Numerical and Statistical Methods for Finance

Generating Continuous Random Variables

Pierpaolo De Blasi

University of Torino
email: pierpaolo.deblasi@unito.it
webpage: sites.google.com/a/carloalberto.org/pdeblasi/

Lecture no. 7
14 November 2013

The Inverse Transform Method

Consider a continuous random variable X with cumulative distribution function (cdf) F. A general method for generating such rv–called the *inverse transform method* –is based on the probability integral transform:

Let $U \sim Unif(0,1)$ *and set*

$$X = F^{-1}(U)$$

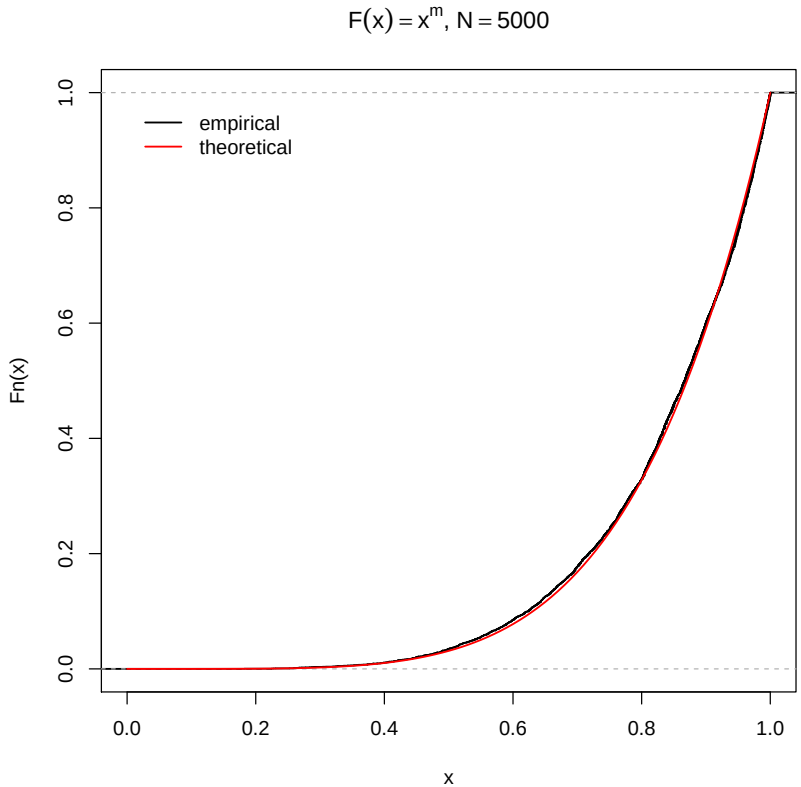
Then the rv X *has cdf* F*.*

Example 5a, page 68, Ross (2006)

$$F(x) = x^m, \quad 0 < x < 1.$$

Then $F^{-1}(u) = u^{1/m}$.

```
# given U ~ Unif(0,1)
F.inv<-function(u) u^(1/m)
X<- F.inv(U)
```



Maximum of Uniform rv's

If we let

$$U_1, \ldots, U_m \sim \text{iid Unif}(0,1), \quad U_{(m)} = \max_j U_j$$

then the cdf of $U_{(m)}$ is $F(x) = x^m$ since

$$\Pr(U_{(m)} \leq x) = \Pr(U_1 \leq x, \ldots, U_m \leq x) = \prod_{j=1}^{m} \Pr(U_j \leq x)$$

Hence an alternative method to simulate from cdf $F(x) = x^m$ is

```
U<-runif(m)
X<-max(U)
```

