Bootstrap

Stat 145

Big Idea

A **sampling distribution** is the distribution of a statistic (number computed from a sample) over all possible samples. The variability of an estimate from sample to sample tells us how precise the estimate is. The problem is that we only ever have one sample, so we can't see the sampling distribution except in illustrative illustrations and simulations.

The Big Idea of the Bootstrap is to approximate a sampling distribution by sampling from our sample instead of sampling from the population. If our sample is a good representation of the population, then this should be a pretty good approximation.

A Wild Sample

In this video, Chris Wild uses a small data set to illustrate how the bootstrap. This data set is too small for bootstrap to work well, but it is nice to be able to see everything that is going on.

First, let's do this the wrong way – using `shuffle()`.

1. Give it a try using `shuffle()` on the data set `WildSample` in the `CalvinData` package.

   Run the R code below several times. What changes each time you run it? What doesn't change? Why is that a bad thing for our purposes?

```
library(CalvinData)
WildShuffled <- shuffle(WildSample)
WildShuffled
```

```
## # A tibble: 8 x 5
##   name     sex     age BPSysAve orig.id
##   <chr>    <chr> <dbl>    <dbl> <chr>
## 1 Romeo    M        49       86 1
## 2 Cloe     F        18       87 8
## 3 Gustav   M        38      100 5
## 4 Michelle F        44      130 6
## 5 Alfred   M        62      124 7
## 6 Janette  F        26      125 2
## 7 Bravo    M        41      134 4
## 8 Fred     M        56      120 3
```

```
mean(~BPSysAve, data = WildShuffled)
```

```
## [1] 113.25
```

```
df_stats(~BPSysAve, data = WildShuffled, mean_BP = mean)
```

```
##   response mean_BP
## 1 BPSysAve  113.25
```

   Note that you can use either `mean()` or `df_stats()` to compute the mean.

In order to create a bootstrap distribution, we must sample from our sample with replacement (allowing some cases to be selected multiple times and some to be skipped). "Sampling from our sample with replacement" is a mouthful, so we'll use the term **resampling** for this. And the R function for this is called `resample()`

2. Repeat the previous exercise using `resample()` in place of `sample()`. Why does this work better than `shuffle()`?

3. Now create a **bootstrap distribution** using `do()`. Use your bootstrap distribution to create a histogram of the resampled means. Here's some code to get you started.

```
library(CalvinData)
Wild_bootstrap <- do(1000) * mean(~BPSysAve, data = resample(WildSample))
```

4. Repeat the exercise above using `df_stats()` instead of mean. Which do you prefer?

NHANES

Let's try a slightly bigger example. We'll pull 12 rows from the larger NHANES data set. That data set as some missing values for pulse and blood pressure. The code below makes sure that the 12 rows we pick have values for both.

```
library(NHANES)
set.seed(123)
NHANES12 <-
  NHANES %>%
  filter(!is.na(BPSysAve), !is.na(Pulse)) %>% # make sure BP and pulse are not missing
  sample(12) # get 12 observations from NHANES
```

5. Create a histogram of a bootstrap distribution for the mean `BPSysAve` using NHANES12.

6. Create a histogram of a bootstrap distribution for the mean `Pulse` using NHANES12.

7. If we used a sample of size 50 instead of a sample of size 12, how would you expect the bootstrap distributions to change? Try it and see if you are right. Call your new data set `NHANES50`. (Mimic the code above to make sure you don't have any missing values for pulse and blood pressure.)

8. The actual NHANES data set is quite large.

   a. How many rows does it have?
   b. What do you expect the boot strap distributions to look like if we use all of `NHANES`?
   c. Create bootstrap distributions for the mean `BPSysAve` and the mean `Pulse` using the fall `NHANES` data set to see if you are right. (Note: now you will have to deal with the problem of missing values. `df_stats()` will take care of this for you. If you use `mean()`, you will need to include `na.rm = TRUE` to tell R that you want the mean of the non-missing values.)

Confidence Intervals

There are several ways to use a bootstrap distribution to create an interval estimate, a range of plausible values for our estimand. Here are two simple methods.

Percentile Method

The percentile method uses the middle 95% of the bootstrap distribution to create what is called a **95% confidence interval**. The function `cdata()` will compute this for us. `cdata()` is short for "center of the data". For example,

```
cdata( ~ mean, data = Wild_bootstrap, p = 0.95)
```

```
##         lower  upper central.p
## 2.5% 100.1219 124.25      0.95
```

9. For each bootstrap distribution you have created, go back and create a 95% confidence interval using the percentile method.

