CS Computer Lab 3

Joshua Hudson, Carles Sans

14 February 2017

Question 1: Cluster sampling

In this section, we used the data file **population.csv**, containing a list of Swedish cities, along with their respective populations. The aim was to select twenty cities at random for an opinion pool, where the random sampling was without replacement and with probabilities proportional to the populations of each city.

In order to do this, we normalised the population proportions so that these were represented as sub-intervals of the interval [0,1]. A uniform random number was then generated; whichever sub-interval this number fell in, the corresponding city was selected. The function is detailed below:

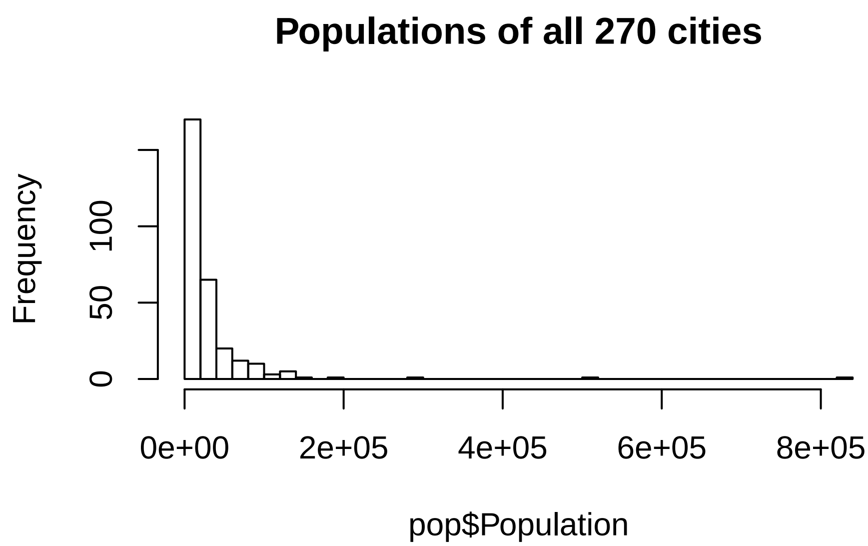
```
selectone <- function (data) {
  interval <- c()
  interval[1] <-
    data$Population[1]/sum(data$Population)
  for (i in 2:dim(data)[1]) {
  interval[i] <- interval[i-1] +
    data$Population[i]/sum(data$Population)
  }
  rand <- runif(1, 0, 1)
  pick <- length(which(interval < rand)) + 1
  return(data[pick, ])
}
```

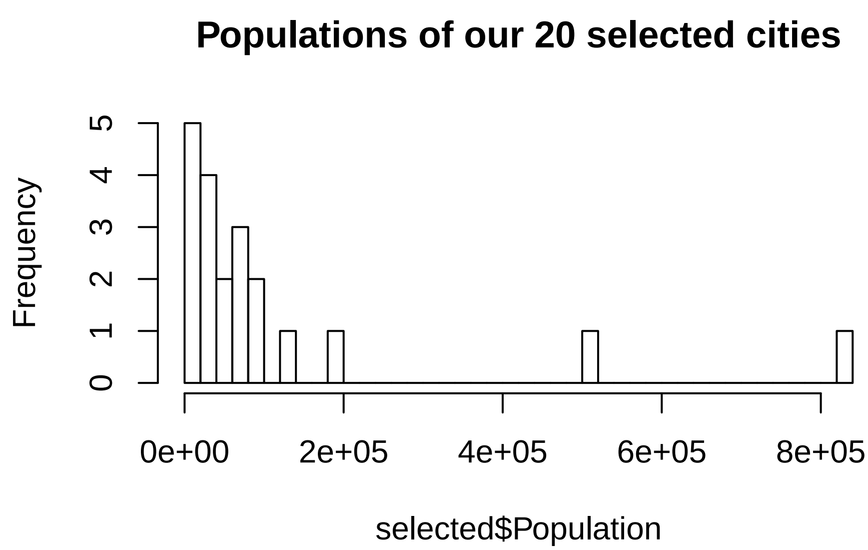
We then applied the function to the list of cities 20 times, each time removing the selected one from the list for the next iteration.

