Bootstrapping

Thomas Scofield

February 26, 2021

Setting 1: Estimating μ

You want to know μ, the mean of some quantitative variable (perhaps healthy body temperature) in a population. Knowing the sample mean is an **unbiased estimator** for μ, you take a random sample of size n, and compute the sample mean $\overline{x}$.

Q1: The **BodyTemp50** data frame contains three variables, `BodyTemp`, `Pulse`, and `Gender`, for a sample of $n = 50$ people. Write an R command that computes $\overline{x}$, the sample mean body temperature from this data.

The $\overline{x}$ you compute serves as a **point estimate** for μ, but being unbiased is no guarantee that your sample mean is exactly equal to the population mean. In fact, that rarely happens. So, it is better to give an interval of values in which you think μ may lie, a 95% 95% confidence interval, perhaps. Yesterday, we discussed one approach, the *centered interval* construction method, which obtains its lower and upper bounds via the calculation

$$(\text{point estimate}) \pm \text{ME}.$$

So long as the sampling distribution of $\overline{X}$ is a **normal** distribution, we obtain approximately 95% coverage in our interval by taking

$$\text{ME} = 2 \times \text{SE}_{\overline{x}}.$$

Estimating SE: ideal, but impractical

The number we need, $\text{SE}_{\overline{x}}$, is the standard deviation of the sampling distribution for $\overline{X}$. So far, this is all we have done:

1. draw a random sample of size n from the population
2. compute $\overline{x}$ from our sample

Doing these steps *once* cannot provide a picture of what sample means are possible, nor their relative frequencies—i.e., the **sampling distribution** of $\overline{X}$. If, however, we *repeated* the steps above many (perhaps several thousand) times, we could use the resulting list of $\overline{X}$-values both to

- confirm our sampling distribution appears normal, and
- compute their standard deviation, which should fairly accurately approximate $\text{SE}_{\overline{x}}$.

This sounds like a great idea, until you consider the difficulty of repeatedly drawing random samples from the population.

Estimating SE: using bootstrapping

In bootstrapping, we mimic the ideal approach above, but with these key differences:

- We use our sample data, already collected, as a stand-in for the population.
- We sample from that data *with replacement*, so the resulting $\overline{X}$-values display variability. The list of values drawn in the sample comprise a _____ and the $\overline{x}$-value computed from that sample is known as a _____

Since the values of $\overline{x}$ we generate arise from using a stand-in population rather than the population itself, we call the distribution of values a **bootstrap distribution**.

Q2: Carry this out in StatKey, selecting the **BodyTemp50** data from the drop-down menu. What is the resulting 95% confidence interval?

In RStudio, we might do the same thing this way.

```
manyXbars <- do(5000) * mean(~BodyTemp, data=resample(BodyTemp50))
```

Setting 2: Estimating p

You want to know p, the proportion of "successes" in a population for a binary categorical variable whose two values are described generically as "success" and "failure". We have seen that $\widehat{p}$ is an unbiased estimator of p. As before, it makes sense to give a confidence interval for p, computing lower and upper bounds as

$$(\text{point estimate}) \pm \text{ME},$$

and using $\widehat{p}$ as the point estimate.

Q3: Say we wish to construct a 95% confidence interval for p. Describe an ideal, but impractical approach to coming up with a reasonable margin of error ME.

The idea of bootstrapping in this setting can be thought of as repeatedly carrying out these steps:

- Place slips of paper marked "S" and "F" in a bag, so that the overall proportion of "S" in the bag matches $\widehat{p}$, the sample proportion. Mix well.
- Draw n times from the bag, with replacement, to obtain a **bootstrap sample** comprised of a list of "S"'s and "F"'s drawn.
- Calculate $\widehat{p}$, the **bootstrap statistic**, the proportion of "S"'s in your bootstrap sample.

Q4: Suppose, in a random sample of $n = 200$ people, there are 23 whose dominant hand is their left. (We generically consider *left*-handedness a "success" here, as it is the trait we are looking for.) Use StatKey to estimate $\text{SE}_{\widehat{p}}$. Then write a 95% confidence interval for p, the true proportion of left-handed people.

We used StatKey to generate a **bootstrap distribution** for $\widehat{p}$, but the same can be done in RStudio. The command

```
rflip(200, prob=23/200)
```

```
##
## Flipping 200 coins [ Prob(Heads) = 0.115 ] ...
##
## T T T T T T T T H T T H T H T T H T T T T T T T H T T T H T T H T T T
## T T T T T H T T T T T T T T T T T T T T T T T T T T H T T T T T T T H
## T T T H T T T T T T T T T H T T T T T T T T T T T H T T T T T T T T H
## T T T T H H T T T T T T T T T T T T T T T T T T H T T T T T T T T T T
## T T T T T T T T T T T T T T T T T T T T T T T T T T T T T T T T H T
## T T T T T T T T T H T T T H T T T T
##
## Number of Heads: 20 [Proportion Heads: 0.1]
```

generates both a single bootstrap sample and its corresponding bootstrap statistic. The command evokes an image of using n coin flips, instead of drawing n times from a bag, where the coin has been appropriately weighted. The resulting bootstrap statistic $\widehat{p}$ is calculated from the viewpoint that an "H" is a success.

We obtain the bootstrap distribution by repeating this process many times:

```
bStrapTrials <- do(5000) * rflip(200, prob=23/200)
```