Title

Computer algebra and algorithms for unbiased moment estimation of arbitrary order

Journal

Cogent Mathematics & Statistics, 6(1)

ISSN

2574-2558

Authors

Gerlovina, Inna

Hubbard, Alan E

Publication Date

2019

DOI

10.1080/25742558.2019.1701917

Peer reviewed

HHS Public Access

Author manuscript

Cogent Math Stat. Author manuscript; available in PMC 2020 October 09.

Published in final edited form as:

Cogent Math Stat. 2019 ; 6(1): . doi:10.1080/25742558.2019.1701917.

Computer Algebra and Algorithms for Unbiased Moment Estimation of Arbitrary Order

Inna Gerlovina[a,b], **Alan E. Hubbard**[a]

[a]University of California, Berkeley, Division of Epidemiology and Biostatistics, Berkeley, CA 94720, USA

[b]University of Caifornia, San Francisco, Department of Medicine, 1001 Potrero Ave, San Francisco, CA 94110

Abstract

While unbiased central moment estimators of lower orders (such as a sample variance) are easily obtainable and often used in practice, derivation of unbiased estimators of higher orders might be more challenging due to long math and tricky combinatorics. Moreover, higher orders necessitate calculation of estimators of powers and products that also amount to these orders. We develop a software algorithm that allows the user to obtain unbiased estimators of an arbitrary order and provide results up to the 6th order, including powers and products of lower orders. The method also extends to finding pooled estimates of higher central moments of several different populations (*e.g.* for two-sample tests). We introduce an R package *Umoments* that calculates one- and two-sample estimates and generates intermediate results used to obtain these estimators.

Keywords

Combinatorics; empirical moments; higher-order approximations; pooled estimates

1. Introduction

Most data analysis methods rely on estimating unknown quantities such as characteristics of an underlying distribution or an effect of a treatment. From a variety of possible estimators of an unknown true parameter, the ones that are typically chosen have certain desirable properties - *e.g.* consistency, efficiency, or unbiasedness. When the sample size is moderate or small, finite sample behavior of an estimator - such as bias, variability, and mean squared error (MSE) - is particularly relevant and is therefore often given special consideration. In addition, when estimation is conducted across multiple samples or studies (pooled estimators), bias may become an important issue.

Moments of a distribution are the most basic building blocks of statistical analysis and their estimates are present in some form in virtually any practical application. A sample average is an estimate of the mean (first moment). Estimates of the variance (second central moment)

CONTACT Inna Gerlovina. innager@berkeley.edu.

are routinely used in statistical inference; they are present in all studentized statistics, of which the most common example is an ordinary *t*-statistic. Higher moments and their estimates, while not as widely used, can be important in various statistical applications and inferential procedures; they also comprise cumulants and their scaled versions (skewness, kurtosis). They are often used in signal processing, financial modeling, and many other areas (for a list of applications see for example [1]). Methods that employ higher order statistics might utilize data in a more efficient way and offer greater insight into the distribution of interest thus providing additional refinement in inference, for example through the use of higher-order approximations to the distribution of a test statistic - such as empirical Edgeworth expansions [2-4]. These methods would require higher-order moment estimation and warrant considerations about estimators' finite sample properties; since moderate or small sample analysis would benefit from such higher-order approaches, unbiased estimates could prove particularly useful.

Naïve estimators $m_k = n^{-1}\sum_{i=1}^{n}(X_i - \bar{X})^k$, $k = 2, 3, \ldots$ of central moments μ_k are biased - that is, $E(m_k) \neq \mu_k$. The first unbiased estimator was introduced for variance by Friedrich Bessel; it is obtained by multiplying m_2 by a factor $n/(n-1)$, thus often called Bessel's correction. That estimator is a part of an ordinary *t*-statistic and therefore plays a role in Student's *t*-distribution; it corresponds to the degrees of freedom in chi-squared distribution that arises from $\sum_{i=1}^{n}(X_i - \bar{X})^2 \sim \chi^2_{n-1}$ when X is a standard normal random variable. The corresponding standard deviation estimator, however, is still biased (though the bias is reduced) and underestimates the true parameter. Interest to unbiased moment and cumulant estimation has a long history, which led to theoretical advances and various strategies to be able to obtain higher-order estimators. The work of R.A. Fisher (1928) [5] provided basis for much of this research, particularly on cumulants; for central moments, unbiased estimators up to fourth order (or "weight" in some literature) have been published by Harald Cramér in 1946 [6]; the results were later expanded for more complex settings (*e.g.* including weights [7]).