Triangular distribution

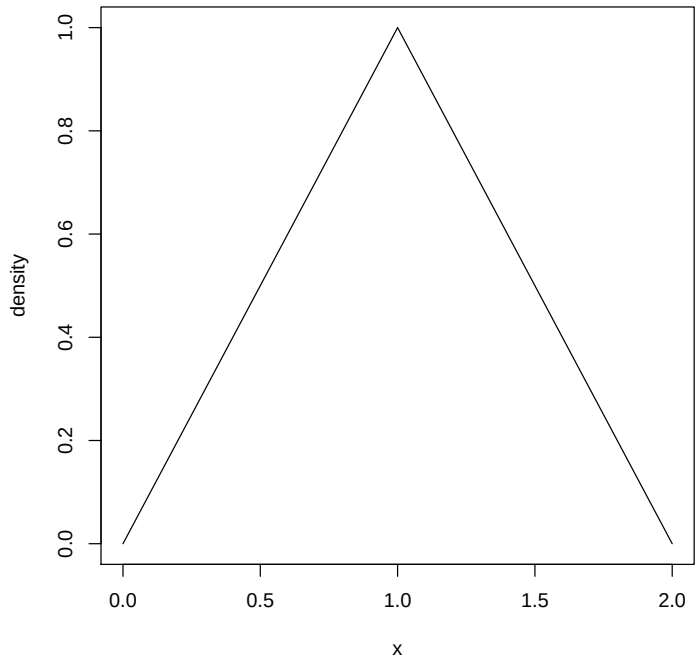
Let now $X = U_1 + U_2$. It has *triangular distribution* with density

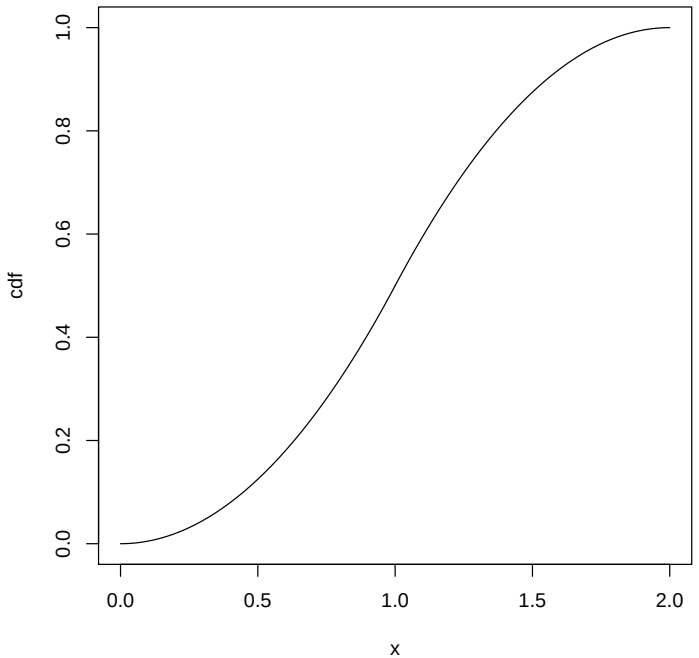
$$f(x) = \begin{cases} x & 0 \le x \le 1 \\ 2 - x & 1 < x \le 2 \end{cases}$$

and cdf

$$F(x) = \begin{cases} x^2/2 & 0 \le x \le 1 \\ 1 - (2-x)^2/2 & 1 < x \le 2 \end{cases}$$

(*why?*)





