# Computational Statistics Lab3

*Andreas C Charitos (andch552) Omkar Bhutra (omkbh878)*

*9 Feb 2019*

## Question 1

### Monte Carlo sampling simulation with no replacement

```
#read data with latin encoding
population<-read.csv("population.csv",sep=";",encoding = "latin1")
#make Municipality as character
population$Municipality<-as.character(population$Municipality)
#create new column with the probability for each city
population$prop<-population$Population/sum(population$Population)


##my function for generating random unoform number
# random<-function(n){
#   vec<-rep(0,n)
#
#   m<-2**30
#   a <- 1103515245
#   c <- 123456
#
#   d <- as.numeric(Sys.time()) * 1000
#
#   for (i in 1:n){
#     d<-(a*d+c)%%m
#     vec[i]<-d/m
#   }
#   return(vec)
#
# }
```

```
#my sample function
sample_func<-function(n,data){
    #create uniform
    uniform<-runif(n)
    #test the uniform to find index which is smaller than cumsum
    t2= sapply(uniform,function (x){ sum(x<=cumsum(data$prop))})
    n=length(data$prop)
    #find the index
    indx=n-t2

  return(list("index"=indx,"sample_city"=data[indx,]))

}
```

### Sampling with no replacement from original data

```r
set.seed(123456)


i<-1
#ordering the original dataset
pop1=population[order(population$prop),]
#dataframe to store the sample
samp=data.frame()
repeat{
  #take 1 sample
  mysample=sample_func(1,pop1)
  #remove this sample from the dataset
  pop1=pop1[-mysample$index,]
  #if to create the dataset columns
  if (i==1){

  samp=mysample$sample_city
  }
  else if(i!=20){
  samp=rbind(samp,mysample$sample_city)
}
  else{
    break
  }
  i=i+1
}

library(kableExtra)
```

```
## Warning: package 'kableExtra' was built under R version 3.5.2
```

```r
cat("The 20 cities that were selected with no replacement from our data are :")
```

```
## The 20 cities that were selected with no replacement from our data are :
```

```r
cat("\n")
```

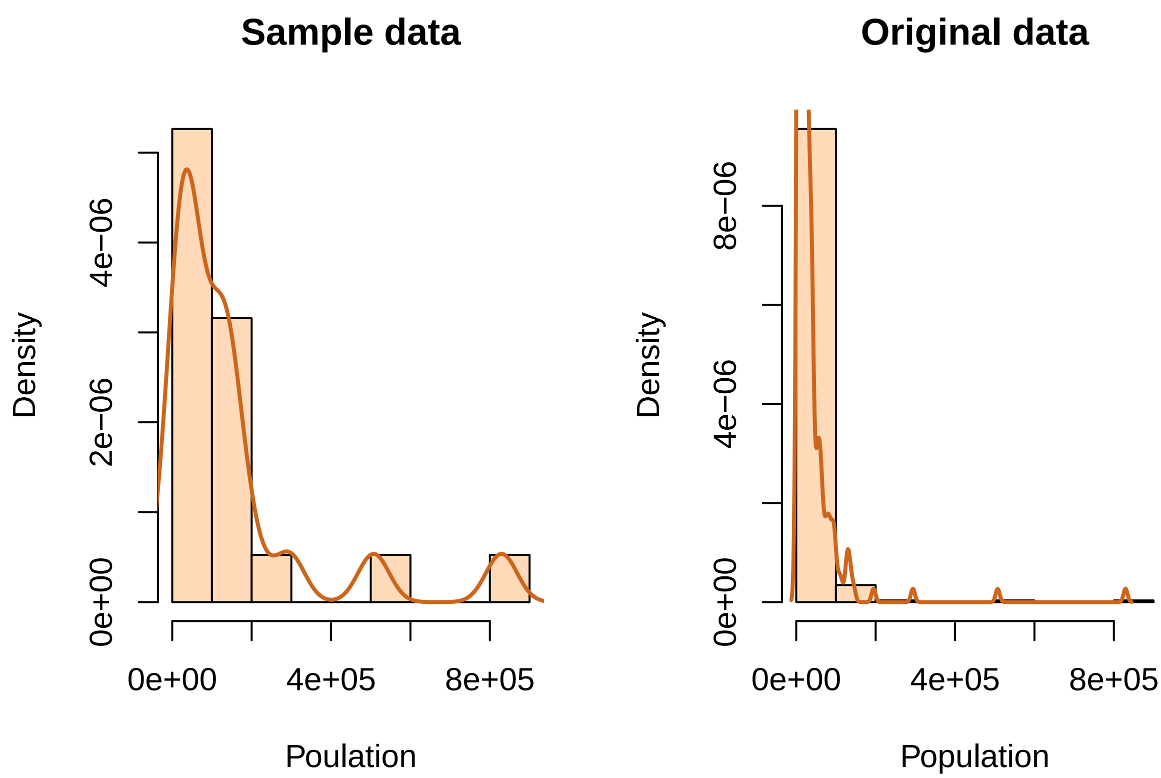
```r
knitr::kable(samp[order(samp$Population,decreasing = T),])%>%kable_styling()
```