Whereas derivation of unbiased moment estimators in general is straightforward, higher order calculations involve long algebra and require obtaining nontrivial coefficients; brute-force calculations of these coefficients become unfeasible fairly quickly. Having observations from different populations or categories, requiring pooled estimators, compounds the problem. Unlike second and third central moments, where naïve biased moment estimators differ from unbiased ones by a constant factor that does not depend on data, subsequent orders require calculation of combinations (integer powers and products) of lower moments that amount to the same order, which in turn creates systems of equations to be solved. With computer algebra, manipulating long algebraic expressions and solving reasonably large systems of linear equations is no longer an issue; the challenge can then be condensed to finding an expectation of the form $E(\overline{X}^{j_1}\overline{X^2}^{j_2}\overline{X^3}^{j_3}\cdots)$, where $\overline{X^j} = n^{-1}\sum_{i=1}^{n} X_i^j$, of an arbitrary order and length, written in terms of sample size and true central moments of the distribution of X. Thus a software algorithm that solves this problem and computer algebra can provide the machinery needed to obtain one-sample and multi-sample pooled estimates of any order, limited only by available processing power.

General order solutions for many problems formulated in the course of unbiased estimation history, including cumulant and moment estimation, are provided in *mathStatica* [8], an add-on package for the proprietory computer algebra system *Mathematica*. Still, given many potential uses for such estimates, there is a need for open-source software and easy to use tools, accessible to a wide range of researchers, that could be seamlessly incorporated into data analysis. Multi-sample pooled estimation, which has not received much attention in higher-order statistics pursuit (and is not included in *mathStatica*), can have many practical applications, especially in two-sample settings (*e.g.* comparing treatment and control groups). In addition, open access to the code and algorithms that are used in generating arbitrary order estimates can be used for obtaining other statistical results, *e.g.* Edgeworth expansions. We introduce an R package *Umoments* [9], which provides pre-programmed functions that calculate one- and two-sample estimates up to sixth order, either from data or from naïve biased estimates, as well as algorithms and tools for generating general order estimators.

In this paper, we break down the procedure of obtaining unbiased moment estimators of an arbitrary order, as well as estimators of products and powers of moments (also referred to as generalized *h*-statistics [10]) such as $\mu_i^k \mu_j^l \cdots$; an analogous procedure is provided for multi-sample pooled estimators. Additionally, this direct approach is illustrated in the *Sage* and *Jupyter* https://github.com/innager/unbiasedMoments templates. Next we describe the algorithm that generates an expression for expectation of raw (non-central) sample moments and their powers and products, thus automating the challenging part of the derivation. Results section provides a full set of one-sample unbiased estimators up to sixth order (or "weight" in some literature); two-sample pooled estimators can be found in *Umoments* package but orders four and higher are too long to include in the paper. Results are followed by a quick overview of *Umoments* package functions; we conclude with a discussion about practical applications of these estimators.

2. Procedure in General

2.1. One-sample estimates

For simplicity, we can consider a mean-zero random variable without any loss of generality. Let $X_1,\ldots,X_n$ be a sample of independent identically distributed random variables with $E(X) = 0$ and central moments μ_k (mean $\mu_1 = 0$, variance μ_2), in this case equal to raw moments; $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$. We also adopt the following useful notation:

$$\overline{X^j} = \frac{1}{n}\sum_{i=1}^{n} X_i^j$$

$m_k = \frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X})^k$ - naïve biased central moment estimators

$M(\cdot)$ - unbiased estimator of an expression inside the parentheses (for quantities such as central moments and their powers and products), e.g. $E\left[M(\mu_3^2)\right] = \mu_3^2$.

The steps to obtain unbiased estimators of a general order are straightforward:

(1) for a desired order, list all the moment combinations for that order (example provided below);

(2) expand their naïve biased estimators (remove the brackets);

(3) take expectations and represent the results in terms of moments μ_k and sample size n; this will produce an equation or a system of equations;

(4) solve this equation or a system of equations for true moments.

