The Stata Journal (2004)
4, Number 3, pp. 312–328

From the help desk: Some bootstrapping techniques

Brian P. Poi
StataCorp

Abstract. Bootstrapping techniques have become increasingly popular in applied econometrics and other areas. This article presents several methods and shows how to implement them using Stata's `bootstrap` command.

Keywords: st0073, bssize initial, bssize refine, bssize analyze, bssize cleanup, bootstrap, confidence intervals, percentile-t, dependent processes

1 Introduction

In a past installment of *From the help desk*, Guan (2003) discusses how to use Stata to obtain bootstrapped standard errors. This article introduces several more bootstrapping techniques. First we show how to obtain bootstrapped confidence intervals for parameters based on asymptotically pivotal statistics. We then illustrate a simple technique that can be used for bootstrapping time-series data. Finally, we introduce a method and Stata command to help us choose the number of bootstrap replications.

Suppose that there is a population distribution $\mathcal{F}$ and that we are interested in estimating a parameter $\theta = \theta(\mathcal{F})$. We have a sample of data F and obtain the estimate $\widehat{\theta} = \theta(F)$. Let N denote the number of observations in F. In the simplest case, the bootstrap proceeds as follows. We draw a random sample F_i of size N with replacement from F and compute $\widehat{\theta}_i = \theta(F_i)$. We repeat this many times, obtaining the set of estimates $\{\widehat{\theta}_i\}_{i=1}^{i=B}$, where B is a large number, such as 1,000. Later we will discuss a method for choosing B, but for now any large value will suffice. Guan (2003) emphasizes using the standard deviation of $\{\widehat{\theta}_i\}$ as the standard error of $\widehat{\theta}$. We instead discuss how to use $\{\widehat{\theta}_i\}$ to obtain confidence intervals for θ.

2 Confidence intervals

We consider four ways to use $\{\widehat{\theta}_i\}$ to construct a $1-\alpha$ confidence interval for $\widehat{\theta}$. The normal approximation, percentile, and bias-corrected methods are implemented by Stata's `bootstrap` command. The fourth, the percentile-t method, is not difficult to implement in many cases, and both theory and Monte Carlo evidence have shown that it provides quite accurate confidence intervals in many applications.

2.1 Normal approximation confidence intervals

One way to obtain a confidence interval for θ is to use a normal approximation. Let $\sigma_{\widehat{\theta}}$ denote the standard error of $\widehat{\theta}$. Then a symmetric $1-\alpha$ confidence interval would be

$$\left(\widehat{\theta} - \sigma_{\widehat{\theta}} z_{1-\alpha/2}, \widehat{\theta} + \sigma_{\widehat{\theta}} z_{1-\alpha/2}\right) \tag{1}$$

where $z_{1-\alpha/2}$ is defined such that $\Phi(z_{1-\alpha/2}) = 1 - \alpha/2$ and $\Phi(\cdot)$ denotes the standard normal distribution function. Traditionally, $\sigma_{\widehat{\theta}}$ is estimated using an asymptotic formula. For example, if θ is a maximum likelihood estimator, $\sigma_{\widehat{\theta}}$ could be based on Fisher's information matrix. If N is small, a t distribution with $N-1$ degrees of freedom could be used instead of the normal.

In finite samples, however, an asymptotically justified estimator of $\sigma_{\widehat{\theta}}$ may perform poorly. Instead, we could estimate $\sigma_{\widehat{\theta}}$ by calculating the sample standard deviation of $\{\widehat{\theta}_i\}$ and then use (1) to compute the confidence interval. Such a confidence interval is known as a "bootstrapped normal-approximation confidence interval", the key being that we use the sampling distribution of the bootstrapped $\{\widehat{\theta}_i\}$ to approximate the true sampling distribution of the estimator. The normal approximation confidence interval is justifiable if the estimator used to obtain $\widehat{\theta}$ has an asymptotically normal distribution, as most estimators typically used in econometrics do. As Mooney and Duval (1993, 36) point out, this method only makes use of the standard deviation of $\{\widehat{\theta}_i\}$ and does not exploit the entire sampling distribution of $\{\widehat{\theta}_i\}$, so other methods may provide confidence intervals with better coverage.

2.2 Percentile confidence intervals

Alternatively, we could calculate a confidence interval for $\widehat{\theta}$ by using $\{\widehat{\theta}_i\}$ as the empirical distribution function for $\widehat{\theta}$. To obtain an asymmetric percentile confidence interval, we simply find $c_{\alpha/2}$ and $c_{1-\alpha/2}$, the $\alpha/2$ and $1-\alpha/2$ centiles of $\{\widehat{\theta}_i\}$. For the endpoints of the confidence interval to be exact, $\alpha(B+1)/2$ must be an integer; if not, interpolation can be used to estimate $c_{\alpha/2}$ and $c_{1-\alpha/2}$. Typically $\widehat{\theta}$ will not lie at the midpoint of $c_{\alpha/2}$ and $c_{1-\alpha/2}$, and asymmetric percentile confidence intervals will therefore provide better coverage than normal approximations when $\{\widehat{\theta}_i\}$ is skewed or even polymodal. Asymmetric percentile confidence intervals are also known as equal-tailed percentile confidence intervals, because equal probability is placed in each tail.

