Lab #3, Part #2 – Simulating Sampling Distributions in R

We recently showed how the Central Limit Theorem (CLT) works with a number of different data distributions in class. In this lab, you'll use data to simulate sampling distributions from data to get a handle on how changes to various parameters impact the convergence of the sampling distribution to a normal approximation.

Activity #1: Randomly Generate Some Normal Data.
There is a function in R that allows us to create data at random, drawn from a normal distribution with whatever mean and standard deviation we want to give it. This function is called `rnorm()`, and it operates something like this:

```
normDat1 <- rnorm(5000, 0, 1)
```

Here, we call the data `normDat1`, then note that we want o draw 5000 observations from a normal distribution with mean 0 and standard deviation 1. `normDat1` is just a vector of random values that R grabs from this distribution. You could do the same with 10000 observations, a mean of 10, and a standard deviation of 20 with:

```
normDat2 <- rnorm(10000, 10, 20)
```

1.1 Do both of the above, and report the mean and standard deviation of `normDat1` and `normDat2` using the `mean()` and `sd()` functions. How do they compare to the known population normal distribution from which they were drawn? Make a histogram of each, side-by-side with `par(mfrow = c())` and compare them to one another.

Activity #2: Multiple Samples and Varying N.
Let's stick with the N(0, 1) version for now. Take a second sample using `rnorm()` of 5,000 observations and call it `normDat1b` and have R calculate the mean and standard deviation.

2.1 Report these in your lab report. Did you end up with the exact same values? Why do you think this is the case?

2.2 Go ahead and take 5 more samples, this time with 5, 10, 30, 100, and 500 observations. Call these `normDat5`, `normDat10`, `normDat30`, `normDat100`, and `normDat500`, respectively. Report the mean and standard deviation of each. What do you notice about your samples?

Activity #3: Replicating Sampling N Times.
Let's now assume we want to draw 100 samples, each of N = 5. We could just copy and paste our code over and over again and take the mean, then do it again and again. But this would be 100 lines of code or hitting CTRL+ENTER 100 separate times to get all the samples.

Instead, we can make use of a new function called `replicate()`. Replicate tells R to do some function however many times you'd like over and over again. For example, if we want to make a vector of the letter A 50 times, we could write:

```
Avec <- replicate(50, "A")
Avec
```

Alternatively, we could make a vector of 50 A's and 50 B's:

```
ABvec <- replicate(50, c("A","B"))
ABvec
```

This actually makes a matrix where row 1 is all the A's and row 2 is all the B's, and each replication is a column. We can also wrap a function inside `replicate()`, in which case we apply it multiple times. For example, we could take the average of 0 and 10 many times:

```
Avg0_10 <- replicate(50, mean(c(0, 10)))
Avg0_10
```

This should just return a vector of 5's, since the average of 0 and 10 is always 5. More interesting to us, of course, would be to have a vector of averages of each sample. Let's start by taking the `mean()` of a single N=5 sample from a N(0,1) distribution using `rnorm()` in one swoop:

```
mean(rnorm(5, 0, 1))
```

Hopefully this printed out something close to zero, since that's the mean of the distribution we're drawing from (but remember it does not have to be near zero).

Ok, now that you've seen how replicate() works and how wrapping rnorm() inside mean() gives us a single value, let's stick them together to take 100 samples and take the mean of each sample with N = 5. We can do this in a single line of code instead of 100 now!

```
meanSamps <- replicate(100, mean(rnorm(5, 0, 1)))
meanSamps
```

3.1 You should see some variation in the samples. Use the `summary()` and `sd()` functions to describe the minimum, maximum, quartiles, mean, and variability across the point estimates from the sample. Then, make a histogram of the sample means. Report these and include the histogram and your description of it in your lab report. Does the histogram look like you expected? How does it compare to a N(0,1) distribution? Explain and be specific about the comparison of the standard deviation of the raw data you sampled from with `rnorm()` and the standard deviation of this sampling distribution you've created.

Activity #4: Looking at the Sampling Distribution of Census Data.
On Canvas, you can download the data called CensusAgeDat.csv, which details the age of everyone in the United States. Note, however, that the file is pared down a bit, as 300+ million rows of data can slow down R a good bit. Instead, we'll assume that these population rates for each age, in years, are represented with a sample of 323,068, with more or less the exact same age distribution as shown from the Census. In other words, each row represents 1,000 people of the given age.

Go ahead and save the file to your Lab 3 folder, set your working directory to that folder with `setwd()`, and then load the data in with the `read.csv()` function. Name the data `censDat`. We'll assume this is all the data from the actual population of interest. Then, we'll take samples from it to build a sampling distribution.

Once you load it in, take a look with `head()`, and then count he rows with `nrow()`. You should see that there is one column, `Age`, and 323,068 observations. We'll sample from this variable.

4.1 First, to get an idea of what the data look like, use `summary()` and `sd()` to summarize it and make a histogram. Be sure to give your histogram a title and label axes appropriately if needed. Report these numbers and paste the histogram in your lab report, and describe what you see in these summaries and figure. Does it look normally distributed?

Now that you have a good idea of what the data look like, let's dive into it. Assume you're running a study about the distribution of Age in the United States. In this scenario, and assuming the Census did not exist, you obviously could not just collect the age of every single person in the U.S. on your own. Instead, you would need to go out and get a representative sample of people to report their age to you and assume it is a good estimate of the population distribution.

