

DS 740

Data Mining

Bootstrapping for Error Estimation

Another way to reuse data

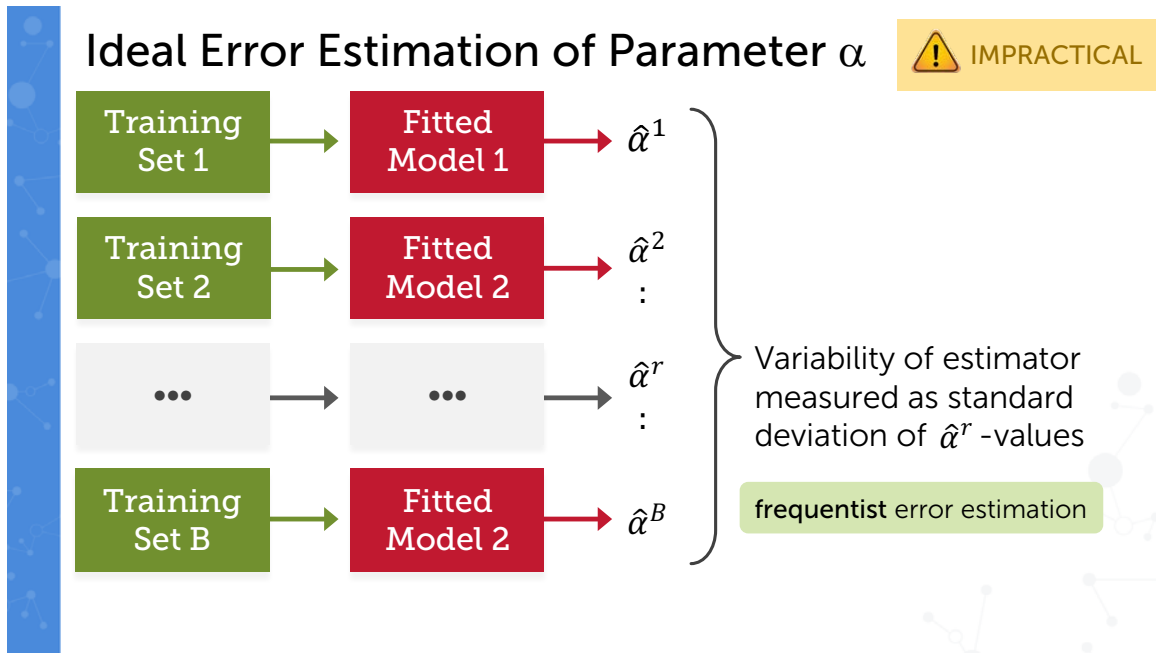
Important note: Transcripts are **not** substitutes for textbook assignments.

Learning Objectives

By the end of this lesson, you will be able to:

- Explain bootstrapping method and understand its application for error estimation.
- Define standard error, $SE_B(\hat{\alpha})$, for estimating error of a point estimator $\hat{\alpha}$.
- Apply bootstrapping method for error estimation.





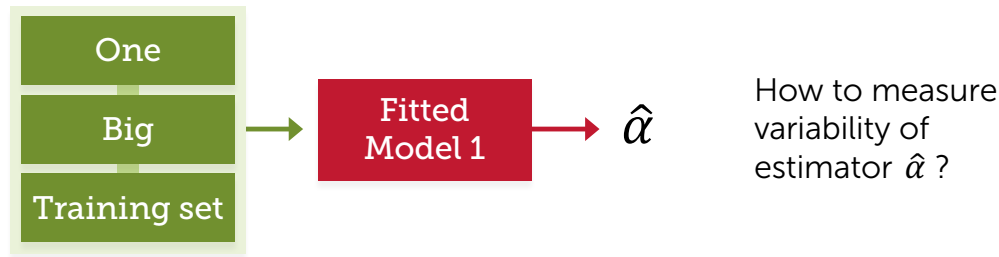
The theory behind an ideal error estimation would mean that we select training sets many, many times from the population and obtain a parameter estimate from each of these training sets, or samples.

Repeating this estimation many, many times would produce many replications of the estimator, α hat. The variability of these values is measured by the usual standard deviation.

This is the traditional approach known as the frequentist statistical approach, and is based in sampling distributions of statistics theory. However, this theoretical approach is not practical in application since we cannot, in nearly every situation, cannot select multiple training sets.

Practical Error Estimation of parameter α

One sample, with one fitted model and one estimate.



