732A90-Lab3-Report

Fengjuan Chen(fench417) Zhixuan Duan(zhidu838)

2/12/2020

Question 1

1. Import neccessary information

```
data=read.csv2("population.csv",encoding = 'latin1')
```

2. Function of sampling

The sampling is based on the proabilities proportional to the number of inhabitants of the city. After we generate a uniform number, u, between 0 and 1, we compare it with the N intervals in the (0,1) which obtained by the proportioan of population of each city, where N is the number of the whole cities in the given data set.

If the uniform number, u, falls into the interval i (1 =< i <= N), then the *ith* city is selected by our function. This makes sense because the random uniform value falls within the interval i with the propbabilites proportional to the ratio of inhabitants of each city.

Therefore, our function to select one city from the whole list by the probabilities proportional to the number of inhabitants of the city is pps() function coded in R.

```
pps <- function(data){
  prop=data$Population/sum(data$Population)
  u=runif(1)
  interval=0
  for (i in 1:nrow(data)) {
    interval=interval+prop[i]
    if(u<=interval) {
      index=i
      return(index)
    }
  }
}
```

3. Sample 20 cities

We use code below to choose 20 cities from the whole list.

```
# samplers is a vector, stored indices of the 20 cities
data_org=read.csv2("population.csv",encoding = 'latin1')
data=cbind(label=c(1:nrow(data_org)),data_org)
set.seed(123456)
samplers=vector(length = 20)
for (i in 1:20) {
  index=pps(data)
  samplers[i]=index
  data=data[-index,]
}
```

4. The selected cities

```
##      Municipality Population
## 204     Karlskoga      29742
## 178    Ulricehamn      22753
## 82         Kalmar      62388
## 62      Jönköping     126331
## 72        Ljungby      27410
## 20           Täby      63014
## 122    Trelleborg      41891
## 16      Stockholm     829417
## 285         Luleå      73950
## 32        Uppsala     194751
## 214 Hallstahammar      15127
## 146      Göteborg     507330
## 250     Sundsvall      95533
## 240    Hudiksvall      36848
## 287         Piteå      40860
## 245     Sandviken      36978
## 238         Gävle      94352
## 38    Katrineholm      32303
## 89      Västervik      36290
## 201     Degerfors       9709
```

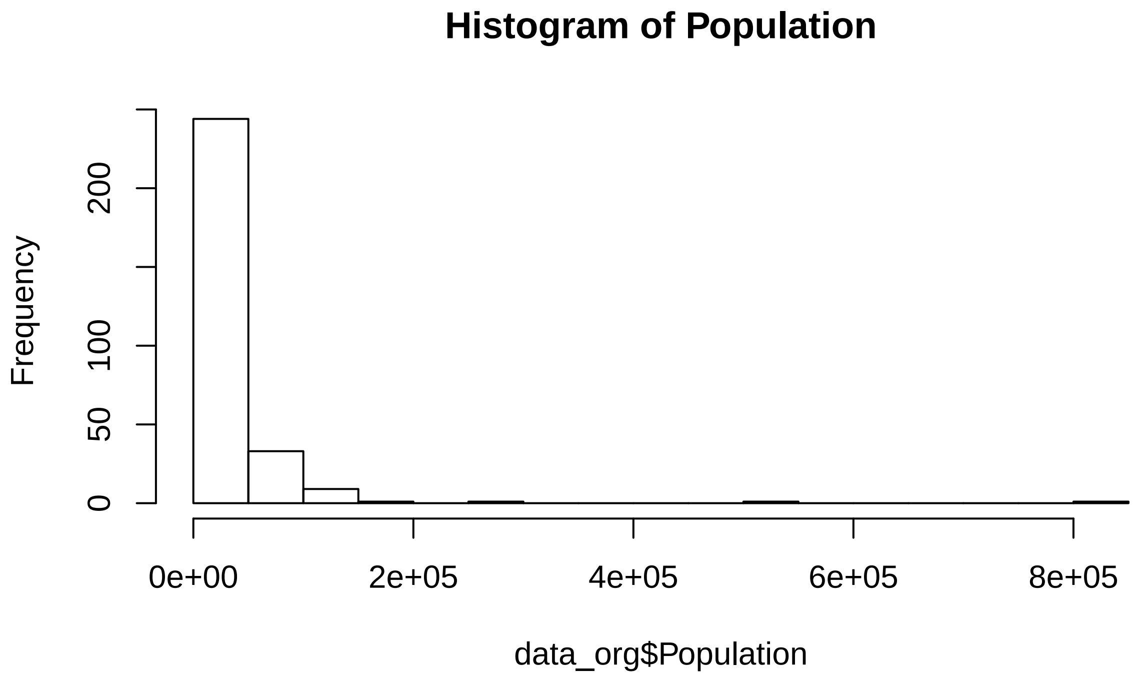
The mean and range of 20 cities populaton are :