As an illustration, we go through these steps for $M(\mu_3)$, an unbiased estimator of a third central moment:

$$\begin{aligned} m_3 &= \frac{1}{n}\sum_{i=1}^{n}(X_i-\bar{X})^3 = \frac{1}{n}\sum_{i=1}^{n}X_i^3 - \frac{3}{n}\sum_{i=1}^{n}X_i^2\bar{X} + \frac{3}{n}\sum_{i=1}^{n}X_i\bar{X}^2 - \frac{1}{n}\sum_{i=1}^{n}\bar{X}^3 \\ E(m_3) &= E(X^3) - 3E(\bar{X}\overline{X^2}) + 3E(\bar{X}^3) - E(\bar{X}^3) = \mu_3 - \frac{3}{n}\mu_3 + \frac{2}{n^2}\mu_3 \\ \mu_3 &= \frac{n^2}{(n-1)(n-2)}E(m_3) \\ M(\mu_3) &= \frac{n^2}{(n-1)(n-2)}m_3 \end{aligned} \tag{1}$$

Steps 2 and 4 are trivial and are performed using computer algebra. Calculation of any unbiased moment estimator of a given order involves all the combinations (powers and products of moments) of that order, which means that for fourth and higher orders there will be a system of equations rather than a single equation (recall that $\mu_1 = 0$). For example, estimators of seventh order will include $M(\mu_7)$, $M(\mu_2\mu_5)$, $M(\mu_2^2\mu_3)$ and $M(\mu_3\mu_4)$ (step 1); step 2 will correspondingly expand m_7, m_2m_5, $m_2^2m_3$ and m_3m_4 producing four equations. Since the equations need to be solved for a given order's combinations of true moments, not individual moments, and all the equations in the system are linear in that order, it makes sense to treat these combinations as single variables, thus solving a system of linear equations.

Step 3 is more challenging but the problem can be reduced to finding an expression for $E\left(\overline{X}^{j_1}\overline{X^2}^{j_2}\overline{X^3}^{j_3}\cdots\right)$ since any term from the right hand side of step 2 equations can be written in that form - e.g. $\frac{1}{n^5}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n}\sum_{m=1}^{n}X_iX_j^4X_k^2X_l^2X_m\bar{X}^3 = \bar{X}^5\overline{X^2}^2\overline{X^4}$. A general solution to this problem is provided in *Umoments* package [9] that generates expressions for these expectations using combinatorics. This algorithm is explained in detail in section 3.

2.2. Pooled estimates

A simple extension of the method can be used to obtain unbiased estimators of central moments for samples that contain observations from several populations or categories. We demonstrate the procedure on a two-category estimation, which extends trivially to any number of categories.

For a sample $X_1,\ldots,X_{n_x}$, $Y_1,\ldots,Y_{n_y}$, $X \perp Y$, let

$$\begin{aligned} m_{xk} &= \frac{1}{n_x}\sum_{i=1}^{n_x}(X_i-\bar{X})^k, \\ m_{yk} &= \frac{1}{n_y}\sum_{i=1}^{n_y}(Y_i-\bar{Y})^k, \text{ and} \\ m_k &= \frac{\sum_{i=1}^{n_x}(X_i-\bar{X})^k+\sum_{i=1}^{n_y}(Y_i-\bar{Y})^k}{n_x+n_y} = \frac{n_x m_{xk}+n_y m_{yk}}{n_x+n_y}, \end{aligned} \tag{2}$$

where m_k is a two-sample analog of the naïve biased estimator described previously. Note that pooled estimation implies an assumption of equality of estimated central moments between distributions of X and Y : $\mu_{xk} = \mu_{yk} = \mu_k$, $k = 2,3,\ldots$. Using this assumption, independence of X and Y, and one-sample results from step 3 in section 2.1, we extend step 3 of the roadmap to incorporate two variables and obtain expectations.

Example: obtain two-sample pooled estimate of the third central moment. Using one-sample result (1), get

$$\begin{aligned} E(m_3) &= \frac{n_x E(m_{x3})+n_y E(m_{y3})}{n_x+n_y} = \frac{n_x^2 n_y+n_x n_y^2-6n_x n_y+2n_x+2n_y}{n_x n_y(n_x+n_y)}\mu_3 \\ &= \left[1-\frac{2(3n_x n_y-n_x-n_y)}{n_x n_y(n_x+n_y)}\right]\mu_3, \end{aligned}$$

which yields

$$M(\mu_3) = \frac{n_x n_y(n_x+n_y)}{n_x^2 n_y+n_x n_y^2-6n_x n_y+2n_x+2n_y}m_3\,. \tag{3}$$

For this particular example, the result matches one-sample case if $n_x = n_y$. That is not true in general, however. All the higher orders involve powers and products of lower moments that need to be expanded before taking expectations, affecting the systems of equations. For example,

$$E\left(m_2^2\right) = E\left[\left(\frac{n_x m_{x2}+n_y m_{y2}}{n_x+n_y}\right)^2\right] = \frac{n_x^2 E(m_{x2}^2)+2n_x n_y E(m_{x2})E(m_{y2})+n_y^2 E(m_{y2}^2)}{(n_x+n_y)^2}\,.$$

3. Generating Expressions for Expectations

To derive general order expectations of naïve moment estimators and their powers and products, one needs to find expectations $E\left(\overline{X}^{j_1}\overline{X^2}^{j_2}\overline{X^3}^{j_3}\cdots\right)$. To build up to this, we first describe the procedure for generating $E\left(\overline{X}^k\right)$, which easily extends to $E\left(\overline{X}^k\overline{X^2}^l\right)$ and then to the general case that involves an arbitrarily long product of $\overline{X^i}^{j_i}$.