This is essentially what we do with the `sample()` function in R. So, let's assume you have a team of 5 researchers (including you) and one week to collect the data. Each of the 5 days of the work week, each person in the team is assigned to some new area, and it is assumed they can collect 50 responses each day. Therefore, each of you will return with 250 age observations by the end of the week. This constitutes your entire sample, but you are also interested in how the mean of each of the daily samples changes. We'll focus on the latter here.

Let's begin with the individual sample means for each researcher-day. This will be 5 samples of 50 from each of the 5 researchers for 25 total samples. You can generate them as follows for the first researcher:

```
Res1 <- replicate(5, mean(sample(censDat$Age, 50), replace = F))
```

Note that we've now wrapped the `sample()` function inside the `mean()` function instead of `rnorm()` because we're sampling 50 observations directly from the U.S. population. Do the same for each of the other 4 researchers, numbering them accordingly.

4.2 Report the minimum and maximum sample mean in the data. How close are they to one another? Does any individual researcher seem to regularly have higher or lower sample means? Do the samples have evidence of a violation of independence or a bias in them? Explain.

Now combine the into a single vector with the code below:

```
AllSamps <- c(Res1, Res2, Res3, Res4, Res5)
```

4.3 Make a histogram of AllSamps and take the mean and standard deviation of the AllSamps sampling distribution. Describe what you see, and again compare the sampling distribution to the raw data population distribution. What do you notice about the relationship between the standard deviation of the population data, N = 50, and standard deviation of the sampling distribution (standard error)? Derive the standard error using the population standard deviation (show your work). Does it look similar?

Activity #5: A Big Grant and A Bigger Team.

You've decided that you need a bigger team to be representative across the country and get a good idea of the distribution of ages of U.S. residents. Let's assume you receive a large grant that allows you to deploy 100 research assistants across the country for a month, during which they work 20 days. Each researcher still gets about 50 responses per day. You'll now have 20*50*100 = 100,000 total observations and 2,000 samples of 50. We'll focus on the variability in the individual samples.

This time around, we'll ignore which researcher collected what data and just do all the samples in one big swoop, then summarize the distribution of sample means and find its standard deviation (standard error). Try coming up with this code on your own and call this `sampsBig`.

5.1 Report the mean and standard deviation and make a histogram of your sample means. Does this differ from what you found with your smaller group of researchers? Does the standard error derivation using the population standard deviation change for this larger group of samples? Why or why not? Describe the distribution in the histogram and compare it to the U.S. population Age distribution.

5.2 Find the smallest and the largest point estimate from your samples and report them. How far do they deviate from the population mean and sampling distribution mean? Do you think something is amiss in these samples (i.e. do you think your researchers messed something up those days)? Why or why not? What is the probability of observing each of these sample means in any given sample?

Activity #6: Slow Research Assistants.

Repeat Activity #5, but this time, assume your research assistants can only gather 5 responses per day. All that will change here is the N, from 50 to 5.

6.1 Compare the standard deviation of the sampling distribution to the one from Activity #5. Why do you think there are differences? Does this make sense, given the derivation we went over in class for the standard deviation of the sampling distribution (again, referred to as the standard error)?

6.2 Just like Question 5.2, find the smallest and the largest point estimate from your samples and report them. How far do they deviate from the population mean and sampling distribution mean? Are they more or less extreme than in Activity 5? Do you think something is amiss in these samples (i.e. do you think your researchers messed something up those days)? Why or why not? What is the probability of observing each of these sample means in any given sample?

Activity #7: Random Data from Other Population Distributions.

Now that we've established the sampling distribution seems to look pretty good with reasonable N from the non-normal population data, let's look at 2 other types of distributions where we can apply the CLT with sampling distributions.

We went over the Poisson distribution in class, and showed that its sampling distribution converges toward a normal distribution under certain conditions. Two other distributions that do well under the CLT are the Beta distribution and the Gamma distribution. While we won't talk much about these distributions in class, you can generate random data from each with the `rbeta()` and `rgamma()` functions, respectively, in R.

For this last activity, I want you to generate data from these distributions as follows (for our purposes, don't worry about what the parameters mean – just note that we have a population of 100,000):

```
betaDat <- rbeta(100000, 1, 3)
gammaDat <- rgamma(100000, 4)
```

This will serve as population data, from which we will take samples and calculate point estimates.

7.1 To begin, make a histogram of each of the population data sets side by side with `par(mfrow = c())` and include this in your lab write-up. Describe each in detail and compare them.

In this activity, instead of the mean, we'll use different measures. For the Beta distribution, I want you to use the `sum()` of observations from each sample. For the Gamma distribution, I want you to use the standard deviation of each sample with `sd()`. We'll use these to show that other types of point estimates often follow the CLT from many distributions.

7.2 For the Beta distribution, modify your previous code such that you take 1,000 samples of N = 50 from `betaDat`, and sum all observations as the point estimate of interest from each sample. Make a histogram and include all of this information in your lab write-up, along with a full description of the distribution you see in the histogram.

7.3 For the Gamma distribution, modify your previous code such that you take 1,000 samples of N = 50 from `betaDat`, and take the standard deviation as the point estimate of interest from each sample. Make a histogram and include all of this information in your lab write-up, along with a full description of the distribution you see in the histogram.

7.4 Given what you've seen in this lab, use 4-5 sentences to fully explain what you've learned about sampling distributions and the CLT. Be sure to include information related to the derivation of the sampling distribution standard deviation (standard error), the role of N, and the role of the population mean and population distribution in the resulting sampling distribution shape and location.