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 (note: it must have finite mean) 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 $\frac{\sigma}{\sqrt{n}}$ (where σ is the population standard deviation).

A New Example

Suppose you are trying to estimate the mean commuting time for all graduate students at NYU.

Suppose for a moment that the truth, which you can't know in real life, is this: If you could ask every single graduate student to report their commuting time, the true mean would be 45 minutes (sd=15 minutes).

Suppose that, in order to estimate the mean commuting time for the full population, you randomly select a sample of 50 graduate students and observe a sample mean ($\bar{x}$) equal to 35 minutes. If you randomly sampled 50 different graduate students, you might have found a sample mean of 50 minutes.

If you could take all possible samples of 50 graduate students and compute the sample mean for each, then the resulting distribution of means is called the **sampling distribution of the mean**. From the **central limit theorem** the sampling distribution of the mean is normal with a mean of μ and standard deviation $\frac{\sigma}{\sqrt{n}}$ (where μ=the population mean and σ=the population standard deviation).

Simulation in R

We can simulate this in R by taking repeated samples of size 50 from a population with known parameters and storing the mean of each sample.

There are a number of base R functions that randomly sample values from probability distributions with specified parameters. All of these functions start with the letter "r". For example, `rbinom` samples from a binomial distribution, `rpois` samples from a Poisson distribution, and `rnorm` samples from a normal distribution.

First, let's create a simulated population of graduate student commute times.

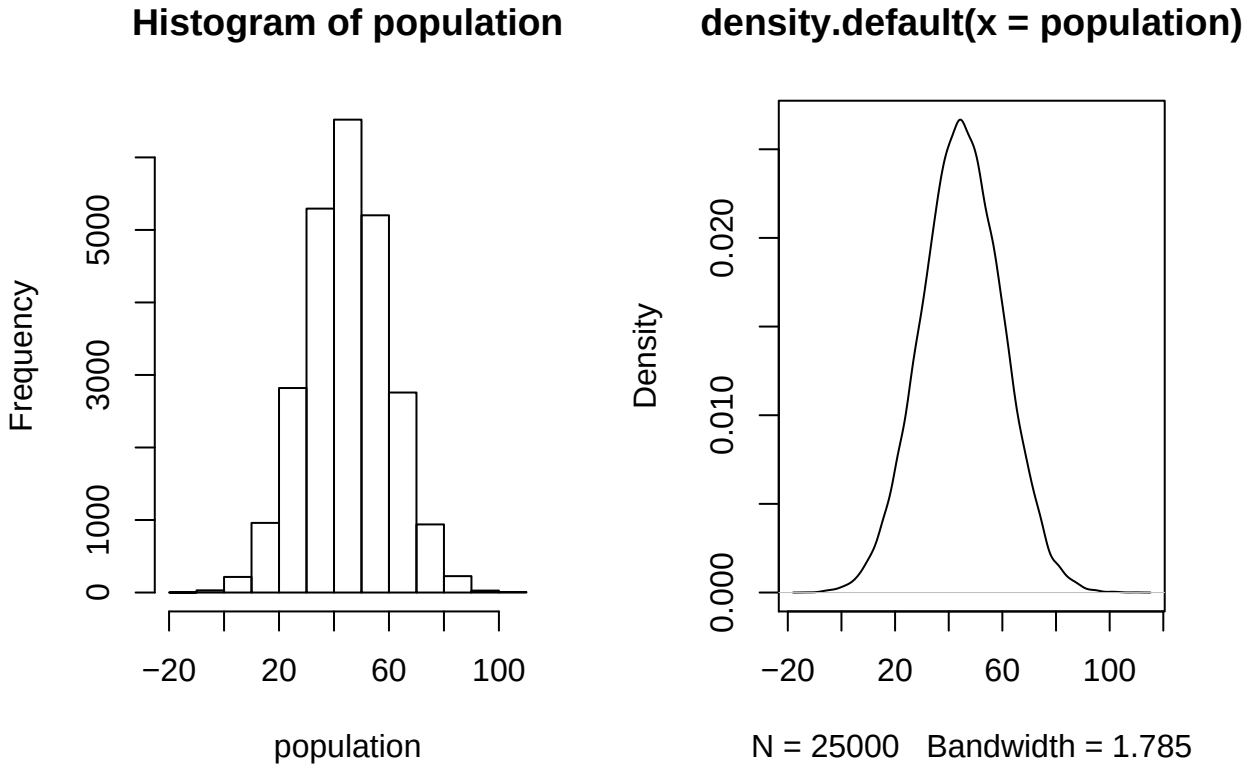
```
# Set.seed allows reproducible results when using random sampling functions
set.seed(123)

# Draw 25000 values from a normal(45,15) distribution and save them in a
# vector: population
population <- rnorm(n = 25000, mean = 45, sd = 15)
pop_mean <- mean(population)  # Computes the mean of the population
pop_mean  # Prints the mean of the population
```

```
## [1] 44.90227
```

```
pop_sd <- sd(population)  # Computes the sd of the population
pop_sd  # Prints the sd of the population
```

```
## [1] 15.02874
```

```r
par(mfrow = c(1, 2))  # Creates a 1 by 2 grid to plot figures in
hist(population)  # Plots a histogram of the draw
plot(density(population))  # Creates a density plot of the draw
```



