Question 1: Cluster sampling

An opinion poll is assumed to be performed in several locations of Sweden by sending interviewers to this location. Of course, it is unreasonable from the financial point of view to visit each city. Instead, a decision was done to use random sampling without replacement with the probabilities proportional to the number of inhabitants of the city to select 20 cities. Explore the file population.xls. Note that names in bold are counties, not cities.

1. Import necessary information to R.

2. Use a uniform random number generator to create a function that selects 1 city from the whole list by the probability scheme offered above (do not use standard sampling functionspresent in R).

```
select_city <- function(data){
  rand_num <- runif(1, min=0,max=sum(data$Population))
  pop=0 #population
  city_pop = cbind(City=data$city, Population=data$Population)

  for(i in 1:nrow(data)){
    pop = data$Population[i] + pop ## why add the previous population?
    if (pop >= rand_num)
      return (city_pop[i,])
  }
}
```

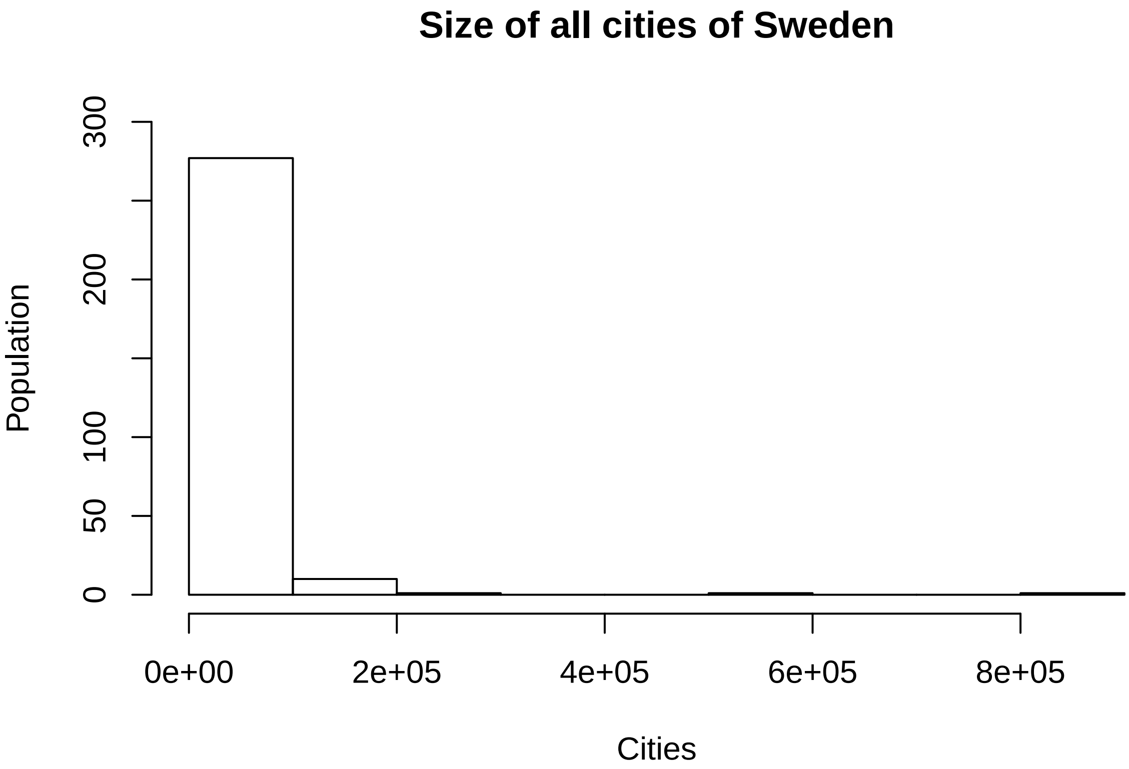
3. Use the function you have created in step 2 as follows:

(a) Apply it to the list of all cities and select one city
(b) Remove this city from the list
(c) Apply this function again to the updated list of the cities
(d) Remove this city from the list
(e) … and so on until you get exactly 20 cities.

