovery profile after incidents or spikes.

Queueing metrics translate these business objectives into measurable random variables.

Use Case: Hospitals

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 $\rightarrow$ 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.

Modeling Workflow

1. **Observe**: collect arrival/service traces, understand constraints.
2. **Idealise**: pick parsimonious assumptions (Poisson arrivals, exponential service, finite buffers).
3. **Analyse**: derive quick estimates (Little's Law, stability bounds).
4. **Simulate**: stress-test policy choices, explore what-if scenarios.
5. **Validate**: compare with real data, refine assumptions, communicate insights.

Each lecture revisits this loop with richer building blocks.

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 λ.
- Examples: arrivals per minute, defects per meter, emails per hour.

Key Properties

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

Poisson as a Binomial Limit

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

Estimating λ in Practice

1. Choose a consistent window (per minute/hour, per location, per channel).
2. Compute $\hat{\lambda} = \frac{1}{n}\sum X_i$ from historical counts.
3. Use variance check: Poisson assumption suggests sample variance $\approx \hat{\lambda}$.
4. Incorporate seasonality via piecewise constant or spline-based rates.
5. Apply shrinkage/Empirical Bayes for low-volume segments.

These diagnostics appear in the notebook; bring your data to test them.

Recognising Poisson Limitations

- **Overdispersion**: variance $\gg$ mean signals bursty or correlated arrivals.
- **Time-of-day effects**: use non-homogeneous Poisson or piecewise rates.
- **Feedback loops**: arrivals depend on system state (e.g., re-tries) $\Rightarrow$ need queueing networks.
- **Heavy tails**: inter-arrival distributions with long memory require renewal/ARIMA/Hawkes models.

Flag these patterns early to avoid mis-specified simulations.

From Counts to Arrival Times

- If counts per interval follow $\mathrm{Poi}(\lambda)$, inter-arrival times are $\mathrm{Exp}(\lambda)$.
- 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 λ.

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: $1 - e^{-2} \approx 0.865$.
- Probability of zero tickets in that window: $e^{-2} \approx 0.135$.

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 λ.

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.”