STAT 2150 Statistics and Computing

Unit 6: Resampling

Keith Uzelmann

Winter Term 2022

Section 1

Bootstrapping

Introduction

- Determining the sampling distribution of a sample statistic is one of the most important tasks in statistics. This has been a major focus of Units 4 and 5.
- However, you may observe that our inferential techniques in each case require knowledge of the parametric form of the population (that is, we have been pursuing parametric methods).
    - For example, we may assume that our population is N(μ, σ) and leave μ and σ to be estimated from data, or we may assume that data is Exp(λ) and leave λ to be estimated from data.
    - Further, from these assumptions we can then determine (either theoretically or through simulation) the sampling distribution of our sample statistic of choice (usually an estimator).
- This is a limiting perspective, especially because in practice *we often do not know with certainty the parametric form of our data*.
- A modern solution to this problem is bootstrapping.

What is Bootstrapping

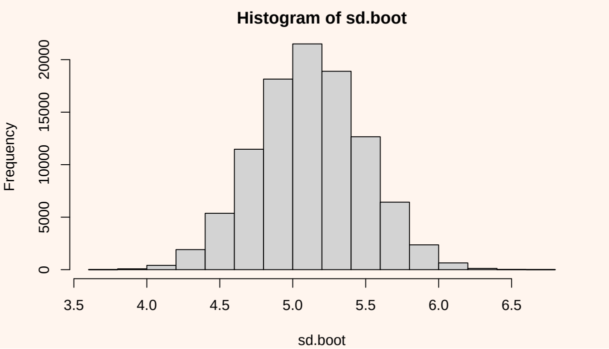
- Bootstrapping is a technique introduced by Bradley Efron in 1979. In essence, the sample data is treated as a so-called surrogate population, and then simulation is done on this surrogate population.
- Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ be an i.i.d. sample f size n from a population $F(x)$, and let $\hat{\theta} = T(\mathbf{X})$ be a sample statistic (such as $\bar{X}$ or any sort of estimator). Our goal is to determine the sampling distribution of $\hat{\theta}$.
- The steps of the bootstrap are as follows:
  1. Draw a sample $\mathbf{x}^*$ *with replacement* of size n from $\mathbf{x}$. This is called a resample.
  2. Calculate $\hat{\theta}^* = T(\mathbf{x}^*)$.
  3. Repeat steps 1 – 2 many times, obtaining a series of outputs $\hat{\theta}^*_1, \hat{\theta}^*_2, \ldots \hat{\theta}^*_B$
- When you are done, the vector $\left(\hat{\theta}^*_1, \hat{\theta}^*_2, \ldots \hat{\theta}^*_B\right)$ will be like simulated values of $\hat{\theta}$. Thus, we have obtained an approximation of the sampling distribution of $\hat{\theta}$.

Bootstrapping Example

- For example, let's load in the `Hospital_200` dataset.
- This dataset contains hospital data for 200 patients admitted to New York state hospitals follow a myocardial infarction. We are in particular interested in `LOS`, the length of stay for each patient.
- Our goal is to determine the standard deviation of this population, and we will use the sample standard deviation S to estimate this.
- However, it is also important to know the sampling distribution of S, so that we can better understand any bias or error incurred through the use of S.
- We will use the bootstrap to determine the sampling distribution of the sample standard deviation.

Bootstrapping Example (Continued)

```
Hospital_Sample = read.csv("~/R_Datasets/Hospital_200.csv")
sd.boot = c()
for(b in 1:100000)
{
  data.boot = Hospital_Sample[sample(1:200, 200, replace = TRUE), ]
  sd.boot[b] = sd(data.boot$LOS)
}
hist(sd.boot)
```



Using the Bootstrap

- In practice, we often do not use the bootstrap to estimate the entire sampling distribution of a sample statistic. Instead, we use it to estimate certain characteristics of that sample statistic.
- We will focus the three main applications of the bootstrap in this course.
  1. Bias Estimation,
  2. Variance Estimation,
  3. Confidence Interval Construction.
- We will find that the bootstrap does quite a good job when estimating these characteristics (especially considering the fact that we are, again, making NO assumptions on the form of the population!)

