Bootstrapping Method Example

D2K Course Staff

April 01 2019

Introduction

When analyzing datasets, we want to maximize the probability that our discoveries or insights will hold true for future, yet-to-be-seen data. One of the ways that this can be done is through a procedure called **bootstrapping**. The basic idea is to replicate the process of gathering "new data" by simply resampling *with replacement* some facet of the data set that we already have. This saves the time and effort of collecting new data in order to validate our results, or losing out on statistical power from holding out a portion of our data. Below, we look at some examples of different bootstrapping methods and discuss when they should be applied.

Case Resampling

The most widely-used bootstrapping method is case resampling. With this method, we resample entire rows of observations with replacement and . Specifically, the procedure works as follows:

1. Fit our desired model to the original data set; get estimated parameters from this original model.
2. Run k bootstrap iterations. For each iteration:
   (a) Resample observations with replacement at size n.
   (b) Fit desired model on resampled data set.
   (c) Record new parameter estimates on resampled data set.
3. Use our collection of k parameter estimates in order to make some sort of statement about the stability of our estimates.

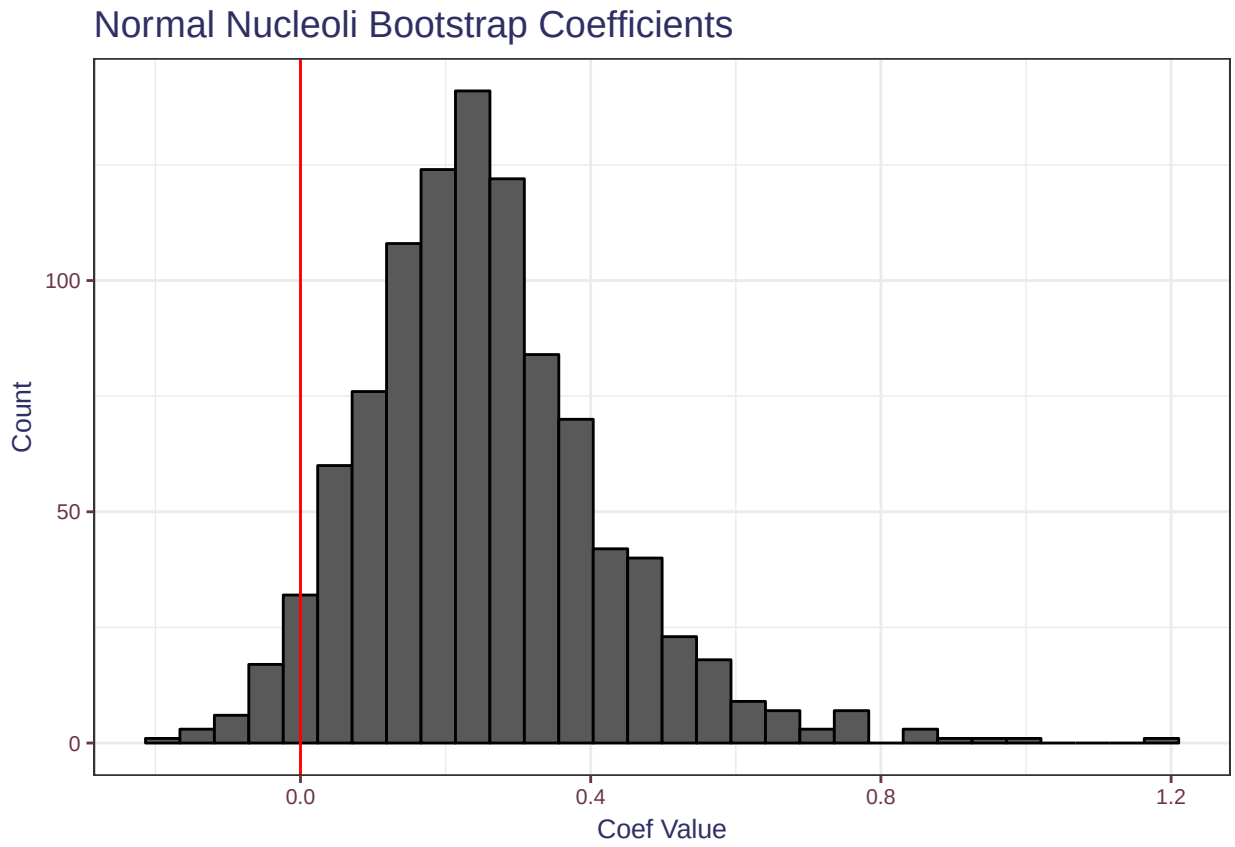