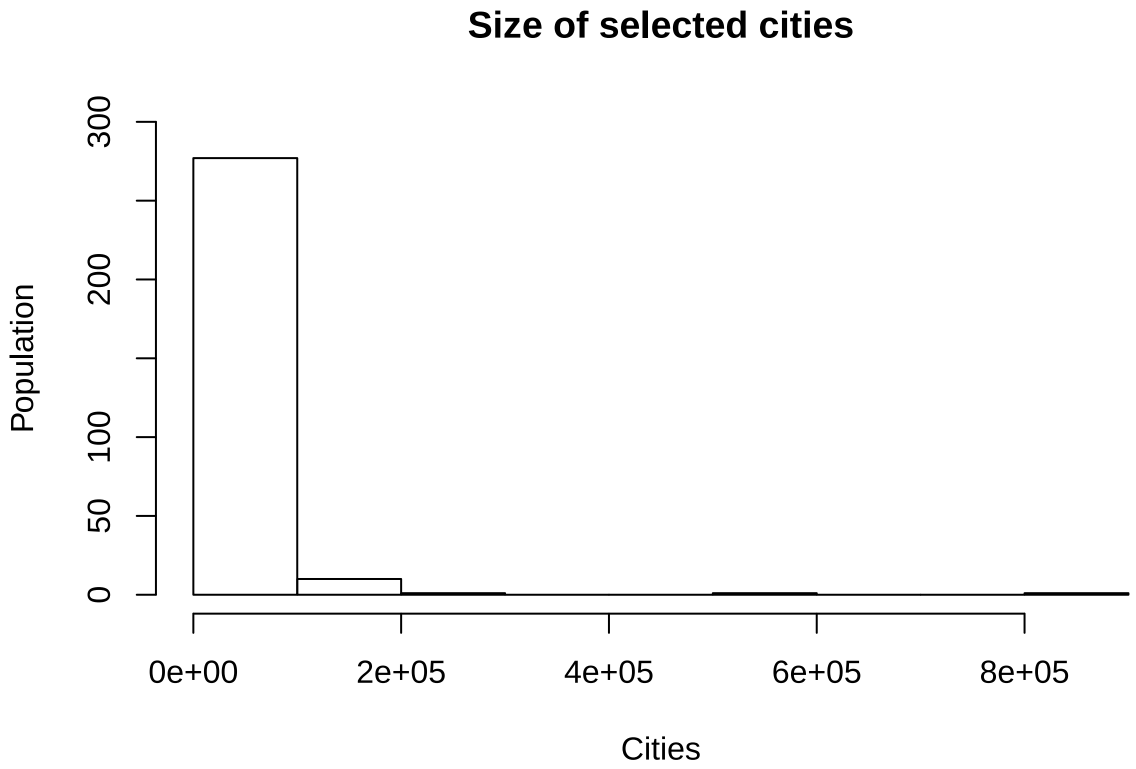
```
cities=data.frame(City=vector(length=20), Population=vector(length=20))

for(i in 1:20){
  cities[i,] <- select_city(data1)
}
```

5. Plot one histogram showing the size of all cities of the country. Plot another histogram showing the size of the 20 selected cities. Conclusions?

```
hist(data1$Population, main = "Size of all cities of Sweden", ylim = c(0,300),xlab="Cities",ylab="Popula
```



```r
hist(data1$Population, main = "Size of selected cities", ylim = c(0,300),xlab="Cities",ylab="Population
```



Question 2: Different distributions

The double exponential (Laplace) distribution is given by formula:
$DE(\mu, \alpha) = \frac{a}{2} exp(-a|x-\mu|)$

1. Write a code generating double exponential distribution DE(0,1) from Unif(0, 1) by using the inverse CDF method. Explain how you obtained that code step by step. Generate 10000 random numbers from this distribution, plot the histogram and comment whether the result looks reasonable.

We want to generate 10000 random numbers from the double exponential distribution DE(0,1) using the inverse CDF method. The steps of the process will be presented below.