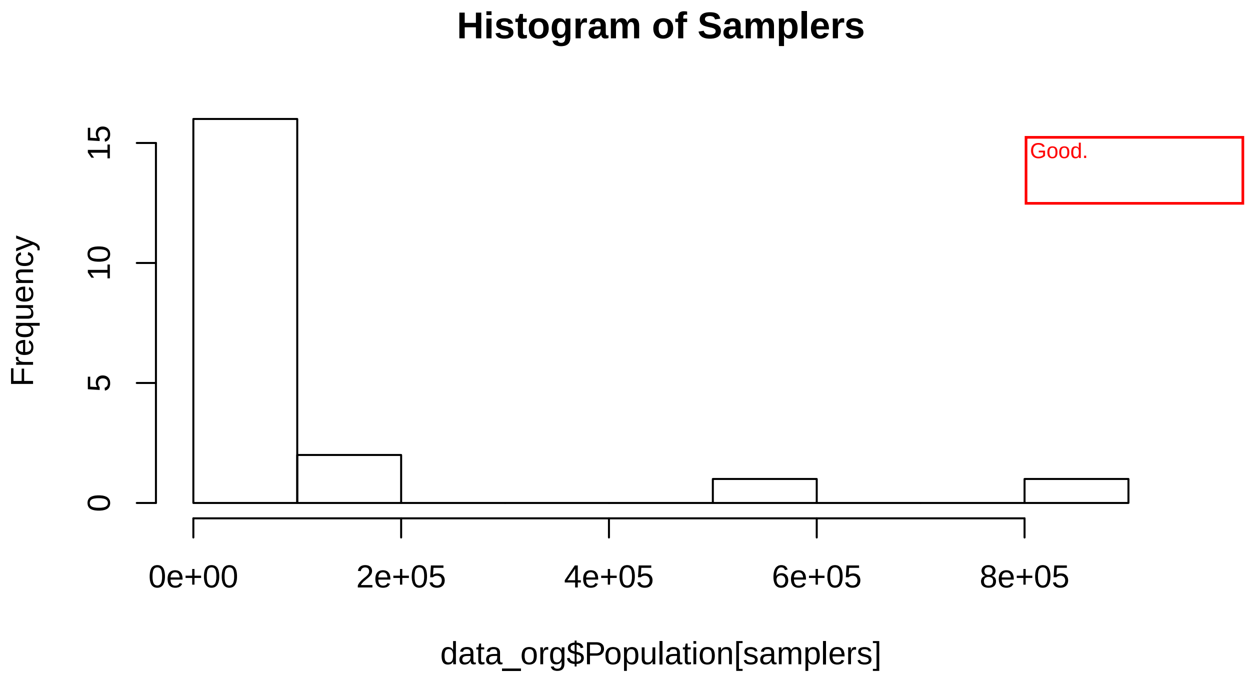
```
## Mean is:  118848.9
```

```
## range is : 9709 829417
```

The size is widely scattered in the whole list.

5. Histograms of population and sample





Good.

Question 2

1. Inverse CDF method

The steps of inverse CDF method to generate numbers from a given density distributin $f(x)$ if the inverse of its cumulative distribution function (CDF) F_x^{-1} exists aer as follows:

Step1. Calculate the CDF, F_x, of the density distribution $f(x)$.

Step2. Obtain inverse of CDF F_x, F_x^{-1}.

Step3. Generate a uniform number u, between 0 and 1, U(0,1).

Step4. Simulate a value of x by calling function $F_x^{-1}(u)$, where $u \in U(0, 1)$.

Here, we have $f(x) = DE(u, \alpha) = \frac{\alpha}{2} exp(-\alpha|x - u|)$ where $u = 0, \alpha = 1$. That is,

$$f(x) = DE(0, 1) = \begin{cases} \frac{1}{2} exp(-x) & x \geq 0 \\ \frac{1}{2} exp(x) & x < 0 \end{cases}$$

According to the steps of inverse CDF method, we first obtain the CDF F_x by calculate the integral $\int_{-\infty}^{+\infty} f(x)dx$. That is

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

Second we get the inverse of Fx by letting $y = F_x$

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

Good.