In this example, we fit a logistic regression model to the `biopsy` data set from the `MASS` package. We will try to predict the class of cancer based on the predictor variables in the data.

```
       ID V1 V2 V3 V4 V5 V6 V7 V8 V9     class
1 1000025  5  1  1  1  2  1  3  1  1    benign
2 1002945  5  4  4  5  7 10  3  2  1    benign
3 1015425  3  1  1  1  2  2  3  1  1    benign
4 1016277  6  8  8  1  3  4  3  7  1    benign
5 1017023  4  1  1  3  2  1  3  1  1    benign
6 1017122  8 10 10  8  7 10  9  7  1 malignant


Call:
glm(formula = class ~ ., family = "binomial", data = biopsy[,
    -1])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.4841  -0.1153  -0.0619   0.0222   2.4698

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -10.10394    1.17488  -8.600  < 2e-16 ***
V1            0.53501    0.14202   3.767 0.000165 ***
V2           -0.00628    0.20908  -0.030 0.976039
V3            0.32271    0.23060   1.399 0.161688
V4            0.33064    0.12345   2.678 0.007400 **
V5            0.09663    0.15659   0.617 0.537159
V6            0.38303    0.09384   4.082 4.47e-05 ***
V7            0.44719    0.17138   2.609 0.009073 **
V8            0.21303    0.11287   1.887 0.059115 .
V9            0.53484    0.32877   1.627 0.103788
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 884.35  on 682  degrees of freedom
Residual deviance: 102.89  on 673  degrees of freedom
  (16 observations deleted due to missingness)
```

```
AIC: 122.89

Number of Fisher Scoring iterations: 8
```