Notice that the mean and standard devation of the simulated commute times are very close to 45 and 15, respectively. Note: because of the set.seed() function, you will get the same exact values every time you run the code chunk above (you can remove it and re-run the code to see a new random sample of 25,000 values if you'd like).

Now, suppose that these commute times (saved as a vector called "population") are the true commute times of all graduate students at NYU. If we take a lot of random samples of 50 commute times from the overall population and save the resulting sample means, we will get a very good approximation for the sampling distribution of the mean. The central limit theorem tells us that the sampling distribution of the mean is normal, with a mean of μ and standard deviation $\frac{\sigma}{\sqrt{n}}$ (where μ=the population mean and σ=the population standard deviation). Let's use the population values to estimate the mean and standard deviation of the sampling distribution of the mean:

```r
samp_size <- 50 # Save a sample size of 50

# Save and print the mean of the sampling distribution of the mean
samp_dist_mean <- pop_mean
samp_dist_mean
```

```
## [1] 44.90227
```

```r
# Save and print the standard deviation of the sampling distribution of the mean
samp_dist_SE <- pop_sd/sqrt(samp_size)
samp_dist_SE
```

```
## [1] 2.125385
```

Thus, we expect the sampling distribution of the means to be normally distributed with mean=44.9 and

sd=2.13. Now we are ready to test this using a simulation in R. To start, we will repeat the process 100 times (in each iteration, we sample 50 values from the population and save their mean). Then, we will inspect the distribution of the sample means and take their mean and standard deviation.

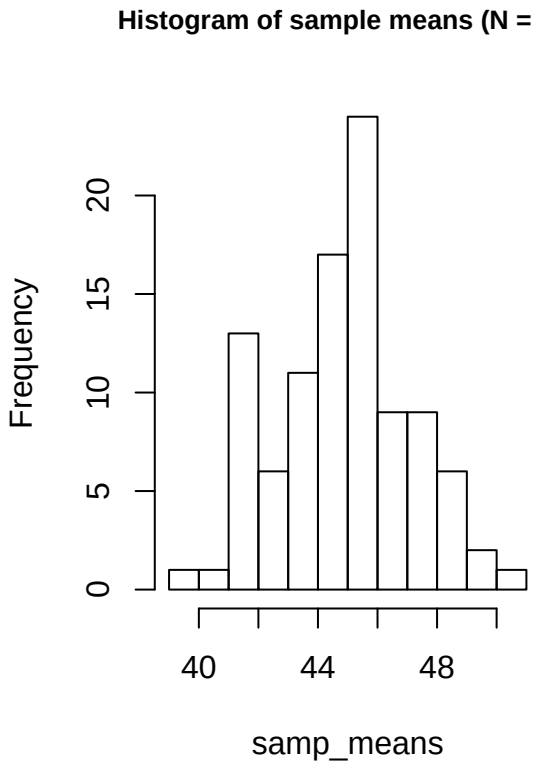
```
samp_size <- 50  # Set the size of sample
ITER <- 100  # Set the number of iterations (i.e., number of samples)

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

# Creates a loop to repeat this process ITER times
for (i in 1:ITER) {
    samp_means[i] <- mean(sample(population, size = 50))
}

# This should be close to 44.9
mean(samp_means)
```