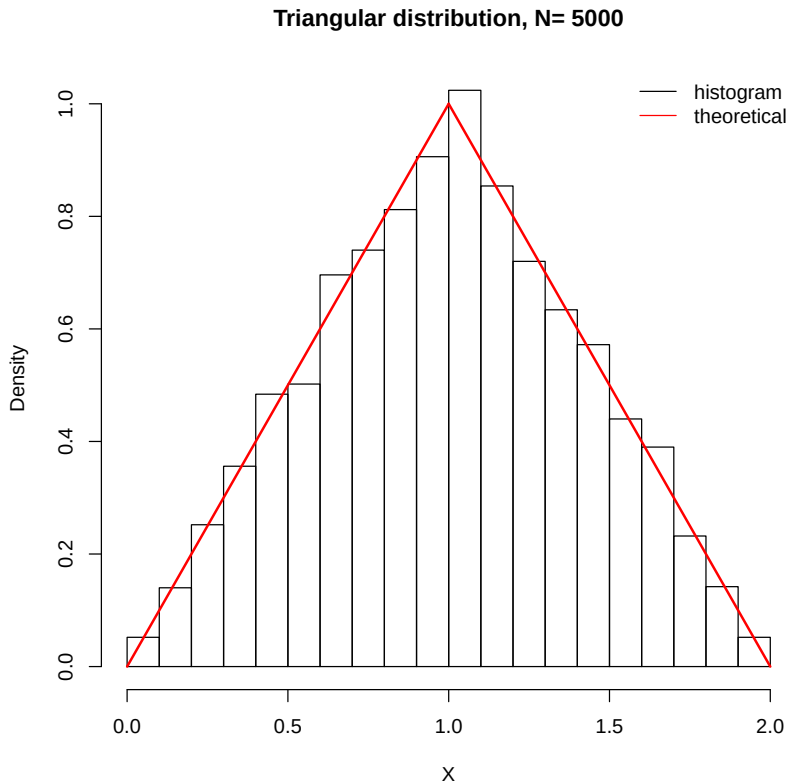
Triangular distribution

Easy to generate values of $X = U_1 + U_2$:

```
U<-runif(2)
X<-sum(U)
```

To implement the inverse transform method we may use a conditional evaluation to select the appropriate range of x in applying the inversion:

```
F.inv1<-function(u) sqrt(2*u)
F.inv2<-function(u) 2-sqrt(2*(1-u))
U<-runif(1)
if (U<0.5) {
    X<-F.inv1(U)
} else X<-F.inv2(U)
print(X)
```



Exponential Distribution

If X is an exponential rv with rate 1, its cdf is given by

$$F(x) = 1 - \exp\{-x\}$$

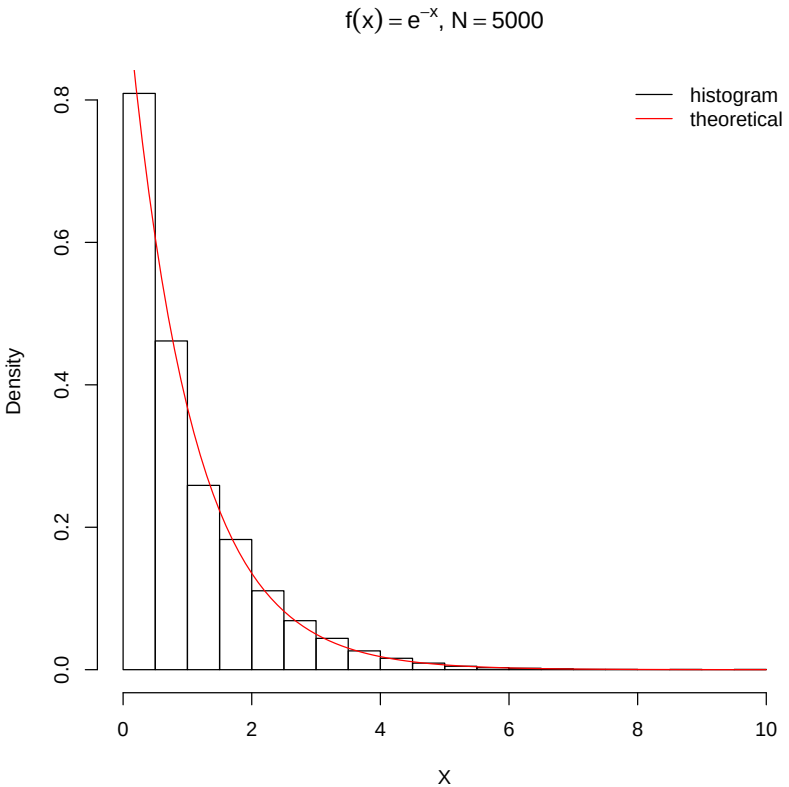
We have $F^{-1}(u) = -\log(1-u)$, hence we can generate X as follows

```
U<- runif(N)
X<- -log(1-U)
```

Note that $1 - U \stackrel{\mathcal{L}}{=} U$, hence one can equally use `X<- -log(U)`.

Recall that cX has exponential distribution with rate $1/c$ (and mean c). Hence an exponential random variable with rate λ can be generated as follows:

```
U<- runif(N)
X<- -log(U)/lambda
```



Exponential and Poisson Distribution

Recall that a Poisson process $\{N(t),\ t \geq 0\}$ with rate λ results when the times between successive events (*interarrival times*) are independent exponentials with rate λ.

