Simulation from Comonotonic random vectors

Daniel Linders

November 17, 2019

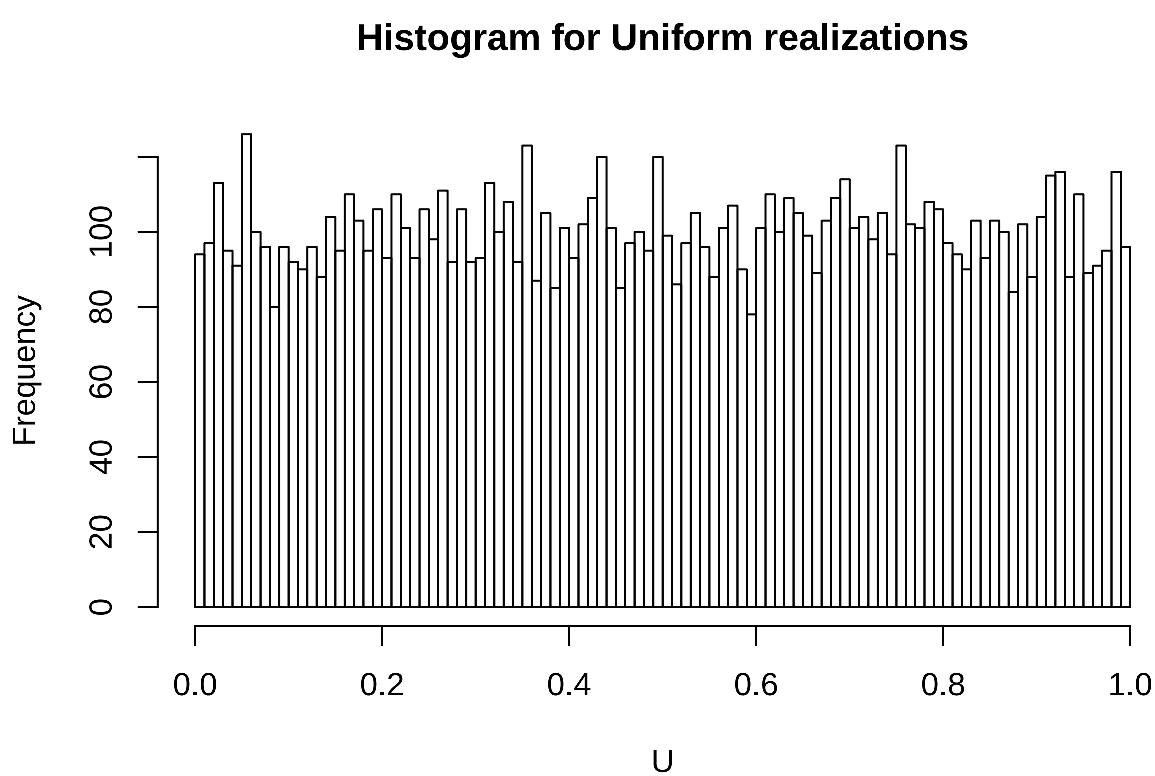
The Quantile Transform Theorem

Consider a random variable X with cumulative distribution function (cdf) denoted by F_X. The **Quantile Transform Theorem** states that:

$$F_X^{-1}(U) \stackrel{d}{=} X,$$

where U is a Uniform distributed random variable. We will employ the quanitle transform theorem to transform m realizations $u_1, u_2, \ldots, u_{N_{sim}}$ from U to realizations $x_1, x_2, \ldots, x_{N_{sim}}$ of X. Below, we start with simulating the realizations u_i, $i = 1, 2, \ldots, N_{sim}$ of U.

```r
NSim=10000
U=runif(NSim)
#Histogram of a uniform distribution
hist(U,100, main="Histogram for Uniform realizations")
```



Assume we want to transform the realizations which are stored in the vector `U` to realizations for an expontial distributed random variable X:

$$X \stackrel{d}{=} \text{Exp}(\lambda).$$

Then:

