The Distribution of Order Statistics for Discrete Random Variables with Applications to Bootstrapping

Diane L. Evans
Department of Mathematics, Rose–Hulman Institute of Technology, 5500 Wabash Avenue, Terre Haute, Indiana 47803, USA, diane.evans@rose-hulman.edu
Lawrence M. Leemis, John H. Drew
Department of Mathematics, The College of William & Mary, P.O. Box 8795, Williamsburg, Virginia 23187–8795, USA, {leemis@math.wm.edu, jhdrew@math.wm.edu}

An algorithm for computing the PDF of order statistics drawn from discrete parent populations is presented, along with an implementation of the algorithm in a computer algebra system. Several examples and applications, including exact bootstrapping analysis, illustrate the utility of this algorithm. Bootstrapping procedures require that B bootstrap samples be generated in order to perform statistical inference concerning a data set. Although the requirements for the magnitude of B are typically modest, a practitioner would prefer to avoid the resampling error introduced by choosing a finite B, if possible. The part of the order-statistic algorithm for sampling with replacement from a finite sample can be used to perform exact bootstrapping analysis in certain applications, eliminating the need for replication in the analysis of a data set.

Key words: combinatorial algorithms; computer algebra systems; probability; probability distributions; statistics
History: Accepted by Susan M. Sanchez, Area Editor; received June 2000; revised August 2001, October 2003, April 2004; accepted June 2004.

---

1. Introduction

As evidenced by over a thousand references cited in the survey text by David and Nagaraja (2003), a large amount of literature has been devoted to the theory and application of order statistics. In conjunction, a growing interest in undergraduate and graduate courses in order statistics (Arnold et al. 1992) has also emerged over the past ten years. Many popular mathematical statistics texts, such as Hogg and Craig (1995), Casella and Berger (2002), or Rice (1995), address only distributions of order statistics for independent and identically distributed random variables drawn from continuous populations due to the mathematical

tractability of their theoretical formulas. Further, order statistics for continuous parent populations have found important applications in many areas, including survival analysis, life testing, reliability, robustness studies, statistical quality control, filtering theory, signal processing, image processing, and radar target detection (Nagaraja et al. 1996, Balakrishnan and Rao 1998).

Work in the late 1980s through the 1990s examined the theory of order statistics for nonidentically distributed and dependent variables, but again, for random variables drawn from continuous populations. Boncelet (1987), Balakrishnan (1988), Bapat and Beg (1989), and Balakrishnan et al. (1992) relate the distributions of order statistics in samples of size k to those in samples of sizes $k-1$, generalizing formulas in the standard independent, identically distributed case. Subsequently, Cao and West (1997) wrote algorithms that extend the known theory of distributions of order statistics of independent, but not identically distributed, random quantities to practical situations.

Results for order statistics drawn from discrete parent populations are sparse and usually specialized to fit one particular discrete population (e.g., Young 1970, Srivastava 1974, Ciardo et al. 1995). Arnold et al. (1992) devote a chapter to discrete order statistics. We will present an algorithm in a computer algebra system (CAS) for determining distributions of order statistics drawn from general discrete parent populations and the application of this algorithm to exact bootstrap analysis. Using Efron and Tibshirani's (1993) notation, we will consider eliminating the generation of B bootstrap samples when performing a bootstrap analysis by calculating the exact distribution of the statistic of interest. This process eliminates the resampling variability that is present in bootstrapping procedures.

The algorithm presented in this paper handles discrete parent populations with finite or infinite support, and sampling with or without replacement. Development of this algorithm provides the scientific community with easy access to many discrete order-statistic distributions. Application areas with respect to bootstrapping that are presented in this paper are the estimation of standard errors for the median, mean, and range, and interval estimation for the range.

2. Algorithm

Let $X_1, X_2, \ldots, X_n$ be n independent and identically distributed (iid) random variables defined on Ω, each with CDF $F_X(x)$ and PDF $f_X(x)$. Let $X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n)}$ denote

these random variables rearranged in nondescending order of magnitude. Thus, $X_{(r)}$ is the rth smallest number in the sample, $r = 1, 2, \ldots, n$. Because order statistics are random variables, it is possible to compute probability values associated with values in their support.

When the population is continuous, the PDF of the rth order statistic can be expressed easily because the probability that any two $X_{(j)}$'s are the same is zero. As is well known (e.g., Casella and Berger 2002, p. 229), the PDF of $X_{(r)}$ is

$$f_{X_{(r)}}(x) = \frac{n!}{(r-1)!(n-r)!} f_X(x) \left[F_X(x)\right]^{r-1} \left[1 - F_X(x)\right]^{n-r}, \quad x \in \Omega$$

for $r = 1, 2, \ldots, n$.

If $X_1, X_2, \ldots, X_n$ is a random sample from a discrete population, then the PDF of the rth order statistic cannot always be expressed as a single formula, as in the continuous case. When working with discrete random variables, the computation of the PDF of the rth order statistic will fall into one of several categories, depending on the sampling convention (with or without replacement), the random variable's support (finite or infinite), and the random variable's distribution (equally likely or nonequally likely probabilities). A taxonomy of these categories appears in Figure 1. The bootstrapping application requires sampling with replacement from a finite population, but other applications require the additional branches of the algorithm as displayed in Figure 1.

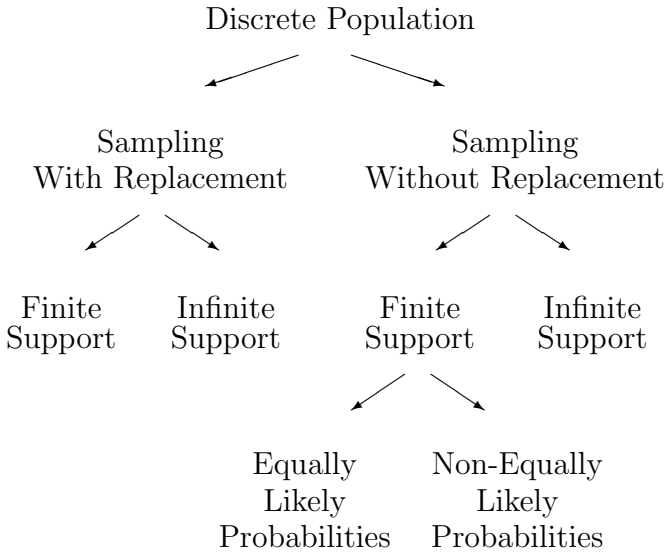

Figure 1: Categorization of Discrete Order Statistics by Sampling Convention, Support, and Probability Distribution

The `OrderStat(X, n, r, [wo])` (henceforth referred to as `OrderStat`) algorithm *requires* three arguments: a random variable `X`, the number of items `n` randomly drawn from

the population with PDF $f_X(x)$, and the index `r` of the desired order statistic. An optional fourth argument `wo` can be specified to indicate that the items are drawn from the population *without* replacement. The implementation steps of the `OrderStat` algorithm are explained in detail in the following two subsections ("Without Replacement" and "With Replacement"). In both cases, the output of the algorithm is $f_{X_{(r)}}(x)$, the PDF of the rth order statistic of the random variable X, where n items have been sampled either with or without replacement from the population with PDF $f_X(x)$. After each branch (displayed in Figure 1) of the algorithm is discussed in its appropriate subsection, an example with solution of the type of problem that fits into that particular branch classification is provided.

