Report lab03

Thijs Quast (thiqu264) and Anubhav Dikshit (anudi287)

30-1-2019

Contents

Collaborations

For this lab our classmates from group 16 explained the intuition behind question 2, and helped with the analytical derivation to solve question 2.

Question 1 - Cluster sampling

1.1

```
options(scipen = 999)
```

```
# Import data
set.seed(12345)
population <- read.csv2("population.csv")
population2 <- population
population2$Municipality <- as.character(population2$Municipality)
```

1.2

```
# Generate probabilities
population2$Probability <- population2$Population/sum(population2$Population)
population2$Cumulativeprobability <- cumsum(population2$Probability)
```

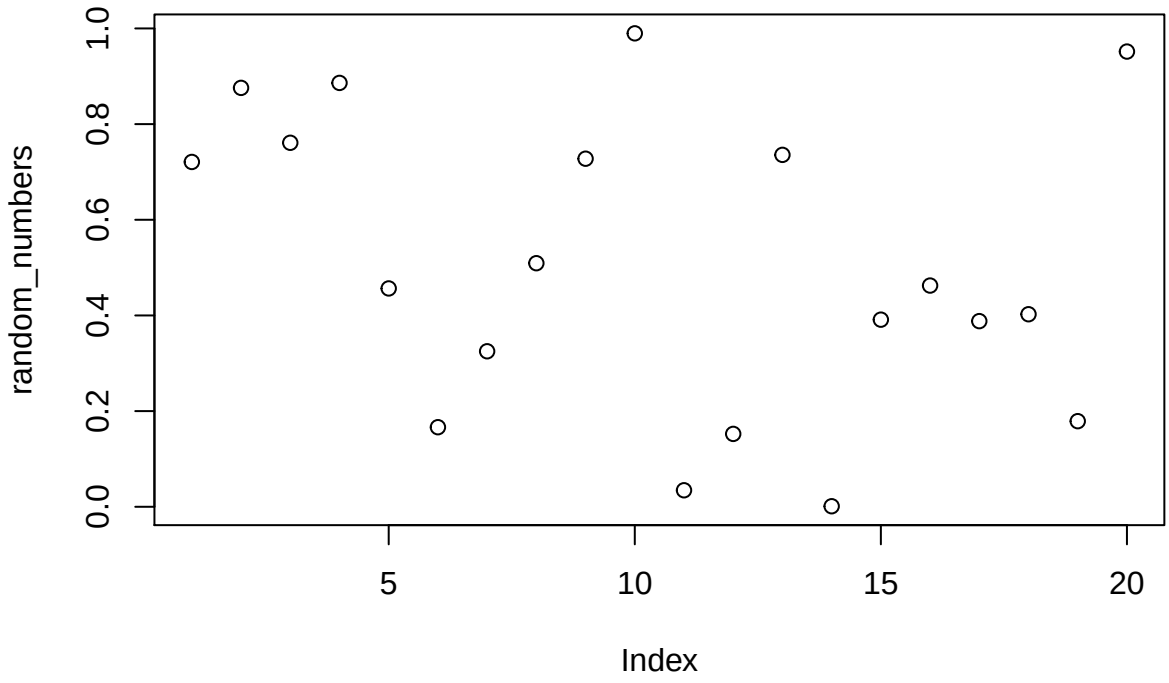
```
# Generate random numbers
#random_numbers <- c()
#x0 <- 1

#for (each in 1:20){
  #a <- 7^5
  #c <- 0
  #m <- 2^31
  #random_number <- (a*x0+c)%%m
  #random_numbers[each] <- random_number
  #x0 <- random_number
#}

#random_numbers <- random_numbers/m
```

```
random_numbers <- runif(20, 0, 1)
```

```
plot(random_numbers)
```



1.3

```
chosen_city <- data.frame(Population=numeric(), Probability= numeric(), Cumulativeprobability = numeric
city_names <- c()

for (i in 1:20){
  generated_probability <- random_numbers[i]
  subdata <- population2[population2$Cumulativeprobability >= generated_probability, ]
  chosen_city[i,] <- subdata[1, -1]
  city_names[i] <- as.character(subdata[1,1])
  drop_index <- (nrow(population2) - nrow(subdata)) + 1
  population2 <- population2[-drop_index,]
  population2$Probability <- population2$Population/sum(population2$Population)
  population2$Cumulativeprobability <- cumsum(population2$Probability)
}

city_names <- as.data.frame(city_names)
chosen_city <- cbind(city_names, chosen_city)
```

