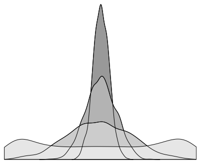

Sampling Distributions

1 A random sample, a statistic

The basic setup of *statistical inference* involves a *target population* for which some characteristics are sought. These may be its center, as measured by its mean or median, or its spread, as measured by the standard deviation or IQR. These have numeric values. A *parameter* is a numeric value summarizing a population. To find out the values of the parameters, a *random sample* may be chosen from the population. With a sample in hand, we can compute *sample statistics*, or numeric summaries of the sample. For instance, the population mean is the (unknown) center of the population, the sample mean is simply the average value of the terms in the sample. These may be completely different. The sample mean typically varies from sample to sample, the population mean is always the same for a given population.

Notationally different but related symbols are used. For the mean, it is traditional to use a μ (mu) for the population mean and "x bar", $\bar{x}$, for the sample mean. For the standard deviation, a σ (sigma) represents the parameter and an s the statistic. For a proportion, p is used for the parameter and a hat, as in $\hat{p}$, indicates a sample statistic.

Although random, the sample mean can be used to give information about the population mean. If we know the distribution of the possible values of the sample mean, then we can infer from that a value of the population mean.

2 simulation

By taking samples and computing the sample statistic many times we can get an understanding of the *sampling distribution* of the statistic. When we do this, we are performing a *simulation*. Although, many properties of sampling distributions can be derived mathematically, it is also possible to get much insight about the population by doing a computer simulation.

2.1 one sample at a time

The basic ingredients of a simulation are

- to take a sample from the population
- to find the value of the statistic from the sample
- to repeat enough times to get an idea of the sampling distribution

Taking a random sample from a finite population may be done using the `sample()` function. For instance, in a popular opinion poll the goal is to find the unknown proportion of the population who are "for" some proposition, or question. Suppose the population is clear about their position and would answer either "yes" or "no" if asked. We can then code the population using a 1 for those who would answer "yes" and 0 for those who would answer "no". For example, if population of 10,000 is split 60-40 in favor, the population could be realized on the computer with

```
> the.pop = rep(c(0, 1), c(4000, 6000))
```

(The `rep()` command repeats the values in the first argument as specified by the second argument. In this case 4,000 `0`'s and then 6,000 `1`'s.)

A random sample with replacement (of size 10) can be found as follows

```
> the.sample = sample(the.pop, 10, replace = TRUE)
> the.sample

 [1] 0 0 0 0 0 0 0 1 0 1
```

The sample mean and median would then be

```
> mean(the.sample)

[1] 0.2

> median(the.sample)

[1] 0
```

Question 1: Define the variable `the.pop`, as above, and find a sample of size 10. For your sample, find the mean, median, standard deviation, and IQR.

Question 2: From the population specified by `the.pop` find a sample of size 100. For your sample, find the mean, median, standard deviation, and IQR.

Question 3: Could you have switched your answers to the two previous questions and not have been able to tell? Discuss

Question 4: For many real-world data sets, for example tax returns or cash register receipts, the first digit is more often a 1, than a 2, which in turn is more often than a 3 etc. Benford's law gives a precise distribution for this population. Samples from this distribution can be formed by assigning probabilities to the possible digits:

```
> n = 10
> sample(1:9, size = n, replace = TRUE, prob = log(1 + 1/(1:9)))

 [1] 8 3 2 2 1 8 2 9 3 3
```

Take a sample of size 100 from this population and find the sample mean. Is it close to 5?

2.2 for loops

A single value of a statistic can be used for statistical inference, but only if you have an idea of the sampling distribution of the statistic. Assuming we don't know this by some other means, we can run a simulation to investigate it. The key is efficiently repeating the sampling procedure.

A `for()` loop is, perhaps, the most familiar way to repeat something on a computer. For instance, the following will define a variable to store our results, `res`, and then do the sampling `m=100` times for a sample of size `n=10`.

```
> res = c()
> m = 25
> n = 10
> for (i in 1:m) {
+     x = sample(the.pop, n, replace = TRUE)
+     res[i] = mean(x)
+ }
> plot(density(res))
```

The first three lines defines some terms so that you can tell where they go in the rest of the command. The second line starts the `for()` loop. The basic template for a `for()` loop is

```
for (i in vec) {
  run these commands
}
```

(It can be more general though.) The `for()` loop runs the commands in the braces as many times are there are items in the data vector `vec`. Each time, the value of `i` changes to the next value of `vec`. For the example, `vec` is `1:m` or the integers starting at `1` and ending at the value of `m`. For each of these values, a sample is taken, its mean computed and then assigned to the `i`th value of `res` for later usage. In this case, the later usage is a density plot resulting in Figure 1.

For reference, the `for()` loop could have been written in one line (thereby eliminating the need for the braces) with

```
> for (i in 1:m) res[i] = mean(sample(the.pop, n, replace = TRUE))
```

Question 5: For the same finite population stored in `the.pop`, perform a simulation with $n = 10$ and $m = 250$. Make a density plot.

Then repeat using $n = 25$. Add a density plot of the new sample to the old one with the command

```
> lines(density(res))
```

Compare the two distributions in the resulting plot. Do they have the same center? The same spread? The same shape?

