

Bootstrap confidence intervals

Class 24, 18.05

Jeremy Orloff and Jonathan Bloom

1 Learning Goals

1. Be able to construct and sample from the empirical distribution of data.
2. Be able to explain the bootstrap principle.
3. Be able to design and run an empirical percentile or basic bootstrap to compute confidence intervals.
4. Be able to design and run a parametric bootstrap to compute confidence intervals.

2 Introduction

The [empirical bootstrap](#) is a statistical technique popularized by Bradley Efron in 1979. Though remarkably simple to implement, the bootstrap would not be feasible without modern computing power. The key idea is to perform computations on the data itself to estimate the variation of statistics that are themselves computed from the same data. That is, the data is ‘pulling itself up by its own bootstrap.’ (A google search of ‘by ones own bootstraps’ will give you the etymology of this metaphor.) Such techniques existed before 1979, but Efron widened their applicability and demonstrated how to implement the bootstrap effectively using computers. He also coined the term ‘bootstrap’ ¹.

The empirical bootstrap is also known as the [nonparametric bootstrap](#).

Our main application of the bootstrap will be to estimate the variation of point estimates; that is, to estimate confidence intervals. An example will make our goal clear.

Example 1. Suppose we have data

$$x_1, x_2, \dots, x_n$$

If we knew the data was drawn from $N(\mu, \sigma^2)$ with the unknown mean μ and known variance σ^2 then we have seen that

$$\left[\bar{x} - 1.96 \frac{\sigma}{\sqrt{n}}, \bar{x} + 1.96 \frac{\sigma}{\sqrt{n}} \right]$$

is a 95% confidence interval for μ .

Now suppose the data is drawn from some completely unknown distribution. To have a name we’ll call this distribution F and its (unknown) mean μ . We can still use the sample mean \bar{x} as a [point estimate](#) of μ . But how can we find a confidence interval for μ around \bar{x} ? Our answer will be to use the bootstrap!

¹Paraphrased from Dekking et al. *A Modern Introduction to Probability and Statistics*, Springer, 2005, page 275.

In fact, we'll see that the bootstrap handles other statistics as easily as it handles the mean. For example: the median, other percentiles or the trimmed mean. These are statistics where, even for normal distributions, it can be difficult to compute a confidence interval from theory alone.

3 Sampling

In statistics to [sample](#) from a set is to choose elements from that set. In a random sample the elements are chosen randomly. There are two common methods for random sampling.

Sampling without replacement

Suppose we draw 10 cards at random from a deck of 52 cards without putting any of the cards back into the deck between draws. This is called [sampling without replacement](#) or [simple random sampling](#). With this method of sampling our 10 card sample will have no duplicate cards.

Sampling with replacement

Now suppose we draw 10 cards at random from the deck, but after each draw we put the card back in the deck and shuffle the cards. This is called [sampling with replacement](#). With this method, the 10 card sample might have duplicates. It's even possible that we would draw the 6 of hearts all 10 times.

Think: What's the probability of drawing the 6 of hearts 10 times in a row?

Example 2. We can view rolling an 8-sided die repeatedly as sampling with replacement from the set $\{1,2,3,4,5,6,7,8\}$. Since each number is equally likely, we say we are sampling uniformly from the data.

Note. In practice if we take a small sample from a very large set then it doesn't matter whether we sample with or without replacement. For example, if we randomly sample 400 out of 300 million people in the U.S., then it is so unlikely that the same person will be picked twice that there is no real difference between sampling with or without replacement.

4 The empirical distribution of data

The [empirical distribution of data](#) is simply the distribution that you see in the data. Let's illustrate this with an example.

Example 3. Suppose we roll an 8-sided die 10 times and get the following data, written in increasing order:

1, 1, 2, 3, 3, 3, 3, 4, 7, 7.

Imagine writing these values on 10 slips of paper, putting them in a hat and drawing one at random. Then, for example, the probability of drawing a 3 is $4/10$ and the probability of drawing a 4 is $1/10$. The full empirical distribution can be put in a probability table

value x	1	2	3	4	7
$p(x)$	$2/10$	$1/10$	$4/10$	$1/10$	$2/10$

Notation. If we label the true distribution the data is drawn from as F , then we'll label

the empirical distribution of the data as F^* . If we have enough data then the law of large numbers tells us that F^* should be a good approximation of F .