Normally, one might just use the p-values from the output of the model to do inference on whether each parameter is not equal to 0 in the model. This, however, relies on the assumption that the residuals are Gaussian distributed. Instead, we an use bootstrapping to obtain a nonparametric confidence interval by bootstrap resampling.

We can then look at the distribution of the variable coefficients in the bootstrap samples; we can also calculate the empirical p-value of the coefficient estimate.




```
[1] 0.039
```

In this case, 39 out of 10000 estimates were less than 0. This is equivalent to a p-value of 0.078 for a two-sided t-test. We can also look at the quantiles of the parameter estimate distribution to get a 95% confidence interval:

```
       2.5%       97.5%
-0.02516126  0.62345640
```

When Does This Fail?

Case resampling will cover your model validation needs in almost all cases. Use it unless you have a good reason not to. However, there do exist some situations for which case resampling will fail. The method relies on the assumption that the observations in your data set are independently and identically distributed. Additionally, we inherently assume that both the original and new data sets are representative samples of the

entire population. When these are violated, the results we get from case resampling may not be valid (or we may not even be able to get any results!)

Residual Resampling

In some cases, case resampling will not be representative of the original data set. This most likely will occur when the model contains categorical variables that have one or more small categories. Thus, when resampling, there is a nontrivial probability that not all of the categories of the variable will be represented in the new data set, meaning that the fitted model will be different from the original one. In this case, one might consider using residual bootstrapping. The process is as follows:

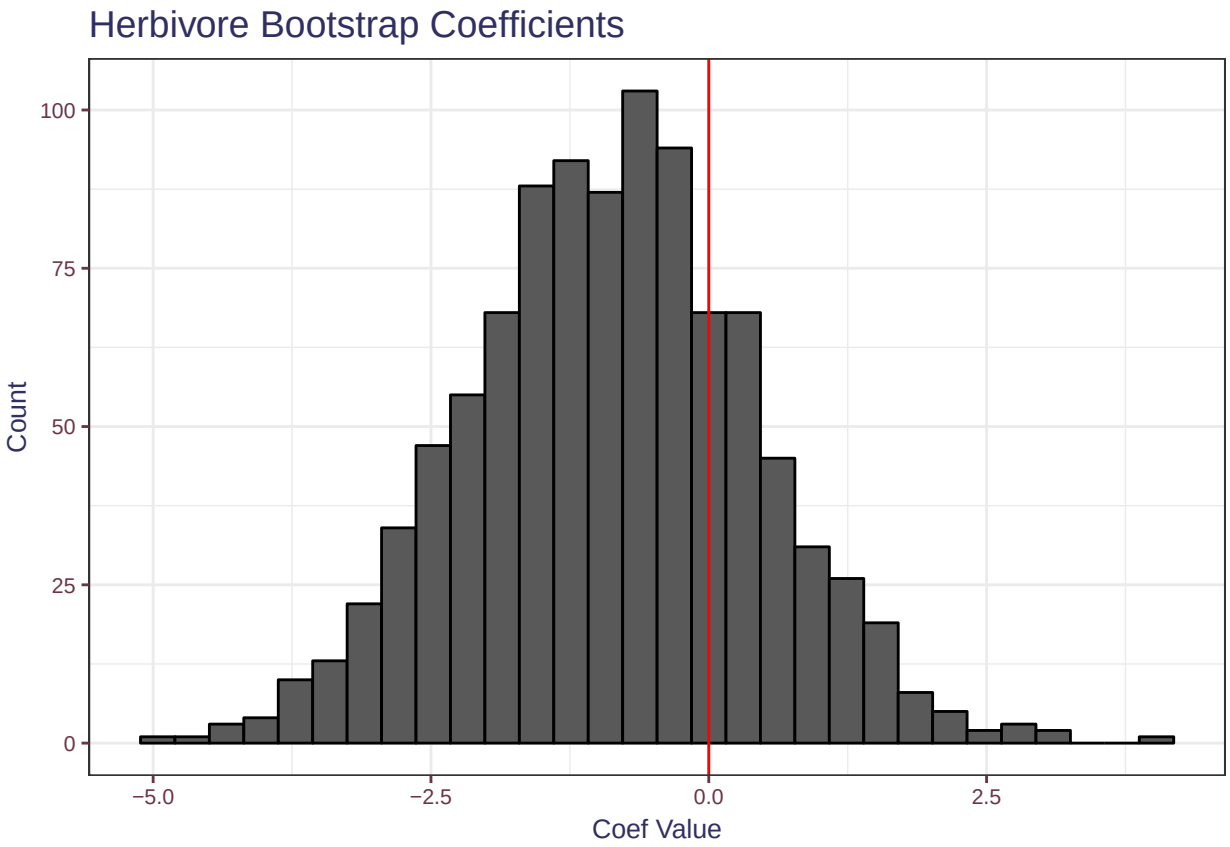
1. Fit our desired model to the original data set; get estimated parameters from this original model.
2. Calculated the fitted values $\hat{y}$ and residuals $\hat{e}$ from the model and the data.
3. Run k bootstrap iterations. For each iteration:
   (a) Resample residuals $\hat{e}$ with replacement at size n.
   (b) Add resampled residuals to fitted values $\hat{y}$.
   (c) Fit desired model on new data set.
   (d) Record new parameter estimates on resampled data set.
4. Use our collection of k parameter estimates in order to make some sort of statement about the stability of our estimates.