Using the Bootstrap for Bias Estimation

- We will start by describing the process for using the bootstrap to estimate the *bias* of an estimator. The application of this is somewhat ingenious.
- Recall that, in practice, the bias of an estimator is given by

$$\text{bias}_\theta(\hat{\theta}) = \mathbb{E}\left[\hat{\theta}\right] - \theta.$$

- When doing the bootstrap, we are treating the sample like a population. Thus, we can actually estimate the bias of $\hat{\theta}$ by

$$\text{bias}_{\hat{\theta}}\left(\hat{\theta}^*\right) = \mathbb{E}\left[\hat{\theta}^*\right] - \hat{\theta}.$$

  where $\hat{\theta}^*$ is the *bootstrap distribution*, and $\hat{\theta}$ is the value of the estimator for the given sample.
- Since the bootstrap results in many observations from $\hat{\theta}^*$, we can use the mean of these values $\overline{\hat{\theta}^*}$ in place of $\mathbb{E}[\hat{\theta}^*]$.

Bootstrap Bias Estimation: Example

- Below we use the bootstrap to estimate the bias of S from the previous example:

```
sd.hat = sd(Hospital_Sample$LOS)
boot.bias = mean(sd.boot) - sd.hat
boot.bias
## [1] -0.02628716
```

- I.e., our estimate of the bias is -0.02433637, indicating that S is underestimating the true population standard deviation by that amount.
- We can subtract off this bias to obtain a bias-corrected estimator:

```
sd.corrected = sd.hat - boot.bias
sd.corrected
## [1] 5.171688
```

Bootstrap Bias Correction

- In the previous slide we obtained a "bias-corrected" estimate by subtracting the approximated bias off of the estimator. I.e., if we let $\hat{\theta}'$ represent the bias-corrected estimate, then we are using the formula

$$\hat{\theta}' = \hat{\theta} - \text{bias}_{\hat{\theta}}\left(\hat{\theta}^*\right)$$

- The idea behind this is as follows. First:

$$\begin{aligned}\mathbb{E}\left[\hat{\theta} - \text{bias}_\theta(\hat{\theta})\right] &= \mathbb{E}\left[\hat{\theta} - (\mathbb{E}[\hat{\theta}] - \theta)\right] \\ &= \mathbb{E}\left[\hat{\theta} - \mathbb{E}[\hat{\theta}] + \theta\right] \\ &= \mathbb{E}[\hat{\theta}] - \mathbb{E}[\hat{\theta}] + \theta \\ &= \theta,\end{aligned}$$

i.e., $\hat{\theta} - \text{bias}_\theta(\hat{\theta})$ is an unbiased estimator of θ. Thus, we use $\hat{\theta} - \text{bias}_{\hat{\theta}}(\hat{\theta}^*)$ as a bootstrap approximation to the unbiased estimate.

Bootstrap Variance

- Using the bootstrap to determine variance is straightforward: we just calculate the sample variance of $\left(\hat{\theta}^*_1, \hat{\theta}^*_2, \ldots \hat{\theta}^*_B\right)$.

```
sd.var = var(sd.boot)
sd.var
## [1] 0.1339911
```

- I.e., our estimate of the variance of S is $\mathbb{V}(S) = 0.1329716$.
- We can use this estimate to compare efficiency of estimators, etc.

