

FUNDAMENTALS OF BOOTSTRAPPING AND MONTE CARLO METHODS

William Howard Beasley, Patrick O’Keefe, and Joseph Lee Rodgers

One of the most modern, and valuable, statistical innovations is the class of statistical procedures that uses simulations based on observed data to generate useful distributions, such as sampling distributions, and features of those distributions, such as standard errors. Until the development of modern, high-speed computing and effective software, such methods were intractable (and therefore undeveloped). Both Fisher and Gosset were aware of the value of simulation-based distributions in the early 20th century (e.g., Rodgers & Beasley, 2013), but were constrained to small and simple statistical settings by computational limitations. By mid-century, Tukey and colleagues were refining and expanding the range of such methods (e.g., Rodgers, 1999). In 1979, Efron developed the bootstrap, which has become the most popular and powerful of the simulation methods to define sampling distributions. Since then, simulation methods have become useful in the distributional requirements of Bayesian statistical settings, through methods such as the Metropolis-Hastings algorithm and Gibbs sampling. This chapter is designed to provide theoretical background, conceptual understanding, and examples so that applied researchers can use this broad and valuable class of statistical methods.

One hundred years ago, a researcher interested in a theoretical distribution or characteristics of that distribution, such as its mean, standard deviation, or 2.5 and 97.5 percentiles, was restricted practically by computing limitations to the types of theoretical distributions that are described by an explicit equation,¹ such as the binomial or multivariate normal distribution. Using mathematical models of distributions often requires considerable mathematical ability and imposes severe and often intractable assumptions (e.g., normality, independence, variance assumptions, and so on). Computer simulations now provide more flexibility specifying distributions, which in turn provide more flexibility specifying models.

Many modern methods rely on simulation. One contemporary simulation technique is Markov chain Monte Carlo (MCMC) simulation, which can specify arbitrarily complex and nested multivariate distributions. It can even combine different theoretical families of variates. Another contemporary technique is the bootstrap, which can construct sampling distributions of conventional statistics that are free from most (but not all) assumptions. It can even create sampling distributions for new or exotic test statistics that the researcher created for a specific experiment.

¹Our present definition of *explicit equation* includes exact equations and well-defined series. An analytic solution relies only on explicit equations, although the definition’s boundaries are fuzzy.

<https://doi.org/10.1037/0000319-024>

APA Handbook of Research Methods in Psychology, Second Edition: Vol. 2. Research Designs: Quantitative, Qualitative, Neuropsychological, and Biological, H. Cooper (Editor-in-Chief)

Copyright © 2023 by the American Psychological Association. All rights reserved.

The field of simulation is a large one, and we try to cover only the aspects that have an immediate benefit for applied behavioral researchers. The field is very wide and extends into almost every area of statistics. It even extends beyond statistics; several influential techniques were developed by physicists in the 1940s and 1950s. The field has a history that itself is almost as long as modern statistics. Many of the founders of modern statistics conceptually described the benefits and justifications of simulations before they were pragmatically possible. The bootstrap and some useful simulation terminology are introduced in the chapter's first section. General simulations and MCMC simulations are covered in the second section.

R code for the chapter's examples is available at <https://github.com/OuhscBbmc/beasley-simulation-methods-2> and can be viewed with a simple text editor. The first example has two versions. The first listing is intended to be a clear and direct translation of the described steps; the second listing is optimized for efficiency and produces the graphs used in this chapter.

THE BOOTSTRAP

The bootstrap is a resampling technique that uses an observed sample to construct a statistic's sampling distribution. Many founders of modern statistics actively developed and promoted resampling, including William Gosset (also known as Student), R. A. Fisher, and John Tukey.

Bootstrapping Univariate Observations

The bootstrap is a flexible tool that can provide inferences in a complicated multivariate space, but the opening example is a simple collection of five scalars.

Example 1a: Standard error of the median.

A psychologist collects waiting times in a sample of $N = 5$ subjects to gain insight into the larger population of people.² She believes the population's distribution is likely skewed and decides the research question is best addressed by the median

and its variability. Unfortunately, the median does not have a closed-form equation for a standard error. One convenient solution is to use a bootstrap, which has five stages.

Stage 1. Collect the sample and calculate the observed median, MD_{Obs} , from the N scores.

Stage 2. Prepare the sampling frame, which can be thought of as a pool of scores. In this example, all five observed scores are placed in the sampling frame.

Stage 3. Draw $N=5$ scores *with replacement* from the sampling frame; this creates one *bootstrap sample*. Repeat this process many times, say $B = 9,999$.

Stage 4. The *bootstrap distribution* is formed by calculating the median of each bootstrap sample. Each bootstrapped statistic is denoted with an asterisk. The bootstrap distribution is the collection of B bootstrapped medians: $MD_1^*, MD_2^*, \dots, MD_{9999}^*$.

Stage 5. The standard error of the median is estimated by the standard deviation of the bootstrap distribution.

$$\overline{SE}_{MD} = \frac{1}{B-1} \sqrt{\sum_{b=1}^B (MD_b^* - MD_{\text{Obs}}^*)^2}, \quad (1)$$

where $MD_{\text{Obs}}^* = \frac{1}{B} \sum_{b=1}^B MD_b^*$.

Suppose the observed scores were 1, 4, 10, 50, and 80, and the summaries are $MD_{\text{Obs}} = 10$ and $\bar{X}_{\text{Obs}} = 29$. Table 24.1 illustrates possible simulation outcomes. In the first bootstrap sample, the values 4 and 50 were drawn twice, whereas 1 and 80 were never drawn. In the second-to-last sample, the five drawn scores were coincidentally the same as the observed sample. In the last sample, 4 was drawn almost every time.

In Stages 2 and 3, a *sampling frame* was formed and five scores were randomly drawn from it repeatedly. The goal was to mimic the median's variability that would occur if additional samples of $N = 5$ were drawn from the *population*. For many

²For a discussion of how to select a worthy research question, see Volume 1, Chapter 7, this handbook.

TABLE 24.1.

Illustration of Bootstrapped Scores and Statistics

Bootstrap index	Bootstrapped sample (Stage 3)	Bootstrapped statistic (Stage 4)
1	4, 4, 50, 10, 50	$MD_1^* = 10$
2	10, 80, 10, 50, 80	$MD_2^* = 50$
3	50, 4, 4, 1, 80	$MD_3^* = 4$
...		
9,998	1, 4, 10, 50, 80	$MD_{9998}^* = 10$
9,999	4, 4, 4, 4, 50	$MD_{9999}^* = 4$

types of bootstraps, the best sampling frame is simply the observed sample.

In Stage 4, a bootstrap distribution of medians was built to make an inference about the median of the population. Using a sample's statistic to estimate a population parameter follows the *plug-in principle*; the median is the *plug-in statistic* in this example (Efron & Tibshirani, 1993, Chapter 4).

A statistic's standard error quantifies the variability in its sampling distribution. Instead of calculating the spread of a *theoretical* sampling distribution (closed-form mathematical solutions that exist for statistics such as \bar{X} , r , and t , but not for MD), we calculate the spread in an *empirical* sampling distribution in Stage 5.

Example 1b: Standard error of the mean. The researcher later reused the collected sample to address a different question—one that is better suited by the mean. The algorithm proceeds as in Example 1a, except the plug-in statistic is now the mean instead of the median.

Stage 1. Collect the sample and calculate the observed mean, \bar{X}_{Obs} , from the N scores.

Stage 2. Prepare the sampling frame, which is the five observed scores in this example.

Stage 3. Draw N scores with replacement from the sampling frame; this creates one bootstrap sample. Repeat this process many times, say $B = 9,999$.

Stage 4. A bootstrap distribution is formed by calculating the mean of each of the B bootstrap samples. The bootstrap distribution is the B bootstrapped means: $\langle eq \rangle \bar{X}_1^*, \bar{X}_2^*, \dots, \bar{X}_{9999}^*$.

Stage 5. The standard error of the mean is estimated by the standard deviation of the bootstrap distribution.

$$\overline{se_{\bar{x}}} = \frac{1}{B-1} \sqrt{\sum_{b=1}^B (\bar{X}_b^* - \bar{X}_{Obs}^*)^2}, \quad (2)$$

$$\text{where } \bar{X}_{Obs}^* = \frac{1}{B} \sum_{b=1}^B \bar{X}_b^*.$$

The bootstrap samples from Example 1a can be reused to calculate the bootstrapped means.³ The last column in Table 24.1 would be replaced with the values $\bar{X}_b^* = 23.6, 46, 27.8, \dots, 29, 13.2$. Stage 5 then calculates the standard deviation of these 9,999 statistics (the reason for choosing $B = 9,999$ is discussed briefly in the section Bootstrap Sample Size).

There are many types of bootstraps, and the two just described are *nonparametric* in the sense that they require no assumptions about the sampling distributions (however, they do assume that the observed scores are drawn independently from the population of interest). A procedure is *parametric* when it relies on assumptions about the population distribution. The typical parametric standard error of the mean is:

$$\text{Parametric } \overline{se_{\bar{x}}} = \frac{1}{\sqrt{N}} \sqrt{\frac{\sum_{i=1}^N (X_i - \bar{X})^2}{(N-1)}} = \frac{s}{\sqrt{N}}. \quad (3)$$

The conventional standard error of the mean measures the variability in a sample (i.e., the standard deviation, s) to estimate the variability in the population of means.⁴ It uses the central limit theorem to relate s to the $\overline{se_{\bar{x}}}$. Unfortunately, many useful statistics do not have a convenient theoretical relationship such as this. For the statistics that do, the required assumptions can

³We want to emphasize that this process is unaffected by the choice of plug-in statistic.

⁴When a large sample is drawn from a normally distributed population, the bootstrap standard error will be very close to the conventional standard error of the mean.