As an illustrative example, we will run residual bootstrapping on the `msleep` data set from `ggplot2`, predicting total hours slept based on diet using an ordinary linear regression model. If we look at a summary of the diet variable, we see that there are only 5 insectivores in the data set; thus, if we take many bootstrap samples of this data, we will likely end up with a data set with no insectivores. In this case, it might be more prudent to use residual resampling. (We will table a discussion on whether fitting this linear model is a good idea on principle.)

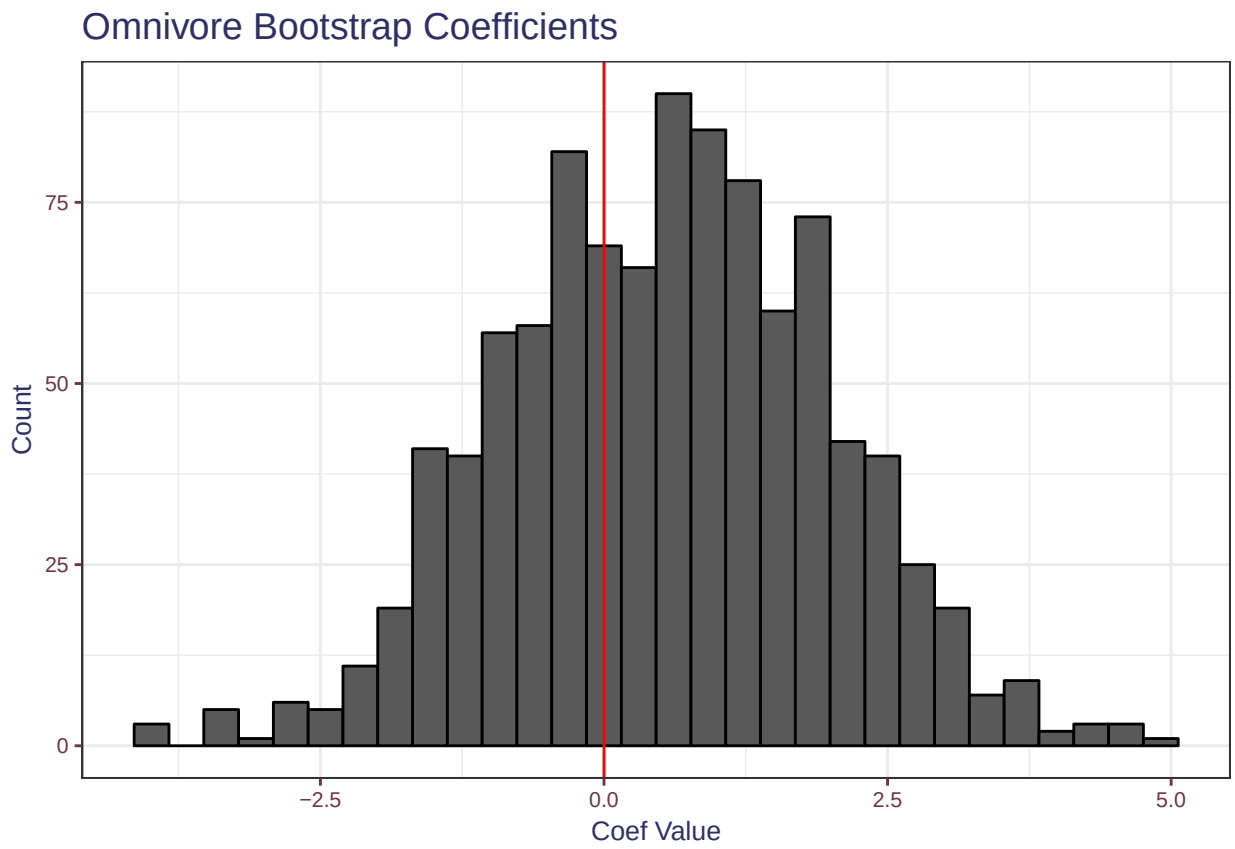
```
  carni   herbi insecti    omni
     19      32       5      20


Call:
lm(formula = sleep_total ~ vore, data = msleep2)

Residuals:
    Min      1Q  Median      3Q     Max
-7.6789 -3.4268 -0.4289  4.0115  9.0211

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  10.3789     1.0245  10.131  1.7e-15 ***
voreherbi    -0.8696     1.2933  -0.672   0.5035
voreinsecti   4.5611     2.2445   2.032   0.0458 *
voreomni      0.5461     1.4306   0.382   0.7038
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.465 on 72 degrees of freedom
Multiple R-squared:  0.0852,    Adjusted R-squared:  0.04708
F-statistic: 2.235 on 3 and 72 DF,  p-value: 0.09143
```

