# Intro to Bootstrapping

Goal: to construct a confidence interval for a parameter in which either (a) the population distribution is not known or (b) the distribution of the statistic is not known. Or both (a) and (b).

Classical Approach: The classical approach requires that one state the population distribution. In many cases, one can then figure out what pdf a statistic from that population would follow. With this information, one can figure out an interval in which covers the true parameter with a given confidence level.

Why you might want it: The classical approach is clearly impossible if one does not know -- or is not willing to make an assumption about -- the pdf of the population. This might happen for any number of reasons. Circumstances in which you might find yourself where you are unwilling to do this:

the sample size is too small to check any distributional assumptions you might make

the sample size is too small for the Central Limit Theorem to "kick in" (applies only to linear estimators)

the estimator is nonlinear and its pdf is unknown (even if the population's pdf is known.)

Sometimes the pdf of the estimator (called the sampling distribution) is known, but requires other assumptions that might or might not be true. The bootstrap can sometimes act as a "check" to see if the estimator is behaving as it should.

Data Set: To illustrate, we'll use the first experimental run from Michelson & Morley's speed-of-light assessments. Here are the data:

```
> speed
 [1]  850  740  900 1070  930  850  950  980  980  880 1000
980  930  650  760
[16]  810 1000 1000  960  960
> summary(speed)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    650     850     940     909     980    1070
```

First we'll estimate the mean and compare to classical results. Next we'll try something a little more exotic: the log of the mean. Finally, we'll estimate the median. (Note: the "Median" and "Mean" given above are the mean and median of the "sample". Here, the population consists of all possible values their measurements would produce. The mean of this population is (hopefully) the "true" value.)

General Idea:

If we do not know the distribution of the population, then our best guess at the distribution is provided by the data. The main idea in bootstrapping is that we (a) pretend

that the data is the population and (b) take samples from this pretend population (which we call "resamples").

Pretending that the sample is the population means that we are interested in the frequency with which the observed values occurred. Thus, if the value 930 occurs in 2% of the sample, we want it to occur in 2% of our resamples, on average.

This is done by sampling *with replacement*. This means that the resamples can be done as follows: write each observation on a ticket. Put all tickets in a hat. Shuffle, and draw one out. Now put it back and repeat.

A "resample" is defined as follows: a sample taken with replacement from the original set of observations and consisting of the same size as the original set.

From the re-sample, we calculate the statistic we're interested in. This is called a "boostrap statistic." We store this value.

Next: repeat the above steps so that you collect a large number (M) of bootstrap statistics.

The general idea is that the relationship of the boostrap statistics to the observed statistic is the same as the relationship of the observed statistic to the true value. That is, in SAT lingo:
bootstrap stat :: observed stat as observed stat::true value.

This means that by studying our collection of boostrap statistics, we learn something about how far off our observed statistic might be from truth.

Example:

Here's some R code that generates a collection of bootstrap means:

```
bstraps <- c()
for (i in 1:1000) {
        bsample <- sample(speed, length(speed), replace=T)
        bstraps <- c(mean(bsample), bstraps)}
```

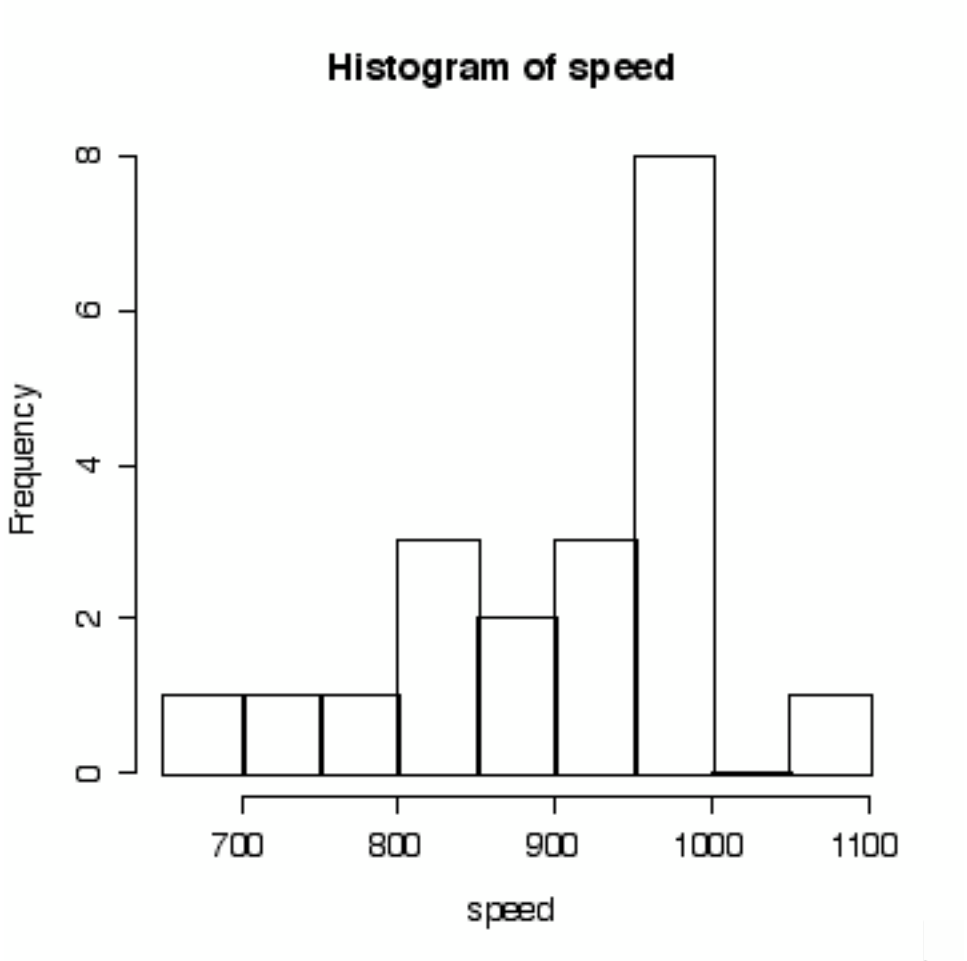
Here are the first 10 bootstrapped means:

```
> bstraps[1:10]
 [1] 890.0 902.0 949.0 926.0 918.0 914.0 887.5 932.0 950.5
916.5
```

