

Simulations: Sampling Distribution of Average

Background: The average of a sample is -- shall we say a popular? -- estimator of the mean of a population. (Alternative estimators might be the median of a sample, or the average of the biggest and smallest value, or the average of the middle 90% of values (called the trimmed mean), to name a few.) Some of the reasons for this popularity are shown in this demonstration.

Imagine: that we **know** the population of a particular variable, say the amount of rain that falls in the Coral Gables, Florida area, measured in 10^7 cubic meters. In fact, we know that this variable follows a normal distribution with mean 14.4 and standard deviation 3.1.

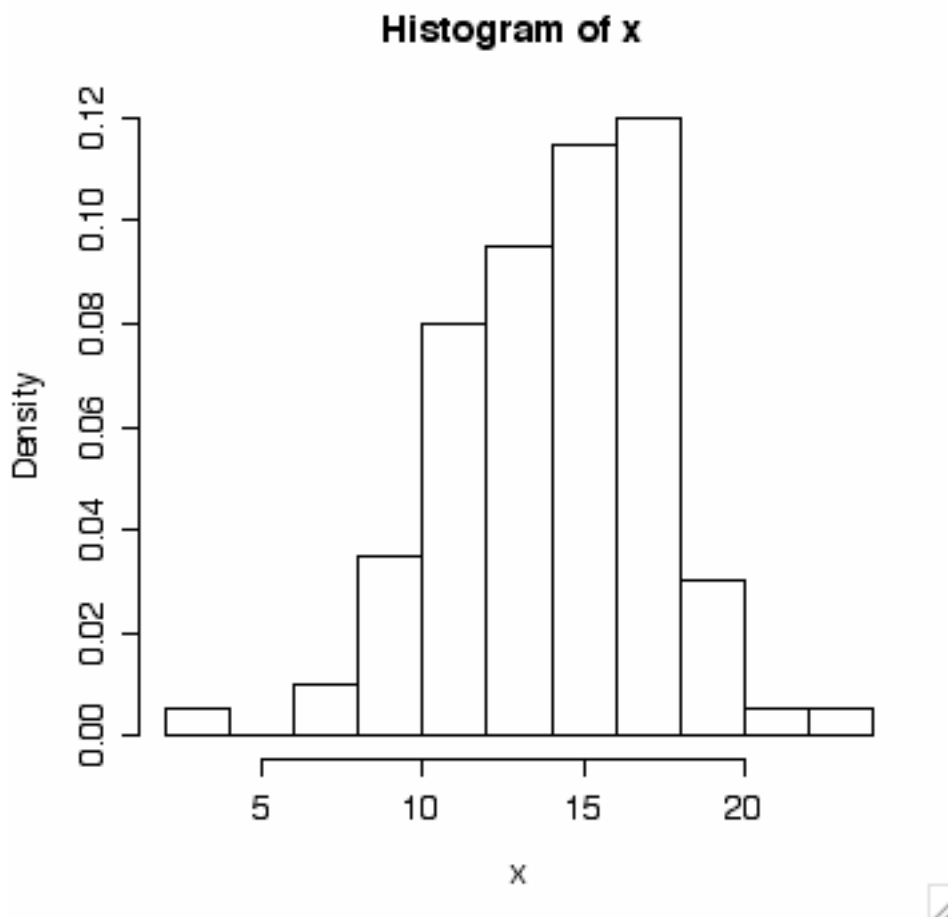
The Simulation: we can simulate a researcher making a measurement on this population with this R command:

```
> rnorm(1,14.4,3.1)
[1] 11.08300
```

Which produces a single random number, drawn from a normal distribution. Of course, the researcher doesn't know this. All he or she sees is the number 11.08 -- the amount of rain recorded.