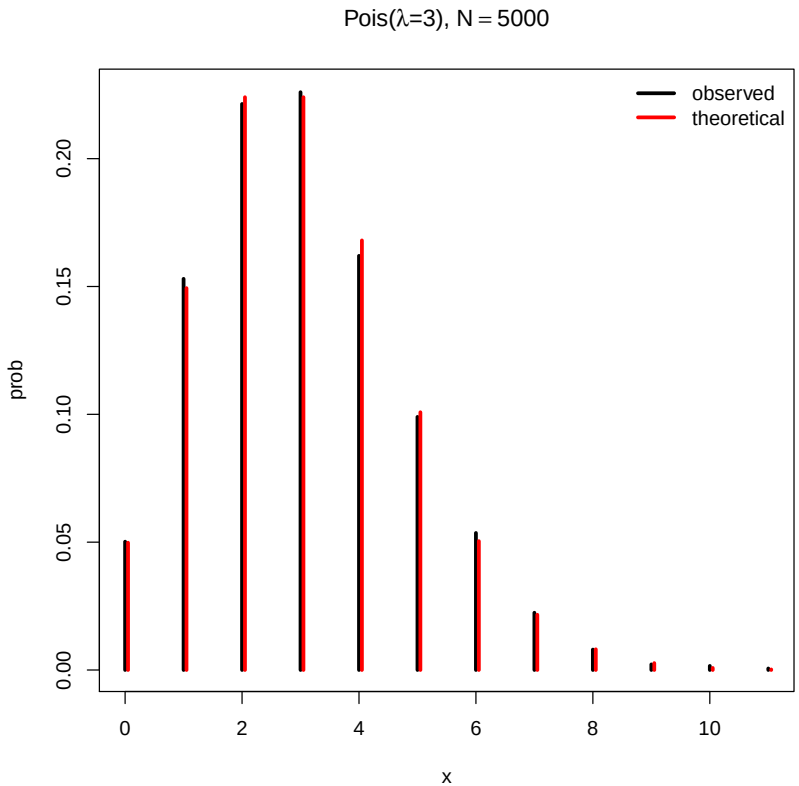
For such process, $N(1) \sim \text{Pois}(\lambda)$. If we let X_i, $i = 1, 2, \ldots$, denote the successive interarrival times, then the nth event will occur at time $\sum_{i=1}^{n} X_i$, and so the number of events by time 1 is

$$N(1) = \max\{n :\ \textstyle\sum_{i=1}^{n} X_i \leq 1\}$$

Hence we can generate $N = N(1) \sim \text{Pois}(\lambda)$ by generating random numbers $U_1, U_2, \ldots$ and setting

$$\begin{aligned} N &= \max\{n :\ \textstyle\sum_{i=1}^{n} -\lambda^{-1} \log U_i \leq 1\} \\ &= \max\{n :\ \textstyle\sum_{i=1}^{n} \log U_i \geq -\lambda\} \end{aligned}$$

```
lambda<-3
log.U<-log(runif(1))
N<-0
while (log.U>= -lambda) {
    log.U<-log.U+log(runif(1))
    N<-N+1
}
print(N)
```



Gamma Random Variable

Suppose we want to generate a gamma(n, λ) rv which has cdf

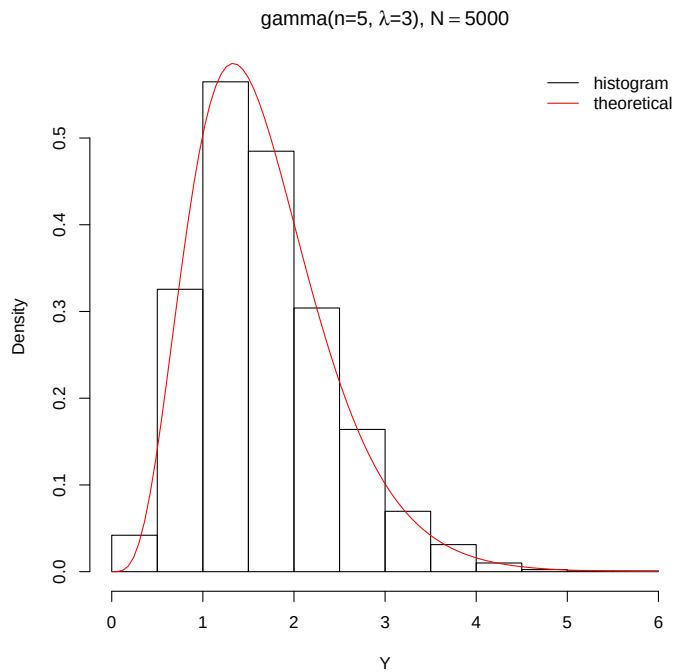
$$F(x) = \int_0^x \frac{\lambda^n}{\Gamma(n)} y^{n-1} \mathrm{e}^{-\lambda y} \mathrm{d}y = \int_0^x \frac{\lambda}{(n-1)!} (\lambda y)^{n-1} \mathrm{e}^{-\lambda y} \mathrm{d}y$$

