Computational Statistics 732A90

Lab 3

Albin Västerlund
albva223

Eric Herwin
Erihe068

Contents

Assigment 1. Cluster sampling

In this assigment we will use random sampling on the data set population without replacement with the probabilities proportional to the number of inhabitants of the city to select 20 cities.

1

Import data.

```
data <- read.csv2("population.csv", fileEncoding = "iso-8859-1")
data <- data[order(data$Population, decreasing = FALSE),]
data$Municipality<-as.character(data$Municipality)
```

2

In this assgigment we will use a uniform random number generator to create a function that selects 1 city from the whole list by the probability scheme offered above.

```
select <- function(data){

  #uniform random number generator
  U <- runif(1, 0,1)

  #slide 15 lecture 3
  prob <- data$Population/sum(data$Population)

  n <- 1
  ##Loooop
  while(U >= sum(prob[1:n])){
    n <- n+1
  }
  n # the selected city
}
```

3 and 4

In this assigment we will use our function that was made in 1.2 to select citys so we can remove them from our list. We will do so until we only have 20 citys left.

```
####### 3 -4  #######
set.seed(12345678)
data_temp<-data
```

```
while(nrow(data_temp) > 20){
  num <- select(data_temp)
  data_temp <- data_temp[-num,]
}
data_temp
```

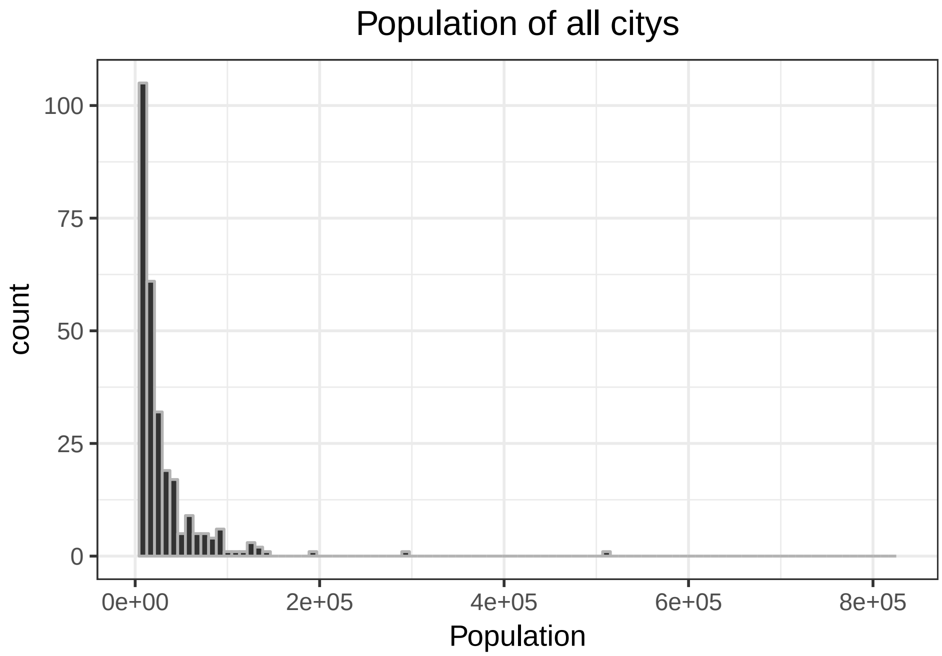
```
##     Municipality Population
## 270      Sorsele       2743
## 265         Malå       3295
## 54          Ydre       3672
## 140      Dals-Ed       4729
## 290   Övertorneå       4920
## 258      Ragunda       5609
## 81        Högsby       5873
## 255       Bräcke       6865
## 268  Robertsfors       6880
## 207     Lekeberg       7123
## 273   Vilhelmina       7156
## 266   Nordmaling       7205
## 52      Vadstena       7420
## 275       Vännäs       8357
## 288      Älvsbyn       8387
## 73      Markaryd       9559
## 229 Malung-Sälen      10408
## 234        Säter      10900
## 182         Åmål      12434
## 172     Tidaholm      12632
```