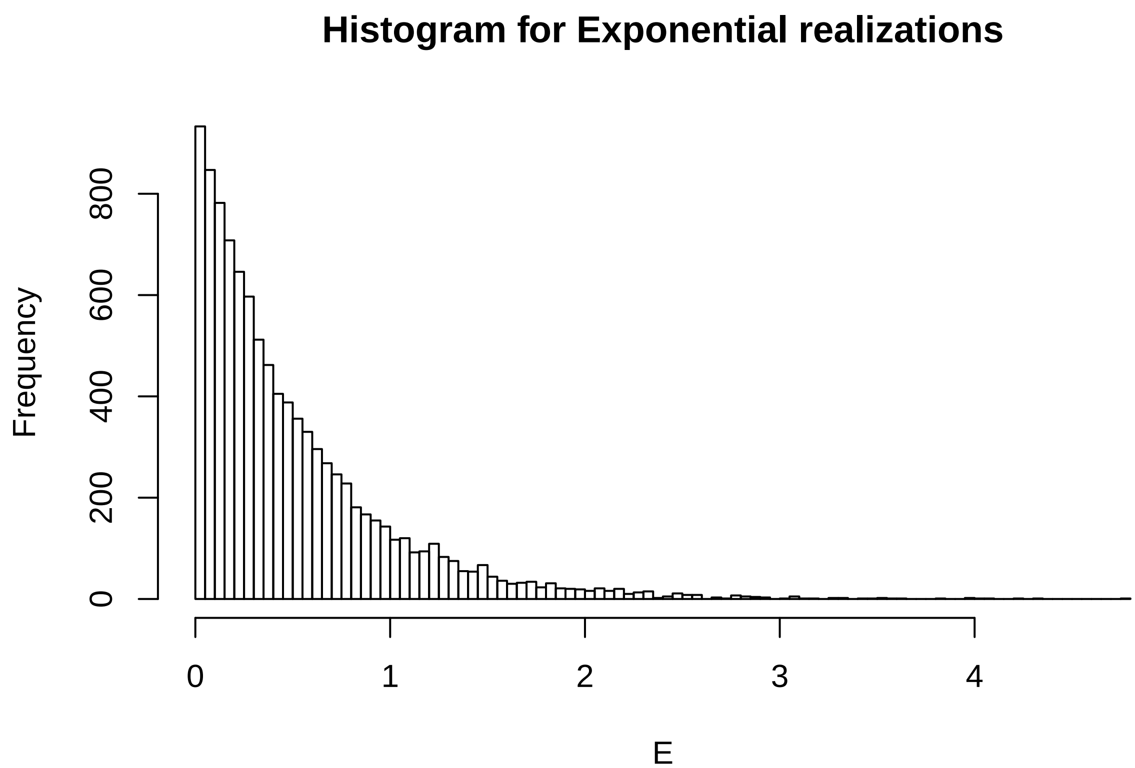
$$F_X(x) = 1 - \mathrm{e}^{-\lambda x}, \ x \geq 0.$$

The inverse cdf of an exponential distribution is given by

$$F_X^{-1}(p) = -\frac{1}{\lambda}\log(1-p),\ p \in (0,1).$$

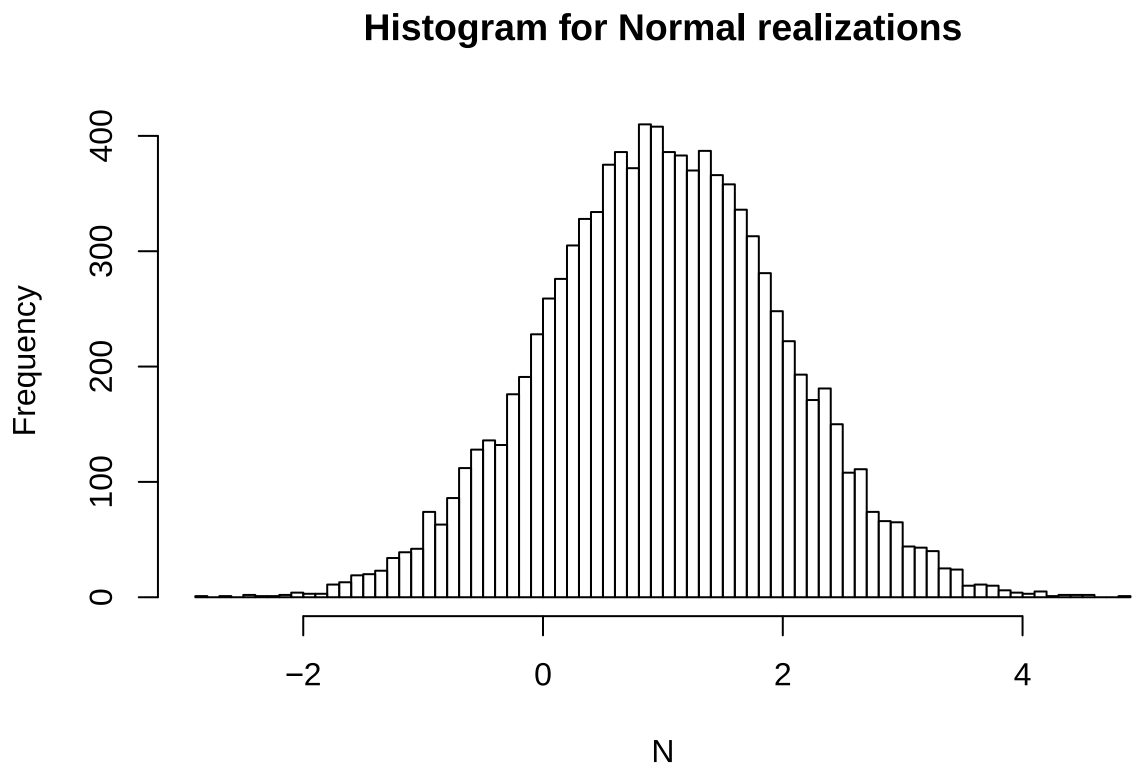
Then, if u is a realization of the Uniform random variable U, then $x = -\frac{1}{\lambda}\log(1-u)$ is a realization of an exponential distribution random variable with parameter λ. The R code below uses the quantile transform theorem to simulate exponential distributed random variables. Because we use the same random numbers as before, there is no simulation required in this step.

```
#Transform the uniform realizations to exponential realizations
lambda=2
E=-(1/lambda)*log(1-U)
hist(E,100, main="Histogram for Exponential realizations")
```