We'll compare these, later, to our observed value: 909.

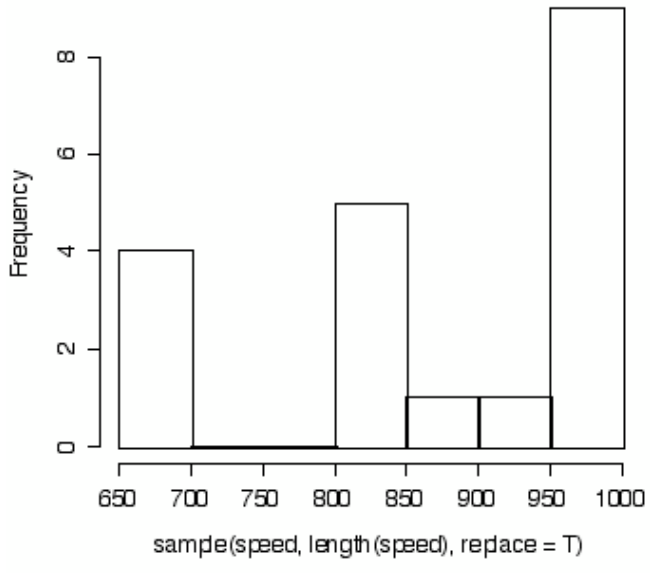
Before going on, let's look at some histograms.

The observed data look like this:

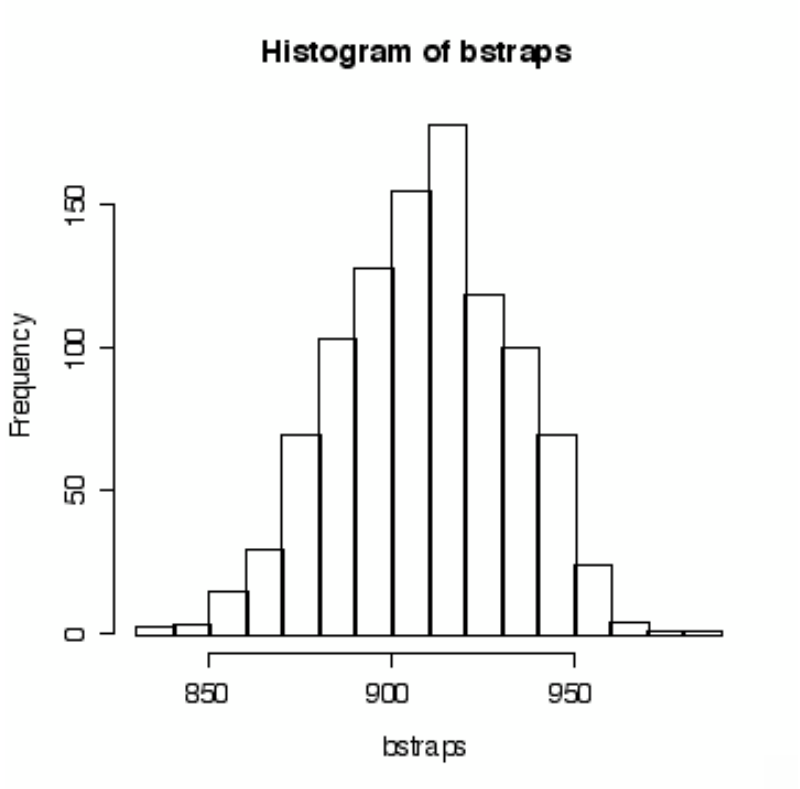



The histogram of a single re-sample looks like this:

It looks similar to, but not exactly the same as, our original data.



