Generating continuous random variables

Math 276 *Actuarial Models*

Spring 2008 semester

EA Valdez
University of Connecticut - Storrs
Lecture Weeks 4-5

The inverse transform algorithm
- Some examples
- Simulating exponentials in R
- Gamma distribution
- Simulating gamma in R
- Poisson distribution

The rejection method
- Simulating half-normal in R

Generating Normal random variables
- Box-Muller transformations
- - continued
- Illustrating Box-Muller in R
- The polar method
- Illustrating polar method in R

Generating a Poisson process
- Introduction
- Generating the first T time units
- Simulating a Poisson process in R
- Alternative approach
- Nonhomogeneous Poisson process

The inverse transform algorithm

- Proposition: Let $U \sim U(0,1)$. For any continuous CDF F, the random variable X defined by

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

has distribution function F.

- Proof: in class.
- So long as we can derive the explicit form for F^{-1}, we can use this result to generate from a continuous distribution with CDF F:

Step 1: generate a random number U.
Step 2: set $X = F^{-1}(U)$ and you are done.

The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Some example

1. Let X come from a distribution with CDF
$$F(x) = x^n, \quad 0 < x < 1.$$
We can generate from this distribution by setting $X = U^{1/n}$ after generating U.

2. Exponential Suppose X comes from an Exp(λ) with PDF
$$f(x) = \lambda e^{-\lambda x}, \quad x > 0.$$
We can generate from this distribution by setting $X = -\frac{1}{\lambda} \log U$ where U is the generated random number.

3. Weibull A random variable X is said to have a Weibull(α, β) distribution if its CDF has the form
$$F(x) = 1 - \exp(-\alpha x^{\beta}), \quad x > 0.$$
Describe a method to generate from this distribution.

The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution

The rejection method
Simulating half-normal in R

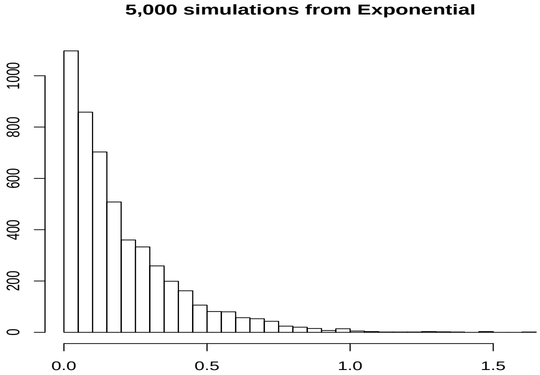
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R

Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Simulating exponentials in R

The following executes commands in R to simulate from exponential with $\lambda = 5$.

```
> urandom <- runif(5000)
> lambda <- 5
> x.exp <- -log(urandom)/lambda
> x.exp[1:20]
 [1] 0.44388841 0.03379772 0.09910273 0.15201846 0.03336014 0.01021211
 [7] 0.22211658 0.14559882 0.49368172 0.18587106 0.01590227 0.07396419
[13] 0.19668067 0.38644089 0.05817653 0.16742193 0.10366767 0.06664743
[19] 0.21755809 0.29353129
> summary(x.exp)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
3.168e-05 5.675e-02 1.321e-01 1.975e-01 2.692e-01 1.479e+00
> hist(x.exp,br=25,xlab="",ylab="",main="5,000 simulations from Exponential")
```



The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution

The rejection method
Simulating half-normal in R

Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R

Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Generating a Gamma random variable

- For a Gamma(n, λ) random variable, note that because we cannot write an explicit form for the expression of the F^{-1}, it is diffcult to directly apply inverse transform method.
- However, recall that the sum of independent Exponentials leads us to a Gamma distribution.
- We can generate from a Gamma distribution by generating n random numbers $U_1, U_2, \ldots, U_n$ and then setting

$$\begin{aligned} X &= -\frac{1}{\lambda}\log U_1 - \cdots - \frac{1}{\lambda}\log U_n \\ &= -\frac{1}{\lambda}\log(U_1 \cdots U_n). \end{aligned}$$

- This works provided n is a positive integer.