Above, we used the uniform realizations together with the quantile transform theorem to get exponential realizations. In the following chunk of R code, we use the same uniform realizations, but now they are transformed to normal realizations by using the inverse cdf of a normal distribution with mean μ and variance σ^2.

```
#Transform the (same) uniform realizations to normal realizations
mu=1
sigma=1
N=qnorm(U,mu,sigma)
hist(N,100, main="Histogram for Normal realizations")
```



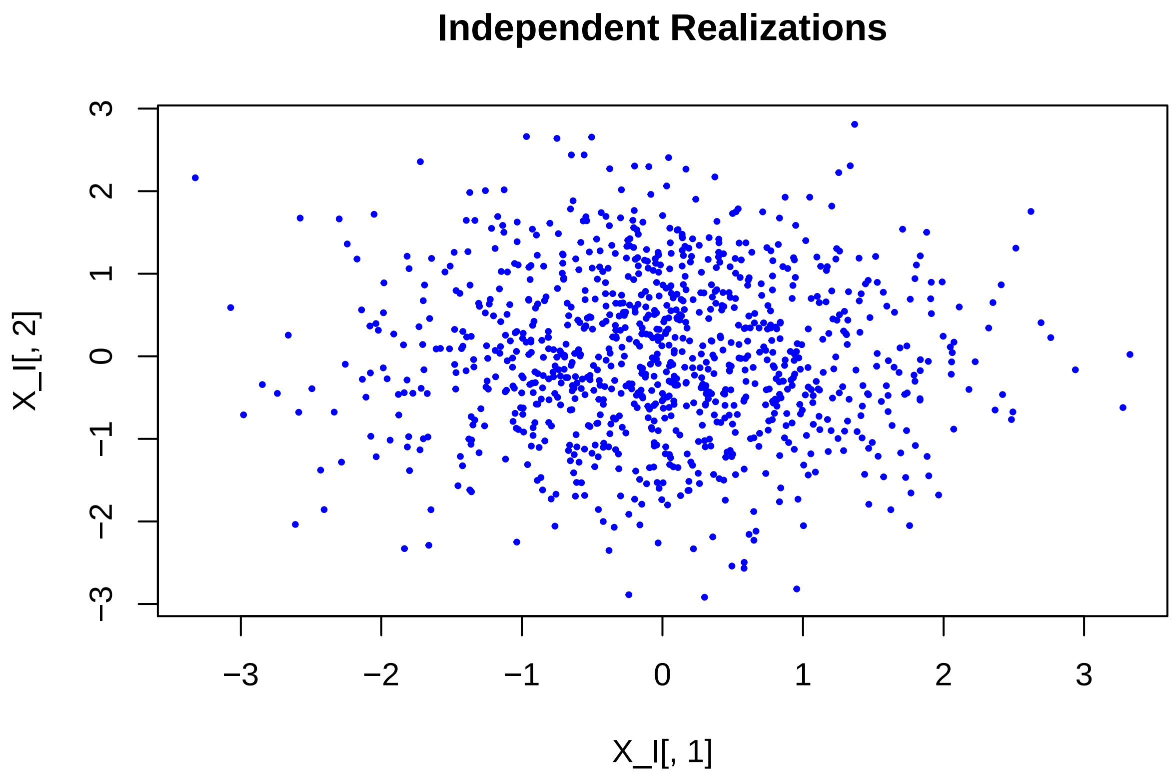
Simulation of comonotonic random vectors

