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 $\frac{\sigma}{\sqrt{n}}$ (Note: this is called the *standard error* of $\bar{x}$) 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.

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 distribution with mean=45 and standard deviation=15.

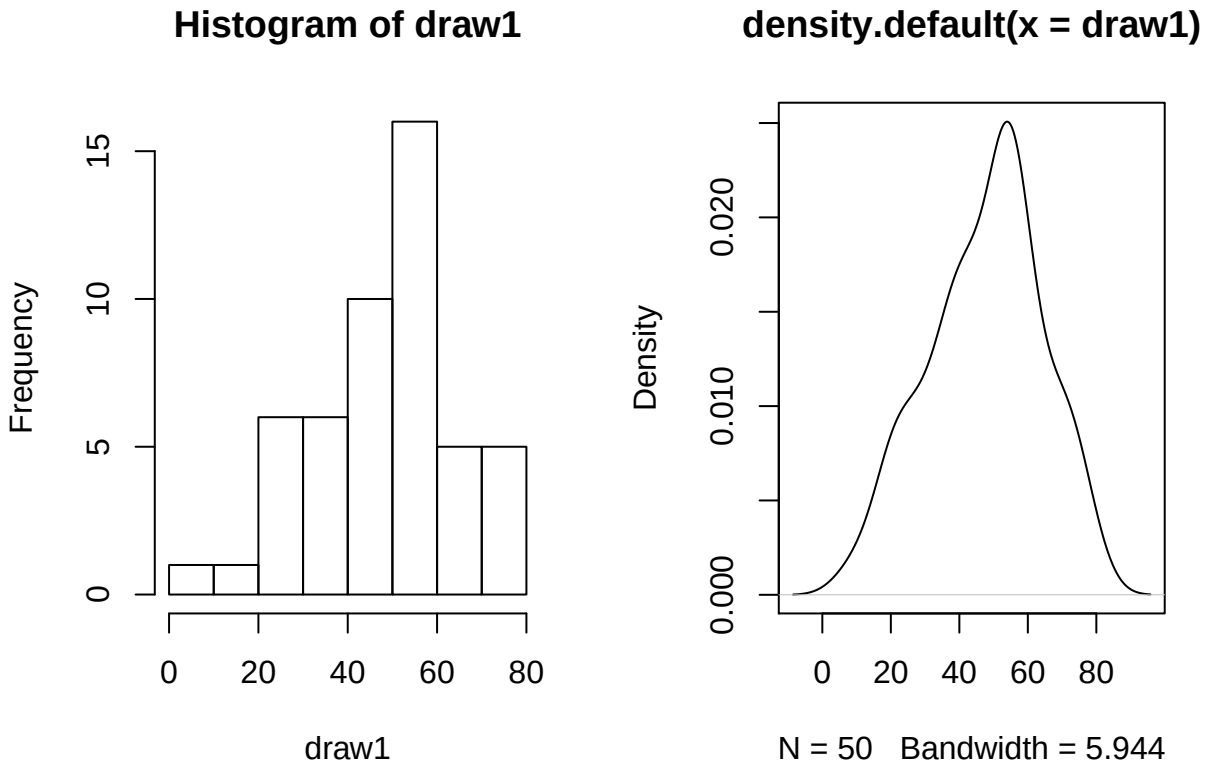
There are a number of base R functions that enable you to randomly sample values from probability distributions with specified parameters. All of these functions start with the letter "r". For example, 'rbinom' samples from a binomial distribution, 'rpois' samples from a Poisson distribution, and 'rnorm'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

We will explore the example above through simulation, assuming that graduate student commute times are normally distributed with mean 45 and sd 15.

```
# Set.seed allows reproducible results when using random sampling functions
set.seed(12345)

# Draw 50 observations and take the mean of the draw
N <- 50  # Storing the number of observations

# A sample of 50 observations from a normal(45,15) distribution
draw1 <- rnorm(n = 50, mean = 45, sd = 15)
mean(draw1)  # Computes the mean of the draw
```

```
## [1] 47.69349
```

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



Note that the mean for this particular sample is about 47.7, a little bit above the true population mean.

Now let's create a sampling distribution of the mean for a sample size of 50. To do this, we will repeat the above process 100 times and plot a histogram of all of the sample means. We should expect the mean (of the means) to be approximately 45 and the standard deviation (of the means) to be approximately $\frac{15}{\sqrt{50}} \approx 2.12$. The more samples we take (i.e., the larger ITER is), the closer we will get to these values (feel free to experiment!).

