

# Computer Lab 3

Martynas Lukosevicius

17/11/2020

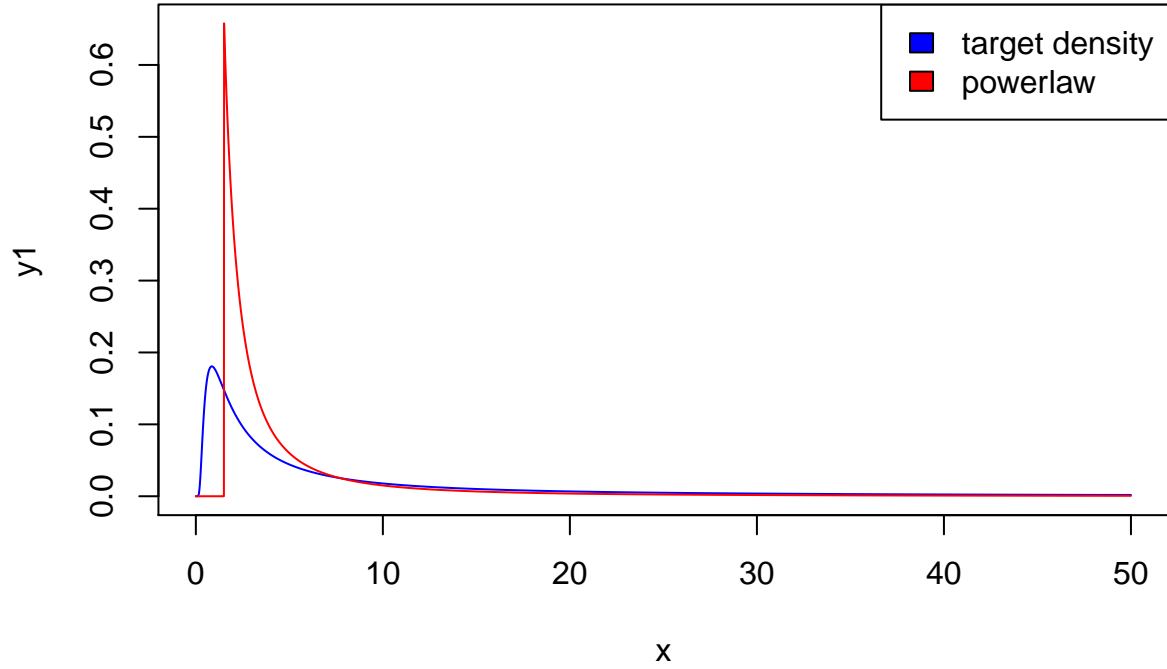
## Question 1

```
distrib1 <- function(x,c){
  if(x>0){
    return(c*(sqrt(2*pi)^(-1))*(exp(-(c^2)/(2*x))) * (x^(-3/2)))
  }
  else return(0)
}

powerlaw <- function(x,a,t){
  if(x>t){
    return(((a-1)/t) * ((x/t)^(-a)))
  }
  else return(0)
}

x <- seq(0,50, by=0.01)
c <- 1.6
#n <- length(x)
#a= 1 + n*(sum(log(x/t)))^(-1)
y1 <- sapply(x, distrib1, c=c)
y2 <- sapply(x, powerlaw, a = 2,t = 1.5)
ymax <- max(y1,y2)
b <- min(c(y1,y2))
e <- max(c(y1,y2))
ax <- seq(b,e,by=(e-b)/200)

plot(x,y1, type="l", col = "blue", ylim = c(0,ymax))
lines(x,y2, col = "red")
legend("topright", c("target density", "powerlaw"), fill=c("blue", "red"))
```



```
# hist(y1, breaks = ax,
#      col = "red",
#      main = "Comparison of rnorm() with our rNorm()",
#      xlab = "values",
#      xlim = range(y1,y2))
#
# hist(y2, breaks = ax, col = "blue", xlim = range(y1,y2), add = TRUE)
```

Power-law distribution cannot be used just by itself because it doesn't support range from 0 to  $T_{min}$ . Because of this we need to use mixture distribution. To support  $x$  from 0 to  $T_{min}$  we choose uniform distribution  $Unif(0, T_{min})$ . As we can see power-law distribution is monotonically decreasing, so we  $T_{min}$  should be equal to  $x$  where target density has maximum value.