2.3 Bias-corrected and BC$_a$ confidence intervals

Both methods discussed so far assume that $\widehat{\theta}$ is an unbiased estimator of θ. In many cases, $\widehat{\theta}$ may be consistent but nevertheless biased in finite samples. Moreover, the standard error of $\widehat{\theta}$ often depends on θ, and percentile methods tend not to work well in these cases (Davison and Hinkley 1997, 195). Bias-corrected (BC) confidence intervals are designed for these situations and assume that there is a monotonic transformation ϕ

such that $[\phi(\widehat{\theta}) - \phi(\theta)] \xrightarrow{D} N(-z_0, \tau^2)$, where z_0 is a bias constant and τ is the constant standard error of $\phi(\widehat{\theta})$. Conditional on z_0, we can then use that limiting distribution to obtain a confidence interval for $\phi(\widehat{\theta})$ and then apply the inverse function ϕ^{-1} to the endpoints to obtain the confidence interval for $\widehat{\theta}$. In fact, we need not even know the functional form of ϕ; all we require is that such a function exist.

Bias-corrected and accelerated (BC$_a$) confidence intervals improve on their bias-corrected counterparts by allowing the variance of $\phi(\widehat{\theta})$ to depend on θ. Specifically, BC$_a$ confidence intervals are predicated on the assumption $[\phi(\widehat{\theta}) - \phi(\theta)] \xrightarrow{D} N(-z_0\tau_\phi, \tau_\phi^2)$, where $\tau_\phi = 1 + a\phi(\widehat{\theta})$ and a is known as an acceleration parameter. Efron (1987) shows that these confidence intervals have better asymptotic properties than the traditional ones based on a normal approximation. To implement BC confidence intervals, we must know z_0, and to implement BC$_a$, we must also know a. Derivations of estimators of these parameters are given in Efron (1987) and Davison and Hinkley (1997, section 5.3), and the formulas are also shown in [R] **bootstrap**.

2.4 Percentile-t confidence intervals

Hall (1992), Horowitz (2001), and others have argued that bootstrap methods based on asymptotically pivotal statistics usually provide better coverage properties than those that are based on nonpivotal statistics. As an example, suppose that we wish to obtain a bootstrapped confidence interval for β_1 in the simple linear regression model

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \tag{2}$$

where we assume that ϵ_i has zero mean and is independent of ϵ_j for $j \neq i$. As described above, one way to proceed would be to draw random samples F_i with replacement from F and record the sequence $\{\widehat{\beta}_{1i}\}$. We could then use the normal approximation, percentile, or BC$_a$ methods to obtain a confidence interval.

Alternatively, suppose that, instead of recording $\widehat{\beta}_{1i}$ at each bootstrap replication, we record the statistic $t_i = (\widehat{\beta}_{1i} - \widehat{\beta}_1)/\widehat{\sigma}_{\widehat{\beta}_{1i}}$, where $\widehat{\sigma}_{\widehat{\beta}_{1i}}$ is an estimate of the standard error of $\widehat{\beta}_{1i}$ and $\widehat{\beta}_1$ is the estimate of β_1 using the original dataset. t_i is asymptotically pivotal, since under very mild conditions $(\widehat{\beta}_{1i} - \widehat{\beta}_1)/\widehat{\sigma}_{\widehat{\beta}_{1i}}$ converges in distribution to a random variable following Student's t-distribution with the appropriate degrees of freedom.

The notation t_i is not accidental, for it is immediately recognized as the usual t statistic for the hypothesis test $H_0\colon \widehat{\beta}_{1i} = \widehat{\beta}_1$. In fact, constructing a confidence interval can be viewed as an exercise in which we treat $\widehat{\beta}_1$ as a fixed parameter and then find the set of $\widehat{\beta}_{1i}$s for which H_0 cannot be rejected. The following is known as the percentile-t confidence interval method. Suppose that we want confidence intervals for $\widehat{\beta}_0$ and $\widehat{\beta}_1$. The technique involves several steps:

1. Fit (2) by OLS using the original dataset F and obtain $\widehat{\beta}_0$ and $\widehat{\beta}_1$; these are the point estimates for the model. Also record their estimated standard errors $\widehat{\sigma}_{\widehat{\beta}_0}$ and $\widehat{\sigma}_{\widehat{\beta}_1}$.

2. Draw a random sample of size N with replacement from F, fit (2) by OLS, and compute

$$t_{\beta_{0i}} = (\widehat{\beta}_{0i} - \widehat{\beta}_0)/\widehat{\sigma}_{\widehat{\beta}_{0i}} \qquad \text{and} \qquad t_{\beta_i} = (\widehat{\beta}_{1i} - \widehat{\beta}_1)/\widehat{\sigma}_{\widehat{\beta}_{1i}}$$

3. Repeat step 2 B times, obtaining the sequences $\{t_{\beta_{0i}}\}_{i=1}^{i=B}$ and $\{t_{\beta_{1i}}\}_{i=1}^{i=B}$.

4. Find the $\alpha/2$ and $1-\alpha/2$ centiles of $\{t_{\beta_{0i}}\}$; call them $t^L_{\beta_0}$ and $t^H_{\beta_0}$, respectively. The $1-\alpha$ confidence interval for β_0 is then

$$\left[\widehat{\beta}_0 - t^H_{\beta_0}\widehat{\sigma}_{\widehat{\beta}_0}, \widehat{\beta}_0 - t^L_{\beta_0}\widehat{\sigma}_{\widehat{\beta}_0}\right]$$

5. Repeat step 4 for $\{t_{\beta_{1i}}\}$, yielding

$$\left[\widehat{\beta}_1 - t^H_{\beta_1}\widehat{\sigma}_{\widehat{\beta}_1}, \widehat{\beta}_1 - t^L_{\beta_1}\widehat{\sigma}_{\widehat{\beta}_1}\right]$$

Of course, one shortcoming of the percentile-t method is the need to have an available estimate of the standard error of the parameter of interest for each bootstrap replicate. In many cases, the bootstrap is used precisely because we have no way to obtain such standard errors analytically. However, for regression models, a heteroskedasticity-consistent covariance matrix is readily available, and for maximum likelihood models, one of several covariance matrix estimators could be employed.