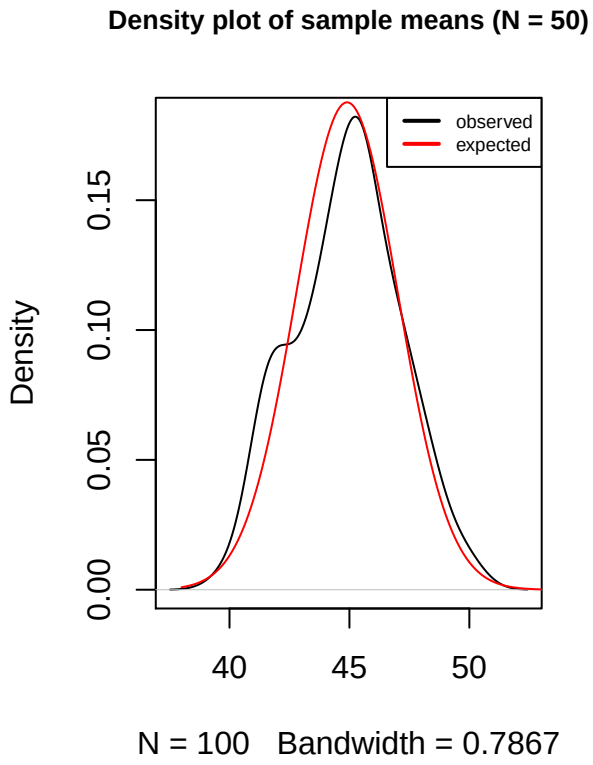
```
## [1] 44.89024
```

```
# This should be close to 2.13
sd(samp_means)
```

```
## [1] 2.195636
```

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

# Creates a density plot of the sample means
plot(density(samp_means), main = "Density plot of sample means (N = 50)", cex.main = 0.8)

# Adds a density plot showing the expected sampling distribution based on
# CLT
lines(seq(38, 53, 0.1), dnorm(seq(38, 53, 0.1), samp_dist_mean, samp_dist_SE),
    col = 2)
legend(x = "topright", legend = c("observed", "expected"), lwd = 2, lty = c(1,
    1), col = c(1, 2), cex = 0.6)
```





We observed a sampling distribution that was fairly similar to what we expected. The more samples we take (i.e., the larger ITER is), the closer the observed sampling distribution should be to our expectations based on the CLT (feel free to experiment!).

Homework (Due after 1st week of classes)

```
# Keep this here when you submit and do not call set.seed() again This way,
# everyone will get the same answers
set.seed(333)
```

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 two parameters: a and b. The PDF of the uniform distribution is defined such that $f(x) = \frac{1}{b-a}$ on the interval [a,b]. Therefore, if you draw values from a Uniform(a,b) distribution, then all values on the interval [a,b] are equally likely.

1. Sample 10,000 values from a uniform distribution with a=0 and b=1 (use 'runif'; note that the parameter a is the same as "min" and b is the same as "max" in the function documentation). Store these variables in a vector called "pop" and plot a histogram of the values. Note: You can type runif in the *Help* tab or type '?runif' into the Console to look at the documentation of the function.

```
# Sample and store (in a vector: pop) 10000 values from a uniform(0,1)
# distribution

# Plot a histogram of pop
```

2. Note that the expected value of a Uniform(a,b) distribution is $E(X) = \frac{a+b}{2}$ and the variance of the

uniform distribution is $Var(X) = \frac{1}{12}(b-a)^2$. Therefore, if X is a Uniform(0,1) random variable, then $E(X) = \frac{0+1}{2} = 0.5$ and $Var(X) = \frac{1}{12}(b-a)^2 = \frac{1}{12}(1-0)^2 \approx 0.083$. Calculate the mean and standard deviation of "pop" and verify that you get values close to 0.5 and $\sqrt{0.083} \approx .289$, respectively. Save the population mean as pop_mean and the population standard deviation as pop_sd

```
# Calculate population mean and save as pop_mean

# Print pop_mean

# Calculation population sd and save as pop_sd

# Print pop_sd
```

3. In question 4, you will take 1000 samples of size 50 from the population ("pop") and calculate the mean of each sample. What do you expect the mean and standard deviation of these means to be? Note: show your calculations and final answers below:

```
# Calculate the expected mean of the means and save as samp_dist_mean

# Print samp_dist_mean

# Calculate the expected sd of the means and save as samp_dist_sd

# Print samp_dist_sd
```

4. Draw 1000 samples of size 50 from "pop"; calculate and save the mean of each sample in a vector called samp_means. Calculate the mean and standard devation of the samp_means. Then, plot a histogram and density plot of samp_means. The code has been outlined for you.

```
# Set the size of sample

# Set the number of iterations

# insert code here to initialize an empty vector called samp_means


for (i in 1:ITER) {
    # Insert code here to sample 50 values from a uniform(0,1) distribution And
    # save the mean of those values in the ith location of samp_means
}

# Calculate the mean of samp_means

# Calculate the standard deviation of samp_means

# Plot a histogram of samp_means

# Plot a smoothed density plot of samp_means
```