Lets find maximum of target density function:

$$\frac{\partial}{\partial x} \frac{ce^{-\frac{c^2}{2x}} x^{-\frac{3}{2}}}{\sqrt{2\pi}} = \frac{ce^{-\frac{c^2}{2x}} (c^2 - 3x)}{2\sqrt{2\pi} x^{7/2}}$$

$$\frac{c \frac{e^{-\frac{c^2}{2x}}}{2x} (c^2 - 3x)}{2\sqrt{2\pi} x^{7/2}} = 0$$

$$x = \frac{c^2}{3}$$

$$T_{min} = \frac{c^2}{3}$$

To make a mixture model we need to know the probability of taking uniform distribution and powerlaw distribution probability that number will be in 0-  $T_{min}$  region is:

$$\int_0^{T_{min}} c x^{-\frac{3}{2}} e^{-\frac{c^2}{2x}} \sqrt{2\pi}^{-1} dx = \frac{\Gamma\left(\frac{1}{2}, \frac{c^2}{2T_{min}}\right)}{\sqrt{\pi}}$$

$$\text{as } T_{min} = \frac{c^2}{3} - \frac{\Gamma\left(\frac{1}{2}, \frac{c^2}{2T_{min}}\right)}{\sqrt{\pi}} = \frac{\Gamma\left(\frac{1}{2}, \frac{3}{2}\right)}{\sqrt{\pi}} \sim 0.08326451666$$

As the result: majorising density function is:

$$g(x) = \frac{2 * 0.08326}{c^2} 1_{[0, T_{min}]} + (1 - 0.08326) * \frac{2^{1-a}(a-1) \left(\frac{x}{c^2}\right)^{-a}}{c^2} * 1_{(T_{min}, \infty)}$$

**2.**

Target density:

$$f(x) = c(\sqrt{2\pi})^{-1} e^{-\frac{c^2}{2x}} x^{-\frac{3}{2}} 1_{(0, \infty)}(x)$$

We need to find  $c_{maj}$

$$c_{maj} > 0; \sup_x (f(x)/g(x)) \leq c_{maj}$$

$$h(x) = \frac{f(x)}{g(x)}$$

$$c_{maj} = h(x_{maj})$$

$$\text{if } x < T_{min}$$

$$\frac{\partial}{\partial x} \frac{f(x)}{\frac{0.16652}{c^2}} = \frac{e^{-\frac{c^2}{2x}} (1.19788c^5 - 3.59364c^3x)}{x^{7/2}}$$

$$\frac{e^{-\frac{c^2}{2x}} (1.19788c^5 - 3.59364c^3x)}{x^{7/2}} = 0$$

$$x_{maj} = \frac{c^2}{3}$$

$$x \text{ for } c_{maj} = x = 0.333333c^2 = T_{min}$$

$$\text{if } x > T_{min}$$

$$\frac{\partial}{\partial x} \frac{f(x)}{g(x)} = \frac{3^{a-1} c^3 \frac{e^{-c^2}}{2x} \left(\frac{x}{c^2}\right)^a ((2a-3)x + c^2)}{2\sqrt{2\pi}(a-1)x^{7/2}}$$

$$\frac{3^{a-1} c^3 \frac{e^{-c^2}}{2x} \left(\frac{x}{c^2}\right)^a ((2a-3)x + c^2)}{2\sqrt{2\pi}(a-1)x^{7/2}} = 0$$