The `OrderStat` algorithm has been implemented in Maple and is one of over 30 procedures included in a probability package developed to automate the naming, processing, and application of random variables. The software is referred to as "A Probability Programming Language" (APPL) and is described in Glen et al. (2001). APPL is available on the first author's website at `http://www.rose-hulman.edu/~evans/appl/index.html` or by e-mail request.

2.1 Without Replacement

Providing the string `wo` as the fourth argument to the `OrderStat` procedure indicates that items are drawn from the discrete population without replacement. Random variables with finite or infinite support are handled by two separate techniques.

If the population distribution has finite support, the population size is denoted by N. To specify this support in a compact form, it is temporarily assumed to be (without loss of generality) the ordered set $\{1, 2, \ldots, N\}$. For example, if

$$f_X(x) = \begin{cases} 0.5 & x = 7 \\ 0.2 & x = 11 \\ 0.3 & x = 15, \end{cases}$$

then $N = 3$, and the support of X is temporarily assumed to be $\{1, 2, 3\}$, instead of $\{7, 11, 15\}$. The original support is reinstated after the order-statistic probabilities are computed.

If the population with PDF $f_X(x)$ has equally likely probability values, e.g., rolling a fair die, then by combinatorial analysis the PDF of the rth order statistic when n items are

sampled is (Arnold et al. 1992, p. 54)

$$f_{X_{(r)}}(x) = \frac{\binom{x-1}{r-1}\binom{N-x}{n-r}}{\binom{N}{n}}, \quad x = r, r+1, \ldots, r+N-n.$$

Example 1. (*Sampling without replacement; finite support; equally likely probabilities*) (Hogg and Craig 1995, p. 231) Draw 15 cards at random and without replacement from a deck of 25 cards numbered $1, 2, \ldots, 25$. Find the probability that the card numbered 10 is the median of the cards selected.

Solution: Let $f_X(x) = \frac{1}{25}$ for $x = 1, 2, \ldots, 25$. The population size is $N = 25$, the size of the sample is $n = 15$, and the sample median $X_{(8)}$ corresponds to $r = 8$. The probability that the sample median equals 10 is given by the above formula as

$$\begin{aligned} \Pr(X_{(8)} = 10) &= \frac{\binom{9}{7}\binom{15}{7}}{\binom{25}{15}} \\ &= \frac{1053}{14858} \\ &\cong 0.0709. \end{aligned}$$

In addition to just computing one probability value as obtained by hand above, the `OrderStat` algorithm returns the PDF of the 8th order statistic (when $n = 15$ items are sampled from the population without replacement), which is

$$f_{X_{(8)}}(x) = \frac{1}{3268760}\binom{x-1}{7}\binom{25-x}{7} \qquad x = 8, 9, \ldots, 18.$$

□

If X has finite support N and nonequally likely probabilities, there are three cases to consider when computing the PDF of the rth order statistic:

1. One item is sampled: $n = 1$. The PDF of the rth order statistic is the same as the population PDF; i.e., $f_{X_{(r)}}(x) = f_X(x)$ for $r = 1, 2, \ldots, N$.

2. The entire population is sampled: $n = N$. The PDF of the rth order statistic is $f_{X_{(r)}}(x) = 1$ for $x = r$.

3. More than one item, but not the entire population, is sampled: $n = 2, 3, \ldots, N-1$. In this nontrivial case, an $n \times N$ array, *ProbStorage*, is defined that eventually contains the probabilities $f_{X_{(r)}}(x)$ for $x = 1, 2, \ldots, N$ and $r = 1, 2, \ldots, n$. The value of $f_{X_{(r)}}(x)$ is found in row r and column x of the array. For example, if $X \sim$ Binomial(5, $p = \frac{1}{4}$) and $n = 3$ items are sampled (without replacement) from this population, then the *ProbStorage* array is 3×6 and has the form displayed in Figure 2.

r \ x	1	2	3	4	5	6
1	*	*	*	*	0	0
2	0	*	*	*	*	0
3	0	0	*	*	*	*

Figure 2: Initial Form of *ProbStorage* Array for $X \sim$ Binomial(5, $\frac{1}{4}$) Where $n = 3$ Items are Sampled Without Replacement. Asterisks Denote Positive Probability Values; Zeros Denote Impossible Situations.

The algorithm's implementation requires use of two APPL combinatorial procedures, `NextCombination` and `NextPermutation`, which were specifically created for use in the `OrderStat` algorithm. Following is a brief description of the algorithm for the without-replacement/finite support/nonequally-likely-probabilities case for $n = 2, 3, \ldots, N-1$.

(a) The first lexicographical combination of n items sampled from the sequence of integers 1 to N is formed; it is $\{1, 2, \ldots, n\}$. For $X \sim$ Binomial(5, $\frac{1}{4}$), the first lexicographical combination of $n = 3$ items from $\{1, 2, \ldots, 6\}$ is $\{1, 2, 3\}$.

(b) Given a combination consisting of n distinct integers, the algorithm generates all possible permutations of that combination and their corresponding probabilities. For the example $X \sim$ Binomial(5, $\frac{1}{4}$), the first combination is $\{1, 2, 3\}$ and its permutations are $[1, 2, 3]$, $[1, 3, 2]$, $[2, 1, 3]$, $[2, 3, 1]$, $[3, 1, 2]$, $[3, 2, 1]$. The probabilities obtained for each of these permutations, respectively, are $\frac{13286025}{150351872}$, $\frac{13286025}{204334592}$, $\frac{13286025}{119164928}$, $\frac{13286025}{110607872}$, $\frac{13286025}{197270528}$, and $\frac{13286025}{134730752}$. As an example, the probability of obtaining the permutation $[3, 1, 2]$ is computed as

$$f_X(3)\,\frac{f_X(1)}{1 - f_X(3)}\,\frac{f_X(2)}{1 - (f_X(3) + f_X(1))}$$

or

$$\frac{135}{512}\,\frac{\frac{243}{1024}}{1 - \frac{135}{512}}\,\frac{\frac{405}{1024}}{1 - \left(\frac{135}{512} + \frac{243}{1024}\right)} = \frac{13286025}{197270528}.$$

(c) After the probability of each permutation generated in (b) is computed, each permutation is "rewritten" lexicographically to determine the corresponding order-statistic probabilities. The permutation $[3, 1, 2]$, for example, is rewritten as $[1, 2, 3]$. The algorithm then adds the probability of obtaining that particular permutation to the appropriate three cells in the *ProbStorage* array. The (r, x)

r \ x	1	2	3	4	5	6
1	$\Sigma(1,1)+pr$	$\Sigma(1,2)$	$\Sigma(1,3)$	$\Sigma(1,4)$	0	0
2	0	$\Sigma(2,2)+pr$	$\Sigma(2,3)$	$\Sigma(2,4)$	$\Sigma(2,5)$	0
3	0	0	$\Sigma(3,3)+pr$	$\Sigma(3,4)$	$\Sigma(3,5)$	$\Sigma(3,6)$

Figure 3: Computation of *ProbStorage* Array for $X \sim$ Binomial(5, $\frac{1}{4}$) Where $n = 3$ Items are Sampled Without Replacement. Given the Permutation [3, 1, 2], its Probability $pr = \frac{13286025}{197270528}$ is Added to the Current Probability Sums $\Sigma(r, x)$ in Cells (1, 1), (2, 2), and (3, 3). Zeros Denote Impossible Situations.

cell accumulates the probabilities of the various permutations for which the rth order statistic has the value x. For example, the final value in the (1, 1) cell represents the probability that the first order statistic assumes the value one. In the Binomial(5, $\frac{1}{4}$) example, the probability of obtaining the permutation [3, 1, 2], $\frac{13286025}{197270528}$, is added to the *ProbStorage* cells (1, 1), (2, 2), and (3, 3). See Figure 3.

