SYMBOLIC COMPUTATION OF NONPARAMETRIC
BOOTSTRAP ESTIMATORS AND THEIR PROPERTIES

by

George Theodore Sampson

A thesis submitted in conformity with the requirements for the
Degree of Doctor of Philosophy
Graduate Department of Statistics
University of Toronto

© Copyright by George Theodore Sampson 1997
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author’s permission.

L’auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L’auteur conserve la propriété du droit d’auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-28050-0
Symbolic Computation of Nonparametric Bootstrap Estimators and their Properties

George Theodore Sampson
Ph.D. 1997
Department of Statistics
University of Toronto

Abstract

The bootstrap was introduced in 1979 as a computer-intensive method for estimating the standard error and other properties of an estimator. This thesis focuses on how the same objective can be achieved, in many cases, by the development and implementation of computer algorithms for the symbolic computation and evaluation of series expansions. The analytic bootstrap permits the calculation of standard errors for a wide variety of estimators in a fraction of the time and with comparable or better accuracy when compared with conventional bootstrap Monte Carlo resampling. In addition, properties such as bias and variance of bootstrap estimators may be assessed. The methodology is applied to examples from Efron and Tibshirani (1993) involving the correlation coefficient, the Behrens-Fisher statistic, double-bootstrapping and $BC_a$ bootstrap confidence intervals. An alternative methodology based on the exhaustive enumeration of bootstrap resamples is also presented and illustrated for the case of the median. Implications for future work are discussed.
Acknowledgements

I would like to express my gratitude to my supervisor David Andrews for his invaluable guidance, patience, encouragement, enthusiasm and humour during the undertaking of this thesis. It has been a privilege and a pleasure working with him.

My thanks also go to Andrey Feuerverger and Rob Tibshirani for making very useful comments and suggestions on the manuscript. I wish to thank Yodit Seifu for her encouragement and assistance. Ruth Croxford, Jenny Tam, Laura Kerr and Tom Glinos have been helpful with computational and administrative details. Gus Vukov has provided me with many valuable teaching experiences.

I would like to thank all the graduate students, staff and faculty in the Department of Statistics for a meaningful and memorable academic experience.

I am indebted to the Department of Statistics and the University of Toronto for providing me with financial support during the first four years of my graduate studies.

This thesis is dedicated to my parents whose initial encouragement instilled in me a resolve to undertake the present studies.
# Contents

1 Introduction

1.1 Overview .................................. 1
1.2 Literature review ............................. 7
1.3 Thesis outline ................................ 9

2 Symbolic Computation of Bootstrapped Order Statistics and their Moments: The Case of the Median

2.1 Introduction .................................. 11
2.2 Review of Bootstrap Estimation of Standard Error .............................. 12
2.3 Properties of Bootstrap Estimators .......................... 14
    2.3.1 Introduction ................................. 14
    2.3.2 Exact Computation of Bootstrap Estimators ............... 16
    2.3.3 Moments of Bootstrap Estimators .......................... 20
2.4 Application to Order Statistics: The Case of the Median ............... 22
    2.4.1 Some Properties of Order Statistics .................... 23
    2.4.2 The Case of the Median ........................... 28
    2.4.3 Some Numerical Results ....................... 38
2.5 Conclusions and Discussion ................................ 42

3 Bootstrapped M-estimators and their Properties .......................... 44

3.1 Introduction .................................. 44
3.2 Bootstrapping M-estimators: An Analytical Approach .................. 46
3.2.1 Introduction ................................................................. 46
3.2.2 An Approximation for Functions of M-estimators .............. 46
3.2.3 Weighting and Decomposing Residual Sums ...................... 51
3.2.4 Analytical Bootstrap Moments of M-Estimators and their Properties ................................................................. 55

3.3 Computational Tools .......................................................... 57

3.4 The Case of the Correlation Coefficient ................................ 60
3.4.1 Introduction ................................................................. 60
3.4.2 Computational Tools ....................................................... 61
3.4.3 Numerical Results .......................................................... 69

3.5 Discussion and Conclusions ................................................. 71

4 Double-Bootstrapping and the Behrens-Fisher statistic ............ 79

4.1 Introduction ................................................................. 79

4.2 Symbolic Computation of the Bootstrap-t Confidence Interval ................................................................. 79
4.2.1 Background ................................................................. 79
4.2.2 Symbolic Computation of the Bootstrap-t Interval for the Correlation Coefficient ................................................................. 82
4.2.3 Discussion ................................................................. 87

4.3 Symbolic Computation of $BC_a$ Confidence Intervals ............ 87
4.3.1 Introduction ................................................................. 87
4.3.2 Background ................................................................. 88
4.3.3 Symbolic Computation of the Acceleration Constant $\hat{a}$ .... 91
4.3.4 Discussion ................................................................. 96

4.4 Analytic Bootstrap Computation for the Behrens-Fisher statistic ................................................................. 97
4.4.1 Introduction ................................................................. 97
4.4.2 Background ................................................................. 97
4.4.3 Analytic Bootstrap Computation ........................................ 98
4.4.4 Numerical Application ...................................................... 103
4.4.5 Discussion ........................................................................ 104
4.5 Conclusions .......................................................................... 106

5 Concluding remarks .................................................................. 108
5.1 Remarks ................................................................................ 108

A Source Code used for this Thesis .............................................. 111
A.1 Source Code used for Chapter 2 ............................................ 111
A.2 Source Code used for Chapter 3 ............................................ 122
A.3 Source Code used for Chapter 4 ............................................ 143
List of Tables

1.1  Bootstrap estimators and associated properties for a statistic $\hat{\theta}$ for which analytic expressions are derived. ................................................................. 4

2.1  Total number of distinct resamples and number of partitions for various sample sizes ................................................................. 18

2.2  The 5 partitions associated with a sample size of 4 with corresponding multinomial probabilities ................................................................. 19

2.3  The 12 distinct permutations of the partition (2.1.1.0) ............................................. 19

2.4  Approximating distributions for various $\lambda$ values ............................................. 26

2.5  Partitions and multinomial probabilities for a sample size of 3 .................................. 28

2.6  Analytical and simulated results for various properties and bootstrap estimates related to the 2nd moment of the median using several sample sizes for the case of the normal distribution ................................................. 39

2.7  Analytical and simulated results for various properties and bootstrap estimates related to the 2nd moment of the median using several sample sizes for the case of the uniform distribution ................................................. 40

2.8  Analytical and simulated results for various properties and bootstrap estimates related to the 2nd moment of the median using several sample sizes for the case of the logistic distribution ................................................. 41

3.1  Bootstrap estimators and related properties for the correlation coefficient for which analytic expansions are derived ................................................. 67

3.2  $n = 15$ Law School Data: original and standardized data ........................................... 69

3.3  $n = 100$ simulated bivariate observations from $N(0,1)$: variable $x$ ................................ 70
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4</td>
<td>$n = 100$ simulated bivariate observations from $N(0, 1)$: variable $y$</td>
<td>71</td>
</tr>
<tr>
<td>3.5</td>
<td>Symbolic bootstrap and classical results for the correlation coefficient using two data sets</td>
<td>72</td>
</tr>
<tr>
<td>3.6</td>
<td>Monte Carlo Bootstrap Estimates of the Mean of the correlation coefficient for various $B$</td>
<td>73</td>
</tr>
<tr>
<td>3.7</td>
<td>Monte Carlo Bootstrap Estimates of Standard Error of the correlation coefficient for various $B$</td>
<td>74</td>
</tr>
<tr>
<td>4.1</td>
<td>Comparison of 90% confidence intervals for $\rho$ using analytic bootstrap-t, conventional bootstrap-t, and standard methods for the law school data ($n = 15$)</td>
<td>86</td>
</tr>
<tr>
<td>4.2</td>
<td>Bootstrap estimators and related properties for the Behrens-Fisher statistic for which analytic expansions are derived</td>
<td>99</td>
</tr>
<tr>
<td>4.3</td>
<td>Survival times (days) of gastric cancer patients ($n = 44$)</td>
<td>103</td>
</tr>
<tr>
<td>4.4</td>
<td>Symbolic bootstrap and classical results for the Behrens-Fisher statistic using the survival data, $n = 44$</td>
<td>104</td>
</tr>
<tr>
<td>4.5</td>
<td>Monte Carlo Bootstrap Estimates of Mean and Standard Error of the Behrens-Fisher statistic for various $B$ for the survival data, $n = 44$ (s.e. in brackets)</td>
<td>105</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Overview

In 1979 Efron proposed the idea of using computer-based simulations instead of mathematical calculations to obtain the sampling properties of random variables. This thesis focuses on how the same objective can be achieved, in many cases, by the development and implementation of computer algorithms for the symbolic computation and evaluation of asymptotic expansions. Our study is largely confined to a collection of problems that may be treated using classical theory for sums of independent and identically distributed random variables - the so-called "smooth function model" (Hall 1992). The corresponding class of statistics is wide and includes, for example, sample means, variances, ratios and differences of means and variances, correlations, maximum likelihood estimators and M-estimators. But it is by no means comprehensive. It does not include, for example, the sample median or other order statistics: we treat this case in a separate chapter.

The bootstrap was originally introduced as a general method for assessing the statistical accuracy of an estimator. The procedure applies a given estimator \( \hat{\theta} \) to each of B bootstrap samples, and then estimates the standard error of \( \hat{\theta} \) by the empirical standard deviation of the B replications. More formally, suppose that \( \mathbf{x} = (x_1, \ldots, x_n) \) is an observed random sample of size n from a population with distribution function \( F \). We estimate a parameter of interest \( \theta \) by \( \hat{\theta} \) to which we wish to assign a standard
error denoted by

$$se_F(\hat{\theta}) .$$  \hspace{1cm} (1.1)

Now (1.1) may in turn be estimated by

$$se_F(\hat{\theta}) .$$  \hspace{1cm} (1.2)

that is, by replacing $F$ by the empirical distribution function $\hat{F}$ defined to be the discrete distribution that assigns probability $1/n$ to each value $x_i$ for $i = \{1, \ldots, n\}$. When $F$ is unknown or $\hat{\theta}$ is complicated, bootstrap methodology estimates (1.1) by applying the same principle. To this end the bootstrap algorithm for estimating standard errors selects $B$ independent bootstrap samples $x_1^*, \ldots, x_i^*, \ldots, x_B^*$ each consisting of $n$ observations drawn with replacement from $x$. The bootstrap replication $\hat{\theta}^*(b)$ corresponding to each bootstrap sample is evaluated and the standard error $se_F(\hat{\theta})$ is estimated by

$$se_B(\hat{\theta}^*) = \left\{ \frac{1}{B-1} \sum_{b=1}^{B} [\hat{\theta}^*(b) - \hat{\theta}^*(\cdot)]^2 / (B - 1) \right\}^{1/2}$$ \hspace{1cm} (1.3)

where

$$\hat{\theta}^*(\cdot) = \frac{1}{B} \sum_{b=1}^{B} \hat{\theta}^*(b) .$$  \hspace{1cm} (1.4)

We note that (1.3) is an approximation of the ideal (i.e. "infinite") bootstrap estimate of $se_F(\hat{\theta})$ given by

$$\lim_{B \to \infty} se_B(\hat{\theta}^*) = \sqrt{E_F(\hat{\theta}^{*2}) - [E_F(\hat{\theta}^*)]^2} = se_F(\hat{\theta}^*) .$$  \hspace{1cm} (1.5)

Similarly, the bootstrap algorithm uses (1.4) to approximate the ideal bootstrap ex-
CHAPTER 1. INTRODUCTION

Observe that a bootstrap sample $\mathbf{x}^* = (x_1^*, \ldots, x_n^*)$ consists of members of the original data set $\mathbf{x}$, some appearing zero times, some appearing once, some appearing twice, etc. Hence, the bootstrap sample $\mathbf{x}^*$ generates a corresponding set of frequencies $(w_1, \ldots, w_n)$ conditional on the original sample $\mathbf{x}$, that is.

$$\mathbf{x}^* \rightarrow w|\mathbf{x}$$  \hspace{1cm} (1.7)

where $w = (w_1, \ldots, w_n)$ is a multinomial vector of integer weights. In this sense, the ideal bootstrap mean (1.6), for example, may be alternatively expressed notationally as

$$E_{\mathbf{w}|\mathbf{x}}(\hat{\theta}^*) .$$  \hspace{1cm} (1.8)

We shall use this notation throughout this thesis in order to convey (1.7), that is, that bootstrap sampling can be thought of as weighted resampling. Moreover, this notation shall represent ideal bootstrap estimates in the sense of (1.5).

More generally, we may represent a generic bootstrap estimator of $E_F[g(\hat{\theta})]$ by

$$T = E_{\mathbf{w}|\mathbf{x}}[g(\hat{\theta}^*)]$$  \hspace{1cm} (1.9)

for some function $g$. Indeed, when

$$g(\hat{\theta}^*) = \hat{\theta}^*$$

we get (1.8) which may be approximated by (1.4). Further, when

$$g(\hat{\theta}^*) = [\hat{\theta}^* - E_{\mathbf{w}|\mathbf{x}}(\hat{\theta}^*)]^2$$
then (1.9) gives the ideal bootstrap variance estimator

\[
T = E_w|X[\hat{\theta}^* - E_w|X(\hat{\theta}^*)]^2
\]

\[
= Var_w|X(\hat{\theta}^*)
\]

\[
= (se_F(\hat{\theta}^*))^2
\]  

(1.10)

which may be approximated by the square of (1.3). In addition, we may evaluate properties of (1.8) and (1.10) such as the mean \(E_x(\cdot)\) and variance \(Var_x(\cdot)\). Table 1.1 below summarizes the six bootstrap expressions for which representations in terms of analytical expressions comprise the focus of this thesis. These six quantities include the bootstrap mean and the bootstrap variance of an estimator \(\hat{\theta}\) as well as two properties for each of these.

Table 1.1: Bootstrap estimators and associated properties for a statistic \(\hat{\theta}\) for which analytic expressions are derived.

<table>
<thead>
<tr>
<th>Property</th>
<th>Bootstrap Mean ((\hat{\theta}))</th>
<th>Bootstrap Variance ((\hat{\theta}))</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(E_{w</td>
<td>x}(\hat{\theta}^*))</td>
</tr>
<tr>
<td>Mean</td>
<td>(E_x[E_{w</td>
<td>x}(\hat{\theta}^*)])</td>
</tr>
<tr>
<td>Variance</td>
<td>(Var_x[E_{w</td>
<td>x}(\hat{\theta}^*)])</td>
</tr>
</tbody>
</table>

The intent of conventional bootstrap methodology is to estimate, for example,

\[
Var_x(\hat{\theta})
\]  

(1.11)

using the ideal bootstrap variance estimator

\[
Var_{w|x}(\hat{\theta}^*). \tag{1.12}
\]

One motivation for this thesis is to investigate how well the bootstrap procedure (1.12) estimates (1.11). To do this we will evaluate properties of (1.12) such as

\[
E_x[Var_{w|x}(\hat{\theta}^*)]
\]
These properties could be used *diagnostically* to uncover, for example, inherent bias in the bootstrap procedure (1.12) as represented by

$$E_X[\text{Var}_{w|x}(\hat{\theta}^*)] - \text{Var}_X(\hat{\theta}).$$

The objective of this thesis is to develop symbolic computational tools that can be used in practice to derive Monte Carlo-free bootstrap estimates and to help study properties of bootstrap estimators.

We explore *two methods* for the symbolic evaluation of the quantities in Table 1.1. The first approach is based on the exact computation of the ideal bootstrap expectation (1.8) as studied by Fisher and Hall (1991). Their algorithm requires enumeration of all distinct resamples which may be drawn with replacement from the original sample. We develop our method with respect to the class of L-estimators, that is, linear combinations of order statistics. However, the technique may be used for more general estimators. We will focus attention on the sample median, a statistic whose distribution is known exactly: this will facilitate illustration of the appropriate symbolic tools. The second approach involves the symbolic derivation of a bootstrap moment and its properties for statistics belonging to the class of M-estimators, that is, for any statistic $\hat{\theta}$ assumed to be the solution of the estimating equation

$$\tilde{\psi}_n(\theta) \equiv \frac{1}{n} \sum_{i=1}^{n} \psi(x_i, \theta) = 0, \quad (1.13)$$

where the $x_i$'s arise from the density $f(x, \theta)$. The method depends primarily on two key symbolic routines - one for computing asymptotic expansions of roots of estimating equations, and one for computing bootstrap versions of these expansions. The algorithms have been implemented using a computer-algebraic manipulation package called *Mathematica* (Wolfram, 1988). The methodology is applied to examples
from Efron and Tibshirani (1993) involving the correlation coefficient, the Behrens-Fisher statistic, double-bootstrapping and $BC_a$ bootstrap confidence intervals.

Finally, we summarize the features/advantages of the analytic bootstrap developed in this thesis as follows:

1. Analytic derivation and numerical evaluation of bootstrap estimates and their properties entirely avoids Monte Carlo resampling resulting in faster and more efficient calculation.

2. It can be used diagnostically to assess properties such as bias, variance or other moments of a bootstrap procedure.

3. For the class of L-estimators (linear combinations of order statistics), the analytic bootstrap calculates ideal bootstrap estimators exactly for sample sizes up to $n = 20$ and calculates properties of these estimators exactly for much larger sample sizes.

For the class of M-estimators (maximum likelihood-type estimators), the analytic bootstrap calculates ideal bootstrap estimators and properties of these exactly for linear statistics and to within a specified order of accuracy for large sample sizes in the case of non-linear statistics.

4. The analytic bootstrap may be applied to a broad class of statistics, namely, to any estimator which may be expressed as a non-linear smooth function of averages.

5. The analytical approach for the class of M-estimators is less computer intensive for larger data sets. The reverse is true for conventional bootstrap calculations.

6. Analytic bootstrap expressions are potentially insightful because component statistical quantities such as cumulants and sums are identified.

7. Each analytic bootstrap expression (representing a bootstrap moment or a property of this) is a mathematical bootstrap formula which may be further numerically evaluated efficiently for a given data set, or repeatedly evaluated for
different data sets arising within a given application. The computational effort required to derive each formula need only be undertaken once. It is subsequently useable in perpetuity by other practitioners for the purpose of efficient numerical evaluation.

This thesis does not discuss the analytic bootstrap in the context of parametric bootstrap estimation, kernel estimation or data which are not independent and identically distributed. These could be topics for future research.

1.2 Literature review

This thesis represents the marriage between two technologies, symbolic computation and bootstrap estimation, and a broad class of statistics known as maximum likelihood type estimators, or M-estimators. In the literature these three areas have largely been developed independently of each other.

Efron's (1979) article has spawned an entire literature on the topic of the bootstrap. The monograph by Efron and Tibshirani (1993) presents an intuitive account of the bootstrap and its diverse applications in statistical inference. It contains a comprehensive list of references on the subject. The lecture notes of Beran and Ducharme (1991) and Hall's (1992) monograph give a mathematical treatment of the bootstrap. In particular, Fisher and Hall (1991) discuss the exact computation of nonparametric bootstrap estimators using exhaustive enumeration of distinct resamples which may be drawn with replacement from the original sample. We use a related approach in Chapter 2 to develop the symbolic computation of bootstrap estimators and corresponding properties connected with the class of L-estimators. Oldford (1985) considers analytic evaluation of bootstrap estimates of bias and variance for linear and quadratic approximations of an estimator. This forms a basis for Chapter 3 which is developed using a generalized methodology found in Andrews and Feuerverger (1993).

Huber (1964, 1967) considered the class of maximum likelihood type estimators $\hat{\theta}$ that solve estimating equations such as (1.13). A detailed account of the theory of M-estimators can be found in Huber (1981). Andrews and Feuerverger (1993) show
how a single arbitrary M-estimator, or a smooth function of such estimators, may be expressed as a particular sequence of increasingly accurate Taylor series approximations. Alternative presentations may also be found in Ferguson (1989) and McCullagh (1987). This approach is used in Chapter 3 as the basis for the development of algorithms permitting the symbolic representation of nonparametric bootstrap moments (and respective properties) associated with M-estimators.

In this thesis a set of general procedures for the symbolic derivation of bootstrap quantities is presented. One of the two contexts considered involves the use of asymptotic expansions. The derivation of asymptotic expansions by hand is typically a laborious task, it can be extremely time consuming and can involve exceedingly complicated algebra. The potential for clerical error is high. Symbolic computation provides a practical alternative to doing these expansions by hand - the development of appropriate computer-algebraic tools permits the statistician to derive the expansions correctly and reliably, in a fraction of the time taken by hand, and to efficiently evaluate the derived formulae in specific cases. Expansions are obtained quickly without the chance for human clerical error. This approach liberates statisticians from the tedious details of asymptotic calculation and extends their potential capabilities. Expansions derived by computer may moreover be saved in a form suitable for publication and then inserted directly into text, further reducing the chance for errors in publications. Programming errors can be reduced by testing software against known statistical results; the software can then be used to derive new results. The procedures developed in this thesis have been tested against simulated results documented in, for example, Efron and Tibshirani (1993), and against independent results obtained from the use of code written in Splus.

Symbolic computation is a powerful facility available to research statisticians. The use of this technology in statistics is relatively new, its history extending over little more than a decade. Packages like Mathematica, Maple, Macsyma and Reduce are being used increasingly in areas of statistical theory where the algebra may be otherwise prohibitively complex. There are numerous books that describe a variety of useful applications of symbolic computation in statistics - see, for example,

1.3 Thesis outline

The thesis is organized as follows:

Chapter 2 presents algorithms for the symbolic computation of properties of non-parametric bootstrap estimators involving order statistics. The case of the median is considered in detail. Comparisons are made with Monte Carlo simulations.

In Chapter 3 symbolic procedures are developed and implemented for the analytical representation of bootstrap estimators and their properties with respect to statistics defined by estimating equations and functions of these. The technique is illustrated for the case of the correlation coefficient. Comparisons with conventionally computed bootstrap estimates are made.

The methodology is further illustrated in Chapter 4 with respect to the Behrens-
Fisher statistic, double-bootstrapping and $BC_a$ bootstrap confidence intervals.

Finally, Chapter 5 provides concluding remarks and suggestions for future research.
Chapter 2

Symbolic Computation of Bootstrapped Order Statistics and their Moments: The Case of the Median

2.1 Introduction

In this chapter we seek to compute properties of nonparametric bootstrap estimators involving order statistics exactly using symbolic computation. In particular, the method permits evaluation of the bias and the mean square error of such estimators analytically. The resulting expressions may be efficiently evaluated numerically. Such calculations require enumeration of all possible resamples which may be drawn with replacement from the original sample and this will be shown to be feasible for moderate sample sizes.

Section 2 reviews standard bootstrap estimation. In Section 3 we first review an algorithm given in Fisher and Hall (1991) which describes the exact computation of nonparametric bootstrap estimators in small samples. We propose an extension to this procedure for the exact computation of moments of nonparametric bootstrap
estimators in much larger samples. In Section 4 this methodology is illustrated by evaluating moments associated with bootstrapping order statistics, in particular, the median. This is achieved through the use of the $\lambda$-family of distributions that permits expression of moments of order statistics in closed form for a variety of distributional shapes. These results are compared with Monte Carlo simulations. Various algorithms developed in the chapter are also described. Conclusions are discussed in Section 5.

2.2 Review of Bootstrap Estimation of Standard Error

Suppose that $x = (x_1, \ldots, x_n)$ is an observed random sample of size $n$ from a population with distribution function $F$ and we wish to estimate the parameter of interest expressed as some function $t$ of $F$, on the basis of $x$. For this purpose we calculate an estimate of interest

$$\hat{\theta} = s(x).$$

(2.2)

to which we wish to assign a standard error denoted by

$$se_F(\hat{\theta}) = \left[Var(\hat{\theta})\right]^{1/2}.$$  

(2.3)

(The hat symbol "\^" always indicates quantities calculated from the observed data.) Now (2.3) may in turn be estimated by

$$se_F(\hat{\theta}) = \left[Var(\hat{\theta})\right]^{1/2},$$

(2.4)

that is, by replacing $F$ with the empirical distribution function $\hat{F}$ defined to be the discrete distribution that assigns probability $1/n$ to each value $x_i$ for $i = \{1, \ldots, n\}$. 


When $F$ is unknown, or $\hat{\theta}$ is complicated, bootstrap methodology attempts to estimate (2.3) by applying this same principle. The bootstrap algorithm for estimating standard errors consists of several steps which are given as follows. We use notation similar to that used by Efron and Tibshirani (1993, Chapter 6). First, $B$ independent bootstrap samples $x_1^*, \ldots, x_B^*$ are selected, each consisting of $n$ observations drawn with replacement from $x$. Each bootstrap sample, generically represented as

$$x^* = (x_1^*, \ldots, x_n^*), \quad (2.5)$$

is a randomized, or resampled, version of the original data set $x$, and hence consists of members of $x$, some appearing zero times, some appearing once, some appearing twice, etc. Hence, the bootstrap sample $x^*$ generates a corresponding set of frequencies $(w_1, \ldots, w_n)$ conditional on the original sample $x$, that is,

$$x^* \longrightarrow w|x \quad (2.6)$$

where $w = (w_1, \ldots, w_n)$ is a multinomial vector of integer weights. The bootstrap replication corresponding to each bootstrap sample is then evaluated as

$$\hat{\theta}^*(b) = s(x_i^*), \quad b = 1, 2, \ldots, B. \quad (2.7)$$

Finally, the bootstrap estimate of the standard error of $\hat{\theta}$ is calculated as the sample standard deviation of the $B$ replications

$$\hat{se}_B(\hat{\theta}^*) = \left\{ \sum_{b=1}^{B} [\hat{\theta}^*(b) - \hat{\theta}^*(\cdot)]^2 / (B - 1) \right\}^{1/2} = [\text{Var}_F(\hat{\theta}^*)]^{1/2}. \quad (2.8)$$

where $\hat{\theta}^*(\cdot) = \sum_{b=1}^{B} \hat{\theta}^*(b) / B$. The limit of $\hat{se}_B(\hat{\theta}^*) = \hat{se}_F$ as $B \rightarrow \infty$ is called the ideal bootstrap estimate of $se_F(\hat{\theta})$, that is,

$$\lim_{B \rightarrow \infty} \hat{se}_B(\hat{\theta}^*) = se_F(\hat{\theta}^*) = [\text{Var}_F(\hat{\theta}^*)]^{1/2}. \quad (2.9)$$
The bootstrap algorithm described above is a computational method for obtaining an approximation to the numerical value of \( se_F(\hat{\theta}^*) \). This ideal bootstrap estimator and its empirical approximation \( se_B \) are sometimes referred to as nonparametric bootstrap estimates because they are based on \( \hat{F} \), the nonparametric estimate of the population \( F \). Parametric bootstrap estimation uses a different estimate of \( F \) but this will not be discussed in this thesis.

Expressing (2.3), (2.4), (2.9) and (2.8) in terms of variances we have, respectively.

\[
Var(\hat{\theta}) = [se_F(\hat{\theta})]^2
\]

\[
\hat{Var}(\hat{\theta}) = [se_F(\hat{\theta})]^2
\]

\[
Var_F(\hat{\theta}^*) = [se_F(\hat{\theta}^*)]^2
\]

\[
\hat{Var}_F(\hat{\theta}^*) = [se_B(\hat{\theta}^*)]^2.
\]

The quantities \( \hat{Var}_F(\hat{\theta}^*) \) and \( Var_F(\hat{\theta}^*) \) may be made arbitrarily close for large enough \( B \) and hence will be assumed to be effectively equal in this discussion.

In the next section we will develop a diagnostic tool that will enable us to calculate bias and other properties of the ideal bootstrap variance estimator \( Var_F(\hat{\theta}^*) \) and the method will be illustrated in the case of order statistics. This methodology is capable of giving analytical or numerical results and is free of Monte Carlo resampling.

### 2.3 Properties of Bootstrap Estimators

#### 2.3.1 Introduction

In this section we propose a procedure for the exact computation of moments of nonparametric bootstrap estimators. We first review an algorithm given in Fisher and Hall (1991) which describes the exact computation of nonparametric bootstrap estimators in small samples.
As was discussed in Chapter 1 we can express the bootstrap sample as

\[ x^* \rightarrow w|x \]  

(2.14)

and the bootstrap statistic \( \hat{\theta}^* \) as

\[ \hat{\theta}^* = \hat{\theta}(w|x) \]

(2.15)

where \( w \) is a multinomial vector of integer weights. Since bootstrap sampling is identical with weighted resampling we may represent a generic bootstrap estimator as

\[ T = E_{w|x}[g(\hat{\theta}^*)] \]

(2.16)

for some function \( g \). Indeed, when

\[ g(\hat{\theta}^*) = [\hat{\theta}^* - E_{w|x}(\hat{\theta}^*)]^2 \]

(2.17)

then (2.16) gives the ideal bootstrap variance estimator

\[ T = E_{w|x}[\hat{\theta}^* - E_{w|x}(\hat{\theta}^*)]^2 = VarF(\hat{\theta}^*). \]

(2.18)

The quantity \( E_{w|x}(\hat{\theta}^*) \) in (2.18) may be used to assess the ideal bootstrap estimate of bias of \( \hat{\theta} \) by

\[ bias_F = E_{w|x}(\hat{\theta}^*) - \hat{\theta}. \]

(2.19)

For most statistics that arise in practice, (2.19) must be approximated by Monte Carlo simulation. In particular, the bootstrap estimate of bias based on \( B \) replications is (2.19) with \( \hat{\theta}^*(\cdot) = \sum_{b=1}^{B} \hat{\theta}^*(b)/B \) substituted for \( E_{w|x}(\hat{\theta}^*) \), so that

\[ bias_B = \hat{\theta}^*(\cdot) - \hat{\theta}. \]

(2.20)

Fisher and Hall (1991) show how Monte Carlo resampling may be circumvented by describing an algorithm for the exact computation of nonparametric bootstrap
estimators in small samples. This technique requires enumeration of all distinct resamples which may be drawn with replacement from the original sample, and also calculation of the probability and value of the statistic associated with each resample. We shall review this procedure for the exact calculation of $E_{w|x}(\hat{\theta}^*)$ in the next sub-section.

2.3.2 Exact Computation of Bootstrap Estimators

Suppose that we wish to calculate the exact value of the ideal bootstrap estimator of the mean of a statistic $\hat{\theta}$ given by

$$E_{w|x}(\hat{\theta}^*).$$


If the sample $x$ is of size $n$, and if all elements of $x$ are distinct, then the number of different possible resamples $x^*$ of size $n$ equals the number, $N(n)$, of distinct ways of placing $n$ indistinguishable objects into $n$ numbered boxes, the boxes being allowed to contain any number of objects. Indeed, letting $k_i$ denote the number of times $x_i$ is repeated in $x^*$ then the number of different possible resamples equals the number of ways of choosing nonnegative integers $k = (k_1, \ldots, k_n)$ satisfying $k_1 + \ldots + k_n = n$. This is given by

$$N(n) = \binom{2n - 1}{n}. \tag{2.21}$$

See Hall (1992, Appendix 1) for a proof of (2.21).

Let $W(n)$ denote the set of all distinct integer $n$-tuples $w = (w_1, \ldots, w_n)$, henceforth called partitions, such that $w_1 \geq w_2 \geq \ldots \geq w_n \geq 0$ and $\sum w_i = n$. Given $(w_1, \ldots, w_n) \in W(n)$, let $M(w_1, \ldots, w_n)$ be the set of all ordered $n$-tuples $k = (k_1, \ldots, k_n)$ which are simply permutations of $(w_1, \ldots, w_n)$. Defining

$$K(n) = \bigcup_{w \in W(n)} M(w_1, \ldots, w_n), \tag{2.22}$$
then \( K(n) \) contains precisely \( N(n) \) elements given by (2.21). The probability of obtaining a specific resample \( x^* \) based on the \( n \)-tuple \( k = (k_1, \ldots, k_n) \) is the multinomial probability \( p(k) \) given by

\[
p(k) = \binom{n}{k_1, \ldots, k_n} (1/n)^{k_1} \ldots (1/n)^{k_n} = \frac{n!}{k_1! \ldots k_n! n^n}
\]  

(2.23)

with the \( k_i \geq 0 \) as above. Furthermore, let \( \hat{\theta}^*(k) \) denote the calculated bootstrap replication of \( \hat{\theta} \) corresponding to the multinomial \( n \)-tuple \( k \). For example, if \( k = (1, 2, 0, 1) \) in the case \( n = 4 \) then the bootstrap replication \( \hat{\theta}^* \) is correspondingly calculated from the resample \( x^* = (x_1, x_2, x_2, x_4) \). Then the ideal bootstrap estimator of the mean of \( \hat{\theta} \) may be computed exactly and directly as the weighted average of the \( N(n) \) bootstrap values \( \hat{\theta}^*(k) \):

\[
E_{\mathbf{w}|\mathbf{x}}(\hat{\theta}^*) = \sum_{k \in K(n)} p(k) \hat{\theta}^*(k)
\]  

(2.24)

or alternatively,

\[
E_{\mathbf{w}|\mathbf{x}}(\hat{\theta}^*) = \frac{(n-1)!}{n^{n-1}} \sum_{\mathbf{w} \in W(n)} \frac{1}{w_1! \ldots w_n!} \sum_{k \in M(w)} \hat{\theta}^*(k).
\]  

(2.25)

Equation (2.25) is computationally more efficient than (2.24). For example, for \( n = 4 \) (2.25) is 7 times more computationally efficient than (2.24).

If the sequence \( \mathbf{w} = (w_1, \ldots, w_n) \) consists of precisely \( r_i \) \( j_i \)'s for \( 1 \leq i \leq q \), where the \( j_i \)'s are distinct and the \( r_i \)'s sum to \( n \), then \( M(\mathbf{w}) \), that is, the second summation in (2.25), contains exactly

\[
L(\mathbf{w}) = \frac{n!}{r_1! \ldots r_q!}
\]  

(2.26)

elements. Table 2.1 lists respective values of \( N(n) \) and the number of partitions \( \mathbf{w} \) contained in \( W(n) \) for various sample sizes. We consider the example below.
Table 2.1: Total number of distinct resamples and number of partitions for various sample sizes

<table>
<thead>
<tr>
<th>n</th>
<th>N(n)</th>
<th>Number of Partitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>35</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>126</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>462</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>1,716</td>
<td>15</td>
</tr>
<tr>
<td>8</td>
<td>6,435</td>
<td>22</td>
</tr>
<tr>
<td>9</td>
<td>24,310</td>
<td>30</td>
</tr>
<tr>
<td>10</td>
<td>92,378</td>
<td>42</td>
</tr>
<tr>
<td>12</td>
<td>1,352,078</td>
<td>77</td>
</tr>
<tr>
<td>15</td>
<td>7.8 × 10⁷</td>
<td>176</td>
</tr>
<tr>
<td>20</td>
<td>6.9 × 10¹⁰</td>
<td>627</td>
</tr>
<tr>
<td>25</td>
<td>6.3 × 10¹³</td>
<td>1958</td>
</tr>
<tr>
<td>30</td>
<td>5.9 × 10¹⁶</td>
<td>5604</td>
</tr>
<tr>
<td>40</td>
<td>5.3 × 10²²</td>
<td>37338</td>
</tr>
</tbody>
</table>

Example 1

We consider in detail the case for n = 4. Table 2.2 lists the 5 partitions comprising $W(4)$ with the respective number of distinct permutations $L(w)$, the simple multinomial probabilities $p(k)$, and the probabilities $p(w)$ of the partitions. The probabilities $p(w)$ will be discussed more fully in the next section.

