2352-Statistical Computing Homework 1

Firstname Lastname

Simulating the Central Limit Theorem

The Central Limit Theorem

If we take repeated independent samples of size n from a population of interest and use the sample mean (denoted $\bar{x}$) to estimate the population mean (denoted μ), then these sample means will be (approximately, if sample size is large enough) normally distributed with mean μ and standard deviation (which we actually call the *standard error* of $\bar{x}$) $\frac{\sigma}{\sqrt{n}}$ where σ is the population standard deviation (which we can estimate using a sample standard deviation if we don't know it).

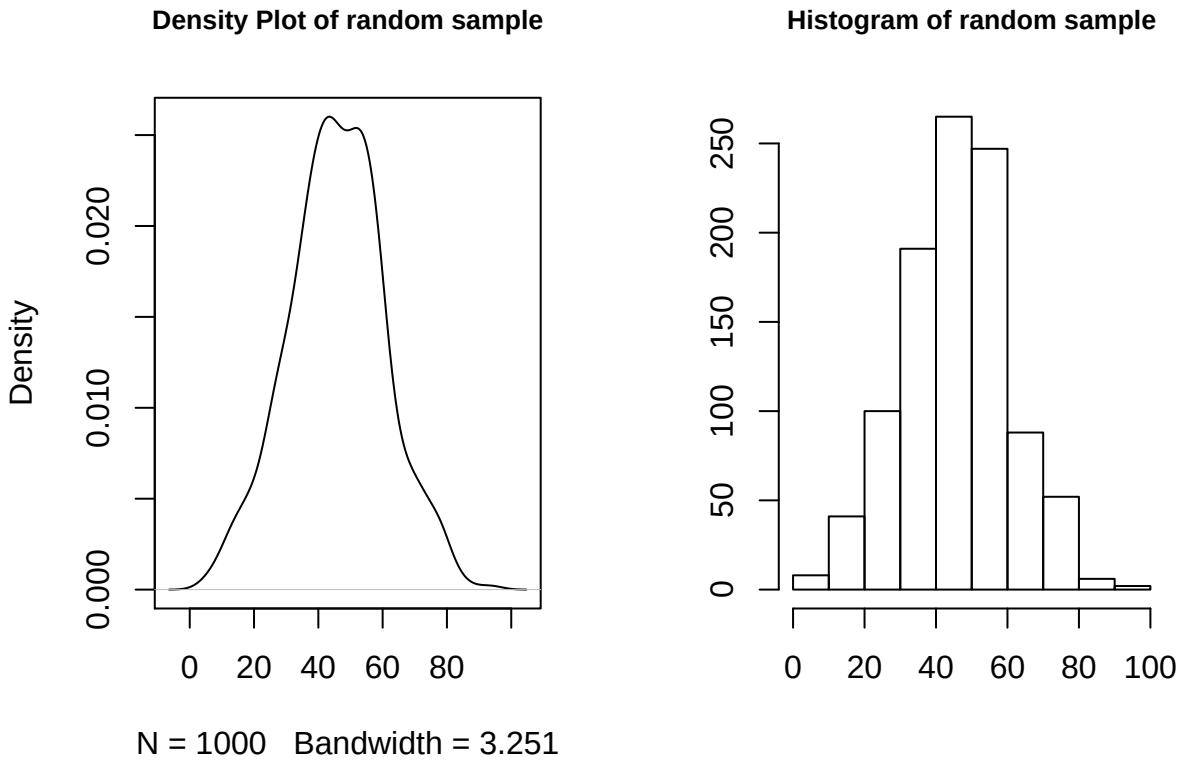
The mean of the sampling distribution of the mean

Suppose we are trying to estimate the mean commuting time for graduate students for a population of 1000 grduate students. Let's say the population mean is 45 minutes. If we randomly select a sample of 50 graduate students we might observe a sample mean ($\bar{x}$) equal to 35 minutes. Suppose we randomly select another 20 graduate students from the popluation and find the mean to be 50 minutes. We repeat this process over and over again to have distribution of means for all the possible samples of 50 graduate students. This is called a **sampling distribution of the mean**. From the **central limit theorem** the sampling distribution of the mean has a mean of μ and standard deviation $\frac{\sigma}{\sqrt{n}}$.