| | Municipality | Population | prop |
|---|---|---|---|
| 16 | Stockholm | 829417 | 0.0887962 |
| 146 | Goteborg | 507330 | 0.0543140 |
| 112 | Malmo | 293909 | 0.0314655 |
| 32 | Uppsala | 194751 | 0.0208498 |
| 47 | Linkoping | 144690 | 0.0154903 |
| 221 | Vasteras | 135936 | 0.0145531 |
| 211 | Orebro | 134006 | 0.0143465 |
| 50 | Norrkoping | 129254 | 0.0138377 |
| 101 | Helsingborg | 128359 | 0.0137419 |
| 191 | Karlstad | 84736 | 0.0090717 |
| 15 | Solna | 66909 | 0.0071632 |
| 26 | Osteraker | 39173 | 0.0041938 |
| 17 | Sundbyberg | 37722 | 0.0040385 |
| 89 | Vastervik | 36290 | 0.0038852 |
| 163 | Partille | 34382 | 0.0036809 |
| 205 | Kumla | 20214 | 0.0021641 |
| 31 | Tierp | 20044 | 0.0021459 |
| 95 | Solvesborg | 16813 | 0.0018000 |
| 259 | Stromsund | 12286 | 0.0013153 |

The above table shows the cities that where selected sampling from our original data.

### Plot of histogram and density for sample and original data

```r
par(mfrow=c(1,2))
hist(samp$Population,main = "Sample data",xlab = "Poulation",
     col="peachpuff", border="black",prob = TRUE)
lines(density(samp$Population),lwd = 2, col = "chocolate3")
hist(population$Population,main="Original data",xlab = "Population",
     col="peachpuff",border="black",prob = TRUE)
lines(density(population$Population),lwd = 2,col = "chocolate3")
```




From the table and the plot we can conclude that using the sampling method we implemented we get more cities with larger population.This is because we sampling using the probabilities of the cities thus citis with larger population are more favour.

## Question 2

### 1 Inverse CDF Method

### Generate Laplace Distribution using the inverse CDF Method

Our target is to find the inverse CDF of the

$$DE(\mu, \alpha) = \frac{a}{2} exp(-\alpha|x-\mu|)(1)$$

First we need to find the CDF for (1)

$$F_x(x) = \int_{-\infty}^{x} f_x(t)\,dt$$

We have 2 cases a)$x < \mu$ and b) $x >= \mu$

Starting with a) $x < \mu$

$$F_x(x) = \int_{-\infty}^{t} \frac{a}{2} e^{(-\alpha|x-\mu|)} = \frac{a}{2}[\frac{e^{(a(x-\mu))}}{a}]_{-\infty}^{t} =$$

$$\frac{a}{2}[\frac{e^{(a(t-\mu))}}{a} - 0] = \frac{e^{(a(x-\mu)}}{2}(2)$$

Next the case b) $x >= \mu$

$$F_x(x) = \int_{-\infty}^{t} \frac{a}{2} e^{(-a|x-\mu|)} = \int_{-\infty}^{0} \frac{a}{2} e^{(-a(x-\mu))} + \int_{0}^{t} \frac{a}{2} e^{(-a(x-\mu))} (3)$$

using (2) we can find the firrst integral so (3) $=>^{(2)}$

$$[\frac{e^{(a(x-\mu))}}{2}]_{-\infty}^{0} + \frac{a}{2}[\frac{e^{(-a(x-\mu))}}{-a}]_{0}^{t} =$$

$$[\frac{1}{2} - 0] + [-\frac{a}{2}\frac{e^{(-a(x-\mu))}}{-a} - \frac{a}{2}\frac{1}{-a}] =$$

$$1 - \frac{1}{2} e^{(-a(x-\mu))} (4)$$

combining (2) and (4) the CDF of the Laplace is :