Bootstrap Confidence Intervals: Percentile Method

- The bootstrap can also be used to estimate confidence intervals.
- There are a few approaches to this, but we will focus on the percentile method.
- The percentile method is not complicated: to estimate a $100(1-\alpha)\%$ confidence interval for θ, we examine the upper and lower $\alpha/2$ quantiles of $\hat{\theta}^*$.
- Below we give an estimate of the 95% confidence interval for the population standard deviation of length of stay, from the earlier example

```
sd.CI = quantile(sd.boot, c(0.025, 0.975))
sd.CI
##     2.5%    97.5%
## 4.407803 5.836259
```

Bootstrap Properties

- There are some properties of the bootstrap that are worth noting.
- First, *the bootstrap is an asymptotic method*. This means that the bootstrap is generally accurate, *so long as the original sample size n is large enough*. However, what exactly constitutes "large enough" is dependent on the underlying population and the sample statistic being investigated, so it is not possible to give a minimum sample size that will work in general.
- Second, *the sample size needed to accurately estimate the bias is smaller than other applications*. This is because when using the bootstrap to estimate the bias, we only need the distribution of $\hat{\theta}^* - \hat{\theta}$ to be similar to $\hat{\theta} - \theta$, which is a weaker condition than requiring $\hat{\theta}^*$ to be close to $\hat{\theta}$.
- Third, *the bootstrap fails on extreme order statistics*. That is, when trying to use the bootstrap to investigate the sampling distribution of things like the sample minimum, the sample maximum, or statistics that involve the extreme ends of the sample, the bootstrap will fail on average.

Simulating $\hat{\theta}$

- Here we will use simulation to determine the true distribution of $\hat{\theta}$.
- The Hospital_200 dataset is a sample drawn from the Hospital_Charges population, so we will draw from that population do perform the following simulations.

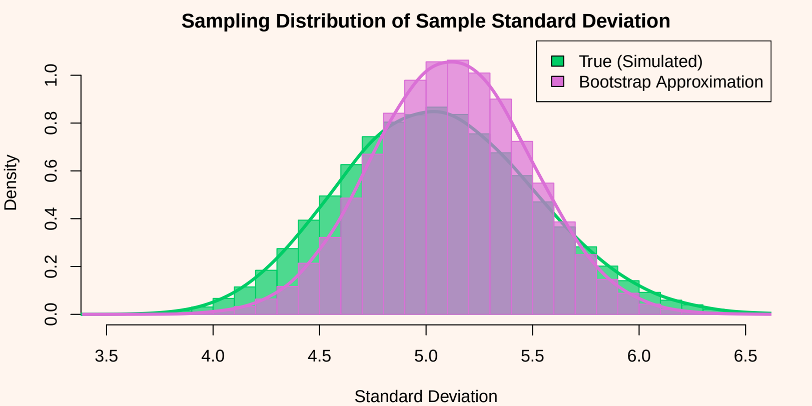
```
Charges = read.csv("~/R_Datasets/Hospital_Charges.csv")
sd.sim = c()
for(b in 1:100000)
{
  data.sim = Charges[sample(1:nrow(Charges), 200, replace = FALSE), ]
  sd.sim[b] = sd(data.sim$LOS)
}
sd.true = sd(Charges$LOS)
true.bias = mean(sd.sim) - sd.true
true.var = var(sd.sim)
true.bias
## [1] -0.01968397
true.var
## [1] 0.2084171
```

Bootstrap Distribution vs True Distribution

- Let's compare the bootstrap distribution $\hat{\theta}^*$ to the true distribution $\hat{\theta}$.

```r
my.col = c("springgreen3", "orchid")
hist(sd.sim, breaks = 30, freq = F, col = alpha(my.col[1], 0.7), border = my.col[1],
     main = "Sampling Distribution of Sample Standard Deviation",
     xlab = "Standard Deviation", ylim = c(0, 1.1), xlim = c(3.5, 6.5))
lines(density(sd.sim, adjust = 2), lwd = 3, col = my.col[1])
hist(sd.boot, breaks = 30, freq = F, border = my.col[2],
     col = alpha(my.col[2], 0.7), add = T)
lines(density(sd.boot, adjust = 2), lwd = 3, col = my.col[2])
legend("topright", legend = c("True (Simulated)", "Bootstrap Approximation"),
       fill = my.col)
```

Bootstrap Distribution vs True Distribution (Continued)



- As you can see, it's not perfect, but it's pretty good considering we made no assumptions on the population.

