Computational Statistics

Lecture 4: Simulating from Discrete Random Variables

Raymond Bisdorff

University of Luxembourg

6 octobre 2021

Content of Lecture 4

Simulating a Bernoulli random variable

Consider a student who guesses on a multiple choice test question which has five options : the student may guess correctly with probability 0.2 and incorrectly with probability $1 - 0.2 = 0.8$. How well is doing this student in a simulated test consisting of 20 questions ?

```
> set.seed(23207)
> guesses = runif(20)
> correctAnswers = (guesses < 0.2)
> table(correctAnswers)
correctAnswers
FALSE  TRUE
   14     6
```

The student would score in this simulated test 6/20, i.e. 6 correct answers out of 20 showing an empirical success probability of $6/20 = 0.3$.

Simulating a binomial random variable

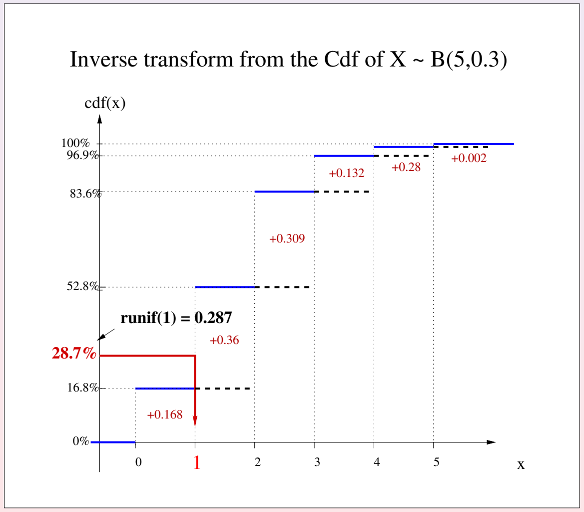
The sum X of m independent Bernoulli random variables, coded : 0 (False) and 1 (True), each having a success probability of p gives a binomial random variable $\sim \mathcal{B}(m, p)$ representing the number of successes in m Bernoulli trials. X can take values in the set $\{0, 1, 2, ..., m\}$ with probability :

$$P(X = x) = \binom{m}{x} p^x (1-p)^{m-x}, \; x = 0, 1, 2, ..., m.$$

We may compute in R the probability of observing 6 successes in 20 trials, when the success probability is 0.2 :
`> dbinom(x=6,size=20,prob=0.2) = 0.1090997` .

Simulating a discrete random variable by inverse transform

```
> db=dbinom(0:5,5,0.3)
[1] 0.16807 0.36015
[3] 0.30870 0.13230
[6] 0.02835 0.00243
# cumsum(db) = cdf
> pbinom(0:5,5,0.3)
[0] 0.16807
[1] 0.52822
[2] 0.83692
[3] 0.96922
[4] 0.99757
[5] 1.00000
> u = runif(1)
[1] 0.287
# inv. cdf = quantile
> qbinom(u,5,0.3)
[1] 1
> rbinom(nSim,5,0.3)
[1] 1 2 3 1 2 ...
```



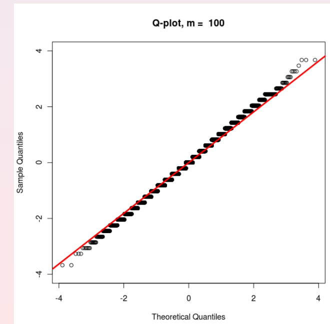
The Central Limit Theorem for binomial variables

If $X \sim \mathcal{B}(m, p)$, and

$$Z = \frac{X - mp}{\sqrt{mp(1-p)}},$$

then $Z \rightsquigarrow \mathcal{N}(0, 1)$ when m gets large.

```
> nSim = 10^4
> m = 100
> p = 0.4
> Z = (rbinom(nSim,size=m,prob=p) - m*p)/
+      sqrt(m*p*(1-p))
> qqnorm(Z, ylim=c(-4,4),
+        main = paste("Q-plot. m = ", m))
> qqline(Z)
```



Simulating a Poisson random variable

The Poisson distribution $X \sim \mathcal{P}(\lambda)$ is the limit of a binomial distribution $\mathcal{B}(n, p_n)$ when $n \to \infty$ and $p_n \to 0$, but where the expected value np_n and the variance $np_n(1 - p_n)$ converge to a same constant value λ, the *rate* of the Poisson distribution.
The possible discrete values a *Poisson variable* can take are the natural numbers $\{0, 1, 2, ..\}$ with probability :

$$P(X = x) = \frac{e^{-x}\lambda^x}{x!}, \quad x = 0, 1, 2, ...$$

The *mean* and the *variance* of a Poisson variable are both equal to the rate λ.

Example of Poisson distribution

Example

Suppose traffic accidents occur at an intersection with a mean rate of 3.7 per year. Assumimg a Poisson model, a simulation of the potential number of accidents per year may be run in R like follows :

```
> nSim = 10
> rate = 3.7
> X = rpois(n=nSim,lambda=rate)
> summary(X)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    1.0     3.0     3.0     3.4     4.0     6.0
```


Poisson processes

A Poisson process is a simple model of the collection of events that occur during a given time period. A *homogenous* Poisson process has the following properties :

1. The number of events during a time period is Poisson distributed with a rate *proportional* to the observation period ;
2. The running process has *no memory of past events*, i.e. the numbers of events in non overlapping time periods are all independent one of the other.