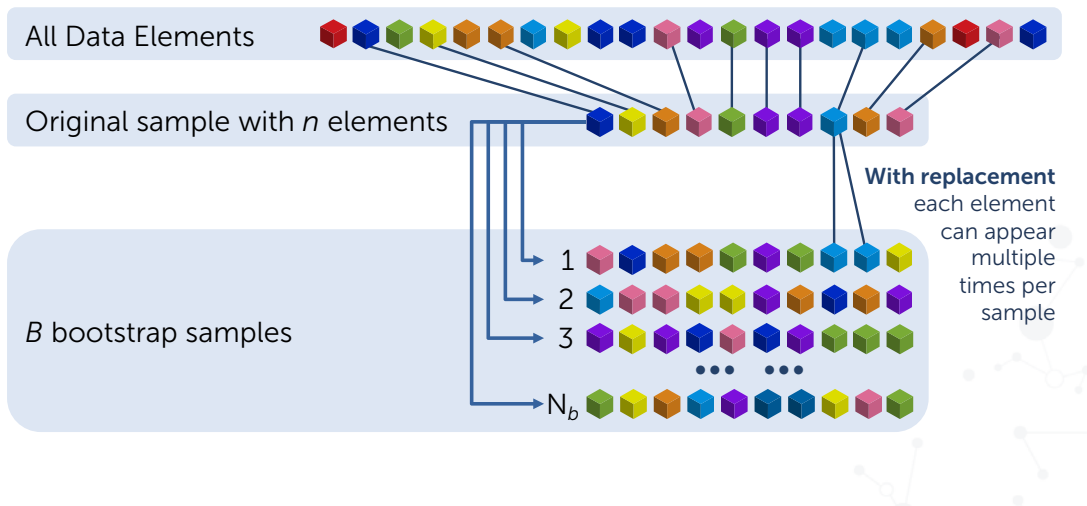
How to get "best estimate" (as above) and also estimate error?

From a practical perspective, adhering to previous principles, we would like to use any and all data we select from the population to get the best possible estimate of the parameter. So how can we use just one parameter estimate to measure variability of that estimate? Well, the simple answer is we can't.

Thus, somehow we have to devise a method of sampling that will still allow us to come up with an overall estimate as shown here, but also will produce multiple sets of data from which we can obtain multiple parameter estimates that will importantly exhibit approximately accurate error.

Re-Sampling to Produce Bootstrap Sets

Bootstrap method: Originally due to Efron (*see note below*).

