Customers arriving

An important event is the **arrival** of a customer to a server.

A “customer” may, in fact, be anything:

- A delivery van with a “fresh” load of hamburgers for the burger stand.
- A weekend or holiday that makes the servers stop working.
- A car door in an assembly line.
- A murder in Los Angeles prompting the attention (“service”) of the LAPD.

In this case the basic property of an arrival is its relation to time. The flow of customers can be represented in two different ways:

- as a continuous random variable describing the time periods separating successive customer arrivals.
- as a discrete random variable giving the average number of arrivals per unit of time.

The simplest and most commonly used arrival model is called the Poisson process and is made of two distributions: Exponential and Poisson.

Continuous flow

Exponential distribution

Exponential distribution: any $x \in \mathcal{R}^+$ is a legal value and the pdf is:

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} E(\mathcal{X}) = \mu &= \int_{-\infty}^{\infty} x f(x) dx = \int_0^{\infty} x \lambda e^{-\lambda x} dx \\ &= \frac{1}{\lambda} e^{-\lambda x} \mid_0^{\infty} = \frac{1}{\lambda} \end{aligned}$$

The variance:

$$V(\mathcal{X}) = \int_0^{\infty} x^2 f(x) dx = \int_0^{\infty} x^2 \lambda e^{-\lambda x} dx = \frac{1}{\lambda^2}$$

It so happens that $\sigma = \mu = \frac{1}{\lambda}$.

To generate an exponentially distributed variate with mean μ the following formula is most useful:

$$e = -\mu \ln(U(0,1))$$

Here $U(0,1)$ is the uniform distribution in the range of (0,1).

Note that the variates are positive because the logarithm of a number less than 1 is negative.

Note also that some uniform random number generators occasionally return 0; that would blow up the above formula, hence a safer alternative is recommended:

$$e = -\mu \ln(1 - U(0,1))$$

Discrete flow

Poisson distribution

The probability that n customers will arrive during a unit of time is:

$$p(\mathcal{X} = n) = \frac{e^{-\lambda}\lambda^n}{n!}$$

The cdf is

$$F(\mathcal{X} \leq n) = e^{-\lambda} \sum_{i=0}^{n} \frac{\lambda^i}{i!}$$

A convenient equality holds: $\sum_{n=0}^{\infty} \frac{\lambda^n}{n!} = e^{\lambda}$. This allows to discover an amazing result: $\mu = \sigma = \lambda$.

Poisson process

Consider a system in which a server process (could be made of many parallel servers) handles events (we call these events "arrivals" but they may represent something else, e.g. local telephone calls). The operation of this process is influenced by the sequence of arrival events; let this sequence be denoted by $E_1^{t_1}, \cdots, E_i^{t_i}, \cdots$. We assume that the sequence is ordered, i.e. that $\forall_i t_i \leq t_{i+1}$.

Let $N(t)$ be the number of these events that occurred in the time interval $[0, t]$ (mathematically speaking: $N(t) = i \mid t_i \leq t, t_{i+1}$).

This process is a Poisson process if

$$P(N(t) = n) = \frac{e^{-\lambda t}(\lambda t)^n}{n!}$$

Here, λ is called the **mean arrival rate**.

$$E[N(t)] = V[N(t)] = \lambda t$$

Comparing the pdf of $N(t)$

$$P(N(t) = n) = \frac{e^{-\lambda t}(\lambda t)^n}{n!}$$

with the pdf of the Poisson distribution:

$$p(\mathcal{X} = n) = \frac{e^{-\lambda}\lambda^n}{n!}$$

we note that the Poisson distribution is just a special case of the Poisson process in which we consider the arrival rate per unit of time (i.e. $t = 1$).

In a Poisson process with a mean rate λ, the distribution of interarrival times is exponential with a mean of $\frac{1}{\lambda}$, not a surprising result.

Pooling and splitting

When two Poisson processes with rates λ_1 and λ_2 are combined into one, the resulting process is a Poisson process with rate $\lambda_1 + \lambda_2$.

When a Poisson process with rate λ is split into two processes with arrivals going to one of them with probability p, the resulting processes are Poisson with rates λp and $\lambda(1 - p)$.

Poisson variates

```
int Poisson( double lambda )
{
   int N ;
   double T = 0 ;
   for( N = 0 ; T <= 1 ; N++ )
      T += Exponential( 1/lambda ) ;
   return N − 1 ;
}
int Poisson( double lambda )
{
   int N ;
   double P = exp( −lambda ) ;
   double T = 1 ;
   for( N = 0 ; T >= P ; N++ )
      T ∗= drand48() ;
   return N − 1 ;
}
```