3.1. Generate $E\left(\overline{X}^k\right)$

$$E(\overline{X}^k) = \frac{1}{n^k}\sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\cdots\sum_{i_k=1}^{n}E\left(X_{i_1}X_{i_2}\cdots X_{i_k}\right) \tag{4}$$

To find (4), we need to consider all the different combinations of ordered indices $i_1, i_2,\ldots,i_k$; $i_j = 1,\ldots,n$ for each j. There are n^k such combinations but many combinations yield the same $E(X_{i_1}\cdots X_{i_k})$ - for example, $E(X_2X_2X_2X_2X_2X_5X_5X_1X_1) = E(X_4X_3X_4X_6X_6X_3X_6X_6X_6) = \mu_2^2\mu_5$. Combinations that produce the same expectation form a set that we will call a *grouping* (similar to "partitions" and "augmented symmetric functions" in some terminology), and the problem therefore reduces to considering all the groupings (each producing a distinct expectation) and calculating their coefficients, which are the number of combinations in each set. Each product $X_{i_1}\cdots X_{i_k}$ can be broken into smaller products, or *groups*, of X's with the same indices such as $\{X_{i_j}: i_j = c\}$, $c = 1,\ldots,n$. The number of groups ranges between 1 (when all the indices are the same: $i_1 = i_2 = \ldots = i_k$) and k (when all the indices are different: $i_1 \neq i_2 \neq \cdots \neq i_k$); sizes of these groups determine $E(X_{i_1}\cdots X_{i_k})$. Thus each grouping is fully characterized by the number of groups and the group sizes.

Let d denote the number of groups in one grouping G and $a_1,\ldots,a_d$ - the numbers of X's in each group, $\sum_{u=1}^{d} a_u = k$; set of group sizes is unordered, so assigning indices to a's is arbitrary (e.g. decreasing). In the example above: $k = 9$, $d = 3$, $a_1 = 5$, $a_2 = 2$, and $a_3 = 1$. If $\sum_{u=1}^{d} I(a_u = 1) > 0$ (at least one group is of size 1), $E(X_{i_1}\cdots X_{i_k}) = 0$ since $E(X) = 0$ and there is no need to calculate a coefficient for this grouping, which is important in terms of computational efficiency; otherwise $E(X_{i_1}\cdots X_{i_k}) = \prod_{u=1}^{d}\mu_{a_u}$. Adding a subscript g to indicate a grouping G, we get

$$E\left(\overline{X}^k\right) = \sum_{all\,g} C_g \prod_{u=1}^{d_g}\mu_{a_{g,u}}, \tag{5}$$

where C_g is the coefficient for G, i.e. the number of combinations that yield $\{a_{g,u}\}$.

$$C_g = (n)_d \frac{k!}{a_{g,1}!a_{g,2}!\cdots a_{g,d}!s_{g,1}!s_{g,2}!\cdots}, \quad (6)$$

where $(n)_d = n(n-1)\cdots(n-d+1)$ and s_g's are the numbers of the same sized groups if there are any - e.g. for group sizes $a_1 = a_2 = 5$, $a_3 = a_4 = 4$, and $a_5 = a_6 = a_7 = 2$, we will get $s_1 = 2$, $s_2 = 2$, and $s_3 = 3$ (from these we can gather that $k = 24$, $d = 7$, and $E(X_1\cdots X_k) = \mu_5^2\mu_4^2\mu_2^3$). In this particular setting, $C_g/(n)_d$ is analogous to the partition coefficient described in the literature [5, 11]. Going back to our original example (group sizes {5, 2, 2}) - there is only one s_g: $s_{g,1} = 2$; the coefficient for that example is $C_g = n(n-1)(n-2)\frac{9!}{5!2!2!2!}$.

One way of arriving at the expression for C_g could be the following: there are $\frac{(n)_d}{s_{g,1}!s_{g,2}!\cdots}$ ways to pick (unordered) indices that satisfy given group sizes (set $\{a_{g,u}\}$) and $\frac{k!}{a_{g,1}!a_{g,2}!\cdots a_{g,d}!} = \binom{k}{a_{g,1}}\binom{k-a_{g,1}}{a_{g,2}}\cdots\binom{a_{g,d-1}+a_{g,d}}{a_{g,d-1}}$ ways to place these indices on k positions (a multinomial coefficient).