$$x_{maj} = \frac{c^2}{3-2a}$$

when  $1 < a < 1.5$

$$c_{maj} = h(x_{maj})$$

```

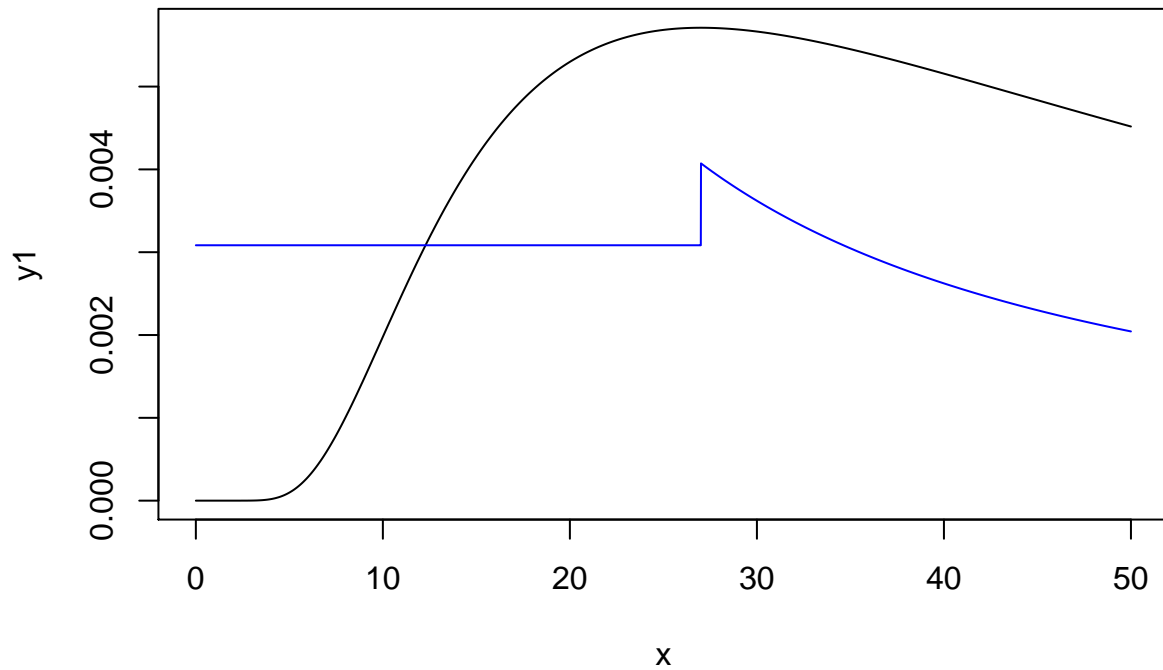
majDensity <- function(x, c,a){
  Tmin <- (c^2)/3

  if(x>Tmin){
    return((1-0.083264) * powerlaw(x, a, Tmin))
  }
  else {return(0.083264 * dunif(x,0,Tmin))}
}

c <- 9
y3 <- sapply(x, majDensity, c = c, a = 1.12)
y1 <- sapply(x, distrib1, c = c)
plot(x,y1, type = "l", ylim = c(0,max(y3,y1)))
#lines(x,dnorm(x,3,1.2), col = "red")

lines(x,y3, col = "blue")