The following cities were selected:

```
##    Municipality Population
## 1     Karlskoga      29742
## 2    Ulricehamn      22753
## 3        Kalmar      62388
## 4     Jönköping     126331
## 5       Ljungby      27410
## 6          Täby      63014
## 7    Trelleborg      41891
## 8     Stockholm     829417
## 9         Luleå      73950
## 10      Uppsala     194751
## 11     Fagersta      12249
## 12     Göteborg     507330
## 13    Sundsvall      95533
## 14        Gävle      94352
## 15        Piteå      40860
## 16     Ovanåker      11530
## 17 Smedjebacken      10758
## 18  Katrineholm      32303
## 19        Nybro      19576
## 20    Hallsberg      15235
```

Generally our program selected large cities: the mean population of our selected ones was 115,569, much larger than the sample mean of 32,209. This is also shown by comparing the histograms of the two.





These histograms confirm what we observed initially; our selection process favours large cities. This makes sense when considering the approach taken here, however the people from these selected cities may be unrepresentative of Swedish people as a whole, as clearly an overwhelming proportion of people live in lots of small cities.

Question 2: Different distributions

In this section we consider the double exponential distribution, with pdf:

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

The first goal was to generate random numbers from the $DE(0,1)$ distribution from $Unif(0,1)$ using the inverse CDF method. For this we need to derive to the inverse CDF of DE(0,1), starting from the pdf:

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

Therefore the CDF is:

$$F(x) = \int_{-\infty}^{x} \frac{1}{2} e^{-|u|} du \tag{3}$$

For $x \geq 0$:

$$\begin{aligned}
F(x) &= \int_{0}^{x} \frac{1}{2} e^{-|u|} du + \int_{-\infty}^{0} \frac{1}{2} e^{-|u|} du && (4) \\
&= \int_{0}^{x} \frac{1}{2} e^{-u} du + \int_{-\infty}^{0} \frac{1}{2} e^{u} du && (5) \\
&= \frac{1}{2} \left[e^{-u}\right]_{0}^{x} + \frac{1}{2} \left[e^{u}\right]_{-\infty}^{0} && (6) \\
&= 1 - \frac{1}{2} e^{-x} && (7)
\end{aligned}$$

For $x < 0$ (and therefore u<0):

$$\begin{aligned}
F(x) &= \int_{-\infty}^{x} \frac{1}{2} e^{-|u|} du && (8) \\
&= \frac{1}{2} \left[e^{u}\right]_{-\infty}^{x} && (9) \\
&= \frac{1}{2} e^{x} && (10)
\end{aligned}$$

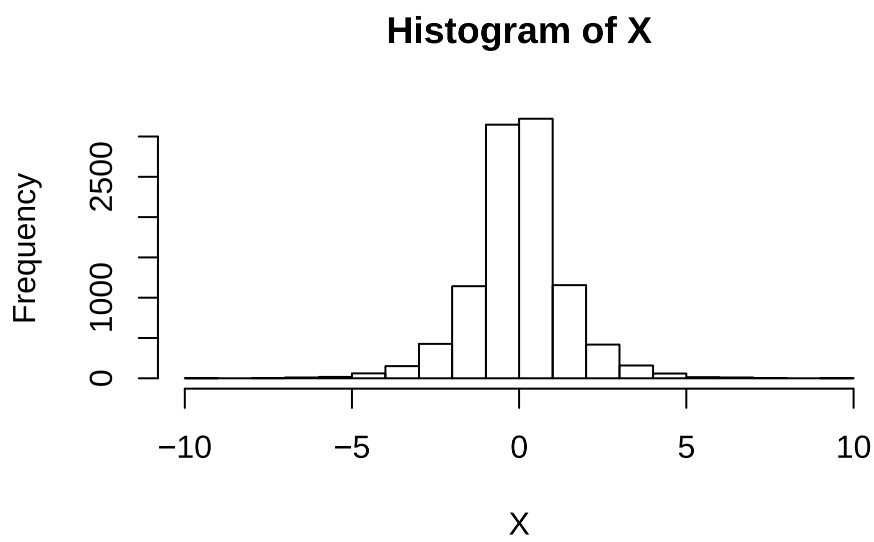
To get the inverse CDF, we set:

$$y = \begin{cases} 1 - \frac{1}{2} e^{-x}, & \text{for } x \geq 0 \\ \frac{1}{2} e^{x}, & \text{for } x < 0 \end{cases} \tag{11}$$