1. The propability density function of ED(0,1) is:

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

and the Cumulative density function is:

For $\underline{x \geq 0:}$

$$F_x(x) = \int_{-\infty}^{x} \frac{1}{2}e^x = \frac{1}{2}e^x\Big|_{-\infty}^{x} = \frac{1}{2}e^x$$

For $\underline{x < 0:}$

$F_x(x) = \int_{-\infty}^{x} \frac{1}{2}e^{-x} = \int_{-\infty}^{0} \frac{1}{2}e^{-x} + \int_{0}^{x} \frac{1}{2}e^{-x} =$
$\frac{1}{2}(-e^{x})\Big|_{-\infty}^{0} + \frac{1}{2}(-e^{x})\Big|_{0}^{x} = \frac{1}{2} - \frac{1}{2}e^{-x} + \frac{1}{2} = 1 - \frac{1}{2}e^{-x}$

So,

$$F_x(x) = \begin{cases} 1 - \frac{e^{-x}}{2} & x \geq 0 \\ \frac{e^{x}}{2} & x < 0 \end{cases}$$

Now we have to find the inverse CDF.

$\underline{x \geq 0:}$

$y = 1 - \frac{e^{-x}}{2} \Rightarrow e^{-x} = 2(1-y) \Rightarrow$
$x = -ln2(1-y)$, $\frac{1}{2} < y < 1$

$\underline{x < 0:}$

$y = \frac{e^{x}}{2} \Rightarrow e^{x} = 2y \Rightarrow$
$x = ln2y$, $0 < y < \frac{1}{2}$

Thus, we have that

$$F_x^{-1}(y) = \begin{cases} ln2y & 0 < y < \frac{1}{2} \\ -ln2(1-y) & \frac{1}{2} < y < 1 \end{cases}$$

Hence, if $U \sim U(0,1)$ then

$$F_x^{-1}(U) = \begin{cases} ln2U & 0 < U < \frac{1}{2} \\ -ln2(1-U) & \frac{1}{2} < U < 1 \end{cases}$$

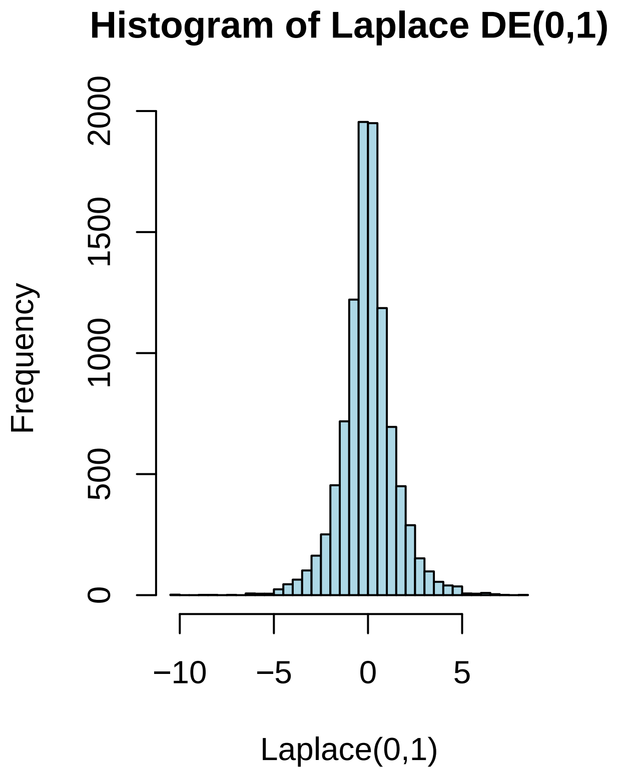
= X ~ E(0,1).