The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution

The rejection method
Simulating half-normal in R

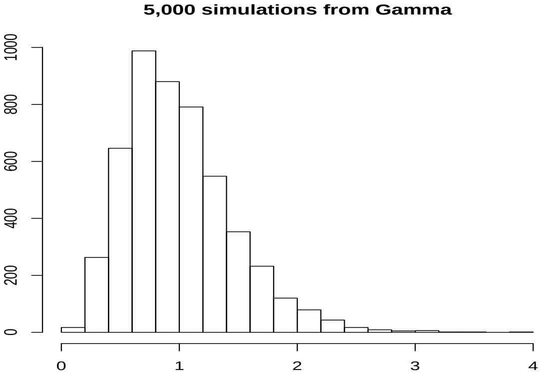
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R

Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Simulating gamma in R

The following executes commands in R to simulate from gamma with $n = 5$ and $\lambda = 5$.

```
> urandom1 <- runif(5000)
> urandom2 <- runif(5000)
> urandom3 <- runif(5000)
> urandom4 <- runif(5000)
> urandom5 <- runif(5000)
> u.prod <- urandom1*urandom2*urandom3*urandom4*urandom5
> lambda <-5
> x.gamma <- -log(u.prod)/lambda
> summary(x.gamma)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.1140  0.6689  0.9277  0.9998  1.2490  3.9810
> hist(x.gamma,br=25,xlab="",ylab="",main="5,000 simulations from Gamma")
```



Generating continuous random variables
EA Valdez
The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Generating a Poisson random variable

- Exploit the property of a Poisson process.
- Recall that for a Poisson process with rate λ, the times between successive events are independent exponentials with rate λ, and the number of events by time 1, $N(1)$, is Poisson with mean λ.
- Thus, to generate from a Poisson with mean λ, we can
  - generate n successive interarrival times which are exponentials, $X_1, X_2, \ldots, X_n$.
  - determine the number of events by time 1 using

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

The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

- continued

- Equivalently, we generate $U_1, \ldots, U_n, \ldots$ and set

$$\begin{aligned} N &= \max\left(n : \sum_{i=1}^{n} -\frac{1}{\lambda} \log U_i \leq 1\right) \\ &= \max\left(n : \sum_{i=1}^{n} \frac{1}{\lambda} \log U_i \geq -\lambda\right) \\ &= \max(n : \log(U_1 \cdots U_n) \geq -\lambda) \\ &= \max(n : U_1 \cdots U_n \geq e^{-\lambda}) \end{aligned}$$

- This is also equivalent to setting

$$N = \min(n : U_1 \cdots U_n < e^{-\lambda}) - 1.$$

The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

The rejection method

- Suppose we wish to generate X from a distribution with PDF $f(x)$.
- Assume that we are able to generate Y from a distribution with PDF $g(y)$ and that there is a constant c such that

$$\frac{f(y)}{g(y)} \leq c, \quad \text{for all } y.$$

- According to the rejection method, we can generate X using the following steps:

Step 1: generate Y from distribution with density g.

Step 2: generate a random number U.

Step 3: if $U \leq \frac{f(Y)}{cg(Y)}$, set $X = Y$.

Step 4: else, return to step 1.

The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

- continued

Theorem:

1. The random variable X generated by the rejection method has density f.
2. The number of iterations required for the rejection algorithm has a geometric distribution with mean c.

Proof: to be discussed in class.

The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Some examples

- Example 5d: Use the rejection method to generate from

$$f(x) = 20x(1-x)^3, \;\; 0 < x < 1.$$

- Example 5e: Use the rejection method to generate from a Gamma$(3/2, 1)$ density with

$$f(x) = Kx^{1/2}e^{-x}, \;\; x > 0,$$

  where $K = 1/\Gamma(3/2) = 2/\sqrt{\pi}$.
- Example 5f: Suggest a rejection method for generating from a standard Normal random variable $Z \sim \mathrm{N}(0,1)$.
- Example 5g: Suggest a rejection method for generating from a truncated Gamma$(2,1)$, conditional on its value exceeding 5.
- These examples to be discussed in details in class.