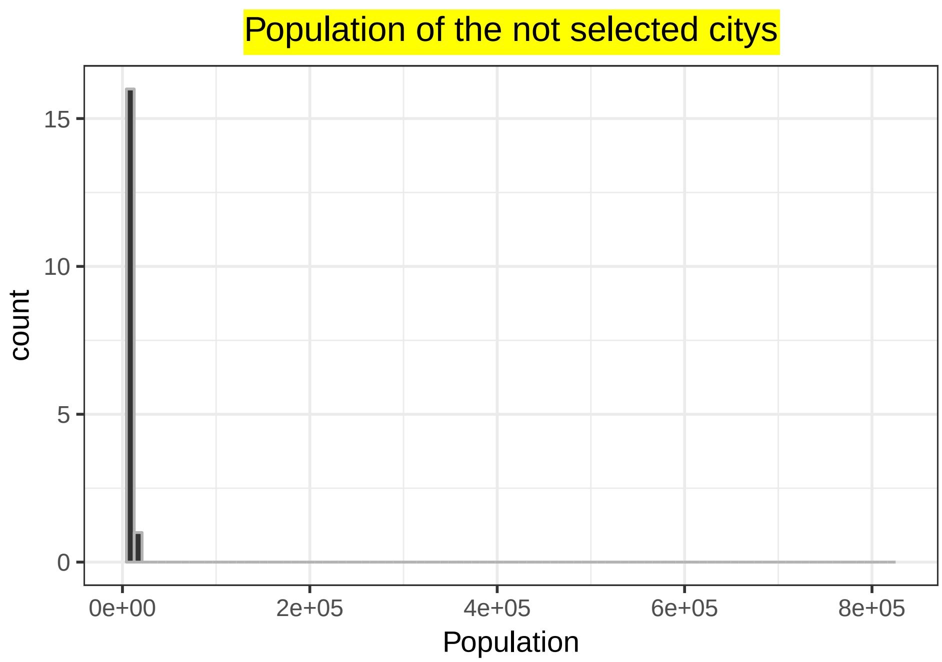
Here we see a list with 20 citys that was not selected from our function.

We can see that the biggest citys was removed from our list. Thats because they have a higher chans to get selected because we use a probability that is proportional to the size of the citys.

5

In this assigment we will make a histogram of the size of the citys thats still in the list.





By looking at the graphs it seems like the list with the 20 citys is representative of the density of the whole population.

If we would have bigger data it would probably be more convincing in the graphs.

Assigment 2. Different distributions

1

In this assigment we will siumulate new values from the double exponential (Laplace) distribution with $\mu = 0$ and $\alpha = 1$ by using the inverse CDF method. We are allowed to simulate values from the uniform distribution by using the standard function in `r`.

We start by doing it mathematically.

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

$$F(x) = p(X \leq x) = \int_{-\infty}^{x} f(t)dt$$

We have two possible outcomes because we have a absolute value in the PDF function. One outcome is when $x \geq \mu$ and the other one is when $x < \mu$. When we calculate we will also set $\mu = 0$ and $\alpha = 1$.

We start with the $x \geq \mu$ outcome.

$$F(x) = \int_{-\infty}^{x} f(t)dt = \int_{-\infty}^{\mu} f(t)dt + \int_{\mu}^{x} f(t)dt = \int_{-\infty}^{0} \frac{e^{t-0}}{2}dt + \int_{\mu}^{x} \frac{-(t-0)}{2}dt = \frac{1}{2}\Big[e^t\Big]_{-\infty}^{0} + \frac{1}{2}\Big[-e^{-t}\Big]_{o}^{x} = 1 - \frac{e^{-x}}{2} = y$$

Which gives us:

$$F_x^{-1}(y) = x = -ln(2-2y)$$

The second outcome is when $x < \mu$.

$$F(x) = \int_{-\infty}^{x} f(t)dt = \int_{-\infty}^{x} \frac{e^{t-0}}{2} = \frac{1}{2}\Big[e^t\Big]_{-\infty}^{x} = \frac{e^x}{2}$$

Which gives us:

$$F_x^{-1}(y) = x = ln(2y)$$

We know that when $F(x) = 0.5$ then must $x = \mu$. So when $y \geq 0.5$ we use the formula $x = -ln(2-2y)$ and when $y < 0.5$ we use $x = ln(2y)$.