It is not possible to invert analytically $F(x)$. However we can use the result

$$X_1, \ldots, X_n \sim \text{iid Exp}(\lambda) \Longrightarrow Y = \sum_{i=1}^n X_i \sim \text{gamma}(n, \lambda)$$

(*why?*).

```
n<-5
lambda=3
U<-runif(n)
Y<-sum(-log(U)/lambda)
print(Y)
```



Exponential through conditioning

Let X, Y be independent Exp(1), so that $X + Y \sim \text{gamma}(2, 1)$. We can exploit the result

$$X|X + Y = t \ \sim \ \text{Unif}(0, t)$$

(*why?*) to generate an exponential rv:

Step 1. Generate $U_1, U_2 \sim \text{Unif}(0, 1)$;

Step 2. Set $t = -logU_1 - \log U_2$;

Step 3. Generate $U_3 \sim \text{Unif}(0, 1)$;

Step 4. Set $X = tU_3$, $Y = t - X$.

```
U<-runif(2)
t<-sum(-log(U))
X<-runif(1)*t
Y<-t-X
```

We can also generate k independent exponential with rate 1 with a similar method.

Let $X_1, \ldots, X_k \sim$ iid Exp$(k, 1)$ so that $T = \sum_{i=1}^{k} X_i \sim$ gamma$(k, 1)$. Then

$$X_{(1)}, \ldots, X_{(k)} | T = t \sim t[U_{(i)} - U_{(i-1)}], \quad i = 1, \ldots, k$$

where $U_{(0)} = 0$, $U_{(k)} = 1$ and $U_{(1)}, \ldots, U_{(k-1)}$ are the ordered values of k independent Unif$(0, 1)$.

```
k<-5
lambda<-1
U<-runif(k)
t<-sum(-log(U)/lambda)
U.prime<-runif(k-1)
U.prime<-c(0,sort(U.prime),1)
U.prime<-U.prime[-1]-U.prime[-(k+1)]
X<-U.prime*t
```

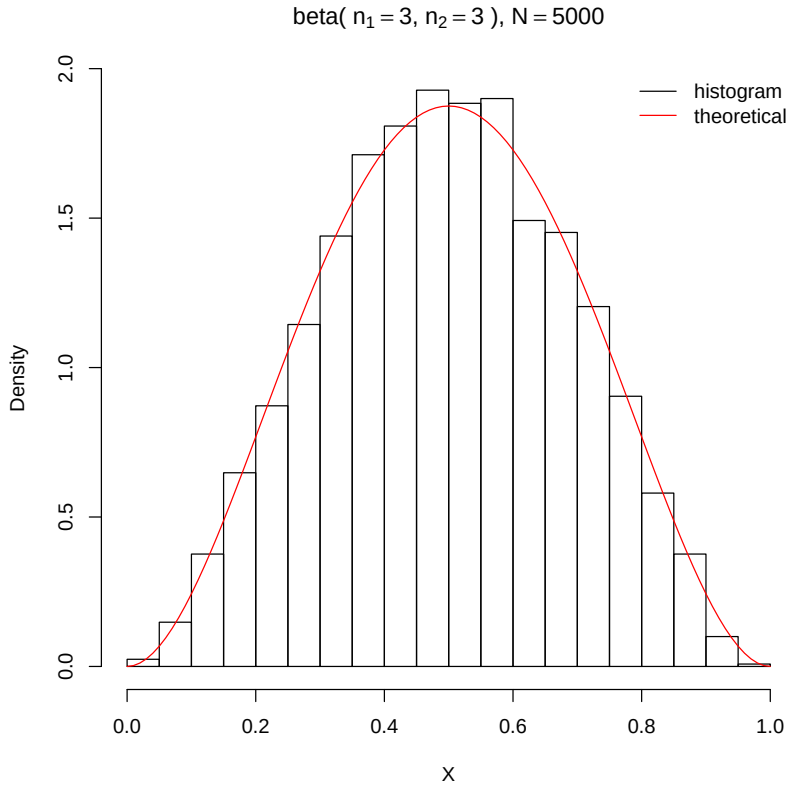
Note that by using an arbitrary value for `lambda`, the vector `X` corresponds to the ordered values of k independent Exp(λ) rv's.