We can simulate this in R. You can sample from a variety distribtions in base R, given the distribution parameters. For example, we can sample from a binomial distribution using **rbinom**, a poisson distribution using **rpois**, and a normal distribution using **rnorm**. The R function rnorm() takes in parameters of the distribution: n (the number of observations), mean (the mean of the sample), and sd (the standard deviation) to randomly draw from the distribution.

We will explore the example from above through simulation, assuming that the population of graduate students is normally distributed.

```
# Set seed allows for reproducible results and for the document to produce
# the exact sample numbers and samples
set.seed(12345)

# A sample of 400 observations from a normal distribution with mean = 2.25
# and sd = 2.
dat1 <- rnorm(1000, 45, 15)

# We can visualize our data using the R functions plot(), density(), and
# hist()

# This creates matrix of m rows by n columns to be able to plot more than
# one plot
par(mfrow = c(1, 2))

# Creates a density plot of the data
plot(density(dat1), main = "Density Plot of random sample", cex.main = 0.8)

# Creates a histogram of the data
hist(dat1, main = "Histogram of random sample", ylab = "", xlab = "", cex.main = 0.8)
```

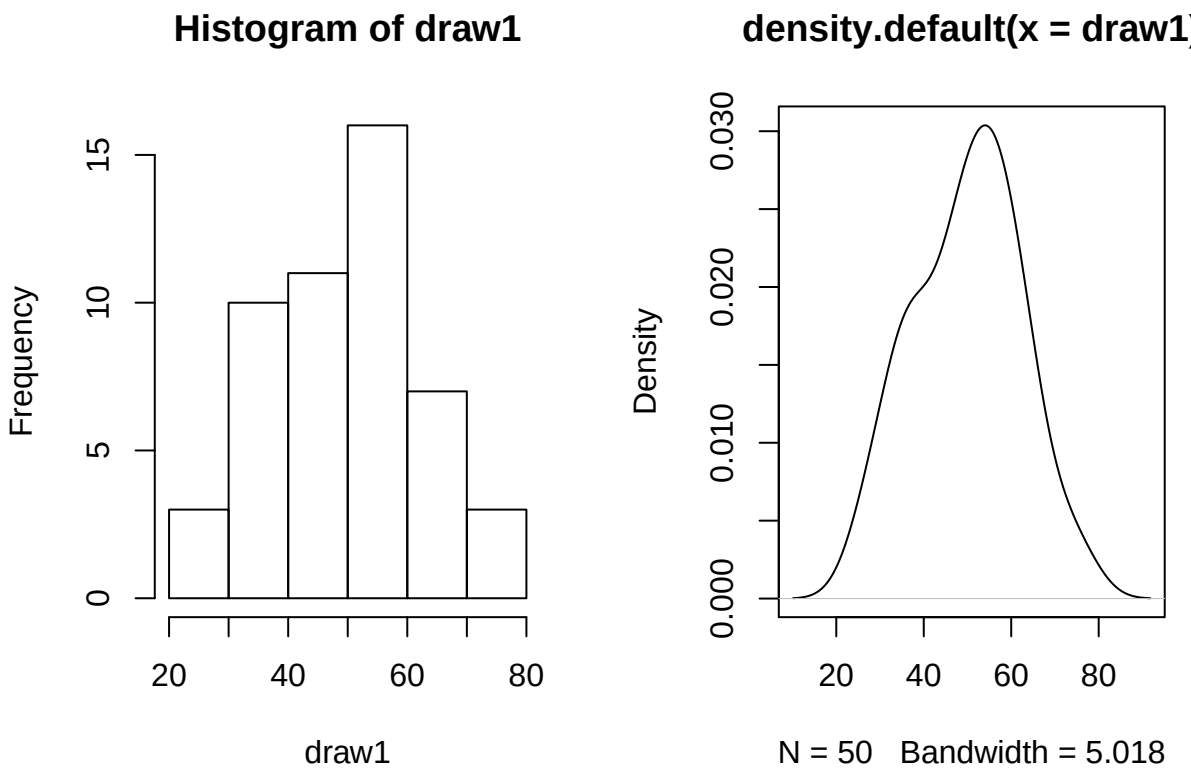


We see that while we sample from a normal distribution, our sample might not necessarily look normal.

```
# Draw 50 observations and take the mean of the draw
N <- 50  # Storing the number of observations
# Randomly draw from the above simulated data N times and stores it in draw1
draw1 <- sample(dat1, N)
mean(draw1)  # Computes the mean of the draw
```

```
## [1] 49.92169
```

```
par(mfrow = c(1, 2))
hist(draw1)  # Plots a histogram of the draw
plot(density(draw1))  # Creates a density plot of the draw
```