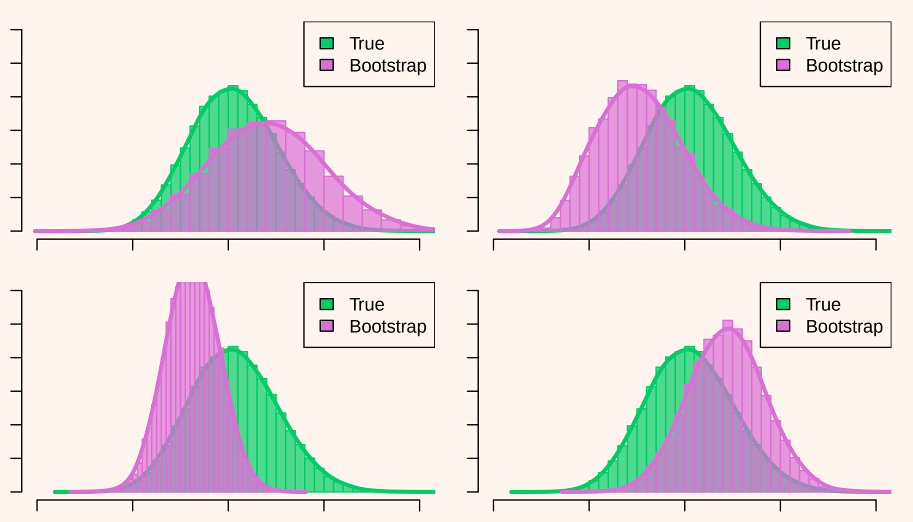
- Note that if we started from a different sample, we would end up with a different estimate of the sampling distribution of S.

Bootstrap Distribution Depends on Sample

- Below we will take three new samples of size $n = 200$ from the population and compare the resulting bootstrap distribution of each sample.

```
par(mfrow = c(2, 2), mar = c(1, 1, 1, 1))
for(i in 1:4)
{
  set.seed(2*i)
  Charges.sample = Charges[sample(1:nrow(Charges), 200, replace = FALSE), ]
  sd.boot2 = c()
  for(b in 1:10000)
    sd.boot2[b] = sd(Charges.sample[sample(1:200, 200, replace = TRUE), ]$LOS)
  hist(sd.sim, breaks = 30, freq = F, col = alpha(my.col[1], 0.7), border = my.col[1
      main = "", axes = FALSE, ylab = "", ylim = c(0, 1.2), xlim = c(3, 7))
  axis(1, labels = FALSE)
  axis(2, labels = FALSE)
  lines(density(sd.sim, adjust = 2), lwd = 3, col = my.col[1])
  hist(sd.boot2, breaks = 30, freq = F, border = my.col[2],
       col = alpha(my.col[2], 0.7), add = T)
  lines(density(sd.boot2, adjust = 2), lwd = 3, col = my.col[2])
  legend("topright", legend = c("True", "Bootstrap"), fill = my.col)
}
```

Bootstrap Distribution Depends on Sample (Continued)



Bootstrap Simulations: Confidence Intervals

- Below we will try taking many samples and see how often our confidence interval holds up (note: it should hold up for 95% of all samples).

```
CI.success = c()
for(i in 1:1000)
{
  Charges.sample = Charges[sample(1:nrow(Charges), 200, replace = F), ]
  sd.boot2 = c()
  for(b in 1:10000)
    sd.boot2[b] = sd(Charges.sample[sample(1:200, 200, replace = TRUE), ]$LOS)
  CI.new = quantile(sd.boot2, c(0.025, 0.975))
  CI.success[i] = sd.true < CI.new[2] && sd.true > CI.new[1]
}
mean(CI.success)
## [1] 0.923
```

- We can see that the true proportion of times that our CI contains σ is under 95%, but close.

Example

Example 1

Repeat the above analysis, but investigate the sampling distribution of the sample correlation between Charges and LOS. In particular, using the `Hospital_200` dataset,

1. Estimate the bias of the sample correlation
2. Estimate the variance of the sample correlation.
3. Construct a 95% confidence interval for the population correlation.
4. Use simulation to compare the true distribution of the sample correlation to your bootstrap estimate.