The $N(2,1,1,0) = 12$ distinct permutations $k$ contained in $M(2,1,1,0)$, for example, are listed in Table 2.3 below:

The ideal bootstrap estimator (2.25) may now be expressed as

$$
E_{w|x}(\hat{\theta}^*) = 3!4^{-3}[(4!)^{-1} \sum_{k \in M(4,0,0,0)} \hat{\theta}^*(k) + (3!1!)^{-1} \sum_{k \in M(3,1,0,0)} \hat{\theta}^*(k) + (2!2!)^{-1} \sum_{k \in M(2,2,0,0)} \hat{\theta}^*(k) + (2!1!)^{-1} \sum_{k \in M(2,1,1,0)} \hat{\theta}^*(k)]
$$
Table 2.2: The 5 partitions associated with a sample size of 4 with corresponding multinomial probabilities

<table>
<thead>
<tr>
<th>( n = 4 )</th>
<th>( W(4) )</th>
<th>( L(w) )</th>
<th>( p(k) )</th>
<th>( p(w) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4,0,0,0)</td>
<td>4</td>
<td>0.00390625</td>
<td>0.015625</td>
<td></td>
</tr>
<tr>
<td>(3,1,0,0)</td>
<td>12</td>
<td>0.015625</td>
<td>0.1875</td>
<td></td>
</tr>
<tr>
<td>(2,2,0,0)</td>
<td>6</td>
<td>0.0234375</td>
<td>0.140625</td>
<td></td>
</tr>
<tr>
<td>(2,1,1,0)</td>
<td>12</td>
<td>0.046875</td>
<td>0.5625</td>
<td></td>
</tr>
<tr>
<td>(1,1,1,1)</td>
<td>1</td>
<td>0.09375</td>
<td>0.09375</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>35</td>
</tr>
</tbody>
</table>

Table 2.3: The 12 distinct permutations of the partition \((2.1.1.0)\)

\[
\begin{array}{l}
M(2.1.1.0) \\
(2.1.1.0) \\
(2.0.1.1) \\
(2.1.0.1) \\
(1.0.1.2) \\
(1.0.2.1) \\
(1.1.0.2) \\
(1.1.2.0) \\
(1.2.0.1) \\
(1.2.1.0) \\
(1.0.1.2) \\
(0.1.1.2) \\
(0.1.2.1) \\
(0.2.1.1) \\
\end{array}
\]

\[
+(1!!1!!1!)^{-1} \sum_{k \in M(1.1.1.1)} \hat{\theta}^*(k). \tag{2.27}
\]

For example, the 4th summation in (2.27) represents the sum of 12 bootstrap replicates calculated from distinct resamples associated with the permutations listed in Table 2.3. A similar expression for exactly evaluating the ideal bootstrap estimator of variance of \( \hat{\theta} \) could also be obtained.
2.3.3 Moments of Bootstrap Estimators

We seek to calculate the mean and variance of the ideal bootstrap quantity

\[ T = E_{w|x}[g(\hat{\theta}^*)] \]  

(2.28)

for some function \( g \), with respect to the original distribution \( F \). Indeed, the first moment of (2.28) may be evaluated exactly, using (2.25), as

\[
E_x \left[ E_{w|x}[g(\hat{\theta}^*)] \right] = \frac{(n - 1)!}{n^{n-1}} \sum_{w \in \mathcal{W}(n)} \frac{1}{w_1! \ldots w_n!} \sum_{k \in M(w)} E_{x|w}[g(\hat{\theta}^*(k))] \\
= \sum_{w \in \mathcal{W}(n)} p(k) L(w) E_{x|w}[g(\hat{\theta}^*)] \\
= \sum_{w \in \mathcal{W}(n)} \left[ p(w) \cdot E_{x|w}[g(\hat{\theta}^*)] \right] 
\]

(2.29)

where

\[ p(w) = L(w) p(k) \]

for any \( k \in M(w) \) and the quantities \( p(k) \) and \( L(w) \) are given respectively by (2.23) and (2.26). Clearly, \( \sum_{w \in \mathcal{W}(n)} L(w) = N(n) \) as given by (2.21).

Note that the final expression in (2.29) may be alternatively obtained by considering that

\[
E_x \left[ E_{w|x}[g(\hat{\theta}^*)] \right] = E_w \left[ E_{x|w}[g(\hat{\theta}^*)] \right] \\
= \sum_{w \in \mathcal{W}(n)} \left[ p(w) \cdot E_{x|w}[g(\hat{\theta}^*)] \right].
\]

Applying this in the context of Example 1 where

\[ g(\hat{\theta}^*) = \hat{\theta}^* \]  

(2.30)
permits the first moment of (2.27) to be expressed exactly, that is,

\[ E_x \left[ E_{w|x}[g(\hat{\theta}^*)] \right] = E_x \left[ E_{w|x}(\hat{\theta}^*) \right] \]

\[ = \sum_{w \in W(n)} \left[ p(w) \cdot E_{x|w}(\hat{\theta}^*) \right] \]

\[ = p(4,0,0,0) \times \left[ E_{x|w=(4,0,0,0)}(\hat{\theta}^*) \right] \]
\[ + p(3,1,0,0) \times \left[ E_{x|w=(3,1,0,0)}(\hat{\theta}^*) \right] \]
\[ + p(2,2,0,0) \times \left[ E_{x|w=(2,2,0,0)}(\hat{\theta}^*) \right] \]
\[ + p(2,1,1,0) \times \left[ E_{x|w=(2,1,1,0)}(\hat{\theta}^*) \right] \]
\[ + p(1,1,1,1) \times \left[ E_{x|w=(1,1,1,1)}(\hat{\theta}^*) \right]. \]  

(2.31)

where \( E_{x|w=(w_1,w_2,w_3,w_4)}(\hat{\theta}^*) = E_x[\hat{\theta}^*(w_1, w_2, w_3, w_4)] \) represents the first moment of the bootstrap replication \( \hat{\theta}^* \) with respect to the specific partition \( w = (w_1, w_2, w_3, w_4) \).

The algorithm associated with (2.29) could be expressed as:

**Algorithm 2.1 (Property of a bootstrap moment):**

- Construct the list \( l \) of all partitions \( w \in W(n) \) for a given positive integer \( n \)
- Over all the partitions \( w \) in \( l \) sum

\[ p(w) \cdot E_{x|w}[g(\hat{\theta}^*)] \]
where \( p(w) = L(w) p(k) \).

The list \( l \) of all partitions \( w \in W(n) \) for a given positive integer \( n \) can be created from the algorithm given in Wang (1992, Chapter 2). The tool for calculating such partitions is called FPL\([n]\) and the code is available in Appendix A.1 in a file called partition.m.

We shall now apply this methodology to the case of assessing the mean and variance of bootstrap quantities associated with order statistics, in particular, the sample median.

### 2.4 Application to Order Statistics: The Case of the Median

In this section we apply the methodology developed in the preceding discussion to the case where

\[
\hat{\theta} = \text{sample median} = \tilde{x}.
\]  

(2.32)

Analogously,

\[
\hat{\theta}^* = \tilde{x}^*
\]  

(2.33)

denotes a replication of the bootstrapped sample median.

The median is a statistic whose distribution is known exactly - this will facilitate illustration of the methodology described above.

Before proceeding we review some general properties of order statistics.
2.4.1 Some Properties of Order Statistics

Let \( x_1, \ldots, x_n \) be a random sample of \( n \) distinct observations from an absolutely continuous population with pdf \( f(x) \) and cdf \( F(x) \). Let

\[
x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}
\]

be the order statistics obtained by arranging the above sample in increasing order of magnitude. The density function of \( x_{(i)} \) (\( 1 \leq i \leq n \)) is well-known to be

\[
f_{(i)}(x) = \frac{n!}{(i-1)!(n-i)!} [F(x)]^{i-1} [1-F(x)]^{n-i} f(x), \quad -\infty < x < \infty
\]

(2.34)

from which it follows that the \( k \)-th moment of \( x_{(i)} \) for \( k = 1, 2, \ldots \) may be presented as

\[
E(x_{(i)}^k) = \int_{-\infty}^{\infty} x^k f_{(i)}(x) dx
= \frac{n!}{(i-1)!(n-i)!} \int_{-\infty}^{\infty} x^k [F(x)]^{i-1} [1-F(x)]^{n-i} f(x) dx.
\]

(2.35)

The variance of \( x_{(i)} \) may then be expressed in terms of (2.35) as

\[
Var(x_{(i)}) = E(x_{(i)}^2) - [E(x_{(i)})]^2.
\]

(2.36)

In many situations (2.35) does not exist explicitly in closed form. An example of this is the case where \( F(x) \) represents the cumulative normal distribution. In order to circumvent this difficulty we shall express (2.35) alternatively in terms of the inverse probability integral transformation. As will be seen shortly, this provides a means of carrying an initial variable with a uniform distribution into a new variable with a special distribution of interest. For a description of this transformation see, for example, Fraser (1976, page 86-87). This representation of (2.35) will then be amenable to closed form expression through the use of a special family of approximating distributions to be described below.
Let $u_1, u_2, \ldots, u_n$ be a random sample from the $U[0, 1]$ distribution and let

$$u_{(1)} \leq u_{(2)} \leq \ldots \leq u_{(n)}$$

be the order statistics obtained from this sample. Define the inverse cumulative distribution function of the population as

$$F^{-1}(u) = \sup\{x : F(x) \leq u\}, \quad 0 < u < 1. \quad (2.37)$$

Now it is well-known that if $u$ is a $U[0, 1]$ random variable, then $F^{-1}(u)$ has distribution function $F$. We have

$$F(x_{(i)}) \overset{d}{=} u_{(i)}, \quad i = 1, 2, \ldots, n \quad (2.38)$$

since $F$ is continuous. Further, it can easily be verified by using (2.37) that

$$F^{-1}(u_i) \overset{d}{=} x_i, \quad i = 1, 2, \ldots, n$$

and

$$F^{-1}(u_{(i)}) \overset{d}{=} x_{(i)}, \quad i = 1, 2, \ldots, n. \quad (2.39)$$

where $\overset{d}{=} \overset{d}{=} \overset{d}{=}$ is to be read as "has the same distribution as". The distributional relations in (2.38) and (2.39) were observed by Scheffé and Tukey (1945). Hence, using (2.37) - (2.39) permits (2.35) to be written alternatively and more compactly as

$$E(x_{(i)}^k) = \frac{n!}{(i-1)!(n-i)!} \int_0^1 \left\{F^{-1}(u)\right\}^k u^{i-1}(1-u)^{n-i} du \quad (2.40)$$

for $1 \leq i \leq n$, $m \geq 1$. 
In the particular case when

\[ x_i \overset{iid}{\sim} U[0, 1], \quad i = 1, 2, \ldots, n \]

then (2.40) can be expressed explicitly in closed form. Indeed, (2.39) is modified to

\[ F^{-1}(u(i)) \overset{d}{=} u(i) \]

from whence it follows that

\[ E(x_{(i)}^k) = E(u_{(i)}^k) \]

\[ = \int_0^1 u^k f_i(u) du \]

\[ = \frac{B(i + k, n - i + 1)}{B(i, n - i + 1)} \tag{2.41} \]

where

\[ f_i(u) = \frac{n!}{(i - 1)!(n - i)!} u^{i-1}(1 - u)^{n-i}, \quad 0 < u < 1. \tag{2.42} \]

is the density function of a two-parameter beta distribution, and \( B(... \) is the complete beta function defined by

\[ B(p, q) = \int_0^1 t^{p-1}(1 - t)^{q-1} dt, \quad p, q > 0 \]

\[ = \frac{\Gamma(p)\Gamma(q)}{\Gamma(p + q)}. \tag{2.43} \]

Upon simplification, (2.41) yields

\[ E(x_{(i)}^k) = \frac{n!}{(n + k)!} \frac{(i + k - 1)!}{(i - 1)!} \tag{2.44} \]

For example, the mean and variance of the sample median \( \hat{x} \) for odd sample size may
Table 2.4: Approximating distributions for various \( \lambda \) values

<table>
<thead>
<tr>
<th>( \lambda )</th>
<th>Approximating Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.135</td>
<td>Normal</td>
</tr>
<tr>
<td>0</td>
<td>Logistic</td>
</tr>
<tr>
<td>1</td>
<td>Uniform</td>
</tr>
</tbody>
</table>

be shown to be

\[
E(\hat{x}) = \frac{1}{2} \quad \text{and} \quad \text{Var}(\hat{x}) = \frac{1}{4(n + 2)}.
\]

The above example illustrates the use of the particular inverse function given by

\[
x = F^{-1}(u) = u
\]

which permits (2.40) to be expressed in closed form in this particular case. A special form of \( F^{-1}(u) \) permitting closed form expression of moments of order statistics in a variety of cases was provided by Hastings et al (1947). They suggested the class of approximating distributions implicit in the transformation

\[
x = F^{-1}(u)
\]

\[
= u^\lambda - (1 - u)^\lambda, \quad \text{for} \ \lambda > 0. \tag{2.45}
\]

where \( u \) has a \( U[0, 1] \) distribution, \( x \) is the random variable whose order statistics interest us and \( \lambda \) is a distributional shape parameter. All the distributions represented by (2.45) for various values of \( \lambda \) are symmetrical. Table 2.4 presents 3 examples of approximating distributions generated by (2.45) for the indicated values of \( \lambda \).

Simulated data sets generated according to (2.45) for the 3 values of \( \lambda \) shown in Table 2.4 may be seen to exhibit properties closely mimicking those of the normal, logistic and uniform distributions, respectively.

By using the representation (2.45) we are now able to express moments of order statistics in closed form. Indeed, substitution of (2.45) into (2.40) yields linear
combinations of beta functions. For general sample size \( n \) and distributional shape parameter \( \lambda \), the second moment of the \( i \)-th order statistic can be expressed explicitly and exactly in terms of beta functions as

\[
E(x_{(i)}^2) = \frac{n!}{(i-1)!(n-i)!} \int_0^1 \left( u^\lambda - (1 - u)^\lambda \right)^2 u^{i-1} (1 - u)^{n-i} du \\
= \frac{n!}{(i-1)!(n-i)!} \left[ \int_0^1 u^{i-1}(1 - u)^{2\lambda+n-i} du \right. \\
- 2 \int_0^1 u^{\lambda+i-1}(1 - u)^{\lambda+n-i} du \\
+ \int_0^1 u^{2\lambda+i-1}(1 - u)^{n-i} du \right. \\
= \frac{1}{B(i, n - i + 1)} \left[ B(i, 2\lambda + n - i + 1) - 2 B(\lambda + i, \lambda + n - i + 1) \right. \\
+ B(2\lambda + i, n - i + 1) \right].
\]

(2.46)

A simple algorithm was implemented in Mathematica that computes (2.46). The function is called `SecondMomOrder [i, n, \lambda]`.

For example, when \( \lambda = 0.135 \) the distribution of the random variable \( x \) generated by

\[
x = F^{-1}(u) = u^{0.135} - (1 - u)^{0.135}
\]

approximates a normal distribution. When \( n = 3 \) the second moment of the median \( E(\hat{x}^2) \) is

\[
\text{SecondMomOrder} [2, 3, 0.135] = 0.0174397
\]

(2.48)
which is precisely (2.46) for \( i = 2, \ n = 3, \ \lambda = 0.135 \). When \( \lambda = 1 \) then (2.45) becomes

\[
x = F^{-1}(u) \\
= 2u - 1
\]

which is distributed as \( U[-1, 1] \). In this case, \( E(\bar{x}^2) \) is

\[
\text{SecondMomOrder} \ [2, 3, 1] = 0.2 \quad .
\]

### 2.4.2 The Case of the Median

We now use the tools discussed in the previous sections to evaluate properties of bootstrap estimators associated with the median. We commence by considering in detail the first moment of the bootstrap estimator of the second moment of the median in the case when the random sample is of size \( n = 3 \). This result will suggest that properties (such as moments or cumulants) of bootstrapped moments of order statistics are linear combinations of bootstrapped moments of order statistics of different sample sizes. Later we shall consider computations of such moments for various sample sizes and distributions. We use the following Table 2.5 which is analogous to Table 2.2 in the present context:

<table>
<thead>
<tr>
<th>( n = 3 )</th>
<th>( W(3) )</th>
<th>( N(w) )</th>
<th>( p(k) )</th>
<th>( p(w) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3,0,0)</td>
<td>3</td>
<td>1/27</td>
<td>1/9</td>
<td></td>
</tr>
<tr>
<td>(2,1,0)</td>
<td>6</td>
<td>1/9</td>
<td>6/9</td>
<td></td>
</tr>
<tr>
<td>(1,1,1)</td>
<td>1</td>
<td>2/9</td>
<td>2/9</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>10</td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Noting that there are three partitions for \( n = 3 \) and substituting

\[
g(\hat{\theta}^*) = (\hat{\theta}^*)^2 = (\tilde{z}^*)^2
\]

into (2.31) gives

\[
E_x \left[ E_w \left[ (\tilde{z}^*)^2 \right] \right] = \sum_{w \in W(n)} \left[ p(w) \cdot E_{x|w} \left[ (\tilde{z}^*)^2 \right] \right]
\]

\[
= \sum_{i=1}^{3} \left[ p(w_i) \cdot E_x((\tilde{z}_i^*)^2|w_i) \right]
\]

\[
= p(3,0,0) \times \left[ E_x((\tilde{z}_1^*)^2|(3,0,0)) \right]
+ p(2,1,0) \times \left[ E_x((\tilde{z}_2^*)^2|(2,1,0)) \right]
+ p(1,1,1) \times \left[ E_x((\tilde{z}_3^*)^2|(1,1,1)) \right]
\]

\[
= \left( \frac{1}{9} \right) \times \left[ E_x((\tilde{z}_1^*)^2|(3,0,0)) \right]
+ \left( \frac{6}{9} \right) \times \left[ E_x((\tilde{z}_2^*)^2|(2,1,0)) \right]
+ \left( \frac{2}{9} \right) \times \left[ E_x((\tilde{z}_3^*)^2|(1,1,1)) \right]
\]

(2.50)

where \( \tilde{z}_i^* \) denotes the sample median associated with the \( i \)-th partition \( w_i \).

Consider first \( \tilde{z}_1^* | (3,0,0) \) in (2.50). This denotes that the sample median is associated with the partition \( (3,0,0) \). The partition \( (3,0,0) \) places all the weight on the first observation; it may be interpreted as representing three identical realizations of a single random variable \( x \) in which case \( n = 1 \). Hence,

\[
\tilde{z}_1^* | (3,0,0) = \tilde{z}^* | (3,0,0) = x_{(1)} = x.
\]
Hence, the median of three identical realizations reduces to a median of a sample of size \( n = 1 \), i.e. a single random variable. Therefore,

\[
E_x[(\hat{x}^*_1)^2|\{3, 0, 0\}] = E_x(x^2)
\]

Next consider \( \hat{x}^*_2 | (2, 1, 0) \). The partition \((2, 1, 0)\) may be interpreted as representing the realizations of two distinct random variables so that \( n = 2 \). One of the variables provides 2 identical observations while the other has generated a single different observation. There are thus two order statistics each of which has probability \( \frac{1}{2} \) of representing the median. Hence,

\[
\hat{x}^*_2 | (2, 1, 0) = \hat{x}^* | (2, 1, 0)
\]

\[
= \begin{cases} 
  x_{(1)} & \text{w.p. } 0.5 \\
  x_{(2)} & \text{w.p. } 0.5 
\end{cases}
\]  

Applying the expectation operator gives

\[
E_x[(\hat{x}^*_2)^2|(2, 1, 0)] = \left[0.5 \times E(x_{(1)}^2)\right] + \left[0.5 \times E(x_{(2)}^2)\right]
\]

\[
= \int_{-\infty}^{\infty} x^2 f(x) \, dx
\]

\[
= E_x(x^2)
\]

by using (2.35).

Finally, consider \( \hat{x}^*_3 | (1, 1, 1) \). The partition \((1, 1, 1)\) may be interpreted as representing respective realizations of 3 distinct random variables. In this case \( n = 3 \). There are thus three distinct order statistics and it follows that

\[
\hat{x}^*_3 | (1, 1, 1) = \hat{x}^* | (1, 1, 1)
\]

\[
= x_{(3)}
\]
which provides

\[ E_x[(\hat{x}_3^*)^2 \mid (1,1,1)] = E_x(x_{(2)}^2) \, . \]

Equation (2.50) now becomes

\[
E_x \left[ E_{w \mid x}[(\hat{x}^*)^2] \right] = (1/9) \times E(x^2) \\
+ (6/9) \times E(x^2) \\
+ (2/9) \times E(x_{(2)}^2) \\
= (7/9) \times E(x^2) + (2/9) \times E(\hat{x}^2) \tag{2.52}
\]

where \( E(\hat{x}^2) = E(x_{(2)}^2) \). Equation (2.52) represents a linear combination of a second moment of sample size 1 and a second moment of the median for a sample size 3. It is interesting to note that this expression is weighted in favour of the sample of size 1. It is an analytical expression, a formula, which is true for any distribution. It can be efficiently evaluated numerically and exactly, for example, for any distribution of interest belonging to the \( \lambda \)-family of distributions generated by (2.45). Similar analytical formulas may be generated for any sample size although we consider only odd sample sizes in this thesis. The method allows us to study, for example in this context, the properties of the bootstrap procedure which estimates the second moment of the sample median.

The algorithm describing the calculation of the conditional expectation of the squared sample median for a given partition for odd values of \( n \) and arbitrary parent distribution function \( F(x) \) is given by:

**Algorithm 2.2 (Calculation of \( E_{w \mid x}[(\hat{x}^*)^2] \) for \( F(x) \) and odd \( n \)):**

- Compute the number of distinct order statistics for the given partition. This equals the number of non-zero elements in the partition. In the case of repeated observations the corresponding order statistic is the common value.
• Compute all permutations of the given partition. Each permutation assumes observations arranged in descending order.

• For each permutation record the order statistic which corresponds to the median.

• Count the number of permutations in which the first order statistic is the median. Do the same for all order statistics.

• Divide each count by the number of elements in the partition. Each order statistic is hence associated with a numerical probability that it represents the median.

• The conditional expectation of the squared sample median for a given partition is the sum of products of the expectation (over an arbitrary parent $F$) of each squared order statistic and its associated probability of representing the median for that partition. The expectation is evaluated, in general, by numerical integration.

• If $F$ is a member of the $\lambda$-family then closed forms of moments of order statistics are evaluated using the symbolic function SecondMomOrder[$i, n, \lambda$].

The tool which gives the order statistics of a partition $w$ with the respective probabilities they have of representing the median is called Med[$w$] and the code is available in Appendix A.1 in a file called os.m.

Similarly, the first moment of the bootstrap estimate of variance of the median may be generally expressed as

$$E_x [Var_w[x|x^*]] = E_x \left[ E_{w|x}(x^*)^2 - \left( E_{w|x}(x^*) \right)^2 \right]$$

$$= E_x \left[ E_{w|x}(x^*)^2 \right] - E_x \left[ \left( E_{w|x}(x^*) \right)^2 \right]$$

(2.53)

which for the case $n = 3$ becomes

$$E_x [Var_w[x|x^*]] = \frac{7}{9} \times Var(x) + \frac{2}{9} \times Var(\bar{x}) .$$

(2.54)
The exact bias of the bootstrap estimator of variance of the median for \( n = 3 \) follows immediately from (2.54) as

\[
BIAS \left[ Var_{w|x}(\hat{x}^*) \right] = E_x \left[ Var_{w|x}[\hat{x}^*] \right] - Var(\hat{x})
\]

\[
= (7/9) \times Var(x) + (2/9) \times Var(\bar{x}) - Var(\hat{x})
\]

\[
= (7/9) \times \left[ Var(x) - Var(\bar{x}) \right]
\]

(2.55)

A similar evaluation with respect to (2.52) yields

\[
BIAS \left[ E_{w|x}[(\hat{x}^*)^2] \right] = (7/9) \times \left[ E(x^2) - E(\bar{x}^2) \right].
\]

(2.56)

Note that the discussion in this chapter centers on the evaluation of

\[
E_x \left[ E_{w|x}(\hat{x}^*)^2 \right]
\]

(2.57)

which is identical to

\[
E_x \left[ E_{w|x}(\hat{x}^* - E_x(\hat{x}))^2 \right]
\]

(2.58)

assuming that

\[
E_x(\hat{x}) = 0.
\]

(2.59)

that is, that the population median is zero. More generally, for an estimator \( \hat{\theta} \) where we denote \( E_x(\hat{\theta}) = \theta \), the inner expectation in (2.58) may be written as

\[
E_{w|x}(\hat{\theta}^* - \theta)^2 = E_{w|x}(\hat{\theta}^* - \hat{\theta} + \hat{\theta} - \theta)^2
\]

\[
\approx E_{w|x}(\hat{\theta}^* - \hat{\theta})^2 + (\hat{\theta} - \theta)^2.
\]

(2.60)

assuming that \( E_{w|x}(\hat{\theta}^* - \hat{\theta}) \) is approximately uncorrelated with \( \hat{\theta} - \theta \). Considering
the expectation of (2.60) over \( x \) we then observe that

\[
E_x \left[ E_{w|x}(\hat{\theta}^* - \theta)^2 \right] = E_x \left[ E_{w|x}(\hat{\theta}^* - \hat{\theta})^2 \right] + E_x(\hat{\theta} - \theta)^2 \\
= E_x(\hat{\theta} - \theta)^2 + E_x(\hat{\theta} - \theta)^2 \\
= 2 \cdot E_x(\hat{\theta} - \theta)^2. \tag{2.61}
\]

assuming \( E_{w|x}(\hat{\theta}^* - \hat{\theta})^2 \) is close to \( (\hat{\theta} - \theta)^2 \). It follows that

\[
E_x \left[ E_{w|x}(\hat{\theta}^*)^2 \right] \simeq 2 \cdot E_x(\hat{\theta}^2) \tag{2.62}
\]

if \( \theta = 0 \) is assumed. In the case of the median we have assumed (2.59) in this chapter and thus we will observe

\[
E_x \left[ E_{w|x}(\hat{x}^*)^2 \right] \simeq 2 \cdot E_x(\hat{x}^2). \tag{2.63}
\]

The relationship given by (2.63) is corroborated by numerical results discussed later. This does not signify that the bootstrap second moment of the median, \( E_{w|x}(\hat{x}^*)^2 \), is positively biased but only that (2.63) is due to the assumption (2.59) that has been made.

The Mathematica functions \texttt{ExpBoot}[n, \lambda] and \texttt{VarBoot}[n, \lambda] respectively compute

\[
E_x \left[ E_{w|x}((\hat{x}^*)^2) \right] = \sum_{w \in W(n)} [p(w) \cdot E_{x|w}((\hat{x}^*)^2)] \tag{2.64}
\]

and

\[
Var_x \left[ E_{w|x}((\hat{x}^*)^2) \right] = \sum_{w \in W(n)} [p(w)^2 \cdot Var_{x|w}((\hat{x}^*)^2)] \tag{2.65}
\]

by using the function \texttt{SecondMomOrder}[i, n, \lambda] seen above and so work only for the \( \lambda \)-family of approximating distributions and odd sample size \( n \). The bias of the
bootstrap estimator of the second moment of the median may be expressed as

$$\text{BIAS} \left[ E_w|x[(\hat{x}^*)^2] \right] = E_x \left[ E_w|x[(\hat{x}^*)^2] \right] - E_x(\hat{x}^2) \quad (2.66)$$

where $E_x(\hat{x}^2)$ represents the true value of the second moment of the median. The latter can be computed by using `SecondMomOrder[i, n, \lambda]` whenever $x(i) = \hat{x}$.

The algorithms describing the evaluation of each of (2.64) and (2.65) is given next:

**Algorithm 2.3 (Calculation of ExpBoot[n, \lambda]):**

- Derive all partitions of the positive integer $n$ using `FPL[n]`.
- Compute the multinomial-type probability of occurrence of each partition $w$ using `Prob[w]` from Appendix A.1.
- Use `Med[w]` to find the order statistics for each partition $w$ having non-zero probabilities of representing the median. Compute the probabilities.
- Compute the conditional expectation of the squared sample median for each partition by summing the products of `SecondMomOrder[i, n, \lambda]` and the respective probabilities the order statistics have of representing the median.
- Sum the conditional expectations found in the previous step weighted by the respective multinomial-type probabilities of occurrence of the partitions.

The code corresponding to the tool `ExpBoot[n, \lambda]` is available in Appendix A.1 in a file called `Expboot.m`.

**Algorithm 2.4 (Calculation of VarBoot[n, \lambda]):**

- Derive all partitions of the positive integer $n$ using `FPL[n]`.
- Compute the square of the multinomial-type probability of occurrence of each partition $w$ using `Prob[w]` from Appendix A.1.
• Use Med[w] to find the order statistics for each partition w having non-zero probabilities of representing the median. Compute the probabilities.

• Compute the conditional variance of the squared sample median for each partition by summing the products of VarSqOrder [i, n, λ] with the respective probabilities the order statistics have of representing the median. This tool is a function of SecondMomOrder [i, n, λ].

• Sum the conditional variances found in the previous step weighted by the respective squared multinomial-type probabilities of occurrence of the partitions.

The code corresponding to the tool VarBoot[n, λ] is available in Appendix A.1 in a file called Varboot.m.

The expressions (2.64), (2.65) and (2.66) represent various properties of the bootstrap estimator of the second moment of the median which can be explicitly evaluated for given n and λ. For example, when n = 3 and λ = 0.135 then (2.64) is

$$\text{ExpBoot}[3, 0.135] = 0.0342303$$

However, setting n = 3 only in ExpBoot[n, λ] yields the formula

$$\text{ExpBoot}[3, \lambda] = 0.111111 \left( \frac{2}{1 + 2l} - 2 \text{Beta}[1 + l, 1 + l] \right) +$$

$$0.666667(0.5 \left[ \frac{2}{2 + 2l} + \frac{2}{(1 + 2l)(2 + 2l)} - 4 \text{Beta}[2 + l, 1 + l] \right] +$$

$$0.5 \left[ \frac{4}{1 + 2l} - \frac{2}{2 + 2l} - \frac{2}{(1 + 2l)(2 + 2l)} -$$

$$4 \text{Beta}[1 + l, 1 + l] + 4 \text{Beta}[2 + l, 1 + l]] \right) +$$

$$0.222222(\frac{6}{2 + 2l} + \frac{6}{(1 + 2l)(2 + 2l)} - \frac{6}{3 + 2l} -$$
where $l$ represents $\lambda$. This is precisely the $\lambda$-family equivalent of the first expression in (2.52). Similar analytical representations may be given for any $n$. We qualify this, however, as follows. Note that (2.67) contains 3 major terms. Table 2.1 indicates that for $n = 3$, ten terms would be involved if we were evaluating the bootstrap estimator directly. The exact calculation of the bootstrap estimator (based on exhaustive enumeration of resamples) for a sample size of $n = 15$ would involve the enumeration of almost 80,000,000 resamples whereas a mere 176 partitions would be required for the exact computation of a property of the same bootstrap estimator! This indicates that the calculations of moments of bootstrap estimators using the methodology developed in this chapter may be carried out for much larger sample sizes than would be feasible if bootstrap estimators were to be evaluated directly.

Recall that we have shown by first principles that for $n = 3$ (2.64) reduces to (2.52). We can hence compute the right hand side of (2.52) directly as a check on the Mathematica code underlying `ExpBoot`[$n, \lambda$]. Indeed, $E(x^2)$ and $E(\hat{x}^2)$ in (2.52) are computed as

\[
E(x^2) = \int_{-\infty}^{\infty} x^2 f(x) \, dx
\]

\[
= \int_{0}^{1} \left( u^{0.135} - (1 - u)^{0.135} \right)^2 \, du
\]

\[
= \text{Beta}(1, 1.27) - 2\text{Beta}(1.135, 1.135) + \text{Beta}(1.27, 1)
\]

\[
= 0.0390276
\]

and $E(\hat{x}^2)$ was given in (2.48) as

\[
\text{SecondMomOrder}[2, 3, 0.135] = 0.0174397
\]
Thus the right hand side of (2.52) is

$$(7/9) \times E(x^2) + (2/9) \times E(\hat{x}^2) = 0.0342303$$

which substantiates the left hand side of (2.52) as computed by $\text{ExpBoot}[n, \lambda]$. Similarly, when $n = 3$, $\lambda = 1$ we have $x = 2u - 1 \sim U[-1, 1]$ and then

$$\text{ExpBoot}[3, 1] = 0.303704$$

for which

$$E(x^2) = Var(x) + [E(x)]^2 = \frac{1}{3}$$

and (2.49) permit

$$(7/9) \times E(x^2) + (2/9) \times E(\hat{x}^2) = 0.303703$$

as desired.

The next section presents tabulated results of the Mathematica functions described here for various $n$ and $\lambda$. Included in the Tables are the results of independent simulations which provide checks of the algorithms underlying the functions.

2.4.3 Some Numerical Results

Tables 2.6, 2.7 and 2.8 below present numerical results for various small sample sizes and three values of $\lambda$ associated with 10 quantities. The quantities have been multiplied by $n$ so as to remove the decreasing effect due to increasing sample size.

The ten quantities presented in the above tables are described as follows:

1. ExpBoot represents the Mathematica function $\text{ExpBoot}[n, \lambda]$ defined in the preceding section.
Table 2.6: Analytical and simulated results for various properties and bootstrap estimates related to the 2nd moment of the median using several sample sizes for the case of the normal distribution

<table>
<thead>
<tr>
<th>$\lambda = 0.135$</th>
<th>[Normal]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>$3$</td>
</tr>
<tr>
<td>$n$ ExpBoot</td>
<td>$0.1027$</td>
</tr>
<tr>
<td>$n$ ExpBoot$_{sim}$</td>
<td>$0.1101$</td>
</tr>
<tr>
<td>$n$ ExpBoot$_{sim-stderr}$</td>
<td>$(0.0047)$</td>
</tr>
<tr>
<td>$n$ $E_{true}(\hat{x}^2)$</td>
<td>$0.0523$</td>
</tr>
<tr>
<td>$n$ $E_{sim}(\hat{x}^2)$</td>
<td>$0.0538$</td>
</tr>
<tr>
<td>$n$ $E_{sim}(\hat{x}^2)_{stderr}$</td>
<td>$(0.0024)$</td>
</tr>
<tr>
<td>$n$ Bias</td>
<td>$0.0504$</td>
</tr>
<tr>
<td>$n$ VarBoot</td>
<td>$0.0043$</td>
</tr>
<tr>
<td>$n$ boot$_{reg}$</td>
<td>$0.1276$</td>
</tr>
<tr>
<td>$n$ boot$_{reg-stderr}$</td>
<td>$(0.0019)$</td>
</tr>
</tbody>
</table>

2. ExpBoot$_{sim}$ was calculated as a check of the correctness of the Mathematica code underlying ExpBoot$_{n, \lambda}$. ExpBoot$_{sim}$ is an estimate of ExpBoot$_{n, \lambda}$ for a particular $n$ and $\lambda$ and was computed according to the following algorithm. This algorithm was coded using SPlus.

- For each value of $\lambda$ and $n$ carry out the following:
  - Randomly select 1000 samples of size $n$ (odd) from the U[0.1] distribution
  - For each sample:
    - select $B = 1$ resamples of size $n$
    - calculate median $\hat{x}$ of the resample
    - transform to $\lambda$-family variate using $x = (\hat{x})^\lambda - (1 - \hat{x})^\lambda$
- Calculate $\text{ExpBoot}_{sim} = \text{mean}(x_1, \ldots, x_{1000})$
- Calculate $\text{ExpBoot}_{sim-stderr} = \left(\frac{\text{Variance}(x_1, \ldots, x_{1000})}{1000}\right)^{\frac{1}{2}}$

3. ExpBoot$_{sim-stderr}$ is defined in 2. above.

4. $E_{true}(\hat{x}^2)$ is computed by the function SecondMomOrder[$i, n, \lambda$] whenever the $i$-th order statistic is the median in a sample of odd sample size $n$. This function
Table 2.7: Analytical and simulated results for various properties and bootstrap estimates related to the 2nd moment of the median using several sample sizes for the case of the uniform distribution

<table>
<thead>
<tr>
<th>n</th>
<th>λ = 1.0</th>
<th>[Uniform]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.9111</td>
<td>1.1821</td>
</tr>
<tr>
<td>5</td>
<td>0.9309</td>
<td>1.2427</td>
</tr>
<tr>
<td>7</td>
<td>0.9745</td>
<td>1.3704</td>
</tr>
<tr>
<td>9</td>
<td>1.4479</td>
<td>1.5102</td>
</tr>
</tbody>
</table>

was defined in the preceding section.