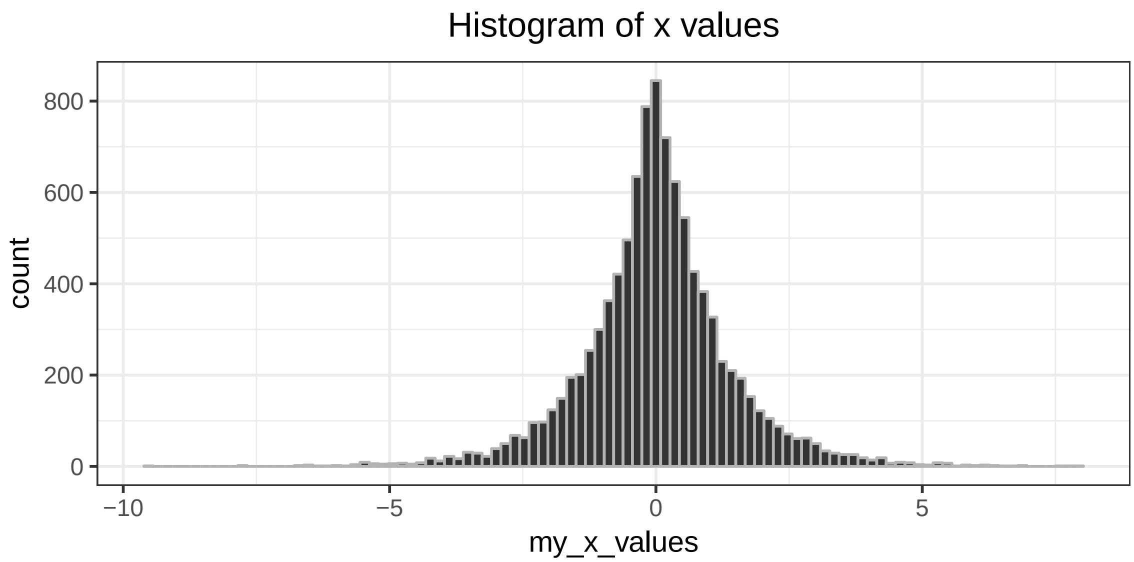
Now when we have the formulas we can simulate some values. We start by simulate 10000 values from the Unif(0,1).