Example 4. In the dice example just above, the true and empirical distributions are:

value x	1	2	3	4	5	6	7	8
true $p(x)$	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
empirical $p(x)$	2/10	1/10	4/10	1/10	0	0	2/10	0

The true distribution F and the empirical distribution F^* of the 8-sided die.

Because F^* is derived strictly from data we call it the **empirical distribution** of the data. We will also call it the **resampling distribution**. Notice that we always know F^* explicitly. In particular the expected value of F^* is just the sample mean \bar{x} .

5 Resampling

The empirical (or nonparametric) bootstrap proceeds by resampling from the data. We continue the dice example above.

Example 5. Suppose we have 10 data points, given in increasing order:

$$1, 1, 2, 3, 3, 3, 3, 4, 7, 7$$

We view this as a **sample** taken from some underlying distribution. To **resample** is to sample with replacement from the empirical distribution, e.g. put these 10 numbers in a hat and draw one at random. Then put the number back in the hat and draw again. You draw as many numbers as the desired size of the resample.

To get us a little closer to implementing this on a computer we rephrase this in the following way. Label the 10 data points x_1, x_2, \dots, x_{10} . To resample is to draw a number j from the uniform distribution on $\{1, 2, \dots, 10\}$ and take x_j as our resampled value. In this case we could do so by rolling a 10-sided die. For example, if we roll a 6 then our resampled value is $x_6 = 3$, the 6th element in our list.

If we want a resampled data set of size 5, then we roll the 10-sided die 5 times and choose the corresponding elements from the list of data. If the 5 rolls are

$$5, 3, 6, 6, 1$$

then the resample is

$$3, 2, 3, 3, 1.$$

Notes: 1. Because we are sampling with replacement, the same data point can appear multiple times when we resample.

2. Also because we are sampling with replacement, we can have a resample data set of any size we want, e.g. we could resample 1000 times.

Of course, in practice one uses a software package like R to do the resampling.

5.1 Star notation

If we have sample data of size n

$$x_1, x_2, \dots, x_n$$

then we denote a **resample of size m** by adding a star to the symbols

$$x_1^*, x_2^*, \dots, x_m^*$$

Similarly, just as \bar{x} is the mean of the original data, we write \bar{x}^* for the **mean of the resampled data**.

6 The empirical bootstrap

Suppose we have n data points

$$x_1, x_2, \dots, x_n$$

drawn from a distribution F . An **empirical bootstrap sample** (or nonparametric bootstrap sample) is a resample of the **same size n** :

$$x_1^*, x_2^*, \dots, x_n^*.$$

You should think of the latter as a sample of size n drawn from the empirical distribution F^* . For any statistic v computed from the original sample data, we can define a bootstrap statistic v^* . They both use the same formula but v^* is computed using the resampled data. With this notation we can state the bootstrap principle.

6.1 The bootstrap principle

The bootstrap setup is as follows:

1. x_1, x_2, \dots, x_n is a data sample drawn from a distribution F .
2. u is a statistic computed from the sample.
3. F^* is the empirical distribution of the data (the resampling distribution).
4. $x_1^*, x_2^*, \dots, x_n^*$ is a resample of the data **of the same size** as the original sample
5. u^* is the statistic computed from the resample.

Then the **bootstrap principle** says that

1. F^* is approximately equal to F .
2. The statistic u is approximated by u^* .
3. The variation of u is approximated by the variation of u^* .

Our real interest is in point 3: **we can approximate the variation of u by that of u^*** . It turns out that, in practice, the bootstrap gives a reasonable approximation of the variation. We will exploit this to estimate the size of confidence intervals.

6.2 Why the resample is the same size as the original sample

This is straightforward: the variation of the statistic u will depend on the size of the sample. If we want to approximate this variation we need to use resamples of the same size.

7 The empirical bootstrap confidence interval

Here we will show two methods of computing an empirical confidence interval: the [percentile bootstrap confidence interval](#) in part (c) below and the [basic bootstrap confidence interval](#) in part (d).