```



```
#lines(x,y1, col = "red")
```

2

```
library(powerLaw)
```

```
## Warning: package 'powerLaw' was built under R version 4.0.3
```

```
randomnumber <- function(t,a){
  numb <- runif(1)
  if(numb<= 0.08326){
    return(runif(1,0,t))
  }
  else{
    return(rplcon(1,t,a))
  }
}

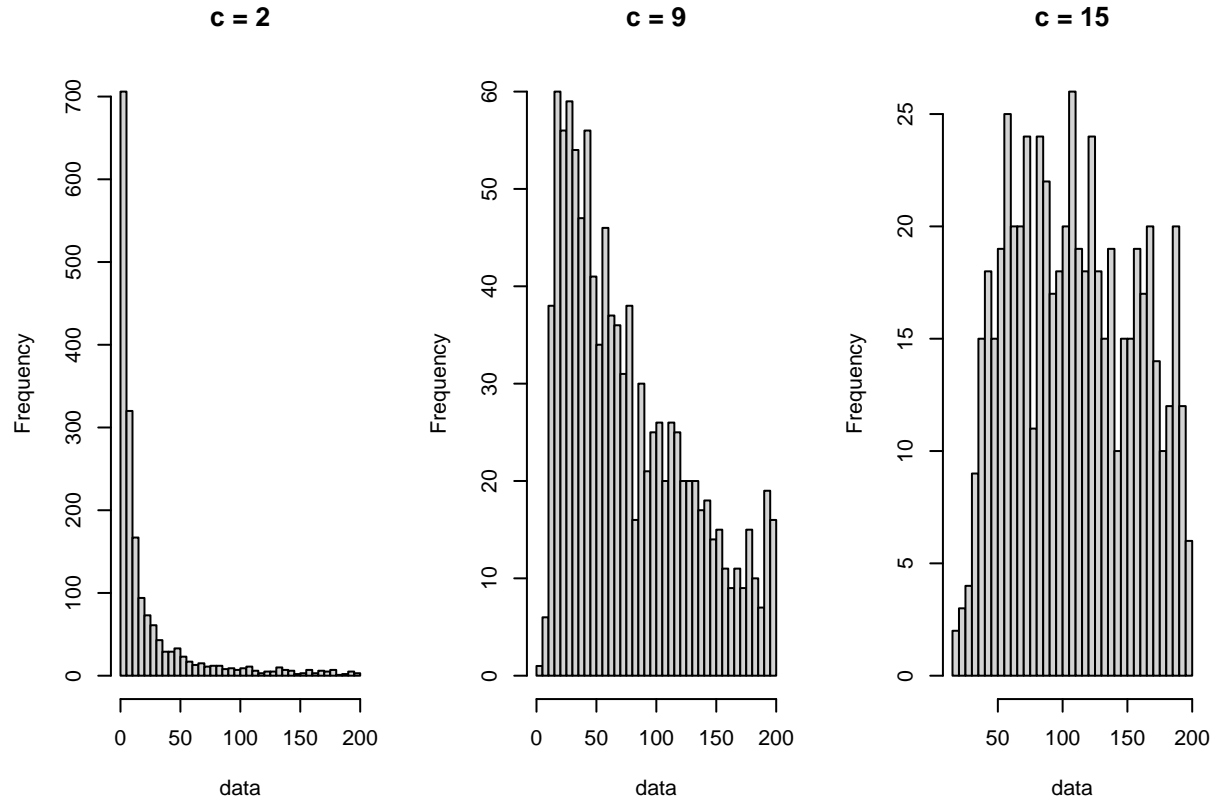
CompleteDist <- function(c, a, rej){
  z <- TRUE
  res <- 0

  Tmin <- (c^2)/3
  xmaj <- (c^2)/(3-2*a)
  cmaj <- distrib1(xmaj,c)/majDensity(xmaj,c, a)

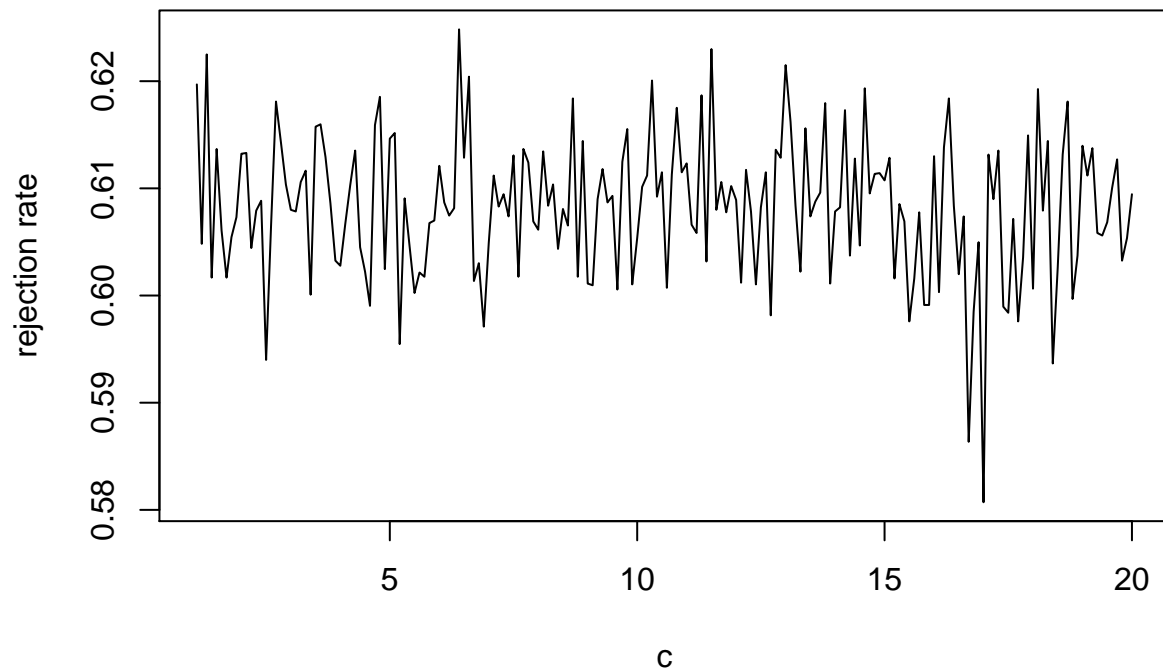
  while (z == TRUE) {
    y <- randomnumber(Tmin,a)
    u <- runif(1)
    if(u <= distrib1(y, c) / (cmaj*majDensity(y,c,a))){
      res <- y
      z <- FALSE
    }
    if(rej){
      rejected <- rejected + 1
    }
  }
  return(res)
}

rDist <- function(n,c,a ,rej = FALSE){
  return(replicate(n, CompleteDist(c, a, rej)))
}
```