Another alternative is to use a bootstrap-within-bootstrap procedure, also known as a nested bootstrap. Suppose that we wish to obtain the percentile-t confidence interval for a parameter θ but we do not have an analytic formula for its standard error. In a nested bootstrap procedure, we again begin by drawing a bootstrap sample from the original dataset and compute $\widehat{\theta}_i$. We then treat this bootstrap sample as if it were our original dataset and perform bootstrapping on it to obtain an estimate of the standard error of $\widehat{\theta}_i$, which we can then use to compute a t statistic. We repeat this entire process many times and then calculate the percentile-t confidence interval.

2.5 A Stata example

Using the venerable `auto` dataset, we illustrate how to obtain nominal 95% percentile-t confidence intervals for β_0, β_1, and β_2 in the regression model

$$\texttt{mpg}_i = \beta_0 + \beta_1\texttt{gear\_ratio}_i + \beta_2\texttt{foreign}_i + \epsilon_i$$

Note that, instead of recording the t statistics at every bootstrap replication, we can just record the estimated parameters and their standard errors and then compute the t

ratios later. That shortcut simplifies the use of the bootstrap command. To allow for the possibility of heteroskedasticity of ϵ_i, we use the hc2 option with regress to obtain heteroskedasticity-consistent standard errors.

First we load the dataset and perform the bootstrap:

```
. sysuse auto, clear
. set seed 1
. bootstrap "regress mpg gear_ratio foreign, hc2" _b _se, reps(1000) dots
> saving(bsdata) replace
```

The output shows the normal approximation, percentile, and bias-corrected confidence intervals discussed above for both the parameters and their standard errors. Typically, we are not concerned with obtaining confidence intervals for estimated standard errors, but they are included in the output since we told the bootstrap command to record the standard errors.

Next we estimate the parameters using the original dataset and then generate the t ratios:

```
. regress mpg gear_ratio foreign, hc2
. use bsdata
. generate t_gear_ratio = (b_gear_ratio - _b[gear_ratio]) / se_gear_ratio
. generate t_foreign = (b_foreign - _b[foreign]) / se_foreign
. generate t_cons = (b_cons - _b[_cons]) / se_cons
```

Finally we find the 2.5th and 97.5th centiles of the t ratios and calculate the confidence intervals. For gear_ratio, we have

```
. _percentile t_gear_ratio, percentiles(2.5 97.5)

. display _b[gear_ratio] - _se[gear_ratio]*r(r2)
5.7053247

. display _b[gear_ratio] - _se[gear_ratio]*r(r1)
11.686533
```

Thus the percentile-t confidence interval for $\widehat{\beta}_1$ is $[\,5.705, 11.687\,]$. You may verify that the percentile-t confidence interval for $\widehat{\beta}_0$ is $[\,-12.769, 3.990\,]$ and that for $\widehat{\beta}_2$ it is $[\,-4.252, 2.478\,]$.

3 Dependent processes

Thus far we have assumed that the observations within the dataset F are independently distributed, and that has allowed us to resample observations from the dataset to generate bootstrap samples. While that assumption is certainly plausible with randomly sampled cross-sectional datasets, it is too restrictive for use with time-series data. An observation on a variable x taken at time t is often correlated with the value of x taken at time $t-1$ and perhaps other time periods as well. Treating x_t as a completely random observation in a bootstrap resampling scheme is inappropriate, because the resulting bootstrap sample would not reflect the fact that x_t depends on x_{t-1}. We therefore need another way to generate bootstrap samples that preserves the temporal dependence of the data.

3.1 Creating bootstrap samples

We illustrate how to create bootstrap samples by using one of the most common applications. Consider a simple linear regression on time-series data with first-order serially correlated errors:

$$y_t = \beta_0 + \beta_1 x_t + \epsilon_t, \qquad \epsilon_t = \rho\epsilon_{t-1} + \nu_t \tag{3}$$

where ν_t is white noise and our sample ranges from $t = 1 \dots T$. For the moment, we will also assume that ν_t is $\mathrm{N}(0, \sigma_\nu)$, though later we will relax the normality assumption. To account for the serially correlated errors, we use Prais–Winsten regression on our dataset to obtain $\widehat{\beta}_0$, $\widehat{\beta}_1$, $\widehat{\rho}$, and $\widehat{\sigma}_\nu$. We then use these estimates to calculate the residuals $\widehat{\epsilon}_t = y_t - \widehat{\beta}_0 - \widehat{\beta}_1 x_t$.