5. $E_{\text{sim}}(\hat{z}^2)$ was calculated as a check of the correctness of the Mathematica code underlying SecondMomOrder[$i, n, \lambda$] which computes $E_{\text{true}}(\hat{z}^2)$ whenever the $i$-th order statistic is the median in a sample of odd sample size $n$. $E_{\text{sim}}(\hat{z}^2)$ is an estimate of $E_{\text{true}}(\hat{z}^2)$ for a particular $n, \lambda$, and $i$ and was computed according to the algorithm described below. This algorithm was coded using SPlus.

- For each value of $\lambda$ and $n$ carry out the following:
  - Randomly select 1000 samples of size $n$ (odd) from the $U[0,1]$ distribution
  - For each sample:
    - calculate median $\hat{z}$
    - transform to $\lambda$-family variate using $x = (\hat{z})^\lambda - (1 - \hat{z})^\lambda$
  - Calculate $E_{\text{sim}}(\hat{z}^2) = \text{mean}(x_1^2, \ldots , x_{1000}^2)$
  - Calculate $E_{\text{sim}}(\hat{z}^2)_{\text{stderr}} = \left(\frac{\text{variance}(x_1^2, \ldots , x_{1000}^2)}{1000}\right)^{\frac{1}{2}}$

6. $E_{\text{sim}}(\hat{z}^2)_{\text{stderr}}$ is defined in 5. above.

7. Bias = $\text{ExpBoot}[n, \lambda] - E_{\text{true}}(\hat{z}^2)$
Table 2.8: Analytical and simulated results for various properties and bootstrap estimates related to the 2nd moment of the median using several sample sizes for the case of the logistic distribution

<table>
<thead>
<tr>
<th></th>
<th>( \lambda = 0 )</th>
<th>[Logistic]</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>n ExpBoot</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>0.5784</td>
<td>0.7172</td>
<td>0.7823</td>
<td>0.8236</td>
<td></td>
</tr>
<tr>
<td></td>
<td>n ExpBoot_sim</td>
<td>0.5785</td>
<td>0.7233</td>
<td>0.7707</td>
<td>0.8240</td>
</tr>
<tr>
<td></td>
<td>(0.0201)</td>
<td>(0.0264)</td>
<td>(0.0303)</td>
<td>(0.0322)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>n ( \bar{E}_{true}(\hat{x}^2) )</td>
<td>0.3493</td>
<td>0.3981</td>
<td>0.4231</td>
<td>0.4382</td>
</tr>
<tr>
<td></td>
<td>0.3648</td>
<td>0.3833</td>
<td>0.3963</td>
<td>0.4281</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.0135)</td>
<td>(0.0153)</td>
<td>(0.0169)</td>
<td>(0.0184)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>n Bias</td>
<td>0.2291</td>
<td>0.3192</td>
<td>0.3592</td>
<td>0.3854</td>
</tr>
<tr>
<td></td>
<td>0.0711</td>
<td>0.0319</td>
<td>0.0206</td>
<td>0.0143</td>
<td></td>
</tr>
<tr>
<td></td>
<td>n VarBoot</td>
<td>0.4673</td>
<td>0.7387</td>
<td>0.7423</td>
<td>1.7976</td>
</tr>
<tr>
<td></td>
<td>(0.0174)</td>
<td>(0.0281)</td>
<td>(0.0227)</td>
<td>(0.0414)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>n ( \text{boot}_{\text{reg}} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.0174)</td>
<td>(0.0281)</td>
<td>(0.0227)</td>
<td>(0.0414)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>n ( \text{boot}_{\text{reg}-\text{stderr}} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8. VarBoot represents the Mathematica function \texttt{VarBoot}[n, \lambda] defined in the previous section.

9. \( \text{boot}_{\text{reg}} \) represents a conventional bootstrap calculation of the second moment of the median and was computed according to the algorithm described below.

   This algorithm was coded using \texttt{SPlus}.

   - For each value of \( \lambda \) and \( n \) carry out the following:
   - Randomly select one sample of size \( n \) (odd) from the \( U[0, 1] \) distribution
   - From this single sample select \( B = 1000 \) bootstrap resamples
   - For each of the bootstrap resamples:
     - calculate median \( \hat{x} \)
     - transform to \( \lambda \)-family variate using \( x = (\hat{x})^\lambda - (1 - \hat{x})^\lambda \)
   - Calculate \( \text{boot}_{\text{reg}} = \text{mean}(x_1^2, \ldots, x_{1000}^2) \)
   - Calculate \( \text{boot}_{\text{reg}-\text{stderr}} = \left(\frac{\text{Variance}(x_1^2, \ldots, x_{1000}^2)}{1000}\right)^{\frac{1}{2}} \)

10. \( \text{boot}_{\text{reg}-\text{stderr}} \) is defined in 9. above.
2.5 Conclusions and Discussion

In this chapter we have detailed a diagnostic methodology that permits the study of properties of bootstrap estimators associated with order statistics. This approach was illustrated for the case where the estimator is the median of a sample of odd sample size. The ideal bootstrap estimator of the second moment of the median can be expressed as a weighted multinomial average of order statistics, conditional on the underlying sample, and this can be computed exactly through exhaustive enumeration of distinct resamples.

Algorithms were developed that permit moments of bootstrap estimators in this context to be expressed symbolically, and if desired, numerically, for any distribution and for any odd sample size. The methodology provides exact results. The correctness of the algorithms was checked by comparing numerical results generated by the algorithms with independent simulations based on conventional Monte Carlo resampling. This has been confirmed by the results given in Tables 2.6, 2.7 and 2.8.

The numerical results corroborate the relationship given by (2.63) especially for a Gaussian shaped distribution. The apparent 2:1 bias suggested in Tables 2.6, 2.7 and 2.8 is largely due to the definitional assumption (2.59) that has been made rather than to any long-run discrepancy between the bootstrap second moment of the median and what it is attempting to estimate.

The methodology developed in this chapter for studying properties of bootstrap estimators involving exhaustive enumeration is deemed useful for sample sizes of moderate size. Indeed, characteristics of bootstrap procedures are more unpredictable for smaller sample sizes rather than for larger sample sizes and the symbolic tools presented here provide a technique for assessing these properties. The consensus in the literature is that bootstrap estimates calculated on the basis of exhaustive enumeration may be competitive with Monte Carlo resampling up to about \( n = 10 \). The reason that \textit{properties} of bootstrap estimates may be assessed for much larger sample sizes is that the calculations involved here use partitions which are substantially fewer than the respective number of all possible distinct resamples.
Table 2.1 illustrates this. For example, the exact calculation of a bootstrap estimator (based on exhaustive enumeration of resamples) for a sample size of $n = 15$ would involve the enumeration of almost 80,000,000 resamples whereas a mere 176 partitions would be required for the exact computation of a property (i.e. a moment) of the same bootstrap estimator. It appears feasible that the algorithms discussed in this chapter could be further developed to permit symbolic expression of higher moments of bootstrap estimators associated not only with more general L-estimators but with M-estimators as well. In addition, greater computational efficiency could be achieved by eliminating multinomial weights possessing negligible probabilities.

In the next chapter, we will discuss an alternative methodology that permits analytical expression of bootstrap estimators and corresponding properties associated with the class of M-estimators.
Chapter 3

Bootstrapped M-estimators and their Properties

3.1 Introduction

In this chapter, procedures are developed and implemented for analytically computing bootstrap estimators and their properties for statistics defined by estimating equations and functions of these. The resulting symbolic quantities are expressed in terms of series expansions to any desired order. Moreover, these formulas may then be evaluated numerically for any particular set of \( n \) iid observations. This method entirely avoids the use of conventional Monte Carlo resampling.

The basis for the proposed methodology comprises the six following steps:

1. A statistic defined by estimating equations is approximated to any order by a finite linear combination of products of normalized, centered sums of independent and identically distributed random variables.

2. A bootstrap replicate of such a statistic may be similarly (analytically) expressed as a finite linear combination of products of weighted sums. The weights arise from the multinomial distribution. The weighted sums are decomposed into simple sums and products of sums involving distinct observations (henceforth referred to as disjoint sums).
3. The bootstrap expectation of the statistic is obtained by considering the expectation of the series expansion with respect to the multinomial weights arising in (2). The moments of the weights are found from the moment generating function of the multinomial distribution.

4. Products of sums involving distinct observations are re-expressed as linear combinations of simple sums and ordinary products of sums.

5. Properties of the preceding expressions may be evaluated. This corresponds to calculating various moments of the analytical bootstrap quantities with respect to the original random variable $x$, for example, in the one-sample case.

6. Symbolic expressions representing bootstrap moments and their properties may be further assessed numerically if desired by evaluating these for a particular set of data.

In Sections 3.2.2—3.2.4 the main methodological details are presented. Specifically, we establish (1) in Section 3.2.2 by considering the inversion of the Taylor series representation of an estimating equation. Sections 3.2.3—3.2.4 discuss (2) — (3) while (4) — (6) are discussed in Section 3.2.4. Section 3.3 focuses on computational issues. In Section 3.4 we apply the algorithms to the case of the correlation coefficient. Specifically, we compare our results to those obtained by Efron and Tibshirani (1993) using their Law School data. We also consider an artificially generated sample of $n = 100$ bivariate Gaussian observations.

The methods described in the present chapter have been implemented for machine computation, and in fact depend primarily on two key symbolic routines - one for computing asymptotic expansions of roots of estimating equations, and one for computing bootstrap analogues to these expansions. The algorithms have been implemented using Mathematica (Wolfram, 1988). The routines are very simple to apply to a broad range of statistics. The Mathematica code is presented in Appendix A.2 of this thesis.
3.2 Bootstrapping M-estimators: An Analytical Approach

3.2.1 Introduction
The following sub-sections present the methodological details relevant to steps (1)–(6) given above.

3.2.2 An Approximation for Functions of M-estimators
In this section we show how a single arbitrary M-estimator \( \hat{\theta} \), or a smooth function of such estimators, may be expressed as a particular sequence of increasingly accurate series approximations. The discussion below follows Andrews and Feuerverger (1993). Alternative presentations may also be found in Ferguson (1989) and McCullagh (1987).

Let \( x_1, \ldots, x_n \) represent independently and identically distributed observations from a distribution indexed by a parameter \( \theta \), with \( \theta_0 \) being the assumed true value. As developed by Huber (1964, 1967) we consider first a one-dimensional M-estimator \( \hat{\theta} \) assumed to be the solution of the equation

\[
\hat{\psi}_n(\theta) \equiv \frac{1}{n} \sum_{i=1}^{n} \psi(x_i, \theta) = 0. \tag{3.1}
\]

If the \( x_i \)'s arise from the density \( f(x, \theta) \), and if \( \psi(x, \theta) = \frac{\partial}{\partial \theta} \log f(x, \theta) \), it follows that \( \hat{\theta} \) is the maximum likelihood estimate of \( \theta \).

Denoting expectation with respect to \( f_{\theta_0} \) by \( E_0 \), we shall require that

\[
E_0 \psi(x, \theta_0) = 0 \tag{3.2}
\]

and

\[
E_0 \psi'(x, \theta_0) \neq 0
\]

so that \( \hat{\theta} \) is \( \sqrt{n} \)-consistent for \( \theta_0 \) (see, for example, Huber (1977)). The following
construction enables us to write \( \hat{\theta} \) in the form of a series approximation. Consider the \( k \)-th order Taylor expansion of (3.1) about \( \theta_0 \):

\[
0 = \frac{1}{n} \sum_{i=1}^{n} \psi(x_i, \theta) = \sum_{j=0}^{k} \frac{(\hat{\theta} - \theta_0)^j}{j!} \tilde{\psi}_n^{(j)}(\theta_0) + R_{k+1} \tag{3.3}
\]

where

\[
\tilde{\psi}_n^{(j)}(\theta_0) = \frac{1}{n} \sum_{i=1}^{n} \psi^{(j)}(x_i, \theta_0).
\]

The superscripts \((j)\) on \( \psi \) represent differentiation with respect to \( \theta \). The remainder term appearing in (3.3) may be expressed as

\[
R_{k+1} \equiv \frac{(\theta^* - \theta_0)^{k+1}}{(k+1)!} \tilde{\psi}_n^{(k+1)}(\theta^*) = O_P(n^{-(k+1)/2})
\]

for some \( \theta^* \) between \( \theta \) and \( \theta_0 \), and is of the order indicated whenever \( \theta^* = \theta_0 + O_P(n^{-1/2}) \) provided only that \( \psi^{(k+1)}(x, \theta) \) is uniformly integrable in a neighborhood of \( \theta_0 \).

Now observe that (3.3) may alternatively be written as

\[
\tilde{\psi}_n(\theta) = \sum_{j=1}^{k} \frac{(\hat{\theta} - \theta_0)^j}{j!} E_0 \psi^{(j)}(x, \theta_0) + \sum_{j=0}^{k} \frac{(\hat{\theta} - \theta_0)^j}{j!} \frac{Z_n^{(j)}(\theta_0)}{\sqrt{n}} + R_{k+1} \tag{3.4}
\]

where

\[
Z_n(\theta) \equiv \sqrt{n} \left[ \frac{1}{n} \sum_{i=1}^{n} \{ \psi(x_i, \theta) - E_0 \psi(x, \theta) \} \right] \tag{3.5}
\]

and so

\[
Z_n^{(j)}(\theta) \equiv \sqrt{n} \left[ \frac{1}{n} \sum_{i=1}^{n} \{ \psi^{(j)}(x_i, \theta) - E_0 \psi^{(j)}(x, \theta) \} \right]. \tag{3.6}
\]
Note that the \( j = 0 \) term in the first sum of (3.4) has been omitted in view of (3.2).

Andrews and Feuerverger (1993) present an iterative inversion of the power series representation (3.4) so as to provide increasingly accurate approximations to the root \( \hat{\theta} \) of the equation (3.1). It is important to note that the terms \( Z_n^{(l)}(\theta_0) \), henceforth referred to as residual sums, appearing in (3.4) are normalized, centered averages and hence are of precise order \( O_P(1) \) provided \( E|\psi^{(l)}(\theta_0)|^2 < \infty \); i.e. are of order \( O_P(1) \) and not of any lower order. (See Hall (1992), pp xii-xiii. for a fuller discussion). Indeed, the use of \( O_P(1) \) random variables of the form (3.5) and (3.6) in (3.4) will conveniently allow the expression of successive iterative approximations to \( \hat{\theta} \) in terms of stochastic asymptotic expansions in powers of the sample size \( n \).

The specific sequence of approximate roots \( \hat{\theta}_l \) to (3.1) is constructed inductively to have the form

\[
\sqrt{n} \cdot (\hat{\theta}_{l+1} - \theta_0) = \delta_0 + \frac{\delta_1}{n^{1/2}} + \frac{\delta_2}{n} + \ldots + \frac{\delta_l}{n^{l/2}} \tag{3.7}
\]

for \( l = 1, 2, \ldots \) where the \( \delta_i \)'s are \( O_P(1) \), depend upon \( n \). and further.

\[
\hat{\theta}_{l+1} \equiv \hat{\theta}_l + \frac{\delta_l}{n^{(l+1)/2}} = \hat{\theta} + O_P(n^{-(l+2)/2}) \tag{3.8}
\]

Each \( \delta_i \) has the special form

\[
\delta_i = \sum_{j=1}^i c_j \cdot Z_{j_1} \ldots Z_{j_{i+1}} \tag{3.9}
\]

that is, a linear combination of products of exactly \( i + 1 \) normalized, centered averages \( Z_j \) of the type (3.5), (3.6). The inductive construction (3.7) is rendered complete by finally defining

\[
\delta_i = \frac{-n^{(i+1)/2}}{E_0 \psi'(X, \theta_0)} \cdot \left\{ \sum_{j=1}^{i+1} \frac{(\hat{\theta}_j)^j}{j!} E_0 \psi^{(j)}(X, \theta_0) + \sum_{j=0}^{i+1} \frac{\hat{\theta}_j^j}{j!} Z_n^{(j)}(\theta_0) \right\} \tag{3.10}
\]

for \( l = 1, 2, \ldots \). It is observed that the representation (3.10) is of order \( O_P(1) \), satisfies (3.8), and has the form given by (3.9). This procedure is a symbolic analogue...
of a quasi Newton-Raphson iteration; see Andrews and Stafford (1993).

For example, setting $l = 2$, the third order expansion of the root $\hat{\theta}$ in iterative form may be expressed as

$$
\hat{\theta}_3 = \theta_0 + \frac{\delta_0}{n^{1/2}} + \frac{\delta_1}{n^{3/2}} + R_4
$$

$$
= \theta_0 + \frac{1}{n^{1/2}} \left\{ -\frac{Z_n(\theta_0)}{E_0\psi'(X, \theta_0)} \right\}
$$

$$
+ \frac{1}{n} \left\{ \frac{Z_n(\theta_0)Z_n^{(1)}(\theta_0)}{[E_0\psi'(X, \theta_0)]^2} - \frac{Z_n(\theta_0)^2E_0\psi^{(2)}(X, \theta_0)}{2E_0\psi'(X, \theta_0)^3} \right\}
$$

$$
+ \frac{1}{n^{3/2}} \left\{ -\frac{Z_n(\theta_0)Z_n^{(1)}(\theta_0)^2}{E_0\psi'(X, \theta_0)^3} + \frac{Z_n(\theta_0)^2Z_n^{(2)}(\theta_0)}{2E_0\psi'(X, \theta_0)^3} \right\}
$$

$$
+ \frac{3Z_n(\theta_0)^2Z_n^{(1)}(\theta_0)E_0\psi^{(2)}(X, \theta_0)}{2E_0\psi'(X, \theta_0)^4}
$$

$$
+ \frac{Z_n(\theta_0)^3E_0\psi^{(3)}(X, \theta_0)}{6E_0\psi'(X, \theta_0)^4}
$$

$$
+ \frac{-Z_n(\theta_0)^3E_0\psi^{(2)}(X, \theta_0)^2}{2E_0\psi'(X, \theta_0)^5} \right\} + O_P(n^{-2}) \right) . \tag{3.11}
$$

We now consider the analogous multiparameter problem. The multivariate M-estimator

$$
\hat{\theta} = (\hat{\theta}_1, \ldots, \hat{\theta}_p)
$$

for the component parameters

$$
\theta = (\theta_1, \ldots, \theta_p)
$$
having the true values

$$\theta_0 = (\theta_{01}, \ldots, \theta_{0p})$$

is the solution of the system of equations:

$$\sum_{i=1}^{n} \psi_j(X_i, \theta) = 0$$

for $j = 1, \ldots, p$. The methods of the previous paragraphs extend directly (care should be used in the matrix-algebra since, for example, $E_0 \psi'(X, \theta_0)$ is now a matrix) to permit us to write each component $\hat{\theta}_j$ of $\hat{\theta}$ in the form (3.7), (3.9) where any of the terms $Z_j$'s appearing in (3.9) can now be any of the terms (3.5) or (3.6) corresponding to any of the $\psi$-functions $\psi_1, \ldots, \psi_p$. Now suppose that we are interested in some smooth function, say $g$, of the component M-estimators so that $\hat{g} \equiv g(\hat{\theta}_1, \ldots, \hat{\theta}_p)$ is viewed as being the statistic of interest, and $g \equiv g(\theta_{01}, \ldots, \theta_{0p})$ as the quantity of inferential interest. Then by substituting approximations which are the vector analogues of (3.7) (for each component of the M-estimator) into a straightforward Taylor expansion for $g$ we are lead to the fact that

$$\sqrt{n} \cdot (\hat{g} - g) = \nu_1 + O_P(n^{-(l+1)/2}) \quad (3.12)$$

where

$$\nu_1 = \gamma_0 + \frac{\gamma_1}{n^{1/2}} + \frac{\gamma_2}{n} + \ldots + \frac{\gamma_l}{n^{l/2}} \quad (3.13)$$

and the $\gamma_i$'s are each $O_P(1)$ linear combinations of products of $i + 1$ normalized, centered averages as in (3.9), that is,

$$\gamma_i = \sum_{j=1}^{J} c_j \cdot Z_{j1} \ldots Z_{j+1} \quad (3.14)$$

As before, this procedure enables us to consider the members of the sequence of
truncated expansions \( \nu_i \) as approximants to the quantity \( \sqrt{n} \cdot (\hat{g} - g) \).

### 3.2.3 Weighting and Decomposing Residual Sums

Conventional bootstrapping is based on the simulated generation of a finite number of resamples drawn with replacement from the original sample. Bootstrap replicates of a statistic \( \hat{\theta} \) of interest are computed and the desired moment associated with the statistic can then be estimated. We observed in the previous section that \( \hat{\theta} \) may be approximated by using the expansion (3.7) together with (3.10). In this section we consider converting (3.7) and (3.10) into bootstrap replicates analytically, that is, without resorting to Monte Carlo simulation. We shall do so by algebraically introducing those weights which have been implicitly generated by repeated sampling with replacement inherent in bootstrapping. We consider the single parameter case below.

From (2.5) and (2.6) it follows that

\[
\sum_{i=1}^{n} x_i^* = \sum_{i=1}^{n} w_i x_i.
\]

Consequently, the introduction of multinomial weights \( w_i \) into (3.6) yields the weighted residual sum

\[
Z_n^{(j)*}(\theta) \equiv \sqrt{n} \left[ \frac{1}{n} \sum_{i=1}^{n} w_i \{ \psi(x_i, \theta) - E_\theta \psi(x_i, \theta) \} \right]. \tag{3.15}
\]

Weighted (i.e. bootstrap) versions of (3.7) and (3.10) may hence be similarly obtained by observing that

\[
\hat{\theta}^* = \hat{\theta}
\]

and

\[
\delta_i^* = \delta_i
\]
with (3.6) replaced by (3.15).

Consider the example in (3.11) in this context. The third order expansion of the bootstrap replicate \( \hat{\theta}^* \) is observed to represent a function of moments and weighted residual sums as well as products of these. That is, the third order expansion of \( \hat{\theta}^* \) can be expressed as a linear combination of products of weighted residual sums. Hence,

\[
\hat{\theta}_3^* = \theta_0 + \frac{\delta_0^*}{n^{1/2}} + \frac{\delta_1^*}{n} + \frac{\delta_2^*}{n^{3/2}} + R_4
\]

\[
= \theta_0 + \frac{1}{n^{1/2}} \left\{ \frac{-Z_n^*(\theta_0)}{E_0 \psi'(X, \theta_0)} \right\}
\]

\[
+ \frac{1}{n} \left\{ \left[ \frac{Z_n^*(\theta_0)Z_n^{(1)*}(\theta_0)}{E_0 \psi'(X, \theta_0)} \right]^2 - \frac{Z_n^*(\theta_0)^2 E_0 \psi^{(2)}(X, \theta_0)}{2 E_0 \psi'(X, \theta_0)^3} \right\}
\]

\[
+ \frac{1}{n^{3/2}} \left\{ \frac{-Z_n^*(\theta_0)Z_n^{(1)*}(\theta_0)^2}{E_0 \psi'(X, \theta_0)^3} + \frac{-Z_n^*(\theta_0)^2 Z_n^{(2)*}(\theta_0)}{2 E_0 \psi'(X, \theta_0)^3} \right\}
\]

\[
+ \frac{3Z_n^*(\theta_0)^2 Z_n^{(1)*}(\theta_0) E_0 \psi^{(2)}(X, \theta_0)}{2 E_0 \psi'(X, \theta_0)^4}
\]

\[
+ \frac{Z_n^*(\theta_0)^3 E_0 \psi^{(3)}(X, \theta_0)}{6 E_0 \psi'(X, \theta_0)^4}
\]

\[
+ \frac{-Z_n^*(\theta_0)^3 E_0 \psi^{(2)}(X, \theta_0)^2}{2 E_0 \psi'(X, \theta_0)^5} \right\} + O_P(n^{-2}) . \tag{3.16}
\]

We now observe that products of residual sums algebraically decompose into linear combinations of simple sums, disjoint sums, and products of these. This fact will be important for subsequent development of the present discussion as it will facilitate computation of moments corresponding to the multinomial weights. A computational procedure for providing such a decomposition for any product of residual sums will be given in Section 3.3. We note that in (3.16) the second order term contains products of two weighted residual sums, the third order term contains products of three weighted residual sums, and so on. Consider \( Z_n^*(\theta_0) \cdot Z_n^{(1)*}(\theta_0) \). Disregarding centering and
normalizing operations and letting \( \psi(x_i, \theta) = \psi_i \) for sake of clarity, we have

\[
Z_n^*(\theta_0) \cdot Z_n^{(1)*}(\theta_0) \equiv \sum_{i=1}^{n} w_i \psi_i \cdot \sum_{j=1}^{n} w_j \psi_j^{(1)}
\]

\[
= \sum_{i=1}^{n} w_i^2 \psi_i \psi_i^{(1)} + \sum_{i \neq j}^{n} w_i w_j \psi_i \psi_j^{(1)}
\]  

(3.17)

Similarly, a product of three weighted residual sums produces, for example,

\[
Z_n^*(\theta_0) \cdot Z_n^{(1)*}(\theta_0) \cdot Z_n^{(2)*}(\theta_0) \equiv \sum_{i=1}^{n} w_i \psi_i \cdot \sum_{j=1}^{n} w_j \psi_j^{(1)} \cdot \sum_{k=1}^{n} w_k \psi_k^{(2)}
\]

\[
= \sum_{i=1}^{n} w_i^3 \psi_i \psi_i^{(1)} \psi_i^{(2)} + \sum_{i \neq j}^{n} \sum_{i \neq j}^{n} w_i w_j w_k \psi_i \psi_j^{(1)} \psi_j^{(2)}
\]

\[
+ \sum_{i \neq j}^{n} \sum_{i \neq j}^{n} w_i w_j w_k \psi_i \psi_j^{(1)} \psi_j^{(2)} + \sum_{i \neq j}^{n} \sum_{i \neq j}^{n} w_i w_j w_k \psi_i \psi_j^{(1)} \psi_j^{(2)}
\]

(3.18)

The combinatorial array of terms in (3.18) suggests a well-known pattern that corresponds to a partitioning of a list of three items. The computational tool \texttt{FP[list]} provides a full partition of a list of any number of items. For example,

\[
\text{FP[\{a, b, c\}]} = \{\{a, b, c\}\}, \{\{b\}, \{a, c\}\}, \{\{a\}, \{b, c\}\}, \{\{a, b\}, \{c\}\}, \{\{a\}, \{b\}, \{c\}\}\}
\]

(3.19)

contains the five partitions from which the respective terms in (3.18) have been constructed. Similarly,

\[
\text{FP[\{a, b, c, d\}]} = \{\{a, b, c, d\}\}, \{\{a, c\}, \{b, d\}\}, \{\{b\}, \{a, c, d\}\}, \{\{b, c\}, \{a\}\}, \{\{b\}, \{c\}, \{d\}\}, \{\{a\}, \{b\}, \{c\}\}, \{\{a\}, \{b, c\}, \{d\}\}, \{\{a\}, \{b\}, \{c\}, \{d\}\}
\]
is used as the basis for decomposing a product of four weighted residual sums into a sum of fifteen component terms, and so on. The above expressions may be readily evaluated using tools for symbolic computation. The procedures described below were based on those developed by Andrews et al. (1993) and subsequent unpublished work. These basic procedures were adapted for the present study. The tool \( FP[\text{list}] \) is described by the following algorithm:

**Algorithm 3.1:**
- Let \( FP[\{a\}] = \{ \{a\} \} \)
- \( FP[\{a,b\}] = \{ \{a, b\}, \{\{a\}, \{b\}\} \} \)
- To find the partitions in \( FP[\{a,b,c\}] \) do the following:
  - To each element of each partition in the preceding list: join \( c \).
  - To each partition in the preceding list: append \( c \).
- Thus, \( FP[\{a,b,c\}] \) is given by (3.19) above.
- Apply this rule to find the partitions of any list \( \{a, b, c, \ldots\} \).

In this section we have indicated that the bootstrap replicate \( \hat{\theta}^* \) may be approximated arbitrarily well by weighted analogues of (3.7) and (3.10), that is, by a truncated series in products of weighted residual sums. Moreover, these products may further be decomposed into linear combinations of simple and disjoint sums. It follows that any smooth function \( g \) of \( \hat{\theta}^* \), for example \( g(\hat{\theta}^*) = (\hat{\theta}^*)^2 \), may also be expressed as a polynomial series in decomposed products of weighted residual sums. Finally, the bootstrap expectation of \( g(\hat{\theta}^*) \) is an expectation over a multinomial vector of weights \( w \) conditional on a sample \( x \) as in (2.16), that is,

\[
E_{w|x}[g(\hat{\theta}^*)]
\]

which can be expressed in a similar form. We consider this in the next section.
3.2.4 Analytical Bootstrap Moments of M-Estimators and their Properties

In this section we consider analytical representations of the bootstrap estimator

$$E_{w|x}[g(\hat{\theta}^*)]$$  \hspace{1cm} (3.22)

and its properties. For example, when $g(\hat{\theta}^*) = \hat{\theta}^*$ or $g(\hat{\theta}^*) = [\hat{\theta}^* - E_{w|x}(\hat{\theta}^*)]^2$ then (3.22) provides the bootstrap estimates of mean or variance given, respectively, by

$$E_{w|x}[\hat{\theta}^*]$$  \hspace{1cm} (3.23)

and

$$E_{w|x}[\hat{\theta}^* - E_{w|x}(\hat{\theta}^*)]^2.$$  \hspace{1cm} (3.24)

As before, the discussion here shall consider only ideal bootstrap estimates, in the sense that $B \to \infty$ (see Section 2.2).

Since expectation is a linear operator, $E_{w|x}[g(\hat{\theta}^*)]$ may be expressed as a truncated polynomial series in decomposed residual sums with some coefficients arising from various moments of the weights $w$. For example, taking the bootstrap expectation over $w$ of $\sum_{i=1}^{n} w_i \psi_i \cdot \sum_{j=1}^{n} w_j \psi_j^{(1)}$ in (3.17) gives

$$E_{w|x}\left[\sum_{i=1}^{n} w_i \psi_i \cdot \sum_{j=1}^{n} w_j \psi_j^{(1)}\right] = E_{w|x}\left[\sum_{i=1}^{n} w_i^2 \psi_i^{(1)}\psi_i^{(1)}\right] + E_{w|x}\left[\sum_{i \neq j}^{n} w_i w_j \psi_i \psi_j^{(1)}\right]$$

$$= E_{w|x}[w_1^2] \cdot \sum_{i=1}^{n} \psi_i \psi_i^{(1)}$$

$$+ E_{w|x}[w_1 w_2] \cdot \sum_{i \neq j}^{n} \psi_i \psi_j^{(1)}. \hspace{1cm} (3.25)$$

Note that the second step in (3.25) is possible because the multinomial weights $(w_1, \ldots, w_n)$ are identically distributed, and though not independent, are exchangeable. That is, $E[w_i \cdot w_j] = E[w_1 \cdot w_2]$ for $i \neq j$ but $E[w_1 \cdot w_2] \neq E[w_1] \cdot E[w_2]$. The
coefficients $E_{w|x}[w_i^2]$ and $E_{w|x}[w_1 w_2]$ are known and may be simply computed from the moment generating function of the multinomial distribution. Moreover, the disjoint sum in the second term on the right side of (3.25) may be expressed as a linear combination of products of simple sums. Similar evaluations would take place when evaluating the bootstrap expectation of any product of weighted residual sums.

The series evaluation of the ideal bootstrap estimator $E_{w|x}[g(\hat{\theta}^*)]$ to any level of precision thus depends solely on the specification of the appropriate $\psi$-functions defining the M-estimator $\hat{\theta}$ while numerical evaluation would require observation of the sample $(x_1, \ldots, x_n)$. The specified $\psi$-functions must be differentiable up to the desired order of the expansion of $\hat{\theta}$. For example, the bootstrap estimates of mean and variance, given respectively by (3.23) and (3.24), can each be expressed analytically as a linear combination of residual sums - symbolic evaluation in each case would depend only on the specification of the appropriate $\psi$-functions. and subsequent numerical evaluation, if desired, would require the sample $x = (x_1, \ldots, x_n)$.

Finally, one may consider properties of the series representation of $E_{w|x}[g(\hat{\theta}^*)]$. Indeed, any expectation - over $x$ - of such a series may be calculated directly. For example, the mean and variance of the bootstrap estimator $E_{w|x}[g(\hat{\theta}^*)]$ are given by

$$E_x \left[ E_{w|x}[g(\hat{\theta}^*)] \right] \quad (3.26)$$

and

$$Var_x \left[ E_{w|x}[g(\hat{\theta}^*)] \right] \quad (3.27)$$

and such expressions can be evaluated in a straightforward manner.

The salient feature of the representations described in this section is that they can be expressed in symbolic form. This precludes necessity for the use of Monte Carlo simulation. In the next section we shall describe and illustrate several computational procedures developed using Mathematica that permit analytic expression of the bootstrap quantities considered above. This methodology will then be applied to the case of the correlation coefficient.
3.3 Computational Tools

In this section we will describe and illustrate a number of computational procedures developed using Mathematica that permit analytic expression of various quantities considered in the previous two sub-sections.

In Section 3.2.2 it was observed that (3.7) together with (3.10) provide increasingly accurate iterative approximations to a single-dimensional M-estimator $\hat{\theta}$. An algorithm has been implemented in Mathematica that computes (3.7), (3.10). This function is called ThetaHat[i] and it returns the expansion for $\hat{\theta}$, correct to order $n^{-i/2}$. For example, the second order approximation to the root $\hat{\theta}$ is computed as

$$\text{ThetaHat}[2] = \theta - \frac{Z_n[\psi(X, \theta)]}{\sqrt{n} C \text{um}[\psi'(X, \theta)]} + \frac{Z_n[\psi(X, \theta)] Z_n[\psi^{(1)}(X, \theta)]}{C \text{um}[\psi'(X, \theta)]^2} - \frac{Z_n[\psi(X, \theta)]^2 C \text{um}[\psi^{(2)}(X, \theta)]}{2 C \text{um}[\psi'(X, \theta)]^3} / n. \quad (3.28)$$

This corresponds to the sum of the first four terms in (3.11) where $Z_n(\psi(X, \theta)) = Z_n(\theta_0)$ and expectations are represented as cumulants. When $i = 3$ an expression with nine terms is returned. When $i = 4$ the function returns an expression with 19 terms. Similarly, the 6-th order expansion contains 75 terms. The expansion for the M-estimator $\hat{\theta}$ may thus be derived for any order. The code for the tool ThetaHat[i] is available in Appendix A.2 in a file called Bootcum.m.

Now the bootstrap quantity $\hat{\theta}^*$, or any power of this, can always be represented symbolically as a linear combination of products of weighted residual sums (factors in $Z_n^*$); these are now functions in the bootstrap resample $X^*$. Thus, the bootstrap version of (3.28) would look identical with (3.28) where $Z_n$ and $X$ are replaced by $Z_n^*$ and $X^*$. Moreover, this analytic bootstrap quantity represents (3.16) truncated to second order.