If we take many observations, say 100, we should see that the values are centered around 14.4. Approximately 68% will be within 1 SD of 14.4 -- so about 68% between $14.4 - 3.1 = 11.1$ and $14.4 + 3.1 = 17.5$. About 95% within 2 SDs: (8.0, 20.6), and almost all -- 99.7% -- between (4.9, 23.7). Let's try it. This is what a researcher might see after taking 100 independent observations from this population:

```
> x <- rnorm(100,14.4,3.1)
> x[1:20]
[1] 17.818969 15.619306 23.356292 11.496608 18.815953
14.712355 11.330867
[8] 12.140779 16.314541 12.384856 11.764926 16.754506
11.379684 8.520755
[15] 11.607767 15.366866 16.387866 9.983140 10.893217
12.147995
> hist(x,freq=F)
```



The average and SD of this sample are:

```
> mean(x)
[1] 14.12239
> sd(x)
[1] 3.226484
```

In this case the average of our sample was pretty close to the mean of the population. (So was the SD of the sample, but we'll talk about that later.) But was this lucky?

If we were to repeat this simulation, would the average of our next sample be close to 14.4? Let's see what we get:

```
> x <- rnorm(100,14.4,3.1)
> mean(x)
[1] 14.91696
```

This time our estimate of the mean, 14.92, is bigger than the true mean, 14.4. But again it's "in the ballpark". The SD of the sample is 3.24, again close to the true value.

Let's speed things up. Let's imagine that we have 1000 researchers who are all conducting the same experiment: take 100 observations from this population. (We assume these are independent, so if two researchers get the same values, it is a coincidence.)

We can simulate this in R with the following algorithm:

- 1) initialize a vector to store our results
- 2) generate a random sample of 100 observations
- 3) calculate the mean of this sample
- 4) store it in our results vector
- 5) repeat steps 2-4 1000 times (once for each hypothetical researcher)

The code to do this is

```
results <- c()           #Step 1: initialize
for (i in 1:1000){       #repeat following commands 1000 times
  x <- rnorm(100,14.4,3.1) #x is a random sample of size 100
  mu <- mean(x)           #calculate the mean of this sample
  results <- c(results, mu)} #store the result in the results vector
```

We can actually condense this quite a bit (although it becomes harder to read):

```
results <- c()
for (i in 1:1000){
  results <- c(results, mean(rnorm(100,14.4,3.1)))}
```

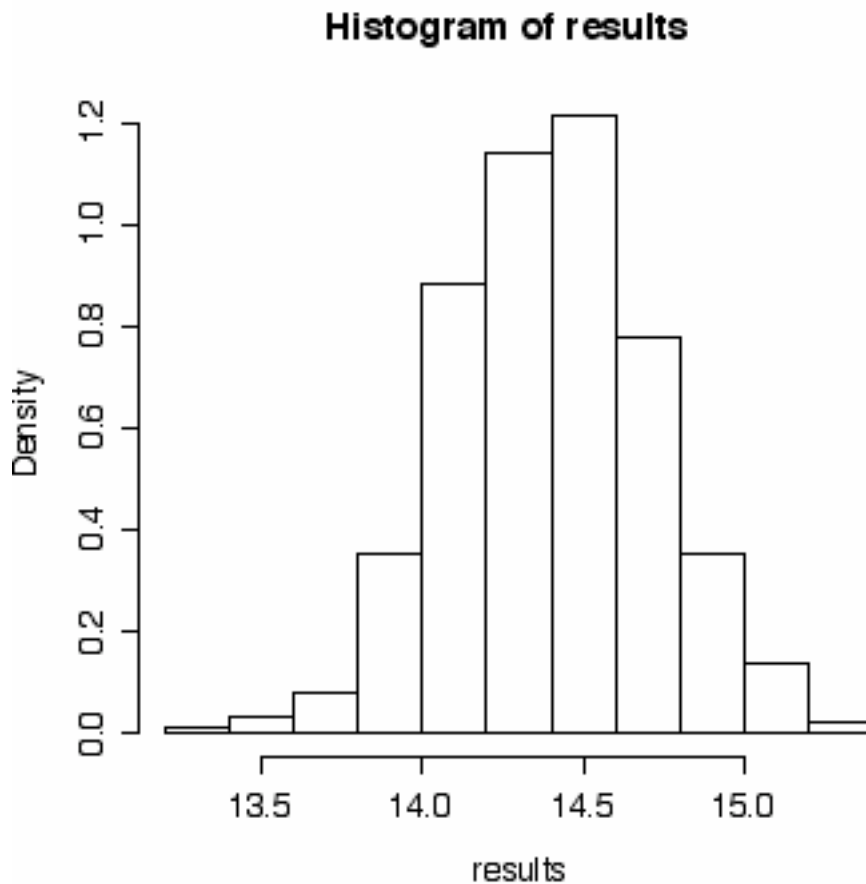
Here are the first 10 averages. Think of these as the estimates of the population mean for the first 10 researchers:

```
> results[1:10]
[1] 14.07758 14.38313 14.07664 14.29438 14.38933 14.27152
14.96627 14.59530
[9] 14.60337 14.48110
```

Note that:

- they are not very spread out -- there is great consistency and
- all are in a "neighborhood" of the true value: 14.4.