Obtaining bootstrapped standard errors or confidence intervals for β_0, β_1, and ρ involves simulating the data-generating process for ϵ_t. First consider period $t = 1$. We have $\epsilon_1 = \rho\epsilon_0 + \nu_1$. The unconditional expected value of ϵ_t is zero for all t, so we could create a simulated value ϵ_1^* by taking a random draw from the $\mathrm{N}(0, \widehat{\sigma}_\nu)$ distribution. For $t = 2$, we have $\epsilon_2 = \rho\epsilon_1 + \nu_2$, and we can simulate this value as $\epsilon_2^* = \widehat{\rho}\epsilon_1^* + \nu_2^*$, where we again draw from the $\mathrm{N}(0, \widehat{\sigma}_\nu)$ distribution to obtain a value for ν_2^*. We then repeat this process for the remaining observations, obtaining the sequence $\epsilon_1^*, \dots, \epsilon_T^*$.

Now we form a bootstrap sample as

$$y_t^* = \widehat{\beta}_0 + \widehat{\beta}_1 x_t + \epsilon_t^*$$

where, by construction, $\epsilon_t^* = \widehat{\rho}\epsilon_{t-1}^* + \nu_t^*$ and ν_t^* is a white noise process with a $\mathrm{N}(0, \widehat{\sigma}_\nu)$ distribution. Notice the striking similarity between our bootstrap sample and the underlying data-generating process shown in (3). The only difference is that our bootstrap sample was constructed using consistent estimates of the population parameters. As in any bootstrapping procedure, the goal is to draw bootstrap samples that approximate the true data-generating process as closely as possible.

We use Prais–Winsten regression on this bootstrap sample, obtaining estimates $\widehat{\beta}_{01}$, $\widehat{\beta}_{11}$, and $\widehat{\rho}_1$. Repeatedly creating bootstrap samples and fitting the model yields a sequence of parameter estimates. To obtain bootstrapped standard errors, we simply calculate the sample standard deviations of $\{\widehat{\beta}_{0i}\}$, $\{\widehat{\beta}_{1i}\}$, and $\{\widehat{\rho}_i\}$. To obtain confidence intervals, we can use any of the techniques discussed in section 2.

Two shortcomings of the previous procedure are the conditioning on $\epsilon_0 = 0$ and the assumption of normality of ν_t. Instead of simply setting ϵ_0 to zero, we could instead pick a random number s between 1 and T and set $\epsilon_0 = \widehat{\epsilon}_s$. To relax the normality assumption, we can use a technique known as "residual resampling". Given the residuals $\widehat{\epsilon}_t$ and $\widehat{\rho}$ for $t = 2 \dots T$, we can compute

$$\widehat{\nu}_t = \widehat{\epsilon}_t - \widehat{\rho}\widehat{\epsilon}_{t-1}$$

In computing the bootstrap samples, instead of drawing ν_t^* from a $\mathrm{N}(0, \widehat{\sigma}_\nu)$ distribution, we pick a random number r between 2 and T and set $\nu_t^* = \widehat{\nu}_r$. This is often called a nonparametric bootstrap, in contrast to a parametric bootstrap in which we draw ν_t^* from a $\mathrm{N}(0, \widehat{\sigma}_\nu)$ distribution. However, in the literature on bootstrapping time-series data, the terms *parametric* and *nonparametric* are often used more generally. Both of the techniques we have discussed are parametric bootstraps in the sense that we have parametrized the data-generating process for y_t by (3)[1].

Many time-series applications, such as ARIMA and VAR models, do not contain any exogenous variables, so a further refinement is possible. Consider the simple AR(1) model

$$y_t = \rho y_{t-1} + \nu_t$$

where again ν_t is white noise. Instead of creating a series of length T as before, we can instead create a series of length $T_0 + T$, where T_0 is, say, $1,000$. We then discard the first T_0 observations and estimate the parameters of our model using the remaining T observations. By doing this, the series is not influenced as much by the starting value y_0 used.

3.2 A Stata example

The manual entry [TS] **prais** fits the model

$$\texttt{usr}_t = \beta_0 + \beta_1 \texttt{idle}_t + \epsilon_t$$

where ϵ_t is believed to exhibit serial correlation. Here we will illustrate how to obtain bootstrapped standard errors and percentile-t confidence intervals for β_0 and β_1 estimated using the Prais–Winsten technique. Because Stata's `bootstrap` command is not amenable to residual resampling, we will create our own bootstrap samples and use the `post` commands to save the results to a file. We can, however, use the `bstat` command afterward to obtain the standard errors and several confidence intervals.

The first step is to fit the model on the original dataset and save all the estimated parameters and variables that we will need later on:

```
. webuse idle, clear
. tsset t
. prais usr idle
. predict double ehat, residuals
. scalar bidle = _b[idle]
. scalar bcons = _b[_cons]
. scalar seidle = _se[idle]
. scalar secons = _se[_cons]
. scalar rho = e(rho)
. scalar sigmanu = e(rmse)
. scalar N = e(N)
. generate double nuhat = ehat - rho*L.ehat
```

With these estimates in hand, we now write a short program that opens a post file, repeatedly creates bootstrap samples using the nonparametric technique just discussed, calls `prais`, and posts the results. Our program takes two arguments: the number of observations in the dataset and the number of bootstrap replications to perform.

[1] See Berkowitz and Kilian (2000) for an overview of parametric and nonparametric bootstrap procedures for time-series data.