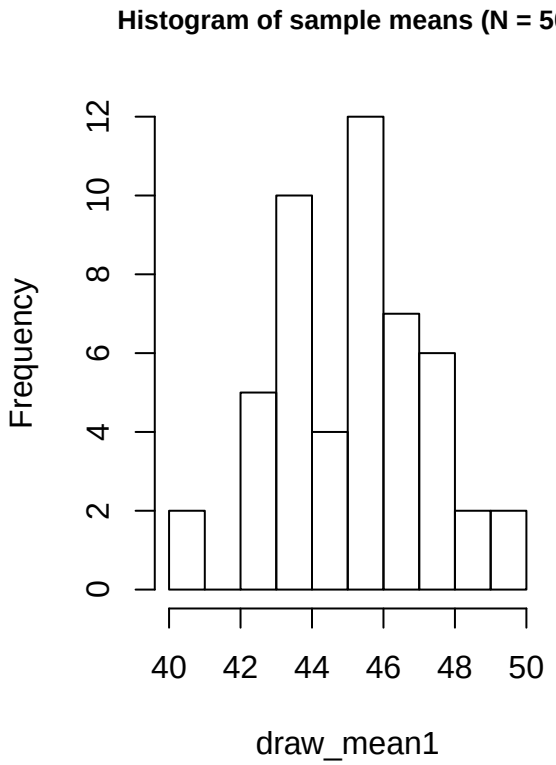


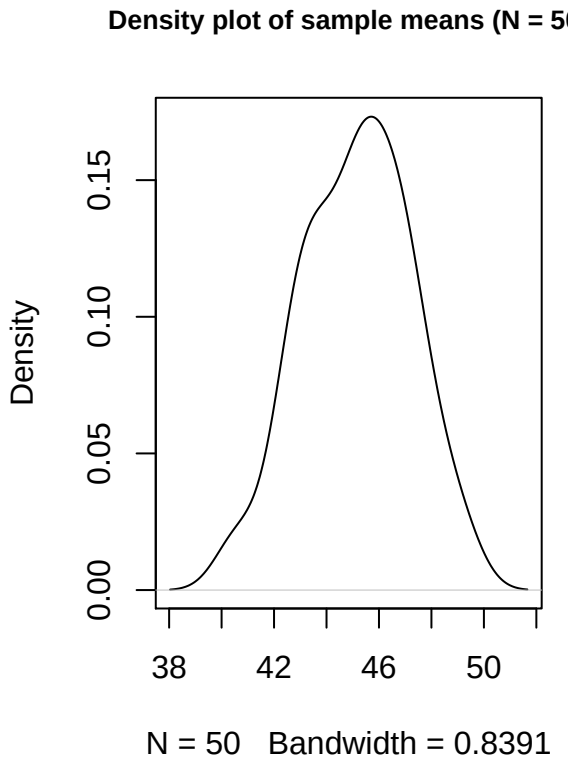
Now let's create a sampling distribution of the mean of size 50.

```
N <- 50  # Set the size of sample
ITER <- 50  # Set the number of iterations

# Create empty vector of length ITER to store results
draw_mean1 <- rep(NA, length = ITER)

# Creates a loop to draw 20 observations from the simulated data (dat1),
# takes the mean of the 20 observations, and stores the mean and repeates
# this, over the loop, ITER times (in this case 50 times)
for (i in 1:ITER) {
    draw_mean1[i] <- mean(sample(dat1, N))
}

mean(draw_mean1)
```

```
## [1] 45.14804
```

```
par(mfrow = c(1, 2))
# Plots a histogram of the sample means
hist(draw_mean1, main = "Histogram of sample means (N = 50)", cex.main = 0.8)
# Creates a density plot of the sample means
plot(density(draw_mean1), main = "Density plot of sample means (N = 50)", cex.main = 0.8)
```