Standard Error Method

If the bootstrap distribution looks approximately normal, we could use the fact that the middle 95% should be within two standard deviations to either side of our estimate. For example,

```
xbar <- mean(~BPSysAve, data = WildSample); xbar
```

```
## [1] 113.25
```

```
SE <- sd( ~ mean, data = Wild_bootstrap); SE
```

```
## [1] 6.331157
```

So our interval would be 113.25 plus or minus `2 * 6.33`.

```
xbar - 2 * SE
```

```
## [1] 100.5877
```

```
xbar + 2 * SE
```

```
## [1] 125.9123
```

Notes

- We are taking the standard deviation of our bootstrap distribution, not of the original data.

  To avoid confusion between these to standard deviations, the standard deviation of a sampling distribution or a bootstrap distribution (or a null distribution) is called the **standard error**. When our estimate is a mean, this would be called the **standard error of the mean**, if we want to be really precise.

- Use the estimate from your data, not the mean of the bootstrap distribution.

  The purpose of the bootstrap is to estimate the amount of sampling variability, not the to give a better single number estimate. The mean of the bootstrap distribution will usually be quite close to the estimate from our sample, but it not a better estimate. In fact, it just adds some additional imprecision from the resampling process.

10. Look at a histogram of the bootstrap distribution for this example. Does it look approximately normal? (Usually, we should do this step first, not after creating the SE interval.)
11. Go back and create an 95% SE confidence interval for each of the examples above where the bootstrap distribution is approximately normal. (The distribution should at least be unimodal and approximately symmetric.)

Comparing the two methods

The two methods do not produce exactly the same interval, but they are usually quite similar – provided the bootstrap distribution is approximately normal. Other, more complicated, methods also exist and are even better, but we won't cover those in this class.

Other Estimands

What if you wanted to create an interval estimate for something other than the mean. Perhaps you are interested in how much pulse *varies*. The standard deviation would be a good way to measure variability.

12. Using your `NHANES50` data set, create a 95% percentile confidence interval for the standard deviation of pulse.

13. Using your `NHANES50` data set, check whether the bootstrap distribution looks approximately normal. If it does, create a 95% SE confidence interval for the standard deviation of pulse.

Review

Creating bootstrap distributions is pretty easy. We just compute our statistic over and over using resamples. The main commands we need in R are

- `resample()` to sample from our sample with replacement.
- a function to compute our estimate (`mean()` or `sd()` or `df_stats()` or something else)
- `do()` do repeat the process many times
- `gf_histogram()` or some other plotting function to see what the distribution looks like

Putting that all together, a template for creating a bootstrap distributon looks something like this:

```
do(_______) * estimate(________, data = resample(________))
```

where `estimate()` is replaced by some function that computes your estimate from data. The other blanks are filled in with the number of resamples to compute, a formula describing the variables involved, and the name of your data set.

To create a percentile confidence interval, we can use

- `cdata()` to find the middle 95% of the bootstrap distribution.

To create a standard error confidence interval, we can use

- `sd()` to compute the standard error and then do a bit of arithmetic.

14. Where will a bootstrap distribution be centered? Why?
15. Where will a randomization distribution be centered? Why?

Looking ahead

Creating confidence intervals from a bootstrap distribution is straightforward, once you have the bootstrap distribution. So where do we go from here? Here are some things we will be learning:

- How to create bootstrap distributions for other estimands (a proportion, the difference between two proportions, the difference between two means, the correlation coefficient, the slope of a regression line, etc., etc.).
- How to create confidence intervals (in a some special situations) without using a bootstrap distribution.
- How to understand what the "95%" means when we say we have a 95% confidence interval (and what would change if we wanted a 90% confidence interval or a 98% confidence interval).
- How confidence intervals and hypothesis tests are related.

Give it a try

15. If you have extra time, here are some things you can try using the NHANES data.
    a. Create a 95% confidence interval for the mean of some other quantitative variable.
    b. Create a 95% confidence interval for the proportion of some categorical variable in the NHANES data set. (Try `Smoke100`, for example, to estimate what proportion of Americans have smoked at least 100 cigarettes in their life.)
    c. See if you can figure out how to create a 95% confidence interval the *difference* in the mean pulse for men and women using the NHANES data.

Be mindful of potentially missing data. (One reason for missing data is that not all subjects were asked all questions. For example, younger subjects were not asked about smoking, men were not asked about pregnancies, etc.)