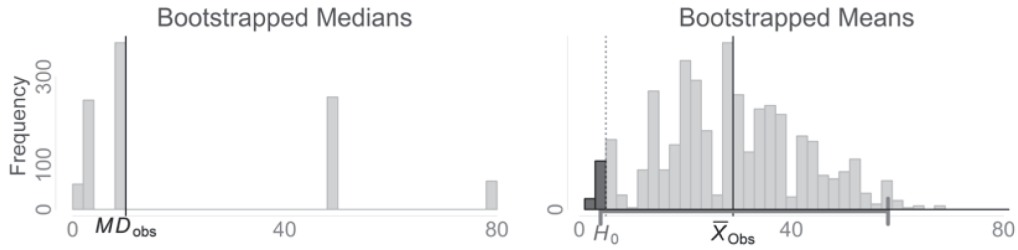


FIGURE 24.1. Bootstrap distributions. The right panel includes the bootstrap CI and p value (dark gray area).

be unreasonable in some applied scenarios. The bootstrap can help in both cases: calculating the standard error is simple even for complicated plug-in statistics. The choice of the plug-in statistic is very flexible, and this will be discussed later.

Example 1c: Confidence interval for the mean.

A 95% confidence interval (CI) for the mean⁵ can be estimated from the bootstrap distribution created in Stage 4. The bootstrap samples and bootstrap distribution can be reused. Only the final stage is different.

Stages 1 to 4.

Proceed as in Example 1b.

Stage 5. Order the $B = 9,999$ bootstrapped statistics from smallest to largest.

The CI bounds are marked by the 250th smallest value and the 250th largest value (i.e., the .025 and .975 quantiles). The number of scores in each tail is calculated by $\alpha(B + 1)/2$; α is .05 with a 95% CI.

A CI determined from this type of bootstrap distribution has an additional advantage over a CI determined from a parametric, theoretical normal distribution. The parametric distribution relies on the central limit theorem for normality, and thus the tails are an equal distance from \bar{X} ; the CI is defined by $\bar{X} \pm 1.96 \times \overline{se}_{\bar{X}}$. The parametric procedure can be justified as N grows infinitely large, but it can be misleading when a small sample is drawn from a skewed distribution. In fact, the parametric CI in this example is $(-1.4, 59.4)$,

which produces a nonsensical negative value for waiting time.

This bootstrap CI method has the appealing feature that it is *range-preserving*; in this case, the CI for waiting time will never be negative. The bootstrap CI is $(4.0, 58.8)$; its boundaries are guaranteed to be values that could be observed in a sample (because they were calculated from values that were actually observed in a sample; Efron & Tibshirani, 1993, Section 13.7). The bootstrap distribution is shown in Figure 24.1, along with the CI.

Example 1d: The p value for the mean. A one-tailed p value is determined in an intuitive way, as the proportion of bootstrapped statistics that are more extreme than the value of the null hypothesis. A two-tailed p value is easy to determine as well but would not make theoretical sense with the waiting time example. If $H_0: \text{time} \leq 5$, the five stages are:

Stages 1 to 4. Proceed as in Example 1b.

Stage 5. Tally the number of \bar{X}_b^* values equal to or less than the hypothesized value, expressed as:

$$\#\{\bar{X}_b^* \leq \text{time}_{\text{Null}}\}. \quad (4)$$

The p value is $(1 + \#\{X_b^* \leq \text{time}_{\text{Null}}\})/(B + 1)$.

Notice that the choice of plug-in statistic in Stage 2 is unrelated to the choice of statistic that summarizes the bootstrap distribution in Stage 5. A standard deviation can be calculated on the B statistics regardless of plug-in equation used in

⁵A frequentist 95% CI is built so that 95% of similarly constructed CIs will contain the population parameter value.

Stage 2 (e.g., the median or mean). Similarly in Stage 5, the distribution of B means can be summarized in a variety of ways (e.g., standard error, CI, or p).

The code accompanying the chapter replicates the steps in our examples, including plotting simplified versions of the figures. These examples are intended to supplement the knowledge of novice bootstrappers (with limited exposure to R) and to provide a template for more complicated bootstraps that can arise in applied research. Software is further discussed at the end of the chapter.

Terminology. Before we move to slightly more complicated examples, we summarize the entities and notation. Typically, a researcher draws a sample X to gain insight into its population distribution of single scores, F (this F is unrelated to the analysis of variance [ANOVA] F distribution). If we are interested in the mean of the population, μ , the appropriate plug-in statistic is the mean of the sample, \bar{X} . An inferential procedure mimics F with a theoretical distribution called \hat{F} , to assess the accuracy of \bar{X} (or any other plug-in statistic). Conceptually, \hat{F} stands in for F because we don't know F , but we do know \hat{F} . In the world of resampling, \hat{F} is more specifically called an empirical distribution. Examples 1b to 1d calculate three common expressions of the uncertainty in the estimate of μ : the standard error, CI, and p value.

The empirical distribution, \hat{F} , should not be confused with the bootstrap distribution, which is a type of empirical *sampling* distribution. For instance, in Example 1a, \hat{F} is a distribution of N single *observations*, whereas the bootstrap distribution is a collection of B *statistics* (that were each calculated from a bootstrap sample of N scores randomly drawn from the sampling frame). The distinction between these different types of distributions is explained in detail in Rodgers (1999).

The sampling frame is the mechanism behind \hat{F} , because it is the pool of single points from which the bootstrap samples are drawn. The previous examples have used a sampling frame that was

built directly from the observed sample. We will show three other types of bootstraps that are only indirect expressions of the sample. In the second half of the chapter, we discuss Monte Carlo methods, which are simulations in which \hat{F} is entirely unconnected to an observed sample.

So far, the sampling frames produced empirical distributions that represent an observed population. We start using the notation \hat{F}_{Obs} to distinguish it from an empirical distribution representing a null hypothesis, \hat{F}_{Null} . Examples 2a and 2b focus on this difference.

Bootstrapping with Novel Designs

The mean is a well-known statistic with an accessible theoretical sampling distribution—yet the bootstrap can help when the central limit theorem assumptions are not justifiable. The median is well known, but it does not have a good theoretical sampling distribution; the bootstrap can help by providing an accessible empirical sampling distribution.

In some scenarios, an established sampling distribution exists but does not fit the profile of an experimental design. For instance, the longitudinal, nested factorial design of Smith and Kimball (2010, Experiment 1) benefited from the flexibility of a bootstrap in two ways. First, a subject's final outcome was conditioned on their initial response in a way that prevented the ANOVA sampling distribution from representing it appropriately. This linking created correlated error terms for the linked observations, which invalidated the traditional ANOVA distribution as an appropriate sampling distribution (and we know that the ANOVA is not robust to violations of independence of errors). Second, there was substantial heterogeneity in the variability, making it difficult to model appropriately. After the sampling frame was customized to fit the researchers' specific contrasts, a bootstrap was able to test hypotheses with $N = 110$ subjects that a parametric generalized linear model or multilevel model could not.

The bootstrap's flexibility perhaps is demonstrated best when it provides a sampling distribution for a *new statistic* that is created for

a specific design protocol. In fact, “subject to mild conditions,” the selected bootstrapped statistic “can be the output of an algorithm of almost arbitrary complexity, shattering the naïve notion that a parameter is a Greek letter appearing in a probability distribution and showing the possibilities for uncertainty analysis for the complex procedures now in daily use, but at the frontiers of the imagination a quarter of a century ago.” (Davison et al., 2003, p. 142).

It is difficult to give concise examples of this flexibility, because several paragraphs would be needed just to describe a novel design; advice and examples are found in Boos (2003) and Davison and Hinkley (1997).

To provide an approximation, and to stimulate the reader to think deeper about such a constructed statistic, consider the following setting. Tukey's (1977) H-spread was designed to measure the distance across the middle half of a distribution (often referred to as the interquartile range). Suppose a theory implies interest in another distance: the distance across the middle 20% of the distribution (a range-type measure even less influenced by extreme scores than the H-spread). This statistic is sensible and interesting, but in this case, the statistical community has no background or statistical theory to help the applied researcher. But the bootstrap is every bit as facile and useful in this previously undefined setting as it is in applications involving other well-known statistics like the mean, median, or H-spread.

Bootstrapping Multivariate Observations

When two scores are collected from a subject, our definition of an observation is expanded to a bivariate point, $u_i = (x_i, y_i)$.

Example 2a: \hat{F}_{Obs} for a correlation. Diaconis and Efron (1983) bootstrapped a correlation by using the observed sample as the sampling frame. In Example 1, N univariate points were drawn from a sampling frame of N univariate points. Here, N bivariate points are drawn from a sampling frame of N bivariate points.

Stage 1. Collect the sample and calculate r_{obs} from the N data points (pairs of X , Y values).

Stage 2. Prepare the sampling frame. To produce \hat{F}_{Obs} in this case, use the observed sample.

Stage 3. Randomly draw N pairs of scores with replacement while keeping the pairs intact. For instance, if x_3 is selected, the accompanying value must be y_3 (i.e., the x and y scores for the third subject). Repeat this stage to form B bootstrap samples.

Stage 4. Calculate r_{Obs}^* for each bootstrap sample drawn in stage 3.

Stage 5. Calculate the CI [$r_{(250)}^*$, $r_{(9750)}^*$] with $B = 9,999$. If a hypothesis test is desired, the null hypothesis can be rejected if ρ_{null} falls outside of the CI. As before, the standard error is the standard deviation of the B statistics in the bootstrap distribution.

Univariate Sampling Bootstrap

Example 2b: \hat{F}_{Null} for a correlation. As early as 1935, Fisher (1970) developed a resampling method, called the *permutation test* or the *randomization test*. It is very similar to the bootstrap, except that it samples from the sampling frame *without replacement*.⁶ Fisher did not intend to estimate the standard error but rather to calculate the p value of a null hypothesis, which is achieved by constructing a sampling frame that represents the null hypothesis.