Because we are interested in obtaining both standard errors and percentile-t confidence intervals, we record the parameter estimates. If we were only interested in the percentile-t confidence intervals, we could instead record the t ratios directly and save a couple of lines of code later.

```
program bootit
        version 8.2
        args capt numboots

        tempname results
        postfile `results' b_idle b_cons se_idle se_cons ///
                        using bsdata, replace
        tempvar usrhat eps
        forvalues i = 1/`numboots' {
                local s = int(uniform()*`capt' + 1)  // For e_0
                local r = int(uniform()*(`capt' - 1) + 2) // For nu_1
                quietly {
                        generate double `eps' = ehat[`s'] + nuhat[`r'] in 1
                        replace `eps' = rho*L.`eps' +  ///
                                nuhat[int(uniform()*(`capt' - 1) + 2)] ///
                                in 2/`capt'
                        generate double `usrhat' = bcons + bidle*idle + `eps'
                        prais `usrhat' idle
                }
                post `results' (_b[idle]) (_b[_cons]) (_se[idle]) (_se[_cons])
                drop `usrhat' `eps'
                display "." _c
        }
        display
        postclose `results'
end
```

Next we set the random number seed so that the results are replicable, and we call the program with the sample size we previously stored in the scalar N and the number of replications we want:

```
. set seed 1
. bootit `=N' 1000
```

Stata's bstat command produces a table of standard errors and confidence intervals using our post file if we pass it a matrix containing the parameter estimates of the model obtained using the original dataset:

```
. matrix stats = (bidle, bcons)
. bstat b*, stat(stats)
```

Finally we compute the percentile-t confidence intervals using the procedure illustrated in section 2.5. The table below summarizes the results:

Parameter	Observed	Asy. SE	BS SE	95% Conf. Interval		Type*
β_0	15.2042	4.1604	4.5589	6.2580	24.1503	(N)
				6.1315	24.4168	(P)
				6.0490	24.2711	(BC)
				4.5666	25.9120	(Pt)
β_1	−0.1357	0.0472	0.0518	−0.2373	−0.0340	(N)
				−0.2374	−0.0300	(P)
				−0.2359	−0.0277	(BC)
				−0.2622	−0.0181	(Pt)

* Denotes the type of bootstrapped confidence interval: (N) normal approximation, (P) percentile, (BC) bias-corrected, and (Pt) percentile-t

In this particular case, the bootstrapped standard errors exceed their asymptotic counterparts reported by `prais`, and the percentile-t confidence intervals are somewhat wider than the others. Since the dataset has only 30 observations, we are not surprised by either of these findings.

4 On the number of replications

In the previous examples, we have chosen to use 1,000 bootstrap replications, though we have not given any formal justification for using this number. On the one hand, using more replications provides more accurate bootstrapped statistics, yet in practice we must weigh the benefit of more bootstrap replications against the computational cost associated with them. Usually, we tend to use fewer replications with complicated estimators that require computationally intensive iterative procedures because of the time involved, yet arguably those estimators require more replications.

In this section, we outline a method of choosing the number of bootstrap replications B developed by Andrews and Buchinsky (2000), and we present a Stata command to help automate the process. Andrews and Buchinsky's method applies to a wide variety of bootstrapped statistics and applications, though here we focus on selecting B only for standard errors and percentile-t confidence intervals. Davidson and MacKinnon (2000) develop an alternative method for choosing the number of replications, and they show that their method typically selects a smaller B than Andrews and Buchinsky's method; however, Davidson and MacKinnon's method applies only to bootstrap p-values, so we do not consider their method here.

We have an estimator θ and are interested in λ, a measure of the reliability of θ. For example, λ could be the standard error or an endpoint of a percentile-t confidence interval of a regression coefficient. If bootstrap replications were costless, we would use an infinite number of them and obtain the "ideal" bootstrap estimator $\widehat{\lambda}_\infty$. However, bootstrap replications are costly, at least in terms of time, so we perform a finite number

B of them, yielding the estimator $\widehat{\lambda}_B$. We want to choose B such that $\widehat{\lambda}_B$ is "close" to $\widehat{\lambda}_\infty$.

The method developed by Andrews and Buchinsky (2000) requires us to specify two parameters. The percentage deviation, or *pdb*, indicates the maximum desired percentage difference between $\widehat{\lambda}_B$ and $\widehat{\lambda}_\infty$. A typical value of *pdb* would be 5% or 10%. The second parameter, τ, defines the probability $1-\tau$ with which $\widehat{\lambda}_B$ and $\widehat{\lambda}_\infty$ differ by more than *pdb* percent. If $pdb = 5\%$ and $\tau = 0.10$, Andrews and Buchinsky's method lets us choose B such that

$$\Pr\left(100\frac{\left|\widehat{\lambda}_B - \widehat{\lambda}_\infty\right|}{\widehat{\lambda}_\infty} \leq 5\%\right) = 0.90$$

Note that we are not making any claims about the relationship between $\widehat{\lambda}_B$ and any asymptotic estimator of λ or the "true" value of λ; we are only comparing $\widehat{\lambda}_B$ with $\widehat{\lambda}_\infty$. If the bootstrap is an inconsistent estimator of λ, the choice of B is moot[2].