$$F(x) = \frac{1}{2} + \frac{1}{2} sign(x-\mu)(1 - e^{(-a|x-\mu|)})(5)$$

We now need to find the inverse of CDF

for $x >= \mu$

$$y = \frac{1}{2} + \frac{1}{2} sign(x-\mu)(1 - e^{(-a(x-\mu))}) =>$$

$$2ysign(x-\mu) - sign(x-\mu)(1 - e^{(-a(x-\mu))}) =>$$

$$e^{(-a(x-\mu))} = 1 - 2ysign(x-\mu) + sign(x-\mu)$$

taking the ln for both sides we have :

$$\ln \frac{1}{e^{(a(x-\mu))}} = ln(1 - 2ysign(x-\mu) + sign(x-\mu)) =>$$

$$\ln(1) - ln(e^{(a(x-\mu))} = \ln(1 - 2ysign(x-\mu) + sign(x-\mu)) =>$$

$$a(x-\mu) = \ln(1 - 2ysign(x-\mu) + sign(x-\mu)) =>$$

$$x = \mu + \frac{1}{a} \ln(1 - 2ysign(x-\mu) + sign(x-\mu))$$

$$x = \mu + \frac{1}{a} \ln(1 - 2|x-\mu|))(6)$$

following the same steps for $x < \mu$ we obtain $x = \mu - \frac{1}{a} \ln(1 - 2|x-\mu|))(7)$

Finally the inverse of CDF is combining (6),(7)

$$F^{-1}(x) = \mu - \frac{1}{a} sign(x-\mu) \ln(1 - 2|x-\mu|) = \mu - \frac{1}{a} sign(x-0.5) \ln(1 - 2|x-0.5|) \quad (See \quad appendix)$$

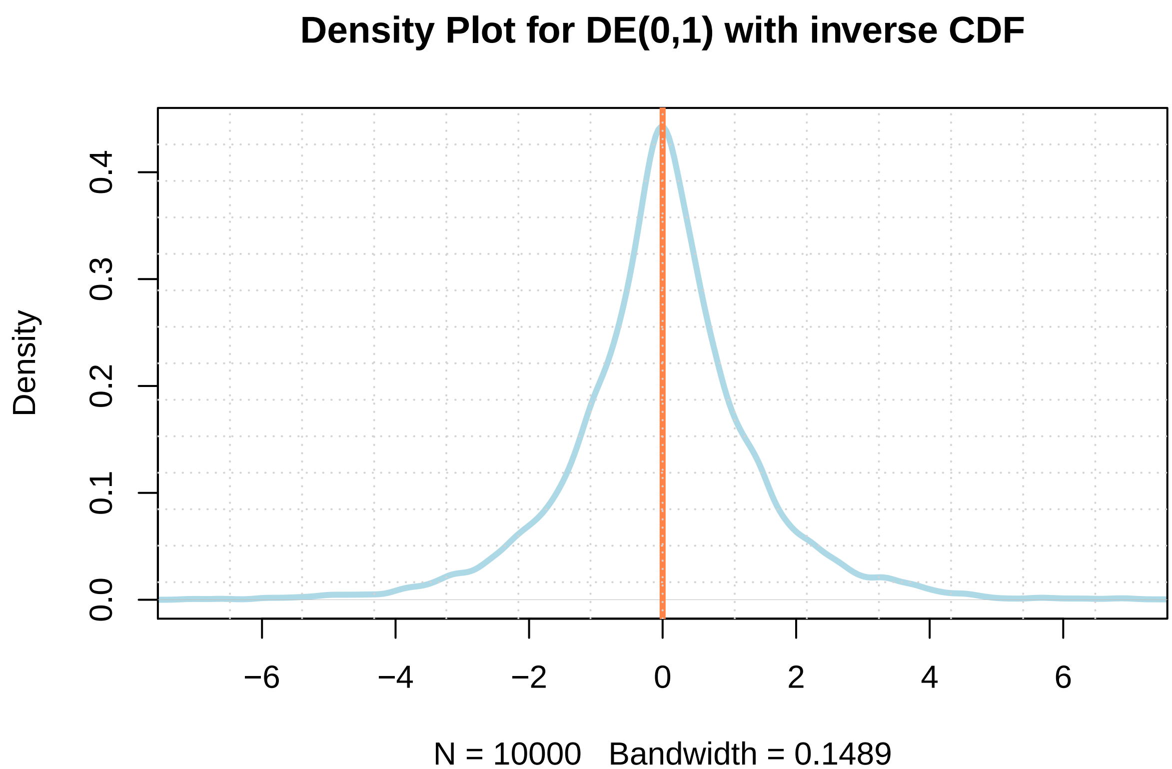
The steps for the Inverse CDF sampling algorithm are :