Beta Random Variable

Let $Y_1 \sim \text{gamma}(n_1, \lambda)$ and $Y_2 \sim \text{gamma}(n_2, 1)$ be independent. Then $X = Y_1/(Y_1 + Y_2)$ has *beta* distribution with shape parameters n_1 and n_2, $X \sim \text{beta}(n_1, n_2)$, with density function

$$f(x) = \frac{\Gamma(n_1+n_2)}{\Gamma(n_1)\Gamma(n_2)} x^{n_1-1}(1-x)^{n_2-1}$$
$$= \frac{(n_1+n_2-1)!}{(n_1-1)!(n_2-1)!} x^{n_1-1}(1-x)^{n_2-1}, \quad x \in (0,1)$$

```
lambda<-4
n1<-2
n2<-3
U1<-runif(n1)
U2<-runif(n2)
Y1<-sum(-log(U1)/lambda)
Y2<-sum(-log(U2)/lambda)
X<-Y1/(Y1+Y2)
```



Exchangeable random vector on Δ_k

Let Δ_{k-1} be the $(k-1)$-dimensional simplex,

$$\Delta_{k-1} = \{(x_1, \ldots, x_{k-1}) : \ x_i \geq 0, \textstyle\sum_{i=1}^{k-1} x_i \leq 1\}$$

We can define a random vector $(X_1, \ldots, X_{k-1})$ with value in Δ_{k-1} as follows:

$$X_1 = Y_1/T, \ldots, X_{k-1} = Y_{k-1}/T, \quad T = \textstyle\sum_{i=1}^{k} Y_i$$

where $Y_1, \ldots, Y_k$ are iid gamma(n, λ). For $n = 1$, we have

$$X_{(i)} = U_{(i)} - U_{(i-1)}, \quad i = 1, \ldots, k-1$$

where $U_{(0)} = 0$, $U_{(k)} = 1$ and $U_{(1)}, \ldots, U_{(k-1)}$ are the ordered values of k independent Unif$(0, 1)$.

The distribution of $(X_1, \ldots, X_{k-1})$ is *exchangeable* in that

$$(X_1, \ldots, X_{k-1}) \stackrel{\mathcal{L}}{=} (X_{\pi_1}, \ldots, X_{\pi_{k-1}})$$

for any permutation $\pi_1, \ldots, \pi_{k-1}$ of the integers $1, \ldots, k-1$ (*why?*).

In particular,

$$X_i \sim \text{beta}(n, n(k-1))$$

so that $E(X_i) = 1/k$ and $\text{Var}(X_i) = \frac{n^2(k-1)}{n^2k^2(nk+1)}$ (*why?*).

