# Chapter 1

# Order Statistics

**This presentation delivered to**
**CE397 Statistics in Water Resources, University of Texas at Austin**
**Spring 2009 on March 31, 2009**
**by William H. Asquith Ph.D.**

**The software used is R from www.r-project.org and is available for all platforms. The `lmomco` package provides some functions discussed in class related to order statitics and L-moments and can be found at `www.cran.r-project.org/package=lmomco`.**

## 1.1 Introduction

A branch of statistics known as **order statistics** plays a prominent role in L-moment theory. The study of order statistics is the study of the statistics of ordered (sorted) random variables and samples. This chapter presents a very brief introduction to order statistics to provide a foundation for later chapters. A comprehensive exposition of order statistics is provided by David (1981), and an R-oriented approach is described in various contexts by Baclawski (2008).

The random variable $X$ for a sample of size $n$, when sorted, forms the order statistics of $X$: $X_{1:n} \leq X_{2:n} \leq \cdots \leq X_{n:n}$. The **sample order statistics** from a random sample are created by sorting the sample into ascending order: $x_{1:n} \leq x_{2:n} \leq \cdots \leq x_{n:n}$. As we will see, the concept and use of order statistics take into account both the value (magnitude) and the relative relation (order) to other observations. Barrett (2004, p. 23) reports that

> ... the effects of ordering can be impressive in terms of both what aspects of sample behavior can be usefully employed and the effectiveness and efficiency of resulting inferences.

and that

> ... linear combinations of all ordered samples values can provide efficient estimators.

This presentation will show that the L-moments, which are based on linear combinations of order statistics, do in fact provide efficient estimators of distributional geometry.

In general, order statistics are already a part of the basic summary statistic repertoire that most individuals—including nonscientists or statisticians—are familiar with. The **minimum** and **maximum** are examples of extreme order statistics and are defined by the following notation

$$\min\{X_n\} = X_{1:n} \tag{1.1}$$

$$\max\{X_n\} = X_{n:n} \tag{1.2}$$

The familiar **median** $X_{0.50}$ by convention is

$$X_{0.50} = \begin{cases} (X_{[n/2]:n} + X_{[(n/2)+1]:n})/2 & \text{if } n \text{ is even} \\ X_{[(n+1)/2]:n} & \text{if } n \text{ is odd} \end{cases} \tag{1.3}$$

and thus clearly is defined in terms of one order statistic or a linear combination of two order statistics.

Other order statistics exist and several important interpretations towards the purpose of this presentation can be made. Concerning L-moments, Hosking (1990, p. 109) and Hosking and Wallis (1997, p. 21) provide an "intuitive" justification for L-moments and by association the probability-weighted moments. The justification follows:

- The order statistic $X_{1:1}$ (a single observation) contains information about the location of the distribution on the real-number line $\mathbb{R}$;
- For a sample of $n = 2$, the order statistics are $X_{1:2}$ (smallest) and $X_{2:2}$ (largest). For a highly dispersed distribution, the expected difference between $X_{2:2} - X_{1:2}$ would be large, whereas for a tightly dispersed distribution, the difference would be small. The expected differences between order statistics of an $n = 2$ sample hence can be used to expression the variability or scale of a distribution; and
- For a sample of $n = 3$, the order statistics are $X_{1:3}$ (smallest), $X_{2:3}$ (median), and $X_{3:3}$ (largest). For a negatively skewed distribution, the difference $X_{2:3} - X_{1:3}$ would be larger (more data to the left) than $X_{3:3} - X_{2:3}$. The opposite (more data to the right) would occur if a distribution where positively skewed.

These interpretations show the importance of the intra-sample differences in the expression of distribution geometry.