In fact, here is a histogram of all 1000 averages:



Some interesting features:

- 1) the 'central' value -- the center of mass -- is very close to the true mean of 14.4. In fact, the average of the averages is $\text{mean}(\text{results}) = 13.39684$. Thus we see that, on average, or "in the long run", the sample average is equal to the population mean.
- 2) Although observations from the population have considerable spread ($\text{SD} = 3.1$, so that 68% are between 11.3 and 17.5), the averages are much less spread out. In fact the SD of these 1000 averages is $\text{sd}(\text{results}) = .31440$, which means that about 68% of them are between 14.09 and 14.71 -- a much narrower interval.
- 3) The distribution of these sample averages looks very close to a normal distribution.

These three observations are reflections of three mathematical facts:

- 1) The expected value of the sample average is equal to the population mean. In other words, the sample average is an unbiased estimator of the population mean. This is always true, no matter which pdf is used to model the population, and whether or not the observations are independent.

2) The spread of sample averages as measured by their standard deviation -- this is called the standard error -- is smaller than the spread of the population. This can be made quite precise. In fact: $SD(\text{average}) = SD(\text{population})/\sqrt{n}$. This is very nice, because it shows us that in fact the spread decreases as the sample size increases. So the bigger the sample, the closer we can expect the average to be to the true value. Don't believe me? In our simulation: $SD(\text{population}) = 3.1$. Sample size = 100. So $SD(\text{average}) = 3.1/10 = 0.31$. In our simulation of 1000 repetitions, we saw that $SD(\text{results}) = 0.3144$, which is quite close. Had we done sample sizes of 10,000, then $SD(\text{average}) = 0.031$.

This fact is true given the following conditions: the observations must be independent of each other. But it does not matter what pdf we draw the observations from.

3) Linear combinations of normally distributed random variables are themselves normally distributed. The sample average consists of a sum of n random variables. Each random variable is taken from the same normal distribution. Therefore their sum is normally distributed and therefore their sum divided by n is normally distributed. This explains why the histogram of the sample averages looks so similar to a normal distribution.

Central Limit Theorem (CLT)