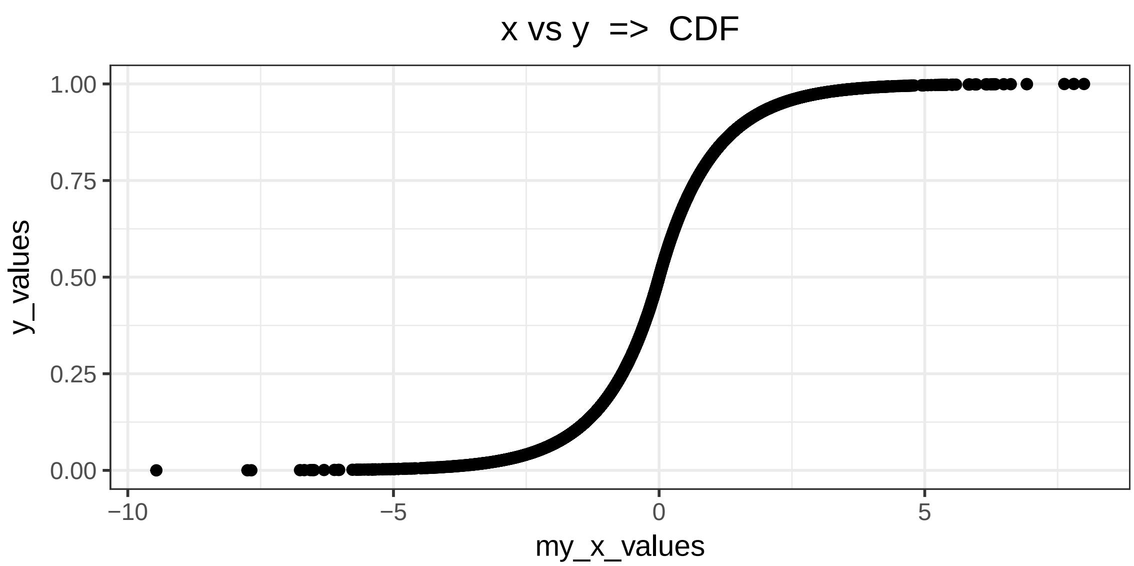
```
y_values<-runif(10000)
```

When we have the y values we can simulate the x values. Which are from double exponential (Laplace) distribution.

```
my_FUN<-function(y){
  if(y<0.5) log(2*y)
  else -log(2-2*y)
}
my_x_values<-sapply(y_values,FUN = my_FUN)
```

Now we want to plot our simulated values to see if it seems like everything is correct.





Both the histogram and the CDF look like they should do. So we are happy.

2

In this assigment we will try to simulate values from the $N(0,1)$ distribution. We will do this through the acceptance/rejection method using the DE(0,1) distribution.

We start to decide a good c value.

$f_y(x) = DE(0,1) = \frac{\alpha}{2}e^{-\alpha|x-\mu|} = \frac{1}{2}e^{-|x|}$

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

$c \cdot f_y(x) \geq f_x(x) \Rightarrow c \geq \frac{f_x(x)}{f_y(x)}$

$\frac{f_x(x)}{f_y(x)} = \frac{\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}}{\frac{1}{2}e^{-|x|}} = \frac{2}{\sqrt{2\pi}}e^{-\frac{x^2}{2}+|x|}$

$c \geq \frac{2}{\sqrt{2\pi}}e^{-\frac{x^2}{2}+|x|} \iff 1 \geq \frac{2}{c\cdot\sqrt{2\pi}}e^{-\frac{x^2}{2}+|x|} \iff ln(1) \geq -\frac{x^2}{2} + |x| + ln\left(\frac{2}{c\cdot\sqrt{2\pi}}\right)$

