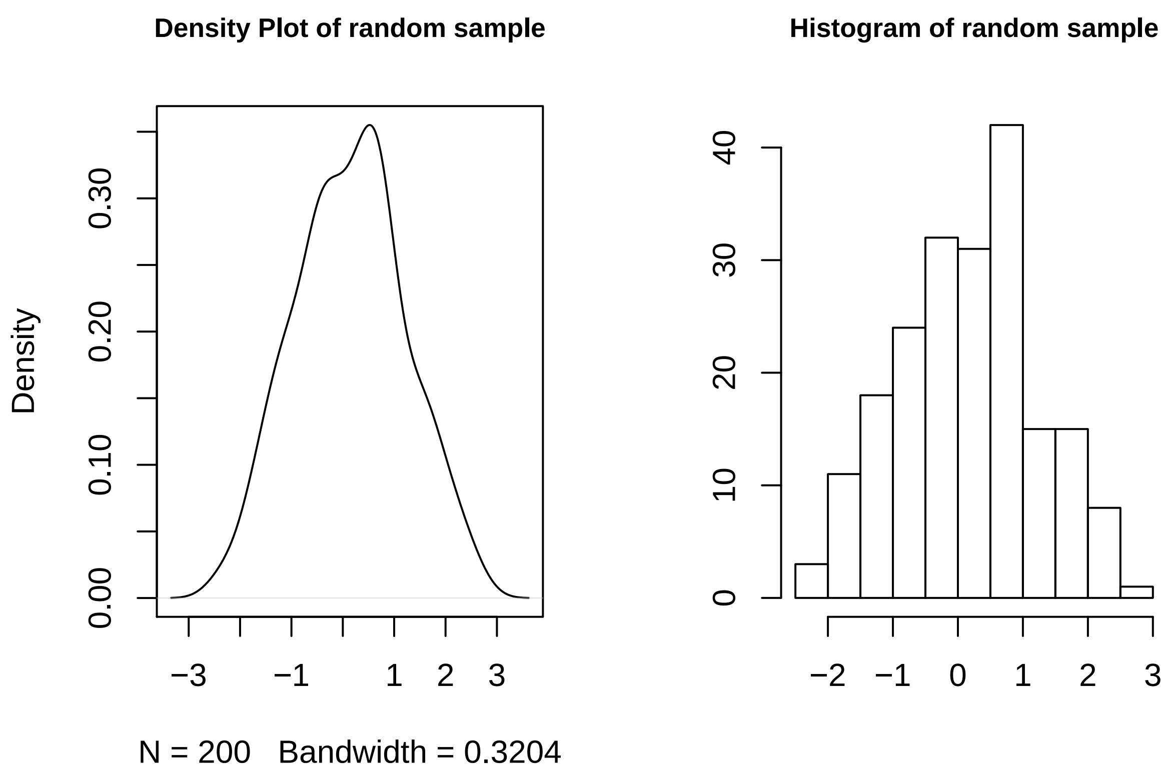
2352-Statistical Computing Assignment

Simulating the Central Limit Theorem

We can sample from a variety of distributions in R. To sample from a normal distribution we would use **rnorm**. 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. The example below creates a sample of size = 200 with mean = 0 and standard deviation = 1.

