Computer Lab 3

Martynas Lukosevicius

17/11/2020

Question 1

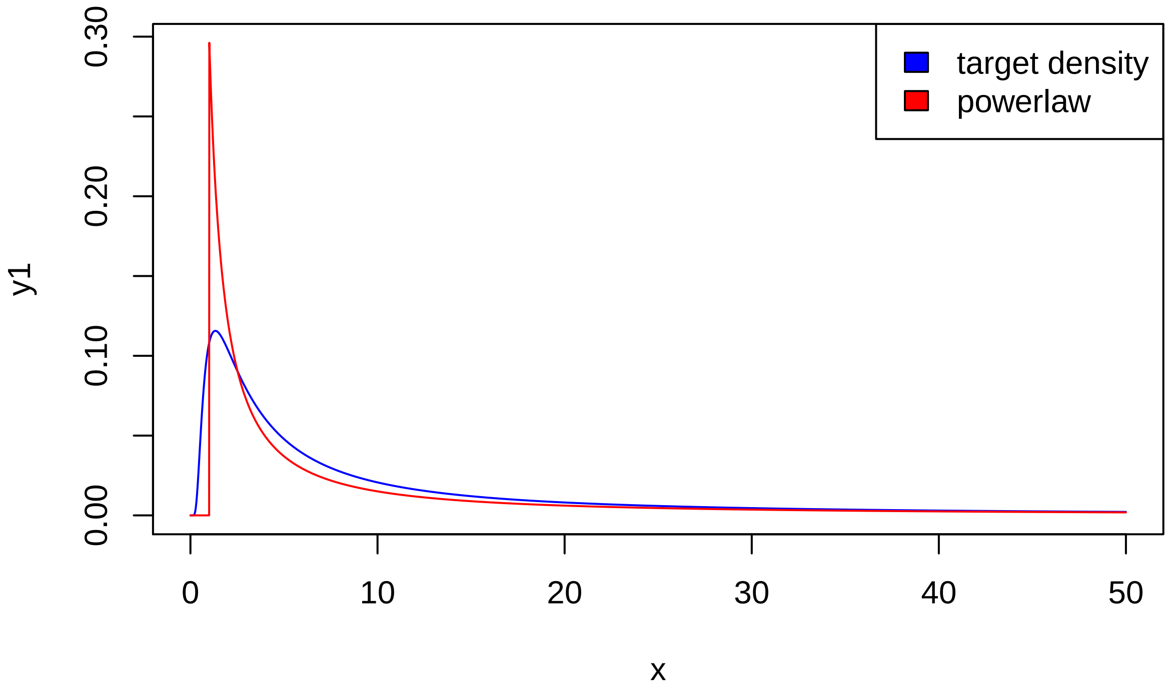
```r
distrib1 <- function(x,c){
  if(x>0){
    return(c*(sqrt(2*pi)^(-1))*(exp(-(c^2)/(2*x))) * (x^(-3/2)))
  }
  else return(0)
}

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 <- 2
y1 <- sapply(x, distrib1, c=c)
y2 <- sapply(x, powerlaw, a = 1.3,t = 1)
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 doesnt support range from 0 to T_{min}. parameters from 0 to 1 can be taken from another distribution

lets assume that $T_{min} = 1$

probability that number will be in 0-1 region is:

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

lets assume that other distribution will be uniform(0,1)

```
majDensity <- function(x){
  if(x>1){
    return(powerlaw(x, 1.6, 1))
  }
  else {return(dunif(x,0,1))}
}
# y3 <- sapply(x, target)
# plot(x,y3, type = "l")
```