A search of the internet credits these names to Efron. Below, we will briefly discuss the merits of each. The basic confidence interval is also called the [reverse percentile confidence interval](#).

We will illustrate both methods with a ‘toy’ example using the same data.

Example 6. Toy example. We start with a made-up set of data that is small enough to show each step explicitly. The sample data is

30, 37, 36, 43, 42, 43, 43, 46, 41, 42

- (a) Use the data to give a point estimate of the mean of the underlying distribution.
- (b) Resample this to get 20 bootstrap samples.
- (c) Use the bootstrap samples to find the 80% bootstrap percentile confidence interval for the mean.
- (d) Use the bootstrap samples to find the 80% bootstrap basic confidence interval for the mean.

Note: R code for this example is shown in the section ‘R annotated scripts’ below. The code is also implemented in the R script, `class24-empiricalbootstrap.r` which is posted with our other R code.

Solution: (a) The sample mean is $\bar{x} = 40.3$, this is our point estimate.

(b) Our original sample contains 10 points. We used R to generate 20 bootstrap samples, each of size 10. Each of the 20 columns in the following array is one bootstrap sample. The values under the line are the means of the columns

36	36	42	42	41	42	42	43	42	36	43	42	42	43	43	43	36	43	36	46
43	37	36	43	43	41	36	41	46	30	43	46	42	30	43	43	41	41	37	43
43	43	43	42	46	42	42	43	43	43	36	43	42	30	36	43	42	41	41	37
42	43	37	37	43	36	43	43	43	41	42	42	37	43	36	42	46	43	43	42
36	46	36	41	43	30	43	42	46	46	43	37	46	42	46	43	41	43	41	36
41	42	43	43	46	30	36	41	36	46	36	30	42	43	42	37	42	41	37	43
42	46	30	46	30	43	42	41	46	42	37	46	43	43	37	43	30	43	46	37
43	42	43	37	42	43	46	43	37	42	42	37	36	43	46	30	43	46	46	41
43	43	41	46	46	43	30	46	36	41	42	42	36	42	37	36	46	43	42	43
30	46	43	42	43	42	41	42	37	43	43	43	43	43	41	36	43	42	43	46
39.9	42.4	39.4	41.9	42.3	39.2	40.1	42.5	41.2	41.0	40.7	40.8	40.9	40.2	40.7	39.6	41.0	42.6	41.2	41.4

(c) For the [percentile method](#), we first compute \bar{x}^* from each of our bootstrap samples. Here they are, sorted in increasing order.

39.2 39.4 39.6 39.9 40.1 40.2 40.7 40.7 40.8 40.9 41.0 41.0 41.2 41.2 41.4 41.9
42.3 42.4 42.5 42.6

The percentile method says to use the distribution of \bar{x}^* as an approximation to the distribution of \bar{x} . The 80% confidence interval stretches from the 10th to the 90th percentile.

Since we have 20 bootstrap means, these are given by the 2nd and 18th elements in our list. Therefore the bootstrap 80% percentile confidence interval is [39.4, 42.4].

To make the example readable, we only computed 20 bootstrap means. The beautiful key to the bootstrap is, that since \bar{x}^* is computed by resampling the original data, we can have a computer simulate \bar{x}^* as many times as we'd like. Hence, by the law of large numbers, we can estimate the distribution of \bar{x}^* with high precision.

Note: In our R code we use the quantile function to find the percentile values. This is easier than figuring out the index. It also is a little more sophisticated in finding the quantiles for a discrete set of values.

(d) The **basic method** is quite similar to the percentile method. The difference is that uses an algebraic 'pivot' analogous to the pivot used in finding z or t confidence intervals. As in Example 1, to make the confidence interval we want to know how much the distribution of \bar{x} varies around μ . That is, we'd like to know the distribution of

$$\delta = \bar{x} - \mu.$$

If we knew this distribution, then we could use the same algebra we saw when pivoting from non-rejection regions to confidence intervals. That is, we could find $\delta_{0.1}$ and $\delta_{0.9}$, the 0.1 and 0.9 critical values of δ . Then we would have

$$P(\delta_{0.9} \leq \bar{x} - \mu \leq \delta_{0.1} | \mu) = 0.8 \Leftrightarrow P(\bar{x} - \delta_{0.9} \geq \mu \geq \bar{x} - \delta_{0.1} | \mu) = 0.8$$

which gives an 80% confidence interval of

$$[\bar{x} - \delta_{0.1}, \bar{x} - \delta_{0.9}].$$

(As always with confidence intervals, we hasten to point out that the probabilities computed above are probabilities concerning the statistic \bar{x} given that the true mean is μ .)