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

# A sample of 200 observations from a normal distribution with mean = 0 and
# sd = 1.
dat1 <- rnorm(200, 0, 1)

# We can visualize our data using the R functions plot(), density(), and
# hist()
par(mfrow = c(1, 2))  #This creates matrix of m rows by n columns to be able to plot more than one plot
plot(density(dat1), main = "Density Plot of random sample", cex.main = 0.8)  # Creates a density plot o
hist(dat1, main = "Histogram of random sample", ylab = "", xlab = "", cex.main = 0.8)  # Creates a hist
```



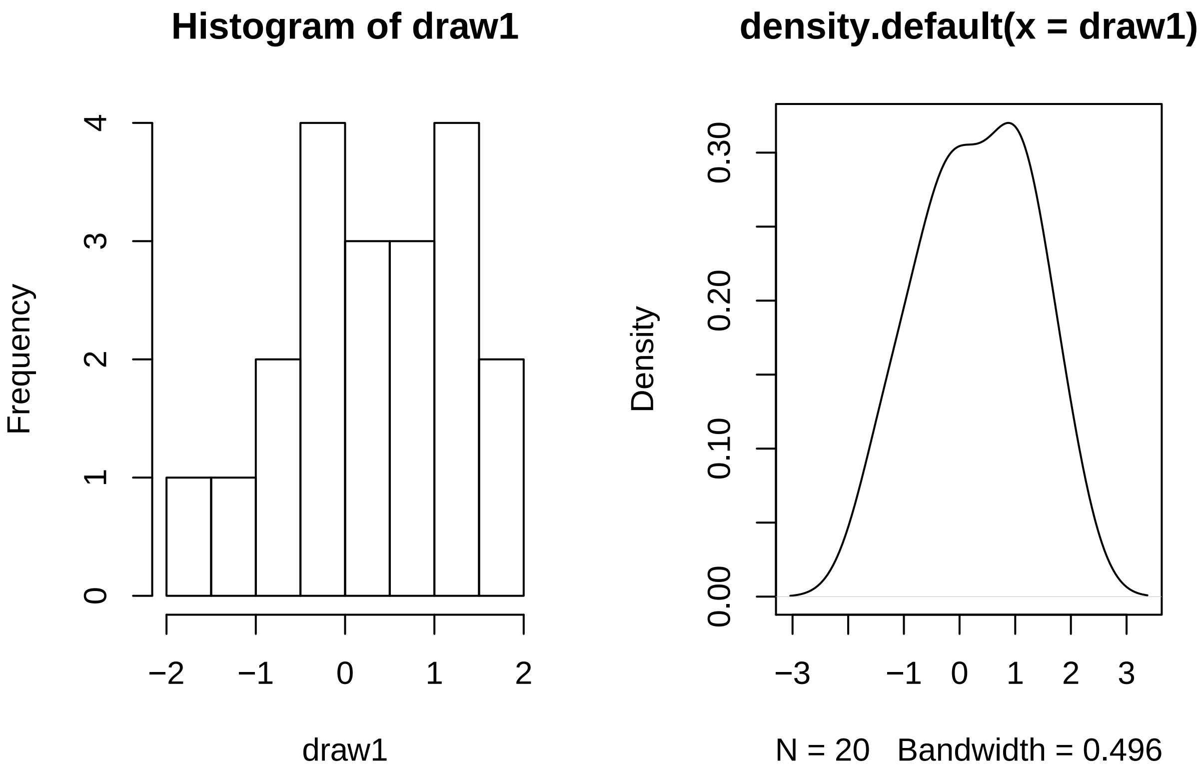

```
# Draw 20 observations and take the mean of the draw

N <- 20  # Storing the number of observations
draw1 <- sample(dat1, N)  # Randomly drawing from the above simulated data N number of times and storin
mean(draw1)  # Computes the mean of the draw
```