Our software generates expressions for $E\left(\bar{X}^k\right)$ for a given k using the method described above. To find all the possible groupings, we impose an ordering on them and use it to generate each consecutive grouping when the previous one is given, thus moving through a complete set of groupings from to $\{a_1 = k\}$ to $\{a_1 = a_2 = \ldots = a_k = 1\}$. For example, in an agglomerative order, a grouping {5, 2, 2} is preceded by {5, 2, 1, 1} and followed by {5, 3, 1}.

The smallest number of groups is $d = 1$, which produces an order of $\frac{n}{n^k} = \frac{1}{n^{k-1}}$ (the highest order in the range); the largest d with a non-zero contribution to $E(\bar{X}^k)$ is $\left\lfloor \frac{k}{2} \right\rfloor$ (when the indices of X appear in pairs and there are no unpaired indices; when K is odd, one of the groups is of size 3), and the order it produces is $\frac{1}{n^{\left\lceil \frac{k}{2} \right\rceil}}$.

3.2. Generate $E\left(\bar{X}^k\overline{X^2}^l\right)$

$$E\left(\bar{X}^k\overline{X^2}^l\right) = \frac{1}{n^{k+l}}\sum_{i_1=1}^{n}\cdots\sum_{i_k=1}^{n}\sum_{j_1=1}^{n}\cdots\sum_{j_l=1}^{n}E(X_{i_1}\cdots X_{i_k}X_{j_1}^2\cdots X_{j_l}^2) \quad (7)$$

To generate expressions for (7), we extend the algorithm described in 3.1 for equation (4). Now groups consist of X^*s and X^*s with the same indices: $\{X_{i_s}, X_{j_t}^2 : i_s = j_t = c\}$, $c = 1,\ldots,n$, and are thus described not by a single number (group size) but by a pair (a, b), where a is the number of i^*s and b is the number of j^*s in the group. Consequently, a grouping in this version is characterized by a set of pairs $\{(a_u, b_u)\}$, $u = 1,\ldots,d$; $\sum_{u=1}^{d} a_u = k$, $\sum_{u=1}^{d} b_u = l$ and its definition is different from the one in 3.1 since for given k and l there can be different

groupings that yield the same expectation, e.g. groupings {(2,3),(3,0),(1,1)}, {(4,2),(1,1),(1,1)}, and {(0,4),(3,0),(3,0)} will all produce $E(X_{i_1}\cdots X_{i_6}X_{j_1}^2\cdots X_{j_4}^2)=\mu_3^2\mu_8$. Analogously to the original version, if $\sum_{u=1}^{d} I(a=1,b=0)>0$ (at least one pair in the grouping is (1,0)), $E(X_{i_1}\cdots X_{i_k}X_{j_1}^2\cdots X_{j_l}^2)=0$; otherwise $E(X_{i_1}\cdots X_{i_k}X_{j_1}^2\cdots X_{j_l}^2)=\prod_{u=1}^{d}\mu_{a_u+2b_u}$. Note that to account for all the possible groupings in this case, permutations need to be used, adding another layer to computational complexity.

Coefficient C_g for a grouping G is calculated in a similar way to 3.1 (equation (6)) with a few adjustments:

$$C_g=(n)_d\frac{k!\,l!}{a_{g,1}!a_{g,2}!\cdots a_{g,d}!b_{g,1}!b_{g,2}!\cdots b_{g,d}!s_{g,1}!s_{g,2}!\cdots}, \tag{8}$$

where $s_{g,1},s_{g,2},\ldots$ are the numbers of the groups with same values for a and b.

In this case the order ranges from $\frac{1}{n^{k+l-1}}$, when $i_1=\ldots=i_k=j_1=\ldots=_{\mathrm{jk}}$ ($d=1$), to $\frac{1}{n^{\left\lceil\frac{k}{2}\right\rceil}}$, when all indices i_s appear in pairs if k is even ("extra" index joining one of the groups if k is odd), and all the j_t's are different from i_s's and each other ($d=\left\lfloor\frac{k}{2}\right\rfloor+l$).

3.3. General Case

The procedure in section 3.2 easily generalizes to finding $E\left(\overline{X}^{j_1}\overline{X^2}^{j_2}\overline{X^3}^{j_3}\cdots\overline{X^m}^{j_m}\right)$ for an arbitrary m, with groups described by a "tuple" of length m and a grouping being a collection of such tuples. Coefficients C_g for groupings G are calculated similarly to (8), accounting for all the elements in each tuple.