3.



	c=2	c=9	c=15
mean	1.119115e+05	6.172831e+05	1.903368e+06
variance	2.337273e+13	5.602875e+14	5.427475e+15



## Question 2

1.

$$DE(\mu, \alpha) = \frac{\alpha}{2} e^{-\alpha|x-\mu|}$$

- $\mu$  - location parameter
- $b > 0$  - scale parameter

inverse CDF of DE:

Source - [https://en.wikipedia.org/wiki/Laplace\\_distribution](https://en.wikipedia.org/wiki/Laplace_distribution)

$$F^{-1}(p) = \mu - b \operatorname{sgn}(p - 0.5) \ln(1 - 2|p - 0.5|)$$

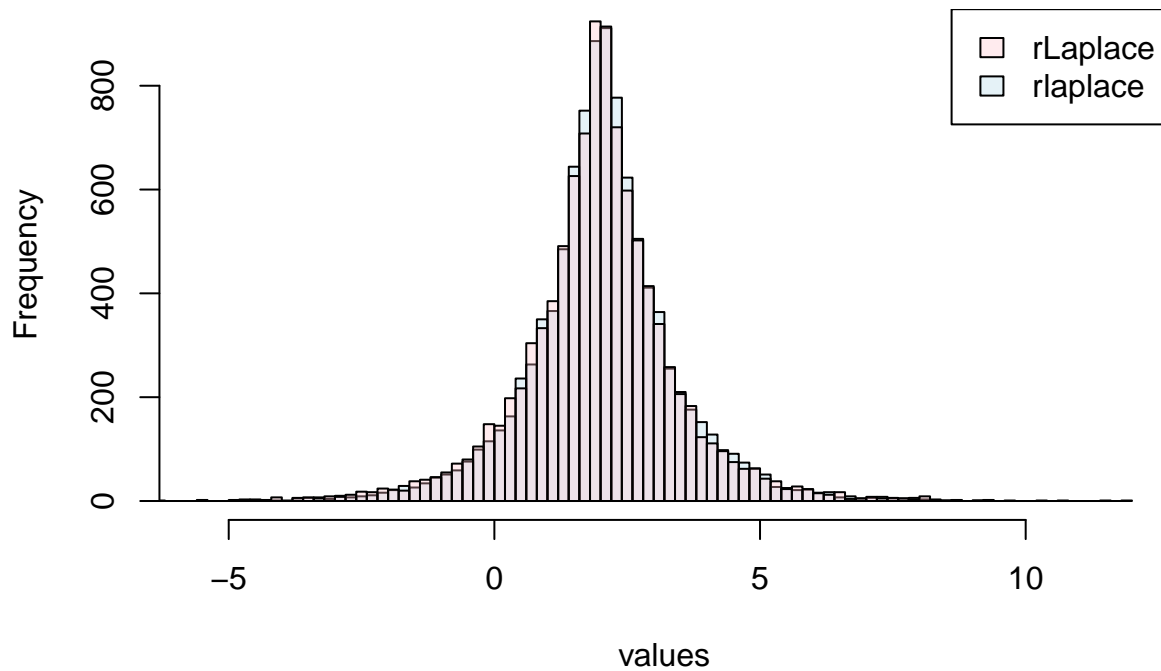
where  $b = \frac{1}{\alpha}$

```
rLaplace <- function(n, mean = 0, alpha = 1){
  b <- 1/alpha
  u <- runif(n)
  res <- mean - (b*sign(u-0.5) * log(1-(2*abs(u-0.5))))
  return(res)
}
```

meaning:

1. calculate  $b$ .
2. take  $n$  random variables from uniform distribution  $[0,1]$
3. calculate random numbers from inverse CDF of laplace distribution where  $x$  is a random variable from uniform distribution

## Comparison of rlaplace function from rmutil with our rLaplace



From histogram we can see that inverse CDF method approximated laplace distribution reasonably, mean and variance very similar.