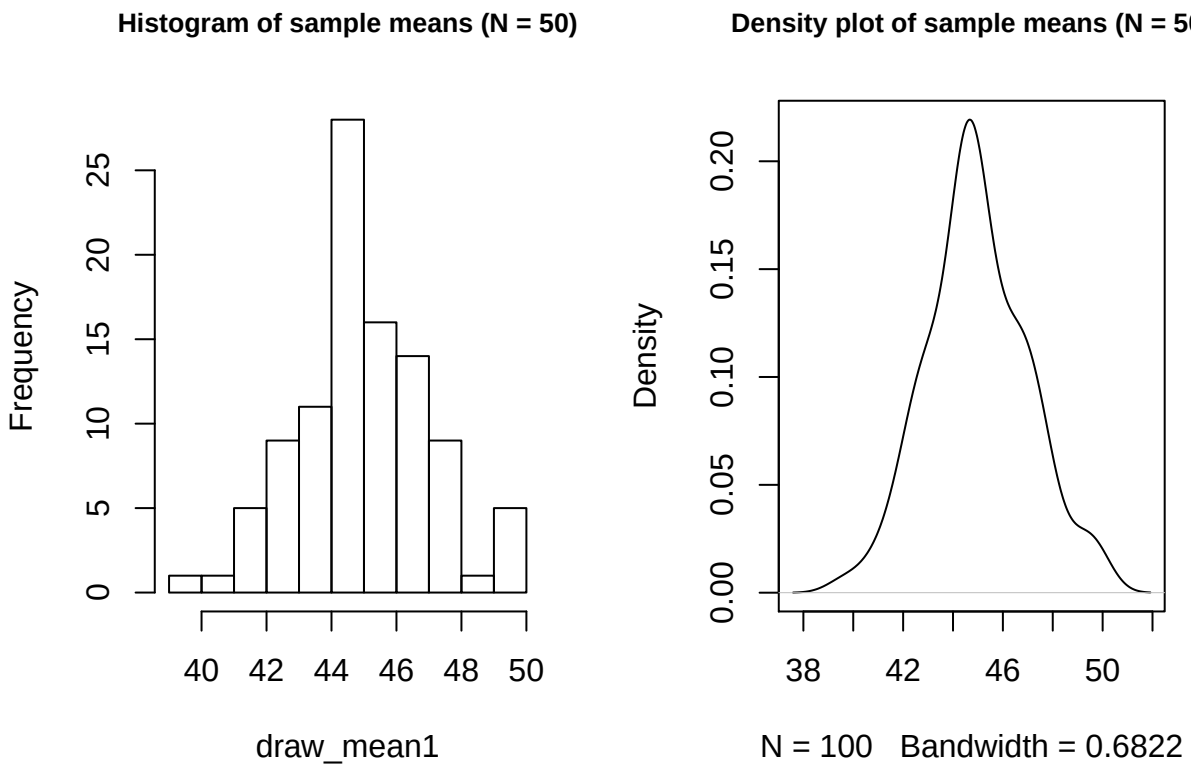
```r
N <- 50  # Set the size of sample
ITER <- 100  # Set the number of iterations (i.e., number of samples)

# Create empty vector of length ITER to store results
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(rnorm(n = 50, mean = 45, sd = 15))
}

# This should be close to 45
mean(draw_mean1)
```

```
## [1] 44.97673
```

```r
# This should be close to 2.12
sd(draw_mean1)
```

```
## [1] 2.018829
```

```r
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)
```



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 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 1000 values from a uniform distribution with a=0 and b=1 (use 'runif'). Store and plot a histogram of 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.

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

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 variance of the data you collected above and verify that you get values close to 0.5 and 0.083, respectively.

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

3. Suppose that you could take infinitely many samples of size 50 from a Uniform(0,1) distribution and calculate the mean of each sample. What would you expect the mean and standard deviation of the resulting sample means to be? Note: show your calculations below:

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

4. Draw 1000 samples of size 50 from a Uniform(0,1) distribution. Calculate the mean and standard devation of the means. Then, plot a histogram and density plot of the means. The code has been outlined for you.

```
# Sampling distribution of the mean from a uniform(0,1) distribution (samp size=50)

 # Set the size of sample
 # Set the number of iterations for the size of the sampling distribution

draw_mean_unif <- # insert code here to initialize an empty vector


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 draw_mean_unif
}

# Calculate the mean of draw_mean_unif

# Calculate the standard deviation of draw_mean_unif

# Plot a histogram of draw_mean_unif

# Plot a smoothed density plot of draw_mean_unif
```