4. Results (up to 6th order)

Below are the results generated with our software (*SymPy* code that produces these results is in a https://github.com/innager/unbiasedMoments *Jupyter notebook*):

$$
\begin{aligned}
M(\mu_3) &= \frac{n^2}{(n-1)(n-2)}m_3 \\
M(\mu_4) &= -\frac{3n(2n-3)}{(n-1)(n-2)(n-3)}m_2^2 + \frac{n\left(n^2-2n+3\right)}{(n-1)(n-2)(n-3)}m_4 \\
M(\mu_2^2) &= \frac{n\left(n^2-3n+3\right)}{(n-1)(n-2)(n-3)}m_2^2 - \frac{n}{(n-2)(n-3)}m_4 \\
M(\mu_5) &= -\frac{10n^2}{(n-1)(n-3)(n-4)}m_2m_3 + \frac{n^2\left(n^2-5n+10\right)}{(n-1)(n-2)(n-3)(n-4)}m_5 \\
M(\mu_2\mu_3) &= \frac{n^2\left(n^2-2n+2\right)}{(n-1)(n-2)(n-3)(n-4)}m_2m_3 - \frac{n^2}{(n-2)(n-3)(n-4)}m_5 \\
M(\mu_6) &= \frac{15n^2(3n-10)}{(n-1)(n-2)(n-3)(n-4)(n-5)}m_2^3 - \frac{15n\left(n^3-8n^2+29n-40\right)}{(n-1)(n-2)(n-3)(n-4)(n-5)}m_2m_4 \\
&\quad - \frac{40n\left(n^2-6n+10\right)}{(n-1)(n-2)(n-3)(n-4)(n-5)}m_3^2 + \frac{n(n^4-9n^3+31n^2-39n+40)}{(n-1)(n-2)(n-3)(n-4)(n-5)}m_6 \\
M(\mu_2\mu_4) &= -\frac{3n^2(2n-5)}{(n-1)(n-3)(n-4)(n-5)}m_2^3 + \frac{n(n^4-9n^3+53n^2-135n+120)}{(n-1)(n-2)(n-3)(n-4)(n-5)}m_2m_4 \\
&\quad + \frac{4n\left(n^2-5n+10\right)}{(n-1)(n-3)(n-4)(n-5)}m_3^2 - \frac{n\left(n^2-3n+8\right)}{(n-2)(n-3)(n-4)(n-5)}m_6 \\
M(\mu_3^2) &= -\frac{3n^2\left(3n^2-15n+20\right)}{(n-1)(n-2)(n-3)(n-4)(n-5)}m_2^3 + \frac{3n\left(2n^3-5n^2-5n+20\right)}{(n-1)(n-2)(n-3)(n-4)(n-5)}m_2m_4 \\
&\quad + \frac{n\left(n^4-8n^3+25n^2-10n-40\right)}{(n-1)(n-2)(n-3)(n-4)(n-5)}m_3^2 - \frac{n\left(n^2-n+4\right)}{(n-2)(n-3)(n-4)(n-5)}m_6 \\
M(\mu_2^3) &= \frac{n^2\left(n^2-7n+15\right)}{(n-1)(n-3)(n-4)(n-5)}m_2^3 - \frac{3n\left(n^2-5n+10\right)}{(n-1)(n-3)(n-4)(n-5)}m_2m_4 \\
&\quad - \frac{2n\left(3n^2-15n+20\right)}{(n-1)(n-2)(n-3)(n-4)(n-5)}m_3^2 + \frac{2n}{(n-3)(n-4)(n-5)}m_6
\end{aligned}
$$

For two-sample pooled estimators up to 6th order, refer to the *Umoments* package [9] and a https://github.com/innager/unbiasedMoments *Sage worksheet*.

5. *Umoments* R Package

Umoments contains a set of pre-programmed functions that calculate one-sample and pooled two-sample unbiased moment estimates, both up to sixth order. This functionality is primarily useful for data analysis. The estimates can be calculated either directly from the sample or from naïve biased estimates, in which case sample size n needs to be provided. For two-sample estimation, input should also include labels indicating which observation belongs to which sample/category, or both n_x and n_y for sample sizes. Below are some examples.