(d) After all $n!$ permutations of a given combination are exhausted, the procedure `NextCombination` determines the next lexicographical combination. Given this new combination, the algorithm repeats steps (b) and (c). This process iterates through the $\binom{N}{n}$ combinations of size n chosen from a population of size N.

Example 2. (*Sampling without replacement; finite support; nonequally likely probabilities*) Let X be a random variable with PDF

$$f_X(x) = \begin{cases} p_1 & x = 1 \\ p_2 & x = 10 \\ p_3 & x = 100 \\ p_4 & x = 1000, \end{cases}$$

where $p_1 + p_2 + p_3 + p_4 = 1$ and $p_i > 0$, $i = 1, 2, 3, 4$. Find the distribution of the median order statistic when $n = 3$ items are sampled without replacement from the population.

Solution: The PDF for the median order statistic computed using the `OrderStat` algorithm is

$$f_{X_{(3)}}(x) = \begin{cases} \frac{p_4\,p_2\,p_1}{(1-p_4)(1-p_2-p_4)} + \frac{p_4\,p_2\,p_1}{(1-p_4)(1-p_1-p_4)} + \frac{p_4\,p_2\,p_1}{(1-p_2)(1-p_2-p_4)} + \frac{p_4\,p_2\,p_1}{(1-p_2)(1-p_1-p_2)} + \\ \frac{p_4\,p_2\,p_1}{(1-p_1)(1-p_1-p_4)} + \frac{p_4\,p_2\,p_1}{(1-p_1)(1-p_1-p_2)} + \frac{p_3\,p_2\,p_1}{(1-p_3)(1-p_2-p_3)} + \frac{p_3\,p_2\,p_1}{(1-p_3)(1-p_1-p_3)} + \\ \frac{p_3\,p_2\,p_1}{(1-p_2)(1-p_2-p_3)} + \frac{p_3\,p_2\,p_1}{(1-p_2)(1-p_1-p_2)} + \frac{p_3\,p_2\,p_1}{(1-p_1)(1-p_1-p_3)} + \frac{p_3\,p_2\,p_1}{(1-p_1)(1-p_1-p_2)} & x = 10 \\ \frac{p_4\,p_3\,p_2}{(1-p_4)(1-p_4-p_3)} + \frac{p_4\,p_3\,p_2}{(1-p_4)(1-p_2-p_4)} + \frac{p_4\,p_3\,p_2}{(1-p_3)(1-p_4-p_3)} + \frac{p_4\,p_3\,p_2}{(1-p_3)(1-p_2-p_3)} + \\ \frac{p_4\,p_3\,p_2}{(1-p_2)(1-p_2-p_4)} + \frac{p_4\,p_3\,p_2}{(1-p_2)(1-p_2-p_3)} + \frac{p_4\,p_3\,p_1}{(1-p_4)(1-p_4-p_3)} + \frac{p_4\,p_3\,p_1}{(1-p_4)(1-p_1-p_4)} + \\ \frac{p_4\,p_3\,p_1}{(1-p_3)(1-p_4-p_3)} + \frac{p_4\,p_3\,p_1}{(1-p_3)(1-p_1-p_3)} + \frac{p_4\,p_3\,p_1}{(1-p_1)(1-p_1-p_4)} + \frac{p_4\,p_3\,p_1}{(1-p_1)(1-p_1-p_3)} & x = 100. \end{cases}$$

□

If the population distribution has a countably infinite support, i.e., $\{1, 2, \ldots\}$, then the pattern established for finding the PDF of the rth order statistic in the finite-support case will not work. At this time, beyond the trivial case when $n = 1$ item is sampled and $f_{X_{(r)}}(x) = f_X(x)$ for $r = 1, 2, \ldots, N$, the `OrderStat` algorithm computes the PDF of the minimum order statistic when at most $n = 2$ items are sampled without replacement from a discrete population with infinite support. The $n = 2$ case is considered in the following paragraph. Future work with `OrderStat` will begin to incorporate the algorithmic pattern for values of $n \geq 3$. Currently, no literature considers these cases.

When $n = 2$ items are sampled without replacement, the probability that the minimum order statistic has value x, for $x = 1, 2, \ldots$, is given by

$$\Pr(X_{(1)} = x) = \Pr(X_1 = x) \Pr(X_2 \geq x+1 \,|\, X_1 = x) + \sum_{y=x+1}^{\infty} \Pr(X_1 = y) \Pr(X_2 = x \,|\, X_1 = y).$$

Thus, the PDF of $X_{(1)}$ when $n = 2$ items are sampled is

$$f_{X_{(1)}}(x) = f_X(x) \left[\frac{S_X(x+1)}{1 - f_X(x)} + \sum_{y=x+1}^{\infty} \frac{f_X(y)}{1 - f_X(y)} \right] \qquad x = 1, 2, \ldots,$$

where $S_X(x)$ is the survivor function defined by $S_X(x) = \Pr(X \geq x)$.

Example 3. (*Sampling without replacement; infinite support*) Let $X \sim \text{Geometric}(\frac{1}{2})$. Find the probability that the minimum order statistic is five when $n = 2$ items are sampled without replacement from the population.

Solution: The PDF of the minimum order statistic as calculated using the `OrderStat` algorithm is

$$f_{X_{(1)}}(x) = \frac{\left(\frac{1}{4}\right)^x}{1 - \left(\frac{1}{2}\right)^x} + \frac{1}{2} \left(\frac{1}{2}\right)^{x-1} \sum_{y=x+1}^{\infty} \frac{1}{2} \frac{\left(\frac{1}{2}\right)^{y-1}}{1 - \frac{1}{2}\left(\frac{1}{2}\right)^{y-1}} \qquad x = 1, 2, \ldots.$$

By substitution, it can easily be determined that $\Pr(X = 5) \cong 0.0020$. □

2.2 With Replacement

If the optional fourth argument `wo` is *not* provided to the `OrderStat` procedure, the items are assumed to be sampled from the discrete population with replacement. When sampling with replacement, the probability density function, cumulative distribution function (CDF) $F_X(x)$, and survivor function (SF) $S_X(x)$ are needed to determine the distribution of the rth

order statistic. The APPL procedures `PDF`, `CDF`, and `SF`, respectively, determine these forms of a distribution. Computing an order-statistic distribution when sampling with replacement from a finite population will be used in the bootstrapping application.

If the random variable X has finite support Ω, then without loss of generality we can assume that $\Omega = \{1, 2, \ldots, N\}$. The PDF of $X_{(r)}$ when n items are sampled with replacement from this finite population is given by

$$f_{X_{(r)}}(x) = \begin{cases} \displaystyle\sum_{w=0}^{n-r} \binom{n}{w} [f_X(1)]^{n-w} [S_X(2)]^w & x = 1 \\ \displaystyle\sum_{u=0}^{r-1} \sum_{w=0}^{n-r} \binom{n}{u, n-u-w, w} [F_X(x-1)]^u [f_X(x)]^{n-u-w} [S_X(x+1)]^w & x = 2, 3, \ldots, N-1 \\ \displaystyle\sum_{u=0}^{r-1} \binom{n}{u} [F_X(N-1)]^u [f_X(N)]^{n-u} & x = N. \end{cases}$$

This formula is not valid in the special case when the discrete population consists of only one item $N = 1$. The PDF of the rth order statistic in this case is simply $f_{X_{(r)}}(1) = 1$.