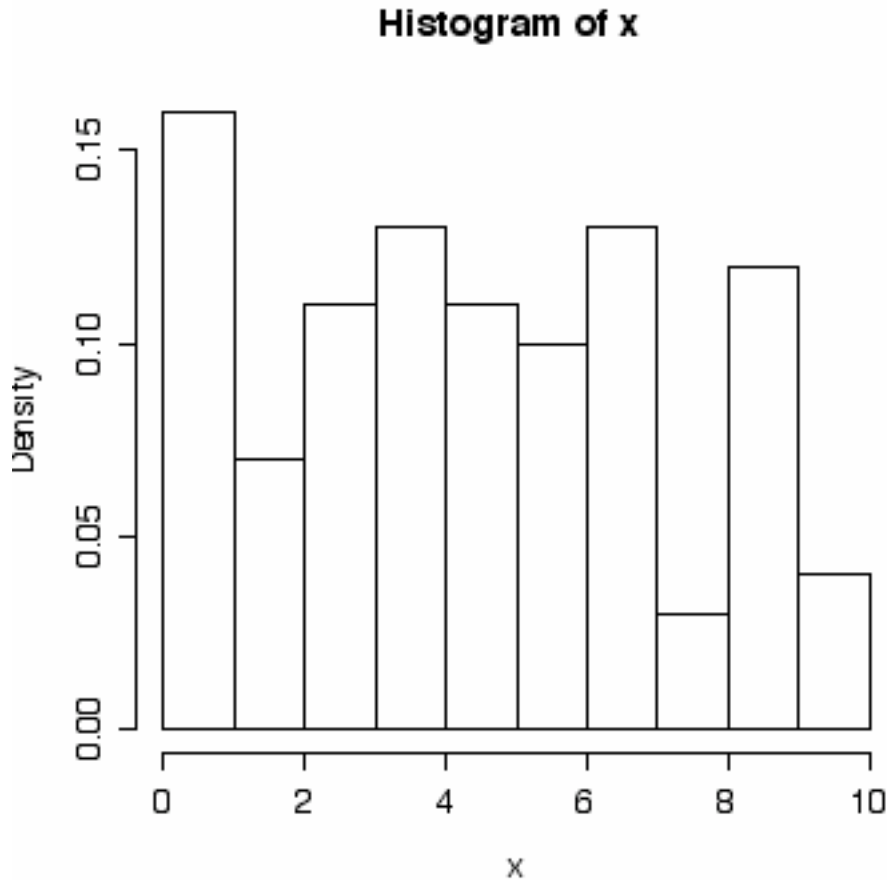
Fact (3) explains that we saw a normal-shaped histogram for our sample averages because the population was normal. In fact, the CLT says that ANY linear combination of independent random variables is *approximately* normally distributed. There are variations of this theorem that apply to slightly different situations. But for now, the important word to focus on is the "approximately". This means that if the population was not a normal distribution, the average will not follow a normal distribution. But its distribution will be close to normal. How close? Depends on the sample size. The greater the sample size, the better the approximation. How big a sample size? Depends on the population distribution. The more skewed it is, the more non-normal, the larger the sample size you'll need.

Let's do another simulation. This time, our researcher draws out 100 samples from a uniform distribution. Here's the command to do it, the first 10 observations, and the average and SD of his sample:

```
> x <- runif(100,0,10)
> x[1:10]
[1] 0.3768652 2.5350590 4.5066203 3.3656516 6.5584788
8.6535154 1.6828664
[8] 4.0691176 0.0757491 0.2422920
> mean(x)
[1] 4.416829
> sd(x)
[1] 2.779954
```

Note that the mean of this population is 5.0. You can see that this is approximately where the histogram above "balances".

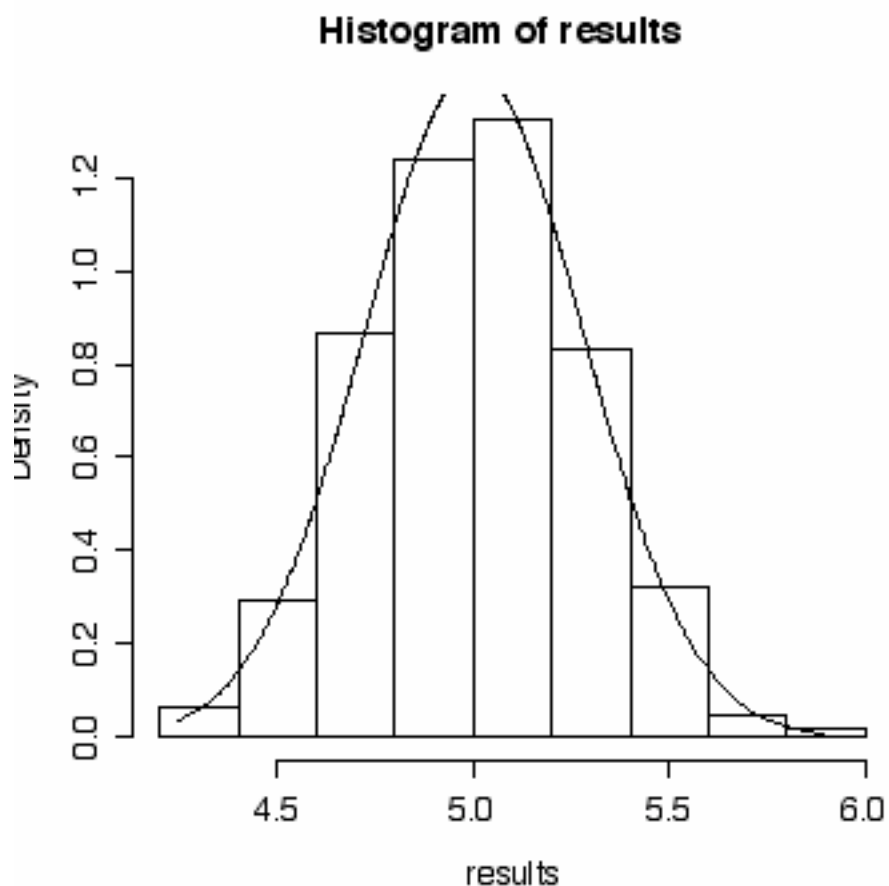
A histogram of this sample will look a little like a uniform distribution (a rectangle):