There appears to be a statistically significant difference between insectivores and carnivores, but not between carnivores and herbivores and omnivores. Let's test this with residual bootstrapping!

Below, we look at the results of bootstrap.

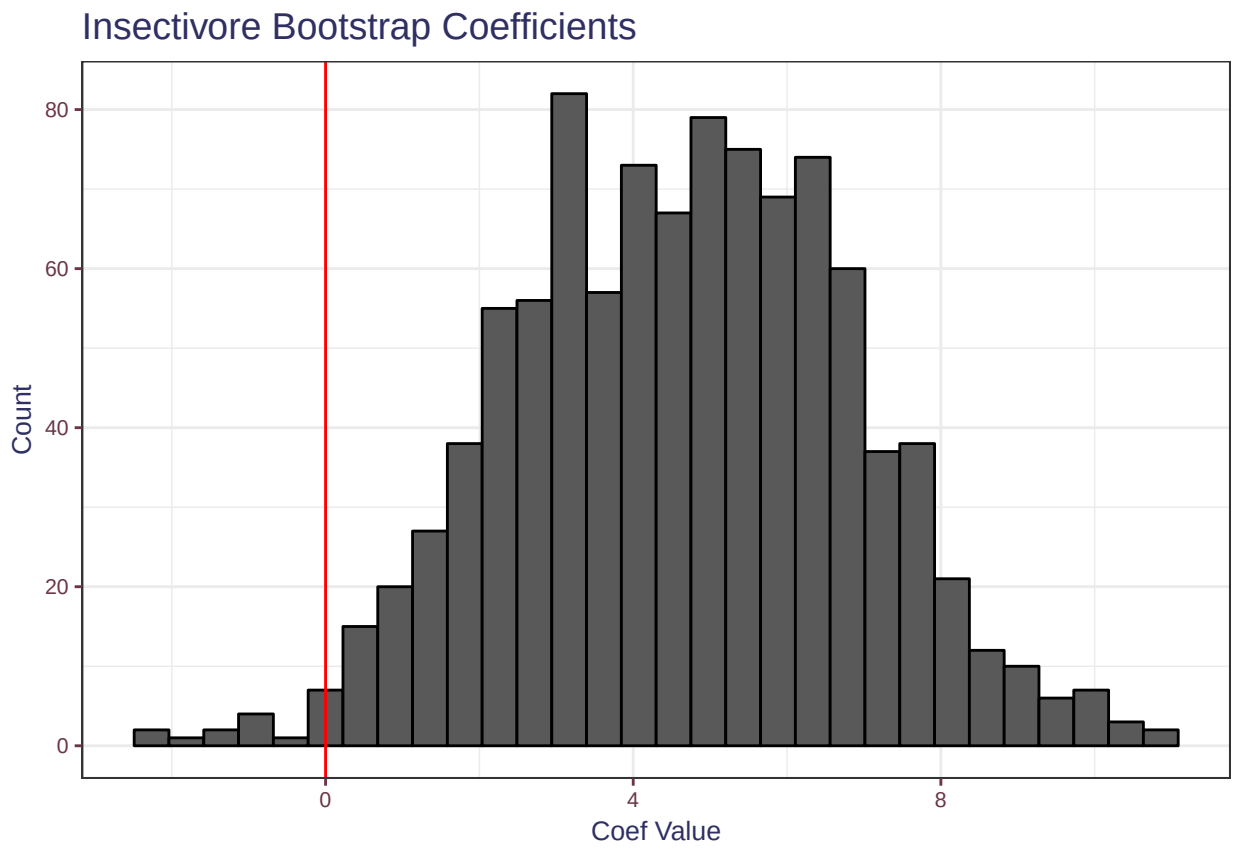



```
[1] 0.246
```




```
[1] 0.356
```