The bootstrap principle offers a practical approach to estimating the distribution of $\delta = \bar{x} - \mu$. It says that we can approximate it by the distribution of

$$\delta^* = \bar{x}^* - \bar{x}$$

where \bar{x}^* is the mean of an empirical bootstrap sample.

Here is the beautiful key: since δ^* is computed by resampling the original data, we can have a computer simulate δ^* as many times as we'd like. Hence, by the law of large numbers, we can estimate the distribution of δ^* with high precision.

Next we compute $\delta^* = \bar{x}^* - \bar{x}$ for each bootstrap sample (i.e. each column) and sort them from smallest to biggest:

-1.1 -0.9 -0.7 -0.4 -0.2 -0.1 0.4 0.4 0.5 0.6 0.7 0.7 0.9 0.9 1.1 1.6 2.0 2.1 2.2 2.3

We will approximate the critical values $\delta_{0.1}$ and $\delta_{0.9}$ by $\delta_{0.1}^*$ and $\delta_{0.9}^*$. Since $\delta_{0.1}^*$ is at the 90th percentile we choose the 18th element in the list, i.e. 2.1. Likewise, since $\delta_{0.9}^*$ is at the 10th percentile we choose the 2nd element in the list, i.e. -0.9.

Therefore our bootstrap 80% basic confidence interval for μ is

$$[\bar{x} - \delta_{0.1}^*, \bar{x} - \delta_{0.9}^*] = [40.3 - 2.1, 40.3 + 0.9] = [38.2, 41.2]$$

In this example we only generated 20 bootstrap samples so they would fit on the page. Using R, we would generate 10000 or more bootstrap samples in order to obtain a very accurate estimate of $\delta_{0.1}^*$ and $\delta_{0.9}^*$.

8 Justification for the bootstrap principle

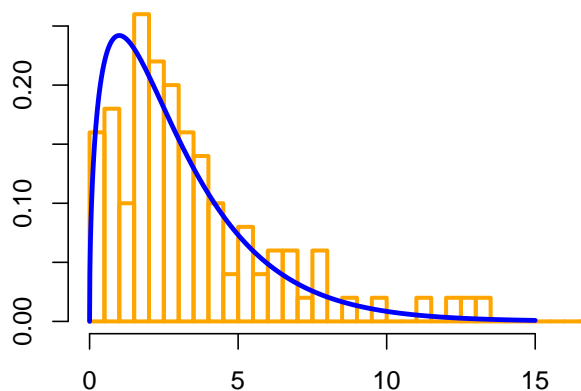
The bootstrap is remarkable because resampling gives us a decent estimate on how the point estimate might vary. We can only give you a ‘hand-waving’ explanation of this, but it’s worth a try. The bootstrap is based roughly on the law of large numbers, which says, in short, that with enough data the empirical distribution will be a good approximation of the true distribution. Visually it says that the histogram of the data should approximate the density of the true distribution.

First let’s note what resampling cannot do for us: it cannot improve our point estimate. For example, if we estimate the mean μ by \bar{x} then in the bootstrap we would compute \bar{x}^* for many resamples of the data. If we took the average of all the \bar{x}^* we would expect it to be very close to \bar{x} . This wouldn’t tell us anything new about the true value of μ .

Even with a fair amount of data the match between the true and empirical distributions is not perfect, so there will be error in our estimates for the mean (or any other value). But the amount of variation in the estimates is much less sensitive to differences between the true density and the data histogram: as long as they are reasonably close, the empirical and true distributions will exhibit the similar amounts of variation. So, in general the bootstrap principle is more robust when approximating the distribution of relative variation than when approximating absolute distributions.