```
## [1] 0.3039873
```

```r
var(draw1)/sqrt(1)  #
```

```
## [1] 1.006714
```

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

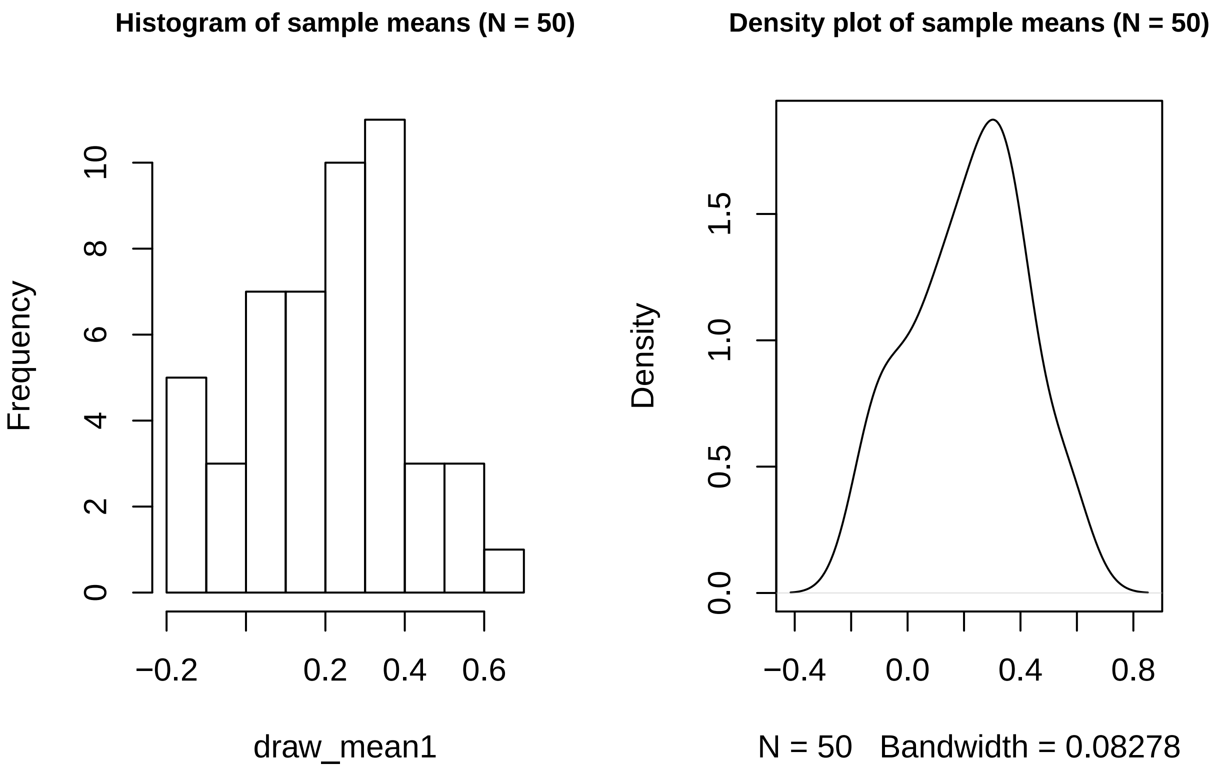


Now let's create a sampling distribution of the mean of size 50.

```r
N <- 20  # 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 original sample (dat1),
# takes the mean of the 20 observations, and stores the mean and iterates
# 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] 0.2171342
```

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

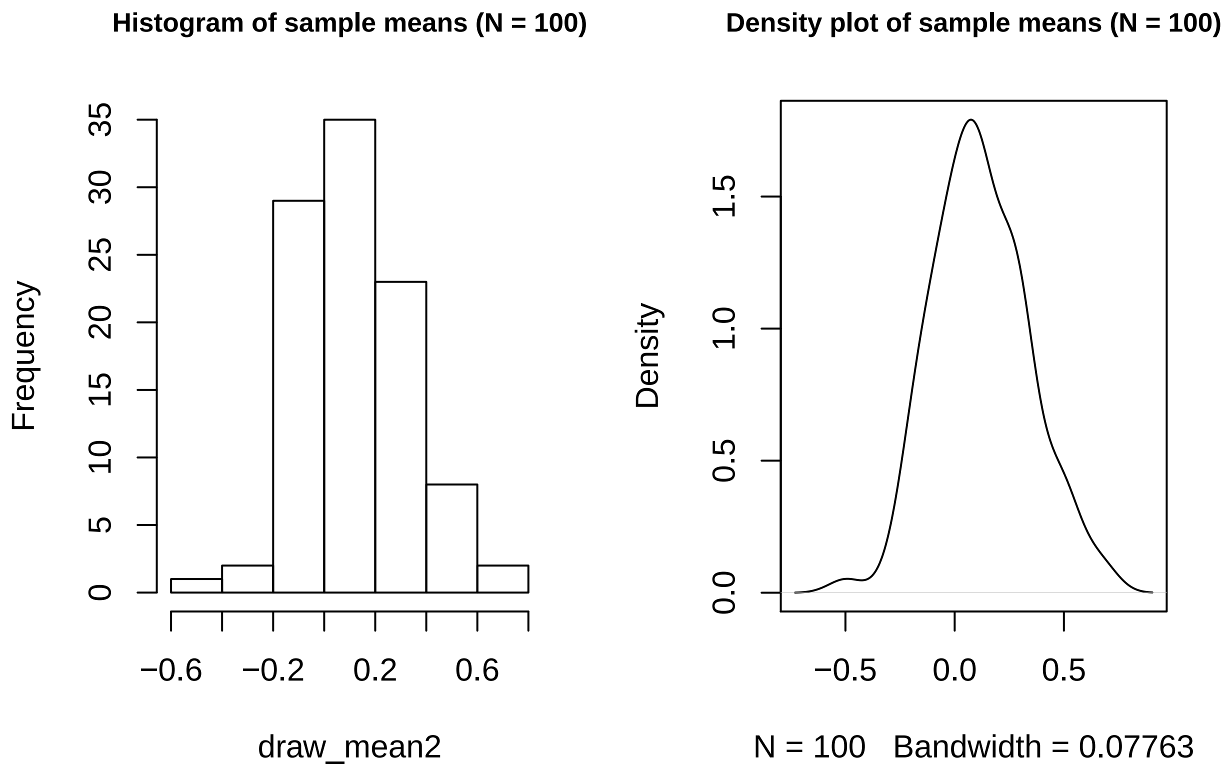


Next we can increase the number of iterations to 100, 1000, and 4000 and see what happens to the sampling distribution as we increase the number of iterations.

