Sampling Distribution Simulation

Joel S Steele, PhD

Winter 2019

Before we get started there are a few house-keeping things to attend to. We want to make sure that is nothing currently defined in our session.

```
# Clean up our workbench (session)
ls() # what's there already?
```

```
## character(0)
```

```
rm(list=ls()) # remove all listed stuff
```

Next we set up some output options to make this document easire to read.

```
knitr::opts_chunk$set(comment=NA)# make this output nice and clean
```

Lastly we set the random number generator seed to a specific value. Spoiler alert random numbers are not really random.

```
set.seed(32) # make random numbers replicable... yeah, that's a thing.
```

Sample vs Population

Here we are going to simulate a class of 50 students who have just completed an exam. We will be using this group as our made up population and will therefore take samples from it. To simulate we will set a range of scores to represent how a class did on the exam.

```
range_o_scores = 50:100 # from a low of 50 to a high of 100.
```

Next, we grab 50 of these scores, with replacement so some scores will repeat. This set of 50 scores is our hypothetical class.

```
scores <- sample(range_o_scores,50,replace=T)
```

Now that we have a sample of 50 scores we can treat this as our class population. Next let's see how the class did overall. But first we ALWAYS look at our data.

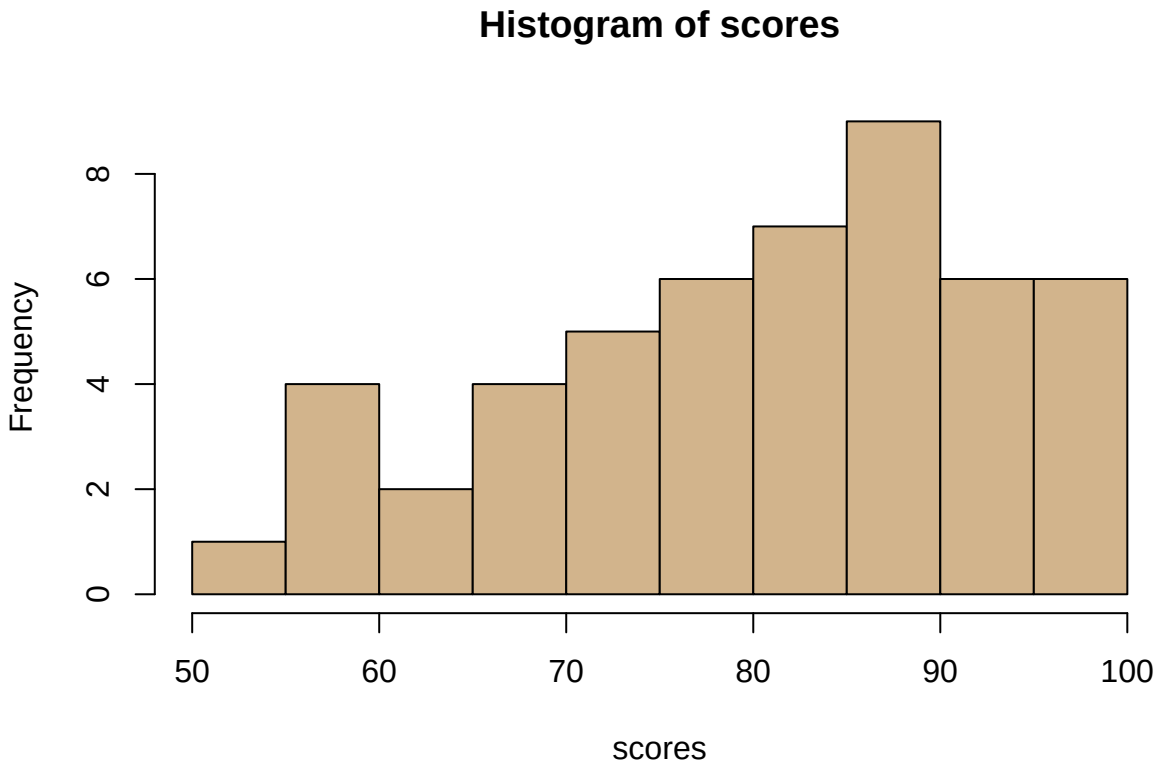
```
scores
```

```
 [1]  75  80  91  87  57  98  88  93  84  69  83  66  81  89  63  88  90
[18]  71  81  96  84  95  73  57  79 100  73  76  77  81  70  92  96  92
[35]  72  89  76  57  90 100  69  88  57  77  92  61  82  99  89  53
```

Step 1 Visualize

Let's start with a basic histogram and stem-and-leaf plot.

```
hist(scores,col='tan') # histogram
```




```
stem(scores) # stem-and-leaf plot
```

```
  The decimal point is 1 digit(s) to the right of the |

   5 | 3
   5 | 7777
   6 | 13
   6 | 699
   7 | 01233
   7 | 566779
   8 | 01112344
   8 | 7888999
   9 | 0012223
   9 | 56689
  10 | 00
```

Step 2 Describe

```
mean(scores) # mean
```

```
[1] 80.52
```

```
sd(scores) # sd
```

```
[1] 12.78909
```

```
range(scores) # min and max
```

```
[1]  53 100
```

Sampling Distribution Simulation

From here we will take some samples from our class scores and see how the selected numbers look.

```
sample(scores,5)
```

```
[1] 69 82 92 90 75
```

Notice you can just use a capital T in place of the word TRUE.

```
sample(scores,5, replace=T) # catch and release.
```

```
[1] 75 88 77 72 99
```