Now we imagine our hypothetical 1000 researchers. Each of them is going to take a random sample of 100 observations from the uniform population and calculate the average of their sample. We'll save their averages in a vector as we did before:

```
> results <- c() #initialize this storage vector
> for (i in 1:1000) {
+   results <- c(results, mean(runif(100,0,10)))}
> mean(results)
[1] 5.00135
> sd(results)
[1] 0.2788632
```

We see that the average of the averages is quite close to the population mean, and the SD of these averages is approximately $SD(\text{population})/\sqrt{n}$. But its the shape of the distribution of averages that's interesting. Here you can see that it really is approximately normal:



A true normal distribution is superimposed so that you can compare.

A programming lesson

These simulations can be sped up considerably (and the frustrations that occur when you mis-type can be greatly decreased) with a little programming.

R allows you to create functions. Thus, we might define a function called `normsims` that imitates the normal simulation as follows:

```
normsims <- function(N,n,mu,sd){  #N is the number of repetitions, n the sample size,
                                   #mu the mean, and sd the standard deviation for the
                                   #population.

  results <- c()
  for (i in 1:N){
    results <- c(results, mean(rnorm(n,mu,sd)))}
  results}
```

A function in R always returns as its value the result of the very last line. Thus, the last line of this function is just the vector "results".

To execute this function we just type

```
output <- normsims(1000,100,14.4,3.1)
```

If we didn't assign the output to a vector, we would be inundated while the screen displayed the 1000 averages it created.

Now, if, while typing these commands into the R window you make a mistake, it can be frustrating trying to fix the mistake. So here's a way around this:

- a) Using a texteditor (appleworks, for example), open a blank file and type the commands above in exactly as they appear above.
- b) Save the file with a memorable name, such as myfile.
- c) Now, go back to R. And from the "File" menu choose "Source file" and select myfile. R automatically loads it in and executes the commands. Your function is now ready to run.

This is a convenient way of editing functions that you write. The reason is that when you "source" the file, R will return error messages. You can then edit the file and source it again. Very quick and convenient way of debugging.

Homework

1. A cola company advertises that a can of their product contains 12 ounces. A cola-dispensing machine at their factory is engineered to dispense mean amount of 12.2 oz of cola into a can. The standard-deviation in the amount dispensed is 0.15 oz. Assume that X , the amount dispensed in a single can, follows a normal distribution. Answer parts (a) - (d) without using a computer.
 - a) Find the probability that less than 12 oz. is dispensed in a can.
 - b) Suppose 1000 cans have been filled. About how many should we expect to have less than 12 oz?
 - c) Cans are sold in packs of 6. What should we expect the average of a "six-pack" to be?
 - d) What's the probability that the average of a six-pack will be less than 12 ounces?
 - e) Write a simulation using R that answers part (d). Note that you won't get the exact probability (because you are using a random number generator, you'll get a slightly different value each time.) The greater your number of repetitions (use at least 1000), the closer to the answer in (d) you'll get. This type of probability is sometimes called an "empirical probability" because it is based on the outcomes of an experiment. (Granted, this is a "virtual" experiment.) To help you calculate how many of your averages were less than 12 oz, you'll need to try something like the following:

```
> x <- rnorm(10)
> x
```



```
[1] -1.0244037 -2.1443087 -1.4294248  0.2338177  0.6728196  
0.1627471  
[7] -0.9231361 -0.3410352  2.0021777 -2.6683111
```

```
> x < 0
```

```
[1]  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE  
TRUE
```

```
> sum(x < 0)
```

```
[1] 6
```