Two-sample pooled estimates from the data up to sixth order (note that smp is a data vector, and treatment is a vector of labels that separates it into two categories):

> uMpool(smp, treatment, 6)

M2 M3 M2pow2 M4 M2M3 M5 M2pow3

1.6443027 1.5188515 2.4878505 6.9794503 2.0615514 17.0989234 3.5236856

M3pow2 M2M4 M6

0.6674177 9.4046220 56.6016025

Unbiased estimate of μ_3^2 from naïve biased 2nd, 3rd, 4th, and 6th moment estimates:

```
> uM3pow2(m[2], m[3], m[4], m[6], n)

[1] -10.00696
```

Other functions in the package can be used to obtain higher-order estimators, pooled estimators across multiple (3 or more) samples, and other statistical results.

Generate $E\left(\bar{X}^3\overline{X^3}^2\overline{X^4}\right)$ for a sample $X_1 \ldots X_{n_X}$ (the output is a string that could be used as a code chunk, fed into a computer algebra system, or converted into latex):

```
> one_combination(c(3, 0, 2, 1), "n_x") [1] "
( 1*n_x*mu13^1 + 3*n_x*(n_x-1)*mu2^1*mu11^1 + 3*n_x*(n_x-1)*(n_x-2)*
(n_x-3)*mu5^1*mu3^2*mu2^1 + 6*n_x*(n_x-1)*(n_x-2)*(n_x-3)*mu4^2*mu3^1*mu2^1 +
6*n_x*(n_x-1)*(n_x-2)*mu8^1*mu3^1*mu2^1 + 9*n_x*(n_x-1)*(n_x-
2)*mu7^1*mu4^1*mu2^1 +
3*n_x*(n_x-1)*(n_x-2)*mu6^1*mu5^1*mu2^1 + 3*n_x*(n_x-1)*mu3^1*mu10^1 + 1*n_x*
(n_x-1)*(n_x-2)*(n_x-3)*mu4^1*mu3^3 + 3*n_x*(n_x-1)*(n_x-2)*mu7^1*mu3^2 +
9*n_x*(n_x-1)*(n_x-2)*mu6^1*mu4^1*mu3^1 + 6*n_x*(n_x-1)*(n_x-2)*mu5^2*mu3^1 +
12*n_x*(n_x-1)*(n_x-2)*mu5^1*mu4^2 + 7*n_x*(n_x-1)*mu9^1*mu4^1 + 9*n_x*(n_x-1)*
mu8^1*mu5^1 + 6*n_x*(n_x-1)*mu7^1*mu6^1 ) / n_x^6"
```

Generate groupings for k=5 (see section 3.1):

```
> Umoments:::groups(5)

[[1]]

[1] 1 1 1 1 1

[[2]]

[1] 2 1 1 1

[[3]]

[1] 2 2 1

[[4]]

[1] 3 1 1

[[5]]

[1] 3 2

[[6]]
```

```
[1] 4 1

[[7]]

[1] 5
```

For further details and examples refer to package vignette and documentation [9].

6. Discussion

The difference between unbiased and biased estimators depends on the sample size and might be considerable for small samples; also, for fixed sample size, it is relatively greater for higher orders. At the same time, variability of the estimators is an important factor to be considered in this bias-variance trade-off, especially in connection with sample size n and the order of the estimators as variability increases with higher orders (which might be offset by the lower contribution/weight of these orders in certain methods) and smaller samples. Another question is the relationship between n and the maximal order that could reasonably be used in a method; besides purely algebraic restrictions on a sample size given the order, apparent from the expressions for unbiased estimators ($n \geqslant k$ for k'th order estimators), there might be another underlying stricter relationship that needs to be explored, either theoretically or numerically.

While unbiased estimators of products and integer powers of moments are possible to obtain, that is not the case with ratios and roots. Of course, such biased estimators, like a square root s of sample variance $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$ or skewness estimator, are widely used in practice. Adding to the complexity is the fact that since unbiased estimator of the ratio cannot be obtained, simplifying expressions should also be questioned - consider, for example, scaled sixth cumulant:

$$\lambda_6 = \frac{\kappa_6}{\mu_2^3} = \frac{\mu_6 - 15\mu_2\mu_4 - 10\mu_3^2 + 30\mu_2^3}{\mu_2^3} = \frac{\mu_6}{\mu_2^3} - 15\frac{\mu_4}{\mu_2^2} - 10\frac{\mu_3^2}{\mu_2^3} + 30$$

For a closest estimate, it is natural to consider the ratio of an unbiased cumulant estimator $M(\kappa_6)$ and an unbiased scaling factor $M(\mu_2^3)$. Then, is $\frac{M(\mu_2\mu_4)}{M(\mu_2^3)}$ preferable to $\frac{M(\mu_4)}{M(\mu_2^2)}$ for the second term?