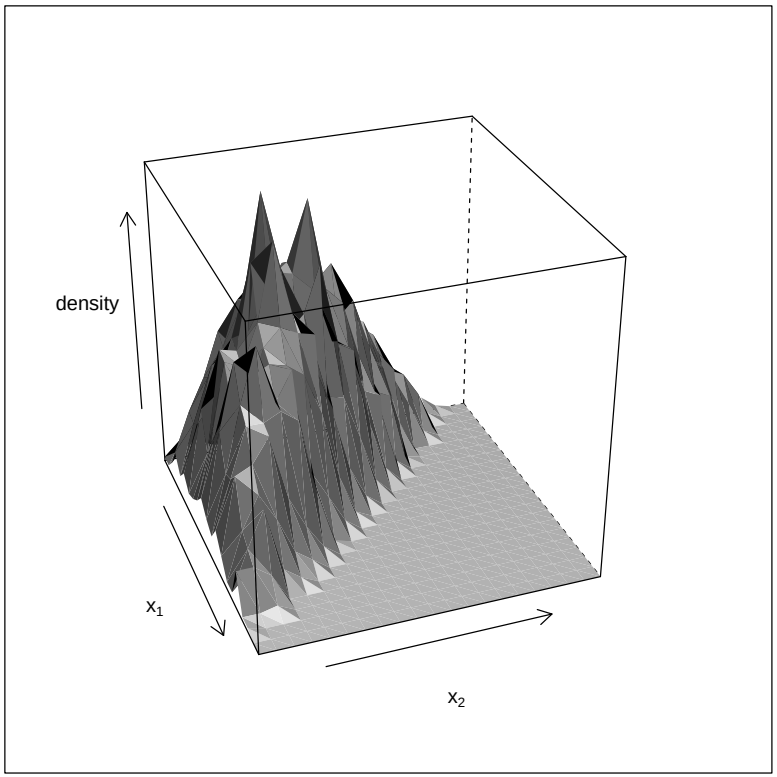
Let $X_k = 1 - \sum_{i=1}^{k-1} X_i$, then $(X_1, \ldots, X_k)$ defines a (*random*) probability distribution on the set of integers $\{1, \ldots, k\}$.

```
lambda<-4
n<-2
k<-3

U<-matrix(runif(k*n),ncol=k)
Y<-apply(-log(U)/lambda,2,sum)
X<-Y/sum(Y)

# n=1

U<-runif(k-1)
U<-c(0,sort(U),1)
X<-U[-1]-U[-(k+1)]
```

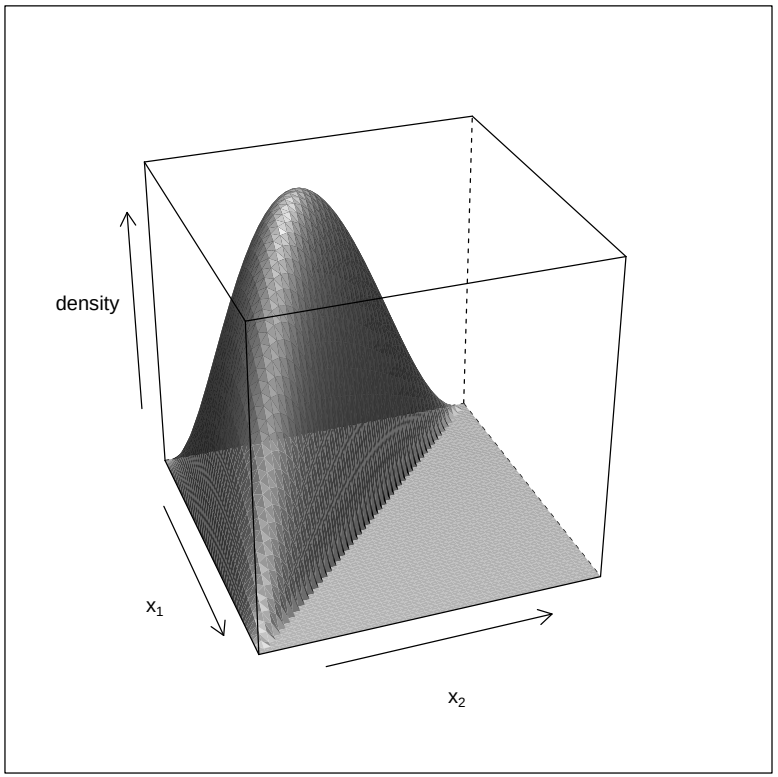
Note that the same distribution is obtained no matter the specific value of λ (*why?*).

Exchangeable random vector in Δ_{k-1} are used in resampling methods like the *weighted bootstrap*.

n=2, empirical



n=2, theoretical



Exercises

- Chapter 5, Ross (2006)
  Ex 1, Ex 2, Ex 3, Ex 4, Ex 5, Ex 6, Ex 10, Ex 12
- Chapter 18, Owen, et al. (2007)
  Ex 13, Ex 14, Ex 16, Ex 17, Ex 18

Resources

- BOOKS
  - Owen J., Maillardet R. and Robinson A. (2009).
    Introduction to Scientific Programming and Simulation Using R.
    Chapman & Hall/CRC.
  - Ross, S. (2006).
    Simulation. 4th edn. Academic Press.
- WEB
  - R software:
    `http://www.r-project.org/`
  - Owen, et al. (2009): `http://www.ms.unimelb.edu.au/spuRs/`