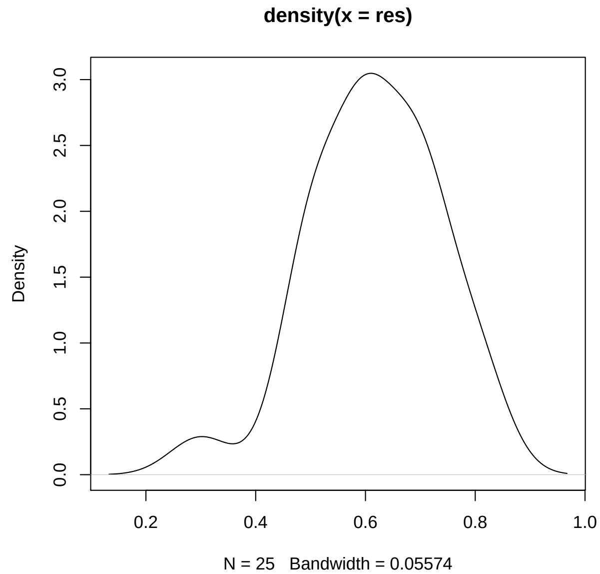

Figure 1: Sampling distribution of $\hat{p}$ for 60-40 population and $n = 10$

2.3 Automating the task

This project comes with a function to help visualize simulations. To install it, enter the following command.

```
> source("http://www.math.csi.cuny.edu/st/R/showSamplingDist.R")
```

This function will illustrate samples for various sizes from a specified, finite population. Different statistics may be used.

For example, to see it work issue these commands (first installing a data set on eruptions at Old Faithful)

```
> data(faithful)
> the.pop = faithful$eruptions
> showSamplingDist(the.pop)
```

For various values of n, a random sample is repeatedly taken and the sample mean is found. A density of the repeated sample means is drawn. To keep track of the different values of n, different colors are used in the graphic.

Question 6: Describe in words what happens to the graphs as the value of n gets larger.

Question 7: The value of n may be specified using the argument `the.steps=`, for instance

```
> showSamplingDist(the.pop, the.steps = c(2, 10, 25, 50, 100))
```

Make the graphic with the step sizes c(4,16,64,256). When you quadruple the size of n what happens to the spread?

Question 8: Does the final shape of the sampling distribution depend on the shape of the population? Let's investigate for the mean. Do simulations for these two populations: one population of 20,000 evenly split on a proposal, and another of 20,000 with 19,000 for and 1,000 against.

Run the simulations and describe how the shape of the sampling distribution for the largest value of n is similar for the two and how they are different, if they are.

Question 9: You may do the simulation for other statistics than the sample mean. For instance, the sample median will be found for each sample in the simulation with

```
> showSamplingDist(the.pop, statistic = "median")
```

Does the same shape happen for the sample median as it does for the sample mean?

Question 10: Perform a simulation using the standard deviation for the statistic. Again, describe the sampling distribution of the sample standard deviation as n gets larger.

Question 11: Based on the above simulations, make a statement about the center, spread, and shape of the sampling distribution as n gets large.

3 Biased estimates

Most statistics used to estimate a parameter are unbiased, or asymptotically unbiased. An *unbiased* statistic is one whose sampling distribution is centered (the mean) on the population parameter. An *asymptotically unbiased* statistic has this property as n gets big.

A *biased* estimator that is not asymptotically unbiased can be an issue with statistical inference. In popular opinion polls, such as political polls, there are many places where bias can enter ranging from bias due to the use of phone interviews, non-response bias, and bias in the wording of the question. Due to these biases, polls with similar intents can produce differences not readily explained by random variation.

Question 12: Explain more fully how the three listed biases actually introduce potential biases.

Question 13: Describe how bias can explain the difference between your weight at home and your weight at the doctor's office.

Question 14: In Moore and McCabe, the statistic $\hat{p} = (X+2)/(n+2)$ is used instead of the unbiased statistic $\hat{p} = X/n$. Perform a simulation showing that the biased statistic is asymptotically unbiased. First define the statistic, as follows:

```
> phat = function(x, n) (x + 2)/(n + 4)
```

Then a simulation can be performed with

```
> showSamplingDist(the.pop, statisitic = phat)
```

Run a simulation for a population that is split 500 for and 500 against a proposition. Is $\hat{p}$ asymptotically unbiased?

4 Continuous populations

There are a number of continuous distributions that have built-in functions in R for generating random samples. Common distributions are the normal distribution, the t-distribution, the chi-squared distribution, the exponential distribution, and the uniform distribution. To use these functions you combine the letter `r` and the family name, which is `norm`, `t`, `chisq`, `exp`, and `unif` for the distributions just named. Each distribution has a number of parameters that may need specification.

For example, to draw 5 samples from the normal distribution with mean 400 and standard deviation 20 we have

```
> rnorm(5, mean = 400, sd = 20)

[1] 409.5 407.4 385.4 387.8 394.3
```

(The arguments set the population mean and standard deviation.)

To draw 7 samples from the exponential distribution with rate 10 (the reciprocal of the mean) we have

```
> rexp(7, rate = 10)

[1] 0.11163 0.09692 0.01257 0.07377 0.32425 0.03987 0.21861
```

In order to use `showSamplingDist()` with a continuous population we can take a big sample, say of size 1,000 from the continuous population and use this as a finite population with similar properties. For instance

```
> the.pop = runif(1000, min = 0, max = 10)
> showSamplingDist(the.pop)
```

Question 15: Look at the sampling distribution of the mean for each of the following finite populations which illustrate a continuous parent population:

1. `rnorm(1000, mean=400, sd=20)`
2. `rt(1000, df=3)`
3. `rexp(1000, rate=1/20)`

How are the sampling distributions similar? How are they different, if at all?