```
# #lines(x,dnorm(x,3,1.2), col ="red")
# lines(x,y1, col ="blue")
# #lines(x,y1, col = "red")
```

2

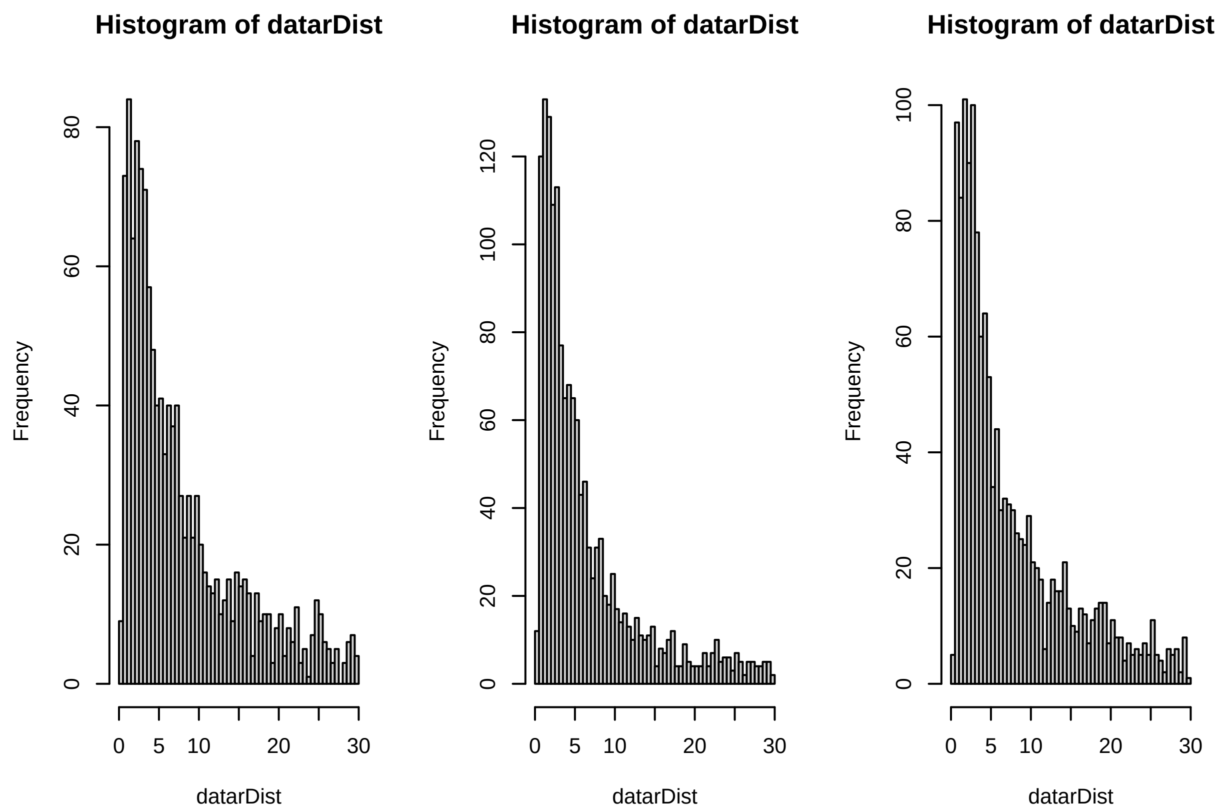
```
library(poweRlaw)
```

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

```
randomnumber <- function(prob){
  numb <- runif(1)
  if(numb<= prob){
    return(runif(1,0,1))
  }
  else{
    return(rplcon(1,1,1.5))
  }
}

CompleteDist <- function(c, rej){
  z <- TRUE
  res <- 0
  while (z == TRUE) {
    y <- randomnumber(0.5)
    u <- runif(1)
    if(u <= distrib1(y, 2) / (c*majDensity(y))){
      res <- y
      z <- FALSE
    }
    if(rej){
    rejected <<- rejected + 1
    }
  }
  return(res)
}

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



Question 2

1.

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

- μ - location parameter
- $b > 0$ - scale parameter

inverse CDF of DE:

Source - https://en.wikipedia.org/wiki/Laplace_distribution

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

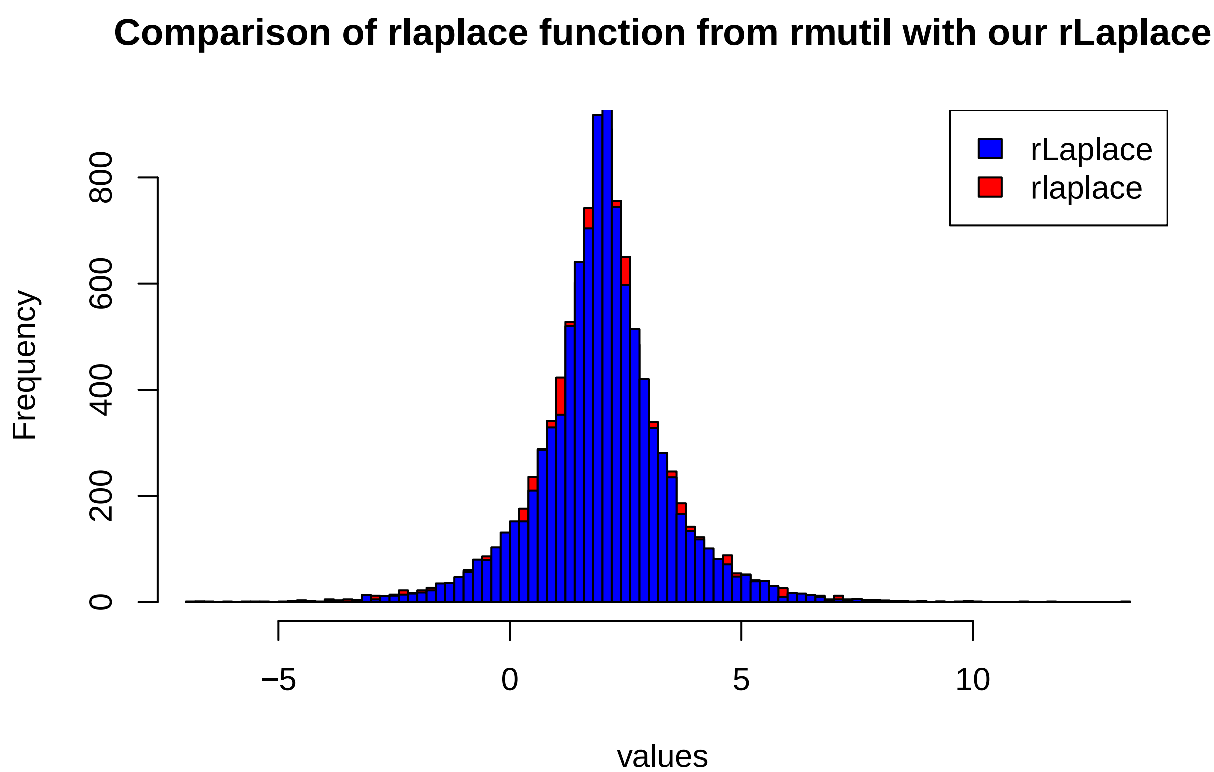
where $b = \frac{1}{\alpha}$

```
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



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)
}
```