4.1 Standard errors

First consider the case in which λ is a standard error. Andrews and Buchinsky (2000) show that

$$\sqrt{B}\frac{\widehat{\lambda}_B - \widehat{\lambda}_\infty}{\widehat{\lambda}_\infty} \xrightarrow{D} \mathrm{N}(0,\omega) \tag{4}$$

where

$$\omega = \frac{2+\gamma}{4}$$

and γ is a measure of the excess kurtosis of $\{\widehat{\theta}_i\}_{i=1}^{i=B}$:

$$\gamma \equiv \frac{\mathrm{E}\left[\left\{\widehat{\theta}_i - \mathrm{E}\left(\widehat{\theta}_i\right)\right\}^4\right]}{\left(\mathrm{E}\left[\left\{\widehat{\theta}_i - \mathrm{E}\left(\widehat{\theta}_i\right)\right\}^2\right]\right)^{1/2}} - 3$$

In the case of standard errors, convergence is defined in terms of a fixed sample size and B tending to infinity, while for sample quantiles (as used, for example, with percentile-t confidence intervals), convergence is defined in terms of both the sample size and B tending to infinity. Given the result shown in (4), we can write

$$\Pr\left\{\frac{\left|\widehat{\lambda}_B - \widehat{\lambda}_\infty\right|}{\widehat{\lambda}_\infty} \leq z_{1-\frac{\tau}{2}}\left(\frac{\omega}{B}\right)^{1/2}\right\} \longrightarrow 1-\tau$$

[2] An example in which the bootstrap is inconsistent is a bootstrap hypothesis test of a parameter on the boundary of the parameter space, such as a test that a variance term is zero.

or

$$pdb \approx 100 z_{1-\frac{\tau}{2}} \left(\frac{\omega}{B}\right)^{1/2} \tag{5}$$

so that

$$B \approx 10{,}000 z_{1-\frac{\tau}{2}}^2 \omega / pdb^2 \tag{6}$$

Because γ is unknown, Andrews and Buchinsky propose a three-step method for choosing B:

1. Set γ equal to zero and evaluate (6) to obtain an initial estimate B_1 of B. Setting γ to zero corresponds to assuming that $\{\widehat{\theta}_i\}$ has no excess kurtosis.
2. Perform B_1 bootstrap replications, and then compute
$$\widehat{\omega}_B = \frac{2+\widehat{\gamma}_B}{4}$$
where
$$\widehat{\gamma}_B \equiv \frac{\frac{1}{B_1-1}\sum_{i=1}^{i=B_1}\left(\widehat{\theta}_i - \frac{1}{B_1}\sum_{j=1}^{j=B_1}\widehat{\theta}_j\right)^4}{\left\{\frac{1}{B_1-1}\sum_{i=1}^{i=B_1}\left(\widehat{\theta}_i - \frac{1}{B_1}\sum_{j=1}^{j=B_1}\widehat{\theta}_j\right)^2\right\}^{1/2}}$$
3. Re-evaluate (6) using $\omega = \widehat{\omega}_B$, and then choose $B^* = \max(B_1, B)$. If $B_1 \geq B^*$, compute the bootstrapped statistics of interest. If $B_1 < B^*$, perform an additional $B - B_1$ replications, then compute the bootstrapped statistics of interest.

In practice, we are usually interested in obtaining bootstrapped confidence intervals or standard errors for several parameters at once. In those cases, we calculate B_1 for each statistic individually and then take the maximum as our final choice. Similarly, in the second and third steps, we compute $\widehat{\omega}_B$ and B^* for each statistic individually and then choose as our final value of B^* the maximum calculated for each statistic. If we use the same (pdb, τ) pair for each statistic, we only need to calculate a single B_1, since it will be the same for all statistics.

4.2 Percentile-t confidence intervals

The three-step method can also be used for percentile-*t* confidence intervals with a few modifications. Consider a $1-2\alpha$ confidence interval. There are two bootstrapped statistics with which we are concerned: the α centile of $\{t_i\}$ and the $1-\alpha$ centile of $\{t_i\}$, where $\{t_i\}$ is the sequence of bootstrapped t statistics. To avoid having to use interpolation to find the α and $1-\alpha$ quantiles of $\{\widehat{\theta}_i\}$, we will choose B in such a way that $\eta/(B+1) = 1-\alpha$ where η is a positive integer, $\alpha = \alpha_1/\alpha_2$, and α_1 and α_2 are positive integers with no common divisors.

Andrews and Buchinsky show that, in the case of percentile-*t* confidence intervals, (4) continues to hold with

$$\omega = \frac{\alpha(1-\alpha)}{z_{1-\alpha}^2 \phi^2(z_{1-\alpha})} \tag{7}$$

For the $1-\alpha$ centile, the three-step method proceeds as follows:

1. Compute ω as given in (7), and then compute

   $$B_1 = \alpha_2 h_1 - 1$$

   where

   $$\begin{aligned} \alpha &= \alpha_1/\alpha_2 \quad \text{(fraction in reduced form)} \\ h_1 &= \text{int}\left(\frac{10,000 z^2_{1-\tau/2}\omega}{pdb^2 \alpha_2}\right) \end{aligned}$$

2. Perform B_1 bootstrap replications, and then compute

   $$\widehat{\omega}_B = \frac{\alpha(1-\alpha)(1/\widehat{g}_B)^2}{\widehat{q}_{1-\alpha,B}}$$

   where

   $$\begin{aligned} 1/\widehat{g}_B &= \frac{B}{2\widehat{m}_B}(t_{\eta+\widehat{m}_B} - t_{\eta-\widehat{m}_B}) \\ \eta &= (1-\alpha)(B+1) \\ \widehat{m}_B &= \text{int}(c_\alpha B^{2/3}) \\ \widehat{q}_{1-\alpha,B} &= t_\eta \\ t_\eta &= \eta\text{th order statistic of } \{t_i\}_{i=1}^{i=B_1} \\ c_\alpha &= \left\{\frac{1.5 z^2_{1-\alpha/2}\phi^2(z_{1-\alpha})}{2z^2_{1-\alpha/2}+1}\right\}^{1/3} \end{aligned}$$