```
N <- 20  # 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 original sample (dat1),
# takes the mean of the 20 observations, and stores the mean and iterates
# 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] 0.1198831
```

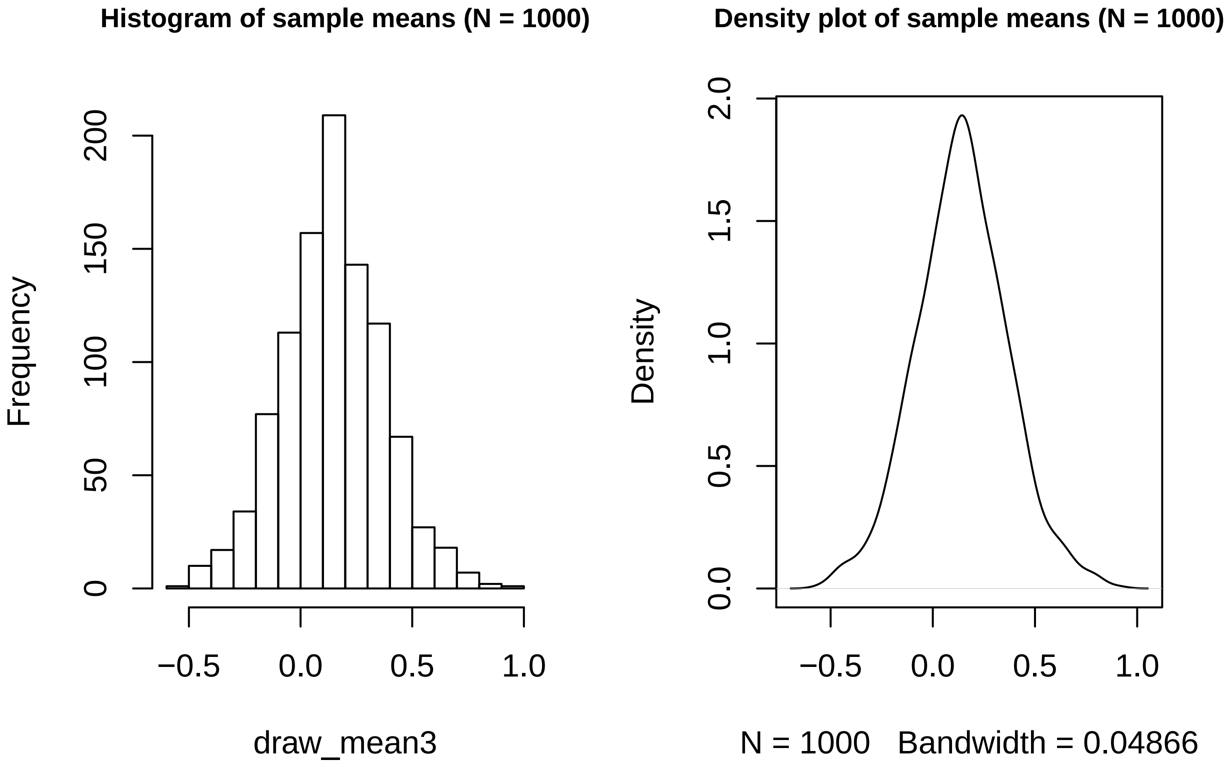
```
par(mfrow = c(1, 2))
hist(draw_mean2, main = "Histogram of sample means (N = 100)", cex.main = 0.8)  # Plots a histogram of
plot(density(draw_mean2), main = "Density plot of sample means (N = 100)", cex.main = 0.8)  # Creates a
```