The formula for the PDF of $X_{(r)}$ is a direct result of the following observation. For the rth order statistic to equal x, there must be u values less than or equal to $x-1$ and w values greater than or equal to $x+1$, where $u = 0, 1, \ldots, r-1$ and $w = 0, 1, \ldots, n-r$. The other $n-u-w$ values must be equal to x. The `CDF` procedure is used to determine the probability of obtaining a value less than or equal to $x-1$, the `SF` procedure is used to determine the probability of obtaining a value greater than or equal to $x+1$, and the `PDF` procedure is used to determine the probability of obtaining the value x. The multinomial coefficient expresses the number of combinations resulting from a specific choice of u and w.

Taking an example from Arnold et al. (1992, p. 43), let X be a discrete uniform random variable with PDF $f_X(x) = \frac{1}{4}$, $x = 1, 2, 3, 4$. Then the CDF and SF of X, respectively, are

$$F_X(x) = \begin{cases} 0 & x < 1 \\ \frac{\lfloor x \rfloor}{4} & 1 \le x < 4 \\ 1 & x \ge 4 \end{cases} \qquad S_X(x) = \begin{cases} 1 & x \le 1 \\ 1 - \frac{\lfloor x \rfloor}{4} & 1 < x \le 4 \\ 0 & x > 4. \end{cases}$$

Suppose $n = 5$ values are sampled with replacement from this population. To calculate $f_{X_{(2)}}(3)$, i.e., the probability that the second order statistic is $x = 3$, evaluate the sum

$$f_{X_{(2)}}(3) = \sum_{u=0}^{1} \sum_{w=0}^{3} \binom{5}{u, 5-u-w, w} [F_X(2)]^u [f_X(3)]^{5-u-w} [S_X(4)]^w .$$

The first term in the summation is the probability of drawing all threes. The second term, in which $u = 0$ and $w = 1$, is the probability of drawing four threes and a value greater than or equal to four (which can only be the value four in this example). This collection of four threes and one four can be drawn five different ways. The subsequent terms in the sum have similar meanings.

Example 4. (*Sampling with replacement; finite support*) (Hogg and Craig 1995, p. 230) A fair die is cast eight independent times. Find the PDF of the smallest of the eight numbers obtained, $X_{(1)}$.

Solution: To compute the numeric PDF by hand,

$$f_{X_{(1)}}(x) = \sum_{w=0}^{7} \binom{8}{w} \left(\frac{1}{6}\right)^{8-w} \left(1 - \frac{x}{6}\right)^{w}$$

is calculated for $x = 1, 2, \ldots, 6$. For example, the probability that the first order statistic is $x = 4$ is

$$\begin{aligned} f_{X_{(1)}}(4) &= \sum_{w=0}^{7} \binom{8}{w} \left(\frac{1}{6}\right)^{8-w} \left(\frac{1}{3}\right)^{w} \\ &= \tfrac{1}{1679616} + \tfrac{1}{104976} + \tfrac{7}{104976} + \tfrac{7}{26244} + \tfrac{35}{52488} + \tfrac{7}{6561} + \tfrac{7}{6561} + \tfrac{4}{6561} \\ &= \tfrac{6305}{1679616} \\ &\cong 0.0038. \end{aligned}$$

Similar calculations for $x = 1, 2, \ldots, 6$ yield the PDF of the first order statistic as

$$f_{X_{(1)}}(x) = \begin{cases} \frac{1288991}{1679616} & x = 1 \\ \frac{36121}{186624} & x = 2 \\ \frac{58975}{1679616} & x = 3 \\ \frac{6305}{1679616} & x = 4 \\ \frac{85}{559872} & x = 5 \\ \frac{1}{1679616} & x = 6. \end{cases}$$

The `OrderStat` algorithm yields the same results, but in addition returns the PDF of the minimum order statistic as a polynomial:

$$f_{X_{(1)}}(x) = -\tfrac{1}{209952}\, x^7 + \tfrac{91}{419904}\, x^6 - \tfrac{889}{209952}\, x^5 + \tfrac{38675}{839808}\, x^4 - \tfrac{63217}{209952}\, x^3 + \tfrac{496951}{419904}\, x^2 - \tfrac{543607}{209952}\, x + \tfrac{4085185}{1679616}$$

for $x = 1, 2, \ldots, 6$. □

If the support of X is countably infinite, i.e., $\Omega = \{1, 2, \ldots\}$, then the calculation of the PDF of $X_{(r)}$ is similar to the finite-support case. The main difference is that the formula

used in the finite-support case is now used for values of x that are unbounded:

$$f_{X_{(r)}}(x) = \begin{cases} \displaystyle\sum_{w=0}^{n-r} \binom{n}{w} [f_X(1)]^{n-w} [S_X(2)]^w & x = 1 \\ \displaystyle\sum_{u=0}^{r-1} \sum_{w=0}^{n-r} \binom{n}{u, n-u-w, w} [F_X(x-1)]^u [f_X(x)]^{n-u-w} [S_X(x+1)]^w & x = 2, 3, \ldots \end{cases}$$

Because the formula assumes that the support of X is $\{1, 2, \ldots\}$, the algorithm works only for distributions with infinite right-hand tails.

Since it is physically impossible to execute the formula for infinitely many values of x, a general expression for $f_{X_{(r)}}(x)$ in terms of x is obtained by taking advantage of Maple's ability to sum and simplify complicated symbolic expressions. Maple computes (in symbolic terms) the double summation of a multinomial coefficient multiplied by the product of the CDF, PDF, and SF raised to various powers.

Example 5. (*Sampling with replacement; infinite support*) Define a geometric random variable X with parameter p $(0 < p < 1)$ to be the trial number of the first success in repeated independent Bernoulli(p) trials. The PDF of X is $f_X(x) = p\, q^{x-1}$, for $x = 1, 2, \ldots$, where $q = 1 - p$. Margolin and Winokur (1967) tabulated values for the mean and variance of the rth order statistic of a geometric distribution for samples of size $n = 1, 5\,(5)\,20$ for $r = 1\,(1)\,5\,(5)\,20$, where $r = 1, 2, \ldots, n$ and $q = 0.25\,(0.25)\,0.75$. The values are calculated to two decimal places.

Margolin and Winokur's treatment of order statistics for the geometric distribution is based on the use of recurrence relations. The formulas they use to compute the first and second moments of the rth order statistic of a geometric distribution (with parameter $p = 1 - q$) when n items are sampled (with replacement) are

$$\mathrm{E}\left[X_{(r)}\right] = n \binom{n-1}{r-1} \sum_{j=0}^{r-1} \frac{(-1)^j \binom{r-1}{j}}{(n-r+j+1)\,(1-q^{n-r+j+1})}$$

and

$$\mathrm{E}\left[X_{(r)}^2\right] = n \binom{n-1}{r-1} \sum_{j=0}^{r-1} \frac{(-1)^j \binom{r-1}{j} (1+q^{n-r+j+1})}{(n-r+j+1)\,(1-q^{n-r+j+1})^2}.$$

The `OrderStat` algorithm can readily produce the *exact value* of any of the rounded figures given in their tables. If $X \sim \text{Geometric}(\frac{1}{4})$, for example, then the exact values of the mean and variance of the third order statistic when $n = 5$ items are sampled can be found with the `OrderStat` algorithm as $\frac{3257984}{1011395} \cong 3.22$ and $\frac{13665723739776}{5114599230125} \cong 2.67$, respectively. (`Mean`

and `Variance` are additional APPL procedures that have the random variable of interest as their argument.) The `OrderStat` algorithm has the capability to compute the mean and variance for a much larger range of arguments, including a symbolic probability p, than provided in their table. If $Y \sim \text{Geometric}(p)$, for example, then the variance of the minimum order statistic when $n = 6$ items are sampled with replacement is $\frac{1-6p+15p^2-20p^3+15p^4-6p^5+p^6}{p^2(p^5-6p^4+15p^3-20p^2+15p-6)}$.