3. Compute B as in step 1, using ω_B in place of ω, and set $B^*_{1-\alpha} = \max(B_1, B)$.

The same steps are used for the α centile, except that $\eta = \alpha(B+1)$ and $\widehat{q}_{\alpha,B}$ is used in place of $\widehat{q}_{1-\alpha,B}$. The total number of bootstrap replications needed is then $B^* = \max(B^*_{1-\alpha}, B^*_\alpha)$. As in the case for standard errors, if you are bootstrapping multiple confidence intervals at once, choose B^* to be the maximum number of replications calculated for any single confidence interval.

4.3 Postestimation analysis

Recall that (5) expresses *pdb* as a function of B, and note that we can rewrite (6) as

$$\tau = 2\left[1 - \Phi\left\{\frac{pdb\,(B/\omega)^{1/2}}{100}\right\}\right] \tag{8}$$

Thus given a set of bootstrap estimates $\{\widehat{\theta}_i\}_{i=1}^{i=B}$, we can use step 2 of the three-step method to obtain an estimate of ω. We can then use (5) to answer the question "With probability $1-\tau$, what is the maximum percentage by which our bootstrapped statistic differs from the one we would obtain if we performed an infinite number of replications?" Equation (8) answers the question, "What is the probability that our bootstrapped statistic differs from the ideal bootstrap by more than *pdb* percent?"

4.4 Stata implementation

The command `bssize` accompanying this article makes the preceding method easy to use. We first provide the syntax diagrams and summary of the options, and then we give an example showing how to use it.

Syntax

`bssize initial ,` [`tau(`#`)` `pdb(`#`)` `pctt(`#`)`]

`bssize refine using` *filename*

`bssize analyze` [`using` *filename*] `,` [`tau(`#`)` `pdb(`#`)` `append(`*filename*`)`]

`bssize cleanup`

filename is the name of a file created by the `bootstrap` command or the `postfile` command.

Description

`bssize initial` provides an initial estimate of the number of bootstrap replications needed; it corresponds to step 1 of Andrews and Buchinsky's three-step method.

`bssize refine` takes as input a file containing the bootstrapped values of the statistics and produces a refined estimate of the number of bootstrap replications needed; it corresponds to steps 2 and 3 of the three-step method.

`bssize analyze` takes as input a file containing the bootstrapped values of the statistics and performs postestimation analysis of the accuracy of the bootstrapped standard errors. See section 4.3 above. `bssize analyze` is not implemented for use with percentile-*t* confidence intervals.

`bssize cleanup` clears all the global macros set by `bssize initial` to store and pass information to `bssize refine`. All such macro names begin with `BSS_`.

Options

`tau(`*#*`)` specifies the probability $1-\tau$ with which $\widehat{\lambda}_B$ and $\widehat{\lambda}_\infty$ differ by no more than *pdb* percent. The default for `bssize initial` is 0.05. `tau()` is required with `bssize analyze` if `pdb()` is not specified.

`pdb(`*#*`)` specifies the maximum percentage deviation between $\widehat{\lambda}_B$ and $\widehat{\lambda}_\infty$ that is acceptable with probability $1-\tau$. The default for `bssize initial` is 5%. `pdb()` is required with `bssize analyze` if `tau()` is not specified.

`pctt(`*#*`)` specifies the size, as a percentage, of the percentile-t confidence intervals being produced. If `pctt()` is not specified, `bssize` assumes that standard errors are being bootstrapped instead of confidence intervals.

`append(`*filename*`)` (`bssize analyze` only) indicates an additional file that is to be appended to the `using` file before doing the postbootstrap analysis. For example, you might perform an initial number of bootstrap replications, save the results under one file name, perform additional replications, and save them under a different name. `append()` obviates the need to use Stata's `append` command to create one dataset before calling `bssize analyze`.

Remarks

If standard errors are being bootstrapped, the `using` dataset must contain a variable named `b_`*statistic* for each *statistic* being bootstrapped.

For percentile-t confidence intervals, the `using` dataset must either contain the variables `b_`*statistic* and `se_`*statistic* representing the point estimate and standard error of *statistic* for each bootstrap replication or contain a variable named `t_`*statistic* representing the t statistic for each *statistic*. If the `t_`*statistics* variables do not exist, the point estimates of *statistics* used to compute the t statistics are obtained from characteristics named `b_`*statistic*`[observed]`; these characteristics are stored with the dataset produced by the `bootstrap` command, so you probably will not need to define the characteristics manually.

Example

Using the `auto` dataset, we want to estimate the parameters of the model

$$\texttt{mpg}_i = \beta_0 + \beta_1\texttt{foreign}_i + \beta_2\texttt{displacement}_i + \epsilon_i$$

and obtain bootstrapped standard errors for β_0, β_1, and β_2. With probability 0.99, we do not want our standard errors to deviate by more than 5% from the ideal bootstrapped values. Thus we choose $pdb = 5$ and $\tau = 0.01$.