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

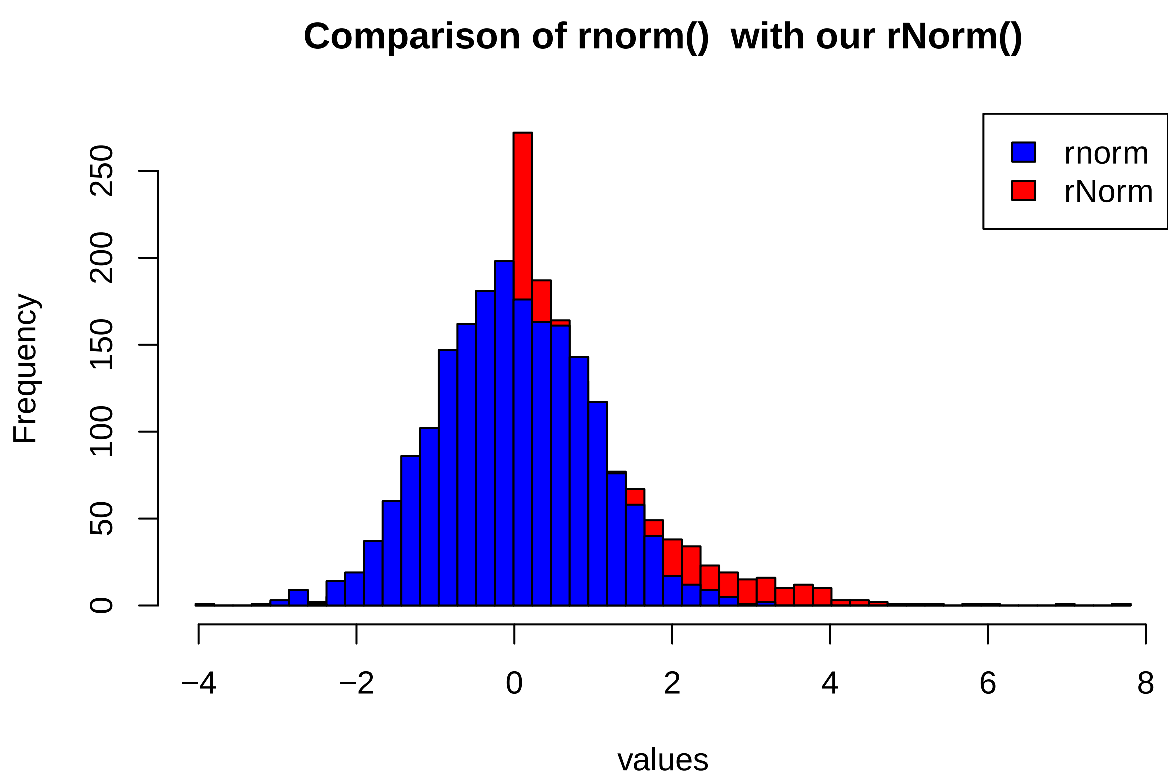
algorithm:

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

Estimated optimal c - 1.3154892

source - http://www.columbia.edu/~ks20/4703-Sigman/4703-07-Notes-ARM.pdf



	mean	variance
rNorm()	0.3648458	1.3887557
rnorm()	-0.0158433	0.9534873

rejection rate: 1.249, expected rejection rate = c = 1.3154892, difference - -0.0664892