This example also provides an illustration for another important consideration that should factor into a decision of which estimators to use - variability of the denominator in studentized statistics. In λ_6, sixth cumulant κ_6 is scaled by μ_2^3; to substitute for this unknown quantity, a variety of estimators can be used: m_2^3, $[M(\mu_2)]^3$, or $M(\mu_2^3)$, to name a few. While expression for $M(\mu_2)$ (and thus its cube) contains m_2 only, the expression for $M(\mu_2^3)$ includes m_4 and m_6 as well. These higher-order quantities may be highly variable, especially in the small sample, and therefore the whole ratio becomes highly sensitive to the small values of estimates in the denominator that can inflate λ_6 dramatically, increasing variability of the

ratio to the point of unusability. Therefore it might be advisable in certain cases, *e.g.* with considerably skewed distributions, to perform some numeric exploration to determine if it might be indeed preferable to use lower-order estimators, naïve biased or unbiased, in place of parameters in denominators because of their relative stability ("power of mean" instead of "mean of power").

Acknowledgments

Funding

This work was supported by NIH under Grant P42ES004705.

Biography

Inna Gerlovina

Inna Gerlovina is a postdoctoral scholar at the University of California, San Francisco. She has worked on small sample inference and error rate control as well as higher-order inferential approaches, developing software packages for high-dimensional data analysis. Her current interests include development, implementation, and application of statistical methods that contribute to understanding of malaria epidemiology and transmission. Inna completed her MA and PhD in Biostatistics at the University of California, Berkeley.

Alan Hubbard

Alan Hubbard, Professor of Biostatistics, Univ. of California, Berkeley, is a director of the Biomedical Big Data pre-doctoral training program, and co-director of the Center of Targeted Learning, is head of the computational biology Core E of the SuperFund Center at UC Berkeley (NIH/EPA), as well a consulting statistician on several federally funded and foundation projects. He has worked as well on projects ranging from molecular biology of aging, epidemiology, and infectious disease modeling, but most all of his work has focused on semi-parametric estimation in high-dimensional data. His current methods-research focuses on precision medicine, variable importance, statistical inference for data-adaptive parameters, and statistical software implementing targeted learning methods. Currently working in several areas of applied research, including early childhood development in developing countries, patient outcomes from acute trauma, environmental genomics and comparative effectiveness research in diabetes care.

References

[1]. Pebay PP, Terriberry T, Kolla H, et al. Formulas for the computation of higher-order central moments. Sandia National Lab.(SNL-CA), Livermore, CA (United States); 2014.

[2]. Bickel P Edgeworth expansions in nonparametric statistics. The Annals of Statistics. 1974;:1–20.

[3]. Hall P, et al. Edgeworth expansion for student's *t* statistic under minimal moment conditions. The Annals of Probability. 1987;15(3):920–931.

[4]. Putter H, van Zwet WR, et al. Empirical edgeworth expansions for symmetric statistics. The Annals of Statistics. 1998;26(4):1540–1569.

[5]. Fisher RA. Moments and product moments of sampling distributions. Proceedings of the London Mathematical Society. 1930;2(1):199–238.

[6]. Cramér H Mathematical methods of statistics (pms-9). Vol. 9 Princeton university press; 2016.

[7]. Rimoldini L Weighted skewness and kurtosis unbiased by sample size and gaussian uncertainties. Astronomy and Computing. 2014;5:1–8.

[8]. Rose C, Smith M. Mathematical statistics with mathematica, chapter 7: Moments of samplng distributions. Springer-Verlag, New York 2002;.

[9]. Gerlovina I, Hubbard AE. Umoments: Unbiased central moment estimates; 2019 R package version 0.1.0; Available from: https://CRAN.R-project.org/package=Umoments.

[10]. Tracy D, Gupta B, et al. Generalized *h*-statistics and other symmetric functions. The Annals of Statistics. 1974;2(4):837–844.

[11]. Dwyer PS. Combined expansions of products of symmetric power sums and of sums of symmetric power products with application to sampling. The Annals of mathematical statistics. 1938;9(1):1–47.

Public Interest Statement

Higher-order statistics are increasingly used in various research fields and data analysis, and central moment estimates are useful for many approaches. Derivation of higher-order unbiased central moment estimators has long been a challenging task; software made the general order solution possible. This paper describes a direct approach to obtaining estimators of any order, including multi-sample pooled estimators. It also introduces an open source R package *Umoments*, which calculates one- and two-sample estimates up to 6th order and contains machinery to obtain even higher order estimates, including a combinatorial algorithm that can be used for solving other problems and assist in long derivations (e.g. Edgeworth expansions).