We start by showing how to simulate independent normal realizations. We assume that X_1 and X_2 are normally distributed:

$$X_i \stackrel{d}{=} N\left(\mu_i, \sigma_i\right), \text{for } i = 1, 2.$$

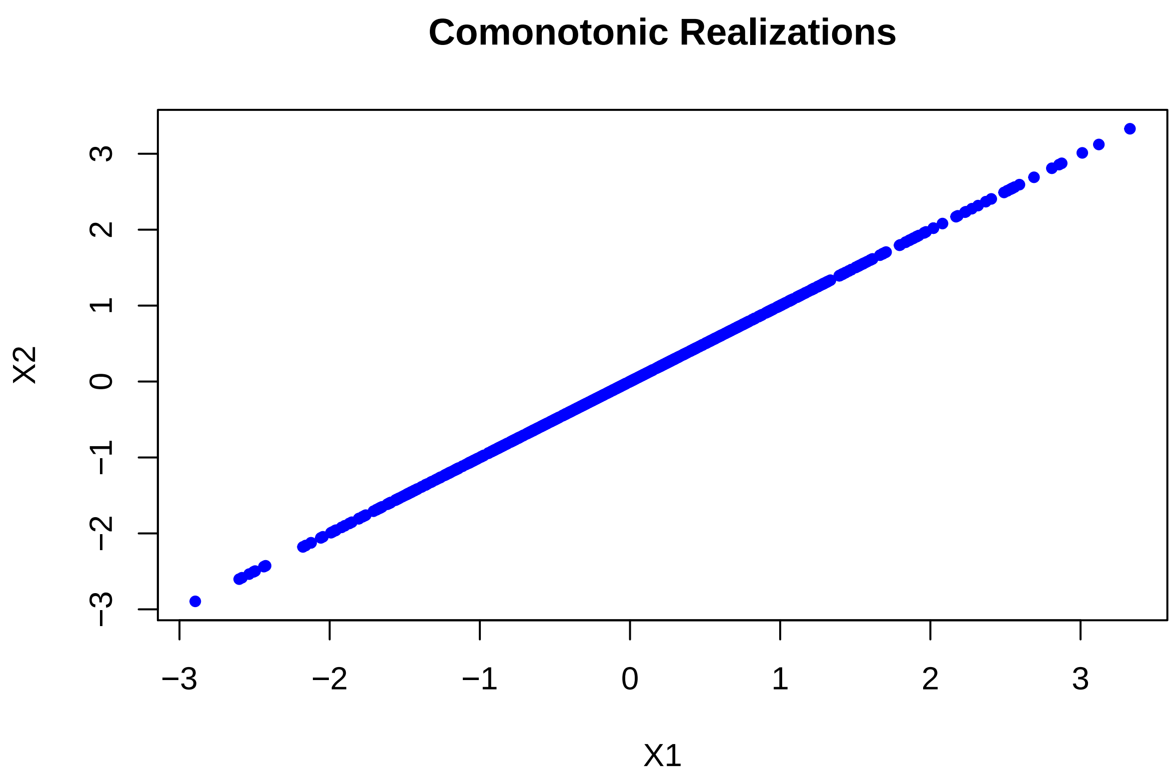
The independent modification of (X_1, X_2) is denoted by (X_1^I, X_2^I) and $X_i^I \stackrel{d}{=} X_i$, for $i = 1, 2$. Realizations from (X_1^I, X_2^I) can be obtained by applying the quantile transform theorem twice, to get independently realizations from X_1^I and then from X_2^I. The code is illustrated below.

```
n=2
NSim=1000
mu=c(0,0)
sigma=c(1,1)
U1=runif(NSim)
U2=runif(NSim)
X_I=matrix(1,NSim,2)
X_I[,1]=mu+sigma*qnorm(U1)
X_I[,2]=mu+sigma*qnorm(U2)
plot(X_I[,1],X_I[,2],pch=20, col="blue",cex=0.5, main="Independent Realizations")
```