1.4

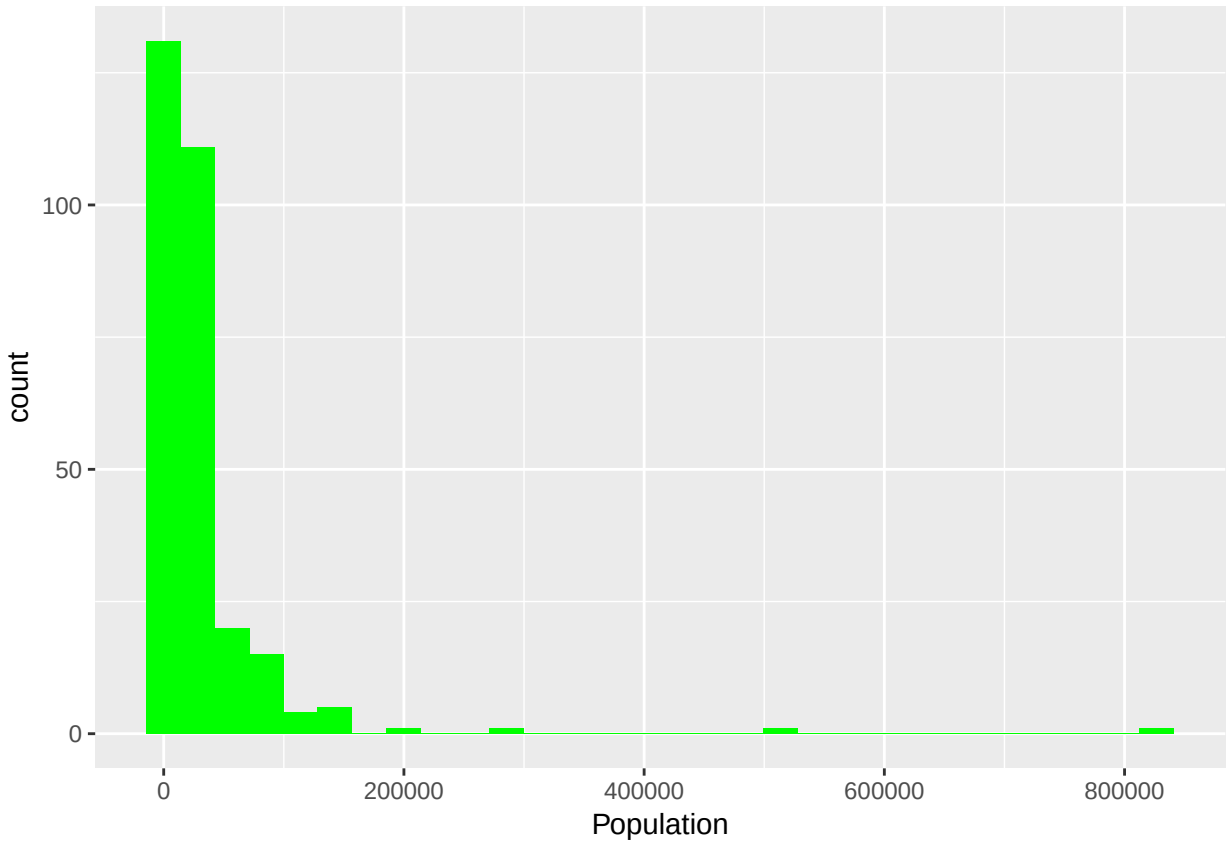
```
chosen_city$city_names
```

```
##  [1] Skövde       Älvdalen     Arvika       Gävle        Helsingborg
##  [6] Stockholm    Växjö        Åstorp       Ulricehamn   Luleå
## [11] Huddinge     Uppsala      Öckerö       Botkyrka     Kristianstad
## [16] Skurup       Klippan      Lund         Nyköping     Skellefteå
## 20 Levels: Älvdalen Arvika Åstorp Botkyrka Gävle Helsingborg ... Växjö
```