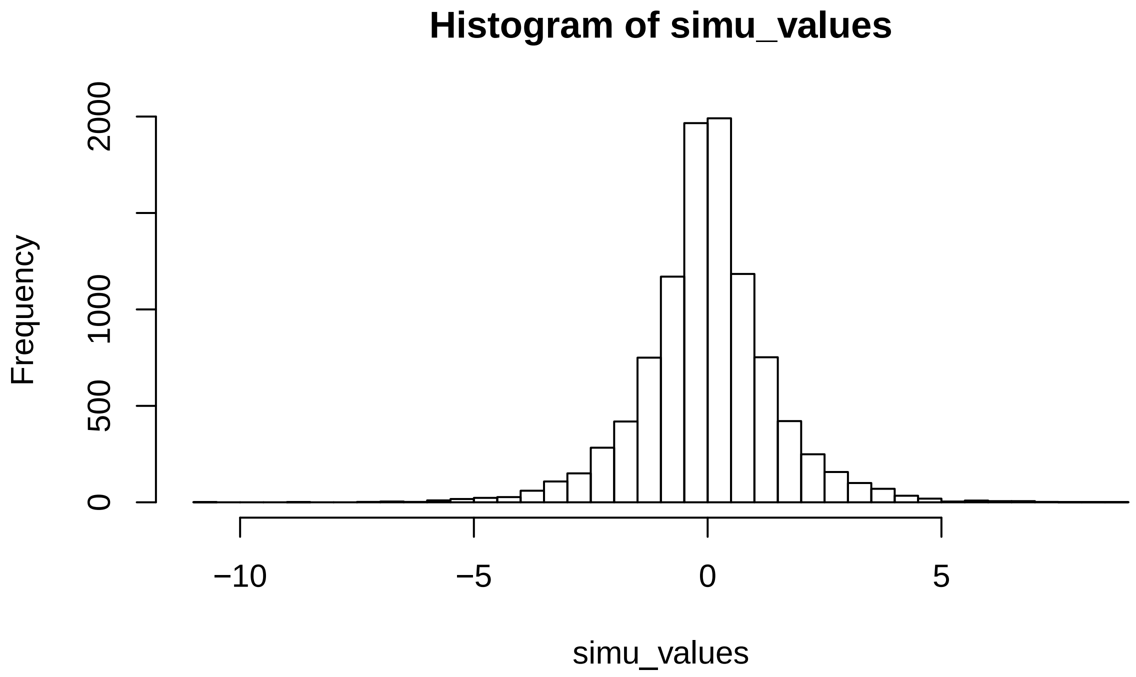
Third, we generate a uniform u belong U(0,1).

Fourth, we simulate x by u replacing y with u in the formula $F_x^{-1}(y)$.

Our function is shown below.

```
inverCDF <- function(u){
  simu_value=ifelse(u>=1/2,-log(2-2*u),log(2*u))
  return(simu_value)
}
```

We first use our function to generate 10000 randonm numbers from the double exponential (Laplace) distribution and then plot the histogram of these values.



2. Acceptance/rejection method

The steps of acceptance/rejection method to generate a number from a given density distributin (target density) by using a proposal density are as follows:

Step1. Generate a number y from the proposal density.

Step2. Generate a number u from the uniform U(0,1).

Step3. Choos a constant (majorizing constant) by maximize the ratio between target function and proposal function.

Step4. If u is less than the ratio=target/(c*proposal), then accept this value y as the simulation of x. Otherwise reject this value y and regenerate a number y and repeat the steps from step1 to step4.

In this question, the target function is $N(0,1)$ and proposal density is DE(0,1) where $N(0,1) = \frac{1}{\sqrt{2\pi}}exp(-\frac{x^2}{2})$, and $DE(0,1) = \frac{1}{2}exp(-|x|)$

How to choose constant c in this method?

To decide the value of majorizing constant c, we calculate the ratio of $\frac{target}{proposal}$ and its optimal value.