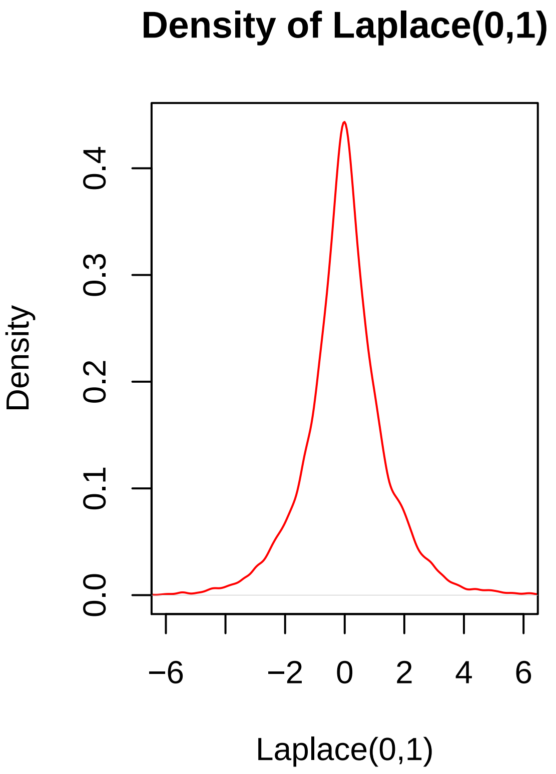
We present also the algorithm for this simulation:

$\underline{Algorithm:}$

1) Generate U ~ U(0,1)
2) Find $F_x^{-1}(y)$
3) X = $F_x^{-1}(U) \sim E(0,1)$

```
## Loading required package: MASS
```





To see if the histogram looks reasonable we used package *smoothmest* to plot the probability density function of the double exponential distribution DE(0,1) and compare it to the histogram. As we see from the plots above, the histogram of the 10000 random numbers generated from U(0,1) using the inverse CDF method looks distributed as DE(0,1) so the results that we obtain from our simulation seem to be correct.

2. Use the Acceptance/rejection method with DE(0,1) as a majorizing density to generate N(0,1) variables. Explain step by step how this was done. How did you choose constant c in this method? Generate 2000 random numbers N(0,1) using your code and plot the histogram. Compute the average rejection rate R in the acceptance/rejection procedure. What is the expected rejection rate ER and how close is it to R? Generate 2000 numbers from N(0,1) using standard rnorm() procedure, plot the histogram and compare the obtained two histograms.

Acceptance/rejection method

Let Y be a continuous random variable with pdf $f_Y(y)$ and we want to simulate values for another continuous random variable X with pdf $f_X(x)$. In our case $Y \sim DE(0,1)$ and $X \sim N(0,1)$. So,we want to simulate values of N(0,1) distribution using DE(0,1) distribution with the Acceptance/Rejection method. What we basically do is to simulate values of the DE(0,1) and then accept them or not as values of X based on a criterion. Let c be a constant, $\frac{f_X(y)}{f_Y(y)} \leq c, \forall y$

$$f_Y(y) = \frac{1}{2}e^{-|x|}, x\epsilon\Re$$

and

$$f_X(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2}, x\epsilon\Re$$

<u>*Algorithm* :</u>

1) Produce a value of the random variable Y with pdf $f_Y(y)$

2) Produce a random number $U \sim U(0,1)$

3) If $U \leq \frac{f_X(y)}{cf_Y(y)}$ set X=Y else return to step 1.

We want to generate 2000 random numbers N(0,1) so the steps of the algorithm will be executed until we get 2000 numbers.