Section 2

Permutation Tests

Two-Sample Testing

- A frequent task in statistics is to tell if two populations X and Y are different.
  - For example, we may wish to tell if there is a difference between a *treatment* group and a *control* group in a study.
- As you learned in Stat 1150 / 2000, one approach is to use a two-sample t-test. Under the assumption that X and Y are independent and Normal, we can take two samples $\mathbf{x}$ and $\mathbf{y}$ and calculate the test statistic:

$$t = \begin{cases} \frac{\bar{x}-\bar{y}}{\sqrt{s_p^2/n_x+s_p^2/n_y}} & \text{if } X \text{ and } Y \text{have equal variances} \\ \frac{\bar{x}-\bar{y}}{\sqrt{s_x^2/n_1+s_y^2/n_y}} & \text{if } X \text{ and } Y \text{have unequal variances} \end{cases}$$

where s_p is the *pooled standard deviation*, given by

$$s_p^2 = \frac{(n_x-1)s_x^2+(n_y-1)s_y^2}{n_x+n_y-2}.$$

- This test statistic follows a $t(\nu)$ distribution, which then may be used to calculate a P-value for the test.

Permutation Test: Idea

- The two-sample t-test is a *parametric* approach to testing, and thus it requires *assumptions* to be made on the form of the data. In particular, we must assume that the sample means $\bar{X}$ and $\bar{Y}$ are Normally distributed. However, in practice, it is often the case that $\bar{X}$ and $\bar{Y}$ fail Normality.
- One non-parametric approach to this type of testing is the permutation test.
- The fundamental principle behind the permutation test is the idea that, if the populations X and Y were to be the same, then shuffling the data labels will have no effect.
- We consider all (or many) possible permutations of the two samples, and calculate the value of our test statistic under each possible permutation. Thus, we obtain an approximation of the sampling distribution of our test statistic.

Permutation Test: Procedure

- The procedure of the permutation test is below:

1. Form the null hypothesis, which is the assumption that the two distributions are identical. I.e., your null hypothesis is $H_0 : F_X = F_Y$.
2. Choose an appropriate test statistic for measuring the difference between F_X and F_Y. Popular options include...
   - $T = t \quad$ (the t-test statistic)
   - $T = \bar{X} - \bar{Y}$
   - $T = \text{median}(X) - \text{median}(Y)$
3. Collect your data (samples must be independent), and calculate T on this data. Call this value T_{cal}.
4. Consider all (or many) permutations of the data, and re-compute T on each permutation. This is an approximation to the sampling distribution of T under H_0, and is called the permutation distribution.
5. Using your test statistic, and your approximated sampling distribution, calculate the P-value for the test.

Permutation Test: Procedure (Continued)

- Thus, the permutation test procedure is quite similar to the bootstrap procedure. The primary difference is that, in the bootstrap procedure, we are estimating a sampling distribution by *resampling with replacement*, and in a permutation test we are shifting the data between groups, which is equivalent to *resampling without replacement*.
- Note that the number of possible permutations is given by $\binom{n_x+n_y}{n_x} = \binom{n_x+n_y}{n_y}$. Even for small sample sizes, it quickly becomes computationally infeasible to consider all possible permutations.
- For example, suppose that our sample sizes are $n_x = 14$ and $n_y = 20$. Then the number of possible permutations of this data is

```r
choose(14 + 20, 20)
## [1] 1391975640
```

-For this reason, we will consider many, but not all of the permutations. Like in our simulation and bootstrap procedures, 10 000 is a good number of repetitions.