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

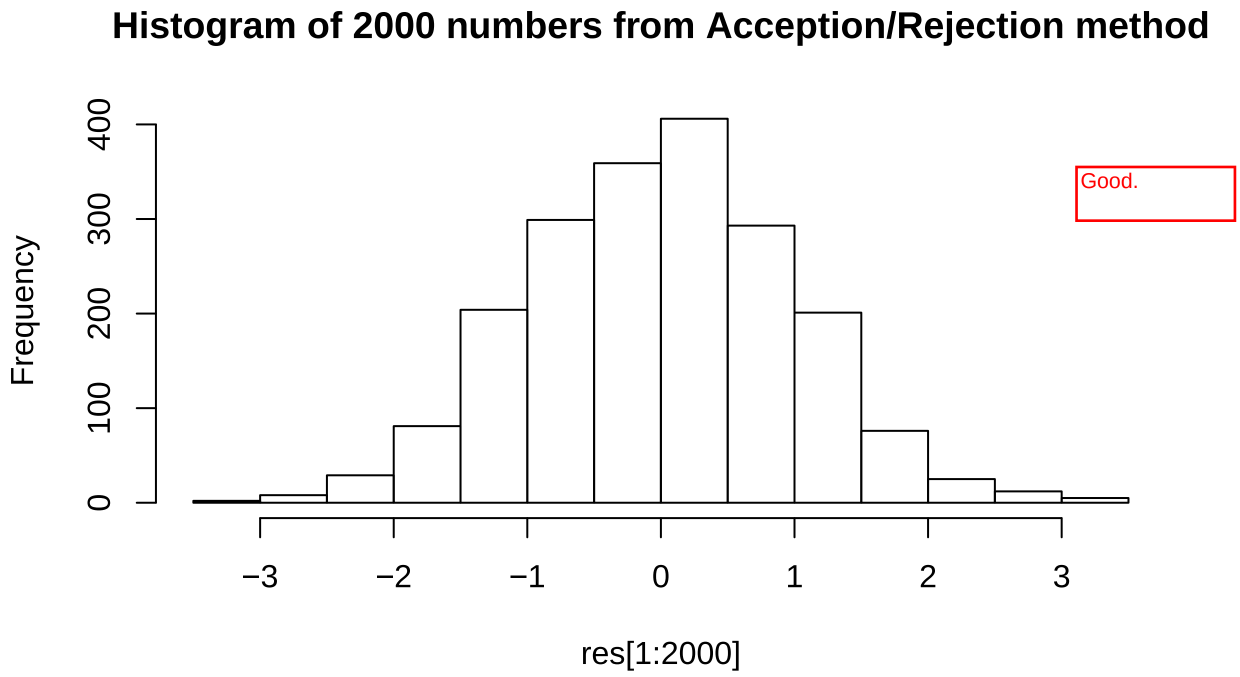
Calculate the derivative of $-\frac{x^2}{2} + |x|$ and let it equals to 0, we can obtain that $x = 1$ and $x = -1$. Putting these values $x = 1$ and $x = -1$ into $-\frac{x^2}{2} + |x|$, we can get $-\frac{x^2}{2} + |x| = \frac{1}{2}$. So, the optimal value is $c = \sqrt{\frac{2}{\pi}}exp(\frac{1}{2}) = \sqrt{\frac{2e}{\pi}} \approx 1.315489$.

Good.

The function for acceptance/rejcetion method

Our function for acceptance/rejection method with DE(0,1) as a majorizing density to generate $N(0,1)$ variables is shown below.

```r
set.seed(123456)
accep_rejec <- function(n){
  accep=0
  rejec=0
  sim_num=vector(length = n)
  while (accep<n) {
    # u1 is used to generate the value from the proposal density
    u1=runif(1)
    y=inverCDF(u1)
    target=exp(-(y^2)/2)/sqrt(2*pi)
    proposal=ifelse(y>=0,exp(-y)/2,exp(y)/2)
    c=sqrt(2*exp(1)/pi)
    # u is used to determin acceptance or rejection
    u=runif(1)
    if(u<=target/(c*proposal)){
          accep=accep+1
          sim_num[accep]=y}
    else{
          rejec=rejec+1
    }
  }
  return(c(sim_num,rejec))
}
```



Good.

The rejection rate

The average rejection rate R in the acceptance/rejection produre is 0.2301771.