The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Simulating half-normal with the rejection method in R

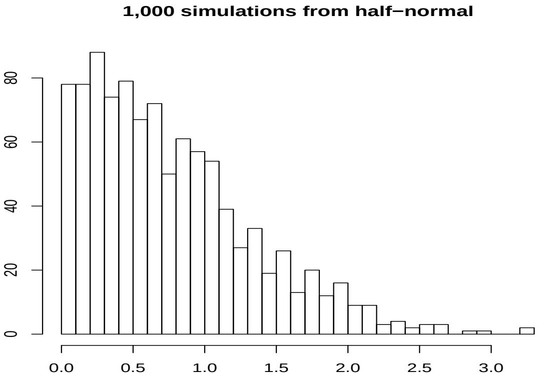
- The following is a routine in R to simulate from the absolute value of a standard Normal random variable, using the rejection method.
- Function is called `simhalfnorm.R`.

```
# function to generate from a half-Normal distribution
# i.e. absolute value of a standard Normal

simhalfnorm <- function(n.gen){
sim.vector <- rep(0,n.gen)
# to start, generate 2 independent exponentials with rate 1
for(i in 1:n.gen){
urandom <- runif(2) ; y <- -log(urandom)
while(y[2] < (y[1]-1)^2/2){
urandom <- runif(2) ; y <- -log(urandom)
}
sim.vector[i] <- y[1]
}
# output
sim.vector
}
```


The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Executing the simulation

```
> source("C:\\...\\Math276-Spring2008\\Rcodes-2008\\Week4\\simhalfnorm.R")
> out1 <- simhalfnorm(1000)
> out1[1:20]
 [1] 0.30606099 1.03767270 0.95139082 1.89421610 0.41375272 1.09521146
 [7] 0.98360575 0.60245821 0.47552683 0.60963746 0.84500891 1.17122770
[13] 1.39765356 0.80675612 1.00840132 1.17619377 0.40388706 0.33065198
[19] 0.01615143 0.94438367
> summary(out1)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
0.0008065 0.3058000 0.6472000 0.7694000 1.0780000 3.2830000
> sd(out1)
[1] 0.5799377
> hist(out1,br=25,xlab="",ylab="",main="1,000 simulations from half-normal")
```



The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

The Box-Muller transformations

- Suppose X and Y are independent standard Normal random variables with R and θ their corresponding polar coordinates:

$$\begin{aligned} R^2 &= X^2 + Y^2 \\ \tan\theta &= \frac{Y}{X} \end{aligned}$$

- It can be shown that R^2 and Θ has joint density

$$f(d,\theta) = \frac{1}{2}\frac{1}{2\pi}, 0 < d < \infty, 0 < \theta < 2\pi.$$

- R^2 and Θ are independent with the following marginals:
  - R^2 is exponential with mean 2, and
  - Θ is Uniform over $(0, 2\pi)$.

The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

- continued

- The idea then is to first generate their polar coordinates and then transform back to rectangular coordinates.
- This can be done with the following steps:

Step 1: generate two random numbers U_1 and U_2.
Step 2: compute $R^2 = -2 \log U_1$ and $\theta = 2\pi U_2$.
Step 3: then set

$$\begin{aligned} X &= R \cos \Theta = \sqrt{-2 \log U_1} \cos(2\pi U_2) \\ Y &= R \sin \Theta = \sqrt{-2 \log U_1} \sin(2\pi U_2) \end{aligned}$$

- These are known as Box-Muller transformations.

The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Simulating standard normal using Box-Muller in R

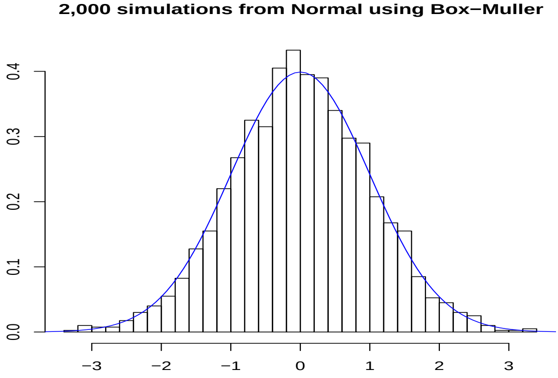
- The following is a routine in R to simulate from standard normal using the Box-Muller transformations.
- Function is called `simnormbm.R`.

```
# function to generate standard Normal using the Box-Muller transformations

simnormbm <- function(n.gen){
urandom1 <- runif(n.gen)
urandom2 <- runif(n.gen)
Rsq <- -2*log(urandom1)
theta <- 2*pi*urandom2
x <- sqrt(Rsq)*cos(theta)
y <- sqrt(Rsq)*sin(theta)
# combine independent x and y
z  <- (x+y)/sqrt(2)
# output
z
}
```