The function BootCum[$\hat{\theta}$][i,j] returns the i-th bootstrap cumulant of some esti-
mator $\hat{\theta}$ expressed as an expansion correct to order $n^{-j/2}$. This calculation represents the kernel upon which the methodology described in the present chapter is based. In particular, $\text{BootCum}[\hat{\theta}][1, 2]$ represents the analytical expansion of the bootstrap expectation $E_{w|x}(\hat{\theta}^*)$ correct to second order, that is,

$$E_{w|x}(\hat{\theta}^*) = \text{BootCum}[\hat{\theta}][1, 2] + O_p(n^{-1}) \quad (3.29)$$

The code $\text{BootCum}[\hat{\theta}][2, 2]$ represents the analytical expansion of the bootstrap variance $\text{Var}_{w|x}(\hat{\theta}^*)$ correct to second order, that is,

$$\text{Var}_{w|x}(\hat{\theta}^*) = \text{BootCum}[\hat{\theta}][2, 2] + O_p(n^{-1}) \quad (3.30)$$

For example, the second order expansions of the bootstrap estimates of mean and variance of $\hat{\theta}$ are displayed symbolically, respectively, as

$$\text{BootCum}[\hat{\theta}][1, 2] = \theta + \left[ \frac{\text{Cum}(\psi(X^*, \theta), \psi'(X^*, \theta))}{\text{Cum}[\psi'(X, \theta)]^2} \right]$$

$$- \frac{\text{Cum}(\psi(X^*, \theta), \psi(X^*, \theta)) \text{Cum}[\psi^{(2)}(X, \theta)]}{2 \text{Cum}[\psi'(X, \theta)]^3} / n \quad (3.31)$$

and

$$\text{BootCum}[\hat{\theta}][2, 2] = \frac{\text{Cum}(\psi(X^*, \theta), \psi(X^*, \theta))}{n \text{Cum}[\psi'(X, \theta)]^2} . \quad (3.32)$$

The code for the tool $\text{BootCum}[\hat{\theta}][i, j]$ is available in Appendix A.2 in a file called $\text{Bootcum.m}$.

The main steps of the algorithms associated with $\text{ThetaHat}[i]$ and $\text{BootCum}[\hat{\theta}][i, j]$ are given below.

We suppose that $\hat{\theta}$ is the root of the equation $f(\hat{\theta}) = 0$. The solution $\hat{\theta}$ is expressed as a sequence of increasingly accurate series approximations. A starting value $\hat{\theta}_0$ must be provided. The algorithm is a Newton iteration as follows:
Algorithm 3.2 (ThetaHat[i]):

- Provide a starting value $\hat{\theta}_0$.
- Compute $\hat{\theta}_1 = \frac{-f_1(\hat{\theta}_0)}{f_0(\theta_0)}$
- In general, $\hat{\theta}_i = \frac{-f_i(\hat{\theta}_{i-1})}{f_0(\theta_0)}$

where $f_i(\hat{\theta}_j)$ is a series expansion of $f(\hat{\theta})$ correct to order $n^{-i/2}$.

Algorithm 3.3 (BootCum[\hat{\theta}][i,j]):

- A statistic defined by estimating equations can be approximated to any order by a finite linear combination of products of normalized, centered sums of independent and identically distributed random variables.
- A bootstrap replicate of such a statistic, or any smooth function of this, may be similarly (analytically) expressed as a finite linear combination of products of weighted sums. The weights arise from the multinomial distribution. The weighted sums are decomposed into simple sums and products of sums involving distinct observations (referred to as disjoint sums).
- The bootstrap expectation of the statistic is obtained by considering the expectation of the series expansion with respect to the multinomial weights arising in the preceding step. The moments of the weights are found from the moment generating function of the multinomial distribution.
- Products of sums involving distinct observations are re-expressed as linear combinations of simple sums and ordinary products of sums.

Finally, we consider the procedure that evaluates properties such as (3.26) and (3.27). The function Cum[BootCum[\hat{\theta}][i]][j,k] returns a series expansion correct to order $n^{-k/2}$ of the $j$-th cumulant - over $x$ - of the $i$-th bootstrap cumulant of the
statistic $\hat{\theta}$. In particular,

$$E_x \left[ E_w(x|\hat{\theta}^*) \right] = \text{Cum}[\text{BootCum}[\hat{\theta}][1]][1, 2] + O_p(n^{-1}) \quad (3.33)$$

$$\text{Var}_x \left[ E_w(x|\hat{\theta}^*) \right] = \text{Cum}[\text{BootCum}[\hat{\theta}][1]][2, 2] + O_p(n^{-1}) \quad (3.34)$$

$$E_x \left[ \text{Var}_w(x|\hat{\theta}^*) \right] = \text{Cum}[\text{BootCum}[\hat{\theta}][2]][1, 2] + O_p(n^{-1}) \quad (3.35)$$

$$\text{Var}_x \left[ \text{Var}_w(x|\hat{\theta}^*) \right] = \text{Cum}[\text{BootCum}[\hat{\theta}][2]][2, 2] + O_p(n^{-1}) \quad (3.36)$$

The functional representations given in this section are generic and apply to any bootstrapped M-estimator $\hat{\theta}^*$ except those which cannot be expressed in terms of smooth functions of averages. An example of this is the median; this case was treated alternatively in Chapter 2. In the following section we shall illustrate the functions developed here by applying them to the case of the correlation coefficient.

### 3.4 The Case of the Correlation Coefficient

#### 3.4.1 Introduction

In this section we illustrate the methodology developed in the preceding sections by considering the case of a bivariate correlation coefficient using both real and simulated data: Efron and Tibshirani's (1993) Law School data relating LSAT and GPA scores of students entering 15 American law schools and a data set simulated from a bivariate standardized Gaussian distribution.
3.4.2 Computational Tools

The symbolic function Cor\[i\] expands the bivariate sample correlation coefficient given by

\[
\hat{\rho} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{[\sum_{i=1}^{n}(x_i - \bar{x})^2]^{\frac{1}{2}} \cdot [\sum_{i=1}^{n}(y_i - \bar{y})^2]^{\frac{1}{2}}} \tag{3.37}
\]

using a Taylor series in residual sums correct to \(i\)-th order. Since the correlation coefficient is invariant with respect to linear transformations we may simplify the expansion by assuming, without loss of generality, that \(E(x) = E(y) = 0\) and \(E(x^2) = E(y^2) = 1\). For example, the 2nd order expansion of \(\hat{\rho}\) is given analytically as

\[
\text{Cor}[2] = \rho + \left[ \frac{-\rho Z(n)(X^2)}{2} + Z(n)(X Y) - \frac{\rho Z(n)(Y^2)}{2} \right] / \sqrt{n} +
\]

\[
\left[ \frac{\rho Z(n)(X)^2}{2} + \frac{3\rho Z(n)(X^2)^2}{8} - Z(n)(X) Z(n)(Y) +
\right. \\
\left. \frac{\rho Z(n)(Y)^2}{2} - \frac{Z(n)(X^2) Z(n)(X Y)}{2} + \frac{\rho Z(n)(X^2) Z(n)(Y^2)}{4} -
\right. \\
\left. \frac{Z(n)(X Y) Z(n)(Y^2)}{2} + \frac{3\rho Z(n)(Y^2)^2}{8} \right] / n \tag{3.38}
\]

where, for example, the residual sum \(Z(n)(X)\) according to (3.5) is given by

\[
Z(n)(X) = \sqrt{n} \left[ \bar{X} - E(X) \right]. \tag{3.39}
\]

The main steps of the algorithm associated with Cor\[i\] are as follows:

\textbf{Algorithm 3.4:}

- Assume w.l.o.g. that \(E(x) = E(y) = 0\) and \(E(x^2) = E(y^2) = 1\).
- Express \(\text{Cov}(x, y)\) as a series expansion in terms of residual sums of the form (3.39).
• Express $\text{Cov}(x, x)$ and $\text{Cov}(y, y)$ as series expansions in terms of residual sums.

• The product of the series expansions of $\text{Cov}(x, x)^{-1/2}$ and $\text{Cov}(y, y)^{-1/2}$ is then multiplied by the expansion for $\text{Cov}(x, y)$ to produce terms in the series expansion of $\hat{\rho} = \text{Cov}(x, y) \cdot \text{Cov}(x, x)^{-1/2} \cdot \text{Cov}(y, y)^{-1/2}$.

The code for the tool $\text{Cor}[i]$ may be found in Appendix A.2 in a file called $\text{Cor1.m}$. Subsequent code used for the asymptotic bootstrap expansions associated with the correlation as derived below can be found in the file $\text{Cor2.m}$.

The tool $\text{Cum}[\text{Cor}][i, j]$ returns the $i$-th cumulant - over the $(x, y)$ distribution - of $\hat{\rho}$ expressed as an expansion correct to order $n^{-1/2}$. (In the notation, $x$, $y$ and $x$, $y$ are used interchangeably.) For example, the mean $E_{x,y}(\hat{\rho})$ and the variance $\text{Var}_{x,y}(\hat{\rho})$ expressed as expansions to order $n^{-1}$ are given, respectively, by

$$
\text{Cum}[\text{Cor}][1, 2] = \rho - \frac{\rho}{2n} + \frac{\rho^3}{2n^2} + \frac{3\rho \text{Cum}(X, X, X)}{8n} - \frac{\text{Cum}(X, X, X, X)}{2n} + \frac{\rho \text{Cum}(X, X, Y, Y)}{4n} - \frac{\text{Cum}(X, Y, Y, Y)}{2n} + \frac{3\rho \text{Cum}(Y, Y, Y, Y)}{8n}.
$$

(3.40)

and

$$
\text{Cum}[\text{Cor}][2, 2] = \frac{1}{n} - \frac{2\rho^2}{n} + \frac{\rho^4}{n} + \frac{\rho^2 \text{Cum}(X, X, X)}{4n} - \frac{\rho \text{Cum}(X, X, X, Y)}{n} + \frac{\text{Cum}(X, X, Y, Y)}{2n} \frac{\rho^2 \text{Cum}(X, X, Y, Y)}{2n} - \frac{\rho \text{Cum}(X, Y, Y, Y)}{n} + \frac{\rho^2 \text{Cum}(Y, Y, Y, Y)}{4n}.
$$

(3.41)

where, for example, $\text{Cum}(X, X, X)$ represents the third cumulant of $X$, that is.

$$
\text{Cum}(X, X, X) = E(X - \mu)^3
$$
if $\mu = E(X)$. See, for example, McCullagh (1987) for a discussion of multivariate cumulants.

The function `BootCum[Cor][i, j]` returns the $i$-th bootstrap cumulant of the correlation coefficient expressed as an expansion correct to order $n^{-j/2}$. For example, the 2nd order analytic expressions for the bootstrap mean

$$E_{w|x,y}(\hat{\rho}^*)$$

and the bootstrap variance

$$Var_{w|x,y}(\hat{\rho}^*) = E_{w|x,y}[\hat{\rho}^{*2}] - [E_{w|x,y}(\hat{\rho}^*)]^2$$

of the correlation coefficient are given, respectively, as

$$\text{BootCum[Cor][1, 2]} = \rho - \frac{\rho}{2n} + \frac{\rho^3}{2n} + \frac{3 \rho \text{Cum}(X, X, X, X)}{8n} -$$

$$\frac{\text{Cum}(X, X, X, Y)}{2n} + \frac{\rho \text{Cum}(X, X, Y, Y)}{4n} - \frac{\text{Cum}(X, Y, Y, Y)}{2n} +$$

$$\frac{3 \rho \text{Cum}(Y, Y, Y, Y)}{8n} + \frac{\rho \text{Z}(n)(X)^2}{2n} - \frac{\rho \text{Z}(n)(X^2)}{2\sqrt{n}} +$$

$$\frac{3 \rho \text{Z}(n)(X^2)^2}{8n} - \frac{\text{Z}(n)(X) Z(n)(Y)}{n} + \frac{\rho \text{Z}(n)(Y)^2}{2n} + \frac{\text{Z}(n)(X Y)}{\sqrt{n}} -$$

$$\frac{\text{Z}(n)(X^2) Z(n)(X Y)}{2n} - \frac{\rho \text{Z}(n)(Y)^2}{2\sqrt{n}} + \frac{\rho \text{Z}(n)(X^2) Z(n)(Y^2)}{4n} -$$

$$\frac{\text{Z}(n)(X Y) Z(n)(Y^2)}{2n} + \frac{3 \rho \text{Z}(n)(Y^2)^2}{8n}$$

(3.45)
CHAPTER 3. BOOTSTRAPPED M-ESTIMATORS

and

$$\text{BootCum}[\text{Cor}][2,2] = \frac{1}{n} - \frac{2 \rho^2}{n} + \frac{\rho^4}{n} + \frac{\rho^2 \text{Cum}(X,X,X,X)}{4n} - \frac{\rho \text{Cum}(X,X,X,Y)}{n} + \frac{\text{Cum}(X,X,Y,Y)}{n} + \frac{\rho^2 \text{Cum}(X,X,Y,Y)}{2n} - \frac{\rho \text{Cum}(X,Y,Y,Y)}{n} + \frac{\rho^2 \text{Cum}(Y,Y,Y,Y)}{4n}. \quad (3.46)$$

The 4th order expansion of the bootstrap variance of the correlation (3.44) is much lengthier than (3.46) and possesses 226 terms. We provide the first and final few terms of this expansion as follows:

$$\text{BootCum}[\text{Cor}][2,4] = n^{-2} + \frac{1}{n} + \frac{7 \rho^2}{2n^2} - \frac{2 \rho^2}{n} - \frac{10 \rho^4}{n^2} + \frac{\rho^4}{n} + \frac{11 \rho^6}{2n^2} - \frac{3 \rho^2 \text{Cum}(X,X,X)^2}{2n^2} + \frac{5 \rho \text{Cum}(X,X,X) \text{Cum}(X,X,Y)}{n^2} - \frac{2 \text{Cum}(X,X,Y)^2}{n^2} - \frac{5 \rho^2 \text{Cum}(X,X,Y)^2}{2n^2} - \frac{2 \text{Cum}(X,X,X) \text{Cum}(X,Y,Y)}{n^2} + \frac{6 \rho \text{Cum}(X,X,Y) \text{Cum}(X,Y,Y)}{n^2} - \frac{2 \text{Cum}(X,Y,Y)^2}{n^2} - \frac{5 \rho^2 \text{Cum}(X,Y,Y)^2}{2n^2} - \frac{2 \text{Cum}(X,X,Y) \text{Cum}(Y,Y,Y)}{n^2} + \frac{5 \rho \text{Cum}(X,Y,Y) \text{Cum}(Y,Y,Y)}{n^2} - \frac{3 \rho^2 \text{Cum}(Y,Y,Y)^2}{2n^2} - (\ldots + 198 \text{ TERMS} + \ldots) -$$
Note that the 4th order expansion (3.47) contains residual sums whereas the 2nd order expansion (3.46) does not. Moreover, 2nd and 3rd order expansions of (3.44) are identical, 4th and 5th order expansions are identical, etc.

Our methodology now permits us to consider properties, indeed, cumulants of the bootstrap quantities (3.43) and (3.44). For illustration we consider the mean and variance - over x and y - of (3.43) and (3.44). In practice, any cumulant expanded to an arbitrary order of accuracy may be considered. The function

\[ \text{Cum}[\text{BootCum}[\text{Cor}][i]][j,k] \]

computes a series expansion correct to order \( n^{-k/2} \) of the j-th cumulant - over x and y - of the i-th bootstrap cumulant of the correlation coefficient. In particular.

\[ E_{x,y} \left[ E_{w|x,y}(\hat{\rho}^*) \right] = \text{Cum}[\text{BootCum}[\text{Cor}][1]][1, 2] + O_p(n^{-1}) \] (3.48)

\[ \text{Var}_{x,y} \left[ E_{w|x,y}(\hat{\rho}^*) \right] = \text{Cum}[\text{BootCum}[\text{Cor}][1]][2, 2] + O_p(n^{-1}) \] (3.49)

\[ E_{x,y} \left[ \text{Var}_{w|x,y}(\hat{\rho}^*) \right] = \text{Cum}[\text{BootCum}[\text{Cor}][2]][1, 2] + O_p(n^{-1}) \] (3.50)
\[ \text{Var}_{x,y} \left[ \text{Var}_{w|x,y}(\hat{\rho}^*) \right] = \text{Cum} [\text{BootCum} [\text{Cor}] [2]] [2,2] + O_p(n^{-1}) \quad (3.51) \]

For example, the second order expansions considered in (3.48) - (3.50) are given, respectively, as:

\[
\text{Cum} [\text{BootCum} [\text{Cor}] [1]] [1,2] = \rho - \frac{\rho}{n} + \frac{\rho^3}{n} + \frac{3 \rho \text{Cum}(X,X,X,X)}{4n} - \]
\[
\frac{\text{Cum}(X,X,X,Y)}{n} + \frac{\rho \text{Cum}(X,X,Y,Y)}{2n} - \]
\[
\frac{\text{Cum}(X,Y,Y,Y)}{n} + \frac{3 \rho \text{Cum}(Y,Y,Y,Y)}{4n} \quad (3.52) \]

\[
\text{Cum} [\text{BootCum} [\text{Cor}] [1]] [2,2] = \frac{1}{n} - \frac{2 \rho^2}{n} + \frac{\rho^4}{n} + \frac{\rho^2 \text{Cum}(X,X,X,X)}{4n} - \]
\[
\frac{\rho \text{Cum}(X,X,X,Y)}{n} + \frac{\text{Cum}(X,X,Y,Y)}{n} + \]
\[
\frac{\rho^2 \text{Cum}(X,X,Y,Y)}{2n} - \frac{\rho \text{Cum}(X,Y,Y,Y)}{n} + \]
\[
\frac{\rho^2 \text{Cum}(Y,Y,Y,Y)}{4n} \quad (3.53) \]

and

\[
\text{Cum} [\text{BootCum} [\text{Cor}] [2]] [1,2] = \frac{1}{n} - \frac{2 \rho^2}{n} + \frac{\rho^4}{n} + \frac{\rho^2 \text{Cum}(X,X,X,X)}{4n} - \]
\[
\frac{\rho \text{Cum}(X,X,X,Y)}{n} + \frac{\text{Cum}(X,X,Y,Y)}{n} + \]
\[
\frac{\rho^2 \text{Cum}(X,X,Y,Y)}{2n} - \frac{\rho \text{Cum}(X,Y,Y,Y)}{n} + \]
\[
\frac{\rho^2 \text{Cum}(Y, Y, Y, Y)}{4n}.
\]

We observe that

\[
\text{Cum[BootCum[Cor][1]][2, 2]} = \text{Cum[BootCum[Cor][2]][1, 2]}
\]

\[
= \text{BootCum[Cor][2, 2]}.
\]

The corresponding 4th order and higher order expansions are not identical, in general, due to the appearance of residual sums.

Table 3.1 below summarizes the six quantities for which symbolic expressions associated with the correlation coefficient have been considered in this chapter. These represent the bootstrap mean and bootstrap variance and the mean and variance - over \( x \) and \( y \) - for each of these. However, the computational tools developed here can provide symbolic expansions for any bootstrap cumulant of the sample correlation to any desired order as well as properties of these.

Table 3.1: Bootstrap estimators and related properties for the correlation coefficient for which analytic expansions are derived

<table>
<thead>
<tr>
<th>Properties</th>
<th>Bootstrap Mean (( \hat{\rho} ))</th>
<th>Bootstrap Variance (( \hat{\rho} ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>( E_{x,y}[E_{w</td>
<td>x,y}(\hat{\rho}^*)] )</td>
</tr>
<tr>
<td>Variance</td>
<td>( \text{Var}<em>{x,y}[E</em>{w</td>
<td>x,y}(\hat{\rho}^*)] )</td>
</tr>
</tbody>
</table>

The series expansions associated with Table 3.1 for the correlation coefficient constitute symbolic representations, in fact, formulas, which may be evaluated for any bi-variate set of the data for which calculations of the correlation, its bootstrap moments, as well as properties of these are desired. These expansions, such as the ones in (3.45) - (3.50), represent analytic bootstrap expressions which may be evaluated numerically for any pair of distributions \( F \) and \( G \) associated with the dependent
variables $x$ and $y$ for which cumulants are known. In particular, we will use the empirical (nonparametric) distribution functions $F_n$ and $G_n$ as defined in Chapter 2, Section 2.2. With reference to the law school data, for example, we make the following empirical substitutions in expressions (3.45)-(3.50):

1. $n \to 15$
2. $\rho \to \hat{\rho} = 0.776374$
3. All residual sums vanish when moments are replaced by empirical moments. For example, $E(X) \to \bar{X}$ yields $Z(n)(X) = 0$ in (3.39). For distributions other than the empirical distribution this is not always the case. If, for example, a standard normal parent is assumed then not all residual sums vanish.
4. Cumulants $\to$ empirical cumulants. For example, the third cumulant

$$Cum(X, X, X) = E(X - \mu)^3$$

$$= E(X^3) - 3\mu E(X^2) + 2\mu^3$$

is replaced by the third empirical cumulant

$$EmpCum(X, X, X) = \frac{SUM(X^3)}{n} - \frac{3 SUM(X) SUM(X^2)}{n^2} + \frac{2 SUM(X)^3}{n^3}$$

where, for example, $SUM[X^2] = \sum_{i=1}^{n} x_i^2$.

An advantage of the analytical approach given above is that once a bootstrap expansion to a specified order has been calculated, the derivational work for the particular problem at hand has been done for all time - it remains merely to numerically evaluate the formula for any bivariate data set for which estimation of the correlation coefficient is desired. This contrasts with the conventional approach which requires that a bootstrap variance estimate for the correlation coefficient based on Monte Carlo simulation be determined each time that a different data set is encountered.
3.4.3 Numerical Results

We will numerically evaluate the second and fourth order expansions for the six bootstrap expressions given in Table 3.1 for two bi-variate data sets described below. The methodology is illustrated for real and simulated data sets of different sizes. Table 3.2 shows a random sample of size \( n = 15 \) drawn from a population of \( N = 82 \) American law schools. What is shown on the left side are two measurements made on the entering classes of 1973 for each school in the sample: LSAT, the average score of the class on a national law test, and GPA, the average undergraduate grade point average achieved by the members of the class. The data were obtained from Efron and Tibshirani (1993, p. 19). The symbolic expansions given previously assume standardized variables, and hence the same data shown on the right side of Table 3.2 have also been standardized.

<table>
<thead>
<tr>
<th>LSAT</th>
<th>GPA</th>
<th>LSAT.stand</th>
<th>GPA.stand</th>
</tr>
</thead>
<tbody>
<tr>
<td>576</td>
<td>3.39</td>
<td>-0.600997</td>
<td>1.25538</td>
</tr>
<tr>
<td>635</td>
<td>3.30</td>
<td>0.860219</td>
<td>0.872812</td>
</tr>
<tr>
<td>558</td>
<td>2.81</td>
<td>-1.04679</td>
<td>-1.21003</td>
</tr>
<tr>
<td>578</td>
<td>3.03</td>
<td>-0.551465</td>
<td>-0.274879</td>
</tr>
<tr>
<td>666</td>
<td>3.44</td>
<td>1.62798</td>
<td>1.46791</td>
</tr>
<tr>
<td>580</td>
<td>3.07</td>
<td>-0.501932</td>
<td>-0.104851</td>
</tr>
<tr>
<td>555</td>
<td>3.00</td>
<td>-1.12109</td>
<td>-0.4024</td>
</tr>
<tr>
<td>661</td>
<td>3.43</td>
<td>1.50414</td>
<td>1.4254</td>
</tr>
<tr>
<td>651</td>
<td>3.36</td>
<td>1.25648</td>
<td>1.12785</td>
</tr>
<tr>
<td>605</td>
<td>3.13</td>
<td>0.117227</td>
<td>0.150192</td>
</tr>
<tr>
<td>653</td>
<td>3.12</td>
<td>1.30601</td>
<td>0.107685</td>
</tr>
<tr>
<td>575</td>
<td>2.74</td>
<td>-0.625764</td>
<td>-1.50758</td>
</tr>
<tr>
<td>545</td>
<td>2.76</td>
<td>-1.36875</td>
<td>-1.42257</td>
</tr>
<tr>
<td>572</td>
<td>2.88</td>
<td>-0.700063</td>
<td>-0.912485</td>
</tr>
<tr>
<td>594</td>
<td>2.96</td>
<td>-0.155203</td>
<td>-0.572429</td>
</tr>
</tbody>
</table>

Tables 3.3 and 3.4 display \( n = 100 \) simulated bivariate observations generated from a standard Gaussian distribution. The data was also standardized to produce exact zero sample means and unit sample variances for each of the variables.

Table 3.5 gives the numerical results obtained by calculating the 2nd and 4th
Table 3.3: $n = 100$ simulated bivariate observations from $N(0,1)$: variable $x$

<table>
<thead>
<tr>
<th>$x$</th>
<th>-0.956714</th>
<th>0.686892</th>
<th>-0.564691</th>
<th>-1.14333</th>
<th>-0.0559606</th>
<th>-0.874619</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.436588</td>
<td>1.12566</td>
<td>-0.793298</td>
<td>0.297573</td>
<td>0.249147</td>
<td>0.0330676</td>
<td>1.62495</td>
</tr>
<tr>
<td>0.225940</td>
<td>-1.21993</td>
<td>0.318041</td>
<td>0.291848</td>
<td>0.0120344</td>
<td>1.62495</td>
<td>0.0036076</td>
</tr>
<tr>
<td>1.498340</td>
<td>-1.8793</td>
<td>-1.30035</td>
<td>-1.40383</td>
<td>-1.37385</td>
<td>0.392384</td>
<td>1.62495</td>
</tr>
<tr>
<td>-1.087850</td>
<td>1.08452</td>
<td>-0.948535</td>
<td>-0.750636</td>
<td>-0.528461</td>
<td>-1.03515</td>
<td>0.0036076</td>
</tr>
<tr>
<td>0.299149</td>
<td>-0.0431774</td>
<td>-1.81281</td>
<td>0.933136</td>
<td>-0.914287</td>
<td>-0.353526</td>
<td>1.62495</td>
</tr>
<tr>
<td>-0.0696765</td>
<td>-1.11197</td>
<td>0.239538</td>
<td>-0.854338</td>
<td>2.07333</td>
<td>-0.0952752</td>
<td>1.62495</td>
</tr>
<tr>
<td>-0.6538</td>
<td>-0.248606</td>
<td>-0.857544</td>
<td>0.284146</td>
<td>1.71851</td>
<td>-0.875886</td>
<td>1.62495</td>
</tr>
<tr>
<td>-0.935791</td>
<td>0.0546828</td>
<td>-0.662648</td>
<td>-0.130106</td>
<td>-2.59892</td>
<td>1.29063</td>
<td>1.62495</td>
</tr>
<tr>
<td>0.61013</td>
<td>0.0920582</td>
<td>-0.665875</td>
<td>-0.394396</td>
<td>-0.426476</td>
<td>0.373098</td>
<td>1.62495</td>
</tr>
<tr>
<td>0.0048362</td>
<td>-1.41674</td>
<td>0.926444</td>
<td>0.353601</td>
<td>-0.203354</td>
<td>0.320726</td>
<td>1.62495</td>
</tr>
<tr>
<td>0.161122</td>
<td>0.471373</td>
<td>-1.17339</td>
<td>0.489503</td>
<td>0.39107</td>
<td>1.95425</td>
<td>1.62495</td>
</tr>
<tr>
<td>0.0301679</td>
<td>0.511621</td>
<td>-1.41243</td>
<td>0.425221</td>
<td>-0.558879</td>
<td>-1.2051</td>
<td>1.62495</td>
</tr>
<tr>
<td>1.18088</td>
<td>0.762738</td>
<td>1.36058</td>
<td>0.25875</td>
<td>1.0521</td>
<td>0.665511</td>
<td>1.62495</td>
</tr>
<tr>
<td>-0.0377158</td>
<td>0.353328</td>
<td>0.0265674</td>
<td>-0.51515</td>
<td>0.887074</td>
<td>-1.72355</td>
<td>1.62495</td>
</tr>
<tr>
<td>-0.262615</td>
<td>-0.0142491</td>
<td>2.73824</td>
<td>-0.895298</td>
<td>0.868991</td>
<td>1.67417</td>
<td>1.62495</td>
</tr>
<tr>
<td>1.37254</td>
<td>0.521282</td>
<td>2.86879</td>
<td>0.653835</td>
<td>1.0521</td>
<td>0.665511</td>
<td>1.62495</td>
</tr>
</tbody>
</table>

order expansions associated with the six moments identified in Table 3.1 for each of the data sets. The classical results are also presented where $\hat{\rho}$ has been computed using (3.37) and $E_{x,y}(\hat{\rho})$. $Var_{x,y}(\hat{\rho})$ have been computed to order $n^{-1}$, for example, by (3.40) and (3.41). The asymptotic variance of the correlation is given by

$$Var_{x,y}(\hat{\rho}) = \frac{(1 - \rho^2)^2}{n}.$$  \hfill (3.55)

Tables 3.6 and 3.7 document the conventional Monte Carlo bootstrap estimates of the mean and standard error of the sample correlation coefficient (the standard error of each estimate is shown in brackets alongside). These are then compared with the corresponding analytically computed bootstrap mean and standard error of the correlation given in lines 1 and 4 of Table 3.5. The Mathematica and Splus code underlying the computation of the results within these tables is given in Appendix A.2.

Note that calculations exhibited in Table 3.5 were carried out using Mathematica
Table 3.4: \( n = 100 \) simulated bivariate observations from \( N(0, 1) \): variable \( y \)

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.97725</td>
<td>1.10446</td>
<td>1.32156</td>
<td>-0.715398</td>
<td>0.1296</td>
<td>-1.40333</td>
<td></td>
</tr>
<tr>
<td>-1.11252</td>
<td>-0.44703</td>
<td>-1.15785</td>
<td>0.12367</td>
<td>-0.166621</td>
<td>0.385073</td>
<td></td>
</tr>
<tr>
<td>0.730527</td>
<td>-1.66333</td>
<td>0.345996</td>
<td>-1.74982</td>
<td>-0.485517</td>
<td>0.937286</td>
<td></td>
</tr>
<tr>
<td>1.63829</td>
<td>-1.13556</td>
<td>-1.44134</td>
<td>0.500906</td>
<td>-0.99789</td>
<td>0.228176</td>
<td></td>
</tr>
<tr>
<td>-1.1454</td>
<td>1.73745</td>
<td>-1.60854</td>
<td>-1.06215</td>
<td>-1.07815</td>
<td>-1.31326</td>
<td></td>
</tr>
<tr>
<td>0.11412</td>
<td>1.28792</td>
<td>-0.709178</td>
<td>0.844933</td>
<td>0.291268</td>
<td>-0.776762</td>
<td></td>
</tr>
<tr>
<td>0.71912</td>
<td>-0.94738</td>
<td>-1.66528</td>
<td>0.00613181</td>
<td>0.625405</td>
<td>0.310604</td>
<td></td>
</tr>
<tr>
<td>-0.88552</td>
<td>0.635544</td>
<td>-0.052398</td>
<td>-0.459871</td>
<td>0.354249</td>
<td>-0.927752</td>
<td></td>
</tr>
<tr>
<td>-0.52729</td>
<td>0.158014</td>
<td>-0.0303813</td>
<td>0.128091</td>
<td>-1.5813</td>
<td>1.36929</td>
<td></td>
</tr>
<tr>
<td>0.799488</td>
<td>-0.615692</td>
<td>-0.853587</td>
<td>-0.0215823</td>
<td>0.232778</td>
<td>-0.0593525</td>
<td></td>
</tr>
<tr>
<td>-0.277299</td>
<td>-1.26071</td>
<td>1.02008</td>
<td>1.151</td>
<td>-0.161112</td>
<td>0.405654</td>
<td></td>
</tr>
<tr>
<td>-0.54128</td>
<td>0.876874</td>
<td>-1.24908</td>
<td>1.74748</td>
<td>0.36598</td>
<td>0.830054</td>
<td></td>
</tr>
<tr>
<td>1.85925</td>
<td>0.0531251</td>
<td>-0.930298</td>
<td>0.522718</td>
<td>0.413684</td>
<td>-1.12877</td>
<td></td>
</tr>
<tr>
<td>-0.32736</td>
<td>1.66135</td>
<td>1.86295</td>
<td>0.328983</td>
<td>1.534</td>
<td>0.33729</td>
<td></td>
</tr>
<tr>
<td>-0.680876</td>
<td>0.372099</td>
<td>0.562604</td>
<td>-1.22691</td>
<td>1.03807</td>
<td>-0.0531184</td>
<td></td>
</tr>
<tr>
<td>-0.680441</td>
<td>-0.59769</td>
<td>0.895902</td>
<td>-2.43549</td>
<td>0.0267913</td>
<td>1.11253</td>
<td></td>
</tr>
<tr>
<td>1.37863</td>
<td>0.798041</td>
<td>2.59144</td>
<td>0.580531</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

and those in Tables 3.6 and 3.7 were performed using SpIus. Therefore, variances from Table 3.5 should be multiplied by a factor

\[
\frac{n}{n - 1}
\]

when being compared with analogous variances associated with Table 3.7.

We discuss the results and conclusions based on Tables 3.5, 3.6 and 3.7 in the next section.

### 3.5 Discussion and Conclusions

A number of points arise from consideration of the results described in the previous section, amongst which we note the following:

1. The symbolic bootstrap methodology developed in this chapter may be applied to a broad class of statistics, namely, to any estimator which may be expressed
Table 3.5: Symbolic bootstrap and classical results for the correlation coefficient using two data sets

<table>
<thead>
<tr>
<th>Analytical Bootstrap</th>
<th>Empirical Distribution $F_n$ and $G_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Law School Data ($n = 15$)</td>
</tr>
<tr>
<td></td>
<td>2nd order</td>
</tr>
<tr>
<td>$E_{w</td>
<td>x,y} (\hat{\rho}^*)$</td>
</tr>
<tr>
<td>$E_{x,y} [E_{w</td>
<td>x,y} (\hat{\rho}^*)]$</td>
</tr>
<tr>
<td>$[\text{Var}<em>{x,y} (E</em>{w</td>
<td>x,y} (\hat{\rho}^*))]^{1/2}$</td>
</tr>
<tr>
<td>$[\text{Var}_{w</td>
<td>x,y} (\hat{\rho}^*)]^{1/2}$</td>
</tr>
<tr>
<td>$\text{Var}_{w</td>
<td>x,y} (\hat{\rho}^*)$</td>
</tr>
<tr>
<td>$E_{x,y} [\text{Var}_{w</td>
<td>x,y} (\hat{\rho}^*)]$</td>
</tr>
<tr>
<td>$[\text{Var}<em>{x,y} (\text{Var}</em>{w</td>
<td>x,y} (\hat{\rho}^*))]^{1/2}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Classical results</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{\rho}$</td>
</tr>
<tr>
<td>$E_{x,y} (\hat{\rho})$</td>
</tr>
<tr>
<td>$[\text{Var}_{x,y} (\hat{\rho})]^{1/2}$</td>
</tr>
</tbody>
</table>

as a smooth (possibly non-linear) function of averages. The sample correlation coefficient is an example of such an estimator.

2. The computational tools presented in this chapter with respect to the correlation coefficient could provide symbolic expansions for any bootstrap cumulant to any desired order as well as properties of these.

3. The numerical evaluation of the 2nd and 4th order analytical expansions for $E_{w|x,y} (\hat{\rho}^*)$ and $[\text{Var}_{w|x,y} (\hat{\rho}^*)]^{1/2}$ given in Table 3.5 for the sample correlation are confirmed by the analogous Monte Carlo calculations in Tables 3.6 and 3.7 for both data sets.