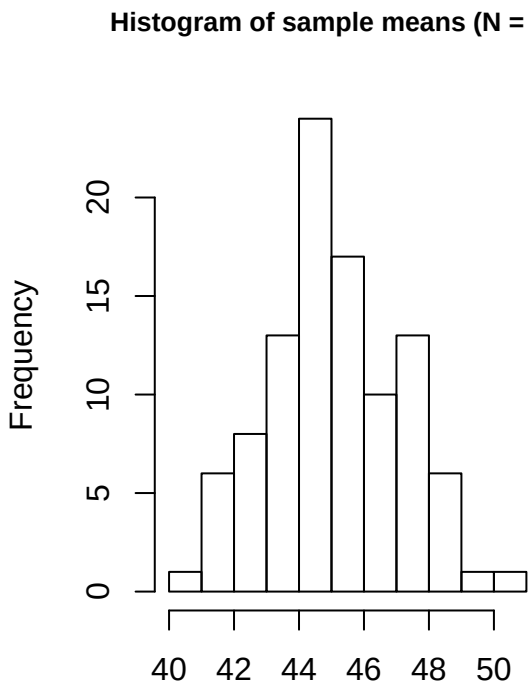


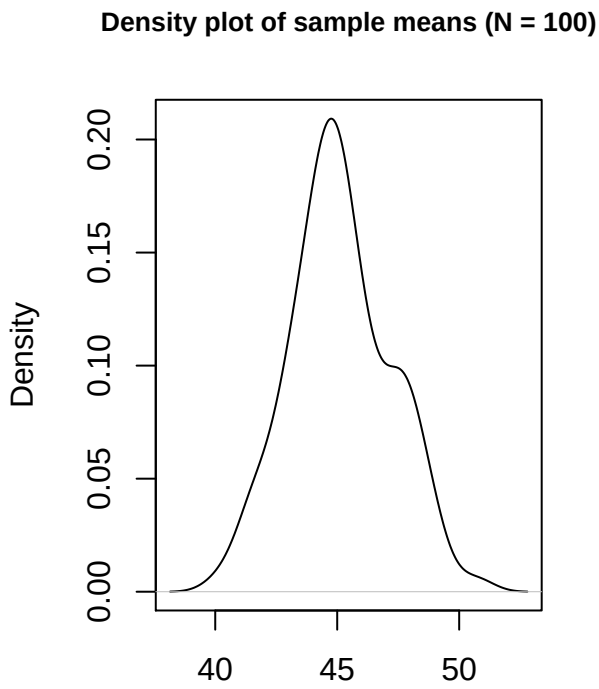
Next we can increase the number of iterations to 100, 1000, and 4000 and see what happens to the sampling distribution as we increase the number of iterations.

```
N <- 50  # Set the size of sample
ITER <- 100  # Set the number of iterations

# Create empty vector of length ITER to store results
draw_mean2 <- rep(NA, length = ITER)

# Creates a loop to draw 20 observations from the simulated data (dat1),
# takes the mean of the 20 observations, and stores the mean and repeates
# this, over the loop, ITER times (in this case 100 times)
for (i in 1:ITER) {
    draw_mean2[i] <- mean(sample(dat1, N))
}

mean(draw_mean2)
```

```
## [1] 45.12611
```

```
par(mfrow = c(1, 2))
# Plots a histogram of the sample means
hist(draw_mean2, main = "Histogram of sample means (N = 100)", cex.main = 0.8)
# Creates a density plot of the sample means
plot(density(draw_mean2), main = "Density plot of sample means (N = 100)", cex.main = 0.8)
```