In the case of a bivariate correlation, suppose the null hypothesis states that X and Y are linearly independent in the population. An interesting special case of linear independence (Rodgers et al., 1984) that is often tested is $\rho_{\text{Null}} = 0$. One approach is to conceptualize this as “every value of X has an equal chance of being associated with any value of Y .” To reflect \hat{F}_{Null} , the sampling frame enumerates all possible X and Y pairs—creating a sampling frame with N^2 bivariate points (see Lee & Rodgers, 1998). Figure 24.2 portrays the two different sampling approaches.

This procedure for bootstrapping \hat{F}_{Null} resembles Example 2a, with three exceptions. First, the sampling frame has N^2 points instead of N . Second,

⁶“[The bootstrap] was designed to extend the virtues of permutation testing” (Efron & Tibshirani, 1993, p. 218).

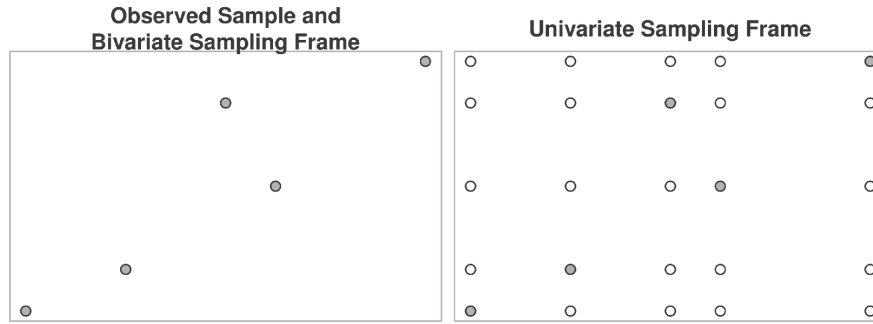


FIGURE 24.2. Scatter plots of a bivariate sampling frame based on \hat{F}_{Obs} (left) and a univariate sampling frame based on \hat{F}_{Null} (right).

each of these points has a $1/N^2$ probability of being selected on each draw, instead of $1/N$. Finally, a hypothesis is tested by comparing r_{Obs} with the CI, instead of comparing ρ_{Null} with the CI.

Stage 1. Collect the sample and calculate r_{Obs} from the N data points (pairs of X , Y values).

Stage 2. Prepare the univariate sampling frame by combining every x with every y value.

Stage 3. Randomly draw N pairs of scores with replacement from the N^2 possible points in the sampling frame. Repeat this stage to form $B = 9,999$ bootstrap samples.

Stage 4. Calculate r_{Obs}^* for each bootstrap sample drawn in Stage 3.

Stage 5. Calculate the $\text{CI}[r_{(250)}^*, r_{(9750)}^*]$. If a hypothesis test is desired, the null hypothesis can be rejected if r_{Obs} falls outside of the region of nonrejection defined by the CI. The standard error is again the standard deviation of the bootstrap distribution.

This CI (derived from \hat{F}_{Null}) represents the variability around ρ_{Null} , whereas the previous CI (derived from \hat{F}_{Obs}) represents the variability around r_{Obs} . The two contrasting p -value equations

for $H_0: \rho > \rho_{\text{Null}}$ are $p_{\hat{F}_{\text{Obs}}} = \frac{1 + \#\{r_b^* < \rho_{\text{Null}}\}}{B + 1}$ and

$p_{\hat{F}_{\text{Null}}} = \frac{1 + \#\{r_b^* < r_{\text{Obs}}\}}{B + 1}$. Notice that the value of

ρ_{Null} is not present in the latter p -value equation because it is reflected within the sampling frame, which is constrained by its construction to have a correlation of zero.

The univariate sampling bootstrap is easily extended to cases where correlations are non-zero. Using a method called “diagonalization” the sampling frame produced by the univariate sampling framework can be shaped to exhibit any correlation a researcher might require. This is particularly useful in two ways. The first is to test non-nil null hypotheses. In some cases, simply rejecting the null hypothesis of no correlation may not be of interest. In order to use the univariate sampling bootstrap for this test of the non-nil null, the researcher can set the correlation in the sampling frame to the null hypothesized correlation coefficient. The bootstrap procedure then proceeds as before. This creates a null distribution *around the non-nil null*.

The second case where diagonalization of the univariate sampling frame is useful is when a confidence interval is desired. By diagonalizing the univariate sampling frame so that it has the same correlation as the originally observed data, and then bootstrapping as before, the resulting confidence interval is the confidence interval of the observed statistic.

While the alternative step of using the univariate sampling frame, instead of the raw data, may seem to be trivial, this alteration to the bootstrap has repeatedly shown itself to provide advantages over the traditional bootstrap, particularly with regards to Type I error rates. Research on correlation coefficients has demonstrated this in a number of settings (e.g., Beasley et al., 2007; Bishara & Hittner, 2012, 2017). Research for uses other than correlation coefficients has also

recently found an advantage for the univariate sampling bootstrap over other bootstrap alternatives and some nonbootstrap CI methods (O'Keefe & Rodgers, 2020). Software implementations of the univariate bootstrap are available in R (e.g., Omisc) that manage the creation of the univariate sampling frame, diagonalization if desired, and the bootstrapping procedure itself.

Hutson (2019) defined an almost identical approach to the univariate sampling bootstrap, and named it the *surrogate bootstrap*. (The only difference is his focus on defining the p-value, instead of the confidence interval; if the confidence interval is used for hypothesis testing, the outcome must always be identical. Hutson noted in relation to the Lee and Rodgers, 1998, procedure that “the test can also be inverted to provide precise confidence interval for ρ ”). Hutson provided mathematical justification for the univariate sampling bootstrap, along with additional simulation support for the excellent operating characteristics of this bootstrap method.

Example 3a: Parametric bootstrap. The *parametric bootstrap* is similar to the nonparametric bootstrap in previous examples, except that \hat{F}_{Obs} and its sampling frame have distributional assumptions. In a correlational setting, an analyst might be able to assume the variables approximately follow a bivariate normal distribution with a linear relationship of r_{Obs} (Efron & Tibshirani, 1993, Section 6.5). In this case, scores in the sampling frame do not contain any observed scores. The sample influences the sampling frame only through r_{Obs} . For a given bootstrap sample, the N bivariate points are generated as follows:

Stage 1. Collect the sample and calculate r_{Obs} from the N data points (pairs of X , Y values).
Stage 2. State the parametric form of the estimated population. A linear, normal distribution is:

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim N \left(\begin{pmatrix} \bar{X} \\ \bar{Y} \end{pmatrix}, \begin{pmatrix} \sigma_X^2 & \sigma_{XY} \\ \sigma_{XY} & \sigma_Y^2 \end{pmatrix} \right). \quad (5)$$

Stage 3. Randomly draw N bivariate points. The random number generator produces a

unique point every draw. Repeat to form B bootstrap samples.

Stage 4. Calculate r_{Obs}^* for each bootstrap sample drawn in stage 3.

Stage 5. If desired, calculate the CI and p value as in Example 2a (and not like Example 2b).

Although r_{Obs}^* is now parametric, the bootstrap distribution itself is still considered nonparametric. The shape of the collection of r_{Obs}^* values has no equation or restrictions. The parametric bootstrap can be a good tool when the population's characteristics can be reasonably assumed, but the statistic's characteristics are not well known. This situation occurs with statistics like the median (that lack a closed-form sampling distribution) or for novel statistics that are tailored to a specific experimental protocol (e.g., Boos, 2003).

Example 3b: Semiparametric bootstrap. A *semiparametric bootstrap* draws observations from an \hat{F} that is constructed from some parametric and some nonparametric assumptions. In a multiple regression setting, one could assume F has a linear relationship and the residuals are exchangeable but not assume the residuals are normally distributed. In this model, the i th subject's predicted score is $y_i = b_0 + b_1x_{1,i} + b_2x_{2,i} + e_i$, and e_i is their residual.

Stage 1. Collect the sample and calculate the sample coefficients (b_0, b_1, b_2) that estimate the population parameters ($\beta_0, \beta_1, \beta_2$).

Stage 2. The sampling frame is formed from the N residuals (e_1, \dots, e_N).

Stage 3a. Randomly draw N residuals with replacement ($e_1^*, e_2^*, \dots, e_N^*$).

Stage 3b. If the independent variables (the x values) are considered fixed, each bootstrap sample is:

$$\begin{aligned} y_1^* &= b_0 + b_1x_{1,1} + b_2x_{2,1} + e_1^* = \hat{y}_1 + e_1^* \\ y_2^* &= b_0 + b_1x_{1,2} + b_2x_{2,2} + e_2^* = \hat{y}_2 + e_2^* \\ &\dots \\ y_N^* &= b_0 + b_1x_{1,N} + b_2x_{2,N} + e_N^* = \hat{y}_N + e_N^*. \end{aligned} \quad (6)$$

This creates a bootstrap sample of N values: $(y_1^*, y_2^*, \dots, y_N^*)$. Repeat this stage to form B bootstrap samples.

Stage 4. Calculate b_0^* , b_1^* , and b_2^* with the same three-parameter linear model for each bootstrap sample created in Stage 3.

Stage 5. Calculate the desired statistics (similar to Example 2a) on the trivariate bootstrap distribution of (b_0^*, b_1^*, b_2^*) .

The x values are considered fixed in this specific example, so they are not drawn randomly in Stage 3b. Bootstrap distributions of other plug-in statistics such as R^2 may better address the specific research question (e.g., Manly, 2007, Chapter 7). The linear model does not necessarily have to minimize squared error (e.g., it could minimize the median of absolute values of deviations). Semiparametric bootstraps can provide a foundation for many generalized linear models (Davison & Hinkley, 1997, Section 7.2) and exploratory approaches, such as loess curves and cubic splines (Hastie et al., 2009).

If additional assumptions are justifiable, a semiparametric bootstrap can model dependencies more naturally than a nonparametric bootstrap. Drawing residuals as if they were interchangeable requires the assumption of homogenous variance (drawing observed samples, as described in Examples 2a and 2b, does not). Adjustments such as standardizing the residuals may improve the robustness of semi-parametric approaches (for this and other techniques, see Davison & Hinkley, 1997, Sections 3.3, 6.2–6.3).