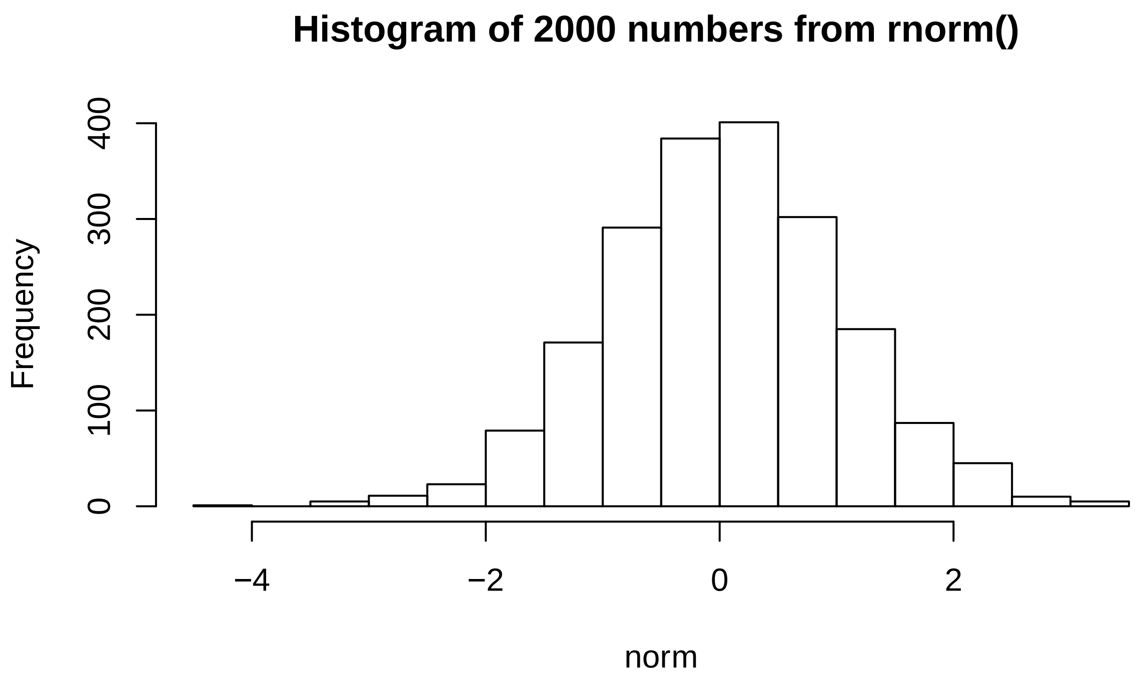
The expected rejection rate ER is equalt to $1 - 1/c$ since $1/c$ is the expected acceptance rate, where $c == \sqrt{\frac{2e}{\pi}} \approx 1.315489$ is the constant in this method. Then the expected rejection rate $ER = 1 - 1/c = 0.2398265$.

From above, we have $R = 0.2301771$ and $ER = 0.2398265$. They are very colse.

Correct.

Compare histograms

The histogram of 2000 numbers from $N(0, 1)$ using standard rnorm() procedure is shown below.



We can see that these two histograms are very familiar with each other and the histogram generated by acceptance/rejection method shows normal distribution with 2000 numbers. This means the acceptance/rejection method works for this situation.

Appendix

```r
RNGversion(min(as.character(getRversion()),"3.6.2"))
set.seed(123456)

# Question 1

## 1
#library(readxl)
data_org=read.csv2("population.csv",encoding = 'latin1')
data=cbind(label=c(1:nrow(data_org)),data_org)

## 2 function() of sampling design
set.seed(123456)
pps <- function(data){
  prop=data$Population/sum(data$Population)
  u=runif(1)
  interval=0
  for (i in 1:nrow(data)) {
    interval=interval+prop[i]
    if(u<=interval) {
      index=data$label[i] # record the label of this city
      cat(u,index,prop[i],interval,"\n")
      return(index) # return the label of the chosen city
    }
  }
}

index=pps(data)

## 3 use the function 20 times to obtain 20 samplers

samplers=vector(length = 20)
for (i in 1:20) {
  index=pps(data)
  samplers[i]=index
  data=data[-index,]
}


## 4 the 20 cities selected by the function

a=data_org[samplers,]

## 5 the histogram

hist(data_org$Population,breaks = 20,
     main = "Histogram of Population")
hist(data_org$Population[samplers],
     main = "Histogram of Samplers")
```

```r
# Question 2

## 1 Inverse CDF method
## function of simulating the values of given distribution
inverCDF <- function(u){
  simu_value=ifelse(u>=1/2,-log(2-2*u),log(2*u))
  return(simu_value)
}

## generate 10000 random numbers from the distribution
n=10000
simu_values=vector(length = n)
for (i in 1:n) {
  u=runif(1)
  simu_values[i]=inverCDF(u)
}

## plot the histogram
hist(simu_values,breaks = 30)

## 2 Acceptance/rejection method
set.seed(123456)
accep_rejec <- function(n){
  accep=0
  rejec=0
  sim_num=vector(length = n)
  while (accep<n) {
    u=runif(1)
    y=inverCDF(u)
    target=exp(-(y^2)/2)/sqrt(2*pi)
    proposal=ifelse(y>=0,exp(-y)/2,exp(y)/2)
    c=sqrt(2*exp(1)/pi)
    if(u<=target/(c*proposal)){
      accep=accep+1
      sim_num[accep]=y}
    else{
      rejec=rejec+1
    }
  }
  return(c(sim_num,rejec))
}
res=accep_rejec(n=2000)
n=2000
hist(res[1:2000],breaks = 20)

## rejection rate
rejec_num=res[2001]
rejec_rate=rejec_num/(rejec_num+2000)

norm=rnorm(2000)
hist(norm,breaks = 20)
```