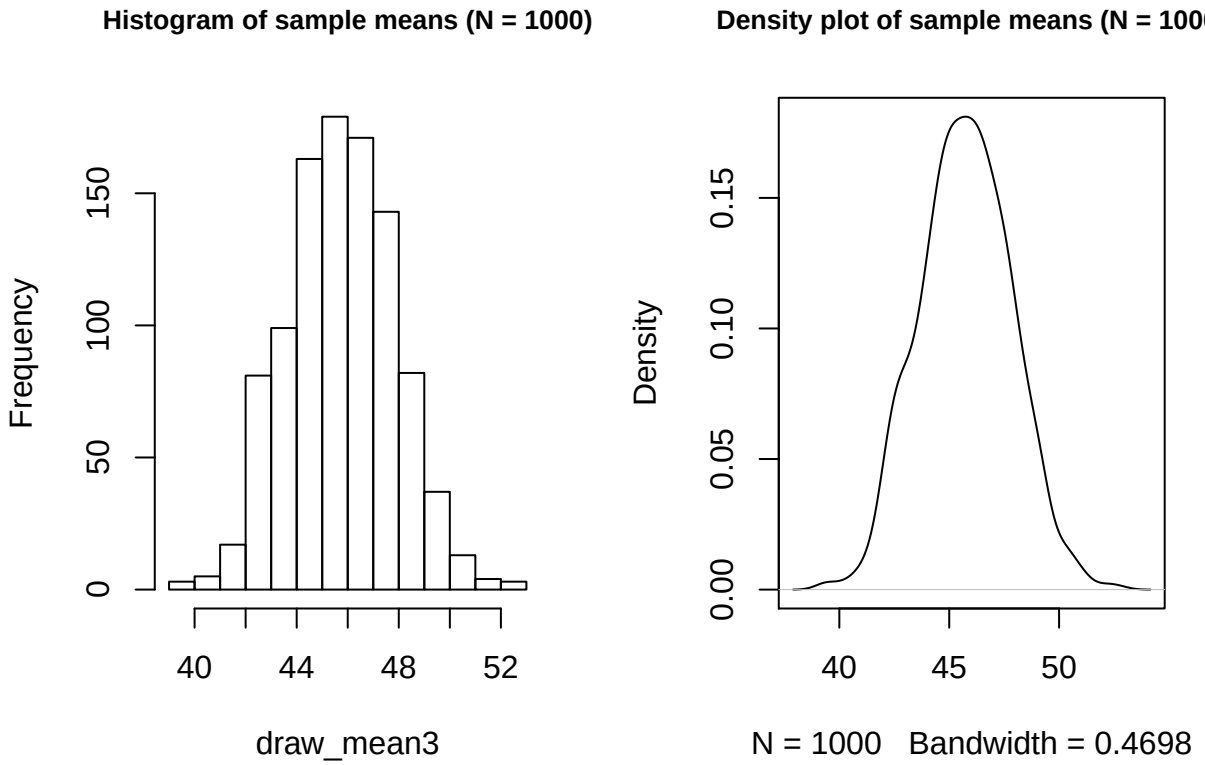





```r
N <- 50  # Set the size of sample
ITER <- 1000  # Set the number of iterations

# Create empty vector of length ITER to store results
draw_mean3 <- rep(NA, length = ITER)

# Creates a loop to draw 20 observations from the simulated data (dat1),
# takes the mean of the 20 observations, and stores the mean and repeates
# this, over the loop, ITER times (in this case 1000 times)
for (i in 1:ITER) {
    draw_mean3[i] <- mean(sample(dat1, N))
}

mean(draw_mean3)
```

```
## [1] 45.74402
```

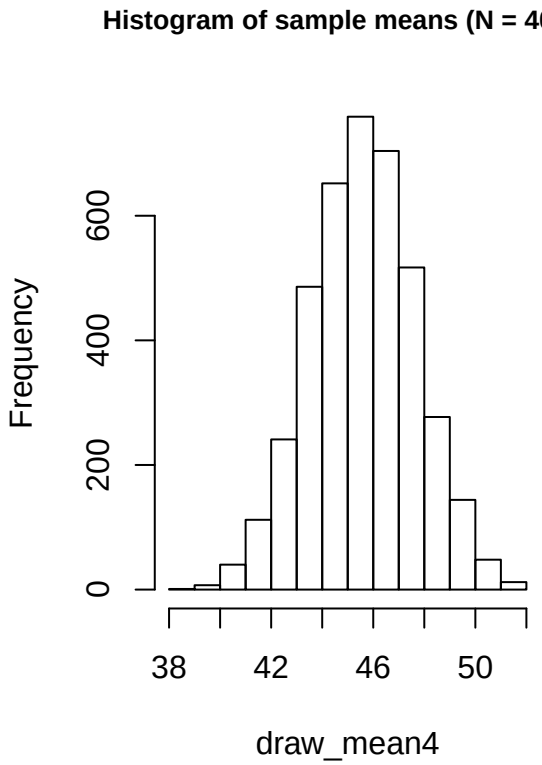
```r
par(mfrow = c(1, 2))
# Plots a histogram of the sample means
hist(draw_mean3, main = "Histogram of sample means (N = 1000)", cex.main = 0.8)
plot(density(draw_mean3), main = "Density plot of sample means (N = 1000)",
    cex.main = 0.8)  # Creates a density plot of the sample means
```