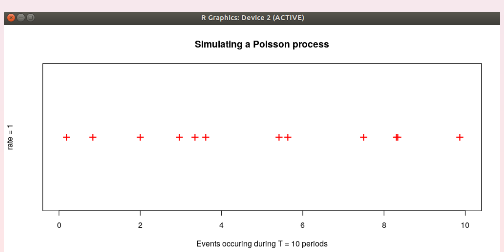
In particular, a Poisson process with rate λ observed in a period $[0, T]$ shows on average λT events.

Simulating a Poisson processes

One way to simulate a Poisson process is the following :

1. Generate n as a Poisson random number with parameter λT,
2. Generate n independent uniform random numbers on the interval $[0, T]$.

```
> lambda = 1
> T = 10
> n = rpois(1,lambda*T)
[1] 12
> events = runif(n,0,T)
> x = sort(events)
[1] 0.1841019 0.8309076 2.0048382
[4] 2.9605278 3.3489711 3.6107790
[7] 5.4219458 5.6337490 7.5043275
[10] 8.2991724 8.3431913 9.8656030
> y = rep(1,n)
> plot(x,y,pch="+",xlim=c(0,T),cex=2,
  col="red",yaxt='n',ylab='rate = 1')
```



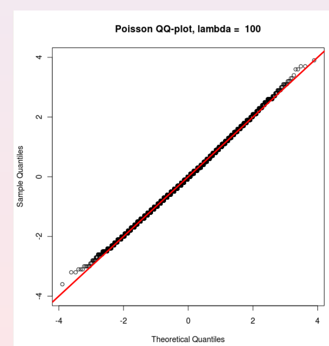
The Central Limit Theorem for Poisson variables

If $X \sim \mathcal{P}(\lambda)$, and

$$Z = \frac{X - \lambda}{\sqrt{\lambda}},$$

then $Z \rightsquigarrow \mathcal{N}(0, 1)$ if λ gets large.

```
> nSim = 10^4
> lambda = 100
> Z = (rpois(nSim,lambda) - lambda)/
+      sqrt(lambda)
> qqnorm(Z, ylim=c(-4,4),
+  main = paste("Poisson QQ-plot,/
+  lambda = ", lambda)
> qqline(z)
```



Exponential random numbers

Exponential random variables model usually such things as failure times T of mechanical or electronic components, or the time T it takes a server to complete service to a customer. The exponential distribution is characterized by a *constant failure rate*, denoted λ.
Random variable T has an exponential distribution with rate $\lambda > 0$ if its cdf F_T is the following :

$$F_T(t) \; = \; P(T \leqslant t) \; = \; 1 \, - \, e^{-\lambda t}$$

for any nonnegative t. Differentiating the distribution function with respect to t gives the exponential density function :

$$f_T(t) \; = \; \lambda e^{-\lambda t}$$

The *expected value* of an exponential random variable is $1/\lambda$ and its *variance* is $1/\lambda^2$.

Simulating T by inverse transform

Suppose $T \sim \exp(\lambda)$. Then $F_T(t) = 1 - e^{-\lambda t} \; = P(T \leqslant t)$.
If u denotes $P(T \leqslant t)$, solving for t in $u \; = \; 1 \; - \; e^{-\lambda t}$ gives

$$t = \frac{-\log(1-u)}{\lambda}.$$

Therefore, if $U \sim \mathcal{U}(0,1)$, then $1 - U \sim U$ and

$$T \; = \; -\frac{\log U}{\lambda} \; \sim \; exp(\lambda)$$

See Lesson 3 for an R example code.

Simulating a Poisson process – another way

It can be shown that the time separating two subsequent events occuring in a Poisson process of rate λ is exponentially distributed with rate λ,
This leads to a simple way for simulating a Poisson process on the fly.

Example

Simulate the moments in time where the first 25 events may occur in a Poisson process of rate 1.5.

```
> X = rexp(25, rate = 1.5)
> cumsum(X)
 [1]  0.7999769  1.0924413  2.2480730  2.6270703  2.8888372  4.5510017
 [7]  5.4118919  5.6875902  5.8969009  6.5536986  7.6601004  7.8540837
[13]  8.2793790  9.4287367 10.5200363 10.5464784 11.4369748 11.7930954
[19] 11.9409715 12.5444665 13.2704827 14.5333422 14.6247818 16.0576074
[25] 16.1842825
```

$\Gamma(\alpha, \beta)$ variables

The Gamma random variable $X \sim \Gamma(\alpha, \beta)$, with real parameters $\alpha > 0$ and $\beta > 0$, has *density* $p(x)$ for $x > 0$:

$$p(x) \; = \; \frac{\beta^\alpha}{\int_0^\infty t^{\alpha-1} e^{-t} dt} x^{\alpha-1} e^{-\beta x}.$$

The *mean and variance* are respectively given by α/β and α/β^2. In the $\Gamma(\alpha, \beta)$ probability law, the β parameter enters only as a scaling :

$$\Gamma(\alpha, \beta) \; \sim \; \frac{1}{\beta} \Gamma(\alpha, 1).$$

To generate a $\Gamma(\alpha, \beta)$ random number, it is hence sufficient to generate a $\Gamma(\alpha, 1)$ random number and divide it by β.