$a = -\frac{x^2}{2}$

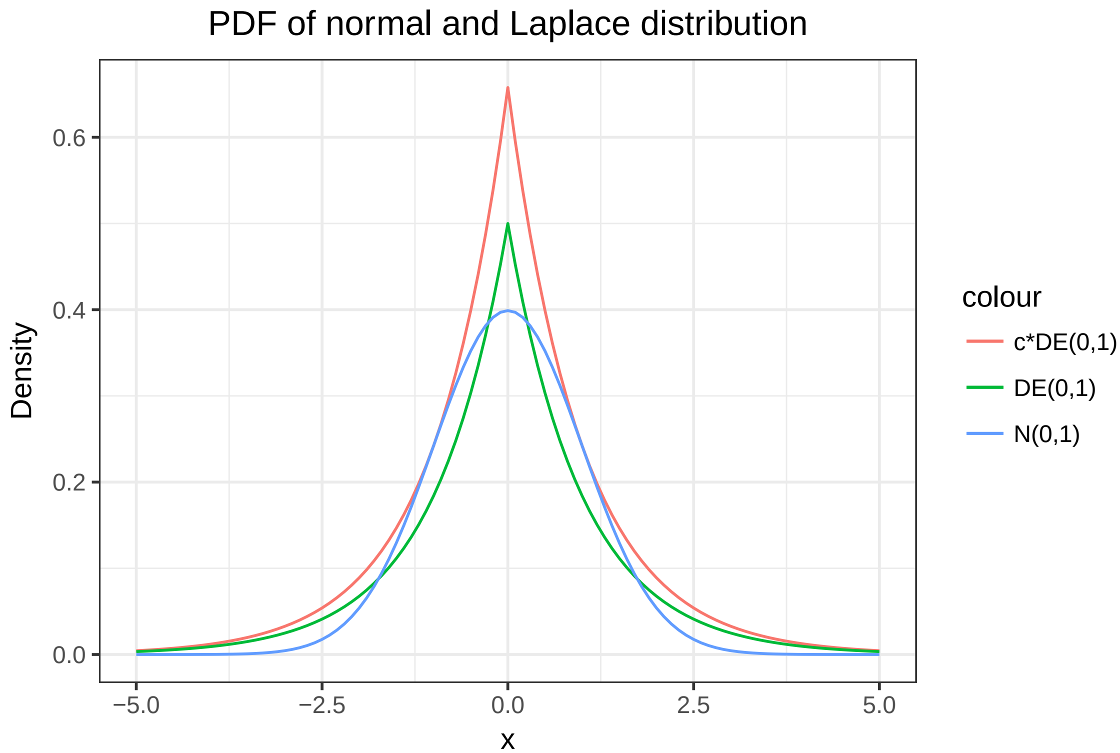
$b = |x|$

$d = ln\left(\frac{2}{c\cdot\sqrt{2\pi}}\right)$

$\frac{-b\pm\sqrt{b^2-4ad}}{2a} = 0$

We now set the c so that $b^2 - 4ad = 0$ so we only get one root. A such c is $c = e^{ln(2)+\frac{1}{2}-ln(\sqrt{2\pi})} = 1.32$.

We can make a plot to see that $f_y(x)$ always is bigger or equal then $f_x(x)$



The selected c seems to be the optimal one.

Now when we have the c we can use the acceptance/rejection method.

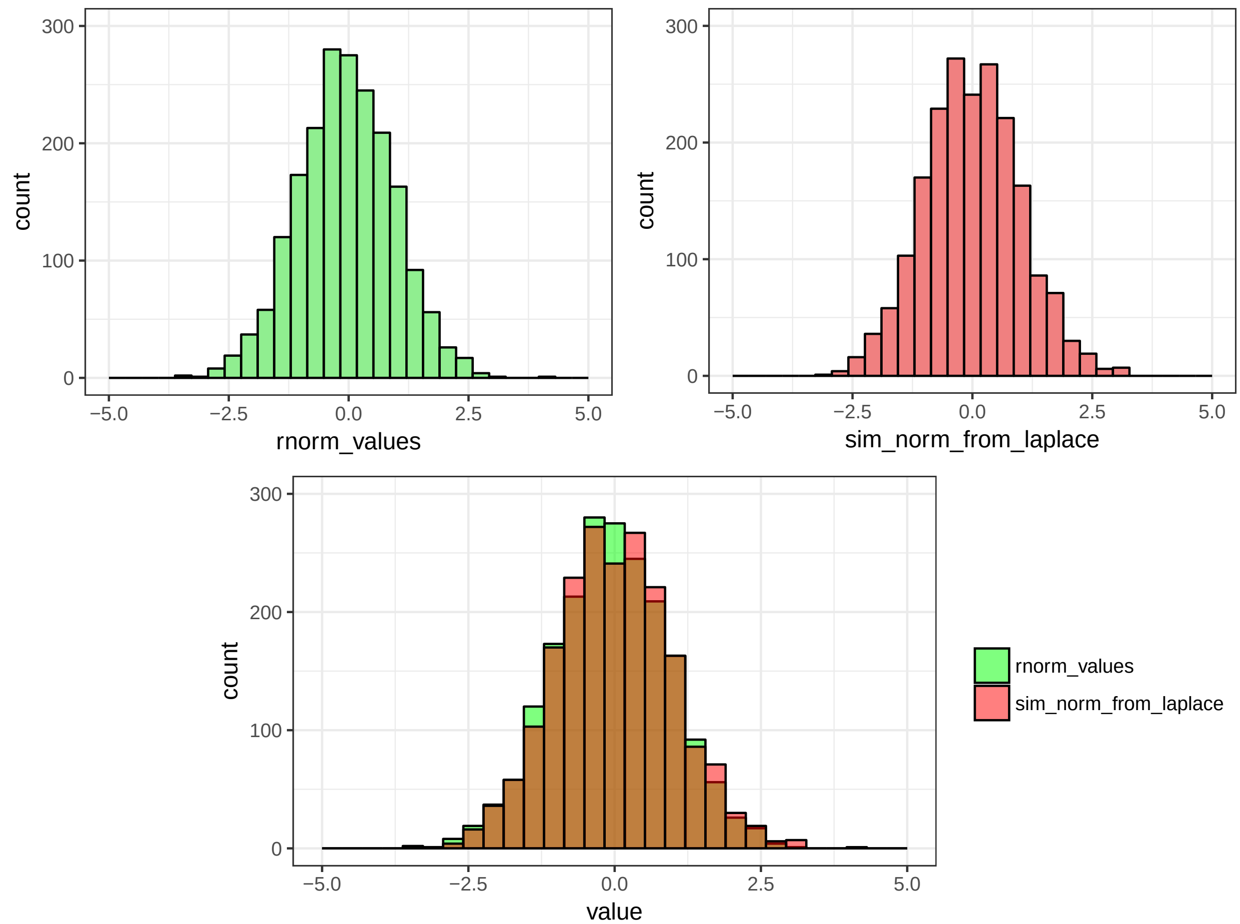
```
c_value <- exp(log(2) + 1/2 - log(sqrt(2*pi)))
sim_norm_from_laplace <- NULL
count <- 0
n <- 2000

for(i in 1:n){
  #Set to start the loop
  u <- Inf
  dn <- dl <- 1
  while(u > dn/(c_value*dl)){
    count <- count + 1
    u <- runif(1)                        #Sim Uniform value
    y<-sapply(runif(1),FUN = my_FUN) #Sim Laplace value

    dn <- 1/sqrt(2*pi) * exp(-(y)^2 / 2) #PDF-value Normal-dist
    dl <- 1/2 * exp(-1*abs( y  -0))       #PDF-value Laplace-dist
  }
  sim_norm_from_laplace[i] <- y
}
```