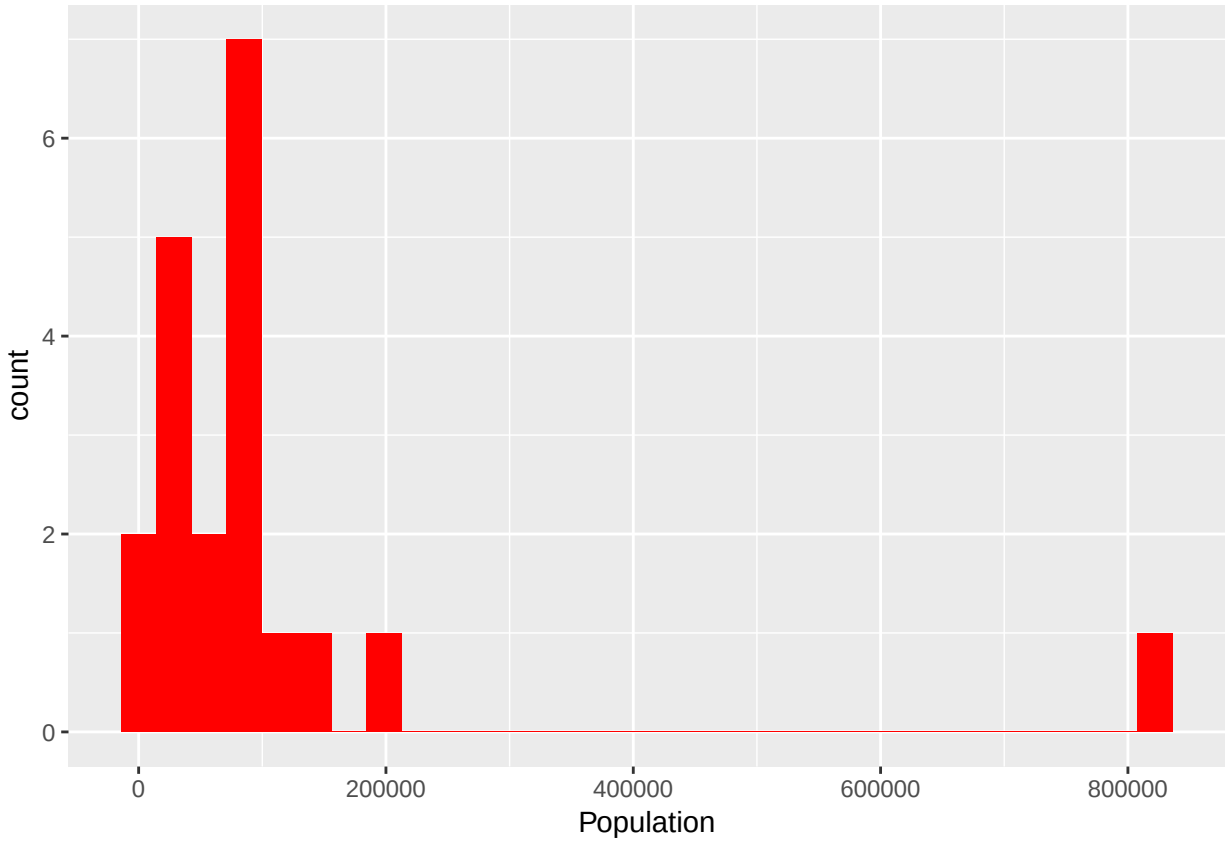
As expected cities with large populations like Stockholm and Uppsala are selected. This is due that the probability of these cities being selected is larger because there population is larger.

1.5

```
library(ggplot2)
plot <- ggplot(data = population, aes(Population)) + geom_histogram(bins = 30, fill = "green")
plot
```




```
plot2 <- ggplot(data=chosen_city, aes(Population)) + geom_histogram(bins = 30, fill = "red")
plot2
```



The distribution of the populations of the sampled cities roughly seems to follow the same distribution as the full dataset.

Question 2 - Different distributions

2.1

source: https://math.stackexchange.com/questions/2021342/how-is-this-inverse-function-calculated-laplace-distribution

Double exponential (Laplace):

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

The CDF is given by:

$$F(x) = \int_{-\infty}^{x} f(x)dx$$

$$F(x) = \int_{-\infty}^{x} \frac{\alpha}{2} e^{-\alpha(x-\mu)} dx, \qquad (if \quad x > \mu)$$

$$= 1 - \int_{x}^{\infty} \frac{\alpha}{2} e^{-\alpha(x-\mu)} dx$$

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

$$F(x) = \int_{-\infty}^{x} \frac{\alpha}{2}e^{\alpha(x-\mu)}dx, \qquad (if \quad x \leq \mu)$$