The comonotonic modification of (X_1, X_2) is denoted by (X_1^c, X_2^c). This comonotonic modification has the same marginals as (X_1, X_2) but a comonotonic dependence structure. Then, simulating from the comonotonic modification can be done by using the quantile tranform theorem. However, when applying the quantile transform theorem to generate realizations for X_1^c and X_2^c, we have to use the *same uniform numbers*.

```
# Comonotonic random vector
U=runif(NSim)
X=rep(1,1,NSim)%o%mu+qnorm(U)%o%sigma
plot(X[,1],X[,2],pch=20, col="blue", main="Comonotonic Realizations", xlab="X1", ylab="X2")
```



Assume the random vector (X_1, X_2) has marginals given by:

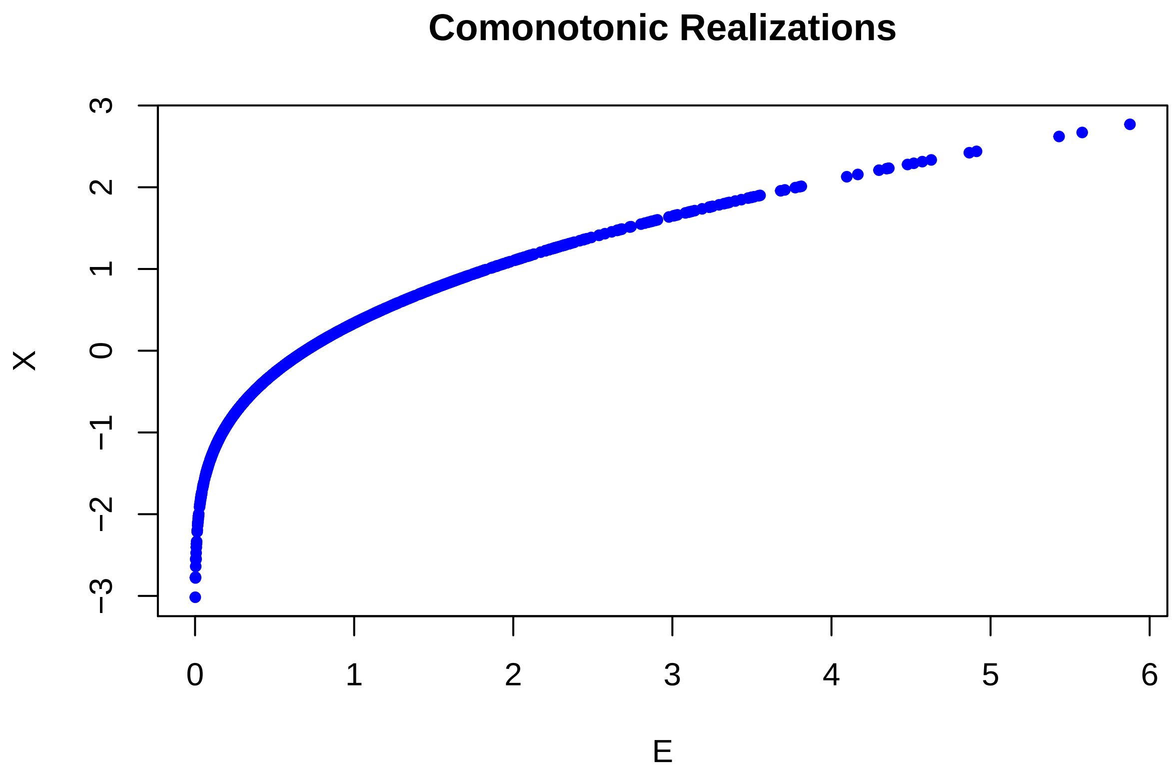
$$X_1 \stackrel{d}{=} \text{Exp}(\lambda),$$

and

$$X_2 \stackrel{d}{=} N(0, 1).$$

Then we can then simulate from the comonotonic modification by changing the inverse cdfs in the R Code.

```
# Exponential/normal marginals and comonotonic dependence
lambda=1
U=runif(NSim)
E=-(1/lambda)*log(1-U) # Inverse cdf of an exponential distribution applied in U
X=qnorm(U) # Inverse cdf of a standard normal distribution applied in U
plot(E,X,pch=20, col="blue", main="Comonotonic Realizations")
```


Comonotonic Realizations
X
E
3
2
1
0
−1
−2
−3
0
1
2
3
4
5
6