Moreover, the numerical evaluation of the 4th order symbolic expansion for the bootstrap standard error of the correlation $[\text{Var}_{w|x,y} (\hat{\rho}^*)]^{1/2}$ for the law school data as given in Table 3.5 matches the conventional bootstrap calculation given in Efron and Tibshirani (1993, p. 50), i.e. $\text{SE}(\hat{\rho}^*) = 0.132$ for $B = 3200$.

This demonstrates that the symbolic tools developed here for calculating bootstrap quantities comprise a viable alternative methodology to conventional re-
Table 3.6: Monte Carlo Bootstrap Estimates of the Mean of the correlation coefficient for various B

<table>
<thead>
<tr>
<th>B</th>
<th>Monte Carlo Bootstrap Estimates of the Mean $E_{B}(\hat{\rho}^*)$ of $\hat{\rho}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Law School Data $(n = 15)$</td>
</tr>
<tr>
<td>50</td>
<td>0.7446 (0.0175)</td>
</tr>
<tr>
<td>100</td>
<td>0.7893 (0.0119)</td>
</tr>
<tr>
<td>200</td>
<td>0.7732 (0.0088)</td>
</tr>
<tr>
<td>400</td>
<td>0.7762 (0.0065)</td>
</tr>
<tr>
<td>800</td>
<td>0.7729 (0.0046)</td>
</tr>
<tr>
<td>1600</td>
<td>0.7637 (0.0033)</td>
</tr>
<tr>
<td>3200</td>
<td>0.7704 (0.0024)</td>
</tr>
<tr>
<td>6400</td>
<td>0.7715 (0.0017)</td>
</tr>
<tr>
<td>12,800</td>
<td>0.7713 (0.0012)</td>
</tr>
<tr>
<td>125,000</td>
<td>0.7708 (0.0008)</td>
</tr>
</tbody>
</table>

sampling. Any bootstrap moment of a sufficiently smooth estimator, and properties of this, may be derived symbolically and subsequently evaluated numerically to within a specified order of accuracy without recourse to conventional Monte Carlo resampling. This is not true with conventional bootstrap estimates which are unavoidably subject to random variability (note the SE's enclosed in brackets in Tables 3.6 and 3.7).

4. For $n = 100$ the 2nd and 4th order analytical calculations are identical up to at least three decimal places for the six quantities in Table 3.5. For $n = 15$ this is true only up to the first decimal place. We conclude that fewer terms are required in the analytical bootstrap expansions as the size of a data set increases. Indeed, 4th order accuracy was required to achieve identical results to at least two significant figures for bootstrap standard errors of the correlation between the analytical and conventional Monte Carlo methods for $n = 15$. However, only 2nd order accuracy was sufficient to achieve this for the larger data set $n = 100$. Comparatively then, the analytical approach is less computer intensive for larger data sets. The reverse is true for conventional bootstrap calculations.
Table 3.7: Monte Carlo Bootstrap Estimates of Standard Error of the correlation coefficient for various B

<table>
<thead>
<tr>
<th>B</th>
<th>Monte Carlo Bootstrap Estimates of Standard Error ($\hat{se}_B$) for $\hat{\rho}$</th>
<th>Law School Data (n = 15)</th>
<th>Simulated Data (n = 100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.1241 (0.0084)</td>
<td>0.0605 (0.0059)</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0.1194 (0.0084)</td>
<td>0.0463 (0.0028)</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>0.1251 (0.0055)</td>
<td>0.0505 (0.0026)</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>0.1292 (0.0051)</td>
<td>0.0544 (0.0020)</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>0.1291 (0.0044)</td>
<td>0.0527 (0.0013)</td>
<td></td>
</tr>
<tr>
<td>1600</td>
<td>0.1336 (0.0030)</td>
<td>0.0506 (0.0009)</td>
<td></td>
</tr>
<tr>
<td>3200</td>
<td>0.1330 (0.0020)</td>
<td>0.0535 (0.0007)</td>
<td></td>
</tr>
<tr>
<td>6400</td>
<td>0.1326 (0.0015)</td>
<td>0.0507 (0.0005)</td>
<td></td>
</tr>
<tr>
<td>12,800</td>
<td>0.1330 (0.0010)</td>
<td>0.0516 (0.0003)</td>
<td></td>
</tr>
<tr>
<td>125,000</td>
<td>0.1340 (0.0007)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. The closeness between analytical results for $E_{w|x,y}(\hat{\rho})$ and $[Var_{w|x,y}(\hat{\rho})]^{1/2}$ and the corresponding Monte Carlo evaluations increases as B increases for both data sets.

6. The series expansions associated with Table 3.1 for the correlation coefficient constitute symbolic representations, in fact, formulas, which may be applied to any bivariate set of data for which calculations of the correlation, its bootstrap moments, as well as properties of these are desired. An advantage to the analytical approach is that once a particular expansion to a specified order has been calculated the derivational work for the problem at hand has been done and need not be duplicated by other practitioners. The formula can then be numerically evaluated efficiently for any bivariate data set for which estimation of the correlation coefficient is required. This contrasts with the conventional approach which requires that a particular bootstrap moment for the correlation coefficient based on Monte Carlo simulation be generated each time that a different data set is encountered.

7. The symbolic bootstrap method can represent bootstrap moments analytically which is not possible in the context of conventional bootstrap computation.
Moreover, diagnostic properties such as bias and variance of bootstrap moments may be represented analytically. The reason for this is that any bootstrap moment when expressed symbolically represents a random variable for which various moments (i.e. properties) may in turn be considered. This permits the user to study the analytical structure of a bootstrap quantity of interest and identify component statistical quantities such as moments and sums. This can be potentially insightful.

For example, the bias of the bootstrap variance of the correlation is represented by

\[
Bias = E_{x,y}[Var_{w|x,y}(\hat{\rho}^*)] - Var_{x,y}[\hat{\rho}]
\]  

(3.56)

This expression could be evaluated for any distribution, for example, the Gaussian distribution. Using the empirical distribution functions \(F_n\) and \(G_n\) for the law school data we calculate the 4th order bootstrap estimate of bias of the bootstrap variance of the correlation as

\[
\hat{Bias} = \hat{E}_{x,y}[Var_{w|x,y}(\hat{\rho}^*)] - \hat{Var}_{x,y}[\hat{\rho}]
\]

\[
= 0.016806 - 0.017372 \\
= -0.000566
\]  

(3.57)

The corresponding calculation for the simulated data \((n = 100)\) yields \(\hat{Bias} = -0.000047\). Similarly, the 4th order bootstrap estimate of bias of the bootstrap expectation of the correlation for the law school data is

\[
\hat{Bias} = \hat{E}_{x,y}[E_{w|x,y}(\hat{\rho}^*)] - \hat{E}_{x,y}[\hat{\rho}]
\]

\[
= 0.765075 - 0.770879 \\
= -0.005804
\]  

(3.58)

The corresponding calculation for \(n = 100\) is \(\hat{Bias} = -0.000238\).
These results suggest that the bootstrap variance of the correlation and the bootstrap procedure for estimating the mean of the correlation are largely free of bias. In both cases, the negligible bias that exists decreases for increasing sample size.

8. We observe that

\[ E_{w|x,y}(\hat{\rho}^*) = E_{x,y}(\hat{\rho}) \]

and

\[ Var_{w|x,y}(\hat{\rho}^*) = Var_{x,y}(\hat{\rho}) \]

for both the 2nd and 4th order expansions.

9. The methodology discussed in Chapter 2 in which exhaustive enumeration of bootstrap resamples was used as a basis for the exact computation of bootstrap estimates and associated properties could alternatively be used for the case of the correlation coefficient. For example, exact evaluation of the bootstrap estimate of standard error of the correlation for the law school data \( n = 15 \) would involve the enumeration according to (2.21) of

\[ N(15) = \binom{2(15) - 1}{15} \]

(3.59)

or almost 80,000,000 distinct bootstrap resamples. This very large computation was avoided and conventional bootstrap sampling with \( B = 125,000 \) was used as a proxy. The results shown at the bottom of Tables 3.6 and 3.7 for the law school data support corresponding analytically-based calculations in Table 3.5.

10. The issue of error analysis with respect to asymptotic expansions of bootstrap moments and their properties is not investigated quantitatively in this thesis. Considerations such as deciding how many terms to include in a particular
expansion and how to tell how much error has been induced by truncating the series would play a role in such an analysis.

The answer to the first question depends on the sample size and the smoothness of the estimator in question. Higher order terms involve higher order derivatives which can escalate in magnitude. In the case of smooth estimators the magnitude of higher order derivatives is dampened. Although no useful bound is known for these they may be evaluated in any application. Hence, series expansions involving very smooth estimators and large sample sizes would converge more rapidly than those involving less smooth estimators and smaller sample sizes. Consequently, in the former case, the error induced by truncating a series at a given point would be less than the error induced by truncating a series at the same place in the latter case. For example, we observe in Table 3.5 that for \( n = 15 \) the error between the 2nd and 4th order expansions of the bootstrap standard error of the correlation coefficient is \( 0.131804 - 0.124276 = 0.007528 \) and this decreases to \( 0.051680 - 0.051133 = 0.000547 \) for \( n = 100 \). The results in Tables 3.5-3.7 suggest that a 4th order expansion of the bootstrap standard error of the correlation evaluated for \( n = 15 \) data points is necessary to achieve the numerical stability that a corresponding 2nd order expansion similarly enjoys when evaluated for the larger sample size of \( n = 100 \). We would make similar conclusions for estimators differing in smoothness characteristics.

The analytic bootstrap developed in this chapter calculates ideal bootstrap estimates and corresponding properties exactly for the case of linear statistics. Such computations are also exact when performed on truncated expansions of non-linear smooth statistics. For example, consider the truncated series expansion of the correlation coefficient given by \( \text{Cor}[i] \) for some finite \( i \). Then the bootstrap variance of this, given by \( \text{BootCum}[\text{Cor}[i]][2. j] \), will be exact for large enough \( j \). More generally, we may conjecture that if the evaluation of the truncated expansion \( \text{ThetaHat}[i] \) for a given data set and finite \( i \) is very close to \( \hat{\theta} \) then the evaluation of the exact expression \( \text{BootCum}[\text{ThetaHat}[i]][2. j] \)
(for large enough $j$) would similarly be very close to the evaluation of the series $\text{BootCum}[\hat{\Theta}[2, j]$.

Finally, analytical error analysis involving truncated series expansions of bootstrap expressions as referred to above may be contrasted with sampling error induced by the Monte Carlo bootstrap. For example, this could entail taking blocks of 100 bootstrap replicates of the estimator and studying the empirical variation among the block estimates. However, such an analysis is limited because it considers only the central region of the empirical distribution of the bootstrap estimate. In contrast, truncated analytical bootstrap expansions can permit expression of subtler higher moments of the distribution, particularly those providing information about the tails. Moreover, such expansions are true for any distribution and not just the empirical distribution.

We will further illustrate the methodology developed in the present chapter by next considering analytic bootstrap expansions associated with the bootstrap-t interval, $BC_a$ confidence intervals, and the Behrens-Fisher statistic.
Chapter 4

Double-Bootstrapping and the Behrens-Fisher statistic

4.1 Introduction

The methodology developed in the preceding chapter will presently be further illustrated by considering analytic expansions associated with the bootstrap-t interval, BCₐ confidence intervals, and the Behrens-Fisher statistic.

Bootstrap-t intervals are described in Efron (1981) and the bias-corrected, accelerated interval (BCₐ) is suggested in Efron (1987). The two-sample problem with unequal variance has a long history: see, for example, Cox and Hinkley (1974, page 143) and Robinson (1982).

4.2 Symbolic Computation of the Bootstrap-t Confidence Interval

4.2.1 Background

We present the conventional bootstrap-t procedure in this section. Our description follows Efron and Tibshirani (1993). The notation follows that of Section 2.2.
CHAPTER 4. DOUBLE-BOOTSTRAPPING

The standard large-sample central $1 - 2\alpha$ confidence interval

$$
(\hat{\theta} - z^{(1-\alpha)} \cdot \hat{s}e, \hat{\theta} - z^{(\alpha)} \cdot \hat{s}e)
$$

(4.1)

is derived from the assumption that

$$
Z = \frac{\hat{\theta} - \theta}{\hat{s}e} \sim N(0, 1).
$$

(4.2)

where $z^{(\alpha)}$ refers to the $\alpha$th percentile of the standard normal distribution and $\hat{s}e$ is an estimate of the standard deviation of $\hat{\theta}$ . This is valid as $n \to \infty$, but is only an approximation for finite samples. In 1908, for the case $\hat{\theta} = \bar{x}$, Gosset derived the better approximation

$$
t = \frac{\hat{\theta} - \theta}{\hat{s}e} \sim t_{n-1}.
$$

(4.3)

where $t_{n-1}$ represents the Student's $t$ distribution on $n - 1$ degrees of freedom. This approximation yields the interval

$$
(\hat{\theta} - t_{n-1}^{(1-\alpha)} \cdot \hat{s}e, \hat{\theta} - t_{n-1}^{(\alpha)} \cdot \hat{s}e).
$$

(4.4)

with $t_{n-1}^{(\alpha)}$ denoting the $\alpha$th percentile of the $t$ distribution on $n - 1$ degrees of freedom.

The use of the $t$ distribution doesn't adjust the confidence interval to account for skewness in the underlying population or other errors that can result when $\hat{\theta}$ is not the sample mean. The bootstrap-t interval is a procedure which does adjust for these errors. We describe this next.

From an observed random sample $x = (x_1, \ldots, x_n)$ we generate $B$ bootstrap samples

$$
x_1^*, x_2^*, \ldots, x_B^*
$$
and for each we compute the quantity

$$t^*(b) = \frac{\hat{\theta}^*(b) - \hat{\theta}}{\hat{se}^*(b)},$$

where $\hat{\theta}^*(b)$ is the value of $\hat{\theta}$ for the bootstrap sample $x_b^*$ and $\hat{se}^*(b)$ is the estimated standard error of $\hat{\theta}^*$ for the same bootstrap sample $x_b^*$. The $\alpha$th percentile of the $t^*(b)$ is estimated by the value $\bar{t}^{(\alpha)}$ such that

$$\#\{t^*(b) \leq \bar{t}^{(\alpha)}\} / B = \alpha.$$ \hfill (4.6)

The bootstrap-t confidence interval is thus given by

$$\left(\hat{\theta} - \bar{t}^{(1-\alpha)} \cdot \hat{se}, \hat{\theta} - \bar{t}^{(\alpha)} \cdot \hat{se}\right).$$ \hfill (4.7)

The ‘bootstrap-t’ approach thus estimates the distribution of (4.3) directly from the data. A bootstrap table is built by generating $B$ bootstrap samples, and then computing the bootstrap version of (4.3) for each: the bootstrap table consists of the percentiles of these $B$ values. Note that $\hat{\theta}$ and $\hat{se}$ represent estimates using the original sample: they are not bootstrap estimates. However, $\hat{se}$ could be replaced by the bootstrap estimate of standard error of $\hat{\theta}$, i.e. $\hat{se}_B$. We will use this option when we consider the analytical bootstrap-t approach in the next section.

There are a number of advantages that the bootstrap-t procedure enjoys over the standard methods. First, accurate intervals can be obtained without having to make normal theory assumptions. Second, bootstrap-t intervals often enjoy better coverage. Third, these intervals offer second order accuracy versus first order accuracy associated with standard methods. The bootstrap-t approach is particularly applicable to location statistics like the sample mean or a sample percentile. However, at least in its simple form, it cannot be trusted for more general problems, like setting a confidence interval for a correlation coefficient. Moreover, it may perform erratically in small-sample, nonparametric settings. The latter drawback can however be alleviated by the use of appropriate transformations; see Tibshirani (1988) for the
implementation of the variance-stabilized bootstrap-t interval.

Finally, there is a computational problem associated with the bootstrap-t approach which can be circumvented by employing the methodology developed in this thesis. In the denominator of the statistic (4.5) we require $se^\star(b)$, the estimated standard error of $\hat{\theta}$ for the bootstrap sample $x^\star$. The difficulty arises when $\hat{\theta}$ is a complicated statistic, for which there is no simple standard error formula. Thus we would need to compute a bootstrap estimate of standard error for each bootstrap sample. Thus, two (nested) levels of bootstrap sampling are required. The quantity (4.5) now becomes

$$t^\star(b) = \frac{\hat{\theta}^\star(b) - \hat{\theta}}{se^{**}(b)} .$$

where the double star notation in the denominator is used to signify that two levels of bootstrapping are required. It is considered sufficient to use $B = 25$ for the estimation of each $se^{**}(b)$, while $B = 1000$ is needed for the computation of percentiles. Hence, the overall number of bootstrap samples needed is perhaps $25 \cdot 1000 = 25,000$. A formidable number if $\hat{\theta}$ is costly to compute.

### 4.2.2 Symbolic Computation of the Bootstrap-t Interval for the Correlation Coefficient

The analytic bootstrap-t approach constructs confidence intervals in a similar way that standard normal and t tables are used to establish (4.4) from (4.3). There are two key differences. First, the conventional estimate of standard error $se$ used in the denominator of (4.3) is replaced by its bootstrap analogue

$$\sqrt{Var_{w|x}(\hat{\theta}^\star)} .$$

Second, both $\hat{\theta}$ and (4.9) may each be expressed in series format (to some order) as was illustrated in the previous chapter at which point the user may choose to compute these numerically for given data. This gives rise to two advantages not enjoyed by
the conventional bootstrap-t procedure. Firstly, as will be shown below, maintaining the quantities \( \hat{\theta} \) and (4.9) in analytical form allows the subsequent computation of certain bootstrap moments which may be used to further standardize the analytical t-statistic described here. Moreover, properties of these bootstrap moments with respect to the original underlying distribution may also be assessed although we shall not consider these. Secondly, calculation of an analytical bootstrap-t interval precludes the need for Monte Carlo resampling at any level. For example, by circumventing the requirement for building a distribution of \( B \) bootstrap percentiles of (4.3) the analytic bootstrap-t method thus avoids the necessity for undertaking double-bootstrapping in order to calculate quantities such as \( \hat{se}^{**}(b) \). We illustrate the approach for the sample correlation coefficient.

Replacing \( \theta = \rho \) in (4.3) we modify

\[
t = \frac{\hat{\rho} - \rho}{\hat{se}}
\]

(4.10)

by considering the \( t \)-like statistic

\[
t = \frac{\hat{\rho} - \rho}{\sqrt{Var_{w|x}(\hat{\rho}^*)}}
\]

(4.11)

which can be re-written as the difference

\[
t = \hat{\rho} \cdot \left[ Var_{w|x}(\hat{\rho}^*) \right]^{-1/2} - \rho \cdot \left[ Var_{w|x}(\hat{\rho}^*) \right]^{-1/2} .
\]

(4.12)

Observe that (4.12) may be expressed analytically as a series expansion to any order desired because the component factors \( \hat{\rho} \) and \( Var_{w|x}(\hat{\rho}^*) \) may each be similarly expressed (i.e. algebraically) according to Section 3.4. In the second term of (4.12) \( \rho \) is kept constant; when bootstrap moments of \( t \) are computed \( \rho \) is then replaced by \( \hat{\rho} \) at which point it is expressed analytically to some order.

The bootstrap expectation

\[
E_{w|x}(t)
\]

(4.13)
and the bootstrap variance

\begin{equation}
Var_{w|x}(t)
\end{equation}

may each be represented analytically to any desired order. Moreover, properties of \((4.13)\) and \((4.14)\) such as \(E_x[\cdot]\) and \(Var_x[\cdot]\) may also be assessed although for brevity’s sake we shall not consider these here. For example, the first order expansion of the bootstrap mean \((4.13)\) may be derived immediately using the tool \(\text{BootCum}[t][i, j]\) as

\begin{equation}
\text{BootCum}[t][1, 1] = \frac{3 \rho \sqrt{(-1 + \rho^2)^2}}{\sqrt{n} (2 - 2 \rho^2)}
\end{equation}

which reduces to

\begin{equation}
\text{BootCum}[t][1, 1] = \frac{3 \rho}{2\sqrt{n}}.
\end{equation}

For the law school data we observe that \(\text{BootCum}[t][1, 1] = 0.29595\). The Mathematica code underlying the tool \(\text{BootCum}[t][i, j]\) may be found in Appendix A.3 in a file called t.m.

It is well-known that if we construct a confidence interval for Fisher’s variance-stabilizing transformation of the correlation coefficient \(\phi = \tanh^{-1}\rho\) given by

\begin{equation}
\phi = 0.5 \cdot \log \left(\frac{1 + \rho}{1 - \rho}\right)
\end{equation}

and then transform the endpoints back with the inverse transformation

\begin{equation}
\frac{e^{2\phi} - 1}{e^{2\phi} + 1},
\end{equation}

then a more accurate interval is obtained. We will use \(\phi\) in the context of comparing three pairs of confidence intervals below.

First, the standard \(1 - 2\alpha\) t-interval for the correlation coefficient follows \((4.4)\)
and is given by

\[ \hat{\rho} - t^{(1-\alpha)}_{n-1} \cdot \hat{\sigma} < \rho < \hat{\rho} + t^{(1-\alpha)}_{n-1} \cdot \hat{\sigma}. \]  

(4.17)

This can be improved by using \( \phi \) as in

\[ \phi(\hat{\rho}) - t^{(1-\alpha)}_{n-1} \cdot \hat{\sigma} \phi(\hat{\rho}) < \phi(\rho) < \phi(\hat{\rho}) + t^{(1-\alpha)}_{n-1} \cdot \hat{\sigma} \phi(\hat{\rho}). \]

(4.18)

and transforming back to an interval for \( \rho \) using (4.16). We assume that the corresponding percentiles \( t^{(1-\alpha)}_{n-1} \) are different in (4.17) and (4.18). Second, the bootstrap-t interval for \( \rho \) given by (4.17) as

\[ \hat{\rho} - \hat{t}^{(1-\alpha)} \cdot \hat{\sigma} < \rho < \hat{\rho} + \hat{t}^{(1-\alpha)} \cdot \hat{\sigma}. \]

(4.19)

may also be improved by considering

\[ \phi(\hat{\rho}) - \hat{t}^{(1-\alpha)} \cdot \hat{\sigma} \phi(\hat{\rho}) < \phi(\rho) < \phi(\hat{\rho}) + \hat{t}^{(1-\alpha)} \cdot \hat{\sigma} \phi(\hat{\rho}). \]

(4.20)

Again, we assume that the corresponding values of \( \hat{t}^{(1-\alpha)} \) are different in each case. Third, the analytic bootstrap-t interval is based on (4.11) and is given by

\[ \hat{\rho} - t^{(1-\alpha)}_{n-1} \cdot \sqrt{\text{Var}_{w|x}(\hat{\rho})} < \rho < \hat{\rho} + t^{(1-\alpha)}_{n-1} \cdot \sqrt{\text{Var}_{w|x}(\hat{\rho})}. \]

(4.21)

Finally, a standardized analytical bootstrap-t interval may be formulated by standardizing (4.11) using (4.13) and (4.14). Thus upon rearranging

\[ -t^{(1-\alpha)}_{n-1} \frac{t - E_{w|x}(t)}{\sqrt{\text{Var}_{w|x}(t)}} < t^{(1-\alpha)}_{n-1}. \]

where \( t \) is given by (4.11) we have

\[ \hat{\rho} - \sqrt{\text{Var}_{w|x}(\hat{\rho})} \cdot \left[ E_{w|x}(t) + t^{(1-\alpha)}_{n-1} \cdot \sqrt{\text{Var}_{w|x}(t)} \right] < \rho < \]
For example, to calculate a 90% confidence interval for $\rho$ for the law school data using (4.22) we set

1. $\hat{\rho} = 0.776374$

2. $\sqrt{Var_{w|x}(\hat{\rho})} = 0.132$ from Chapter 3

3. $\sqrt{Var_{w|x}(t)} \to 1$

4. $E_{w|x}(t) = 0.29595$ (first order result seen above).

Table 4.1 compares the six different 90% confidence intervals (4.17) to (4.22) for the correlation coefficient for the $n = 15$ law school data.

Table 4.1: Comparison of 90% confidence intervals for $\rho$ using analytic bootstrap-t, conventional bootstrap-t, and standard methods for the law school data ($n = 15$)

<table>
<thead>
<tr>
<th></th>
<th>standard $t$</th>
<th>(.57, .98)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>standard $t_0$</td>
<td>(.51, .91)</td>
</tr>
<tr>
<td>3</td>
<td>boot-$t$</td>
<td>(.03, .90)</td>
</tr>
<tr>
<td>4</td>
<td>boot-$t_0$</td>
<td>(.45, .93)</td>
</tr>
<tr>
<td>5</td>
<td>analytical$^1$ boot-$t$</td>
<td>(.54, 1.01)</td>
</tr>
<tr>
<td>6</td>
<td>analytical$^1$ standard boot-$t$</td>
<td>(.58, .98)</td>
</tr>
</tbody>
</table>

Intervals #3 and #4 were obtained by Efron and Tibshirani (1993, p. 162-3) using conventional double-bootstrapping. They used $B = 25$ bootstrap samples to calculate the bootstrap estimate of standard error $se^{**}(b)$ for each of 1000 values of $\hat{\rho}$ that were generated so that a total of 25,000 bootstrap samples were used. The superscript “1” in intervals #5 - #6 signifies that all analytic component terms were computed to first order except for the bootstrap variance of the correlation which was calculated to fourth order as in Chapter 3.
4.2.3 Discussion

We have shown that the two nested levels of bootstrap sampling associated with the conventional bootstrap-t interval can be avoided by representing the bootstrap-t statistic in terms of a symbolic expansion. We compared confidence intervals for the correlation using standard and conventional bootstrap-t methods with the same intervals evaluated analytically.

It is observed that the conventional bootstrap-t interval #4 for the correlation based on Fisher's $z$ transformation has the same length as the analytical bootstrap-t interval #5 computed without use of Fisher's transformation. However the latter interval has partially strayed outside the allowable range for the correlation. Moreover, the analytical standardized bootstrap-t interval #6 is shorter than the conventional interval #4.

This suggests that the analytical bootstrap-t approach provides intervals which not only circumvent computer-intensive double-bootstrapping calculations but may also be more accurate.

4.3 Symbolic Computation of $BC_a$ Confidence Intervals

4.3.1 Introduction

In this section we explore the development of computer algebraic tools which permit the analytic expression of an integral component associated with $BC_a$ confidence intervals. We first provide a general overview of the $BC_a$ confidence interval and its precursor, the percentile method. We follow Efron and Tibshirani (1993) and Efron (1987).
4.3.2 Background

A number of problems connected with conventional bootstrap-t confidence intervals were outlined in Section 4.2.1. Instead of focusing on a statistic of the form \( t = (\hat{\theta} - \theta)/s.e \) a different approach for remedying such problems works directly with the bootstrap distribution of \( \theta \). We describe first the percentile method followed by its improved version, the bias-corrected and accelerated (BCa) interval.

Suppose the empirical distribution \( \hat{F} \) generates the bootstrap data set \( x^* \) by random sampling, giving the bootstrap replication \( \hat{\theta}^* = s(x^*) \). This is repeated an infinite number of times. Let \( \hat{G} \) be the cumulative bootstrap distribution function of \( \hat{\theta}^* \), that is,

\[
\hat{G}(s) = \#\{\hat{\theta}^*(b) < s\}/B.
\]

The \( 1 - 2\alpha \) percentile interval is defined by the \( \alpha \) and \( 1 - \alpha \) percentiles of \( \hat{G} \) as

\[
\left( \hat{\theta}_{lo}, \hat{\theta}_{up} \right) = \left( \hat{G}^{-1}(\alpha), \hat{G}^{-1}(1 - \alpha) \right) = \left( \hat{\theta}^{*(0)}, \hat{\theta}^{*(1-\alpha)} \right)
\]

since by definition \( \hat{G}^{-1}(\alpha) = \hat{\theta}^{*(\alpha)} \), the 100 \cdot \alpha th percentile of the bootstrap distribution. Expression (4.23) refers to the ideal bootstrap situation in which the number of bootstrap replications is infinite. In practice we use some finite number \( B \) of replications in which case the approximate \( 1 - 2\alpha \) percentile interval is

\[
\left( \hat{\theta}_{lo}, \hat{\theta}_{up} \right) \approx \left( \hat{\theta}^{*(0)}_B, \hat{\theta}^{*(1-\alpha)}_B \right).
\]

The lower confidence limit \( \hat{\theta}^{*(\alpha)}_B \) is the 100 \cdot \alpha th empirical percentile of the \( \hat{\theta}^*(b) \) values, that is, the \( B \cdot \alpha \) th value in the ordered list of the \( B \) replications of \( \hat{\theta}^* \).

The percentile interval for a parameter \( \theta \) is more accurate than a standard normal interval when the bootstrap distribution of \( \hat{\theta}^* \) is non-normal. Both methods agree well when the standard interval is constructed on an appropriate transformation \( \hat{\theta} \)
of \( \theta \) and then mapped back to the \( \theta \) scale. The advantage of the percentile method is that it automatically makes this transformation; unlike the standard interval, we don't need to know the correct transformation. All we need assume is that such a transformation exists.

The percentile method is not the last word in bootstrap confidence intervals. There are other ways the standard intervals can fail, besides non-normality. For example, \( \hat{\theta} \) might be a biased normal estimate.

\[
\hat{\theta} \sim N(\theta + \text{bias}, se^2)
\]  

(4.25)

in which case no transformation can fix things up. The \( BC_\alpha \) confidence interval represents an extension of the percentile method that automatically handles both bias and transformations. Moreover, it allows the standard error in (4.25) to vary with \( \theta \), rather than being forced to stay constant. We discuss the \( BC_\alpha \) confidence procedure next.

The \( BC_\alpha \) interval endpoints are given by percentiles of the bootstrap distribution, but not necessarily the same ones as in (4.23). The percentiles used depend on two numbers \( \hat{\alpha} \) and \( \hat{\alpha}_0 \), called the acceleration and bias-correction, respectively. The \( 1 - 2\alpha \) \( BC_\alpha \) interval is given by

\[
\left( \hat{\theta}_{lo}, \hat{\theta}_{up} \right) = \left( \hat{\theta}^{*(\alpha_1)}, \hat{\theta}^{*(\alpha_2)} \right).
\]  

(4.26)

where

\[
\alpha_1 = \Phi \left( \hat{z}_0 + \frac{\hat{z}_0 + z^{(\alpha)}}{1 - \hat{\alpha}(\hat{z}_0 + z^{(\alpha)})} \right)
\]

and

\[
\alpha_2 = \Phi \left( \hat{z}_0 + \frac{\hat{z}_0 + z^{(1-\alpha)}}{1 - \hat{\alpha}(\hat{z}_0 + z^{(1-\alpha)})} \right).
\]  

(4.27)

Here \( \Phi(\cdot) \) is the standard normal cumulative distribution function and \( z^{(\alpha)} \) is the
100 \cdot \alpha \text{th percentile point of a standard normal distribution. Notice that if } \hat{a} \text{ and } \hat{z}_0 \text{ equal zero, then } \alpha_1 = \alpha \text{ and } \alpha_2 = 1 - \alpha \text{ so that the } BC_\alpha \text{ and percentile intervals coincide.}

The } BC_\alpha \text{ interval given above is based on the following model. Recall that standard confidence intervals are based on the assumption}

$$\frac{\hat{\theta} - \theta}{se} \sim N(0, 1). \quad (4.28)$$

The } BC_\alpha \text{ interval assumes, more generally, that there exists a monotone transformation } g \text{ and constants } z_0 \text{ and } a \text{ such that}

$$\hat{\phi} = g(\hat{\theta}), \quad \phi = g(\theta) \quad (4.29)$$

satisfy

$$\frac{\hat{\phi} - \phi}{se_\phi} \sim N(-z_0, 1) \quad (4.30)$$

with

$$se_\phi = se_{\phi_0}[1 + a(\phi - \phi_0)/se_{\phi_0}] \quad (4.31)$$

for any fixed value of } \phi_0. The } BC_\alpha \text{ interval is attractive because one does not need to know the transformation } g(\cdot); \text{ the only three quantities needed to construct the interval are the bootstrap distribution } \hat{G} \text{ and the constants } z_0 \text{ and } a.

The estimated value of the bias-correction } z_0 \text{ is obtained directly from the proportion of bootstrap replications less than the original estimate } \hat{\theta},

$$\hat{z}_0 = \Phi^{-1}\left(\frac{\#\{\hat{\theta}^*(b) < \hat{\theta}\}}{B}\right). \quad (4.32)$$

Roughly speaking, the quantity } \hat{z}_0 \text{ measures the median bias of } \hat{\theta}^*. \text{ that is, the discrepancy between the median of } \hat{\theta}^* \text{ and } \hat{\theta}, \text{ in normal units.
The quantity \( a \) is called the acceleration because it measures the rate of change of the standard error of \( \hat{\theta} \) with respect to the true parameter value \( \theta \), measured on a normalized scale. The normal approximation \( \hat{\theta} \sim N(\theta, se^2) \) assumes that the standard error of \( \hat{\theta} \) is the same for all \( \theta \). This is often unrealistic and the acceleration adjustment \( a \) corrects for this. For the multinomial distribution, which corresponds to the nonparametric bootstrap, \( \hat{\theta} \) may be computed in terms of the jackknife values of a statistic \( \hat{\theta} = s(x) \). Let \( x(i) \) represent the original sample with the \( i \)th point \( x_i \) deleted, let \( \hat{\theta}(i) = s(x(i)) \), and define

\[
\hat{\theta}(i) = \sum_{i=1}^{n} \frac{\hat{\theta}(i)}{n}.
\] (4.33)

A simple expression for the acceleration is given by

\[
\hat{a} = \frac{\sum_{i=1}^{n} (\hat{\theta}(i) - \hat{\theta}(i))^3}{6\{\sum_{i=1}^{n} (\hat{\theta}(i) - \hat{\theta}(i))^2\}^{3/2}}.
\] (4.34)

In summary, the percentile method generalizes the standard normal approximation (4.28) by allowing a transformation \( g(\cdot) \) of \( \theta \) and \( \hat{\theta} \). The \( BC_a \) method adds the further adjustments \( z_0 \) and \( a \). The bootstrap-\( t \) approach is second-order accurate, but not transformation-respecting. The percentile method is transformation-respecting but not second-order accurate. The standard method is neither, while the \( BC_a \) method is both.

In the next section we show how recourse to resampling methods such as the jackknife for estimating the acceleration constant \( a \) can be avoided by using the symbolic bootstrap methodology developed in the previous chapter.

### 4.3.3 Symbolic Computation of the Acceleration Constant \( \hat{a} \)

We show how the computer algebraic tools developed in Chapter 3 may be used for the symbolic representation of the acceleration adjustment used in \( BC_a \) confidence intervals. As will be shown below, the key reason that the acceleration constant is conducive to symbolic representation (whereas the bias-correction \( z_0 \) is not) is that
it can be represented as a smooth function of averages. We provide a setting for this representation by justifying the expression for $se_\phi$, as given by (4.31), as follows.