2.

```
DE <- function(x, mean = 0, alpha = 1){
  return((0.5*alpha)*exp((-alpha)*abs(x-mean)))
}

genNorm <- function(c, rej){
  z <- TRUE
  res <- 0
  while (z == TRUE) {
    y <- rLaplace(1)
    u <- runif(1)
    if(u <= pnorm(y) / (c*DE(y))){
      res <- y
    }
  }
}
```



```

    z <- FALSE
  }
  if(rej){
    rejected <- rejected + 1
  }
}
return(res)
}

rNorm <- function(n,c,rej = FALSE){
  return(replicate(n, genNorm(c, rej)))
}

```

algorithm:

1. write Laplace probability function
2. assign 0 to result value res
3. generate random number y from rLaplace function
4. generate random number u from uniform distribution
5. check if u is less or equal to probability of y in normal distribution / c \* probability of y in laplace distribution
  - a) if yes, return y
  - b) repeat steps from 3

$$c > 0; \sup_x (f(x)/g(x)) \leq c$$

$$h(x) = \frac{f(x)}{g(x)}$$

$$f(x) = N(0,1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

$$g(x) = \frac{1}{2} e^{-|x|}$$

$$h(x) = \sqrt{\frac{2}{\pi}} e^{|x| - \frac{x^2}{2}}$$

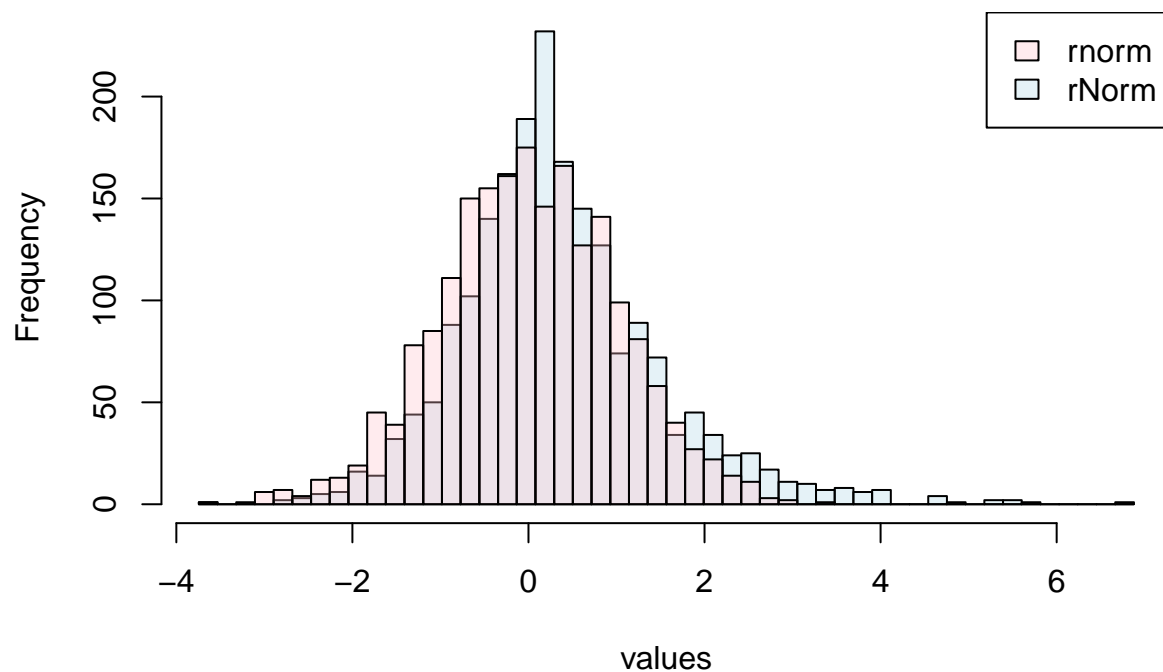
$$\frac{d}{dx} \sqrt{\frac{2}{\pi}} e^{|x| - \frac{x^2}{2}} = \sqrt{\frac{2}{\pi}} x e^{|x| - \frac{x^2}{2}} \left( \frac{1}{|x|} - 1 \right)$$

$$\frac{\sqrt{2} e^{x - \frac{x^2}{2}} (x - 1)}{\pi} = 0$$

$$x = \pm 1$$

$$c = h(1) = 1.3154892$$

## Comparison of `rnorm()` with our `rNorm()`



	mean	variance
<code>rNorm()</code>	0.3461223	1.283130
<code>rnorm()</code>	0.0146288	1.012949

rejection rate: 0.2119779, expected rejection rate =  $c = 0.2398265$ , difference = -0.0464892

We can see that mean and variance of generated distribution slightly differs.