Bootstrapping data with dependencies. Bootstrapping is reasonably straightforward when the data are independently and identically distributed. However, psychological designs frequently model dependency among the observations (e.g., time series), variables (e.g., multiple regression, repeated measures designs), or sampling levels (e.g., multi-level models). Sometimes a nonparametric bootstrap may be unable to accommodate these designs because it is difficult to incorporate the appropriate dependency into the sampling frame and also avoid distributional assumptions; instead, parametric

and semiparametric bootstraps can be used. For more strategies and applications, see Davison and Hinkley (1997) and Beasley and Rodgers (2009). Lahiri (2003) is a mathematically oriented book dedicated to dependent data.

Pragmatic Bootstrapping Issues

A statistical analysis can accommodate both bootstrap and parametric procedures. A researcher may believe a χ^2 distribution is appropriate for the fit statistic for testing a structural equation model (SEM), while also believing the CIs around the means and covariances are asymmetric. In this case, a parametric fit statistic can be complemented by bootstrapped standard errors. If a parametric distribution is problematic, the Bollen-Stine (Bollen & Stine, 1992) bootstrap distribution could be used instead (Enders, 2010, Sections 5.11 and 5.15). Another illustration of a heterogeneous strategy is using parametric standard error of the mean and a bootstrapped H-spread. In short, adopting the bootstrap can be a gradual transition.

Confidence interval adjustments. The CI calculated in Example 2 is commonly called the percentile CI. Its simple definition is that the percentile of the bootstrap distribution maps directly to the percentile of the inferred population. For instance, the 250th smallest r^* (out of $B = 9,999$) estimates the population's 2.5% percentile, assuming the null hypothesis is true. However, this effortless relationship can produce biased estimates in common conditions and several CI adjustments have been developed to have less bias and greater efficiency.

At least eight CI adjustments have been proposed (many authors frequently use ambiguous or conflicting names; surveyed in Beasley & Rodgers, 2009, pp. 372–375). We prefer the BCa (which stands for “bias-corrected and accelerated”) adjustment because it has a favorable combination of efficiency, robustness, and wide applicability. It attempts to correct for bias in the bootstrap distribution and for heterogeneous variability in the plug-in statistic (Efron & Tibshirani, 1993, Chapter 14).

Bootstrap sample size. Nonparametric bootstraps are randomly drawn from the empirical sampling frame because complete enumeration of all possible bootstrap samples is rarely practical.⁷ This randomness introduces simulation error (which can be thought of a type of sampling error from \hat{F}) and fortunately increasing B to a reasonable number makes this error negligible. All the chapter's bootstrap examples complete in less than 5 seconds, even when $N = 500$.

We recommend that at least 10^3 and 10^4 replications be run for standard errors and 95% CIs, respectively. Additional discussion and references are found in Beasley and Rodgers (2009, pp. 378–379), but reading this takes longer than completing $B = 99,999$. It may seem strange that our suggested B values have been chosen so that $(B + 1)\alpha$ is an integer (e.g., 9,999 instead of the more natural 10,000). Boos (2003) explained the “99 Rule” and how it slightly improves CI accuracy.

Additional bootstrap applications. Most psychological research questions and designs are more complex than the chapter's examples, but the principles remain the same. Examples and references to sophisticated designs and plug-in statistics are found in Beasley and Rodgers (2009, pp. 375–378). These include designs like time series, stratified samples, circular variables, and models like generalized linear models, multilevel linear models, survival models, Bayesian analysis, mediation models, and SEM. The resampling procedures that influenced the development of the bootstrap are also discussed, including the permutation test and jackknife (also see Rodgers, 1999).

Limitations. Two commonly encountered limitations of parametric procedures that apply to the bootstrap and are worth stating here. First, inferences can be misleading when dependencies in the data are not appropriately modeled. Second,

a flawed sampling process can produce problematic inferences (although the bootstrap may be less susceptible to this problem than traditional parametric procedures).⁸

The bootstrap does have problems if the plug-in statistic estimates a boundary, or a value close to a boundary, such as a minimum reaction time (Andrews, 2000). In this case, the estimate will be biased upward because the bootstrapped statistic of reaction time cannot be negative. Notice that it is acceptable to estimate a quantity near the boundary of a bootstrap distribution (such as the 2.5th percentile in Stage 5) but not near the boundary of the population distribution (Stage 4). Andrews (2000, Section 2) and LePage and Billiard (1992) discussed other potential concerns that are less likely to affect psychologists.

Beran (2003) wrote,

Success of the bootstrap, in the sense of doing what is expected under a probability model for data, is not universal. Modifications to Efron's (1979) definition of the bootstrap are needed to make the idea work for estimators that are not classically regular. (p. 176)

When a novel plug-in statistic is developed (either bootstrap or parametric), good inferential performance is not assured. We advise that the new statistic be studied with a small simulation to assess if it has acceptable Type I error and adequate power, for the observed N . This proactive analysis (Steiger, 2007) should include several likely population values and nonnormal distributions. Many of the same tools and skills used to bootstrap can be applied to the proactive analysis.

We occasionally are asked whether the validity of bootstrap inferences suffers with small sample sizes. We feel that if one or more outliers, or even an unrepresentative sample caused by natural

⁷In Example 1, a small-data example, complete enumeration requires $5^5 = 3,125$ bootstrap samples, which actually requires less work than the suggested $B = 9,999$. However, this is rarely the case, because the sample size is usually larger than $N = 5$; if one more score had been collected, complete enumeration requires $B = 6^6 = 46,656$. Even a moderate size of $N = 30$ requires $B \approx 10^{14}$. This number can be reduced by accounting for and reweighting redundant samples (e.g., the sample $\{11, 11, 4\}$ produces the same statistic as $\{4, 11, 11\}$), but programming these shortcuts would take much longer than running a large B , and the sample still may not be small enough to be practical.

⁸With respect to the correlation, the bootstrap outperformed parametric procedures in simulations of restricted range (Chan & Chan, 2004; Mendoza et al., 1991), nonnormal correlated populations (Beasley et al., 2007), and composite populations (Lee & Rodgers, 1998).

sampling variability, can mislead a bootstrap distribution, then it is likely to be even more disruptive to a parametric sampling distribution. For instance, parametric inferences were more susceptible than bootstrap inferences when a bivariate correlation was calculated from a sample of five observations (Beasley et al., 2007). With a multivariate normal population, the procedures had comparable Type I error, whereas the parametric had slightly better power than the bootstrap. However, when the assumptions were violated by using skewed populations, the parametric procedure had liberal Type I error (reaching 0.15), whereas the bootstrap did not. Summarizing across all simulated values of N , the parametric procedure benefited when its assumptions were met, but could be unreliable when they were not. Of course, it is irresponsible to claim this pattern will hold for all statistics and population distributions, which is another reason to perform a proactive analysis before using a novel plug-in statistic.

Software. Software for parametric procedures is much more available and user-friendly than for the equivalent bootstraps. The flexibility that empowers the bootstrap also prevents automation. Twenty years later, Fan's (2003) assessment of available bootstrapping software still applies. When bootstrapping a statistic, it is likely that writing code will be necessary.

R has the most complete support for two reasons. First, it has many concise routines useful to bootstrapping. For instance, the line ``sample(x=obs, size=15, replace=TRUE)`` randomly draws $N = 15$ scores from a vector called ``obs``. Second, most developments and publications involving applied bootstrapping have come from statisticians (and especially biostatisticians) who publish their examples in this language. Examples and documentation

also can be found in Stata and SAS.⁹ The SEM programs EQS and Mplus provide bootstrapping for better fit statistics and for more robust CIs (Enders, 2010, Table 11.1).

Two of the most popular bootstrap books use R and S-PLUS exclusively (Davison & Hinkley, 1997; Efron & Tibshirani, 1993).¹⁰ They accommodate some common designs with less than 10 lines of code from the practitioner. The user defines their specific plug-in statistic, and then passes this definition to a reusable base routine provided by the package.

It can be tricky to define this specialized function, however, even for common analyses such as those that (a) incorporate multiple groups, (b) draw from \hat{F}_{Null} , or (c) use sampling frames that do not have exactly N points. If the base routine has trouble accommodating the plug-in function, we suggest that users create their own routine by starting with the code for a routine (like `bcanon`) in the bootstrap package and modifying it to fit the current design.¹¹

The defined plug-in statistic needs to detect and react to atypical samples. In Example 2a, it is likely that one of the 9,999 bootstrap samples will have no variation, so that r_{Obs}^* is undefined. If unanticipated, this will either halt the program's execution or insert an undefined value into the bootstrap distribution (depending on the statistical software).

If the software supports a "Try-Catch" block, it can be used to recover from this event. One implementation of Example 2a catches the undefined statistic and forces another bootstrap sample to be drawn and calculated. Another implementation simply replaces the undefined values with zeros (which is much faster than having the computer construct a Try-Catch block). Even if this behavior is not ideal theoretically, it will happen too infrequently to have any noticeable effect.¹² If the software language does

⁹Good starting points are <http://www.stata.com/help.cgi?bootstrap>, Poi (2004), and <http://support.sas.com/kb/24/982.html>.

¹⁰Their routines are included in the "bootstrap" and "boot" packages. After loading the package, documentation appears after typing "?bootstrap" or "?boot". Both packages have good help files, with "boot" being slightly more thorough. Packages are discussed in "An Introduction to R," which is available on the help menu of R.

¹¹In R, a routine's underlying code is presented when its name is entered by itself (e.g., "bcanon" when Efron & Tibshirani's [1993] "bootstrap" package has been installed and loaded). Saving the code in a script allows it to be modified, executed, and saved.

¹²When $N = 5$ in Example 2a, roughly $5-4 = 0.16\%$ of bootstrap statistics will be undefined. When $N = 10$, this proportion drops to 10–9. We believe this source of error is overwhelmed by sampling error and can be ignored.

not provide error handling (and zero is not an appropriate substitute value for the statistic), the custom code should anticipate and test for illegal conditions.