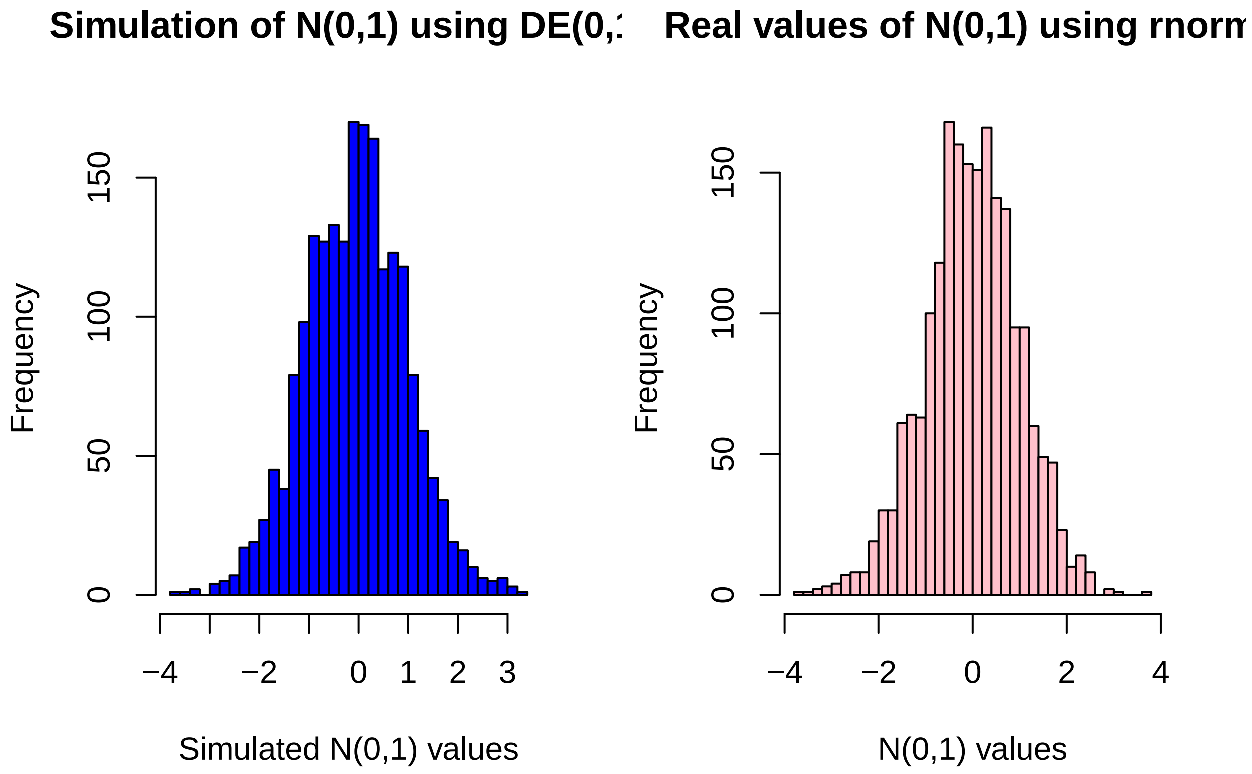
A value of X will be given for sure but what interests us as well is to get it as faster as possible, i.e. have an efficient algorithm. This is achieved when c is small, because the larger the c the bigger the rejection rate.

To find c, we will derivate the ratio $\frac{f_X(y)}{f_Y(y)}$ and see for which value of x we get the maximum. Then we put this x into the ratio and this will be the c.

Note: Laplace is symmetric around zero so we will consider here that x>0 to simplify the derivation.

$$(\frac{f_X(y)}{f_Y(y)})' = (\frac{2}{\sqrt{2\pi}}e^{x-\frac{1}{2}x^2})' = \frac{2}{\sqrt{2\pi}}e^{x-\frac{1}{2}x^2}(-x+1) = 0 \Rightarrow (-x+1) = 0 \Rightarrow x = 1$$

Now we put this x into the ratio and we get that $c = \frac{f_X(1)}{f_Y(1)} = \frac{2}{\sqrt{2\pi}}e^{1-\frac{1}{2}} = \frac{2}{\sqrt{2\pi}}e^{\frac{1}{2}} = 1.31$ (rounded in two decimals) which is the optimal value of c.



We see that the histogram generated from the simulation of N(0,1) looks similar to the one obtained using the build-in function *rnorm()* which shows that the simulation worked well. The simulated N(0,1) histogram looks normally distributed as it has the characteristic bell curve.

The probability of having acceptance in one repetition of the algorithm is $\frac{1}{c}$.

Thus,the expected rejection rate will be $1 - \frac{1}{c}$.

```
## The rejection rate is

## [1] 0.235474

## The expected rejection rate is

## [1] 0.2366412
```

We see that the rejection rate we obtained from the algorithm we implemented(~ 22)% is very close to the expected rejection rate(~ 0.24)%.