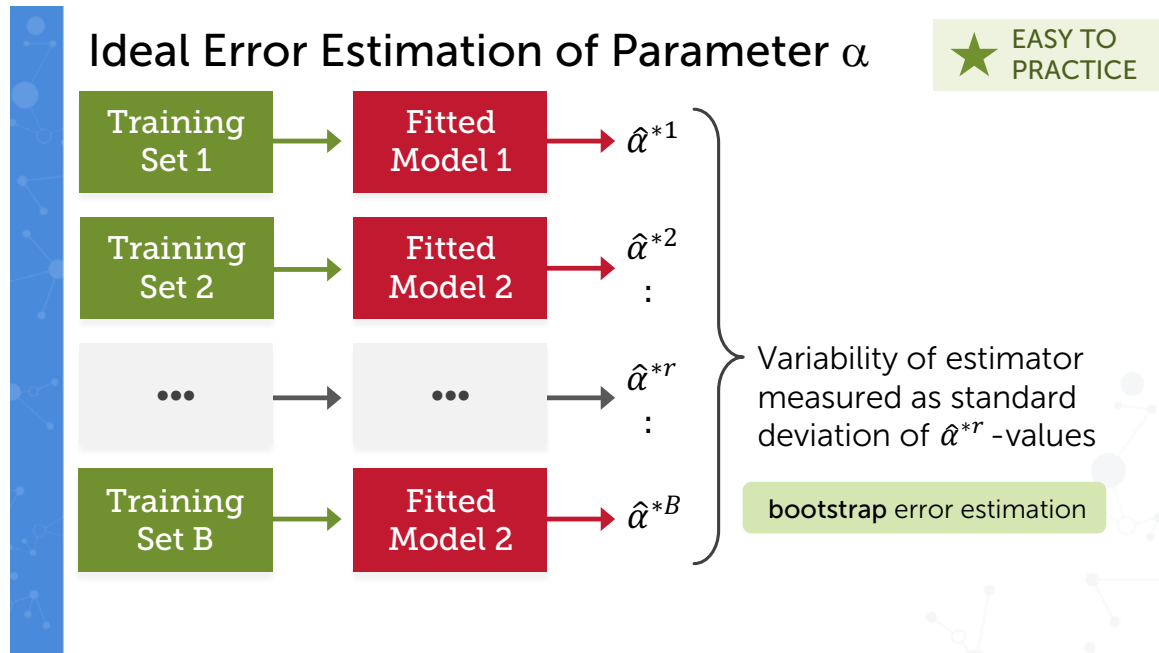


The bootstrap is an effective method of accurately estimating the error for a parameter estimation. This method was introduced by Bradley Efron in 1979, and has recently gained more prominence as a computationally practical way of estimating error, even within basic statistical methods.

The concept is simple. From the initial sample, which was selected from the much larger population, we re-sample a number of times-- let's call that b times-- with replacement, obtaining b bootstrap samples. Sampling with replacement means that each individual or object from the original sample from which we're selecting, from which we're re-sampling, can appear multiple times in a single bootstrap sample.

Notes:

For more about Bradley Efron's creation of the bootstrap method, see [The Annals of Statistics, Volume 7, Number 1](#) (1979, p. 1-26.)



Once we get these multiple bootstrap samples, bootstrap error estimation involves taking the bootstrap samples, fitting the model on each sample and obtaining the estimate of the parameter from each of the samples. We take these many estimated values and compute their standard deviation to approximate the error. This is easy to put in practice, particularly with powerful computation tools.

Error Estimation from Re-samples

- Variability of the $\hat{\alpha}^{*r}$ -values ($r = 1, 2, \dots, B$) from the re-samples is approximately accurate for the standard deviation of the estimates.
- Let the average estimator value be denoted: $Avg(\hat{\alpha}^*) = \frac{1}{B} \sum_{r=1}^B \hat{\alpha}^{*r}$

Then define the standard error of the estimator to be:

$$SE_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1} \sum_{r=1}^B (\hat{\alpha}^{*r} - Avg(\hat{\alpha}^*))^2}$$

Theory, as well as empirical studies, have verified that the bootstrap method produces accurate estimates of variability for a wide variety of models. The formula for computing the so-called standard error based on the b bootstrap samples is in fact just our familiar formula for standard deviation. It is applied to the bootstrap estimates, the alpha hat stars, for estimating the parameter alpha.

Question 1

DS740 - Bootstrapping for Error Estimation

🔊 Question for Self Assessment: Multiple Choice

Suppose we are estimation a parameter α . When we select bootstrap samples, our purpose is to compute:

- ☐ The estimated value of the parameter.
- ☐ The variability of the estimated values for the parameter.

SUBMIT

Answer is at the end of this transcript

Bootstrapping in R

Package: *boot* [Download x1](#)

Input function: fits model and outputs the parameters to be estimated.

Application: the function, *boot*, uses data set and input function, producing the bootstrap re-samples and the coefficient estimates from those samples.



The package *boot* must be downloaded into R again. As with all packages, it only needs to be done once in order to most easily do bootstrapping in R. While one can write their own program to do bootstrapping, *boot* is a very intuitive package with a function of the same name that requires simply a data set and an input function.

The input function must fit the model and output the parameters that are desired to be estimated. The function then does all the heavy lifting. So running *boot* will produce the bootstrap re-samples as well as the coefficient estimates from those samples. And you will see a demonstration of how to work with those coefficient tests.

Body Fat Example Revisited

Using `bodyfat.csv` data set, with author's description and adjustments (errors in density and height have been corrected).

Response: *BodyFatSiri* (using Siri's equation)

Predictors: *Abs, Weight, Wrist, Forearm, Neck, Biceps, Age, Thigh, Hip, Ankle, BMI, Height, Chest, Knee.*

Model 1 uses first predictor (Abs) only; Model 2 uses all 14 predictors via multiple linear regression.