Permutation Test: Procedure (Continued)

- Since the permutation test results involves the creation of the (approximated) sampling distribution of our test statistic, we can calculate the P-value as below:
  - If you are performing a right-tailed test (e.g., $\mu_x > \mu_y$), then the P-value in the permutation test is

$$P = \mathbb{P}(T > T_{cal}) \approx \frac{N(T > T_{cal})}{N(\text{Permutations})}$$

  - If you are performing a left-tailed test (e.g., $\mu_x < \mu_y$), then the P-value in the permutation test is

$$P = \mathbb{P}(T < T_{cal}) \approx \frac{N(T < T_{cal})}{N(\text{Permutations})}$$

  - If you are performing a two-tailed test (e.g., $\mu_x \neq \mu_y$), then the P-value in the permutation test is

$$P = \mathbb{P}(|T| > |T_{cal}|) \approx \frac{N(|T| > |T_{cal}|)}{N(\text{Permutations})}$$

Permutation Test: Example

- Below we will perform a permutation test to see if the length of stay for women is longer than the length of stay for men, from the Hospital200 dataset. We will use the the (unequal variances) *t*-statistic as our test statistic, which is a good default test statistic to use.

```r
x.og = Hospital_Sample[Hospital_Sample$SEX == "M", ]$LOS
y.og = Hospital_Sample[Hospital_Sample$SEX == "F", ]$LOS
tcal = (mean(x.og) - mean(y.og))/sqrt(var(x.og)/127 + var(y.og)/73)
data.combined = c(x.og, y.og)
tstat = c()
for(i in 1:10000)
{
  data.shuffled = sample(data.combined, 200, replace = FALSE)
  x.pm = data.shuffled[1:127]
  y.pm = data.shuffled[128:200]
  tstat[i] = (mean(x.pm) - mean(y.pm))/sqrt(var(x.pm)/127 + var(y.pm)/73)
}
mean(tstat < tcal)
## [1] 0.0137
```

Permutation Test: Example (Continued)

- In the last slide, we found an (approximate) P-value of $P = 0.0137$. This means that we reject $H_0 : \mu_M = \mu_F$ at the $\alpha = 0.05$ level of significance in favour of $H_a : \mu_M < \mu_F$. I.e., we have significant evidence that the length of stay for women is greater than the length of stay for men.
- Note that our P-value found is an approximation. The quality of this approximation depends on two things:
    1. How good of a representation $\mathbf{x}$ and $\mathbf{y}$ are of their respective populations X and Y. In short, smaller sample sizes will lead to less powerful tests.
    2. How many permutations are considered. With 10 000, permutations considered, this should have very little effect on P.
- Note also that we chose to use the unequal variances t-statistic as our test statistic. This is a good general choice, due to the important below property:

If sample sizes are unequal, then the permutation distribution will pick up any differences in the variance and will inflate Type I error rate

Permutation Test: Choosing a Test Statistic

- Since the unequal variances t-test statistic automatically handles unequal variances, we can avoid the concerns given at the bottom of the last slide.
- If our sample sizes are equal, or if we can assume equal variances, then the following hypotheses / test statistics may be preferred.
- For testing $H_0 : \mu_X = \mu_Y$, we may prefer to use...
  - the equal-variances t-test statistic for increased power.
  - $\bar{X} - \bar{Y}$.
  - $\bar{X}' - \bar{Y}'$, where $\bar{X}'$ and $\bar{Y}'$ are trimmed means. This will make our test less sensitive to outliers.
- We can use median(X) − median(Y) to test $H_0 : m_X = m_Y$ where m is the population median.
- We can even use s_x/s_y to test for differences in the population standard deviation. There's really no limitations to what kinds of two-sample testing we can do with the permutation test technique!

Examples

Example 2

Investigate the Length of Stay variable from the `Hospital200` dataset.

1. Use a permutation test to determine if the standard deviation women is greater than the standard deviation for men.
2. Based on your results of the last test, is it appropriate to update our earlier test for testing the difference of means? If so, consider possible ways of updating your test.

Example 3

Load in the `Height_and_Handedness` dataset. This dataset contains heights of several twelfth grade students across the US, as well as whether they are left- or right-handed. Use a permutation test to determine if left-handed and right-handed individuals have a different mean height.