To mimic our sampling distribution we will take the mean of each sample and repeat this a couple of times. Remember that the mean of the entire population (our hypothetical class) is 80.52. Let's take 5 samples from our class, each sample will have 5 people

```
mean(sample(scores,5,replace=T)) # average of sample 1
```

```
[1] 83.2
```

```
mean(sample(scores,5,replace=T)) # average of sample 2
```

```
[1] 74.8
```

```
mean(sample(scores,5,replace=T)) # average of sample 3
```

```
[1] 84
```

```
mean(sample(scores,5,replace=T)) # average of sample 4
```

```
[1] 79.6
```

```
mean(sample(scores,5,replace=T)) # average of sample 5
```

```
[1] 75.6
```

Pretty close to our average, but not great below we increase our sample size to 20 people. Let's take ten samples and see how they compare. Just a reminder our population mean is 80.52

```
mean(sample(scores,20,replace=T)) # average of sample 1
```

```
[1] 82.9
```

```
mean(sample(scores,20,replace=T)) # average of sample 2
```

```
[1] 83.25
```

```
mean(sample(scores,20,replace=T)) # average of sample 3
```

```
[1] 83.95
```

```
mean(sample(scores,20,replace=T)) # average of sample 4
```

```
[1] 78.05
```

```
mean(sample(scores,20,replace=T)) # average of sample 5
```

```
[1] 79.85
```

```
mean(sample(scores,20,replace=T)) # average of sample 6
```

```
[1] 79.25
```

```r
mean(sample(scores,20,replace=T)) # average of sample 7
```

```
[1] 80.1
```

```r
mean(sample(scores,20,replace=T)) # average of sample 8
```

```
[1] 84.25
```

```r
mean(sample(scores,20,replace=T)) # average of sample 9
```

```
[1] 79
```

```r
mean(sample(scores,20,replace=T)) # average of sample 10
```

```
[1] 81.7
```

Pretty cool, but it would be nice if we could keep track of all the sample averaged that we are calculating so we could look directly at our simulated sampling distribution. Turns out we can using the replicate() function below. This will repeat the steps above and store things in a vector.

```r
# Read up on replicate
#?replicate
```

Replication

Ten Samples of Size 5

To start we will take 10 samples of size 5 again but this time we will store the results.

```r
S10N5 = replicate(10,mean(sample(scores,5,replace=T)))
S10N5 # what are the sample averages?
```

```
 [1] 79.2 86.0 80.6 77.6 63.2 83.0 83.2 80.6 84.2 82.6
```

Let's compare.

```r
mean(scores)# population mean
```

```
[1] 80.52
```

```r
mean(S10N5) # mean of sample means
```

```
[1] 80.02
```

Now let's compute the Standard Error, this is really just the standard deviation of our sampling distribution. We know from the Central Limit Theorem that this is defined as the standard deviation of the population divided by the square root of the sample size. Let's compute this below and see how we did.

```r
sd(scores) # population sd
```

```
[1] 12.78909
```

```r
sd(S10N5) # sampling distribution sd (Standard Error)
```

```
[1] 6.40309
```

```r
sd(scores)/sqrt(5) # what would the formula predict?
```

```
[1] 5.719455
```

One hundred Samples of Size 25

Not bad, not great, but not bad either. Next we increase the sample size to 25 and take 100 samples. This is equal to half of our population!

```
S100N25 = replicate(100,mean(sample(scores,25,replace=T)))
S100N25
```

```
  [1] 78.84 75.68 77.32 80.60 85.00 80.60 77.24 81.04 81.08 82.64 84.32
 [12] 79.60 79.44 82.68 83.76 78.52 81.04 78.32 77.96 78.28 78.84 82.60
 [23] 85.40 78.48 77.56 85.32 78.84 82.32 85.16 80.20 76.48 83.08 79.96
 [34] 82.36 79.72 81.20 77.44 78.68 79.20 78.56 81.24 80.04 83.04 80.84
 [45] 83.28 77.28 78.68 81.72 79.56 81.04 77.60 82.48 81.48 83.52 81.84
 [56] 81.04 81.40 85.48 79.20 79.84 82.00 78.68 80.76 83.12 74.04 79.40
 [67] 79.88 81.64 78.84 81.20 80.04 82.76 79.28 85.68 75.44 83.08 79.76
 [78] 79.20 81.08 82.84 82.28 82.52 79.84 79.92 79.72 81.68 75.80 79.04
 [89] 81.72 79.44 79.24 79.64 79.44 81.52 80.00 82.60 78.64 79.48 79.40
[100] 81.36
```

```
mean(scores)# population mean
```

```
[1] 80.52
```

```
mean(S100N25)# mean of sample means
```

```
[1] 80.4796
```

standard error

```
sd(scores) # population sd
```

```
[1] 12.78909
```

```
sd(S100N25) # sampling distribution sd (Standard Error)
```

```
[1] 2.326008
```

```
sqrt(var(scores)/25) # what would the formula predict?
```

```
[1] 2.557818
```

Five thousand samples of Size 45

Things are getting better, but let's push our computer a bit.

```
S5000N45 = replicate(5000,mean(sample(scores,45,replace=T)))
```

We will skip taking a look at all five thousand sample means directly, but I'm sure you can imagine they will all be rather close to the population mean.

```
mean(scores)# population mean
```

```
[1] 80.52
```

```
mean(S5000N45)# mean of sample means
```

```
[1] 80.49172
```

standard error

```r
sd(scores) # population sd
```

```
[1] 12.78909
```

```r
sd(S5000N45) # sampling distribution sd (Standard Error)
```

```
[1] 1.894891
```

```r
sqrt(var(scores)/45) # what would the formula predict?
```

```
[1] 1.906485
```

How does our sampling distribution look?

```r
hist(S5000N45, col='tan',freq = F,
     main='Means of samples of size 45', xlab='sample means')
lines(density(S5000N45),col='red',lwd=2)
```