1.Initialize a random uniform number $U = (0, 1)$

2.Next we plug the U to inverse CDF $F^{-1}(U)$ and we obtain a sample for our laplace distribution

```
set.seed(123456)
#creating laplace using the inverse CDF

rlaplace = function(n,mu,alpha){
  U = runif(n)
  sign = ifelse(U-0.5>0,1,-1)
  y = mu - sign*(1/alpha)*log(1-2*abs(U-0.5))
  return(y)
}
#plot of 10000 samples
plot(density(rlaplace(10000,0,1)),xlim=c(-7,7),main="Density Plot for DE(0,1) with inverse CDF",
     lwd=3,col="lightblue")
abline(v=0,col="sienna1",lwd=3)
grid(14,14)
```




The above plot shows the density plot of a 10000 sample that was created for the $DE(0, 1)$ from $Uniform(0, 1)$ using the inverse CDF method. We know that the Laplace distribution(-double exponential distribution) is a function that is symmetric and can be though as having 2 exponential distributions seperated by the mean similar to the normal distribution.As we see from the plot our results seems reasonable and a very good approximation of the Laplace with mean zero and deviation 1.

## 2 Acceptance/Rejection Method

### Use Acceptance/Rejection Method to generate Normal Distribution

We are asked to generate noramal distribution $f(x) = \frac{1}{\sqrt{(2\pi)}}e(-\frac{x^2}{2}) \sim N(0,1)$ with majorizing denstity $g(x) = \frac{1}{2}e^| - x| \sim DE(0,1))$ using the acceptance rejection algorithm.

The steps or the algorithm are :

1. Generate random variable Y from our majorizing denstity function.In our case we use the previous funciton we built that samples using the inverse CDF method.
2. Generate random $U \sim (0,1)$ independent from Y
3. Next using the obtained Y from step 1 we calculate $f(Y)$ and $g(Y)$ using the normal distribution we asked to approximate and the majorizing density function given and we check the condition

$$if \quad U <= \frac{f(Y)}{cg(Y)} \quad then "accept \quad the \quad sample \quad Y" and \quad X = Y$$

$$else \quad go \quad back \quad to \quad step1 "reject"$$

We run the algorithm as many times needed to obtain a fixed sample n (2000-in our case).

In order to find the optimal value of parameter c we work as follows:

Let

$$h(x) = \frac{f(x)}{g(x)} = \frac{\frac{1}{\sqrt{2}\pi \quad e^{\frac{x2}{2}}}}{\frac{1}{2e^{|x|}}} =$$

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

now we maximize (I) solving $h'(x) = 0$ using to cases a)x>=0 and b) x<0

Setting $t = |x| - \frac{x^2}{2}$

for x>=0 we have

$$h'(x) = 0 => (\sqrt{\frac{2}{\pi}}e^t)' = 0 =>$$

$$\sqrt{\frac{2}{\pi}}e^t t' = 0 => \sqrt{\frac{2}{\pi}}e^{(x-\frac{x^2}{2})}(1-x) = 0$$

and in order for that expression to be zero we have $x = 1$

Simillarly for $x < 0$ we get $x = -1$

For $x = 1$ we have $h(1) = \sqrt{\frac{2}{\pi}}e^{(1-\frac{1}{2})} = \sqrt{\frac{2e}{\pi}}$

and for $x = -1$ we have $h(-1) = \sqrt{\frac{2}{\pi}}e^{(1-\frac{1}{2})} = \sqrt{\frac{2e}{\pi}}$

Thus the optimal value for $c = \sqrt{\frac{2e}{\pi}} = 1.32$

```r
set.seed(123456)
#our target function N(0,1)
f<-function(x){
  exp(-x^2/2)/sqrt(2*pi)
}
#majorizing density function laplace DE(0,1)
g<-function(x){
  exp(-abs(x))/2

}

counter<-0
#function to generate the normal
norm_gen<-function(n,c){
  #initialize a vector size n
  fvec<-rep(0,n)
  #counter for repeat

  #for loop to fill the vector
  for (i in 1:length(fvec)){
    #repeat in order to check or
    #condition for only one sample
    repeat{
      counter<<-counter+1
      #we take only one sample
      y<- rlaplace(1,0,1)
      u<-runif(1)

      if (u<=f(y)/(c*g(y))){

      break

      }
    }
    if (runif(1)<0.5){
      Z=y
    }
    else{Z=-y}
    #fill the vector
    fvec[i]<-Z
  }

  return(list("sample"=fvec,"iter"=counter))
}

optimal_c=f(1)/g(1)

res<-norm_gen(2000,optimal_c)

cat("The number of optimal_c is :",optimal_c,"\n",
    "The number of iterations to create the sample are :",res$iter)
```

```
## The number of optimal_c is : 1.315489
##  The number of iterations to create the sample are : 2621
```