The first step is to call `bssize initial` and obtain an estimate of the number of bootstrap replications needed:

```
. bssize initial, tau(0.01) pdb(5)
Initial estimate of bootstrap size needed for standard errors
---------------------------------------------------------------
Percent deviation from Binfinity (pdb)                    5.000
Probability (1 - tau)                                     0.990
---------------------------------------------------------------
Required size (B1)                                         1326
```

Next we call `bootstrap` and perform 1,326 replications. We save the results in the file `one.dta`.

```
. sysuse auto, clear
. set seed 1
. bootstrap "regress mpg foreign displacement" _b, reps(1326) saving(one)
```

With an initial set of bootstrap replications in hand, we can refine our estimate of the final number needed:

```
. bssize refine using one
Refined estimate of bootstrap size needed for standard errors
---------------------------------------------------------------
Percent deviation from Binfinity (pdb)                    5.000
Probability (1 - tau)                                     0.990
---------------------------------------------------------------
   Parameter |  Initial Size    Current Size    Revised Size
-------------+-------------------------------------------------
     foreign |          1326            1326            1447
displacement |          1326            1326            1454
        cons |          1326            1326            1383
---------------------------------------------------------------
Maximum revised size                                       1454
Additional replications needed                              128
```

The column marked “Initial Size” contains the initial estimate of the number of bootstrap replications needed, the column marked “Current Size” contains the number of replications contained in the dataset, and the column marked “Revised Size” shows the revised number of bootstrap replications needed based on steps 2 and 3 of Andrews and Buchinsky’s method. In this case, we need to do an additional 128 replications.

```
. bootstrap "regress mpg foreign displacement" _b, reps(128) saving(two)
```

With all the bootstrapping done, we use `append` to join the two datasets and then call the `bstat` command to obtain standard errors:

```
. use one, clear
(bootstrap: regress mpg foreign displacement)
. append using two
. bstat
Bootstrap statistics                              Number of obs    =        74
                                                  Replications     =      1454

------------------------------------------------------------------------------
Variable     |  Reps  Observed      Bias  Std. Err. [95% Conf. Interval]
-------------+----------------------------------------------------------------
   b_foreign |  1454 -.8006818 -.0198747  1.420449  -3.587032   1.985668   (N)
             |                                      -3.444365   2.136814   (P)
             |                                      -3.393448   2.169905  (BC)
b_displace~t |  1454 -.0469161  .0000582  .0058629  -.0584168  -.0354154   (N)
             |                                      -.0585276  -.0360198   (P)
             |                                      -.0594264  -.0367062  (BC)
      b_cons |  1454  30.79176 -.0504927   1.39613   28.05312   33.53041   (N)
             |                                       27.99683    33.5871   (P)
             |                                       28.05231   33.66282  (BC)
------------------------------------------------------------------------------
Note:  N   = normal
       P   = percentile
       BC  = bias-corrected
```

How confident can we be that our bootstrapped standard errors deviate by no more than 2.5% from the ideal? We can use `bssize analyze` to find out:

```
. bssize analyze using one, append(two) pdb(2.5)
Analysis of bootstrap results for standard errors
---------------------------------------------------------------
Percent deviation (pdb)                                   2.500
---------------------------------------------------------------
   Parameter |   Final Size                tau          1 - tau
-------------+-------------------------------------------------
   b_foreign |         1454              0.195            0.805
b_displace~t |         1454              0.197            0.803
      b_cons |         1454              0.185            0.815
-------------+-------------------------------------------------
Maximum                                  0.197            0.815
```

We see that there is a 19.7% chance that the standard error for the coefficient on `displacement` is more than 2.5% away from the value we would obtain if we could perform an infinite number of bootstrap replications.

5 Conclusion

With the power of modern computers, the bootstrap is a viable alternative to asymptotic standard errors and confidence intervals. This paper has illustrated an alternative method of computing confidence intervals and has shown how to bootstrap simple time-series models. The `bssize` commands introduced here allow you to select a reasonable number of bootstrap replications.

6 References

Andrews, D. W. K. and M. Buchinsky. 2000. A three-step method for choosing the number of bootstrap repetitions. *Econometrica* 68(1): 23–51.

Berkowitz, J. and L. Kilian. 2000. Recent developments in bootstrapping time series. *Econometric Reviews* 19(1): 1–54.

Davidson, R. and J. G. MacKinnon. 2000. Bootstrap tests: How many bootstraps? *Econometric Reviews* 19(1): 55–68.

Davison, A. C. and D. V. Hinkley. 1997. *Bootstrap Methods and their Application.* Cambridge: Cambridge University Press.

Efron, B. 1987. Better bootstrap confidence intervals. *Journal of the American Statistical Association* 82(397): 171–185.

Guan, W. 2003. From the help desk: Bootstrapped standard errors. *Stata Journal* 3(1): 71–80.

Hall, P. 1992. *The Bootstrap and Edgeworth Expansion.* New York: Springer.

Horowitz, J. L. 2001. The bootstrap. In *Handbook of Econometrics*, ed. J. J. Heckman and E. Leamer, vol. 5, 3159–3228. Amsterdam: North-Holland.

Mooney, C. Z. and R. D. Duval. 1993. *Bootstrapping: A Nonparametric Approach to Statistical Inference.* Newbury Park, CA: Sage Publications.

About the Author

Brian P. Poi is a Senior Economist at StataCorp.