Despite the additional issues to consider, bootstrapping can be valuable to a practitioner when it holds a statistical advantage. The bootstrap is a good candidate when the desired statistic lacks a closed-form standard error equation, when necessary parametric assumptions are not met, or especially when small sample sizes are combined with the previous restrictions.

BROADER SIMULATION METHODS

When simulation uses repeated random sampling to build a distribution, it is frequently called a *Monte Carlo method*. The bootstrap is a specific type of Monte Carlo simulation. It can create a distribution of statistics that lacks an equation for the probability density function (pdf) and the cumulative distribution function (CDF; i.e., the integral of the pdf). Thus, in the bootstrap, a collection of B points are simulated and substituted for the desired pdf or CDF.

In most Monte Carlo simulations, the distribution of the relevant statistic(s) has a tractable pdf but an intractable CDF. In other words, equations are available to calculate the probability for a single parameter value (e.g., $p(\theta = 2)$) but not for a range of parameter values (e.g., $p(0 \leq \theta \leq 2)$ or $p(\theta \leq 1.7)$); the standard error and other moments typically are not available either. Like the bootstrap, the general Monte Carlo method builds a collection of B points as a substitute for the desired distribution. Simulation literature commonly calls this the *target distribution*, f .¹³

The following simulation techniques are general and can evaluate many types of distributions, although we will discuss them in the context of the posterior distribution. A Bayesian posterior distribution is proportional to the product of the

prior and likelihood distributions (as explained in Chapter 26 of this volume). Many posterior distributions have an equation for the pdf, but not for the CDF or standard error, and so simulation methods are an attractive tool.

Before the 1990s, most Bayesian analysts had to choose their prior and likelihood distributions carefully, so that the posterior's CDF had a closed-form equation.¹⁴ This was not a weakness of Bayesian theory but rather a limitation of available Bayesian methods. This restriction was a common inconvenience for single-parameter models, but it made the use of many multiparameter models completely intractable (especially when the posterior distribution included parameters from different families of distributions). With the development of simulation, Bayesian methods are now arguably more flexible than frequentist (i.e., standard parametric) methods.

General Simulation

Simulation is unnecessary when the posterior describes a small number of parameters. A distribution can be systematically partitioned into small areas, which are calculated separately before being recombined. This deterministic technique, called *numerical integration*, can be a rectangular approximation used to estimate the area under a curve and is taught to all calculus students before the more elegant *analytical integration*. Analytical integration is not possible with most posterior distributions used in research, however, and even numerical integration is not practical when the posterior has many parameters. A target distribution has one dimension for every parameter; it is common for f to have too many dimensions to integrate deterministically.

When analytical and numerical integration are not feasible, simulation can be the next best method. As Monahan (2001) said, "Monte Carlo should be viewed as just another way to compute an integral; numerical integration should be

¹³The target distribution, f , should not be confused with the bootstrap literature's F (or \hat{F}). F is the theoretical population distribution of single observations, whereas f is the desired distribution of statistics. If the simulation notation were applied to the bootstrap, f would be the bootstrap distribution.

¹⁴One common conjugate relationship is a Gaussian prior and a Gaussian likelihood, resulting in a Gaussian posterior. Another common relationship is a beta prior and a binomial likelihood, resulting in a beta posterior.

viewed as just another way to sample points in a space” (p. 235). Although our simple simulation examples include only one or two parameters, simulation’s real benefit is evident in high-dimensional problems.

Example 4a: Rejection sampling with bounded support. *Rejection sampling* is a simple simulation technique in which points are generated and then accepted or rejected into the final collection of points (it is sometimes called *acceptance–rejection sampling*). To focus on rejection sampling, we will assume that the sample has been collected, and the prior and likelihood distributions have been defined so that the posterior’s pdf can be found. Thus, the posterior pdf is the target distribution, f .

Suppose the researcher has found f for a parameter, θ , that ranges between -0.5 and $+0.5$. The height of f (the bimodal solid line in Figure 24.3, left panel) can be found directly, but not the area underneath it (say from $\theta = 0$ to $\theta = 0.2$). To find this area and other quantities, five stages are needed:

Stage 1. Specify and graph f (the curved solid line in Figure 24.3, left panel).

Stage 2a. Determine the *candidate bounds*, represented by the endpoints of the horizontal axis in Figure 24.3, left panel. It should cover the minimum and maximum values of the target parameter (which is $[-0.5, 0.5]$).

Stage 2b. Determine the *density bounds*, $[0, c]$. It should start at zero and extend slightly

beyond the tallest point f . The height is called the *scaling constant*, c .

Stage 2c. Plot the box for the candidates and the densities. Stage 2a determines the horizontal coordinates of the box, and Stage 2b determines the vertical coordinates. It should completely envelope f .

Stage 3a. Draw a random uniformly distributed variate, x_b , from the candidate bounds $[-0.5, 0.5]$ (i.e., the width of the dashed box). Repeat this B times to generate $x_1, x_2, \dots, x_b, \dots, x_B$.

Stage 3b. For every candidate, draw a uniformly distributed variate, y_b , from the density bounds $[0, c]$ (i.e., the height of the dashed box).

Stage 4. For every candidate, find the corresponding height of the target pdf, $f(x_b)$. Accept the candidate if $f(x_b) \geq y_b$. Accepted candidates are stored in a collection of target points. Plot each (x_b, y_b) point; in Figure 24.3, left panel, an accepted point is a dark gray circle, whereas a rejection is a light gray x .

Stage 5. Calculate the summary statistics of the distribution. Like the bootstrap, the inferences are estimated by calculating statistics on the distribution of accepted candidates. For instance, the estimated mean of the posterior is simply the mean of the accepted candidates. Similarly, the 95% Bayesian CI is marked by the .025 and .975 quantiles of the accepted candidates.

After the candidate and density bounds are established in Stage 2, a pair of random numbers

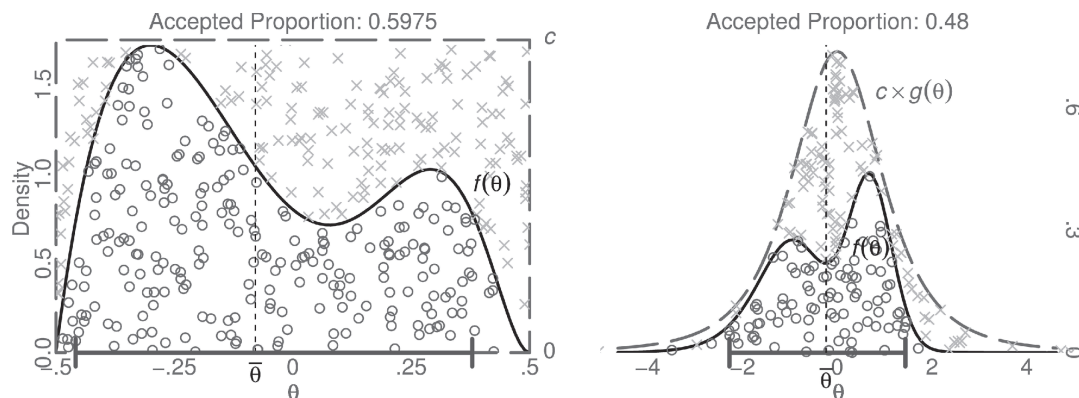


FIGURE 24.3. Rejection sampling of a bounded parameter (a) and unbounded parameter (b). The target distribution is solid, whereas the candidate distribution is dashed. A dark gray circle indicates an accepted candidate, whereas a light gray x is a rejected candidate.

is drawn for every candidate in Stage 3. The first variate is a parameter value (i.e., the point's horizontal position). The second variate is a density value (i.e., the vertical position). It is important that these variates can cover the range of both dimensions.

The target distribution is taller at $\theta = -.3$ than at $\theta = .1$, indicating that $-.3$ is more likely. Therefore in Stage 4, we want more of the accepted candidates to be in the neighborhood of $-.3$ than in the neighborhood of $.1$. The height of f at $\theta = -.3$ is roughly 1.7 and c (the height of the dashed box) is 1.75. As a result, a candidate of $\theta = -.3$ has a 97% ($= 1.7/1.75$) chance of being accepted. For comparison, candidates in the neighborhood of $\theta = .1$ will be accepted 41% ($= .72/1.75$) of the time. When enough candidates are evaluated, the collection of accepted candidates will have more than twice as many values near $-.3$ than near $.1$. This allows the summary statistics calculated in Stage 5 to assess the properties of the posterior distribution.

The example's f was defined to be a proper probability distribution¹⁵ (i.e., the total area under the curve, its integral, equals 1), which allows us to verify that the proportion of accepted candidates is approximately correct. The area of the box is 1.75 ($= (.5 - [-.5]) \times (1.75 - 0)$) and the area under the target distribution is 1; 57.1% ($= 1/1.75$) of candidates should be accepted. In this example 57.8% were accepted, which will vary slightly between simulation runs.

Example 4b: Rejection sampling with unbounded support. The parameter in Example 4a was bound by $[-.5, .5]$, which permitted a convenient box to be drawn around f . Two primary changes are necessary when θ is unbounded. First, an unbounded *candidate distribution*, g , is needed. In the previous example, the candidate distribution was the uniform distribution, $U(-.5, .5)$, but now g should be chosen more carefully. Second,

the density variate drawn in Stage 3b will depend on the candidate drawn in Stage 3a. It will no longer be fixed at $U(0, c)$. The range of the uniform distribution will differ for each candidate. For instance, sometimes it is $U(0, .35)$, and sometimes $U(0, 1.3)$.

Stage 1. Specify and graph the target distribution, f (the solid bimodal line in Figure 24.3, right panel). Because f extends $(-\infty, \infty)$, decide on reasonable bounds for the graph. The target's tails should practically be zero at the graph's boundaries.

Stage 2a. Choose an appropriate g . When f covers $(-\infty, \infty)$, g also should be unbounded.