Integer alpha parameter

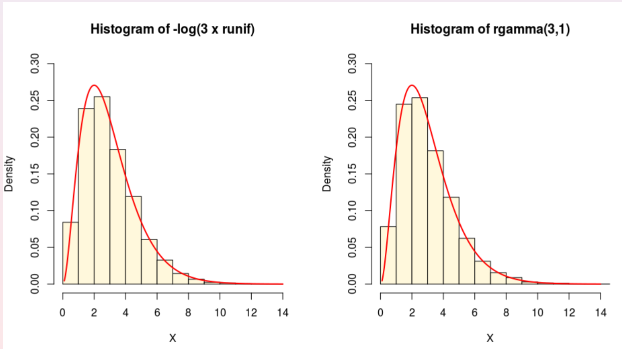
If $X \sim \Gamma(\alpha, 1)$ with α a small integer, X is in fact distributed as the waiting time to the αth event in a random Poisson process of unit mean.

Since the waiting time between two consecutive events is distributed following an exponential law with $\lambda = 1$, we can hence simply add up α exponentially distributed waiting times, i.e. logarithms of uniform random numbers.

Furthermore, since the sum of logarithms is equal to the logarithm of the product, we may simulate X by computing the product of α uniform random numbers and then take minus the log. .

Simulation and visual checking of a random variable $X \; \sim \; \mathcal{G}(\alpha = 3, \beta = 1)$

```
> nSim = 10^4
> rl3 = -log(
+   runif(nSim) *
+   runif(nSim) *
+   runif(nSim) )
> ra =
+   rgamma(nSim,3,1)
> x =
+   seq(0,14,by=0.1)
> dg = dgamma(x,3,1)
> par(mfrow=c(1,2))
> hist(rl3,freq=F)
> lines(x,dg,lwd=2)
> hist(ra,freq=F)
> lines(x,dg,lwd=2)
```



Sum rule and CLT for gamma variables

Useful properties of the gamma distribution :

1. If we have to simulate the sum of a set of independent $X_i \sim \Gamma(\alpha_i, \beta)$ variables with different α_i's, but sharing the same β parameter, we may consider that their sum $Y = \sum_i X_i$ is again distributed like a gamma variable :

$$Y \; \sim \; \Gamma(\sum_i \alpha_i, \beta).$$

2. If $X \sim \Gamma(\alpha, \beta)$ when $\alpha \ggg \beta$, then $X \; \rightsquigarrow \; \mathcal{N}(\alpha/\beta, \alpha/\beta^2)$.
3. If the α_i are integers, we may directly simulate X with the minus log of the product of the corresponding number $\sum_i \alpha_i$ of uniform random numbers, divided by β.

Simulate a Bernoulli variable

Exercise

1. *Suppose a class of* 100 *students writes a* 20*-question True-False test, and everyone in the class guesses the answers with a success probabilioty of* 0.2 *:*
   1.1 *Use simulation to estimate the average mark over the* 100 *students as well as the standard deviation of the marks.*
   1.2 *estimate the proportion of students who would obtain a mark of* 30% *or higher.*
2. *Write an R function which simulates* 500 *light bulbs, each of which has probability* 0.99 *of working. Using simulation, estimate the expected value and variance of the random variable X, which is* 1 *if the light bulb works and* 0 *if it does not work. What are the theoretical values ?*

Simulate a binomial variable

Exercise

1. *Suppose the proportion p of defective production is* 0.15 *for a manufactoring operation. Simulate the number of defectives for each hour of a* 24*-hour period, assuming* 25 *units are produced every hour. Check if the number of defectives ever exceeds* 5*. Repeat assuming $p = 0.2$ and then* 0.25*.*
2. *Write a binomial random variable generator in R with parameters : 'n' successes, 'm' trials, and success probability 'p', using the cumulated density function (cdf) inversion method.*
3. *Write a similar binomial random variable generator in R based on the summing up of corresponding independent Bernoulli random variables.*
4. *The previous generator requires m uniform pseudo random numbers for one simulated binomial number. Design a similar generator for a binomial random variable which requires only one uniform random number for each simulated binomial number.*

Simulating a Poisson process

Exercise

1. *Conduct a simulation experiment to check, on a large number ($nSim = 10^4$) of realizations on a period of* 10 *minutes, the reasonableness of the assumption that the numbers X of events from a rate* 1.5 *per minute Poisson process which occur between the fourth and fifth minute of these processes are indeed Poisson distributed with rate* 1.5*.*
2. *Use the incremental quantile agent from Lesson 5 for estimating the quantiles of distribution X.*
3. *Use the* `qqplot` *R command to graphically compare the quantiles of distribution X with the quantiles of a corresponding theoretical Poisson distribution.*