What we have in mind is the scenario of our examples. The distribution (over different sets x of experimental data) of \bar{x} is ‘centered’ at μ and the distribution (over different bootstrap samples x^* of x) of \bar{x}^* is centered at \bar{x} . If there is a significant separation between \bar{x} and μ then these two distributions will also differ significantly. On the other hand the distribution of $\delta = \bar{x} - \mu$ describes the variation of \bar{x} about its center. Likewise the distribution of $\delta^* = \bar{x}^* - \bar{x}$ describes the variation of \bar{x}^* about its center. So even if the centers are quite different the two variations about the centers can be approximately equal.

The figure below illustrates how the empirical distribution approximates the true distribution. To make the figure we generate 100 random values from a chi-square distribution with 3 degrees of freedom. The figure shows the pdf of the true distribution as a blue line and a histogram of the empirical distribution in orange.



The true and empirical distributions are approximately equal.

9 Other statistics

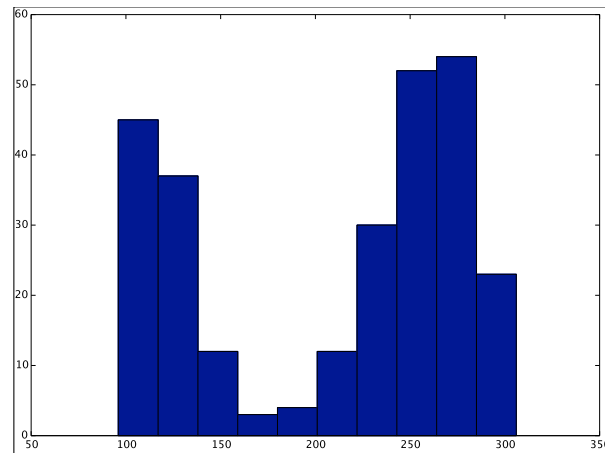
So far in this class we've avoided confidence intervals for the median and other statistics because their sample distributions are hard to describe theoretically. The bootstrap has no such problem. In fact, to handle the median all we have to do is change 'mean' to 'median' in the R code from Example 6.

Example 7. Old Faithful: confidence intervals for the median

Old Faithful is a geyser in Yellowstone National Park in Wyoming:

https://en.wikipedia.org/wiki/Old_Faithful

There is a publicly available data set which gives the durations of 272 consecutive eruptions. Here is a histogram of the data.



Question: Estimate the median length of an eruption and give a 90% percentile bootstrap confidence interval for the median.

Solution: The full answer to this question is in the R file `oldfaithful.r` and the Old Faithful data set. Both are posted on the class R code page. (Look under 'Other R code' for the old faithful script and data.)

Note: the code in `oldfaithful.r` assumes that the data `oldfaithful-data.txt` is in the current working directory.

Let's walk through a summary of the steps needed to answer the question.

1. Data: x_1, \dots, x_{272}
2. Data median: $x_{\text{median}} = 240$
3. Find the median x_{median}^* of a bootstrap sample x_1^*, \dots, x_{272}^* . Repeat 1000 times.

Put these 1000 values in order and pick out the 0.05 and 0.95 quantiles, i.e. the 50th and 950th biggest values. (In R we do this using the `quantile()` function) Call these $m_{0.05}^*$ and $m_{0.95}^*$.

5. The 90% percentile confidence interval for the median is $[m_{0.05}^*, m_{0.95}^*]$.

The bootstrap 90% CI we found for the Old Faithful data was $[230, 246]$. Since we used 1000 bootstrap samples a new simulation starting from the same sample data should produce a similar interval. If in Step 3 we increase the number of bootstrap samples to 10000, then

the intervals produced by simulation would tend to be even more similar to each other. One common strategy is to increase the number of bootstrap samples until the resulting simulations produce intervals that vary less than some acceptable level.

Example 8. Using the Old Faithful data, estimate $P(|\bar{x} - \mu| > 5 \mid \mu)$.

Solution: We proceed exactly as in the previous example in finding a basic bootstrap confidence interval, except we use the mean instead of the median.