Finally, here's a histogram of all 1000 boostrap statistics. Note that, not surprisingly (because of the Central Limit Theorem) it looks pretty Normal.




There are two methods for generating bootstrap confidence intervals. The first is fairly intuitive (assuming you have intuited the general philosophy) and the second "fixes" some shortcomings of the intuitive method that arise in some situations.

I. The Percentile Method

The percentile method is this: say you want a (1-alpha)*100% confidence interval. Then generate lots of bootstrap statistics and look at the histogram. Find the points that cut-off the bottom (alpha/2)*100% and the top (alpha/2)*100%. That's your confidence interval.

Let's apply this to the problem of finding a 95% confidence interval for the mean. I'm going to re-do my bootstrapping.

Now for a 95% CI, alpha = .05, so we want to find the values that "cut off" the lower 2.5% and the upper 2.5% of our data. Since I did 1000 bootstraps, this means I find the value that divides the bottom 25 values and the value that cuts off the upper 25 values (since 25 is 2.5% of 1000).

```
> sorted <- sort(bstraps)
```

```
> sorted[c(25, 975)]
[1] 862.0 949.5
```

Thus, my 95% CI is (862, 949.5).
(Note: I'm being a little sloppy here. Strictly speaking, I want the value that has 2.5% of the observations BELOW it, and the 25th observation here has 2.4% below it. If I wanted to be very careful, I could use the average of the 25th and 26th observations. A similar argument applies to the upper end-point.)

Let's compare this to the "classical" method. In this method, we would use the formula
xbar +- t(n-1) * s/sqrt(n)
xbar = mean(speed) = 909
n = length(speed) = 20
t(n-1) = qt(.975, 19) = 2.09
s = sd(speed) = 104.9

So the 95% CI is (860.0, 958.0) which is slightly wider.

One thing to note: bootstrap confidence intervals vary. If I redo my bootstrapping, I will get a slightly different interval. This variability can be reduced by using more bootstrap samples. Often, 1000 is sufficient for most applications, but 10,000 is even better if you have the time.

Now let's imagine that we wanted to find a confidence interval for the standard-deviation of this population. An common estimate is the standard-deviation of the sample, given by the square root of the sum of the squared deviations from the average, divided by (n-1). For these data, this is s = sd(speed) = 104.926.

As it turns out, there is a formula for the confidence interval, but it relies fairly strongly on distributional assumptions. Suppose we didn't know it? Then we could bootstrap it using almost exactly the same commands:

```
bstraps <- c()
for (i in 1:1000) {
        bsample <- sample(speed, length(speed), replace=T)
        bstraps <- c(sd(bsample), bstraps)}
sort(bstraps)[c(25, 975)]
```

The results is (66.5, 134.9).

Final Example: Bootstrapping the median.

There is a theoretical "large sample" confidence interval for the median. One can show that the sampling distribution of the median (that is, the pdf of the median of a sample) approaches the normal distribution with large sample size. The mean is the true value of

the median, and the SD is the square root of (1/4n) $f(m)^2$ where f(m) is the value of the pdf at the true value of the median.

A problem, then, is that to calculate the CI, we need to know f(m). But this is unknown, and must also be estimated. Estimation of a density is part art, and requires some experimenting. But even if done well, this introduces additional error into our estimation routine. The bootstrap avoids this:

```
bstraps <- c()
for (i in 1:1000) {
        bsample <- sample(speed, length(speed), replace=T)
        bstraps <- c(median(bsample), bstraps)}
sort(bstraps)[c(25,975)]
(865, 980)
```

Part II: Bias Correction

Some estimates are biased, and if so, the bootstrap procedure above tends to magnify this. A correction to this is called the "Bias Corrected" bootstrap. It's based on the principle that, while the bootstrap estimates might be biased, one can still get an estimate of this bias by examining the differences between the bootstrap estimates and the observed estimate. The basic idea makes use of a theoretical result that makes use of the normal distribution.

The idea is to still choose percentiles from the collection of bootstrap estimates. But rather than choose the alpha/2 and (1 - alpha/2) percentiles, we instead use corrected percentiles. These are somewhat tricky (but not terribly so) to compute. So I recommend you use pre-packaged software.

R

R has many bootstrap routines built-in. The problem is that at the time of writing I haven't been able to get them to work reliably. So stay tuned...

References:
There are many, many references on this. But a fairly accessible one to a general, science-oriented reader is
An Introduction to the Bootstrap, Bradley Efron, Robert J. Tibshirani, Chapman & Hall