3. Applications

3.1 Range Statistics

One natural extension to the `OrderStat` algorithm is a `RangeStat` algorithm, which finds the PDF of the range of a sample of n items drawn from a discrete population, either with or without replacement. This procedure will also be used in the bootstrapping application.

Let X be a discrete random variable with PDF $f_X(x)$ and CDF $F_X(x)$, and let Y be a random variable representing the range of X. We can assume (without loss of generality) that the support of X is $\{1, 2, \ldots, N\}$, where N is a positive integer or infinity. If we are sampling with replacement (which implies that the n draws are independent), then the probability that Y assumes the value y, where $y = 0, 1, \ldots, N-1$, is given by Burr (1955):

$$\Pr(Y = y) = \begin{cases} \sum_{x=1}^{N} [f_X(x)]^n & y = 0 \\ \sum_{x=1}^{N-y} \{[\Pr(x \le X \le x+y)]^n - [\Pr(x+1 \le X \le x+y)]^n \\ -[\Pr(x \le X \le x+y-1)]^n + [\Pr(x+1 \le X \le x+y-1)]^n\} = \\ \sum_{x=1}^{N-y} \{[F_X(x+y) - F_X(x-1)]^n - [F_X(x+y) - F_X(x)]^n \\ -[F_X(x+y-1) - F_X(x-1)]^n + [F_X(x+y-1) - F_X(x)]^n\} & y = 1, 2, \ldots N-1. \end{cases}$$

Of the four terms being summed in the $y = 1, 2, \ldots, N-1$ case, the first term is the probability that all sampled elements lie between x and $x+y$ inclusive. The second term removes the probability that these elements do not include x, because this would result in a range that is less than y. The third term removes the probability that these elements do not include $x+y$, because this would also result in a range that is less than y. The fourth term adds back the probabilities of those elements that include neither x nor $x+y$, which were removed by both the second and third terms.

In the without-replacement case, the `RangeStat` algorithm basically follows the same steps as the `OrderStat` algorithm, including use of the procedures `NextCombination` and `NextPermutation`. In the trivial case when the entire population is sampled, the PDF of the range Y is $f_Y(y) = 1$ for $y = N - 1$. If $n = 2, 3, \ldots, N$, where N is a positive integer less than or equal to n, then a single-dimensional array of length $N - n + 1$ is defined that will eventually contain the values of $f_Y(y)$ for $y = n - 1, n, \ldots, N - 1$. As in the `OrderStat` algorithm, the first lexicographical combination of n items sampled from the sequence of integers 1 to N is formed. Given a combination, the algorithm generates all possible permutations of that combination. The probability for each permutation is calculated (as described in Section 2.1), and then the maximum and minimum values of that permutation is determined. The permutation's range, which is the difference between its maximum and minimum values, is computed and then the appropriate array cell is incremented by that permutation's probability.

3.2 Bootstrapping

"One of the most influential developments in statistics in the last decade has been the introduction and rapid dissemination of bootstrap methods" (Rice 1995, Preface). "Much of the current appeal of bootstrap, without doubt, stems from the not unrealistic hope of obtaining — as much of the research effort has been geared to show — higher-order accuracy in an automatic and simple manner ... Only recently has attention been paid to the practically crucial question of providing the user with some means of assessing how well-determined, or accurate, the bootstrap estimator is" (Young 1994, p. 384).

Using Efron and Tibshirani's (1993) notation, this section considers eliminating the generation of B bootstrap samples when performing a bootstrap analysis by calculating the exact distribution of the statistic of interest. There are several reasons, as alluded to in the previous paragraph, for considering this approach:

- Although bootstrap methods are relatively easy to apply, determining the number of bootstrap repetitions, B, to employ is a common problem facing practitioners (Andrews and Buchinsky 2002). A practitioner needs to be concerned about problem-specific requirements for B, e.g., "B in the range of 50 to 200 usually makes $\hat{\text{se}}_{\text{boot}}$ a good standard error estimator, even for estimators like the median" (Efron and Tibshirani 1993, p. 14) or "B should be $\geq$ 500 or 1000 in order to make the variability of [the

estimated 95th percentile] acceptably low" for estimating 95th percentiles (Efron and Tibshirani 1993, p. 275). In fact, "one can obtain a 'different answer' from the same data merely by using different simulation draws if B is too small, but computational costs can be great if B is chosen to be extremely large" (Andrews and Buchinsky 2000, p. 23). Because of this, Andrews and Buchinsky (2000) introduced a three-step method to determine B to attain a specified level of accuracy.

- Exact values are always preferred to approximations. There is no point in adding resampling error to sampling error unnecessarily.

- A bootstrapping novice can easily confuse the sample size n and number of bootstrap samples B. Eliminating the resampling of the data set B times simplifies the conceptualization of the bootstrap process.

- In many situations, computer time is saved using the exact approach.

By way of example, this section shows how the `OrderStat` and `RangeStat` algorithms (along with additional APPL procedures) can be used to perform exact bootstrap analysis. The use of these algorithms eliminates the resampling variability that is present in a bootstrap procedure. "The only exact solutions that are available for the bootstrap variance of an L-statistic (e.g., mean, trimmed mean, median, quick estimators of location and scale, upper and lower quartiles) are for the specific cases of the sample mean and sample median (when the sample size is odd)" (Hutson and Ernst 2000, p. 89). The application areas that will be presented here are the estimation of standard errors for the median, mean, and range, and interval estimation for the range.

3.2.1 Estimation of Standard Errors

The standard error of the sample mean, $s/\sqrt{n}$, is useful when comparing means, but standard errors for comparing other quantities (e.g., fractiles) are often intractable. The following examples consider the estimation of standard errors associated with the rat-survival data given in Table 1 (Efron and Tibshirani 1993, p. 11). Seven rats are given a treatment and their survival times, given in days, are shown in the first row of the table. Nine other rats constitute a control group, and their survival times are shown in the second row of the table.

Example 1. Comparing Medians. Consider first the estimation of the standard error of the difference between the medians of the two samples. The standard bootstrap approach

Table 1: Rat-Survival Data

Group	Data	n	Median	Mean	Range
Treatment	16, 23, 38, 94, 99, 141, 197	7	94	86.86	181
Control	10, 27, 30, 40, 46, 51, 52, 104, 146	9	46	56.22	136

Table 2: Bootstrap Estimates of the Standard Error of the Median

	$B = 50$	$B = 100$	$B = 250$	$B = 500$	$B = 1000$	$B = +\infty$
Treatment	41.18	37.63	36.88	37.90	38.98	37.83
Control	20.30	12.68	9.538	13.10	13.82	13.08

to estimating the standard error of the median for the treatment group is to generate B bootstrap samples, each of which consists of seven samples drawn with replacement from 16, 23, 38, 94, 99, 141, and 197. The sample standard deviation of the medians of these B bootstrap samples is an estimate of the standard error of the median. Using the S-Plus commands

```
set.seed(1)
tr <- c(16, 23, 38, 94, 99, 141, 197)
medn <- function(x){quantile(x, 0.50)}
bootstrap(tr, medn, B = 50)
```

yields an estimated standard error of 41.18 for the treatment data with $B = 50$ bootstrap replicates. With the `set.seed` function used to call a stream number corresponding to the associated column, Table 2 shows the estimated standard errors for several B values.