Stage 2b. Choose the density bounds. The scaling constant, c , should be defined $f(\theta) \leq c \times g(\theta)$, at all points (i.e., the solid line never exceeds the dashed line).

Stage 2c. Plot the scaled candidate distribution, $c \times g(\theta)$. Make adjustments in Stages 2a to 2b until the candidate envelopes the target completely. In Figure 24.3, right panel, we ultimately settled on $g(\theta) = t_{df=3}(\theta)$ with $c = 2$.

Stage 3a. Draw random variate x_b from g . Repeat this B times.

Stage 3b. For every candidate, find the corresponding height of the dashed line (i.e., $c \times g(x_b)$).

Draw the density variate y_b from $U(0, c \times g(x_b))$.

Stage 4. For every candidate, find $f(x_b)$. Accept and store the candidate if $f(x_b) \geq y_b$.

Stage 5. Calculate the desired summary statistics of the distribution as in Example 4a.

In Example 4a, the only explicit adjustment in Stages 2a to 2c was the c value because the candidate distribution already covered the range of the θ parameter. In this example, however, the analyst determines c and the family of the candidate distribution (along with distribution parameters like df). In practice, these are decided together with trial and error.¹⁶

¹⁵Rejection sampling can estimate improper probability distributions whose total area is not 1. The total area underneath does not matter, as long as the heights along f are correctly proportioned. This is useful in Bayesian statistics, in which the posterior is known only up to a proportional constant.

¹⁶Albert (2009, p. 99) provided an automatic way to find the scaling constant with a multivariate target distribution (although the candidate distribution and its parameter are still decided by a human). This approach improves efficiency because as c grows, more candidates are rejected and the simulation becomes less efficient. It also is useful with multivariate distributions where graphically determining c is difficult.

The choice of candidate distribution has three requirements. First, after it is multiplied by c , it must be equal to or greater than the target distribution for all values in the target. For this reason, a heavy tailed distribution is a good initial try (like a t with few degrees of freedom). Second, the target distribution should have a quick and accessible random number generator. Third, the height of the target distribution should be easily calculated. Most statistical software provides a function for producing random variates from a t distribution and calculating its pdf.

Markov Chain Monte Carlo

An MCMC simulation introduces dependencies between the B statistics. The theoretical justification and foundations of MCMC are covered in Robert and Casella (2004) and Gamerman and Lopes (2006). Only a few details differ between rejection sampling and the simplest MCMC.

Example 5a: Independent metropolis-hastings.

Rejection sampling candidates are generated independently—for example, the 53rd candidate has no effect on the value or the rejection chances of the 54th candidate. This differs from the independent Metropolis-Hastings (IMH) sampler. On the b th step, there is a competition between the *incumbent*, z_b , and the candidate, x_b . The accepted candidate becomes the incumbent for the subsequent step, z_{b+1} . The sequence of z_b values is called a *chain*.

This example reuses f and g from Example 4b. The heights of these two distributions are $f(x_b)$ and $g(x_b)$ at point x_b .

Stage 1. Specify f .

Stage 2. Choose g . From g , draw the incumbent for the chain's first step, z_1 .

Stage 3a. Draw the candidate x_b from g .

Stage 3b. Calculate a_b , which affects the candidate's chances of acceptance:

$$a_b = \frac{g(z_b)}{f(z_b)} \times \frac{f(x_b)}{g(x_b)}. \quad (7)$$

This is the ratio of the incumbent at the candidate and target distribution, multiplied by

the ratio of the new candidate at the target and candidate distribution.

Stage 4. If $a_b \geq 1$, the new candidate wins and becomes the incumbent for the next step (so, $z_{b+1} = x_b$). If $a_b < 1$, there is a runoff election in which the new candidate's probability of winning is a_b . Draw y_b from $U(0, 1)$. The new candidate wins if $a_b > y_b$; otherwise, the incumbent is reelected and survives another step (so, $z_{b+1} = z_b$).

Repeat *Stages 3a, 3b, and 4* for $b = 1, 2, \dots, B$ steps.

Stage 5. Calculate any summary statistics on the B incumbents, as in Example 4a.

As seen in the upper left panel of Figure 24.4, the candidate does not have to envelope the target distribution in an IMH. The histogram of the B accepted points matches the theoretical target distribution nicely. Compared with rejection sampling, it is less important to graph f and g because c does not exist. However, g is still required to support all possible values of f . For instance if f supports $(-\infty, \infty)$, g cannot be $\chi^2_{(df=10)}$, which supports only $(0, \infty)$.

The top right panel identifies three points (D, E, F) to illustrate the logic of jumping. Assume the incumbent is D and the candidate is E at step 70. The first ratio in a_{70} (i.e., $g(z_{70})/f(z_{70})$) equals 1 because the target and candidate distribution are equal at the incumbent's position. Point E is at the mode of g incidentally, so it is the most likely position for a candidate. However, $g(E)$ would overestimate $f(E)$ by a factor of 2 if all candidates at E were accepted; to account for the disparity between the distributions, the second ratio in a_{70} (i.e., $f(x_{70})/g(x_{70})$) is roughly .5—indicating that half of the candidates are accepted.

Assume the candidate E was rejected at Step 70, and F is the new candidate for Step 71. The value x_{71} is guaranteed victory because the first ratio in a_{71} is one and the second ratio is greater than one. The MCMC's first 100 steps are shown in the bottom panel of Figure 24.4. Flat chain links indicate the incumbent was reelected. Notice there are many longtime incumbents with values

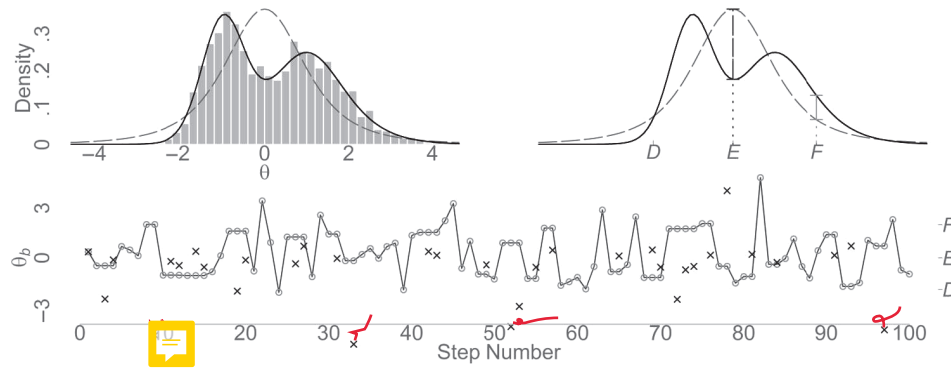


FIGURE 24.4. The target (f ; solid) and candidate (g ; dashed) distributions of an IMH (top left). A histogram of the accepted candidates (top left) closely matches the target distribution (top right). In the bottom panel, the chain's history is overlaid with victorious candidates (light gray circles) and rejected new candidates (dark gray \times symbols)

around point F (e.g., the flat sequence for Steps 71–76). Furthermore, there are many candidates around point E but few victories (e.g., the x values for Steps 11–17 and 71–76).

If f and g are equal at both x_b and z_b , then a_b equals 1 and a jump is guaranteed. If f and g are always equal, every jump is guaranteed. We later discuss the Gibbs sampler, which exploits this property in a multivariate context. In a univariate context, it would be better to simply draw x_b from f (instead of g) and always accept it. However, if it is possible to simulate directly from the univariate f , it is very likely that f has tractable equations for its CDF and standard error—so simulation is unnecessary.

The IMH is called *independent* because the candidate distribution never changes, and thus g is independent of z_b . The IMH may be practical when f is tight and has well-defined boundaries. However, when f is complex and highly dimensional, capturing “the main features of the target distribution is most often impossible” (Robert & Casella, 2004, p. 284). An MCMC can cover a multivariate space better if the candidate distribution is able to wander, which is a feature of the next sampler.

Example 5b: Metropolis-hastings. In a Metropolis-Hastings (MH) sampler, the incumbent influences g . In Example 5a, g was unaffected by the previous step and remained centered on

$\theta = 0$; g could be expressed $g_0(x_b) = t(x_b|df = 3, \text{mean} = 0)$. The MH adds a location parameter to g : $g_z(x) = g(x_b|z_b) = t(x_b|df = 3, \text{mean} = z_b)$ and $g_x(z_b) = g(z_b|x_b) = t(z_b|df = 3, \text{mean} = x_b)$. Only two procedural changes are necessary. In Stage 3a, x_b is drawn from g_{z_b} , which is centered around z_b . In Stage 3b, the acceptance variable is:

$$a_b = \frac{g(z_b|x_b)}{f(z_b)} \times \frac{f(x_b)}{g(x_b|z_b)}. \quad (8)$$

We revisit the scenario depicted Figure 24.4, upper right panel. When D is the incumbent at Steps 70 and 71, the candidates are generated from a t_3 distribution centered around D . When point F wins Step 71, g_z will shift right, and the next candidate will be drawn from a t_3 distribution centered around point F . The target distribution never moves. The candidate distribution jumps around for each x_b and z_b as it tries to recover a chain of points that are representative of the target.

Inferences are calculated directly from the chain's B points. For instance, a multilevel model uses no explicit formula for the shrinkage from a level-one parameter toward a level-two parameter (e.g., Gelman & Hill, 2007, Equation 12.1) when estimated with an MCMC. The challenging aspect of an MCMC is getting the chain to represent f . Like a bootstrap, the equations for the estimates are simply summary statistics.

The MH is the oldest and most general and flexible of the MCMC samplers. A seminal article by Metropolis and Ulam (1949) established the term *Monte Carlo method*. Newer MCMC samplers can be more efficient, but more knowledge of the target distribution is required.