1. Data: x_1, \dots, x_{272}
2. Data mean: $\bar{x} = 209.27$
3. Find the mean \bar{x}^* of 1000 empirical bootstrap samples: x_1^*, \dots, x_{272}^* .
4. Compute the bootstrap differences

$$\delta^* = \bar{x}^* - \bar{x}$$

5. The bootstrap principle says that we can use the distribution of δ^* as an approximation for the distribution $\delta = \bar{x} - \mu$. Thus,

$$P(|\bar{x} - \mu| > 5 \mid \mu) = P(|\delta| > 5 \mid \mu) \approx P(|\delta^*| > 5)$$

One bootstrap simulation for the Old Faithful data gave 0.230 for this probability.

10 Parametric bootstrap

Before getting started, we note that we only show the simplest algorithm for parametric bootstrap confidence intervals. In practice, more sophisticated algorithms are used. Since the bootstrap principle is the same in all cases, we feel it is worth encountering the simple algorithm in 18.05.

The examples in the previous sections all used the empirical bootstrap, which makes no assumptions at all about the underlying distribution and draws bootstrap samples by resampling the data. In this section we will look at the [parametric bootstrap](#). The main difference between the parametric and empirical bootstrap is the source of the bootstrap sample. For the parametric bootstrap, we generate the bootstrap sample from a parametrized distribution.

Another difference, is that, for the parametric bootstrap, we will use basic bootstrap confidence intervals. It's hard to find a definitive answer on whether the percentile or basic interval is better in this case. As we noted above, the real answer seems to be that, in practice, parametric bootstrap confidence intervals are computed with more sophisticated algorithms.

Here are the elements of using the parametric bootstrap to estimate a confidence interval for a parameter.

0. Data: x_1, \dots, x_n drawn from a distribution $F(\theta)$ with unknown parameter θ .
1. A statistic $\hat{\theta}$ that estimates θ .
2. Our bootstrap samples are drawn from $F(\hat{\theta})$.

3. For each bootstrap sample

$$x_1^*, \dots, x_n^*$$

we compute $\hat{\theta}^*$ and the bootstrap difference $\delta^* = \hat{\theta}^* - \hat{\theta}$.

4. The bootstrap principle says that the distribution of δ^* approximates the distribution of $\delta = \hat{\theta} - \theta$.

5. Use the bootstrap differences to make a bootstrap confidence interval for θ .

Example 9. Suppose the data x_1, \dots, x_{300} is drawn from an $\text{exp}(\lambda)$ distribution. Assume also that the data mean $\bar{x} = 2$. Estimate λ and give a 95% parametric bootstrap confidence interval for λ .

Solution: This is implemented in the R script `class24-parametricbootstrap.r` which is posted with our other R code.

It's will be easiest to explain the solution using commented code.

```
# Parametric bootstrap
# Given 300 data points with mean 2.
# Assume the data is exp(lambda)
# PROBLEM: Compute a 95% parametric bootstrap confidence interval for lambda
# We are given the number of data points and mean
n = 300
xbar = 2
# The MLE for lambda is 1/xbar
lambda_hat = 1.0/xbar
# Generate the bootstrap samples
# Each column is one bootstrap sample (of 300 resampled values)
n_boot = 1000
# Here's the key difference with the empirical bootstrap:
# We draw the bootstrap sample from Exponential(lambda_hat).
x = rexp(n*n_boot, lambda_hat)
bootstrap_sample = matrix(x, nrow=n, ncol=n_boot)
# Compute the bootstrap lambda_star
lambda_star = 1.0/colMeans(bootstrap_sample)
# Compute the differences
delta_star = lambda_star - lambda_hat
# Find the 0.05 and 0.95 quantile for delta_star
d = quantile(delta_star, c(0.05,0.95))
# Calculate the 95% confidence interval for lambda.
ci = lambda_hat - c(d[2], d[1])
# This line of code is just one way to format the output text.
# sprintf is an old C function for doing this. R has many other
# ways to do the same thing.
s = sprintf("Confidence interval for lambda: [%.3f, %.3f]", ci[1], ci[2])
```

cat(s)

11 Building a better bootstrap

The first thing to say, is that we have just scratched the surface of bootstrap techniques. There are more sophisticated methods that correct for bias in the original sample or for skewness in the underlying distribution. There are also methods for when the original sample size is small.