There is considerable resampling error introduced for smaller values of B. The $B = +\infty$ column of Table 2 corresponds to the *ideal bootstrap estimate of the standard error* of $\hat{\theta}$, or $\mathrm{se}_{\hat{F}}(\hat{\theta}^*) = \lim_{B\to+\infty} \hat{\mathrm{se}}_B$, to use the terminology and notation in Efron and Tibshirani (1993, p. 46).

An additional APPL procedure was constructed, `BootstrapRV(data)` (henceforth referred to as `BootstrapRV`), whose only argument is a list of n data values. The `BootstrapRV` procedure creates a discrete random variable X that can assume the provided data values, each with probability $\frac{1}{n}$. If we put the treatment data values 16, 23, 38, 94, 99, 141, and 197 in a list and provide this list to the `BootstrapRV` procedure, it creates a discrete random variable X that can assume these data values each with probability $\frac{1}{7}$.

Assign the random variable Y to the distribution of the fourth order statistic (the median) in seven draws with replacement from the rat-treatment data. This is done with the APPL

Table 3: Bootstrap Estimates of the Standard Error of the Mean

	$B = 50$	$B = 100$	$B = 250$	$B = 500$	$B = 1000$	$B = +\infty$
Treatment	23.89	24.29	23.16	24.36	23.75	23.36
Control	17.07	13.83	13.40	13.13	13.55	13.35

statement `Y := OrderStat(X, 7, 4)`. The distribution of the random variable Y is

$$f(y) = \begin{cases} \frac{8359}{823543} & y = 16 \\ \frac{80809}{823543} & y = 23 \\ \frac{196519}{823543} & y = 38 \\ \frac{252169}{823543} & y = 94 \\ \frac{196519}{823543} & y = 99 \\ \frac{80809}{823543} & y = 141 \\ \frac{8359}{823543} & y = 197. \end{cases}$$

Taking the square root of the variance of Y (which can be done in APPL) returns the standard error as $\frac{2}{823543}\sqrt{242712738519382} \cong 37.83467$. In a similar fashion, the *ideal bootstrap estimate of the standard error* of the median can be calculated in the control case and is $\frac{1}{387420489}\sqrt{25662937134123797402} \cong 13.07587$. Finally, note that the seemingly large difference between the two sample medians ($94 - 46 = 48$) is only $48/\sqrt{37.83^2 + 13.08^2} \cong 1.19$ standard-deviation units away from zero, indicating that the observed difference in the medians is not statistically significant. Had the standard bootstrap procedure been applied with $B = 50$ bootstrap replications, Table 2 indicates that the number of standard-deviation units would have been estimated to be $48/\sqrt{41.18^2 + 20.30^2} \cong 1.05$. Although the conclusion in this case is the same, the difference between using $B = 50$ and $B = +\infty$ could result in different conclusions for the same data set.

Example 2. Comparing Means. Although the standard error of the mean is tractable, we continue with the previous analysis and attempt to compare the sample means to illustrate how additional APPL procedures are used for comparing means. As before, S-Plus can be used to create bootstrap estimates given in Table 3. The code

```
set.seed(1)
x <- c(16, 23, 38, 94, 99, 141, 197)
bootstrap(x, mean, B = 50)
```

produces the upper-left-hand entry in Table 3.

After the bootstrap random variable X is created as in the previous example, the APPL procedure `ConvolutionIID` sums $n = 7$ of these iid X's to create a convoluted random

variable W. The APPL procedure `Transform` transforms the random variable W using the transformation $Y = W/n$. These manipulations yield the probability density function of the mean Y as

$$f(y) = \begin{cases} 1/7^7 = 1/823543 & y = 16 \\ \binom{7}{1}/7^7 = 1/117649 & y = 17 \\ \binom{7}{2}/7^7 = 3/117649 & y = 18 \\ \binom{7}{3}/7^7 = 5/117649 & y = 19 \\ \binom{7}{1}/7^7 = 1/117649 & y = 134/7 \\ \binom{7}{4}/7^7 = 5/117649 & y = 20 \\ \vdots & \vdots \\ 1/7^7 = 1/823543 & y = 197, \end{cases}$$

and the standard error as $\frac{2}{49}\sqrt{327649}$, or approximately 23.36352. This, of course, is equal to $\sqrt{\frac{n-1}{n}}\,\frac{s}{\sqrt{n}} = \sqrt{\frac{6}{7}}\,\frac{s}{\sqrt{7}}$, where s is the standard deviation of the treatment survival times. This fact is the fortunate consequence of the mathematical tractability of the standard error for means. Other, less fortunate, situations can be handled in a similar manner.

Similar APPL code for the treatment case yields an estimated standard error in the $B = +\infty$ case of $\frac{1}{27}\sqrt{129902}$, or approximately 13.34886. The difference between the treatment and control group means ($86.86 - 56.22 = 30.64$) is not statistically significant since it is only $30.64/\sqrt{23.36^2 + 13.35^2} \cong 1.14$ standard deviations away from zero.

Example 3. Comparing Ranges. The previous two examples have estimated the standard errors of measures of central tendency (e.g., the median and mean). Estimation of the standard error of a measure of dispersion, the sample range R, will now be considered. The standard error of the range R for the treatment case is $\frac{2}{16807}\sqrt{88781509983}$, or approximately 35.45692, which is obtained using the `BootstrapRV` and `RangeStat` procedures. Similar APPL statements for the control case yield an estimated standard error for the range as $\frac{1}{9}\sqrt{129902}$, or approximately 40.04658.

The observed difference in the ranges between the treatment and control groups ($181 - 136 = 45$) is not statistically significant since it is only $45/\sqrt{35.46^2 + 40.05^2} \cong 0.84$ standard deviations away from zero.

3.2.2 Confidence Intervals

A final example is presented here to show how one of the shortcomings of the standard bootstrap approach can be overcome by using the parametric bootstrap. APPL is used in both settings to eliminate resampling error.

Example 4. Confidence Interval for Range. The previous three examples estimated the standard errors of measures of central tendency and a measure of dispersion. This example constructs a confidence interval for the sample range of the rat-treatment group.

Let R be the range of the $n = 7$ observations. The APPL procedures `BootstrapRV`, `RangeStat`, and `IDF` (a procedure that determines inverse distribution functions or values) produces the 95% confidence interval $76 < R < 181$. This confidence interval has the unappealing property that the point estimator, $\hat{R} = 181$, is also the upper limit of the confidence interval.

Trosset (2001) suggested an alternative method for computing a confidence interval for the range R, which involves parametric bootstrapping. First, an exponential distribution with mean $1/\theta$ is fit to the treatment group data using the APPL `MLE` (maximum likelihood estimator) procedure. The procedure identifies the parameter estimate for the distribution as $\hat{\theta} = \frac{7}{608}$. The (continuous) distribution of the sample range of $n = 7$ observations drawn from an exponential population with parameter $\hat{\theta} = \frac{7}{608}$ is then computed. The 95% confidence interval for the range R is $68 < R < 475$.