```r
N <- 20  # 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 original sample (dat1),
# takes the mean of the 20 observations, and stores the mean and iterates
# 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] 0.1424135
```

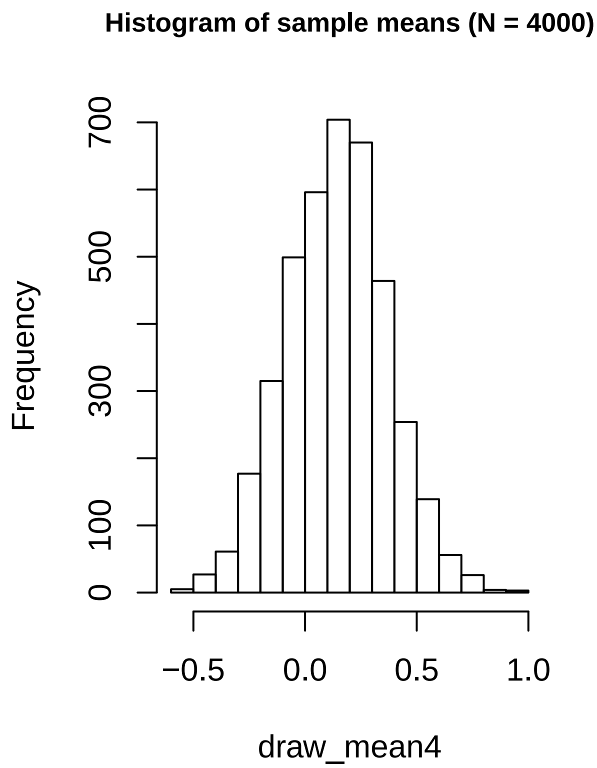
```r
par(mfrow = c(1, 2))
hist(draw_mean3, main = "Histogram of sample means (N = 1000)", cex.main = 0.8)  # Plots a histogram of
plot(density(draw_mean3), main = "Density plot of sample means (N = 1000)",
    cex.main = 0.8)  # Creates a density plot of the sample means
```

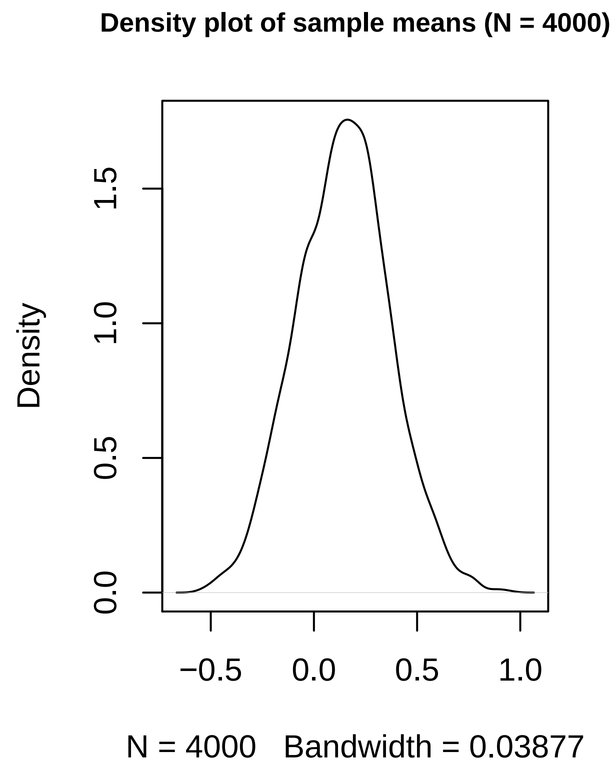