Example 6: Gibbs sampler. The Gibbs sampler has two important differences from the MH. The basic MH changes every dimension at once, whereas Gibbs divides the problem into substeps and jumps in only one direction at a time. Every dimension has its own candidate distribution, which leads to the second difference between Gibbs and the MH—every candidate is accepted. The candidate and target distributions are identical, which permits direct simulation from f . When direct simulation is possible from conditional distributions, Gibbs *can* be more efficient than the MH. If f has four parameters $x = (x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)})$, the Gibbs involves four substeps in every step:

Stage 1. Determine that the joint distribution of f exists (but it does not actually need to be specified).

Stage 2. Choose starting values for each parameter $(x_1^{(1)}, x_1^{(2)}, x_1^{(3)}, x_1^{(4)})$

Stage 3. In each substep, draw a variables' candidate while fixing the other three variables:

$$\begin{aligned} x_b^{(1)} &\sim f_1(x^{(1)} | x_{b-1}^{(2)}, x_{b-1}^{(3)}, x_{b-1}^{(4)}) \\ x_b^{(2)} &\sim f_2(x^{(2)} | x_b^{(1)}, x_{b-1}^{(3)}, x_{b-1}^{(4)}) \\ x_b^{(3)} &\sim f_3(x^{(3)} | x_b^{(1)}, x_b^{(2)}, x_{b-1}^{(4)}) \\ x_b^{(4)} &\sim f_4(x^{(4)} | x_b^{(1)}, x_b^{(2)}, x_b^{(3)}) \end{aligned} \quad (9)$$

Stage 4. Automatically accept the multivariate candidate, $z_b = x_b = (x_b^{(1)}, x_b^{(2)}, x_b^{(3)}, x_b^{(4)})$.

Repeat *Stages 3 and 4* for $b = 2, 3, \dots, B$ steps.

Stage 5. Calculate any summary statistics as in Example 4a.

Stage 3 exhibits a leapfrog pattern. Variables jump one at a time, and then they stay still in

the updated position until the others complete their turn. The jump for the first variable in line, $x_b^{(1)}$, relies on the values from the previous step $(x_{b-1}^{(2)}, x_{b-1}^{(3)}, x_{b-1}^{(4)})$. The jump for the second variable, $x_b^{(2)}$, relies on the current step's value for $x^{(1)}$ but on the previous step's value for $x^{(3)}$ and $x^{(4)}$ because they have not been updated yet. This sequence continues until the last variable is updated entirely from values from the b th step.

Examples 5 and 6 have used a single chain. A recommended practice is to run at least four independent chains (e.g., Robert & Casella, 2004, Chapter 12). The algorithms are modified by running Stages 2 through 4 once for each chain. It is important that chains' positions do not affect each other. However, the summary statistics in Stage 5 combine the chains and treat their points as one large sample.

Metropolis within Gibbs. The Gibbs advantage can be exploited even when it is not possible to simulate directly from the joint f . Suppose f_1, f_2 , and f_3 could produce their respective candidates, but f_4 could not. This last substep could use an MH to draw $x^{(4)}$, while $x^{(1)}, x^{(2)}$, and $x^{(3)}$ are temporarily fixed. In fact, each substep could be replaced by a different MH. Consider a typical growth model in which each subject has three parameters; a study with 100 subjects has a target distribution with more than 300 dimensions. Robert and Casella (2010) explained the advantage:

It is most often the case that designing a Metropolis Hastings algorithm on a large-dimensional target is challenging or even impossible. The fundamental gain in using a Gibbs-like structure is that it breaks down a complex model into a large number of smaller and simpler targets, where local MH algorithms can be designed at little expense. (p. 230)

Pragmatic MCMC Issues

Expectations for learning the MCMC method are different than those for the bootstrap and rejection sampling. For a student or researcher with a solid graduate-level statistics background (say, two or more rigorous statistics courses), we believe 1 or 2 days is a reasonable amount

of time to understand the basics of bootstrap theory, program some necessary routines, and competently interpret the results for a two-factor experiment. However, learning MCMC takes more investment. The techniques are not only more complicated—both conceptually and mathematically—but also usually applied to more complex experimental designs. But their power and flexibility should be obvious. With some (worthwhile) effort, readers can appreciate the capabilities of MCMC and understand applied articles containing an MCMC analysis.

Convergence and mixing. The MH and Gibbs are defined so that f is guaranteed to be recovered after an infinite number of steps. Most applications require fewer steps, but deciding how many are needed is somewhat subjective.

There are two milestones for an MCMC. The chains’ starting values (specified in Stage 2) are not necessarily on f , especially when f has many dimensions. It is recommended to run a chain for several hundred (or several thousand) steps during a *warm-up* or *burn-in* period; these initial points are unlikely to represent f , so they are discarded and not considered by the Stage 5 statistics. Several indicators can assess different aspects of convergence, and the popular indicators are explained in MCMC and contemporary Bayesian books (e.g., Carlin & Louis, 2009; Gelman et al., 2013; Gelman et al., 2020; McElreath, 2020a; Robert & Casella, 2004).

After reaching the warm-up milestone, then determine how many steps are needed to adequately represent f . The primary concern is how well the chains continue to mix with each other and how quickly they cover f . Weak mixing can occur when successive points in a chain are strongly correlated or when a chain gets stuck in an isolated region of f , like a local maximum. One general strategy is to specify an equivalent model in which

the parameters are “as independent as possible” (Robert & Casella, 2004, p. 396; for many specific strategies, see Gelman & Hill, 2007, especially Chapter 19; Gelman et al., 2020).

Failing to converge is rarely a concern for a (properly specified) model that covers a few dimensions, because computers are powerful enough to generate a chain long enough to cover f decisively. But their brute-force nature is not ensured to be adequate for a target distribution with hundreds of dimensions (which occurs even for modest multilevel models, because each subject has multiple individual parameters).

MCMC software and resources. After running a bootstrap for 30 seconds, simulation error is usually negligible (and 1 second is adequate for most one-dimensional distributions). The duration of a nontrivial MCMC is much longer. Compared with bootstrapping, each simulation replication is less efficient and most MCMC models are much more complex. Models may require several minutes of computer time to get a rough estimate and 1 hour or more before simulation error is negligible. To reduce development time, we agree with Gelman and Hill (2007, p. 345) that similar models should be run initially with non-Bayesian software that uses maximum likelihood (ML; see also Gelman et al., 2020, section 5).¹⁷

Most recent Bayesian books use Stan (plus R or Julia) for their computational examples.¹⁸ Stan’s syntax is flexible and can even address frequentist models that may be impossible to run in frequentist software. It decides many technical details; for instance, the user does not need to determine the posterior distribution—only the prior and likelihood equations that ultimately define it.

Typically, a researcher (1) manipulates the data set with R, (2) estimates the model with MCMC software (such as Stan, BUGS, or JAGS), and (3) diagnoses convergence and views the model

¹⁷Although MCMC is less computationally efficient, it has at least three benefits over typical ML approaches. First, ML cannot incorporate prior information. Second, ML approaches fix the estimates of variance parameters instead of allowing their uncertainty to inform lower level parameter estimates appropriately (Gelman & Hill, 2007, p. 345). Third, ML finds only the mode of the likelihood distribution, whereas MCMC can capture many features of the target distribution, like its mean, modes, and quantiles (Robert & Casella, 2004, Section 9.4).

¹⁸The BUGS and JAGS programs were the community’s favorites before Stan’s release in 2012. Their strengths and weaknesses are covered in Stan Development team, 2020, Chapter 32; Lunnn et al., 2009, and their subsequent discussion; and Plummer 2017, Appendix A. Software such as SAS and Mplus have released MCMC routines, although we expect most books will continue to target the Stan syntax.

results in R again. This workflow is demonstrated in most recent applied Bayesian books (e.g., Albert, 2009; Carlin & Louis, 2009; Gelman et al., 2013; Gelman et al., 2020; Gelman & Hill, 2007; Gill, 2008; McElreath, 2020a). Presently there exist a number of R packages that facilitate the use of Stan directly from R. Packages like the ‘brms’ and ‘rstanarm’ packages provide convenient “wrappers” for Stan, so that researchers can use familiar R syntax to conduct Bayesian analyses (Bürkner, 2018; Goodrich et al., 2020). The “rethinking” package, intended for use alongside *Statistical Rethinking* (McElreath, 2020a, 2020b) provides a syntax closer to actual Stan syntax. Finally, the ‘cmdstanr’ and ‘rstan’ packages provide full Stan functionality and use the full Stan syntax (Gabry & Cesnovar, 2021; Stan Development Team, 2020). For most users, there will be little difference between the ‘cmdstanr’ and ‘rstan’ packages, although ‘cmdstanr’ is slightly ahead of the ‘rstan’ package in terms of available features and tends to use a slightly more current version of the Stan program. For non-R users, a variety of similar packages are available in other programs (e.g., Python).

There exist pedagogical and performance advantages to writing a sampler for a specific model, such as the code for Examples 5 and 6. R has many functions that make MCMC development more manageable, as well as packages such as MCMCpack that handle common details automatically but allow the user to specify the exact samplers (Martin et al., 2011; Gelman & Hill, 2007, Chapter 18). Regardless of the software, we recommend starting with the simplest possible model (e.g., the sample’s grand mean) and incrementally increasing complexity (e.g., group- and subject-level covariates). Although this appears pedantic and tedious, any syntax and logic errors are more obvious when only one feature has changed. Common accidents like misspelling a variable or creating an unidentified model are easier to detect, and the overall process is less tedious.

Furthermore, an incremental approach naturally produces a sequence of nested models that can be statistically compared with one another

(see Rodgers, 2010, for a modeling rationale). The complexity of the specified model should be given careful thought. As Fisher (1970) wrote,

No human mind is capable of grasping in its entirety the meaning of any considerable quantity of numerical data. The number of independent facts supplied by the data is usually far greater than the number of facts sought, and in consequence much of the information supplied by any body of actual data is irrelevant. It is the object of the statistical processes employed in the reduction of data to exclude this irrelevant information, and to isolate the whole of the relevant information contained in the data. (p. 6)


CONCLUSION

Simulation methods like MCMC and the bootstrap are tools that allow an applied researcher to approach questions that cannot be addressed with conventional analytic methods. The statistical tools required of well-trained behavioral science researchers now include traditional approaches such as ANOVA and categorical data analysis, along with more recently developed strategies for multilevel latent variable models and missing data. Simulation methods support the feasibility of these approaches. They provide access to many (underlying) distributions that were previously intractable, which permits statisticians to specify models that are more appropriate to their research goals.