4. Discussion

An asset of the `OrderStat` algorithm is that it can be used to explore the properties of order statistics of various distributions, such as the geometric distribution. Arnold et al. (1992, p. 52) show that the distribution of the sample minimum from a geometric distribution with PDF $f(x) = p\,(1-p)^x$ for $x = 0, 1, \ldots$ is geometric with parameter $1-(1-p)^n$. That is, the geometric distribution and the sample minimum from the geometric distribution come from the same family of distributions. Although the `OrderStat` algorithm will not allow us to see this general result in terms of n, it does allow us to verify this conclusion for increasing values of n starting at $n = 2$. The algorithm is a tool for learning about and exploring properties of order statistics. The properties and results can be verified by other means, such as simulation or analytical methods (in some cases), if so desired. Other useful aspects of the algorithm were explored in the applications sections. Below we conclude with some timing results and measurement performances.

We have done timing experiments to determine the practical limitations of using APPL to perform exact bootstrapping. The statistic under consideration is the determining factor as to whether or not one would use the APPL procedures discussed in this paper. Table 4

Table 4: Exact Bootstrap Method Times (in Seconds)

	Rats	Bearings	Presidents
Sample size	7	23	35
Median	0.019	0.549	2.224
Mean	0.039	–	250.610
Range	0.010	0.771	3.585

contains the time (in seconds on a 848 MHz, Intel Pentium III processor laptop machine) for APPL to determine the exact standard error of the median, mean, and range for three different data sets. The first data set is the rat-treatment group discussed in the applications section, and the second two data sets are from Hand et al. (1994) with sample sizes of $n = 23$ and $n = 35$. They are

$$\begin{aligned} \text{BallBearings} &= [17.88, 28.92, 33.00, 41.52, 42.12, 45.60, 48.48, 51.84, 51.96, 54.12, \\ &\quad 55.56, 67.80, 68.64, 68.64, 68.88, 84.12, 93.12, 98.64, 105.12, \\ &\quad 105.84, 127.92, 128.04, 173.40], \end{aligned}$$

and

$$\begin{aligned} \text{Presidents} &= [10, 29, 26, 28, 15, 23, 17, 25, 0, 20, 4, 1, 24, 16, 12, 4, 10, 17, 16, 0, 7, \\ &\quad 24, 12, 4, 18, 21, 11, 2, 9, 36, 12, 28, 3, 16, 9]. \end{aligned}$$

The disproportionate time for computing the standard error of the mean for the Presidents data set is due to the necessary creation of the PDF of the convolution of 35 iid `BootstrapRV(Presidents)` random variables. All APPL procedures only accept discrete random variables with ordered supports. Creating a convolution of more than 10 iid random variables in APPL causes time delays because of sorting algorithm time requirements (based on the number of iid random variables) and data formats (e.g., integer, rational, nonpositive). Although we used an APPL procedure for creating moment generating functions (MGFs) to create PDFs for the convolutions of iid `BootstrapRV`'s in the rat survival times and Presidents data sets, we were unable to employ this method with the BallBearings data set because of its noninteger data values. The original APPL (non-MGF) convolution code `ConvolutionIID` can determine the PDF of at most nine iid discrete random variables, regardless of the format of the support (e.g., noninteger and nonpositive). Both convolution procedures have their limitations, and those limitations are met with the BallBearings data set. Fortunately, theoretical results exist for computing the standard error of the mean. The dash in Table 4 denotes that the standard error of the mean could not be obtained in the BallBearings example; the time required (8+ hours) to determine the MGF of the

Table 5: Standard Errors of the Median for Various Distributions and Sample Sizes

Sample size		Normal(0, 1)	Uniform$(-1, 1)$ (short tailed)	T(5) (heavy tailed)	Gamma(1, 1) (slightly skewed)	Gamma(1, 0.1) (heavily skewed)
	APPL	0.4855290730	0.2351964343	0.7629798392	0.8283462971	0.02503762460
5	$B = 200$	0.4859	0.2290	0.7576	0.8311	0.02516
	$B = 500$	0.4863	0.2325	0.7617	0.8280	0.02511
	$B = 1000$	0.4845	0.2332	0.7621	0.8271	0.02509
	APPL	0.2683256885	0.2032509315	0.6399134179	0.4015447534	0.002422313289
15	$B = 200$	0.2689	0.2047	0.6447	0.4011	0.002361
	$B = 500$	0.2684	0.2052	0.6398	0.3998	0.002156
	$B = 1000$	0.2687	0.2038	0.6367	0.4013	0.002345
	APPL	0.1731572056	0.1923136966	0.1288715334	0.5391931046	0.004471352237
25	$B = 200$	0.1710	0.1937	0.1270	0.5383	0.003225
	$B = 500$	0.1722	0.1926	0.1289	0.5395	0.003584
	$B = 1000$	0.1731	0.1923	0.1290	0.5396	0.003762
	APPL	0.1574494382	0.2003765627	0.2340377215	0.1447994448	0.01427190012
35	$B = 200$	0.1568	0.2004	0.2335	0.1452	0.01401
	$B = 500$	0.1562	0.2008	0.2309	0.1450	0.01425
	$B = 1000$	0.1571	0.2004	0.2348	0.1444	0.01432
	APPL	0.1678335148	0.1401358740	0.2271996278	0.1300985538	0.0007586864666
45	$B = 200$	0.1660	0.1402	0.2261	0.1281	0.0006130
	$B = 500$	0.1675	0.1404	0.2274	0.1299	0.0007100
	$B = 1000$	0.1676	0.1401	0.2271	0.1300	0.0006574
	APPL	0.1183004083	0.1078244912	0.1854762190	0.1324662485	0.001183628210
55	$B = 200$	0.1175	0.1081	0.1856	0.1318	0.001190
	$B = 500$	0.1177	0.1071	0.1851	0.1327	0.001185
	$B = 1000$	0.1179	0.1072	0.1855	0.1324	0.001186
	APPL	0.1623131704	0.1041300347	0.1915669587	0.07657912379	0.0008932239966
75	$B = 200$	0.1627	0.1036	0.1921	0.07565	0.0008918
	$B = 500$	0.1625	0.1042	0.1909	0.07671	0.0008864
	$B = 1000$	0.1624	0.1042	0.1910	0.07641	0.0008984
	APPL	0.1272179181	0.1320419450	0.08651961726	0.06644847628	0.001396997097
95	$B = 200$	0.1259	0.1319	0.08662	0.06629	0.001427
	$B = 500$	0.1267	0.1324	0.08616	0.06634	0.001390
	$B = 1000$	0.1273	0.1323	0.08685	0.06630	0.001412

convolution of 23 of the `BootstrapRV(BallBearings)` random variables exceeded practical expectations.