The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Executing the simulation

```
> source("C:\\...\\Math276-Spring2008\\Rcodes-2008\\Week4\\simnormbm.R")
> out1 <- simnormbm(2000)
> out1[1:20]
 [1] -0.56867837  2.20613520 -0.46835432 -0.55154500 -2.17445397  2.33841314
 [7]  1.12578412  0.36733281 -1.24346577 -0.41904992 -0.17311589 -0.95007900
[13] -0.04571634  2.75877598 -0.86521717  0.88587815 -0.92575967 -1.70796891
[19] -0.50454892 -1.11563401
> summary(out1)
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-3.298e+00 -6.694e-01  6.473e-05  1.055e-02  6.969e-01  3.246e+00
> sd(out1)
[1] 0.9935784
> hist(out1,br=25,xlab="",ylab="",main="2,000 simulations from Normal using Box-Muller",freq=FALSE)
> curve(dnorm(x),from=-4,to=4,col="blue",add=TRUE)
```



The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution

The rejection method
Simulating half-normal in R

Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R

Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

The polar method for generating Normal random variables

- The Box-Muller transformation is believed to be computationally inefficient because of the computation of the sine and cosine functions.
- The Polar method avoids these computations by generating random numbers U_1 and U_2 and setting

$$V_1 = 2U_1 - 1 \;\text{ and }\; V_2 = 2U_2 - 1.$$

- Compute $S = R^2 = V_1^2 + V_2^2$ and then set

$$\begin{aligned} X &= V_1 \sqrt{\frac{-2 \log S}{S}} \\ Y &= V_2 \sqrt{\frac{-2 \log S}{S}} \end{aligned}$$

- Details in class. Algorithm is given on pages 81-82.

The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Simulating standard normal using the polar method in R

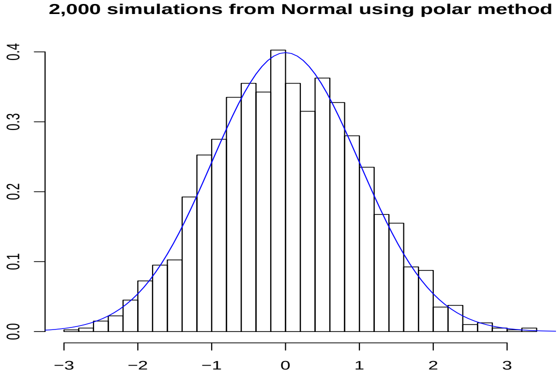
- The following is a routine in R to simulate from standard normal using the polar method.
- Function is called `simnormpolar.R`.

```
# function to generate standard normal using the Polar method

simnormpolar <- function(n.gen){
sim.vector <- rep(0,n.gen)
for (i in 1:n.gen){
urandom <- runif(2)
v1 <- 2*urandom[1]-1; v2 <- 2*urandom[2]-1
s <- v1^2 + v2^2
while(s > 1){
urandom <- runif(2)
  v1 <- 2*urandom[1]-1; v2 <- 2*urandom[2]-1
s <- v1^2 + v2^2
}
insq <- -2*log(s)/s
x <- sqrt(insq)*v1
y <- sqrt(insq)*v2
sim.vector[i] <- (x+y)/sqrt(2)
}
# output
sim.vector
}
```


The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Executing the simulation

```
> source("C:\\...\\Math276-Spring2008\\Rcodes-2008\\Week4\\simnormpolar.R")
> out1 <- simnormpolar(2000)
> out1[1:20]
 [1] -0.53573046  0.58419468 -1.56107977  0.27961148 -1.37491965 -1.33070593
 [7]  0.09831912 -0.88206048  1.17231397  0.44666004  0.90755589 -1.38911674
[13]  1.24109777  0.34494397  0.46953717  1.03635509 -0.53338976  1.18540414
[19]  0.29707077 -0.71940887
> summary(out1)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-2.926000 -0.704400 -0.008467  0.017280  0.720300  3.390000
> sd(out1)
[1] 1.000975
> hist(out1,br=25,xlab="",ylab="",main="2,000 simulations from Normal using polar method",freq=FALSE)
> curve(dnorm(x),from=-4,to=4,col="blue",add=TRUE)
```