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

Inverse of CDF

$$For \quad x > \mu, \ we \ got \ F(x) = 1 - \frac{1}{2}e^{-\alpha(x-\mu)}$$

$$y = 1 - \frac{1}{2}e^{-\alpha(x-\mu)}$$

$$\frac{ln(2-2y) - \alpha\mu}{-\alpha} = x$$

$$For \ U \sim U(0,1), \qquad \frac{ln(2-2U) - \alpha\mu}{-\alpha} = X$$

$$For \quad x \leq \mu, \ we \ got \ F(x) = \frac{1}{2}e^{\alpha(x-\mu)}$$

$$y = \frac{1}{2}e^{\alpha(x-\mu)}$$

$$\frac{ln(2y)}{\alpha} + \mu = x$$

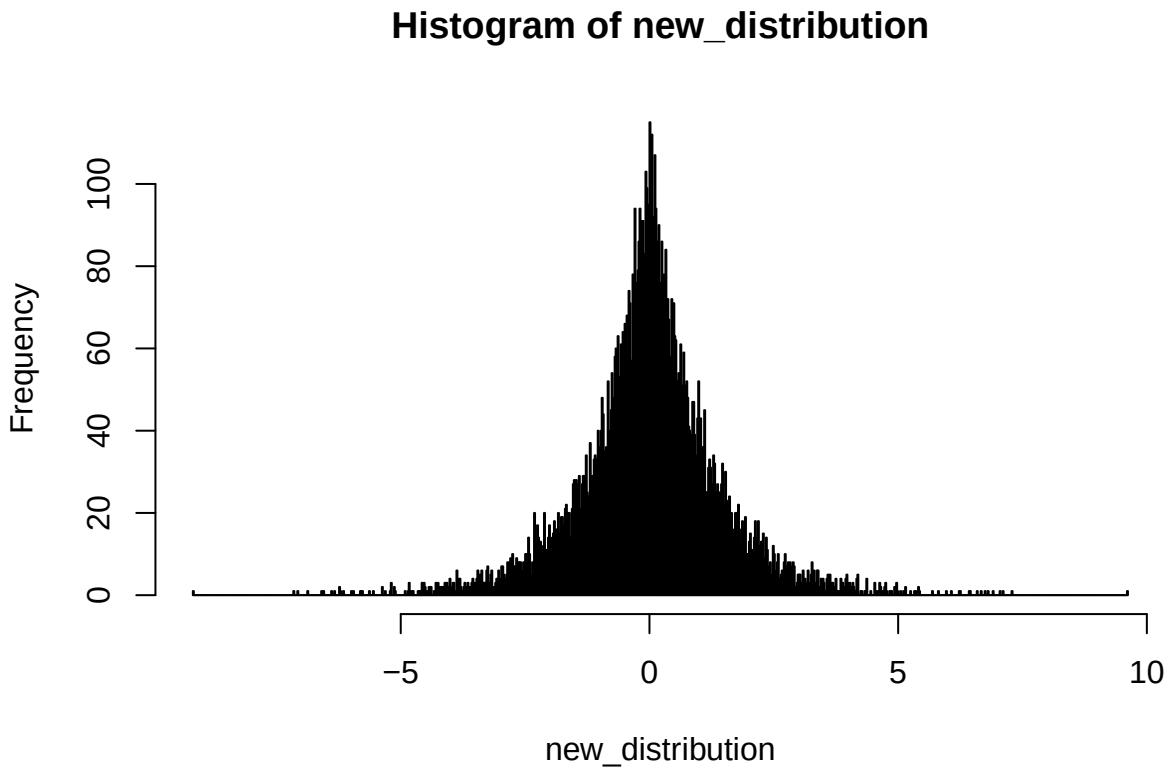
$$For \ U \sim U(0,1), \qquad \frac{ln(2U)}{\alpha} + \mu = X$$

```
uniform <- runif(10000, 0, 1)
```

```
de_dist <- function(u, mu, a){
  de_distribution <- c()

    for (i in 1:length(u)){
      if (u[i] > 0.5){
        de_distribution[i] <- (log(2-2*u[i])-a*mu)/(-a)
      }
      else {
        de_distribution[i] <- (log(2*u[i])/a)+mu
      }
    }
  return(de_distribution)
}
```

```
new_distribution <- de_dist(uniform, mu=0, a=1)
hist(new_distribution, breaks = 1000)
```



2.2

$$\frac{f(y)}{g(y)} \leq c$$

Thus if we maximize the ratio f(y)/g(y) then that would be the value of c. This can be done by using parital derivate of the fraction.

