Bootstrapping

The bootstrap is a newly invented statistical technique for both understanding distributions of statistics and for calculating p-values (a p-value is a the probability that a scientific claim is incorrect). It was invented here at Stanford in 1979 when mathematicians were just starting to understand how computers, and computer simulations, could be used to better understand probabilities.

The first key insight is that: if we had access to the underlying distribution (F) then answering almost any question we might have as to how accurate our statistics are becomes straightforward. For example, in the previous section we gave a formula for how you could calculate the sample variance from a sample of size n. We know that in expectation our sample variance is equal to the true variance. But what if we want to know the probability that the true variance is within a certain range of the number we calculated? That question might sound dry, but it is critical to evaluating scientific claims! If you knew the underlying distribution, F, you could simply repeat the experiment of drawing a sample of size n from F, calculate the sample variance from our new sample and test what portion fell within a certain range.

The next insight behind bootstrapping is that the best estimate that we can get for F is from our sample itself! The simplest way to estimate F (and the one we will use in this class) is to assume that the $P(X = k)$ is simply the fraction of times that k showed up in the sample. Note that this defines the probability mass function of our estimate $\hat{F}$ of F.

```
def bootstrap(sample):
   N = number of elements in sample
   pmf = estimate the underlying pmf from the sample
   stats = []
   repeat 10,000 times:
      resample = draw N new samples from the pmf
      stat = calculate your stat on the resample
      stats.append(stat)
   stats can now be used to estimate the distribution of the stat
```

Bootstrapping is a reasonable thing to do because the sample you have is the best and only information you have about what the underlying population distribution actually looks like. Moreover most samples will, if they're randomly chosen, look quite like the population they came from.

To calculate $\text{Var}(S^2)$ we could calculate S_i^2 for each resample i and after 10,000 iterations, we could calculate the sample variance of all the S_i^2s. You might be wondering why the resample is the same size as the original sample (n). The answer is that the variation of the variation of stat that you are calculating could depend on the size of the sample (or the resample). To accurately estimate the distribution of the stat we must use resamples of the same size.

The bootstrap has strong theoretic grantees, and is accepted by the scientific community. It breaks down when the underlying distribution has a ``long tail" or if the samples are not I.I.D.

Example of p-value calculation

We are trying to figure out if people are happier in Bhutan or in Nepal. We sample $n_1 = 200$ individuals in Bhutan and $n_2 = 300$ individuals in Nepal and ask them to rate their happiness on a scale from 1 to 10. We measure the sample means for the two samples and observe that people in Nepal are slightly happier--the difference between the Nepal sample mean and the Bhutan sample mean is 0.5 points on the happiness scale.

If you want to make this claim scientific you should calculate a p-value. A p-value is the probability that, when the null hypothesis is true, the statistic measured would be equal to, or more extreme than, than the value you are reporting. The null hypothesis is the hypothesis that there is no relationship between two measured phenomena or no difference between two groups.

In the case of comparing Nepal to Bhutan, the null hypothesis is that there is no difference between the distribution of happiness in Bhutan and Nepal. The null hypothesis argument is: there is no difference in the distribution of happiness between Nepal and Bhutan. When you drew samples, Nepal had a mean that 0.5 points larger than Bhutan by chance.

We can use bootstrapping to calculate the p-value. First, we estimate the underlying distribution of the null hypothesis underlying distribution, by making a probability mass function from all of our samples from Nepal and all of our samples from Bhutan.

```
def pvalue_bootstrap(bhutan_sample, nepal_sample):
   N = size of the bhutan_sample
   M = size of the nepal_sample
   universal_sample = combine bhutan_samples and nepal_samples
   universal_pmf = estimate the underlying pmf of the universalSample
   count = 0
   observed_difference = mean(nepal_sample) - mean(bhutan_sample)
   repeat 10,000 times:
      bhutan_resample = draw N new samples from the universalPmf
      nepal_resample = draw M new samples from the universalPmf
      mu_bhutan = sample mean of the bhutanResample
      mu_nepal = sample mean of the nepalResample
      mean_difference = |muNepal - muBhutan|
      if mean_difference > observed_difference:
         count += 1
   pvalue = count / 10,000
```

This is particularly nice because nowhere did we have to make an assumption about a parametric distribution that our samples came from (ie we never had to claim that happiness is gaussian). You might have heard of a t-test. That is another way of calculating p-values, but it makes the assumption that both samples are gaussian and that they both have the same variance. In the modern context where we have reasonable computer power, bootstrapping is a more correct and versatile tool.