```r
N <- 20  # Set the size of sample
ITER <- 4000  # Set the number of iterations

# 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] 0.1422502
```

```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
```






```
par(mfrows=c(2,2))
```

```
## Warning in par(mfrows = c(2, 2)): "mfrows" is not a graphical parameter
```

The Central Limit Theorem

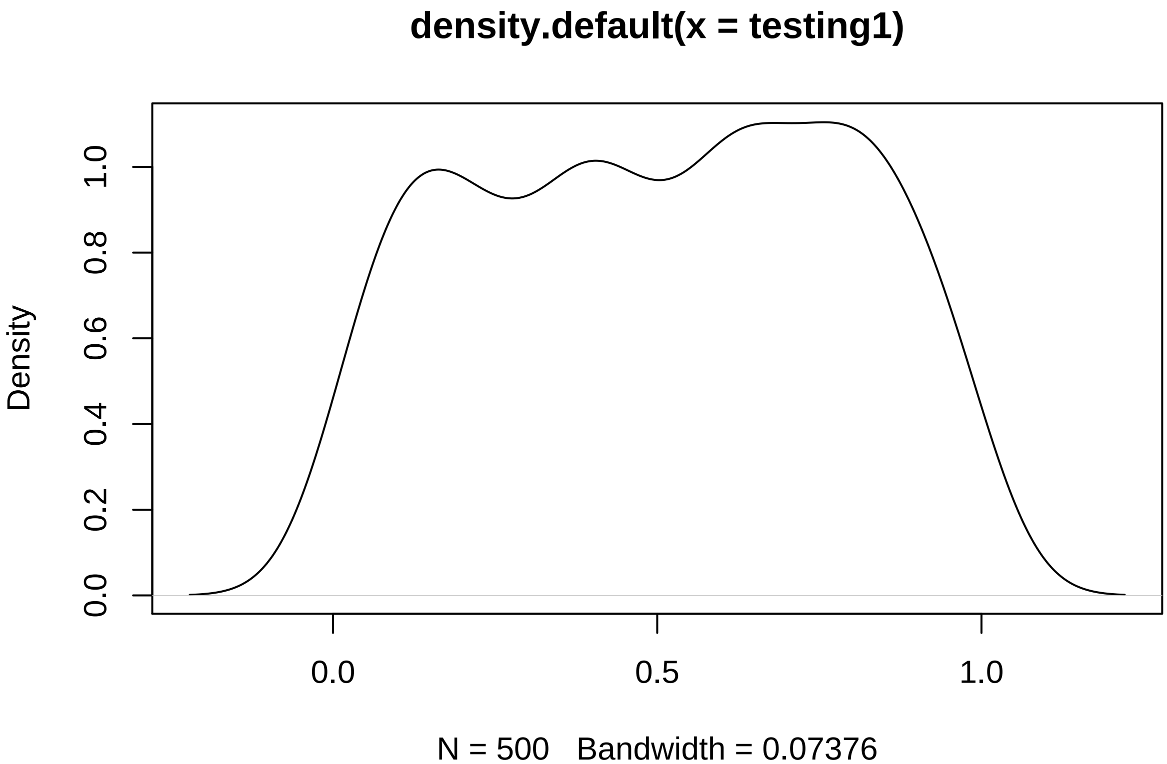
The **central limit theorem** states that given a population with mean of μ and a standard deivation of σ, the sampling distribution of the mean has a mean of μ and a standard deivation of $\frac{\sigma}{\sqrt{N}}$ and approaces a normal distribution as the sample size on which it is based, N, approaches infinity.

1. The mean of the sampling distribution of the mean

*** Describe what a uniform distribution is!!!!**** For a unifrom distribution, the probability density function,

1. Sample from a uniform distribution (use runif) and store your sample.

```
testing1 <- runif(500, 0, 1)
plot(density(testing1))
```



2. Create a sampling distribution of the mean with 20, 100, 1000, and 4000 thousand draws