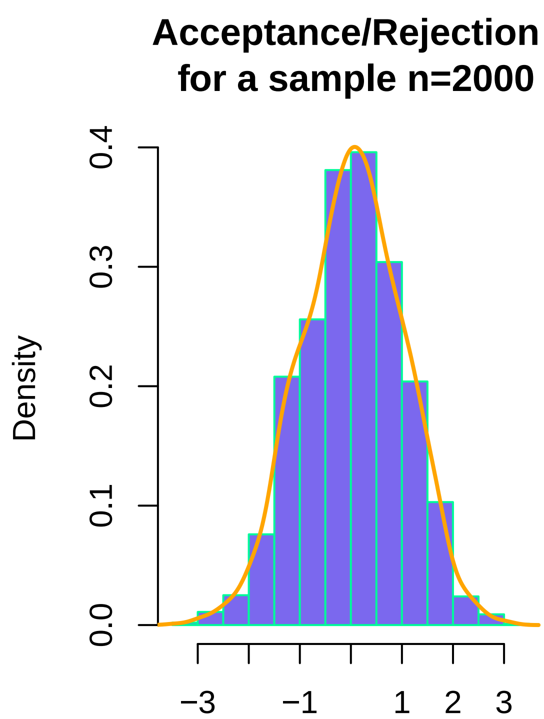
The average Expected Rate is given by the formula $1/c = 1/1.315489$ so the expected rejection rate is $1 - \frac{1}{1.315489} = 0.2398264 = 23.98264\%$ our average rejection rate is $ER = \frac{2000}{2621}$ and thus the average rejection rate is $1 - ER = 0.2369325 = 23.69325\%$.

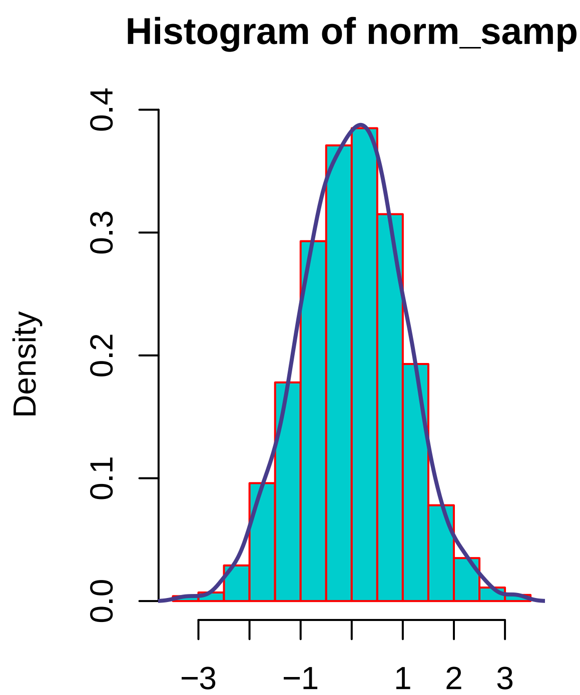
### Plot of the Acceptance/Rejection and rnorm

The above histogram pair plot shows the result of Acceptance/Rejection Method using $DE(0,1) = \frac{1}{2}exp(-|x|)$ for generating $N(0,1) = \frac{1}{\sqrt{2\pi}}e^{(-\frac{x^2}{2})}$ on the left and on theright using rnorm funnction of R for a sample of 2000.

```
norm_samp=rnorm(2000,0,1)

par(mfrow=c(1,2))
#plot a sample with the optimal c parameter x=1
hist(res$sample,col="mediumslateblue",main="Acceptance/Rejection \n for a sample n=2000",
     xlab ="" ,border="mediumspringgreen",prob=T)
lines(density(res$sample),col="orange",lwd=2)
hist(norm_samp,col="cyan3",border="red",xlab="",prob=T)
lines(density(norm_samp),col="slateblue4",lwd=2)
```




## Appendix

We need to prove that $sign(x - \mu) = sign(x - 0.5)$

for $x >= \mu$ we have

$x - \mu >= 0 <=> -\frac{1}{a}\ln(2 - 2y) >= 0 <=> 2 - 2y >= 1 <=> y >= \frac{1}{2}$

for $x < \mu$ we have

$x - \mu < 0 <=> \frac{1}{a}\ln(2y) < 0 <=> y < \frac{1}{2}$

so we can replace $sign(x - \mu)$ with $sign(x - 0.5)$