Bootstrap used to compute standard error based on bootstrap sample estimates:

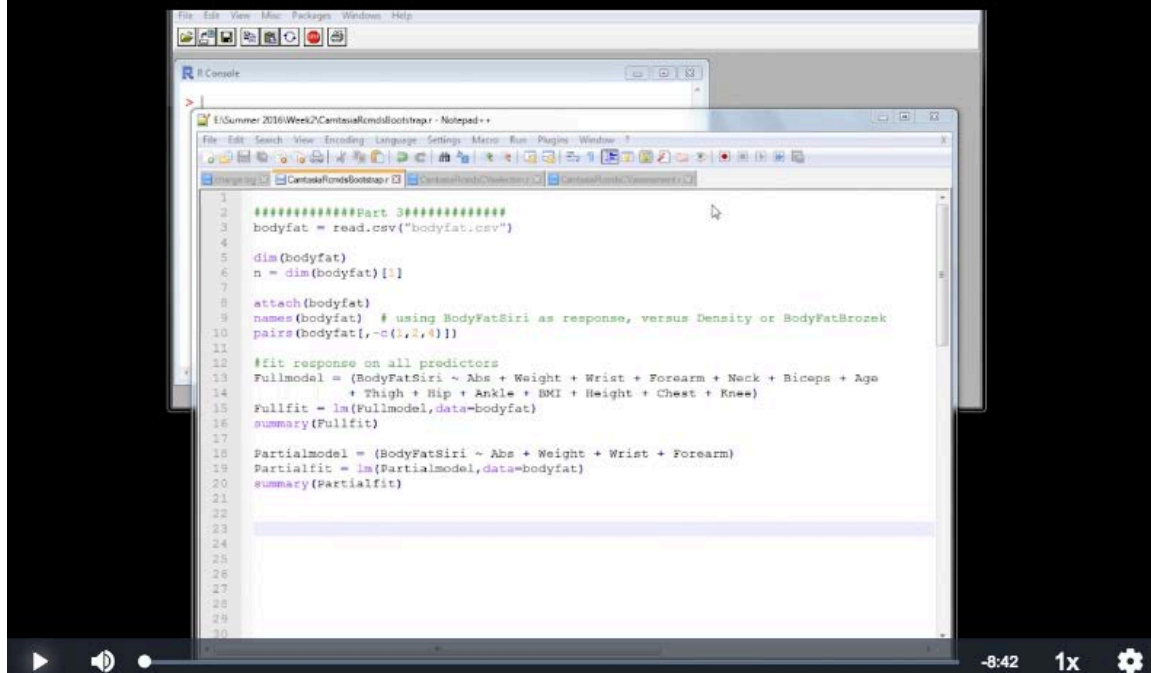
$$SE_B(\hat{\beta}) = \sqrt{\frac{1}{B-1} \sum_{r=1}^B (\hat{\beta}^{*r} - \text{Avg}(\hat{\beta}^*))^2}$$

Notes:

The `bodyfat.csv` file is available in the online course.

The data set author's description and adjustments are available here: [Fitting Percentage of Body Fat to Simple Body Measurements - Roger W. Johnson, Carleton College](#)

DS740 - Bootstrapping for Error Estimation



```
1 #####Part 3#####
2 bodyfat = read.csv("bodyfat.csv")
3
4 dim(bodyfat)
5 n = dim(bodyfat)[1]
6
7 attach(bodyfat)
8 names(bodyfat) # using BodyFatSiri as response, versus Density or BodyFatBrozek
9 pairs(bodyfat[,c(1,2,4)])
10
11 #fit response on all predictors
12 Fullmodel = (BodyFatSiri ~ Abs + Weight + Wrist + Forearm + Neck + Biceps + Age
13 + Thigh + Hip + Ankle + BMI + Height + Chest + Knee)
14 Fullfit = lm(Fullmodel,data=bodyfat)
15 summary(Fullfit)
16
17 Partialmodel = (BodyFatSiri ~ Abs + Weight + Wrist + Forearm)
18 Partialfit = lm(Partialmodel,data=bodyfat)
19 summary(Partialfit)
20
21
22
23
24
25
26
27
28
29
30
```

This slide represents a video/screencast in the lecture. The transcript does not substitute video content.

We're ready to look at our third reuse of data lecture example. Once again, this example is based in the body fat data. So if you have not already read in that data set, please do so now. You may also make sure that you've changed your directory. We're going to start by fitting the response on all predictors to get a full fit of a FOL model. So that's body fat response, calculated by series equation, fit on all 14 predictors. And we see a summary of that, coefficient estimates, and standard error, as computed mathematically. We'll also fit a partial model, as this will give us a little bit easier output to work with on just four predictors.