Indeed, the first-order Taylor series expansion of

$$ (se_\phi)^2 = \frac{\sigma^2_\phi}{n} $$  \hspace{1cm} (4.35)

about $\phi = \phi_0$ is given by

$$ \frac{\sigma^2_\phi}{n} = \frac{\sigma^2_{\phi_0}}{n} + \left( \frac{\sigma^2_{\phi_0}}{n} \right) \cdot (\phi - \phi_0) $$  \hspace{1cm} (4.36)

which may be re-expressed as

$$ \frac{\sigma^2_\phi}{n} = \frac{\sigma^2_{\phi_0}}{n} \left[ 1 + \frac{(\sigma^2_{\phi_0})' \cdot (\phi - \phi_0)}{n(\sigma^2_{\phi_0}/n)} \right] $$  \hspace{1cm} (4.37)

Taking the square root of each side of (4.37) and using the fact that

$$ (1 + e)^{\frac{1}{2}} \approx 1 + \frac{1}{2}e $$

we have

$$ \frac{\sigma_\phi}{\sqrt{n}} \approx \frac{\sigma_{\phi_0}}{\sqrt{n}} \left[ 1 + \frac{(\sigma^2_{\phi_0})' \cdot (\phi - \phi_0)}{2n\sigma_{\phi_0}/\sqrt{n}} \left( \frac{\sigma_{\phi_0}/\sqrt{n}}{\sigma_{\phi_0}/\sqrt{n}} \right) \right] $$  \hspace{1cm} (4.38)

Re-introducing the notation $se_\phi = \sigma_\phi/\sqrt{n}$, we observe that (4.38) and (4.31) are the same, as desired, with

$$ a = \frac{(\sigma^2_{\phi_0})'}{2nse_{\phi_0}} $$  \hspace{1cm} (4.39)

By interpreting (4.36) as a simple linear regression model

$$ E(y) = \beta_0 + \beta_1 \cdot x $$  \hspace{1cm} (4.40)
with \( x = (\phi - \phi_0) \) as the independent variable and the slope \( \beta_1 \) represented by

\[
\beta_1 = \left( \frac{\sigma^2_{\phi_0}}{n} \right)' \tag{4.41}
\]

then (4.39) becomes

\[
a = \frac{\beta_1}{2se_{\phi_0}}. \tag{4.42}
\]

Disregarding the divisor \( n \) in (4.41) for the moment it is well-known that the slope \( \beta_1 \) of a generic regression model (4.40) can always be expressed as

\[
\beta_1 = \frac{\text{cov}(x, y)}{\text{var}(x)} \tag{4.43}
\]

which we re-formulate as

\[
\beta_1 = \frac{E[\{(x - E(x))\{y - E(y)\}]}{\text{var}(x)} = \frac{E[y \cdot x]}{\text{var}(x)} - \frac{E(y) \cdot E(x)}{\text{var}(x)} \tag{4.44}
\]

where the expectation \( E \) and the variance "\( \text{var} \)" are taken with respect to any distribution \( F \).

The expression (4.44) now becomes the basis for the symbolic computation of (4.42). We show this for the case of the correlation coefficient, and then using the law school data we compare our numerical evaluation of (4.42) with that of Efron (1987). Indeed, setting

\[
F = F_n, \quad x = \phi = \rho
\]

we observe that

\[
y = \frac{\sigma^2_{\rho}}{n}
\]
becomes

$$y = \text{Var}_{w|x}(\hat{\rho}^*)$$

and (4.44) becomes

$$\beta_1 = \frac{E_{w|x}[\text{Var}_{w|x}(\hat{\rho}^*) \cdot \hat{\rho}]}{\text{Var}_{w|x}(\hat{\rho}^*)} - \frac{E_{w|x}[\text{Var}_{w|x}(\hat{\rho}^*) \cdot E_{w|x}(\hat{\rho}^*)]}{\text{Var}_{w|x}(\hat{\rho}^*)}. \quad (4.45)$$

Note that the subscript $x$ in (4.45) refers to a bivariate sample of size $n$ whereas the notation $x$ as in (4.40) denotes a single independent random variable.

We now use the same tools developed in Chapter 3 to formulate a new tool called Boot[Acc][i] for expressing (4.45) to any desired order. The Mathematica code underlying the tool Boot[Acc][i] may be found in Appendix A.3 in a file called bca.m. For example, the first-order symbolic bootstrap expansion of (4.45) is given by

$$\text{Boot[Acc][1]} = \left[ 8 \rho - 8 \sqrt{n} \rho - 16 \rho^3 + 16 \sqrt{n} \rho^3 + 8 \rho^5 - 8 \sqrt{n} \rho^5 + 2 \rho^3 \text{Cum}(X, X, X, X) - 2 \sqrt{n} \rho^3 \text{Cum}(X, X, X, X) - 8 \rho^2 \text{Cum}(X, X, X, Y) + 8 \sqrt{n} \rho^2 \text{Cum}(X, X, X, Y) + 8 \rho \text{Cum}(X, X, Y, Y) - 8 \sqrt{n} \rho \text{Cum}(X, X, Y, Y) + 4 \rho^3 \text{Cum}(X, X, Y, Y) - 4 \sqrt{n} \rho^3 \text{Cum}(X, X, Y, Y) - 8 \rho^2 \text{Cum}(X, Y, Y, Y) + 8 \sqrt{n} \rho^2 \text{Cum}(X, Y, Y, Y) + 2 \rho^3 \text{Cum}(Y, Y, Y, Y) - 2 \sqrt{n} \rho^3 \text{Cum}(Y, Y, Y, Y) + 4 \rho^3 \text{Cum}(X, X, X) \text{Z}(n)(X) - 12 \rho^2 \text{Cum}(X, X, Y) \text{Z}(n)(X) + 8 \rho \text{Cum}(X, Y, Y) \text{Z}(n)(X) + 4 \rho^3 \text{Cum}(X, Y, Y) \text{Z}(n)(X) - 4 \rho^2 \text{Cum}(Y, Y, Y) \text{Z}(n)(X) + 8 \rho \text{Z}(n)(X^2) - 20 \rho^3 \text{Z}(n)(X^2) + 12 \rho^5 \text{Z}(n)(X^2) + 4 \rho^3 \text{Cum}(X, X, X, X) \text{Z}(n)(X^2) - 12 \rho^2 \text{Cum}(X, X, X, Y) \text{Z}(n)(X^2) + 8 \rho \text{Cum}(X, X, Y, Y) \text{Z}(n)(X^2) + 6 \rho^3 \text{Cum}(X, X, Y, Y) \text{Z}(n)(X^2) - 8 \rho^2 \text{Cum}(X, Y, Y, Y) \text{Z}(n)(X^2) + 2 \rho^3 \text{Cum}(Y, Y, Y, Y) \text{Z}(n)(X^2) - \rho^3 \text{Z}(n)(X^4) - \right]
4 \rho^2 \text{Cum}(X, X, X) Z(n)(Y) +
8 \rho \text{Cum}(X, X, Y) Z(n)(Y) + 4 \rho^3 \text{Cum}(X, X, Y) Z(n)(Y) -
12 \rho^2 \text{Cum}(X, Y, Y) Z(n)(Y) + 4 \rho^3 \text{Cum}(Y, Y, Y) Z(n)(Y) - 8 Z(n)(XY) +
24 \rho^2 Z(n)(XY) - 16 \rho^4 Z(n)(XY) - 4 \rho^2 \text{Cum}(X, X, X) Z(n)(XY) +
12 \rho \text{Cum}(X, X, X, Y) Z(n)(XY) - 8 \text{Cum}(X, Y, Y, Y) Z(n)(XY) -
8 \rho^2 \text{Cum}(X, X, Y, Y) Z(n)(XY) + 12 \rho \text{Cum}(X, Y, Y, Y) Z(n)(XY) -
4 \rho^2 \text{Cum}(Y, Y, Y, Y) Z(n)(XY) + 4 \rho^2 Z(n)(XY^2) + 8 \rho Z(n)(Y^2) -
20 \rho^3 Z(n)(Y^2) + 12 \rho^5 Z(n)(Y^2) + 2 \rho^3 \text{Cum}(X, X, X, X) Z(n)(Y^2) -
8 \rho^2 \text{Cum}(X, X, X, Y) Z(n)(Y^2) + 8 \rho \text{Cum}(X, X, Y, Y) Z(n)(Y^2) +
6 \rho^3 \text{Cum}(X, X, Y, Y) Z(n)(Y^2) - 12 \rho^2 \text{Cum}(X, Y, Y, Y) Z(n)(Y^2) +
4 \rho^3 \text{Cum}(Y, Y, Y, Y) Z(n)(Y^2) - 4 \rho Z(n)(XY^2) - 2 \rho^3 Z(n)(XY^2) +
4 \rho^2 Z(n)(XY^3) - \rho^3 Z(n)(Y^4)]
/ [4\sqrt{n}(1 - 8 \rho^2 + 4 \rho^4) +
\rho^2 \text{Cum}(X, X, X, X) - 4 \rho \text{Cum}(X, X, X, Y) +
4 \text{Cum}(X, X, Y, Y) + 2 \rho^2 \text{Cum}(X, X, Y, Y) -
4 \rho \text{Cum}(X, Y, Y, Y) + \rho^2 \text{Cum}(Y, Y, Y, Y)]^{\frac{1}{2}} \tag{4.46}

The analytic bootstrap expression (4.46) may be evaluated numerically for any pair of
distributions \( F \) and \( G \) associated with the dependent variables \( x \) and \( y \) for which the
cumulants are known. In particular, we will use the empirical distribution functions
\( F_n \) and \( G_n \) as follows.

Using the law school data with \( \hat{\rho} = 0.776 \) then (4.46) becomes

\[
\text{Boot[Acc][1]} = -0.2772
\]

and upon factoring in the divisor \( n = 15 \) we find

\[
\hat{\beta}_1 = -0.01848
\]
CHAPTER 4. DOUBLE-BOOTSTRAPPING

Using the fourth-order result from Chapter 3

\[ s\varepsilon_{\phi_0} = \sqrt{Var_{lx}(\hat{\rho})} = 0.132 \]

substitution in (4.42) yields

\[ \hat{a} = \frac{-\hat{J}_1}{2s\varepsilon_{\phi_0}} = -0.0700. \]

Using the jackknife approximation (4.34) Efron (1987, p. 178) found

\[ a \doteq -0.0817. \]

Finally, we can perform an independent check on the results obtained above by expressing \( Var(\hat{\rho}) \) as a first-order Taylor expansion about \( \rho = \rho_0 \). Proceeding we get

\[ (1 - \rho^2)^2 \doteq (1 - \rho_0^2)^2 + \left[ (1 - \rho_0^2)^2 \right]' (\rho - \rho_0) \]

with

\[ \hat{J}_1 = \left[ (1 - \rho_0^2)^2 \right]' \]
\[ = 2 \cdot (1 - \rho_0^2) \cdot (-2\rho_0) \]
\[ = -1.23485 \]

where \( \hat{\rho} = \rho_0 = 0.776 \). Dividing by \( n = 15 \) we get \(-0.082323\) which is close to our value of \( \hat{a} = -0.0700 \).

4.3.4 Discussion

We have shown that the acceleration adjustment associated with \( BC_a \) confidence intervals may be represented exactly as a symbolic bootstrap expansion to any order desired. For illustration this was applied to the case of the correlation coefficient. The resulting analytical expression is a formula which may repeatedly be used in any
context for which a $BC_{a}$ confidence interval for the correlation is desired. Numerical evaluation of a first-order analytical bootstrap expansion of the acceleration yielded a result consistent with that produced by jackknife resampling. Higher-order evaluation would render more accurate results. Properties of the symbolic bootstrap representation of the acceleration could be further assessed although for brevity's sake these are not considered here.

4.4 Analytic Bootstrap Computation for the Behrens-Fisher statistic

4.4.1 Introduction

We develop here the analytical representation of bootstrap moments related to the Behrens-Fisher statistic and their properties. Numerical evaluation of the symbolic expressions is illustrated using data from a case study in medicine. We compare some of these results with bootstrap estimates computed using Monte Carlo resampling.

4.4.2 Background

Suppose that $(x_1, \ldots, x_{n_1})$ and $(y_1, \ldots, y_{n_2})$ comprise independent samples from Gaussian distributions $N(\mu_1, \sigma_1^2)$ and $N(\mu_2, \sigma_2^2)$, respectively. Since $\sigma_1^2$ and $\sigma_2^2$ are typically unknown we would consider estimating them by $s_1^2 = \sum_{i=1}^{n_1} (x_i - \bar{x})^2 / (n_1 - 1)$ and $s_2^2 = \sum_{i=1}^{n_2} (y_i - \bar{y})^2 / (n_2 - 1)$. The hypothesis

$$H_0 : \mu_1 = \mu_2$$

may be assessed using the statistic

$$t = \frac{\bar{x} - \bar{y}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \quad (4.47)$$
where \( \bar{x} = \frac{\sum_{i=1}^{n_1} x_i}{n_1} \) and \( \bar{y} = \frac{\sum_{i=1}^{n_2} y_i}{n_2} \). (The test statistic (4.47) would have a Student’s t distribution under the null hypothesis if the two Gaussian populations had the same variance \( \sigma_1^2 = \sigma_2^2 = \sigma^2 \) and a pooled estimate of \( \sigma^2 \) was used.) A number of approximate solutions have been proposed in the literature. This is known as the Behrens-Fisher problem. For illustrative purposes it suffices to consider the balanced case only, namely, when \( n_1 = n_2 = n \), and so the test statistic becomes

\[
\sqrt{n} \left( \frac{\bar{x} - \bar{y}}{\sqrt{s_1^2 + s_2^2}} \right) = bf. \tag{4.48}
\]
say.

In the next section we shall calculate bootstrap estimates of the mean and variance of the Behrens-Fisher statistic (4.48) and their properties by using the analytical bootstrap. The symbolic calculations are facilitated by considering the modified Behrens-Fisher statistic

\[
bf' = \frac{bf}{\sqrt{n}} = \frac{\bar{x} - \bar{y}}{\sqrt{s_1^2 + s_2^2}}. \tag{4.49}
\]

### 4.4.3 Analytic Bootstrap Computation

As was done for the case of the correlation coefficient Table 4.2 below lists the six bootstrap quantities for which symbolic expressions associated with the modified Behrens-Fisher statistic (4.49) are presented in this section. These represent the bootstrap mean and bootstrap variance and the mean and variance - over \( x \) and \( y \) - for each of these. Moreover, if required, the computational tools developed here can provide symbolic expansions for any bootstrap cumulant of the Behrens-Fisher statistic to any desired order as well as properties of these. The tools presented below were coded using Mathematica and are based on the methodology developed in Chapter 3.

The symbolic function \( BF[i] \) expands the modified Behrens-Fisher statistic (4.49)
Table 4.2: Bootstrap estimators and related properties for the Behrens-Fisher statistic for which analytic expansions are derived

<table>
<thead>
<tr>
<th>Properties</th>
<th>Bootstrap Mean (bf)</th>
<th>Bootstrap Variance (bf)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>( E_{w</td>
<td>x,y}(\text{bf}^*) )</td>
</tr>
<tr>
<td>Variance</td>
<td>( Var_{z,y}(E_{w</td>
<td>x,y}(\text{bf}^*)) )</td>
</tr>
</tbody>
</table>

using a Taylor series in residual sums correct to \( i \)-th order. We assume \( H_0 : \mu_1 = \mu_2 = 0 \) and that the variables \( x \) and \( y \) are independent. For example, the 2nd order expansion of (4.49) is given by

\[
BF[2] = \left[ \frac{Z(n)(X)}{\sqrt{Cum(X,X) + Cum(Y,Y)}} - \frac{Z(n)(Y)}{\sqrt{Cum(X,X) + Cum(Y,Y)}} \right] / \sqrt{n} + \\
\left[ \frac{-(Z(n)(X) Z(n)(X^2))}{2(Cum(X,X) + Cum(Y,Y))^\frac{3}{2}} + \frac{Z(n)(X^2) Z(n)(Y)}{2(Cum(X,X) + Cum(Y,Y))^\frac{3}{2}} - \right. \\
\frac{Z(n)(X) Z(n)(Y^2)}{2(Cum(X,X) + Cum(Y,Y))^\frac{3}{2}} + \left. \frac{Z(n)(Y) Z(n)(Y^2)}{2(Cum(X,X) + Cum(Y,Y))^\frac{3}{2}} \right] / n .
\]

(4.50)

The tool \( \text{Cum}[\text{BF}][i, j] \) returns the \( i \)-th cumulant - over \( x \) and \( y \) - of the modified Behrens-Fisher statistic expressed as an expansion correct to order \( n^{-1/2} \). For example, the mean \( E_{x,y}(bf) \) and the variance \( Var_{x,y}(bf) \) expressed as expansions to order \( n^{-1} \) are given, respectively, by

\[
\text{Cum}[\text{BF}][1,2] = \frac{-Cum(X,X,X) + Cum(Y,Y,Y)}{2 n (Cum(X,X) + Cum(Y,Y))^\frac{3}{2}}
\]

(4.51)

and

\[
\text{Cum}[\text{BF}][2,2] = \frac{1}{n} .
\]

(4.52)
CHAPTER 4. DOUBLE-BOOTSTRAPPING

The function \texttt{BootCum[BF][i, j]} returns the \(i\)-th bootstrap cumulant of the modified Behrens-Fisher statistic expressed as an expansion correct to order \(n^{-j/2}\).

The \texttt{Mathematica} code underlying this tool and \texttt{BF[i]} may be found in Appendix A.3. For example, the 4th order analytic expressions for the bootstrap expectation

\begin{equation}
E_{w|x,y}(bf^*)
\end{equation}

and the bootstrap variance

\begin{equation}
Var_{w|x,y}(bf^*) = E_{w|x,y}[bf^{*2}] - [E_{w|x,y}(bf^*)]^2
\end{equation}

of the modified Behrens-Fisher statistic are given, respectively, by

\begin{align*}
\text{BootCum[BF][1, 4]} & = \left[ \text{Cum}(X, X, X) - \text{Cum}(Y, Y, Y) - 2\sqrt{n} Z(n)(X) + \\
& 3\sqrt{n} \text{Cum}(X, X) Z(n)(X) + \sqrt{n} \text{Cum}(Y, Y) Z(n)(X) + \\
& 2n^{3/2} (\text{Cum}(X, X) + \text{Cum}(Y, Y)) \\
& Z(n)(X) + \sqrt{n} Z(n)(X)^3 + 3 Z(n)(X) Z(n)(X^2) + \\
& 2\sqrt{n} Z(n)(Y) - \sqrt{n} \text{Cum}(X, X) Z(n)(Y) - \\
& 3\sqrt{n} \text{Cum}(Y, Y) Z(n)(Y) - 2n^{3/2} (\text{Cum}(X, X) + \text{Cum}(Y, Y)) \\
& Z(n)(Y) - \sqrt{n} Z(n)(X)^2 Z(n)(Y) - Z(n)(X^2) Z(n)(Y) + \\
& \sqrt{n} Z(n)(X) Z(n)(Y)^2 - \sqrt{n} Z(n)(Y)^3 + \\
& Z(n)(X) Z(n)(Y^2) + 3 Z(n)(Y) Z(n)(Y^2) \right] \\
& / \left[ 2n^2 (\text{Cum}(X, X) + \text{Cum}(Y, Y))^{3/2} \right]
\end{align*}

\begin{align*}
\text{BootCum[BF][2, 4]} & = \left[ -2 \text{Cum}(X, X) + 3 \text{Cum}(X, X)^2 + n \text{Cum}(X, X)^2 - \\
& 2 \text{Cum}(Y, Y) + 2 \text{Cum}(X, X) \text{Cum}(Y, Y) + \\
& 2n \text{Cum}(X, X) \text{Cum}(Y, Y) + 3 \text{Cum}(Y, Y)^2 + n \text{Cum}(Y, Y)^2 + \\
& \right]
\end{align*}
CHAPTER 4. DOUBLE-BOOTSTRAPPING

\[ 2 Cum(X, X) Z(n)(X)^2 + \sqrt{n} Cum(X, X) Z(n)(X^2) + \]
\[ \sqrt{n} Cum(Y, Y) Z(n)(X^2) - 2 Cum(X, X) Z(n)(X) Z(n)(Y) - \]
\[ 2 Cum(Y, Y) Z(n)(X) Z(n)(Y) + 2 Cum(Y, Y) Z(n)(Y)^2 + \]
\[ \sqrt{n} Cum(X, X) Z(n)(Y^2) + \sqrt{n} Cum(Y, Y) Z(n)(Y^2) \]
\[ / [n^2 (Cum(X, X) + Cum(Y, Y))^2] . \]  

(4.56)

We now consider the procedure that evaluates properties, indeed, various cumulants - over \( x \) and \( y \) - of the bootstrap mean (4.53) and the bootstrap variance (4.54). The function \( Cum[BootCum[BF][i][j, k]] \) returns a series expansion correct to order \( n^{-k/2} \) of the \( j \)-th cumulant - over \( x \) and \( y \) - of the \( i \)-th bootstrap cumulant of the modified Behrens-Fisher statistic. In particular,

\[ E_{x,y} \left[ E_{w|x,y}(bf^*) \right] = Cum[BootCum[BF][1][1,4] + O_p(n^{-2}) \]  

(4.57)

\[ Var_{x,y} \left[ E_{w|x,y}(bf^*) \right] = Cum[BootCum[BF][1][2,4] + O_p(n^{-2}) \]  

(4.58)

\[ E_{x,y} \left[ Var_{w|x,y}(bf^*) \right] = Cum[BootCum[BF][2][1,4] + O_p(n^{-2}) \]  

(4.59)

\[ Var_{x,y} \left[ Var_{w|x,y}(bf^*) \right] = Cum[BootCum[BF][2][2,4] + O_p(n^{-2}) . \]  

(4.60)

In practice, higher moments of other bootstrap cumulants may be expanded to any order of accuracy with these tools.

For example, the fourth order expansions considered in (4.57) - (4.59) are given by the following:

\[ Cum[BootCum[BF][1][1,4] = \frac{5}{2n^2} \left( Cum(X, X, X) - Cum(Y, Y, Y) \right) \]  

(4.61)
\[ \text{Cum[BootCum[BF][1]][2, 4]} = \left[ -2 \text{Cum}(X, X) + 6 \text{Cum}(X, X)^2 + n \text{Cum}(X, X)^2 - 2 \text{Cum}(Y, Y) + 4 \text{Cum}(X, X) \text{Cum}(Y, Y) + 2 n \text{Cum}(X, X) \text{Cum}(Y, Y) + 6 \text{Cum}(Y, Y)^2 + n \text{Cum}(Y, Y)^2 \right] / \left[ n^2 (\text{Cum}(X, X) + \text{Cum}(Y, Y))^2 \right] \] (4.62)

\[ \text{Cum[BootCum[BF][2]][1, 4]} = \left[ -2 \text{Cum}(X, X) + 5 \text{Cum}(X, X)^2 + n \text{Cum}(X, X)^2 - 2 \text{Cum}(Y, Y) + 2 \text{Cum}(X, X) \text{Cum}(Y, Y) + 2 n \text{Cum}(X, X) \text{Cum}(Y, Y) + 5 \text{Cum}(Y, Y)^2 + n \text{Cum}(Y, Y)^2 \right] / \left[ n^2 (\text{Cum}(X, X) + \text{Cum}(Y, Y))^2 \right] \] (4.63)

Since the fourth order expansion considered in (4.60) is lengthy we provide below the corresponding second order expression:

\[ \text{Cum[BootCum[BF][2]][2, 2]} = \left[ 2 \text{Cum}(X, X)^2 + 2 \text{Cum}(Y, Y)^2 + \text{Cum}(X, X, X, X) + \text{Cum}(Y, Y, Y, Y) \right] / \left[ n (\text{Cum}(X, X) + \text{Cum}(Y, Y))^2 \right] \] (4.64)

The six expressions (4.55), (4.56), and (4.61) - (4.64) represent examples of series expansions of the six bootstrap quantities associated with Table 4.2. They constitute symbolic representations, in fact, formulas, which may subsequently be applied to any pair of independent data sets for which numerical computations of the Behrens-Fisher statistic, its bootstrap moments, or properties of these are desired. Moreover, (4.55), (4.56), and (4.61) - (4.64) represent analytic bootstrap expressions which may be
evaluated numerically for any pair of distributions $F$ and $G$ associated with the independent variables $x$ and $y$ for which cumulants are known. In particular, we will use the empirical (nonparametric) distribution functions $F_n$ and $G_n$.

We next illustrate the numerical evaluation of these expansions within the context of a case study from medicine.

### 4.4.4 Numerical Application

Table 4.3 shows the original survival times of 88 patients with gastric cancer equally divided into two independent treatment groups. The data come from Gamerman (1991). Group 1 received chemotherapy and radiation while Group 2 received only chemotherapy. Interest lies in comparing the average survival times of the two groups. Without loss of generality, prior to evaluation each of the two data sets was standardized to have zero means. It is assumed that both samples come from independent populations possessing unequal variances.

<table>
<thead>
<tr>
<th>Therapy 1</th>
<th>Therapy 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>383</td>
</tr>
<tr>
<td>42</td>
<td>383</td>
</tr>
<tr>
<td>44</td>
<td>577</td>
</tr>
<tr>
<td>48</td>
<td>580</td>
</tr>
<tr>
<td>60</td>
<td>795</td>
</tr>
<tr>
<td>72</td>
<td>855</td>
</tr>
<tr>
<td>74</td>
<td>1174</td>
</tr>
<tr>
<td>95</td>
<td>1214</td>
</tr>
<tr>
<td>103</td>
<td>1232</td>
</tr>
<tr>
<td>108</td>
<td>1366</td>
</tr>
<tr>
<td>122</td>
<td>1455</td>
</tr>
<tr>
<td>144</td>
<td>1585</td>
</tr>
<tr>
<td>167</td>
<td>1622</td>
</tr>
<tr>
<td>170</td>
<td>1626</td>
</tr>
<tr>
<td>183</td>
<td>528</td>
</tr>
</tbody>
</table>

The 2nd and 4th order expansions of the six bootstrap quantities associated with Table 4.2 could be evaluated numerically for any underlying distributions for which
Table 4.4: Symbolic bootstrap and classical results for the Behrens-Fisher statistic using the survival data, \( n = 44 \)

| Analytical Bootstrap | Empirical Distribution \( F_n \) and \( G_n \) 
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2nd order</td>
</tr>
<tr>
<td>( E_w</td>
<td>x,y(bf^*) )</td>
</tr>
<tr>
<td>( E_{x,y}E_w</td>
<td>x,y(bf^*) )</td>
</tr>
<tr>
<td>( [Var_{x,y}[E_w</td>
<td>x,y(bf^*)]]^{1/2} )</td>
</tr>
<tr>
<td>( [Var_w</td>
<td>x,y(bf^*)]^{1/2} )</td>
</tr>
<tr>
<td>( Var_w</td>
<td>x,y(bf^*) )</td>
</tr>
<tr>
<td>( E_{x,y}[Var_w</td>
<td>x,y(bf^*)] )</td>
</tr>
<tr>
<td>( [Var_{x,y}[Var_w</td>
<td>x,y(bf^*)]]^{1/2} )</td>
</tr>
<tr>
<td>Classical</td>
<td></td>
</tr>
<tr>
<td>( bf )</td>
<td>0</td>
</tr>
<tr>
<td>( E_{x,y}(bf) )</td>
<td>-0.01235</td>
</tr>
<tr>
<td>( Var_{x,y}(bf) )</td>
<td>1</td>
</tr>
</tbody>
</table>

cumulants exist. Here we evaluate the analytic expansions for the empirical distribution functions \( F_n \) and \( G_n \) because these are considered close to the true underlying \( F \) and \( G \). Hence, the data in Table 4.3 are used to numerically evaluate the symbolic bootstrap expressions. The results were then scaled by appropriate powers of \( \sqrt{n} \) so that they be associated with (4.48) instead of (4.49). The final results are shown in Table 4.4. For comparison purposes the bootstrap mean and bootstrap standard error of the Behrens-Fisher statistic have also been estimated using conventional Monte Carlo resampling for various values of \( B \). The algorithm upon which these calculations are based is given in Efron and Tibshirani (1993, p. 224). The simulation results are presented in Table 4.5.

### 4.4.5 Discussion

We observe that the numerical evaluations of the 2nd and 4th order symbolic expressions for the bootstrap mean and the bootstrap standard error of the Behrens-Fisher statistic given by \( E_w|x,y(bf^*) \) and \( [Var_w|x,y(bf^*)]^{1/2} \) in Table 4.4 are almost identical to the corresponding Monte Carlo calculations seen in Table 4.5. (The latter computa-
Table 4.5: Monte Carlo Bootstrap Estimates of Mean and Standard Error of the Behrens-Fisher statistic for various B for the survival data, $n = 44$ (s.e. in brackets)

<table>
<thead>
<tr>
<th>B</th>
<th>Bootstrap mean</th>
<th>Bootstrap s.e.</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.028 (0.155)</td>
<td>1.093 (0.125)</td>
</tr>
<tr>
<td>100</td>
<td>-0.088 (0.108)</td>
<td>1.077 (0.077)</td>
</tr>
<tr>
<td>200</td>
<td>-0.060 (0.072)</td>
<td>1.022 (0.058)</td>
</tr>
<tr>
<td>400</td>
<td>-0.015 (0.049)</td>
<td>0.980 (0.036)</td>
</tr>
<tr>
<td>800</td>
<td>-0.000 (0.035)</td>
<td>0.988 (0.024)</td>
</tr>
<tr>
<td>1600</td>
<td>0.018 (0.025)</td>
<td>1.008 (0.019)</td>
</tr>
<tr>
<td>3200</td>
<td>-0.028 (0.018)</td>
<td>1.018 (0.013)</td>
</tr>
<tr>
<td>6400</td>
<td>-0.009 (0.013)</td>
<td>1.029 (0.009)</td>
</tr>
<tr>
<td>12,800</td>
<td>-0.020 (0.009)</td>
<td>1.002 (0.007)</td>
</tr>
</tbody>
</table>

Considerations representing estimates of bootstrap standard error are multiplied by $\sqrt{(n - 1)/n}$ when compared to the symbolic-based computations.) This demonstrates that the analytic bootstrap is capable of providing bootstrap estimates without resorting to conventional bootstrap resampling.

Diagnostic properties such as bias and variance of bootstrap moments may be assessed analytically and then evaluated numerically. For example, the bias of the bootstrap variance of the Behrens-Fisher statistic is represented by

$$Bias = E_{x,y}[\text{Var}_{w|x,y}(bf^*)] - \text{Var}_{x,y}[bf].$$ (4.65)

This expression could be evaluated for any distribution, for example, the Gaussian distribution. Using the empirical distribution functions $F_n$ and $G_n$ we may calculate the 4th order bootstrap estimate of bias of the bootstrap variance given by

$$\hat{Bias} = \hat{E}_{x,y}[\text{Var}_{w|x,y}(bf^*)] - \hat{\text{Var}}_{x,y}[bf]$$

$$= 1.06831 - 1.04659$$

$$= 0.02172.$$ (4.66)
Similarly, the 4th order bootstrap estimate of bias of the bootstrap expectation is

\[
\widehat{\text{Bias}} = \hat{E}_{x,y}[E_w|_{x,y}(bf^*)] - \hat{E}_{x,y}[bf] \\
= 0.00140 - (-0.01309) \\
= 0.01449
\]  

(4.67)

These results suggest that the bootstrap procedure for estimating the mean and variance of the Behrens-Fisher statistic is largely free of bias.

### 4.5 Conclusions

The symbolic bootstrap methodology developed in Chapter 3 may be applied to a broad class of statistics, namely those which may be expressed as smooth non-linear functions of averages. The Behrens-Fisher statistic, the acceleration adjustment in \(BC_a\) confidence intervals, and various quantities associated with bootstrap-t confidence intervals all satisfy this criterion. The present chapter has shown in detail how the analytic bootstrap may be applied to these examples. In all cases numeric evaluation of the analytic bootstrap estimates matched computations based on conventional resampling techniques.

If required, the computational tools presented in this chapter with respect to the three examples studied could provide symbolic expansions for any bootstrap cumulant to any desired order as well as properties of these.

Analytic bootstrap expansions constitute symbolic representations, in fact. bootstrap formulas which are functions of various cumulants and residual sums. They are true for any distribution(s) for which the cumulants are known. In particular, the empirical distribution functions were used for numerical evaluation. The symbolic expressions offer two advantages over conventional Monte Carlo resampling. First, the existence of the particular 2nd and 4th order analytic expansions seen in this chapter means that the derivational work they represent does not need to be duplicated by different practitioners; the expressions are available for efficient and repeated
numerical evaluation for different data sets. Second, analytic bootstrap expansions enable the user to identify component statistical quantities such as particular cumulants or residual sums which may influence the magnitude of properties such as the bias or variance of the bootstrap procedure in question. Hence, analytic bootstrap methodology enables the user to better understand the bootstrap procedure being used.
Chapter 5

Concluding remarks

5.1 Remarks

In this thesis we established the representation of bootstrap moments and their properties as analytical expansions involving cumulants and standardized sums. This methodology was developed for a broad class of statistics, namely, the M-estimators. The approach was illustrated by providing examples involving the correlation coefficient, the Behrens-Fisher statistic, double-bootstrapping, and $BC_a$ bootstrap confidence intervals. Numerical evaluations of the symbolic bootstrap expressions were shown to be consistent with analogous Monte Carlo approximations. An alternative methodology based on the exhaustive enumeration of bootstrap resamples was also presented and illustrated for the class of L-estimators, in particular, the median. The exhaustive enumeration approach used for order statistics provides exact analytical bootstrap expressions and although the approach may also be applied for more general estimators it has sample size restrictions. The use of asymptotic expansions in the case of M-estimators has no such limitation and the symbolic bootstrap quantities may be obtained to an arbitrary precision.

The features/advantages of the analytic bootstrap developed in this thesis may be summarized as they were in Chapter 1 as follows:

1. Analytic derivation and numerical evaluation of bootstrap estimates and their
properties entirely avoids Monte Carlo resampling resulting in faster and more efficient calculation.

2. It can be used diagnostically to assess properties such as bias, variance or other moments of a bootstrap procedure.

3. For the class of L-estimators (linear combinations of order statistics), the analytic bootstrap calculates ideal bootstrap estimators \textit{exactly} for sample sizes up to \( n = 20 \) and calculates properties of these estimators \textit{exactly} for much larger sample sizes.

For the class of M-estimators (maximum likelihood-type estimators), the analytic bootstrap calculates ideal bootstrap estimators and properties of these \textit{exactly} for linear statistics and to within a specified order of accuracy for large sample sizes in the case of non-linear statistics.

4. The analytic bootstrap may be applied to a broad class of statistics, namely, to any estimator which may be expressed as a non-linear smooth function of averages.

5. The analytical approach for the class of M-estimators is \textit{less} computer intensive for larger data sets. The \textit{reverse} is true for conventional bootstrap calculations.

6. Analytic bootstrap expressions are potentially insightful because component statistical quantities such as cumulants and sums are identified.

7. Each analytic bootstrap expression (representing a bootstrap moment or a property of this) is a mathematical \textit{bootstrap formula} which may be further numerically evaluated efficiently for a given data set, or repeatedly evaluated for different data sets arising within a given application. The computational effort required to derive each formula need only be undertaken once. It is subsequently useable in perpetuity by other practitioners for the purpose of efficient numerical evaluation.
This thesis has established that a large class of problems may be addressed by the analytic bootstrap. Further work would be required to investigate the application of the methodology for estimators involving non-iid random variables. The methodology could also be extended to address multi-dimensional parameters. Other work may involve use of the analytic bootstrap on the basis of bootstrap resamples of smaller size $\alpha n$ instead of $n$. Symbolic Cornish-Fisher expansions of bootstrapped quantiles of a statistic $\hat{\theta}$ could be obtained by computing and plugging in various bootstrap cumulants into an Edgeworth expansion of the distribution function of $\hat{\theta}$ and inverting.
Appendix A

Source Code used for this Thesis

In this Appendix we provide the source code that was used for this thesis. This includes the Mathematica code underlying the computational tools developed in Chapters 2 and 3 together with code used for the examples in Chapter 4. SPlus code that was used to carry out comparative Monte Carlo simulations related to these chapters is also given below.

A.1 Source Code used for Chapter 2

This section contains the source code for 5 Mathematica programs associated with the functions developed in Chapter 2, and the source code for examples of 3 programs written in SPlus that were used to conduct simulation checks of the functions.