References

- Albert, J. (2009). *Bayesian computation with R* (2nd ed.). Springer. <https://doi.org/10.1007/978-0-387-92298-0>
- Andrews, D. W. K. (2000). Inconsistency of the bootstrap when a parameter is on the boundary of the parameter space. *Econometrica*, 68(2), 399–405. <https://doi.org/10.1111/1468-0262.00114>
- Beasley, W. H., DeShea, L., Toothaker, L. E., Mendoza, J. L., Bard, D. E., & Rodgers, J. L. (2007).

- Bootstrapping to test for nonzero population correlation coefficients using univariate sampling. *Psychological Methods*, 12(4), 414–433. <https://doi.org/10.1037/1082-989X.12.4.414>
- Beasley, W. H., & Rodgers, J. L. (2009). Resampling methods. In R. E. Millsap & A. Maydeu-Olivares (Eds.), *Quantitative methods in psychology* (pp. 362–386). SAGE.
- Beran, R. (2003). The impact of the bootstrap on statistical algorithms and theory. *Statistical Science*, 18(2), 175–184. <https://doi.org/10.1214/ss/1063994972>
- Bishara, A. J., & Hittner, J. B. (2012). Testing the significance of a correlation with nonnormal data: Comparison of Pearson, Spearman, transformation, and resampling approaches. *Psychological Methods*, 17(3), 399–417. <https://doi.org/10.1037/a0028087>
- Bishara, A. J., & Hittner, J. B. (2017). Confidence intervals for correlations when data are not normal. *Behavior Research Methods*, 49(1), 294–309. <https://doi.org/10.3758/s13428-016-0702-8>
- Bollen, K. A., & Stine, R. A. (1992). Bootstrapping goodness-of-fit measures in structural equation models. *Sociological Methods & Research*, 21(2), 205–229. <https://doi.org/10.1177/0049124192021002004>
- Boos, D. D. (2003). Introduction to the bootstrap world. *Statistical Science*, 18(2), 168–174. <https://doi.org/10.1214/ss/1063994971>
- Bürkner, P. (2018). Advanced Bayesian multilevel modeling with the R package brms. *The R Journal*, 10(1), 395–411. <https://doi.org/10.32614/RJ-2018-017>
- Carlin, B. P., & Louis, T. A. (2009). *Bayesian methods for data analysis* (3rd ed.). Chapman & Hall/CRC.
- Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., & Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1), 1–32. <https://doi.org/10.18637/jss.v076.i01>
- Chan, W., & Chan, D. W. L. (2004). Bootstrap standard error and confidence intervals for the correlation corrected for range restriction: A simulation study. *Psychological Methods*, 9(3), 369–385. <https://doi.org/10.1037/1082-989X.9.3.369>
- Davison, A. C., & Hinkley, D. V. (1997). *Bootstrap methods and their application*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511802843>
- Davison, A. C., Hinkley, D. V., & Young, G. A. (2003). Recent development in bootstrap methodology. *Statistical Science*, 18(2), 141–157. <https://doi.org/10.1214/ss/1063994969>
- Diaconis, P., & Efron, B. (1983, May). Computer-intensive methods in statistics. *Scientific American*, 248(5), 116–130. <https://doi.org/10.1038/scientificamerican0583-116>
- Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, 7(1), 1–26. <https://doi.org/10.1214/aos/1176344552>
- Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. Chapman & Hall/CRC.
- Enders, C. K. (2010). *Applied missing data analysis*. Guilford Press.
- Fan, X. (2003). Using commonly available software for bootstrapping in both substantive and measurement analyses. *Educational and Psychological Measurement*, 63(1), 24–50. <https://doi.org/10.1177/0013164402239315>
- Fisher, R. A. (1970). *Statistical methods for research workers* (14th ed.). Hafner.
- Gabry, J., & Cesnovar, R. (2021). cmdstanr: R Interface to 'CmdStan'. <https://mc-stan.org/cmdstanr>, <https://discourse.mc-stan.org>
- Gamerman, D., & Lopes, H. F. (2006). *Markov chain Monte Carlo*. Chapman & Hall/CRC. <https://doi.org/10.1201/9781482296426>
- Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian data analysis* (3rd ed.). Chapman & Hall. <https://doi.org/10.1201/b16018>
- Gelman, A., & Hill, J. (2007). *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press.
- Gelman, A., Hill, J., & Vehtari, A. (2020). *Regression and other stories*. Cambridge University Press. <https://doi.org/10.1017/9781139161879>
- Gill, J. (2008). *Bayesian methods* (2nd ed.). Chapman & Hall.
- Goodrich, B., Gabry, J., Ali, I., & Brilleman, S. (2020). rstanarm: Bayesian applied regression modeling via Stan. R package version 2.21.1 <https://mc-stan.org/rstanarm>
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer. <https://doi.org/10.1007/978-0-387-84858-7>
- Hutson, A. D. (2019). A robust Pearson correlation test for a general point null using a surrogate bootstrap distribution. *PLoS One*, 14(5), e0216287. <https://doi.org/10.1371/journal.pone.0216287>
- Lahiri, S. N. (2003). *Resampling methods for dependent data*. Springer. <https://doi.org/10.1007/978-1-4757-3803-2>

- Lee, W., & Rodgers, J. L. (1998). Bootstrapping correlation coefficients using univariate and bivariate sampling. *Psychological Methods*, 3(1), 91–103. <https://doi.org/10.1037/1082-989X.3.1.91>
- LePage, R., & Billiard, L. (Eds.). (1992). *Exploring the limits of bootstrap*. Wiley.
- Lunn, D., Spiegelhalter, D., Thomas, A., & Best, N. (2009). The BUGS project: Evolution, critique and future directions. *Statistics in Medicine*, 28(25), 3049–3067. <https://doi.org/10.1002/sim.3680>
- Manly, B. (2007). *Randomization, bootstrap and Monte Carlo methods in biology* (3rd ed.). Chapman & Hall.
- Martin, A. D., Quinn, K. M., & Park, J. H. (2011). MCMCpack: Markov Chain Monte Carlo in R. *Journal of Statistical Software*, 42(9), 1–21. <https://doi.org/10.18637/jss.v042.i09>
- McElreath, R. (2020a). *A Bayesian course with examples in R and Stan*. Chapman & Hall/CRC. <https://doi.org/10.1201/9780429029608>
- McElreath, R. (2020b). rethinking: Statistical rethinking course and book package [Computer software]. <https://github.com/rmcelreath/rethinking>.
- Mendoza, J. L., Hart, D. E., & Powell, A. (1991). A bootstrap confidence interval based on a correlation corrected for range restriction. *Multivariate Behavioral Research*, 26(2), 255–269. https://doi.org/10.1207/s15327906mbr2602_4
- Metropolis, N., & Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Association*, 44(247), 335–341. <https://doi.org/10.1080/01621459.1949.10483310>
- Monahan, J. F. (2001). *Numerical methods of statistics*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511812231>
- O’Keefe, P., & Rodgers, J. L. (2020). A simulation study of bootstrap approaches to estimate confidence intervals in DeFries–Fulker regression models (with application to the heritability of BMI changes in the NLSY). *Behavior Genetics*, 50(2), 127–138. <https://doi.org/10.1007/s10519-020-09993-9>
- Plummer, M. (2017). *JAGS Version 4.3.0 user manual*. https://people.stat.sc.edu/hansont/stat740/jags_user_manual.pdf
- Poi, B. P. (2004). From the help desk: Some bootstrapping techniques. *Stata Journal*, 4, 312–328. <https://journals.sagepub.com/doi/pdf/10.1177/1536867X0400400308>
- Robert, C. P., & Casella, G. (2004). *Monte Carlo statistical methods*. Springer. <https://doi.org/10.1007/978-1-4757-4145-2>
- Robert, C. P., & Casella, G. (2010). *Introducing Monte Carlo methods with R*. Springer. <https://doi.org/10.1007/978-1-4419-1576-4>
- Rodgers, J. L. (1999). The bootstrap, the jackknife, and the randomization test: A sampling taxonomy. *Multivariate Behavioral Research*, 34(4), 441–456. https://doi.org/10.1207/S15327906MBR3404_2
- Rodgers, J. L. (2010). The epistemology of mathematical and statistical modeling: A quiet methodological revolution. *American Psychologist*, 65(1), 1–12. <https://doi.org/10.1037/a0018326>
- Rodgers, J. L., & Beasley, W. H. (2013). Fisher, Gossett, and AHST: Bootstrapping multiple correlation alternative hypotheses. In M. Edwards & R. MacCallum (Eds.), *Current Topics in the Theory and Application of Latent Variable Models* (pp. 217–239). Routledge.
- Rodgers, J. L., Nicewander, W. A., & Toothaker, L. (1984). Linearly independent, uncorrelated, and orthogonal variables. *The American Statistician*, 38(2), 133–134. <https://doi.org/10.2307/2683250>
- Smith, T. A., & Kimball, D. R. (2010). Learning from feedback: Spacing and the delay-retention effect. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 36(1), 80–95. <https://doi.org/10.1037/a0017407>
- Stan Development Team. (2020). *RStan: The R interface to Stan*. R package version 2.21.2. <https://mc-stan.org/> 
- Stan Development Team. (2020). *Stan modeling language users guide and reference manual*, VERSION. <https://mc-stan.org>
- Steiger, J. H. (2007, August). *Statistical games we all should play*. Paper presented at the 115th Annual Convention of the American Psychological Association, San Francisco, CA.
- Tukey, J. W. (1977). *Exploratory data analysis*. Addison-Wesley.