The four predictors, selected by stepwise regression, and four predictors chosen by cross validation. All right, so now that we have these two model fits, a full fit and a partial fit, we're going to take a look at fitting these models and estimating coefficients on simulated samples selected from the original sample, with replacement. And so that process is what we've introduced as bootstrapping. In order to do bootstrap, you need to install the package boot. Once you've done that, you can read in the package. And we need to be able to define functions that output the coefficients that we're aiming to estimate. So beta fn FOL stands for beta function FOL. And this is a function that takes input data with some sort of index, fits the fol model we previously defined,

and fits it to that part, the index part of the input data. We'll then return the LM fit boot coefficients.

So all that is the definition of a function. We'll do something very similar for the partial fit on the partial model and define that function data beta fn partial. We're now ready to run the boot function to simulate the actual re-samples, and then produce the coefficients for each re-sample. And we're going to go with the partial model bootstrap to begin because, as I said, it's a little bit simpler. So we'll all start from the same place. We'll set the random seed to be 2. And our partial boot output is running the boot function on the body fat data set, with the function as defined up above here. And we're going to do 1,000 bootstrap simulations and see what comes out.

Now this particular function can take a little bit of time to run, depending on how long each model fit takes. It turns out here that we're coming up with some summary of our outputs. T1 represents our first term. And in this case, that is simply the intercept. T2, the second term, that's the coefficient for the first predictor-- ABS, in this case. So we might want to take a little bit more of a look at some of this information. So the partial boot output T element gives us all the coefficient estimates. So if I were to type this in up here, I would end up with a matrix that should be 1,000 values for the five coefficient estimates. So let's just not-- rather than printing that out, let's just take a look at the dimension. So the rows are the 1000 estimates, and each of the columns is which coefficient it is. So partial boot output, element T, picking the first column, would be all 1,000 coefficient estimates for the intercept from the 1,000 bootstrapped samples. So I will go ahead and just copy and paste that over. And we see the 1,000 values listed here from the different partial model fits on the bootstrap sample.

Not a great summary. So let's actually summarize. Let's take-- calculate the standard deviation and make a histogram of those values. When we take a look at the histogram, it has a-- not surprisingly-- normal shape. Good assumptions were met for this model fit. So we expect the coefficient estimates to have a good fit. And the standard deviation-- well, the standard deviation is estimated to be 7.953. Now we might wish to compare this to the standard error as mathematically computed based on mathematical statistics theory. And so, that was output in our original summary of the overall fit on the original data-set, on the original sample.

So we're going to pull off just the standard errors, the first element. Copy and paste this server. And we see a slightly lower estimate. Again, this is not surprising. We get a higher estimate based on the bootstrap samples because that's a little bit more realistic. All right, similar sort of thing, when we take a look at the coefficient estimates for the ABS term-- we'll copy and paste all of

this over right away. We see all the coefficient estimates for the 1,000 bootstrap samples. And we obtain a standard deviation estimate-- estimated standard deviation. That is, again, higher than mathematical computation. You could repeat the same thing for the third term, which is weight. And we see a similar sort of result here. If you'd like, you can continue this, and take a look at the full model bootstrap. I'll just run all the commands in one fell swoop. And it takes a little bit. It's going to take a little bit longer here to fit the bootstraps. But we see a similar sort of summary.

Notice that the estimated standard deviation for the intercept is much larger than the mathematically computed one, as is the standard error, the estimated standard error, or the estimated standard deviation from bootstrap for the weight term.

Notes:

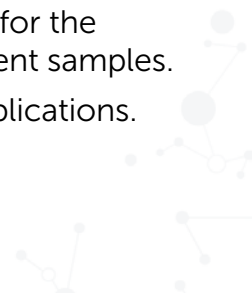
See the online course for a downloadable R file containing the set of commands used in this demonstration.

Summary

Bootstrapping is a method that can be applied to estimate variability of parameter estimates through re-using the original data.

- Re-samples are selected from the original sample, with replacement
- A fitted model is produced for each re-sample, resulting in new parameter estimate(s) for each re-sample.
- The boot-strap standard error, calculated as the simple standard deviation of these estimates, is approximately correct for the standard deviation of parameter estimates from different samples.

The *R* function *boot* (from package *boot*) is used for applications.



Question 1 Answer

DS740 - Bootstrapping for Error Estimation

Feedback for Self Assessment

✓ Correct!

Suppose we are estimation a parameter α . When we select bootstrap samples, our purpose is to compute:

Your answer:

The variability of the estimated values for the parameter.

Correct answer:

The variability of the estimated values for the parameter.