(* --------------Filename: prob.m -------------- *)

(* This program computes the multinomial-type probability of a given *)
(* set of weights (i.e. a list) t = {n1,n2,n3,...} *)
\text{Prob}[t_] := \text{Block} \{\{c,n,k,l,m,\text{mult},\text{lst},\text{lst2},\text{ls}\} , \\

\quad c = \text{Length}[t]; \\
\quad n = \text{Apply}[\text{Plus},t]; \\
\quad k = c-1; \\
\quad l = n/(n-k-1); \\
\quad m = 1/(n^n); \\

\quad \text{Do} [ \\
\quad \quad \text{If} [ i <= c , f[i] = t[[i]] ; \\
\quad \quad \quad g[i] = \text{Cases}[t,f[i]]; \\
\quad \quad \quad h[i] = \text{Count}[t,f[i]], \\
\quad \quad \quad \quad f[i] = 0; g[i] = \{\} ] , \\
\quad \quad \{i,20\} ] ; \\

\quad \text{lst} = \{g[1],g[2],g[3],g[4],g[5],g[6],g[7],g[8],g[9],g[10]\}; \\

\quad \text{lst2} = \text{Union[lst]}; \\

\quad \text{ls} = \text{Length[lst2]}; \\

\quad \text{Do} [ \\
\quad \quad \text{If} [ j <= ls , w[j] = \text{Length[lst2[[j]]]]; \\
\quad \quad \quad y[j] = w[j]!, \\
\quad \quad \quad w[j] = 0 ] , \{j,ls\} ] ; \\

\quad (*) \text{Do} [\text{Print}[i, " ",f[i], " ",g[i], " ",h[i]], \{i,20\} ] ; (*)
Print[1st2];

(* Do [Print[i, " ", w[i], " ", y[i] ], {i,ls} ]; *)


mult = Multinomial[ f[1], f[2], f[3], f[4], f[5],
                     f[6], f[7], f[8], f[9], f[10],
                     f[11], f[12], f[13], f[14], f[15],
                     f[16], f[17], f[18], f[19], f[20] ];

(* Print[mult]; *)

den = Apply[Times,Table[y[i],{i,ls}]]; 

(* Print[den]; *)

mult * 1 * m/den    //N     ]

(* ------------------------ Filename: os.m ------------------------* )

(* This program computes : *

Med [ { list t of weights } ] = { {r(i) , n , p(r(i))} }
for any list t where
r(i) = the ith order statistic having non-zero probability of
occurrence,
n = number of distinct order statistics available, and
p(r(i)) = probability of occurrence of the ith order statistic.
This is done for ODD sample size only. Note that sum of weights
equals the sample size.  


A

APPENDIX: Source Code

[Image 0x0 to 600x773]

e.g. \( t = \{4,2,1\} \implies \text{Med}[t] = \)
\[ \{\{1, 3, 0.333333\}, \{2, 3, 0.333333\}, \{3, 3, 0.333333\}\} \]

\( t = \{2,2,1,1,1\} \implies \text{Med}[t] = \)
\[ \{\{2, 5, 0.1\}, \{3, 5, 0.8\}, \{4, 5, 0.1\}\} \]

\( t = \{3,3,2,2,2,1\} \implies \text{Med}[t] = \)
\[ \{\{3, 6, 0.5\}, \{4, 6, 0.5\}\} \]

*)

\[ F[t_] := \text{Block} \[ \{ c,j,s,q,os \} \], \]

Do [
    If [ s = Apply[Plus,Take[t,i]]; q = Apply[Plus,Drop[t,i]]; s < q,
        Continue[], os=i;
        Break[] ], {i,Length[t]} ]; os ]

\[ \text{OsPerm}[t_] := \text{Map}[F,\text{Permutations}[t]] \]

\[ F2[t_] := \text{Union} \[ \text{Partition} \[ \text{Map}[F,\text{Permutations}[t]], 1 \] \] \]

\[ \text{Med}[t_] := \text{Block} \[ \{ \} \], \]

Do [
    If [ i <= Length[t], g[i] = Count[OsPerm[t],i];
        p[i] = g[i] / Length[Permutations[t]] //N ],
    {i,Length[t]} ] ;

    tab = Table [ p[i], {i, Length[t]} ];
APPENDIX: Source Code

\[
\text{tab2} = \text{DeleteCases}[	ext{tab}, 0];
\]

\[
\text{Do}[
\quad \text{m}[i] = \text{Flatten} \left[ \text{Append} \left[ \text{F2}[[i]], \{\text{Length}[[t]], \text{tab2}[[i]]\} \right] , \{i, \text{Length}[[\text{tab2}]]\}\right];
\quad \text{Table} [\text{m}[i], \{i, \text{Length}[[\text{tab2}]]\}]\]
\]

(* -------------------------- Filename: partition.m --------------------------*)

(* This code finds all partitions of any positive integer *)

ClearAll[I]
I[I[n_], 0] := I[n]
I[I[n_], I[n2_]] := I[Join[n1, n2]]
I[I[n_], I[n2_]+b_] := I[I[n1], I[n2]] + I[I[n1], b]

ClearAll[P]
P[n_, 1] := I[\{n\}]
P[n_, 1, 1] := P[n, 1]
P[n_, m_] := P[n, m, 1]
P[n_, m_, k_] := 0 \quad ; k>n
P[n_, 2, k_] := I[\{k, n-k\}] \quad ; \text{SameQ}[k, \text{Quotient}[n, 2]]
P[n_, 2, k_] := I[\{k, n-k\}] + 
\quad P[n, 2, k+1] \quad ; (k<\text{Quotient}[n, 2])
P[n_, m_, k_] := J[I[\{k\}], P[n-k, m-1, k]] \quad ; (\text{SameQ}[k, \text{Quotient}[n, m]] \quad \&\&
\quad \text{!SameQ}[m, 2])
P[n_, m_, k_] := P[n, m, k+1] + J[I[\{k\}], P[n-k, m-1, k]] \quad ; (k<\text{Quotient}[n, m] \quad \&\& \text{!SameQ}[m, 2])
ClearAll[FP]
FP[nn___1] := P[nn,1]
FP[nn___,n___] := P[nn,n] + FP[nn,n-1]
FP[nn___] := FP[nn]=FP[nn,nn]

rem[I[a___]] := a
FPL[nn___] := Map [ rem, Apply[List,FPL[nn]] ]

(* -------------------------------Filename: Expboot.m ------------------------------- *)

(* This computes the expectation of the bootstrap estimate of the second
 moment of a sample median for given odd sample size and lambda.
 It uses the preceding three programs.
 The function SecondMomOrder[r___,n___,l___] is defined here. *)

<<prob.m
<<os.m
<<partition.m

mbi[a___ + b___] := mbi[a] + mbi[b];
mbi[a___ b___] := a mbi[b] /;FreeQ[a,u];
mbi[u a___ (1-u) b___] := Beta[a+1,b+1];
mbi[u_] = Beta[2,1];
mbi[u a___] := Beta[a+1,1];
mbi[(1-u) b___] := Beta[1,b+1];

x[l___] := u l - (1-u) l
APPENDIX: Source Code

\[ f[r_, n_] := \text{Beta}[r, n-r+1]^{-1} \cdot u^{-r-1} \cdot (1-u)^{n-r} \]

\[ \text{SecondMomOrder}[r_, n_, l_] := \text{mbi}[\text{Expand}[(x[l])^2 \cdot f[r, n]]] \]

\[ \text{ExpBoot}[n_, l_] := \text{Block} \{ \{ \} \}, \]

\[ a = \text{Map}[	ext{Prob}, \text{FPL}[n]]; \]
\[ b = \text{Map}[	ext{Med}, \text{FPL}[a]]; \]

\[ \text{Do} [ \]
\[ \text{If} [ \text{Length}[b[[i]]] == 1, c[i] = \text{Flatten}[b[[i]]]]; \]
\[ d[i] = a[i] \cdot \text{SecondMomOrder}[c[i][[1]], c[i][[2]], 1], \]
\[ \text{Do} [ e[j] = b[[i]][[j]][[3]] \cdot \text{SecondMomOrder}[b[[i]][[j]][[1]], \]
\[ b[[i]][[j]][[2]], 1], \]
\[ {j, \text{Length}[b[[i]]]} ]; \]
\[ d[i] = \text{Apply}[	ext{Plus}, \text{Table}[e[j], \{j, \text{Length}[b[[i]]]\}]] \cdot a[[i]] \]
\[ , \{i, \text{Length}[b]\} ] \]
\[ \]
\[ \text{Print}[a, "", b, "", e[1], "", e[2], "", d[1], "", d[2], "", d[3]]; \]

\[ \text{Apply}[	ext{Plus}, \text{Table} \{ d[i], \{i, \text{Length}[a]\}\}] \]

\*[\text{----------------------------------------Filename: Varboot.m ----------------------------------------}]

\*[\text{This computes the variance of the bootstrap estimate of}]
\*[\text{the second moment of a sample median for given odd sample size}]
\*[\text{and lambda. The function VarSqOrder[r_,n_,l_] is defined here}]}
and this depends on SecondMomOrder[r_, n_, l_].. 

<<prob.m
<<os.m
<<partition.m

mbi[a_ + b_] := mbi[a] + mbi[b];
mbi[a_ b_] := a mbi[b] /; FreeQ[a, u];
mbi[u'a_. (1-u)'b_.] := Beta[a+1, b+1];
mbi[u] = Beta[2, 1];
mbi[u'a_.] := Beta[a+1, 1];
mbi[(1-u)'b_.] := Beta[1, b+1];

x[1_] := u'1 - (1-u)'1
f[r_, n_] := Beta[r, n-r+1]'(-1) u'(r-1) (1-u)'(n-r)
SecondMomOrder[r_, n_, l_] := mbi[Expand[(x[1])^2 f[r, n]]]
mv2[r_, n_, l_] := mbi[Expand[(x[1])^4 f[r, n]]]
VarSqOrder[r_, n_, l_] := mv2[r, n, l] - (SecondMomOrder[r, n, l])^2

VarBoot[n_, l_] := Block [ { } ,

   a = Map[Prob, FPL[n]]^2;
   b = Map[Med, FPL[n]];

   Do [
      If [ Length[b[[i]]] == 1, c[i] = Flatten[b[[i]]];
         d[i] = a[[i]] VarSqOrder[c[i][[1]], c[i][[2]], l],
      Do [ e[j] = b[[i]][[j]][[3]] VarSqOrder[b[[i]][[j]][[1]],

]
APPENDIX: Source Code

\[
\begin{align*}
&b[[i]][[j]][[2]], 1, \\
&\{j, Length[b[[i]]]\} ]; \\
&d[i] = \text{Apply[Plus,Table[e[j],\{j,Length[b[[i]]]\}]] a[[i]]} \\
&\{i, Length[b]\} ] \\
&\text{;} \\
&\text{Print[a, ",", b, ",", e[1], ",", e[2], ",", d[1], ",", d[2], ",", d[3]];}
\end{align*}
\]

Apply[Plus,Table[ d[i], \{i,Length[a]\}]] }

(* ------------------------ SPLUS code ------------------------*)

(* ------------------------ boot.norm.3.B1 ------------------------*)

(* This was used to simulate the ExpBoot function *)
(* The particular example here assumes a Gaussian shape for
  a sample size of 3. There were 12 combinations in all *)

l_0.135
B_1

mat_matrix(runif(3000),3,1000)

boot_function(x){
  med_rep(0,B)
  for(i in 1:B){
    junk_sample(x,length(x),replace=T)
    med[i]_median(junk)^1 - (i-median(junk))^1
  }
  return(med)
}
APPENDIX: Source Code

med_apply(mat,2,boot)  # med contains 1000 transformed median values
mean1_mean(med^2)
stderr1_sqrt(var(med^2)/1000)

# Examples
# mean1 : 0.03669427  stderr1 : 0.001580984

(* ------------------ sim.norm.3 ------------------------------- *)

# This is a simulation CHECK of the Mathematica function
# SecondMomOrder[i,n,1] (= mv[r,n,l]) = true value of E(median^2),
# that is, for the case where the i-th order statistic happens to be the
# median in a sample of size n .

# Below 1000 samples of size 3 have been simulated from a U[0,1] distribution.
# The particular example here assumes a Gaussian shape for
# a sample size of 3. (12 combinations)

mu_matrix(runif(3000),3,1000)
u.med_apply(mu,2,median)
x_=(u.med^0.135) - (1-u.med)^0.135  # x resembles a Normal since lambda = 0.135
mean1_mean(x^2)
stderr1_sqrt(var(x^2)/1000)

(* ------------------ boot.norm.3 ------------------------------- *)

# --- This is the conventional bootstrap, selecting 1000 resamples
# --- from the initial sample of size 3, calculating the transformed median,
# --- then the average of their squares.
# --- This is bootstrap_reg.
# The particular example here assumes a Gaussian shape for
# a sample size of 3. (12 combinations)

l_0.135
B_1000

mat_matrix(runif(3),3,1)

boot_function(x){
    med_rep(0,B)
    for(i in 1:B){
        junk_sample(x,length(x),replace=T)
        med[i]_median(junk)^1 - (1-median(junk))^1
    }
    return(med)
}

med_apply(mat,2,boot) # med contains 1000 transformed median values
mean1_mean(med^2)
stderr1_sqrt(var(med^2)/1000)

(* -------------------------------- END ---------------------------------* )
A.2 Source Code used for Chapter 3

This section contains the Mathematica code that was used in Chapter 3 for the symbolic derivation and numerical evaluation of bootstrap estimators and corresponding properties associated with M-estimators. Code connected with the case of the correlation coefficient is also included. Splus code for producing comparative Monte Carlo simulations for the correlation coefficient is given at the end of this section.

(* ------------------------ Filename : Bootcum.m ------------------------*)

(* This contains the code for : ThetaHat[i] and BootCum[thetahat][i,j] *)
(* Supporting code called astat.3.0.new.m is associated with Andrews et al. (1993). *)

(* Change Cums to Gmoms if rv is x1 *)

Cum[{a_}] := Cum[a]
Mom[{a_}] := Mom[a]
Cum[a_] := Mom[a]/;!FreeQ[a,X1[w]]
Cum[a_,b_] := CfromM[a,b]/;!FreeQ[{a,b},X1[w]]
Mom[a_] := Gmom[a]/;!FreeQ[a,X1[w]]
Mom[a_,b_] := Gmom[Apply[Times,{a,b}]]/;!FreeQ[a,X1[w]]

(* Read in definitions *)

<<astat.3.0.new.m

$Echo = "stdout"

(* Define psi function, root etc. *)
APPENDIX: Source Code

\[ \Psi = \text{psifn}[X[w], #1] \& \]
\[ \text{NrootO}[\text{Cum}[\Psi]] = \theta \]
\[ \text{thetahat} = \text{Nroot}[\text{AF}[n][\Psi]]; \]

(* empir replaces F with Fn in stat *)
(* eCum stands for empirical Cum *)

\[ \text{rsn}[a_] := \text{rsm}[a]/.X[w] \to Xstar[w] \]
\[ \text{NT}[\text{empir}[a_], i_] := \text{Nget}[\text{Expand}[\text{Nsum}[a[i]], i] \]
\[ /./\text{RS}[n] \to \text{rsm}/.\text{RS}[n]/.\text{Cum} \to \text{tCum}/.\text{tCum} \to \text{Cum} \]
\[ \text{tCum}[a_] := s[\text{Length}[\{a\}]] \text{eCum}[a] /;!\text{FreeQ}[\{a\}, Xstar] \]

(* eCum[a_] := n^{-1} S[n][a]/.X[w] \to Xstar[w] *)

(* X* = Xstar, i.e. the weighted or bootstrapped X *)

\[ \text{ThetaHat}[i_] := \text{Nsum}[\text{thetahat}, i] \]

\[ \text{BootCum}[\text{thetahat}][i_, j_] := \]
\[ \text{Nsum}[\text{empir}[\text{Cum}[\text{empir}[\text{Npower}[\text{thetahat}, 1]], i]], j] \]

(* In practice, for a given statistic, this BootCum code must use the tool Nboot. See for example the code used to derive bootstrap moments of the correlation coefficient below. *)

(*
\[ \text{ExpBootExp}[i_, j_, k_] := \]
\[ \text{Nsum}[\text{Ncum}[\text{empir}[\text{Cum}[\text{empir}[\text{Npower}[\text{thetahat}, 1]], k]], j], i] \]
RVQ::usage = "RVQ[a] returns True iff a is a random variable."

ClearAll[RVQ]
RVQ[a_,b_]:= If[RVQ[a],True,RVQ[b]]
RVQ[g_[a_]] :=If[MemberQ[ConsExpectFns,g],False,
If[MemberQ[RandomFns,g],True,RVQ[a]]]
RVQ[z_]:= False
ConsExpectFns = {Expect, Moment, CentralMoment, Theta,Cum,ED,InvED,ICum};
RandomFns = {RD, RS,DF};
AllRVQ[a_,b_]:= If[RVQ[a], AllRVQ[b],False]
APPENDIX: Source Code

AllRVQ[a_] := RVQ[a];

VQ::usage = "VQ[a] returns True iff a is a variable."
(*
  :[font = output; output; inactive]
  "VQ[a] returns True iff a is a variable."
  ;[0]
VQ[a] returns True iff a is a variable.
  :[font = input; initialization; endGroup]
*)

ClearAll[VQ]
VQ[a_, b_] := If[VQ[a], True, VQ[b]]
VQ[g_[a___]] := If[MemberQ[ConsFns, g], False, If[MemberQ[VarFns, g], True, VQ[a]]]
VQ[z_] := False
ConsFns = {SUM};
VarFns = {};
AllVQ[a_, b___] := If[VQ[a], AllVQ[b], False]
AllVQ[a_] := VQ[a];

FP1::usage = "FPQ[a,b,..] returns the full partition of a,b,..."
(*
  :[font = output; output; inactive]
  "FPQ[a,b,..] returns the full partition of a,b,..."
  ;[0]
FPQ[a,b,..] returns the full partition of a,b,...
  :[font = input; initialization]
*)
SetAttributes[PE, Orderless]

ClearAll[addn1]
addn1[n_, a_ + b_] := addn1[n, a] + addn1[n, b]
addn1[n_, (a_ + b_)c_] := addn1[n, a c] + addn1[n, b c]
addn1[n_, a_ PE[b_]] := a addn1[n, PE[b]]
addn1[n_, PE[a_]] :=
Apply[Plus, Table[Apply[PE, addn2[{a}, n, iadd]], {iadd, 1, Length[{a}]})] +
Append[PE[a], {n}]

ClearAll[addn2]
addn2[a_, n_, i_] :=
Table[If[jaddn2 == i, Append[a[[i]], n], a[[jaddn2]]],
jaddn2, 1, Length[a]]

ClearAll[FP1]
FP1[inlist_] := Expand[FP1[PE[inlist], Length[inlist]]]
FP1[pe_, 1] := pe
FP1[a_ + b_] := FP1[a] + FP1[b]
FP1[c_, PE[a_], n_] := c FP1[PE[a], n]
FP1[PE[a_], n_] :=
addn1[Last[a], FP1[PE[Drop[a, -1]], n-1]]

EV::usage =
"EV[a] returns the expected value of a"
(*
:[font = output; output; inactive]
"EV[a] returns the expected value of a"
;o}
APPENDIX: Source Code

EV[a] returns the expected value of a

ClearAll[EV]
EV[a_ + b_] := EV[a] + EV[b]
EV[(a_ + b_)^c_ d_] := EV[Expand[(a+b)^c d]]/;
((c > 0) && (IntegerQ[c]))
EV[a_ b_] := a EV[b] /; !RVQ[a]
EV[a_] := a /; !RVQ[a]
EV[SUM[a_]] := SUM[EV[a]]
EV[SUM[a_] c_] := EV[c, a]
EV[SUM[a_] c_, b_] := EV[c, a, b]
EV[SUM[a_] b_ c_] := EV[SUM[a]^(b-1)c, a]
EV[SUM[a_] b_ c_, d_] := EV[SUM[a]^(b-1)c, a, d]
EV[SUM[a_]^b_ c_, d_] := EV[SUM[a]^(b-1)c, a, d]
EV[1, a_] := Collect[Expand[Map[EVPE, FP1[{a}]]], SUM]

ClearAll[EVPE]
EVPE[(a_ + b_)] := EVPE[a] + EVPE[b]
EVPE[(a_ + b_) c_] := EVPE[a c] + EVPE[b c]
EVPE[a_ PE[b_]] := a EVPE[PE[b]]
EVPE[PE[a_]] := SUM[EV[Apply[Times, a]]]
EVPE[PE[a_]] := Apply[SUM, Map[EV[Apply[Times, #1]]& , PE[a]]]

BEV::usage =

"BEV[a] returns the bootstrap expected value of a"
(*
 :[font = output; output; inactive]
"BEV[a] returns the bootstrap expected value of a"
 ;[o]
BEV[a] returns the bootstrap expected value of a
A

APPENDIX: Source Code

*)
ClearAll[BEV]
BEV[a_ + b_] := BEV[a] + BEV[b]
BEV[(a_ + b_)^c_. d_] := BEV[Expand[(a+b)^c d]]/
((c > 0) && (IntegerQ[c]))
BEV[a_ b_] := a BEV[b]/;!RVQ[a]
BEV[a_] := a /;!RVQ[a]
BEV[SUM[a_]] := SUM[a]
BEV[SUM[a_] c_] := BEV[c,a]
BEV[SUM[a_] c_,b_] := BEV[c,a,b]
BEV[SUM[a_]^b_ c_] := BEV[SUM[a]^(b-1)c,a]
BEV[SUM[a_]^b_ c_,d_] := BEV[SUM[a]^(b-1)c,a,d]
BEV[1,a_] := Collect[Expand[Map[BEVPE,FP1[a]]],SUM]

ClearAll[BEVPE]
BEVPE[(a_ + b_)] := BEVPE[a ] + BEVPE[b ]
BEVPE[(a_ + b_)c_] := BEVPE[a c] + BEVPE[b c]
BEVPE[a_ PE[b_]] := a BEVPE[PE[b]]
BEVPE[PE[a_]] := wt[Length[a]]SUM[Apply[Times,a]]
BEVPE[PE[a_]] :=
wtpe[PE[a]]Apply[SUM,Map[Apply[Times,#1&,a]]]

ClearAll[mgf]
tv={t1,t2,t3,t4,t5,t6,t7,t8,t9}
tvrules = Table[tv[[i]]->0,{i,1,9}]
mgf [0]=((1/n)(Sum[Exp[tv[[i]]],{i,1,9}] + n-9)^n;
mgf[i_] := mgf[i] = D[mgf[i-1],{tv[[1]],1}]
mgf[a_,0] := mgf[a]
mgf[a_,i_] := mgf[a,i] = D[mgf[a,i-1],{tv[[Length[{a}]] +1]],1}]
\[\text{wtpe[PE[a_\_] := Apply[wt, Map[Length, \{a\}]]}\]

\text{ClearAll[wt]}
\text{SetAttributes[wt, Orderless]}
\text{wt[a_] := wt[a] = mgf[a]/.tvrules}

\text{BVAR::usage =}
"BVAR[a] returns the bootstrap variance of a"

\((*)\)
\text{[\text{font = output; output; inactive}]}
"BVAR[a] returns the bootstrap variance of a"
\text{[o]}
\text{BVAR[a] returns the bootstrap variance of a}
\text{[\text{font = input; initialization; endGroup}]}
\text{*)}
\text{BVAR[a_] := Collect[BEV[a^2] - BEV[a]^2, SUM]}

\text{SUM::usage =}
"SUM[a] represents the sum of a_i
SUM[a,b,..] represents the sum of a_i,b_j,...
over i != j != ..."
\((*)\)
\text{[\text{font = output; output; inactive}]}
"SUM[a] represents the sum of a_i  SUM[a,b,..] represents\the sum of a_i,b_j,...  over i != j != ..."
\text{[o]}
\text{SUM[a] represents the sum of a_i  SUM[a,b,..] represents\the sum of a_i,b_j,...  over i != j != ...}
\text{[\text{font = input; initialization}]}

ClearAll[SUM]
SetAttributes[SUM, Orderless]
SUM[a_] := n a; !VQ[a]
SUM[a_ + b_] := SUM[a] + SUM[b]
SUM[a_ + b_, c_] := SUM[a, c] + SUM[b, c]
SUM[a_, b_] := a SUM[b] /; !VQ[a]
SUM[a_] :=
Collect[Expand[Map[CSUM[Length[{a}], #1] &, FP1[{a}]]], SUM];
Length[{a}] > 1

ClearAll[CSUM]
CSUM[l_, a_ + b_] := CSUM[l, a] + CSUM[l, b]
CSUM[l_, a_ b_] := a CSUM[l, b] /; !VQ[a]
CSUM[l_, (b_ + c_)^d_] :=
CSUM[l, Expand[(b + c)^d]] /; IntegerQ[d] && (d > 0)
CSUM[l_, PE[a__]] := Block[{la},
   la = Length[{a}];
   (-1)^(1-la) *
   Apply[Times, Map[(Length[#1] - 1)!SUM2[#1] & , {a}]]]
SUM2[{a__}] := SUM[Times[a]]

Nboot::usage =
"NT[Nboot[a_, i]] represents the ith term of the bootstrap
expectation of a";

NT[Nboot[a_, i_]] := NT[Nboot[a], i] =
Collect[BootSum[a, i, 0], RS]
BootSum[a_, i_, j_] := Nget[BootTerm[a, j], i-j] +
If[j < i, BootSum[a, i, j+1], 0]
APPENDIX: Source Code

BootTerm[a_,i_] := BootTerm[a,i] =
Collect[Expand[(BEV[NT[a,i]/.RS[n]->RtoS]/.SUM->StoR)],n]
StoR[zm[a_]] := n^(1/2)RS[n][a]
RtoS[a_] := n^(-1/2)SUM[ZM[a]]
StoR[a_] := S[n][a/.ZM->ZtoX]
ZtoX[a_] := a - Cum[a]

EmpCum[a_] := n^(-1)SUM[a]
EmpCum[a_] := Map[SignedEV,FP1[{a}]]
SignedEV[b_ PE[a_]] := b SignedEV[PE[a]]
SignedEV[PE[a_]] :=
(-1)^(Length[{a}]-1)(Length[{a}]-1)! *
Apply[Times,Map[BAV,Map[Apply[Times,#1,&,{a}]]]
BAV[a_] := n^(-1) SUM[a]

(* Example *)

RVQ[X]=True
VQ[X] = True
EmpCum[X,X,X,X]

(* Example *)

<<BSTAT.1.0.new.m  (* bootstrap code *)
<<astat.3.0.new.m

(* Some Examples *) .
RVQ[X1]=True
RVQ[X2] = True
RVQ[X3]=True
VQ[X1]=True
VQ[X2] = True
VQ[X3]=True;
SUM[X1]
SUM[X1,X2]
SUM[X1,X2,X3]
SUM[X1 X2 X3] + SUM[X1,X2 X3] + SUM[X1 X2, X3] +
SUM[X1 X3,X2] +SUM[X1,X2,X3]
EV[SUM[X1]SUM[X2]]
BEV[SUM[X1]]
BEV[SUM[X1]SUM[X2]]
BEV[SUM[X1]SUM[X2]SUM[X3]]
BVAR[SUM[X1]/n]
BVAR[BVAR[SUM[X1]/n]]
LX/:VQ[LX[w]]=True
LY/:VQ[LY[w]]=True;
Cum[LX[w]]=0
Cum[LY[w]]=0
LX/:Cum[LX[w],LX[w]]=1
LY/:Cum[LY[w],LY[w]]=1
LX/:Cum[LX[w],LY[w]]=rho

cov/:NT[cov,i_] := NT[cov,i] =
If[i==0,Cum[LX[w],LY[w]],
If[i==1,RS[n][LX[w]LY[w]] -
Cum[LX[w]]RS[n][LY[w]] - Cum[LY[w]]RS[n][LX[w]],
If[i==2,-RS[n][LX[w]]RS[n][LY[w]],0]]
vax/:NT[vax,i_] :=
If[i==0,Cum[LX[w],LX[w]],
If[i==1,RS[n][LX[w]LX[w]] -
Cum[LX[w]]RS[n][LX[w]] - Cum[LX[w]]RS[n][LX[w]],
If[i==2,-RS[n][LX[w]]RS[n][LX[w]],0]]
vay/:NT[vay,i_] :=
If[i==0,Cum[LY[w],LY[w]],
If[i==1,RS[n][LY[w]LY[w]] -
Cum[LY[w]]RS[n][LY[w]] - Cum[LY[w]]RS[n][LY[w]],
If[i==2,-RS[n][LY[w]]RS[n][LY[w]],0]]]

cor/:NT[cor,i_] := NT[cor,i] =
Collect[Expand[NT[Ntimes[Nf[#1^(-1/2)&,Ntimes[vax,vay]],cov],i]],RS]

Cor[i_] := Nsum[cor,i]

NT[cor,2]
Nsum[cor,2]
NT[Nboot[cor],0]
NT[Nboot[cor],1]
NT[Nboot[cor],2]
v2=Expand[Nsum[Nboot[Npower[cor,2]] - Npower[Nboot[cor],2],2]]
Collect[(v2/.RS[n]->RtoS/.Cum->EmpCum),SUM];

(**************************************************************************)

(************************************************************************** Filename : Cor2.m**************************************************************************) 

(* This computes 2nd order symbolic bootstrap expansions*)
for the correlation coefficient and numerically evaluates these for two data sets.

4th order code is similar.

<<Cor1.m
$Echo = "stdout"

(*) This is for \( n = 15 \) AND \( n = 100 \) (*)

(* Uses NEW versions of both BSTAT and ASTAT ! *)

(* Calculates cor2, e2 = BootExp[Cor], v2 = BootVar[Cor] here *)

(* Plus TeXForm and numerical versions for e2, v2 *)

(* Assuming empirical distribution Fn *)

(* i.e. Set RS[n] \rightarrow 0 \) below, and use empirical cums *)

(* The values below are correctly standardized ! *)

(* -----------------------------------------------*)

texr = \{ RS \rightarrow Z, LX[w] \rightarrow X, LY[w] \rightarrow Y \}

(* Expansion of correlation *)

Cor[i_] := Nsum[cor,i]

cor2 = Nsum[cor,2] /. texr

incor2 = TeXForm[cor2]

incor22 = InputForm[cor2]

(* Expansion of bootstrap mean of the correlation *)
**APPENDIX: Source Code**

e2 = Expand[Nsum[Nboot[cor],2]]
inee2 = TeXForm[e2]
inee22 = InputForm[e2]

(* Expansion of E - over x - of bootstrap mean of the correlation *)

ee2 = Expand[Nsum[Ncum[Nboot[cor],1],2]]
inee2 = TeXForm[ee2]
inee22 = InputForm[ee2]

(* Expansion of Var - over x - of bootstrap mean of the correlation *)

ve2 = Expand[Nsum[Ncum[Nboot[cor],2],2]]
inve2 = TeXForm[ve2]
inve22 = InputForm[ve2]

(* ------------------------------- *)

(* Expansion of bootstrap variance of the correlation *)

v2=Expand[Nsum[Nboot[Npower[cor,2]] - Npower[Nboot[cor],2],2]]
inv2 = TeXForm[v2]
inv22 = InputForm[v2]

(* Expansion of E - over x - of bootstrap variance of the correlation *)

ev2 = Expand[Nsum[Ncum[Nboot[Npower[cor,2]] - Npower[Nboot[cor],2],1],2]]
inenv2 = TeXForm[ev2]
inenv22 = InputForm[ev2]

(* Expansion of Var - over x - of bootstrap variance of the correlation *)
APPENDIX: Source Code

vv2 = Expand[Nsum[Ncum[Nboot[Npower[cor, 2]] - Npower[Nboot[cor], 2]], 2], 2]

ininvv2 = TeXForm[vv2]

ininvv22 = InputForm[vv2]

e2SUM = Collect[(e2/.RS[n] -> (0&)/.Cum->EmpCum), SUM];
ee2SUM = Collect[(ee2/.RS[n] -> (0&)/.Cum->EmpCum), SUM];
ve2SUM = Collect[(ve2/.RS[n] -> (0&)/.Cum->EmpCum), SUM];
v2SUM = Collect[(v2/.RS[n] -> (0&)/.Cum->EmpCum), SUM];
ev2SUM = Collect[(ev2/.RS[n] -> (0&)/.Cum->EmpCum), SUM];
vv2SUM = Collect[(vv2/.RS[n] -> (0&)/.Cum->EmpCum), SUM];

PLUS[a_] := PLUS[a]

(* The following is for  n = 15 and n = 100  *)

e2num1 = e2SUM /. {SUM->PLUS, LX[w] -> {-0.600997, 0.860219, -1.04679, 
  -0.551465, 1.62798, 
  -0.501932, -1.12109, 1.50414, 1.25648, 0.117227, 
  1.30601, -0.625764, -1.36875, 0.700063, -0.155203 },
  LY[w] -> {1.25538, 0.872812, -1.21003, -0.274879, 1.46791, 
  -0.104851, -0.4024, 1.4254, 1.12785, 0.150192, 
  0.107685, -1.50758, 
  -1.42257, -0.912485, -0.572429},
  n->15 ,
  rho -> 0.776374 } / . PLUS -> Plus

e2num2 = e2SUM /. {SUM->PLUS, LX[w] -> 
  {-0.956714, 0.686892, -0.564691, -1.14333, -0.0559606, -0.874619, 
  -0.436588, 1.12566, -0.793298, 0.297573, 0.249147, 0.0330676, 0.22594,
\[\begin{array}{l}
-1.21993, 0.318041, 0.291848, 0.0120344, 1.62495, 1.49834, -1.8793, \\
-1.30035, -1.40383, -1.37385, 0.392384, -1.08785, 1.08452, -0.948535, \\
-0.750636, -0.528461, -1.03515, 0.299149, -0.0431774, -1.81281, \\
0.933136, -0.914287, -0.353526, -0.0696765, -1.11197, 0.239538, \\
-0.854338, 2.07333, -0.0952752, -0.6538, -0.248606, -0.857544, 0.284146, \\
1.71851, -0.875686, -0.935791, 0.0546828, -0.662648, -0.130106, \\
-2.59892, 1.29063, 0.61013, 0.0920582, -0.665875, -0.394396, -0.0426476, \\
0.373098, 0.0048362, -1.41674, 0.926444, 0.353601, -0.203354, 0.320726, \\
0.161122, 0.471737, -1.17339, 0.489503, 0.39107, 1.95425, 0.0301679, \\
0.511621, -1.41423, 0.425221, -0.558879, -1.2051, 1.18088, 0.762738, \\
1.36058, 0.25875, 1.0521, 0.665511, -0.0377158, 0.353328, 0.0265674, \\
-0.51515, 0.887074, -1.72355, -0.262615, -0.0142491, 2.73824, -0.895298, \\
0.868991, 1.67417, 1.37254, 0.521282, 2.86879, 0.653835}, \\
\text{LY[w]} \rightarrow \\
\{0.97725, 1.10446, 1.32156, -0.715398, 0.1296, -1.40333, -1.11252, \\
-0.447303, -1.15785, 0.123867, -0.166621, 0.385073, 0.730527, -1.66333, \\
-0.345996, -1.74982, -0.485517, 0.937286, 1.63829, -1.13556, -1.44134, \\
0.500906, -0.999789, 0.228176, -1.1454, 1.73745, -1.60854, -1.06215, \\
-1.07815, -1.31326, 0.11412, 1.28792, -0.709178, 0.844933, 0.291268, \\
-0.776762, 0.71912, -0.94738, -1.66528, 0.00613181, 0.625405, 0.310604, \\
-0.88552, 0.635544, -0.0523398, -0.459871, 0.354249, -0.927752, \\
-0.527289, 0.158014, -0.0303813, 0.128091, -1.58813, 1.36929, 0.799488, \\
-0.615692, -0.853587, -0.0215823, 0.232778, -0.0593525, -0.277299, \\
-1.26071, 1.02008, 1.151, -0.161112, 0.405654, -0.54128, 0.876874, \\
-1.24908, 1.74748, 0.36598, 0.830054, 1.85925, 0.0531251, -0.930298, \\
0.522718, 0.413684, -1.12877, -0.32736, 1.66135, 1.86295, 0.328983, \\
1.534, 0.33729, -0.680876, 0.372099, 0.562604, -1.22691, 1.03807, \\
-0.0531184, -0.680441, -0.59769, 0.895902, -2.43549, -0.0267913, \\
1.11253, 1.37863, 0.798041, 2.59144, 0.580531\}, \\
\text{n} \rightarrow 100,
rho -> 0.692704 } / PLUS -> Plus