Neither the herbivore or omnivore coefficients are statistically significantly different from 0, with p-values of about 0.5 and 0.7, respectively.




```
[1] 0.013
```

In this case, 13 out of 10000 estimates were less than 0. This is equivalent to a p-value of 0.026 for a two-sided t-test. The 95% confidence interval:

```
     2.5%     97.5%
0.3807241 8.9184733
```

We do see that the confidence interval does not include 0, again indicating statistical significance.

Wild Bootstrap

In the case of heteroskedasticity of residuals in the data, the wild bootstrap method can be used for bootstrapping. The general jist of the method is that, in addition to resampling the residuals, we multiply each element by a random factor drawn from a known symmetric mean 0 distribution. A further explanation of the algorithm can be found in Davidson and Flachaire (2008).

Block Bootstrap

The most common case where the iid assumption is violated in data we would like to bootstrap is in the case of time series data. For these types of data sets, we generally assume that there is some autocorrelation in the variables - see the VAR example for more details. This means that we can not independently resample the observations to create a new data set, since this will destroy any time-related structure in the data. In this case, we use a block bootstrap procedure. The idea is similar to case resampling as explained above; however, instead of resampling individual observations, we divide the data in to (possibly overlapping) blocks of observations, then create new data sets by resampling the blocks. This allows most of the time series structure to be preserved.

In R, the block bootstrap can be run using the `tsboot` function in the `boot` package, with some code finagling. In general, the standard block length is $n^{1/3}$ and we create either blocks of all the same size or blocks of a random size that is geometrically distributed.

In the example below, we analyze the `economics` data set from the `ggplot2` package. We will try to fit an autoregressive model to the monthly unemployment rate. We will estimate coefficients up to a lag of order 5 for the purposes of illustration.

```
Call:
arima(x = economics$unemploy, order = c(5, 0, 0))

Coefficients:
         ar1     ar2      ar3      ar4      ar5  intercept
      1.0553  0.1463  -0.0219  -0.0497  -0.1366   7457.870
s.e.  0.0413  0.0604   0.0606   0.0606   0.0415   1161.813

sigma^2 estimated as 39255:  log likelihood = -3853.12,  aic = 7720.24
```

According to the output, time lags 1, 2, and 5 are statistically significant.

	Estimate	Standard Error
AR1	0.8720042	0.0576840
AR2	0.0006514	0.0688333
AR3	-0.0006776	0.0666954
AR4	-0.0034892	0.0638595
AR5	-0.0077190	0.0504206
Intercept	7800.4630660	425.0696686

The block bootstrap shows that only the 1st order lag is statistically significant to predit unemployment rate at the current time.

Parametric Bootstrap

One other method that is mentioned here is the parametric bootstrap. This method uses the assumed distribution of the parameter estimates as the basis for resampling. The process for this is as follows:

1. Fit our desired model to the original data set; get estimated parameters from this original model.
2. Run k bootstrap iterations. For each iteration:
   (a) Resample the estimated parameters using the assumed distribution.
   (b) Generate a new data set from these new parameters.
   (c) Refit the desired model to this new generated data set.
   (d) Record new parameter estimates on generated data set.
3. Use our collection of k parameter estimates in order to make some sort of statement about the stability of our estimates.

It is not recommended to use this method. In general, bootstrapping allows us to get a sense of the variability or stability of our discoveries without making any distributional assumptions. Thus, this method defeats the purpose of bootstrapping itself. We will not glorify this method with an example, because it doesn't deserve it.