Rearranging gives us the following inverse CDF:

$$F^{-1}(y) = \begin{cases} -log(2(1-y)), & \text{for } y \geq \frac{1}{2} \\ log(2y), & \text{for } y < \frac{1}{2} \end{cases} \tag{12}$$

We then wrote a function `laplace()` to generate random numbers from the $DE(0,1)$ distribution, by generating a uniform r.v. and taking the inverse CDF of this. We generated 10000 such random numbers; the results are shown in the histogram below.



This histogram looks reasonable as it ressembles the pdf of the Laplace distribution (on a much larger scale of course).

The next goal was to use the Acceptance/rejection method to generate from the Normal $N(0, 1)$ distribution (target density f_X). We were given that the previously considered $DE(0, 1)$ distribution could be used as a majorising density (f_Y). All we needed was to find a constant c such that, for all x:

$$cf_Y(x) \geq f_X(x) \tag{13}$$

Subbing in the respective pdfs:

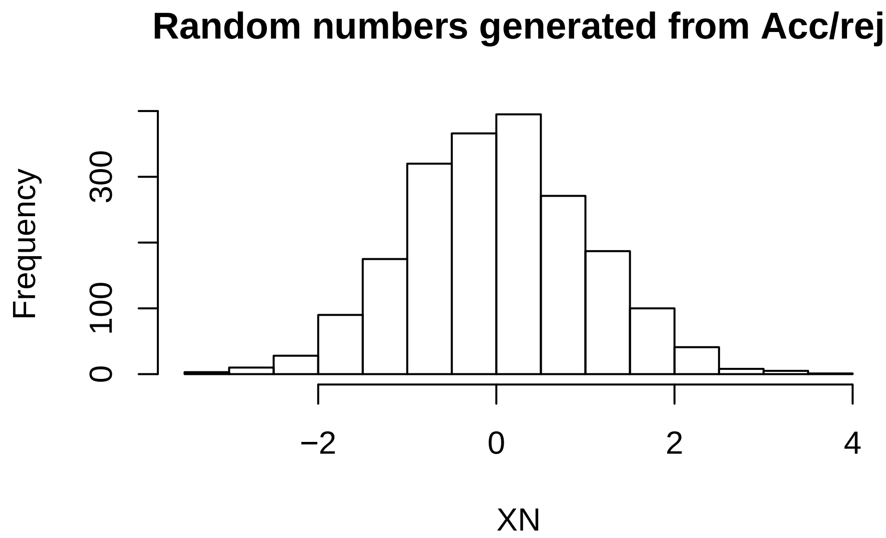
$$\frac{c}{2}e^{-|x|} \geq \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2} \tag{14}$$

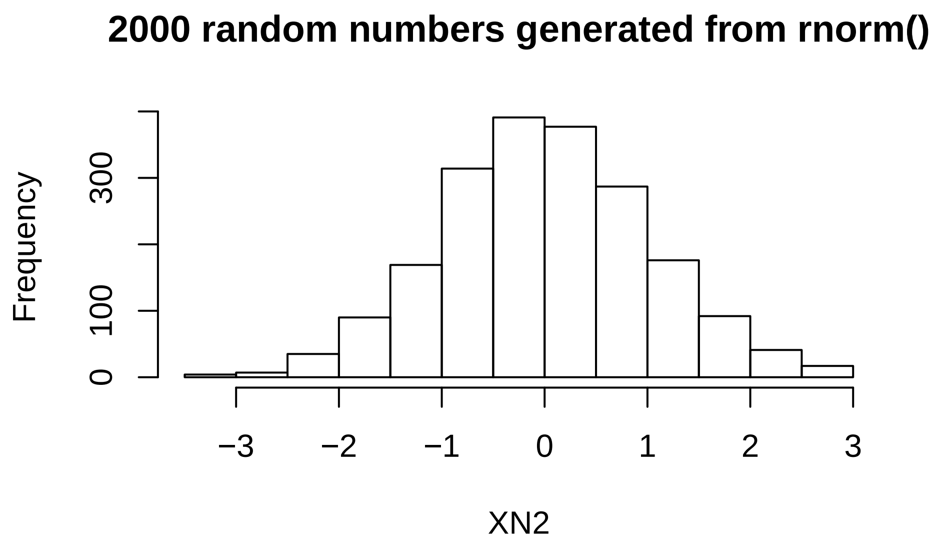
Rearranging gives us:

$$c \geq \frac{2}{\sqrt{2\pi}}e^{|x|-\frac{1}{2}x^2} \tag{15}$$

The right hand side is maximised for $x = \pm 1$ so the limiting case is $c = \frac{2}{\sqrt{2\pi}}e^{0.5}$. Using this constant c, we applied the Acceptance/rejection method to generate 2000 random numbers.

The average rejection was calculated, we obtained 0.2472714. The expected rejection rate was also found, as it is the probability that our acceptance condition is not validated, ie. $ER = 1 - 1/c$. This expected rejection rate was therefore 0.2398265, so very close to the actual rate. We also plotted the histogram of our results. Below it we have the histogram of 2000 normal random numbers using the in-built `rnorm()` function.





The plots are very similar in shape, it looks like our Acceptance/rejection method was pretty effective at simulating draws fron the $N(0, 1)$ distribution.

Appendix

```
knitr::opts_chunk$set(echo = FALSE, message = FALSE, fig.width = 5, fig.asp = 0.66, fig.show = "asis", 
#1.1
setwd("~/Semester2/CS/Lab3")
pop <- read.csv2(paste0(getwd(), "/population.csv"), stringsAsFactors = FALSE)

selectone <- function (data) {
  interval <- c()
  interval[1] <-
    data$Population[1]/sum(data$Population)
  for (i in 2:dim(data)[1]) {
  interval[i] <- interval[i-1] +
    data$Population[i]/sum(data$Population)
  }
  rand <- runif(1, 0, 1)
  pick <- length(which(interval < rand)) + 1
  return(data[pick, ])
}

set.seed(123456)
selected <- data.frame(Municipality=vector(length = 20), Population=vector(length = 20))
data <- pop
for (i in 1:20) {
  selected[i, ] <- selectone(data = data)
  pick <- which(data$Municipality == selected$Municipality[i])
  data <- data[ -pick, ]
}
print(selected)
meanselect <- round(mean(selected$Population))
meanpop <- round(mean(pop$Population))
hist(pop$Population, breaks = 30, main="Populations of all 270 cities")
hist(selected$Population, breaks = 30, main = "Populations of our 20 selected cities")
#2.1
rlaplace <- function(n, mu, alpha) {
  U <- runif(n, min=0, max=1)
  X <- mu - 1/alpha*(sign(U-0.5)*log(1-2*abs(U-0.5)))
  return(X)
}

set.seed(12345)
X <- rlaplace(10000, 0, 1)
hist(X)
c <- 2/sqrt(2*pi) * exp(1/2)

normgen <- function(c) {
  fY <- function(x) {
  y <- 1/2*exp(-abs(x))
   return(y)
  }

  rej <- 0
  repeat {
```

```r
    Y <- rlaplace(1, 0, 1)
    U <- runif(1, 0, 1)
    cond <-  dnorm(Y, 0, 1)/(c*fY(x=Y))
    if (U <= cond) {
      X <- Y
      break
    }
    else {
      rej <- rej+1
    }

  }
  return(list(X=X, rej=rej))
}

XN <- c()
rej <- c()
for (i in 1:2000) {
  X <- normgen(c=c)
  XN[i] <- X$X
  rej[i] <- X$rej
}


R <- sum(rej)/(sum(rej)+2000)
ER <- 1- 1/c

hist(XN, main="Random numbers generated from Acc/rej")
XN2 <- rnorm(2000, 0, 1)
hist(XN2, main="2000 random numbers generated from rnorm()")
```

Collaborations

Methodology and results were shared and discussed with members of Group 6, Chih-Yuan Lin and Sarah Walid. Alsaadi.