The time required to compute the bootstrap estimates for $B = 200$ bootstrap replications for the standard error in these cases in S-Plus is less than a second. The trade-off in using APPL procedures versus a preexisting software package with bootstrapping is precision versus computer time. To further investigate the accuracy of the standard error of the median computed by S-Plus versus the APPL procedure, simulated data from a variety of distributions based on symmetry and shape were created. A single data set was generated from the appropriate distribution for each sample size n. The standard error of the median for each of the simulated data sets appears in Table 5. (Note that the PDF of the gamma

Table 6: Median Relative Errors for S-Plus Approximations vs. Exact APPL Values from the 40 Cells in Table 5

Sample size	Normal(0, 1)	Uniform$(-1, 1)$ (short tailed)	T(5) (heavy tailed)	Gamma(1, 1) (slightly skewed)	Gamma(1, 0.1) (heavily skewed)
5	0.0021	0.0085	0.0011	0.0015	0.0021
15	0.0014	0.0027	0.0050	0.0006	0.0319
25	0.0003	0.0001	0.0010	0.0008	0.1586
35	0.0022	0.0001	0.0033	0.0027	0.0034
45	0.0014	0.0003	0.0004	0.0008	0.1335
55	0.0034	0.0058	0.0001	0.0005	0.0020
75	0.0005	0.0007	0.0030	0.0022	0.0058
95	0.0006	0.0020	0.0038	0.0022	0.0107

distribution with parameters a and b is $f(x) = a\,(ax)^{b-1}e^{-ax}/\Gamma(b)$ for $x \geq 0$.) The table entries are the standard errors of the median as calculated in APPL (which corresponds to $B = +\infty$) and in S-Plus using the `bootstrap` function with $B = 200$, $B = 500$, and $B = 1000$. The numbers in the table associated with the bootstrapping are averages of 100 replications. The only case in which APPL took substantially longer than S-Plus to compute the standard error was when the sample size was $n = 95$ for any of the distributions. To better visually summarize the precision of the results in Table 5, Table 6 displays the relative errors in the S-Plus approximations to the exact APPL value for each individual cell in Table 5 associated with $B = 1000$. As can be seen from Table 6, the relative errors range from as small as 0.0001 to as large as 0.1586. No significant trend is apparent for the various sample sizes, but the relative error is typically highest for the heavily skewed gamma distribution.

In conclusion, for small and moderate sample sizes and test statistics that are tractable for APPL, the exact approach to bootstrapping eliminates resampling error and can even reduce computation time compared to some statistical software packages (such as Minitab) where user-written macros are required to perform bootstrapping.

Appendix

This appendix contains APPL code for solving our examples. The five examples from Section 2 given below consider various sampling mechanisms (without replacement and with replacement) and types of sampling distributions (finite support and infinite support).

Example 1. (*Sampling without replacement; finite support; equally likely probabilities*)

```
X := UniformDiscreteRV(1, 25);
Y := OrderStat(X, 15, 8, "wo");
PDF(Y, 10);
```

Example 2. (*Sampling without replacement; finite support; nonequally likely probabilities*)

```
X := [[p1, p2, p3, p4], [1, 10, 100, 1000], ["Discrete", "PDF"]];
OrderStat(X, 3, 2, "wo");
```

Example 3. (*Sampling without replacement; infinite support*)

```
X := GeometricRV(1 / 2);
OrderStat(X, 2, 1, "wo");
```

Example 4. (*Sampling with replacement; finite support*)

```
X := UniformDiscreteRV(1, 6);
OrderStat(X, 8, 1);
```

Example 5. (*Sampling with replacement; infinite support*)

```
X := GeometricRV(1 / 4);
Y := OrderStat(X, 5, 3);
Mean(Y);
Variance(Y);
```

The three examples from Section 3 given below (rat-survival-treatment case only) consider various statistics that can be used in bootstrapping.

Example 1. Comparing Medians.

```
X := BootstrapRV([16, 23, 38, 94, 99, 141, 197]);
Y := OrderStat(X, 7, 4);
sqrt(Variance(Y));
```

Example 2. Comparing Means.

```
X := BootstrapRV([16, 23, 38, 94, 99, 141, 197]);
W := ConvolutionIID(X, 7);
g := [[x -> x / 7], [-infinity, infinity]];
Y := Transform(W, g);
sqrt(Variance(Y));
```

Example 3. Comparing Ranges.

```
X := BootstrapRV([16, 23, 38, 94, 99, 141, 197]);
Y := RangeStat(X, 7);
sqrt(Variance(Y));
```

Acknowledgments

Diane Evans gratefully acknowledges support from the Clare Boothe Luce Foundation. All authors thank Major Andrew Glen for the original continuous code for order statistics, Dr. Paul Stockmeyer for his discussion and help with the combinatorial algorithms, and the associate editor and referees for their helpful suggestions. Lawrence Leemis gratefully acknowledges a faculty research assignment from The College of William & Mary, which provided time for this work.

References

Andrews, D.W.K., M. Buchinsky. 2002. On the Number of Bootstrap Repetitions for BC_a Confidence Intervals. *Econometric Theory* **18** 962–984.

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

Arnold, B.C., N. Balakrishnan, H.N. Nagaraja. 1992. *A First Course in Order Statistics.* Wiley, New York.

Balakrishnan, N. 1988. Recurrence relations for order statistics from n independent and non-identically distributed random variables. *Ann. Inst. Statist. Math.* **40** 273–277.

Balakrishnan, N., S.M. Bendre, H.J. Malik. 1992. General relations and identities for order statistics from non-independent non-identical variables. *Ann. Inst. Statist. Math.* **44** 177–183.

Balakrishnan, N., C.R. Rao, eds. 1998. *Order Statistics: Applications.* Elsevier, Amsterdam, The Netherlands.

Bapat, R.B., M.I. Beg. 1989. Identities and recurrence relations for order statistics corresponding to non-identically distributed variables. *Comm. Statist. – Theory and Methods* **18** 1993–2004.

Boncelet, C.G. 1987. Algorithms to compute order statistic distributions. *SIAM J. Statist. Comput.* **8** 868–876.

Burr, I.W. 1955. Calculation of exact sampling distribution of ranges from a discrete population. *Ann. Math. Statist.* **26** 530–532. Correction **38** 280.

Cao, G., M. West. 1997. Computing distributions of order statistics. *Comm. Statist.-Theory and Methods* **26** 755–764.

Casella, G., R. Berger. 2002. *Statistical Inference*, 2nd ed. Duxbury Press, Pacific Grove, CA.

Ciardo, G., L. Leemis, D. Nicol. 1995. On the minimum of independent geometrically distributed random variables. *Statistics and Probability Letters* **23** 313–326.

David, H.A., H.N. Nagaraja. 2003. *Order Statistics*, 3rd ed. Wiley, Hoboken, NJ.

Efron, B., R. Tibshirani. 1993. *An Introduction to the Bootstrap.* Chapman & Hall, New York.

Glen, A., L. Leemis, D. Evans. 2001. APPL: A probability programming language. *The American Statistician* **55** 156–166.

Hand, D.J., F. Daly, A.D. Lunn, K.J. McConway, E. Ostrowski, eds. 1994. *A Handbook of Small Data Sets.* Chapman & Hall, London, UK.

Hogg, R.V., A.T. Craig. 1995. *Mathematical Statistics*, 5th ed. Prentice-Hall, Englewood Cliffs, NJ.

Hutson, A.D., M.D. Ernst. 2000. The exact bootstrap mean and variance of an L-estimator. *J. Royal Statist. Soc. B* **62** 89–94.

Margolin, B.H., H.S. Winokur, Jr. 1967. Exact moments of the order statistics of the geometric distribution and their relation to inverse sampling and reliability of redundant systems. *J. Amer. Statist. Assoc.* **62** 915–925.

Nagaraja, H.N., P.K. Sen, D.F. Morrison, eds. 1996. *Statistical Theory and Applications: Papers in Honor of Herbert A. David.* Springer, New York.

Rice, J.A. 1995. *Mathematical Statistics and Data Analysis*, 2nd ed. Wadsworth, Belmont, CA.

Srivastava, R.C. 1974. Two characterizations of the geometric distribution. *J. Amer. Statist. Assoc.* **69** 267–269.

Trosset, M. 2001. Personal communication.

Young, D.H. 1970. The order statistics of the negative binomial distribution. *Biometrika* **57** 181–186.

Young, G.A. 1994. Bootstrap: More than a stab in the dark? *Statistical Science* **9** 382–415.