Now we hopefully have values that are distributed as N(0,1).

To check if they are N(0,1) we will plot the values in a histogram and compare the histogram with a histogram with values simulated from the `rnorm()` function in r.

```
# rnorm ####
rnorm_values<-rnorm(2000)
```

```
## No id variables; using all as measure variables
```



We can see that the histogram is similar. Which means that we have simulated N(0,1) values by using the acceptance/rejection method.

We will also compute the average rejection rate and the expected rejection rate.

```
R<-(count-n)/count
ER<-(c_value-1)/c_value

R
```

```
## [1] 0.2296
```

```
ER
```

```
## [1] 0.2398
```

The average rejection rate is in our case is 0.2296 and the expected rejection rate is $ER = \frac{c-1}{c} = 0.24$. The ER and R differ by 0.0102.

Appendix

R-code

```
# Library ####
library(tidyverse)
library(gridExtra)
options(digits = 4)

data <- read.csv2("population.csv", fileEncoding = "iso-8859-1")
data <- data[order(data$Population, decreasing = FALSE),]
data$Municipality<-as.character(data$Municipality)

select <- function(data){

  #uniform random number generator
  U <- runif(1, 0,1)

  #slide 15 lecture 3
  prob <- data$Population/sum(data$Population)

  n <- 1
  ##Loooop
  while(U >= sum(prob[1:n])){
    n <- n+1
  }
  n # the selected city
}

####### 3 -4  #######
set.seed(12345678)
data_temp<-data

while(nrow(data_temp) > 20){
  num <- select(data_temp)
  data_temp <- data_temp[-num,]
}
data_temp
all<-ggplot(data,aes(x=Population))+
  geom_histogram(bins =100,col="grey70",fill="grey20")+
  theme_bw()+
  ggtitle("Population of all citys")+
   theme(plot.title = element_text(hjust = 0.5))+
  scale_x_continuous(limits = c(0,max(data$Population)))


not_selected<-ggplot(data_temp,aes(x=Population))+
  geom_histogram(bins = 100,col="grey70",fill="grey20")+
  theme_bw()+
  ggtitle("Population of the not selected citys")+
   theme(plot.title = element_text(hjust = 0.5))+
    scale_x_continuous(limits = c(0,max(data$Population)))
```

```r
grid.arrange(all,not_selected,ncol=1)
y_values<-runif(10000)
my_FUN<-function(y){
  if(y<0.5) log(2*y)
  else -log(2-2*y)
}
my_x_values<-sapply(y_values,FUN = my_FUN)
histo<-ggplot()+
  geom_histogram(aes(my_x_values),bins =100,col="grey70",fill="grey20")+
  theme_bw()+
  ggtitle("Histogram of x values")+
  theme(plot.title = element_text(hjust = 0.5))

our_cdf<-ggplot()+
  geom_point(aes(x=my_x_values,y=y_values))+
  theme_bw()+
  ggtitle("x vs y  =>  CDF")+
  theme(plot.title = element_text(hjust = 0.5))

grid.arrange(histo,our_cdf,ncol=1)
x<-seq(-5,5,0.1)
c_value <- exp(log(2) + 1/2 - log(sqrt(2*pi)))
dn <- 1/sqrt(2*pi) * exp(-(x)^2 / 2) #PDF-value Normal-dist
dl <- 1/2 * exp(-1*abs( x  -0))      #PDF-value Laplace-dist
dl2<-dl*c_value

ggplot(mapping = aes(x=x))+
  geom_line(aes(y=dl2,col="c*DE(0,1)"))+
  geom_line(aes(y=dl,col="DE(0,1)"))+
  geom_line(aes(y=dn,col="N(0,1)"))+
  theme_bw()+
  ggtitle("PDF of normal and Laplace distribution")+
  theme(plot.title = element_text(hjust = 0.5))+
  labs(y="Density")
c_value <- exp(log(2) + 1/2 - log(sqrt(2*pi)))
sim_norm_from_laplace <- NULL
count <- 0
n <- 2000

for(i in 1:n){
  #Set to start the loop
  u <- Inf
  dn <- dl <- 1
  while(u > dn/(c_value*dl)){
    count <- count + 1
    u <- runif(1)                    #Sim Uniform value
    y<-sapply(runif(1),FUN = my_FUN) #Sim Laplace value

    dn <- 1/sqrt(2*pi) * exp(-(y)^2 / 2) #PDF-value Normal-dist
    dl <- 1/2 * exp(-1*abs( y  -0))      #PDF-value Laplace-dist
  }
  sim_norm_from_laplace[i] <- y
}
```

```r
# rnorm ####
rnorm_values<-rnorm(2000)
# Histogram ggplot ####
hej<-data.frame(rnorm_values,sim_norm_from_laplace)
hej2<-reshape2::melt(hej)
both<-ggplot(hej2, aes(x=value,fill=variable)) +
  geom_histogram(color="black",bins = 30 ,position = "identity",alpha=0.5)+
  theme_bw()+  scale_fill_manual("",breaks = c("rnorm_values", "sim_norm_from_laplace"),
                                 values=c("green", "red"))+
  scale_y_continuous(limits =c(0,300) )+
  scale_x_continuous(limits =c(-5,5) )

normal_values_hist<-ggplot(mapping = aes(x=rnorm_values)) +
  geom_histogram(color="black",fill="lightgreen",bins = 30 )+
  theme_bw()+
  scale_y_continuous(limits =c(0,300) )+
  scale_x_continuous(limits =c(-5,5) )

sim_normal_values_hist<-ggplot(mapping = aes(x=sim_norm_from_laplace)) +
  geom_histogram(color="black",fill="lightcoral",bins = 30 )+
  theme_bw()+
  scale_x_continuous(limits =c(-5,5) )+
  scale_y_continuous(limits =c(0,300) )


lay <- rbind(c(1,1,1,2,2,2),
             c(4,3,3,3,3,3))
grid.arrange(normal_values_hist,sim_normal_values_hist,both,layout_matrix = lay)

R<-(count-n)/count
ER<-(c_value-1)/c_value

R
ER
##
```