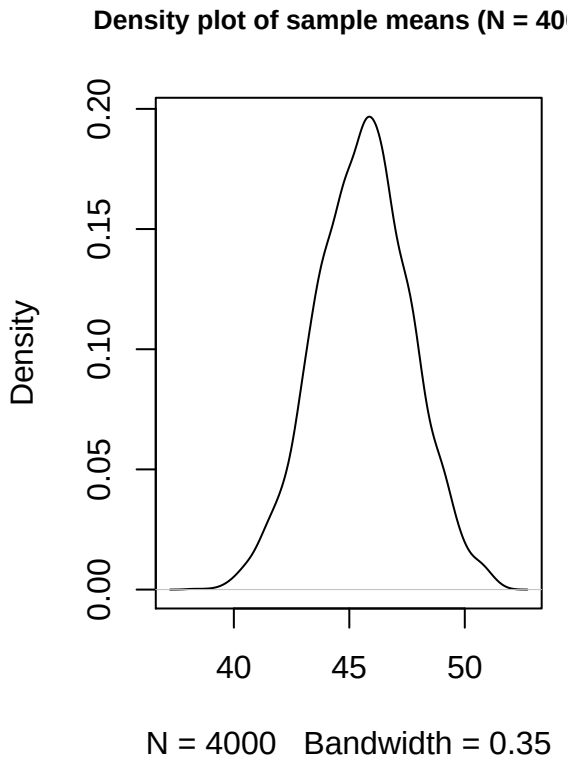



```r
N <- 50  # Set the size of sample
ITER <- 4000  # Set the number of iterations

# Create empty vector of length ITER to store results
draw_mean4 <- rep(NA, length = ITER)

# Creates a loop to draw 20 observations from the original sample (dat1),
# takes the mean of the 20 observations, and stores the mean and iterates
# over the loop ITER times (in this case 4000 times)
for (i in 1:ITER) {
    draw_mean4[i] <- mean(sample(dat1, N))
}

mean(draw_mean4)
```

```
## [1] 45.6065
```

```r
par(mfrow = c(1, 2))
hist(draw_mean4, main = "Histogram of sample means (N = 4000)", cex.main = 0.8)  # Plots a histogram of
plot(density(draw_mean4), main = "Density plot of sample means (N = 4000)",
    cex.main = 0.8)  # Creates a density plot of the sample means
```





From this example we see that as we increase N, the sampling distribution of the mean approaches a normal distribution. This is true for any population distribution (uniform, skewed, etc.) as long as sample size is large enough.

Homework (Due after 1st week of classes)

Note: this assignment should be submitted as a knitted PDF file. Please include the course number, your last name, and first name (for example 2352_LASTNAME_FIRSTNAME). You may delete all notes above the homework heading before submitting.

Uniform Distribution

A Uniform distribution has an equal probability of an event occuring given an interval from [a,b].

1. Simulate a sample from a uniform distribution (use **runif**)of size 500 from an interval between 0 and 1. Store and plot your sample. Note: You can type runif in the *Help* tab or type ?runif into the Console to look at the documentation of the function

```
## Insert your code here
```

2. Draw a sample of size 50 from the simulated data and find the mean. Create loop to creates a sampling distribution of the mean with 50, 100, 1000, and 4000 sample means. Plot the histogram of each sampling distribution. Note: Use the above notes as a reference and some guidance.

```
# Sampling distribution of the mean from a uniform distribution of size 50

N <- 50 # Set the size of sample
ITER <- 50 # Set the number of iterations for the size of the sampling distribution

draw_mean_uni1 <- # inset code here

# Sample from the simulated data (from a uniform distribution stored above)
```

```r
  # find the mean, and store the value ITER times
for (i in 1:ITER){
  # Insert code here to store sample means
}

# Show your results visually
# Insert code here
```

```r
# Sampling distribution of the mean from a uniform distribution of size 100

N <- 50 # Set the size of sample
ITER <- 100
# Insert code here
```

```r
# Sampling distribution of the mean from a uniform distribution of size 1000

N <- 50 # Set the size of sample
ITER <- 1000
# Insert code here
```

```r
# Sampling distribution of the mean from a uniform distribution of size 4000

N <- 50 # Set the size of sample
ITER <- 4000

# Insert code here
```