$$Majorizing\ density\ F_Y(y) \sim DE(0,1) = \frac{1}{2}e^{-|x|}$$

$$Target\ density\ F_X(y) \sim N(0,1) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}$$

$$C \geq \frac{\sqrt{2}}{\sqrt{\pi}}e^{-\frac{x^2}{2}+|y|}$$

Differenitating with respect to x we get that the expression is maximum at x=1, thus C is:

$$\sqrt{\frac{2}{\pi}}e^{\frac{-1}{2}}$$

$$= \frac{\sqrt{2}}{C\sqrt{\pi}} \cdot e^{-\frac{y^2}{2}+|y|}$$

$$= e^{-\frac{y^2}{2}+|y|+\frac{1}{2}}$$

Thus C is

$$C = \sqrt{(\frac{2 * e^1}{\pi})}$$

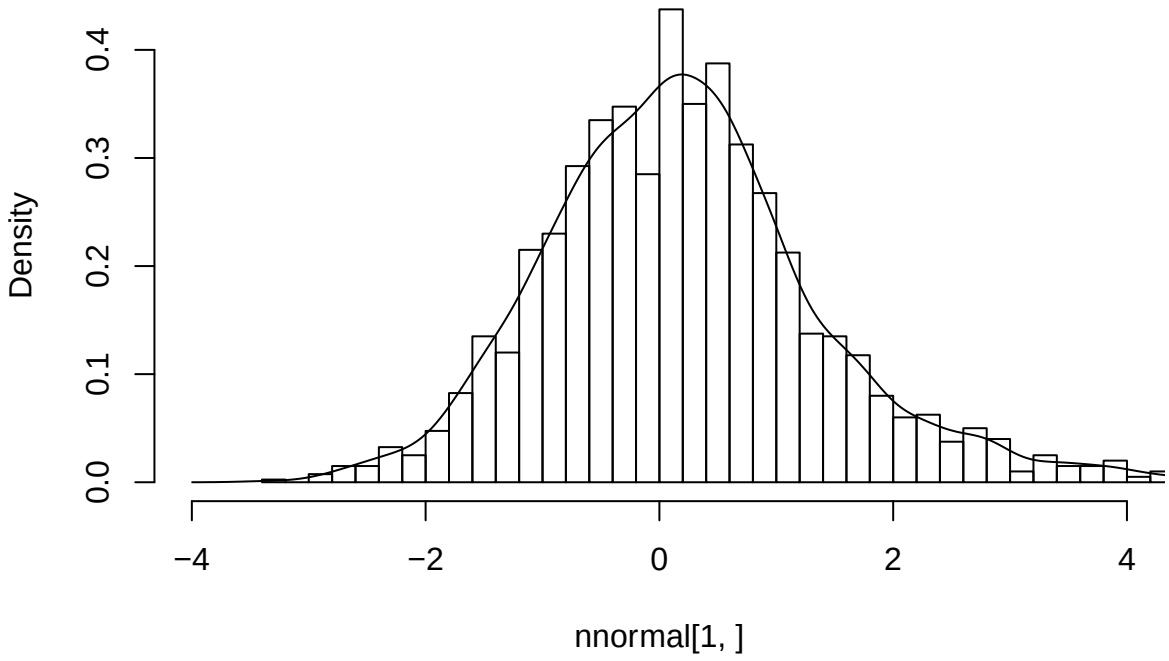
source: Introduction to Probability, Statistics, and Random Processes - Hossein Pishro-Nik

```
generate_n <- function(c){
  x <- NA
  num.reject <- 0
    while (is.na(x)){
      u <- runif(1, 0, 1)
      y <- de_dist(u, 0, 1)
      U <- runif(1)
        if (y>0 && U <= sqrt(2/pi)/c*exp(-(y^2)/2)+y){
          x <- y
        } else if (y<=0 && U<=sqrt(2/pi)/c*exp(-(y^2)/2-y)){
            x <- y
              } else {
                num.reject <- num.reject + 1
            }
    }
  c(x,num.reject)
}
```

```
c <- sqrt(2*exp(1)/pi)
```

```
set.seed(12345)
nnormal <- sapply(rep(c,2000), generate_n)
```

```
hist(nnormal[1,], breaks = 50, freq = FALSE, xlim = c(-4,4),
     main = "Normal distribution generated by Accept/Reject method")
lines(density(nnormal[1,]))
```




```
average_rejection <- sum(nnormal[2,])/(ncol(nnormal)+sum(nnormal[2,]))
average_rejection
```

```
## [1] 0.154334
```

```
expected_rejection <- 1-(1/c)
expected_rejection
```

```
## [1] 0.2398265
```