... ETC ...

e2num1
e2num2
ee2num1
ee2num2
ve2num1
ve2num2
v2num1
v2num2
ev2num1
ev2num2
vv2num1
vv2num2

(*****************************************************************************)

(*****************************************************************************  r.s2 *****************************************************************************)

(* SPLUS code for producing comparative Monte Carlo simulations for the correlation coefficient for n = 15 law school data. Can hence compare bootstrap mean and standard error for the correlation coefficient obtained using the analytical bootstrap with the same using conventional resampling as done below. *)

origlaw1_c(576,635,558,578,666, 580,555 ,661,651, 605,653, 575 ,545, 572,594)
APPENDIX: Source Code

origlaw2_c(3.39,3.30, 2.81,3.03, 3.44, 3.07, 3.00, 3.43, 3.36, 3.13, 3.12,
2.74,2.76, 2.88, 2.96)

law1_c(-0.600997, 0.860219, -1.04679, -0.551465, 1.62798, -0.501932,
-1.12109, 1.50414, 1.25648, 0.117227, 1.30601, -0.625764, -1.36875,
-0.700063, -0.155203)

law2_c(1.25538, 0.872812, -1.21003, -0.274879, 1.46791, -0.104851, -0.4024,
1.4254, 1.12785, 0.150192, 0.107685, -1.50758, -1.42257, -0.912485,
-0.572429)

options(object.size=40e6)

data_c(law1,law2)
xdata_matrix(data,ncol=2)

n_15
x_1:n
nboot_25600

bootstrap_function(x,nboot,theta,xdata) {
    data_matrix(sample(x,size=length(x)*nboot, replace=T), nrow=nboot)
    return(apply(data,1,theta,xdata))
}

theta_function(x,xdata) { cor(xdata[x,1], xdata[x,2]) }
results_bootstrap(x,nboot,theta,xdata)

bs.se_sqrt(var(results))
varvar_(nboot/((nboot-1)\^2))*var((results - mean(results))\^2)

var.bsse_(1/(4*var(results)))*varvar

se.bsse_var.bsse\^0.5

xbar_mean(results)

se_sqrt(var(results)/nboot)

bs.se

se.bsse

xbar

se

(*************************************************************************)

(************************************************************************* r2.s*************************************************************************)

(* SPLUS code for producing comparative Monte Carlo
   simulations for the correlation coefficient for
   n = 100 simulated standard normal data.
   Can hence compare bootstrap mean and standard error for the
   correlation coefficient obtained using the analytical
   bootstrap with the same using conventional resampling
   as done below.

norm1_c(-0.956714, 0.686892, -0.564691, -1.14333, -0.0559606, -0.874619,
   -0.436588, 1.12566, -0.793298, 0.297573, 0.249147, 0.0330676, 0.22594,
   -1.21993, 0.318041, 0.291848, 0.0120344, 1.62495, 1.49834, -1.8793,
   -1.30035, -1.40383, -1.37385, 0.392384, -1.08785, 1.08452, -0.948535,
-0.750636, -0.528461, -1.03515, 0.299149, -0.0431774, -1.81281,
  0.933136, -0.914287, -0.353526, -0.0696765, -1.11197, 0.239538,
-0.854338, 2.07333, -0.0952752, -0.6538, -0.248606, -0.857544, 0.284146,
  1.71851, -0.875686, -0.935791, 0.0546828, -0.662648, -0.130106,
-2.59892, 1.29063, 0.61013, -0.0920582, -0.665875, -0.394396, -0.0426476,
  0.373098, 0.0048362, -1.41674, 0.926444, 0.353601, -0.203354, 0.320726,
  0.161122, 0.471737, -1.17339, 0.489503, 0.39107, 1.95425, 0.0301679,
  0.511621, -1.41423, 0.425221, -0.558879, -1.2051, 1.18088, 0.762738,
  1.36058, 0.25875, 1.0521, 0.665511, -0.0377158, 0.353328, 0.0265674,
-0.51515, 0.887074, -1.72355, -0.262615, -0.0142491, 2.73824,
-0.895298,
0.868991, 1.67417, 1.37254, 0.521282, 2.86879, 0.653835)

\text{norm2_c}(-0.97725, 1.10446, 1.32156, -0.715398, 0.1296, -1.40333, -1.11252,
-0.447303, -1.15785, 0.123867, -0.166621, 0.385073, 0.730527, -1.66333,
  0.345996, -1.74982, -0.485517, 0.937286, 1.63829, -1.13556, -1.44134,
  0.500906, -0.999789, 0.228176, -1.1454, 1.73745, -1.60854, -1.06215,
  -1.07815, -1.31326, 0.11412, 1.28792, -0.709178, 0.844933, 0.291268,
  -0.776762, 0.71912, -0.94738, -1.66528, 0.00613181, 0.625405, 0.310604,
  -0.88552, 0.635544, -0.0523398, -0.459871, 0.354249, -0.927752,
-0.527289, 0.158014, -0.0303813, 0.128091, -1.58813, 1.36929, 0.799488,
  -0.615692, -0.853587, -0.0215823, 0.232778, -0.0593525, -0.277299,
  1.26071, 1.02008, 1.151, -0.161112, 0.405654, -0.54128, 0.876874,
  1.24908, 1.74748, 0.36598, 0.830054, 1.85925, 0.0531251, -0.930298,
  0.522718, 0.413684, -1.12877, -0.32736, 1.66135, 1.86295, 0.328983,
  1.534, 0.33729, -0.680876, 0.372099, 0.562604, -1.22691, 1.03807,
-0.0531184, -0.680441, -0.59769, 0.895902, -2.43549, -0.0267913,
  1.11253, 1.37863, 0.798041, 2.59144, 0.580531)

\text{options(object.size=50e6)}
data_c(norm1,norm2)
xdata_matrix(data,ncol=2)

n_100
x_1:n
nboot_25600

bootstrap_function(x,nboot,theta,xdata) {
    data_matrix(sample(x,size=length(x)*nboot, replace=T), nrow=nboot)
    return(apply(data,1,theta,xdata))
}
theta_function(x,xdata) { cor(xdata[x,1], xdata[x,2]) }
results_bootstrap(x,nboot,theta,xdata)

bs.se_sqrt(var(results))
varvar_(nboot/((nboot-1)^2))*var((results - mean(results))^2)
var.bsse_(1/(4*var(results)))*varvar
se.bsse_var.bsse^0.5

xbar_mean(results)
se_sqrt(var(results)/nboot)

bs.se
se.bsse
xbar
se

(************************ END ************************)*
A.3 Source Code used for Chapter 4

This section contains the Mathematica programs that were used for the examples in Chapter 4. Splus code for producing comparative Monte Carlo simulations for the Behrens-Fisher statistic is given at the end of this section.

(* This defines the analytical bootstrap-t statistic
for the correlation coefficient
and derives the symbolic bootstrap expectation of
it as well as evaluating it for the law school data *)

$Echo = "stdout"

$RecursionLimit = 4096

<<BSTAT.1.0.new.m
<<astat.3.0.new.m

LX/:VQ[LX[w]]=True
LY/:VQ[LY[w]]=True;

Cum[LX[w]]=0
Cum[LY[w]]=0
LX/:Cum[LX[w],LY[w]]=1
LY/:Cum[LY[w],LY[w]]=1
LX/:Cum[LX[w],LY[w]]=rho

cov/:NT[cov,i_] := NT[cov,i] =
  If[i==0,Cum[LX[w],LY[w]],

(************** t.m ***************)
If[i==1,RS[n][LX[w]LY[w]] -      
        Cum[LX[w]]RS[n][LY[w]] - Cum[LY[w]]RS[n][LX[w]], 
If[i==2,-RS[n][LX[w]]RS[n][LY[w]],0])]

vax/:NT[vax,i_] := 
If[i==0,Cum[LX[w],LX[w]],
If[i==1,RS[n][LX[w]]LX[w]] -
        Cum[LX[w]]RS[n][LX[w]] - Cum[LY[w]]RS[n][LX[w]], 
If[i==2,-RS[n][LX[w]]RS[n][LX[w]],0])]

vay/:NT[vay,i_] :=
If[i==0,Cum[LY[w],LY[w]],
If[i==1,RS[n][LY[w]]LY[w]] -
        Cum[LY[w]]RS[n][LY[w]] - Cum[LY[w]]RS[n][LY[w]], 
If[i==2,-RS[n][LY[w]]RS[n][LY[w]],0])]

cor/:NT[cor,i_] := NT[cor,i] =
        Collect[Expand[NT[Ntimes[Nf[#1^(-1/2)&,Ntimes[vax,vay],cov],i]],RS]

cor2/:NT[cor2,i_] := NT[cor2,i] = NT[cor,i+1]

bootvar/:NT[bootvar,i_] := NT[bootvar,i] =
        NT[Nboot[Npower[cor,2]],i] - NT[Npower[Nboot[cor],2],i]

bootvar2/:NT[bootvar2,i_] := NT[bootvar2,i] =
        NT[Nboot[Npower[cor2,2]],i] - NT[Npower[Nboot[cor2],2],i]

t/:NT[t,i_] := NT[t,i] =
        Expand [
            NT[Ntimes[cor2, Nf[#1^(-1/2)&, bootvar2]], i]
            - NT[Ntimes[r1, Nf[#1^(-1/2)&, bootvar2]], i]
        ]

text = { RS -> Z, LX[w] -> X, LY[w] -> Y }
\texttt{bootexp\_t1 = Expand[Nsum[Nboot[Npower[t,1]],1]] ./ r1 \rightarrow cor2}

\%
./ texr
\TeXForm[\%]
\%
./ texr;
\texttt{Simplify[\%]}
\TeXForm[\%]

\texttt{Simplify[bootexp\_t1] ./ texr ./ n \rightarrow 15 ./ rho \rightarrow 0.776374}

\texttt{bootexp = Expand[Nsum[Nboot[t],1]]}
\%
./ texr;
\texttt{Simplify[\%]}
\TeXForm[\%]

\texttt{bootexp\_t1SUM = Collect[(bootexp\_t1/.RS[n] \rightarrow (0&)/.Cum \rightarrow EmpCum),SUM];
PLUS[{a_\_}] := PLUS[a]}

\texttt{bootexp\_t1SUM ./ {SUM \rightarrow PLUS, LX[w] \rightarrow \{-0.600997, 0.860219, -1.04679, -0.551465
1.62798, -0.501932, -1.12109, 1.50414, 1.25648, 0.117227,
1.30601, -0.625764, -1.36875, -0.700063, -0.155203\},
LY[w] \rightarrow \{1.25538, 0.872812, -1.21003, -0.274879, 1.4679
-0.104851, -0.4024, 1.4254, 1.12785, 0.150192,
0.107685, -1.50758,
-1.42257, -0.912485, -0.572429\},
n \rightarrow 15 ,
 rho \rightarrow 0.776374 \} ./ PLUS \rightarrow Plus}
APPENDIX: Source Code

(* This calculates the symbolic bootstrap expansion of the acceleration adjustment associated with BCa C.I.'s. This is done for the correlation coefficient w.r.t the law school data. *)

$Echo = "stdout"

<<BSTAT.1.0.new.m
<<astat.3.0.new.m

RVQ[X]=True
VQ[X] = True
RVQ[Y]=True
VQ[Y]=True

LX/:VQ[LX[w]]=True
LY/:VQ[LY[w]]=True;
Cum[LX[w]]=0
Cum[LY[w]]=0
LX/:Cum[LX[w],LX[w]]=1
LY/:Cum[LY[w],LY[w]]=1
LX/:Cum[LX[w],LY[w]]=rho

cov/:NT[cov,i_] := NT[cov,i] =
   If[i==0,Cum[LX[w],LY[w]],
   If[i==1,RS[n][LX[w]LY[w]] -
APPENDIX: Source Code

```
Cum[LX[w]]RS[n][LY[w]] - Cum[LY[w]]RS[n][LX[w]],
If[i==2,-RS[n][LX[w]]RS[n][LY[w]],0]]

vax/:NT[vax,i_] :=
   If[i==0,Cum[LX[w],LX[w]],
   If[i==1,RS[n][LX[w]LY[w]] -
   Cum[LX[w]]RS[n][LY[w]] - Cum[LX[w]]LS[n][LY[w]],
   If[i==2,-RS[n][LY[w]]RS[n][LY[w]],0]]

vay/:NT[vay,i_] :=
   If[i==0,Cum[LY[w],LY[w]],
   If[i==1,RS[n][LY[w]LY[w]] -
   Cum[LY[w]]RS[n][LY[w]] - Cum[LY[w]]RS[n][LY[w]],
   If[i==2,-RS[n][LY[w]]RS[n][LY[w]],0]]

cor/:NT[cor,i_] := NT[cor,i] =
   Collect[Expand[NT[NTimes[Nf[1^-1/2],NTimes[vax,vay]],cov],i]],RS]

cor2/:NT[cor2,i_] := NT[cor2,i] = NT[cor,i+1]

(* Bootvar gives the ith term, as opposed to the sum that v2 gives. *)
   We need this the ith term as input to get the ith term of the t statistic *)

bootvar/:NT[bootvar,i_] := NT[bootvar,i] =
   NT[Nboot[NTimes[cor,2]],i] - NT[NTimes[Nboot[cor,2],2],i]

bootvar2/:NT[bootvar2,i_] := NT[bootvar2,i] =
   NT[Nboot[NTimes[cor2,2]],i] - NT[NTimes[Nboot[cor2,2],2],i]

(* The next section calculates tha acceleration constant 'a' which
   also represents a slope! *)

prod/:NT[prod,i_] := NT[prod,i] =
```
APPENDIX: Source Code

```
Expand[NT[NTimes[cor, bootvar], i]]

bprod/: NT[bprod, i_] := NT[bprod, i] =
    Expand[NT[NT[Nboot[prod], i]]]

bprod2/: NT[bprod2, i_] := NT[bprod2, i] = NT[bprod, i+1]

bcor/: NT[bcor, i_] := NT[bcor, i] =
    Expand[NT[NT[Nboot[cor], i]]]

bootvar2/: NT[bootvar2, i_] := NT[bootvar2, i] =
    NT[Nboot[Npower[cor2, 2], i]] - NT[Npower[Nboot[cor2], 2], i]

bbvar2/: NT[bbvar2, i_] := NT[bbvar2, i] =
    Expand[NT[NT[Nboot[bootvar2], i]]]

(* Made up of three pieces: bprod2, bootvar2, bcor. *)

acc3/: NT[acc3, i_] := NT[acc3, i] =
    Expand[NT[NTimes[bprod2, Nf[#1^(-1/2)&, bootvar2]], i]] -
    Expand[NT[NTimes[NTimes[bcor, bootvar2], Nf[#1^(-1/2)&, bootvar2]], i]]

Boot[Acc][i_] := Simplify[Nsum[acc3, i]]

(*************** NUMERICAL EVALUATION FOR n = 15 ***************

$RecursionLimit = 4096
```

slope = Simplify[Nsum[acc3,1]] (* first order slope *)
\%/ . text
TeXForm[%]

slopeSUM = Collect[(slope/.RS[n]->(0&)/.Cum->EmpCum),SUM];

PLUS[{a_}] := PLUS[a]

(* The following is for the law school data, n = 15 *)

eslopenum = slopeSUM /. \{SUM->PLUS, LX[w] -> \{-0.600997, 0.860219, -1.04679, 
-0.551465, 1.62798, 
-0.501932,-1.12109, 1.50414, 1.25648, 0.117227 
1.30601,-0.625764, -1.36875,-0.700063, -0.155203 \}
LY[w] -> \{1.25538, 0.872812, -1.21003, -0.274879, 1.4679 
-0.104851, -0.4024,1.4254, 1.12785, 0.150192, 
0.107685, -1.50758, 
-1.42257, -0.912485, -0.572429\},
n->15,
rho -> 0.776374 \} /. PLUS -> Plus

(*******************************************************************************)

(*******************************************************************************)

(*******************************************************************************)

$Echo = "stdout"
APPENDIX: Source Code

<<BSTAT.indep2.LXonly.m (* Indep. version of BSTAT.1.0.new.m *)

<<astat.3.0.new.m

(* Some Examples *)

RVQ[X]=True
VQ[X] = True
RVQ[Y] = True
VQ[Y] = True

LX/:VQ[LX[w]]=True
LY/:VQ[LY[w]]=True;

BEV[ SUM[X]^2 ]
BEV[ SUM[Y]^2 ]
BEV[ SUM[X]^2 SUM[Y]^2 ]
BEV[ SUM[X] SUM[Y] ]
BEV[ SUM[X]^2 SUM[Y] ]
BEV[ SUM[X]^2 SUM[Y]^2 SUM[Y] ]
BEV[ SUM[X] ]
BEV[ SUM[Y] ]
BEV[ 1 ]

(* end of examples *)

(* BERHENS-FISHER problem : corrected

Assume Ex = 0, Ey = 0 with unequal variances
which now really are just second moments !

Equal zero means as opposed to equal non-zero means will speed up the 4th order symbolic calcs!

*)

Cum[LX[w]] = 0
Cum[LY[w]] = 0

cov/: NT[cov,i_] := NT[cov,i] =
    If[i==0, Cum[LX[w],LY[w]],
    If[i==1, RS[n][LX[w]LY[w]] -
            Cum[LX[w]]RS[n][LY[w]] - Cum[LY[w]]RS[n][LX[w]],
    If[i==2, -RS[n][LX[w]]RS[n][LY[w]], 0]])

vax/: NT[vax,i_] :=
    If[i==0, Cum[LX[w],LX[w]] - Cum[LX[w]]Cum[LX[w]],
    If[i==1, RS[n][LX[w]LX[w]] -
            Cum[LX[w]]RS[n][LX[w]] - Cum[LX[w]]RS[n][LX[w]],
    If[i==2, -RS[n][LX[w]]RS[n][LX[w]], 0]])

vay/: NT[vay,i_] :=
    If[i==0, Cum[LY[w],LY[w]] - Cum[LY[w]]Cum[LY[w]],
    If[i==1, RS[n][LY[w]LY[w]] -
            Cum[LY[w]]RS[n][LY[w]] - Cum[LY[w]]RS[n][LY[w]],
    If[i==2, -RS[n][LY[w]]RS[n][LY[w]], 0]])

cor/: NT[cor,i_] := NT[cor,i] =
    Collect[Expand[NT[Ntimes[Nf[#1^(-1/2)&, Ntimes[vax,vay]], cov], i]], RS]

$RecursionLimit = 4096

(* BERHENS-FISHER CODE : *)
APPENDIX: Source Code

diff/:NT[diff,i_] := NT[diff,i] =
   If[i==0, Cum[LX[w]]-Cum[LY[w]] ,
   If[i==1, RS[n][LX[w]] - RS[n][LY[w]] , 0 ]]

bf/:NT[bf,i_] := NT[bf,i] =
   Collect[Expand[NT[Ntimes[Nf[#1^-(-1/2)&, Apply[Plus, {vax,vay} ]],
   diff ] ,i]],RS]

bf1/:NT[bf1,i_] := NT[bf1,i] = NT[bf,i+1]
bf2/:NT[bf2,i_] := NT[bf2,i] = NT[bf,i+2]

(* BF *)

BF[i_] := Nsum[bf, i]

NT[bf,2]
Nsum[bf,2]
NT[Nboot[bf],0]
NT[Nboot[bf],1]
NT[Nboot[bf],2]

bootexp = Expand[Nsum[Nboot[bf],4]]
bootvar = Expand[Nsum[Nboot[Npower[bf,2]] - Npower[Nboot[bf],2],4]]

(* Provides 2nd order symbolic and numerical results for the Behrens-Fisher statistic using n = 44 *)
survival data. 4th order uses similar code. *)

$Echo = "stdout"

<<bf.newtest2.m

(* Uses SURVIVAL data ( n = 44 ) *)

(********** Assuming 0 , 0 , unequal variances! *********)
(* and assuming INDEPENDENCE code in bf3.m *)
(* i.e. BEV tool was MODIFIED *)

$RecursionLimit = 4096

Cum[LX[w], LY[w]] = 0
 Cum[LX[w], LY[w], LY[w]] = 0
 Cum[LX[w], LX[w], LY[w]] = 0
 Cum[LX[w], LY[w], LX[w]] = 0
 Cum[LY[w], LX[w], LY[w]] = 0
 Cum[LX[w], LX[w], LY[w], LY[w]] = 0
 Cum[LX[w], LX[w], LX[w], LY[w]] = 0
 Cum[LX[w], LY[w], LY[w], LY[w]] = 0
 Cum[LX[w], LX[w], LX[w], LX[w], LY[w]] = 0
 Cum[LX[w], LX[w], LX[w], LY[w], LY[w]] = 0
 Cum[LX[w], LX[w], LY[w], LY[w], LY[w]] = 0
 Cum[LX[w], LY[w], LY[w], LY[w], LY[w]] = 0

\texttt{texr = \{ RS -> Z, LX[w] -> X, LY[w] -> Y \}}

(* Expansion of Behrens-Fisher statistic *)
\textbf{APPENDIX: Source Code}

\begin{verbatim}
bf2 = Nsum[bf,2] \\
\% / . texr

(* Expansion of bootstrap mean of the Behrens-Fisher statistic *)

e2 = Expand[Nsum[Nboot[bf],2]] \\
\% / . texr
Simplify[%] \\
TeXForm[%] \\

(* Expansion of E - over x - of bootstrap mean of the B-F stat. *)

ee2 = Expand[Nsum[Ncum[Nboot[bf],1],2]] \\
ee2 / . texr \\
Simplify[%] \\
TeXForm[%] \\

ee4 = Expand[Nsum[Ncum[Nboot[bf],1],4]] \\
ee4 / . texr \\
Simplify[%] \\
TeXForm[%] \\

(* Expansion of Var - over x - of bootstrap mean of the B-F stat. *)

ve2 = Expand[Nsum[Ncum[Nboot[bf],2],2]] \\
ve2 / . texr \\
Simplify[%] \\
TeXForm[%]
\end{verbatim}
APPENDIX: Source Code

(* ------------------ ------------------------------*)

(* Expansion of bootstrap variance of the B-F stat. *)

v2 = Expand[Nsum[Nboot[Npower[bf, 2]] - Npower[Nboot[bf], 2], 2]]

v2 /. texr
Simplify[%]
TeXForm[%]

(* Expansion of E - over x - of bootstrap variance of the B-F stat. *)

ev2 = Expand[Nsum[Ncum[Nboot[Npower[bf, 2]] - Npower[Nboot[bf], 2]], 1, 2]]
ev2 /. texr
Simplify[%]
TeXForm[%]

(* Expansion of Var - over x - of bootstrap variance of the B-F stat. *)

vv2 = Expand[Nsum[Ncum[Nboot[Npower[bf, 2]] - Npower[Nboot[bf], 2], 2], 2]]

vv2 /. texr
Simplify[%]
TeXForm[%]

vv2one = Expand[Nsum[Ncum[Nboot[Npower[bf1, 2]] - Npower[Nboot[bf1], 2], 2], 2]]

vv2one /. texr
Simplify[%]
TeXForm[%]

exp2 = Expand[Nsum[Ncum[bf, 1], 2]]

exp2 /. texr
\textbf{APPENDIX: Source Code}

\begin{verbatim}
Simplify[%]
TeXForm[%]

var2 = Expand[Nsum[Ncum[bf, 2], 2]]
var2 /. texr
Simplify[%]
TeXForm[%]

(* exp4 = Expand[Nsum[Ncum[bf, 1], 4]] /.
   texr *)
(* var4 = Expand[Nsum[Ncum[bf, 2], 4]] /.
   texr *)

bf2SUM = Collect[(bf2/.RS[n]->(0&)/.Cum->EmpCum),SUM];
e2SUM = Collect[(e2/.RS[n]->(0&)/.Cum->EmpCum),SUM];
ee2SUM = Collect[(ee2/.RS[n]->(0&)/.Cum->EmpCum),SUM];
e4SUM = Collect[(ee4/.RS[n]->(0&)/.Cum->EmpCum),SUM];
ev2SUM = Collect[(ve2/.RS[n]->(0&)/.Cum->EmpCum),SUM];
v2SUM = Collect[(v2/.RS[n]->(0&)/.Cum->EmpCum),SUM];
ev2SUM = Collect[(ev2/.RS[n]->(0&)/.Cum->EmpCum),SUM];
v2SUM = Collect[(vv2/.RS[n]->(0&)/.Cum->EmpCum),SUM];
vv2SUM1 = Collect[(vv2one/.RS[n]->(0&)/.Cum->EmpCum),SUM];
exp2SUM = Collect[(exp2/.RS[n]->(0&)/.Cum->EmpCum),SUM];
var2SUM = Collect[(var2/.RS[n]->(0&)/.Cum->EmpCum),SUM];

PLUS[{a_}] := PLUS[a]

(* The following uses the \texttt{SURVIVAL} data *)

(*************** assuming 0 means and unequal variances ***************

bf2num1 = bf2SUM /. \{SUM->PLUS, LX[w] ->
\end{verbatim}
APPENDIX: Source Code

{ 
-466.7272727, -441.7272727, -439.7272727, -435.7272727, -423.7272727, 
-411.7272727, -409.7272727, -388.7272727, -380.7272727, -375.7272727, 
-361.7272727, -339.7272727, -316.7272727, -313.7272727, -300.7272727, 
-298.7272727, -290.7272727, -288.7272727, -286.7272727, -275.7272727, 
-249.7272727, -248.7272727, -229.7272727, -176.7272727, -168.7272727, 
-82.7272727, -38.7272727, -19.7272727, 0.27272727, 0.27272727, 
85.27272727, 93.27272727, 96.27272727, 311.2727273, 
371.2727273, 690.2727273, 730.2727273, 748.2727273, 882.2727273, 
971.2727273, 1101.272727, 1138.272727, 1142.272727 }

LY[w] ->

{ 
-644.9318182, -582.9318182, -540.9318182, -520.9318182, -463.9318182, 
-429.9318182, -395.9318182, -383.9318182, -344.9318182, -344.9318182, 
-303.9318182, -291.9318182, -289.9318182, -287.9318182, 
-265.9318182, -262.9318182, -262.9318182, -257.9318182, -251.9318182, 
-237.9318182, -185.9318182, -156.9318182, -122.9318182, -121.9318182, 
-110.9318182, -83.93181818, -76.93181818, 29.06818182, 30.06818182, 
102.0681818, 132.0681818, 140.0681818, 151.0681818, 309.0681818, 
322.0681818, 331.0681818, 599.0681818, 625.0681818, 774.0681818, 
814.0681818, 870.0681818, 905.0681818, 1044.068182, 
1048.068182 }

n -> 44 } / PLUS -> Plus

e2num1 = e2SUM /. {SUM->PLUS, LX[w] ->

{ 
-466.7272727, -441.7272727, -439.7272727, -435.7272727, -423.7272727, 

\[-411.7272727, -409.7272727, -388.7272727, -380.7272727, -375.7272727, \\
-361.7272727, -339.7272727, -316.7272727, -313.7272727, -300.7272727, \\
-298.7272727, -290.7272727, -288.7272727, -286.7272727, -275.7272727, \\
-249.7272727, -248.7272727, -229.7272727, -176.7272727, -168.7272727, \\
-82.7272727, -38.7272727, -19.7272727, 0.27272727, 44.27272727, \\
58.27272727, 83.27272727, 93.27272727, 96.27272727, 311.2727273, \\
371.2727273, 690.2727273, 730.2727273, 748.2727273, 882.2727273, \\
971.2727273, 1101.272727, 1138.272727, 1142.272727 \\
},

LY[w] ->
{
-644.9318182, -582.9318182, -540.9318182, -520.9318182, -463.9318182, \\
-429.9318182, -395.9318182, -383.9318182, -344.9318182, -344.9318182, \\
-303.9318182, -291.9318182, -289.9318182, -287.9318182, \\
-265.9318182, -262.9318182, -262.9318182, -257.9318182, -251.9318182, \\
-237.9318182, -185.9318182, -156.9318182, -122.9318182, -121.9318182, \\
-110.9318182, -83.93181818, -76.93181818, 29.06818182, 30.06818182, \\
102.0681818, 132.0681818, 140.0681818, 151.0681818, 151.0681818, \\
309.0681818, 322.0681818, 331.0681818, 599.0681818, 625.0681818, 774.0681818, \\
814.0681818, 870.0681818, 905.0681818, 1044.0681818, \\
1048.068182
},

n -> 44 } /.

PLUS -> Plus

... ETC. ...

bf2num1 //N
e2num1  //N

% Sqrt[44]  //N
APPENDIX: Source Code

```plaintext
ee2num1 //N
% Sqrt[44] //N
e4num1 //N
% Sqrt[44] //N
t2num1 //N
% 44 //N
Sqrt[%] //N
t2num1 //N
var = % 44 //N
se = Sqrt[% ] //N
% Sqrt[44/43] //N (* to compare to Splus *)
t2num1 //N
% 44 //N
st2num1 //N
st2num2 //N
% 44^2 //N
exp2num //N
% Sqrt[44] //N
var2num //N
% 44 //N

(******************************************************************************)

(*************** BSTAT.indep2.LXonly.m *****************)

(* Same as BSTAT.1.0.new.m except for extra code below included
   to create the INDEPENDENCE criterion necessary for
   the Behrens-Fisher statistic *)

BEV::usage =
```
"BEV[a] returns the bootstrap expectation value of a"

(*
: [font = output; output; inactive]
"BEV[a] returns the bootstrap expectation value of a"
; [o]
BEV[a] returns the bootstrap expectation value of a
: [font = input; initialization]
*)
ClearAll[BEV]
BEV[a_ + b_] := BEV[a] + BEV[b]
BEV[(a_ + b_)^c_. d_.] := BEV[Expand[(a+b)^c d]]/;
   ((c > 0) && (IntegerQ[c]))
BEV[a_ b_] := a BEV[b]; !RVQ[a]
BEV[a_] := a /; !RVQ[a]
BEV[SUM[a_]] := SUM[a]
BEV[SUM[a_] c_] := BEV[c, a]
BEV[SUM[a_] c_, b_] := BEV[c, a, b]
BEV[SUM[a_]^b_ c_] := BEV[SUM[a]^(b-1)c, a]
BEV[SUM[a_]^b_ c_, d_] := BEV[SUM[a]^(b-1)c, a, d]

(* Setting VARIABLES ! *)

RVQ[X]=True
VQ[X] = True
RVQ[Y]=True
VQ[Y]=True

LX/:VQ[LX[w]]=True
LY/:VQ[LY[w]]=True
LX/:RVQ[LX[w]]=True
APPENDIX: Source Code

LY/ : RVQ[LY[w]] = True
VQ[ZM[LX[w]]] = True
RVQ[ZM[LX[w]]] = True
VQ[ZM[LY[w]]] = True
RVQ[ZM[LY[w]]] = True

(* Examples *)

BEV[ SUM[LX[w]] ]
BEV[ SUM[LX[w]]^2 ]
BEV[ SUM[LX[w]]^2 SUM[LY[w]]^2 ]
BEV[ SUM[LX[w]] SUM[LY[w]] ]
(* end of examples *)

(* THIS PART CREATES INDEPENDENCE *)

h[a_] := Prepend[a, 1]
g[a_] := { h [ Cases [ a, ZM[LX[w]]^c_ ] ] , h [ Cases [ a, ZM[LY[w]]^d_ ] ]
m[a_] := Apply[Times, Apply[BEV1, a, {1}]]
BEV1[a_] := a /; !RVQ[a]
BEV[1,a_] := Collect[Expand [ m [ g [ {a} ] ] ] ], SUM]

BEV[1,a_] := Collect[Expand[Map[BEVPE, FP1[{a}] + PE[0] ]], SUM]

(* Examples *)

BEV[ SUM[LX[w]] ]
BEV[ SUM[LX[w]]^2 ]
BEV[ SUM[LX[w]]^2 SUM[LY[w]]^2 ]
APPENDIX: Source Code

BEV[ SUM[LX[w]] SUM[LY[w]] ]
(* end of examples *)

(******************************************************************************)

(************* bfSurv2.s : Splus code **********************)

# Behrens-Fisher statistic : Monte Carlo calcs.

# INDEP. SAMPLES APPROACH (unlike the corr. coeff. with paired samples!)

# Survival times (days) of gastric cancer patients (n = 44)

# Data have been standardized to have ZERO MEANS only
# i.e. assume H0: equal means = 0

options(object.size=10e6)

c5_c(-466.7272727 , -441.7272727 , -439.7272727 , -435.7272727 , -423.7272727 ,
     -411.7272727 , -409.7272727 , -388.7272727 , -380.7272727 , -375.7272727 ,
     -361.7272727 , -339.7272727 , -316.7272727 , -313.7272727 , -300.7272727 ,
     -298.7272727 , -290.7272727 , -288.7272727 , -286.7272727 , -275.7272727 ,
     -249.7272727 , -248.7272727 , -229.7272727 , -176.7272727 , -168.7272727 ,
     -82.7272727 , -38.7272727 , -19.7272727 , 0.27272727 , 44.27272727 ,
     58.27272727 , 83.27272727 , 93.27272727 , 96.27272727 , 311.27272727 ,
     371.27272727 , 690.27272727 , 730.27272727 , 748.27272727 , 882.27272727 ,
     971.27272727 , 1101.27272727 , 1138.27272727 , 1142.27272727 )

c6_c(-644.9318182 , -582.9318182 , -540.9318182 , -520.9318182 , -463.9318182 ,
n_44
x_1:n
nboot_12800   # Repeat for B = : 50,100,200,400,800,1600,3200,6400,
              # 12,800 , 25600

data1_matrix(sample(c5,size=length(x)*nboot, replace=T), nrow=nboot)
data2_matrix(sample(c6,size=length(x)*nboot, replace=T), nrow=nboot)

Bmeans1_apply(data1,1,mean)
Bvars1_apply(data1,1,var)/n
Bmeans2_apply(data2,1,mean)
Bvars2_apply(data2,1,var)/n

bf_( Bmeans1 - Bmeans2 ) / sqrt( Bvars1 + Bvars2)
# B bootstrap reps of BF statistic
# as on p. 224 Efron&Tibs., step (16.6)

bf_mean_mean(bf) # bootstrap mean (exp.) of bf
se_mean_sqrt(var(bf)/nboot)
APPENDIX: Source Code

bf.se_sqrt(var(bf))  # bootstrap SE of bf
varvar_((nboot/((nboot-1)^2))*var((bf- mean(bf))^2)
var.bsse_(1/(4*var(bf)))*varvar
se.se_var.bsse^0.5

# 1. See if matches with symbolic results, using modified
# BSTAT code assuming INDEP’CE and ZERO means
# 2. Calc. SE’s in brackets later using code below
# 3. Run with B as now preferred (even spacings and up to 25,600)

bf.mean
se.mean
bf.se
se.se

(************************ END ***************************
Bibliography