For the nonparametric bootstrap, there are different opinions about which of the basic and percentile methods gives the most accurate results. What is clear is that, if the empirical distribution is symmetric, then the basic and percentile confidence intervals are equivalent. If the distribution is skewed then the two intervals are skewed in opposite directions.

Hesterberg in (<https://www.tandfonline.com/doi/full/10.1080/00031305.2015.1089789>) is fairly convincing that the percentile method performs better on skewed distributions.

On the other hand, Rice says of the percentile method, “Although this direct equation of quantiles of the bootstrap sampling distribution with confidence limits may seem initially appealing, it’s rationale is somewhat obscure.”²

We’ll give Hesterberg the final word. He contributed several excellent responses to a discussion group. Unfortunately the link is no longer active, but here is one of his posts.

Skewness is a fact of life. So what do you do when there is skewness? A bootstrap percentile interval is usually a good choice – much better than a symmetric interval like t , that pretends there is no skewness. It is fine for a beginning stats student, and for most applications in practice. Someone more advanced (or using easy-to-use software) can use a BCa or bootstrap- t interval, for better accuracy.

Bootstrapping the median is a different issue – for small samples the median is quite sensitive to whether the population being sampled from is continuous or discrete. So if the population is continuous, but you use the nonparametric bootstrap that draws from a discrete distribution (the data), the bootstrap distribution won’t look like the true sampling distribution. Even so, the bootstrap percentile interval is not bad in this case, close to the exact confidence interval for the median (typically the same or one order statistic different).

12 R annotated transcripts

12.1 Using R to generate a empirical bootstrap confidence intervals

This code only generates 20 bootstrap samples. In real practice we would generate many more bootstrap samples. It is making a bootstrap confidence interval for the mean. This code is implemented in the R script `class24-empiricalbootstrap.r` which is posted with our other R code.

²John Rice, *Mathematical Statistics and Data Analysis*, 2nd edition, p. 272.

```
# Data for the example 6
x = c(30,37,36,43,42,43,43,46,41,42)
n = length(x)

# sample mean
xbar = mean(x)

n_boot = 20
# Generate 20 bootstrap samples, i.e. an n x 20 array of
# random resamples from x
tmp_data = sample(x, n*n_boot, replace=TRUE)
bootstrap_sample = matrix(tmp_data, nrow=n, ncol=n_boot)

# Compute the means  $\bar{x}^*$ 
xbar_star = colMeans(bootstrap_sample)

# Calculate the bounds for the 80% percentile confidence interval.
percentile_ci = quantile(xbar_star, c(0.1, 0.9))
cat('80% percentile confidence interval: ',percentile_ci, '\n')

# Compute  $\delta^*$  for each bootstrap sample
delta_star = xbar_star - xbar

# Find the 0.1 and 0.9 quantiles for delta_star
d = quantile(delta_star, c(0.1, 0.9))

# Calculate the bounds for the 80% basic confidence interval.
# Note how pivoting reverses the order of d[1] and d[2]
basic_ci = xbar - c(d[2], d[1])
cat('80% basic confidence interval: ',basic_ci, '\n')

# ALTERNATIVE: the quantile() function is sophisticated about
# choosing a quantile between two data points. A less sophisticated
# approach is to pick the quantiles by sorting xbar_star and delta_star and
# choosing the index that corresponds to the desired quantiles.
# This is what we did in the text above. We show the code using this for the
# percentile method below.

# Sort the results
sorted_xbar_star = sort(xbar_star)

# Find the 0.1 and 0.9 quantiles values of xbar_star
q1_alt = sorted_xbar_star[2]
q9_alt = sorted_xbar_star[18]

# Find and print the 80% percentile confidence interval for the mean
ci_alt = c(q1_alt, q9_alt)
cat('Alternative confidence interval: ',ci_alt, '\n')
```

MIT OpenCourseWare

<https://ocw.mit.edu>

18.05 Introduction to Probability and Statistics

Spring 2022

For information about citing these materials or our Terms of Use, visit: <https://ocw.mit.edu/terms>.