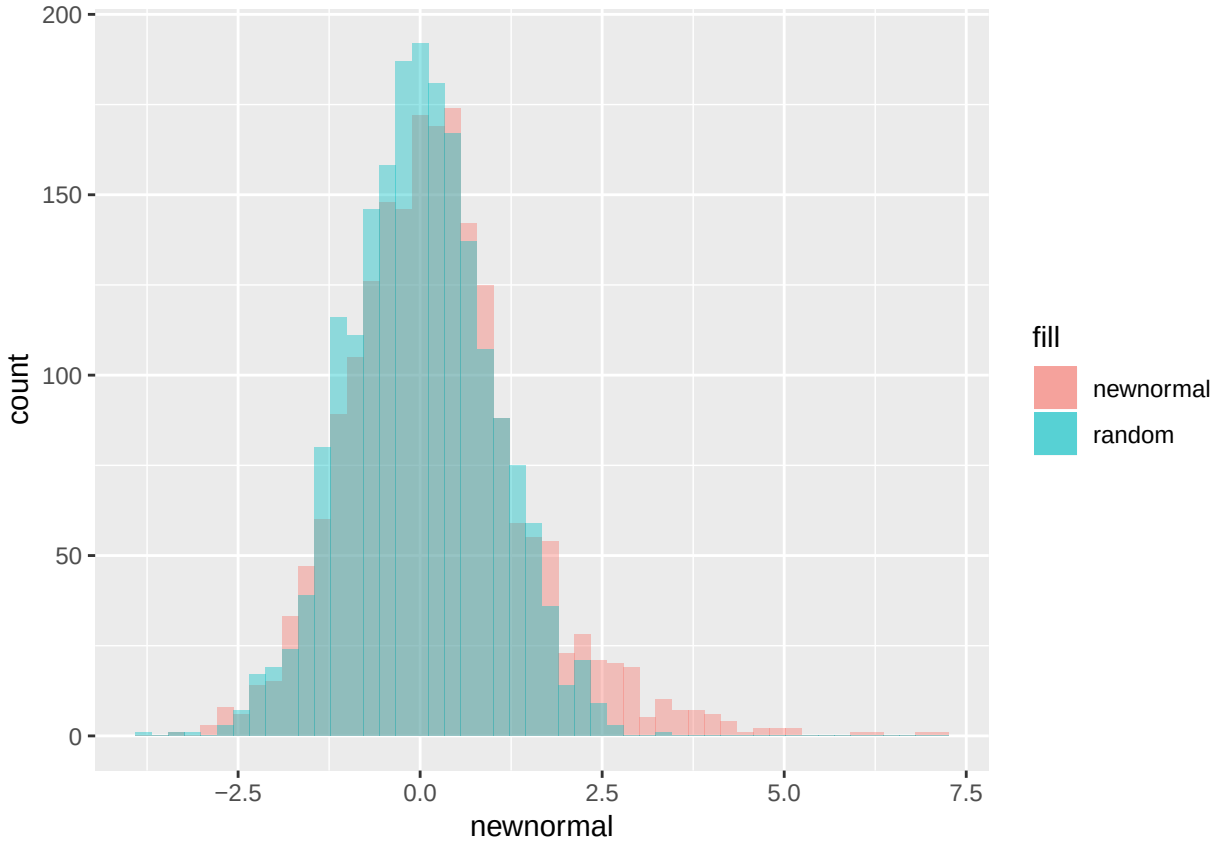
```
rejection_difference <- expected_rejection - average_rejection
rejection_difference
```

```
## [1] 0.08549251
```

The difference between the expected rejection rate and average rejection rate is around 8.5 percentage points, which is quite high.

```
newnormal <- nnormal[1,]
random <- rnorm(2000, 0, 1)
df <- as.data.frame(cbind(newnormal, random))
```

```
library(ggplot2)
histogram1 <- ggplot(df, aes(newnormal, fill = "newnormal")) + geom_histogram(alpha = 0.4, bins = 50) +
  geom_histogram(aes(random, fill = "random"), alpha = 0.4, bins = 50)
histogram1
```



Both the samples distribution using accept/reject method and the actual distribution are close to each other. This means that the sampling method applied was decent.