The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Generating a Poisson process

- To generate a Poisson process with rate λ, use the fact that the times between successive events are independent exponentials each with rate λ.
- Generate n random numbers $U_1, \ldots, U_n$ and set $X_i = -\frac{1}{\lambda} \log U_i$, the time between the $(i-1)$st and the ith event.
- The sum $\sum_{i=1}^{j} X_i$, for $j = 1, \ldots, n$ then gives the actual time of the jth event.
- To generate then the first T time units of the process, follow the previous procedure and stopping when the sum then exceeds T.

The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Generating the first T time units

- In the following algorithm,

  t refers to time,

  I is the number of events occurring by time t, and

  $S(I)$ is the most recent event time.

- Generating the first T time units of a Poisson process with rate λ:

Step 1: start with $t = 0$, $I = 0$.
Step 2: generate a random number U.
Step 3: set $t = t - \frac{1}{\lambda} \log U$ and STOP if $t > T$.
Step 4: reset $I = I + 1$, $S(I) = t$.
Step 5: return to Step 2.

- The values $S(1), S(2), \ldots, S(I)$ are the I event times in increasing order.

The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Simulating the arrival times of a Poisson process in R

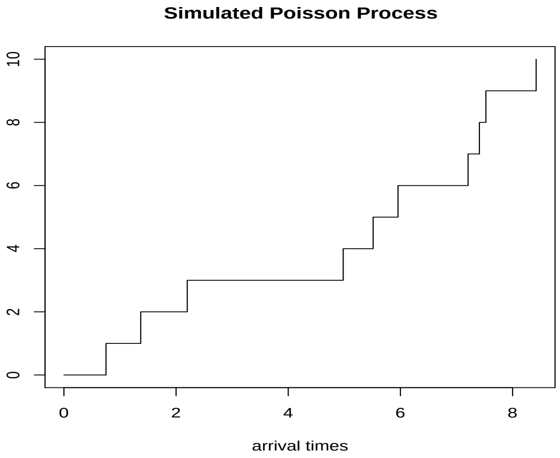
- The following is a routine in R to simulate the arrival times of a Poisson process.
- Function is called `simpproc.R`.

```
# simulating the arrival times of a Poisson process
# inputs are: number of simulations and lambda parameter

simpproc <- function(n.gen,lambda){
x <- rep(0,n.gen)
# generate exponential inter-arrival times
for(i in 1:n.gen){
urandom <- runif(1)
x[i] <- -log(urandom)/lambda
}
# computing the time of the n-th arrival
arrival.times <- c(0, cumsum(x))
# plotting the process
nn <- c(0:n.gen)
plot(arrival.times,nn,type="s",ylab="",xlab="arrival times", main="Simulated Poisson Process")
}
```


The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Executing the simulation

```
> source("C:\\...\\Math276-Spring2008\\Rcodes-2008\\Week4\\simpproc.R")

> simpproc(10,1.5)
```



The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution

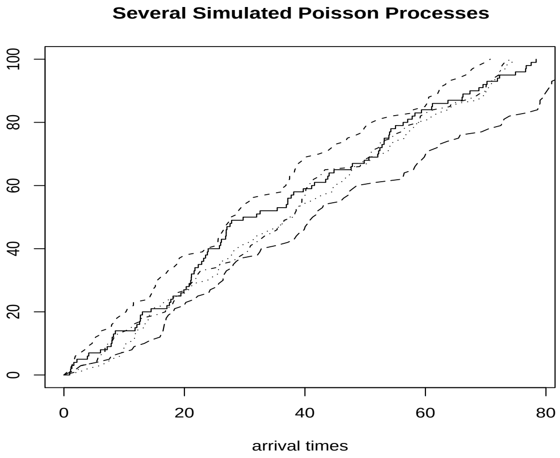
The rejection method
Simulating half-normal in R

Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R

Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Several simulations

```
> source("C:\\...\\Math276-Spring2008\\Rcodes-2008\\Week4\\simpprocs.R")

> simpprocs(100,1.2)
```



The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Alternative approach

- Recall that for a Poisson process with rate λ, the total number of events that occur by time t, $N(T)$, is Poisson with mean λT.
- The following general steps can then be followed:
  1. generate $N(T)$ from a Poisson with mean λT.
  2. generate $N(T)$ random numbers $U_1, \ldots, U_{N(T)}$.
  3. $TU_1, TU_2, \ldots, TU_{N(T)}$ are taken as the event times by time T of the Poisson process.
- This works more efficiently than simulating from exponentials, provided all we are interested in were on the set of event times of the process.
- Often such is not the case however, we would also like the event times in increasing order.

The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Generating a nonhomogeneous Poisson process

- In the nonhomogeneous Poisson process, the rate depends on time t, where $\lambda(t)$ is called the intensity function.
- One approach is called the thinning approach:
  - start by choosing a λ such that
  $$\lambda(t) \leq \lambda, \text{ for all } t \leq T.$$
  - randomly select the event times of a Poisson process with rate λ.
- If an event of a Poisson process with rate λ is counted with probability $\lambda(t)/\lambda$, then the process of counted events is the desired nonhomogeneous Poisson process.

The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Algorithm with the thinning approach

- Applying the thinning approach, the following algorithm provides generating the first T time units of a nonhomogeneous Poisson process with intensity $\lambda(t)$:

Step 1: set $t = 0$, $I = 0$.
Step 2: generate a random number U.
Step 3: set $t = t - \frac{1}{\lambda} \log U$ and if $t > T$, STOP.
Step 4: generate another random number U.
Step 5: if $U \leq \lambda(t)/\lambda$, set $I = I + 1$, $S(I) = t$.
Step 6: return to Step 2.

- The final value of I is the number of events by time T with the event times $S(1), \ldots, S(I)$.

The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Improvement to the thinning approach

- An improvement to the approach, though less efficient, is to break up the whole interval into subintervals and then use the thinning procedure over each subinterval.
- In effect, determine suitable values k, $0 = t_0 < t_1 < t_2 < \cdots < t_k < t_{k+1} = T$, $\lambda_1, \ldots, \lambda_{k+1}$ such that

$$\lambda(s) \leq \lambda_i, \ \text{ if } t_{i-1} \leq s < t_i, \ \ i = 1, \ldots, k+1.$$

- Over each time interval (t_{i-1}, t_i), generate exponentials with rate λ_i and accept the generated event with probability $\lambda(s)/\lambda_i$.
- This works because of the memoryless property.
- See page 85 for the algorithmic steps using subintervals.

The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Alternative approach to generating nonhomogeneous Poisson process

- An alternative approach is to generate event times $S_1, S_2, \ldots$ directly.
- This may be accomplished by generating first S_1 from distribution function F_0; then generate S_2 by adding S_1 to a generated value from F_{s_1}; then generate S_3 by adding S_2 to a generated value from F_{s_2}, and so on.
- This alternative procedure works in cases where there are known efficient procedures for simulating from the F's, e.g. inverse transform.
- Suppose an event occurs at time s, additional time until the next event has distribution function:

$$
\begin{aligned}
F_s(x) &= P(\text{time from } s \text{ until next event} \leq x | \text{ event at s}) \\
&= 1 - \exp\left(-\int_0^x \lambda(s+y)dy\right).
\end{aligned}
$$

The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution
The rejection method
Simulating half-normal in R
Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R
Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process

Example 5h

- Consider the example in Example 5h, pages 86-87.
- Suggest simulation procedure for a nonhomogeneous Poisson process with intensity function given by

$$\lambda(t) = \frac{1}{t+a}, \quad \text{for } t \geq 0.$$

- Details in class.

The inverse transform algorithm
Some examples
Simulating exponentials in R
Gamma distribution
Simulating gamma in R
Poisson distribution

The rejection method
Simulating half-normal in R

Generating Normal random variables
Box-Muller transformations
- continued
Illustrating Box-Muller in R
The polar method
Illustrating polar method in R

Generating a Poisson process
Introduction
Generating the first T time units
Simulating a Poisson process in R
Alternative approach
Nonhomogeneous Poisson process