## *Expectations and Distributions of Order Statistics*

A fundamental definition regarding order statistics, which will be critically important in the computation of L-moments and probability-weighted moments, is the expectation of an order statistic. The expectation is defined in terms of the QDF. The expectation of an order statistic for the $j$th largest of $r$ values is defined (David, 1981, p. 33) in terms of the QDF $x(F)$ as

$$\mathrm{E}[X_{j:n}] = \frac{n!}{(j-1)!(n-j)!} \int_0^1 x(F) \times F^{j-1} \times (1-F)^{n-j} \, \mathrm{d}F \tag{1.4}$$

The expectation of an order statistic for a sample of size $n = 1$ is especially important because

$$\mathrm{E}[X_{1:1}] = \int_0^1 x(F) \, \mathrm{d}F = \mu = \text{arithmetic mean} \tag{1.5}$$

Therefore, the familiar mean can be interpreted thus: The mean is the expected value of a single observation if one and only one sample is drawn from the distribution.

Hosking (2006) reports from references cited therein that "the expectations of extreme order statistics characterize a distribution." In particular, if the expectation of a random variable $X$ is finite, then the set $\{\mathrm{E}[X_{1:n} : n{=}1,2,\cdots]\}$ or $\{\mathrm{E}[X_{n:n} : n{=}1,2,\cdots]\}$ uniquely determine the distribution. Hosking (2006) reports that such sets of expectations contain redundant information and that technically a subset of expectations can be dropped and the smaller set is still sufficient to characterize the distribution.

## USING R

Using eq. (1.4) and R, the expected value of the 123rd-ordered (increasing) value of a sample of size $n = 300$ is computed for an Exponential distribution in example 1–1. The ratio of factorial functions in eq. (1.4) is difficult to compute for large values—judicious use of the fact that $n! = \Gamma(n+1)$ and use of logarithms of the complete Gamma function $\Gamma(a)$ suffices. The results of the integration using QDF of the Exponential by the `qexp()` function and stochastic computation using random variates of the Exponential by the `rexp()` function for $\mathrm{E}[X_{123:300}]$ are equivalent.

1–1

```
nsim <- 10000; n <- 300; j <- 123
int <- integrate(function(f,n=NULL,j=NULL) {
                   exp(lgamma(n+1) - lgamma(j) - lgamma(n-j+1)) *
                   qexp(f) * f^(j-1) * (1-f)^(n-j)
                 }, lower=0, upper=1, n=n, j=j)
E_integrated <- int$value
```

```
E_stochastic <- mean(replicate(nsim, sort(rexp(n))[j]))
cat(c("RESULTS:", round(E_integrated, digits=3),
      "and",      round(E_stochastic, digits=3), "\n"))
RESULTS: 0.526 and 0.527
```

◀

## *Distributions of Order Statistic Extrema*

The extrema $X_{1:n}$ and $X_{n:n}$ are of special interest in many practical problems of distributional analysis. Let us consider the sample maximum of random variable $X$ having CDF of $F(x) = \Pr[X_{n:n} \leq x]$. If $X_{n:n} \leq x$, then all $x_i \leq x$ for $i = 1, 2, \cdots, n$, it can be shown that

$$F_n(x) = \Pr[X \leq x]^n = \{F(x)\}^n \tag{1.6}$$

Similarly, it can be shown for the sample minimum that

$$F_1(x) = \Pr[X > x]^n = \{1 - F(x)\}^n \tag{1.7}$$

Using the arguments producing eqs. (1.6) and (1.7) with a focus on the QDF, Gilchrist (2000, p. 85) provides

$$x_{n:n}(F_{n:n}) = x(F_{n:n}^{1/n}) \tag{1.8}$$

$$x_{1:n}(F_{1:n}) = x(1 - (1 - F_{1:n})^{1/n}) \tag{1.9}$$

for the maximum and minimum, respectively. Gilchrist (2000, p. 85) comments that, at least for $x_{n:n}$ that "the quantile function of the largest observation is thus found from the original quantile function in the simplest of calculations."

For the general computation of the distribution of non extrema order statistics, the computations are more difficult. (Gilchrist, 2000, p. 86) shows that the QDF of the distribution of the $j$th order statistic of a sample of size $n$ is

$$x_{j:n}(F_{j:n}) = x[\mathrm{B}^{(-1)}(F_{j:n},\ j,\ n-j+1)] \tag{1.10}$$

where $x_{j:n}(F_{j:n})$ is to be read as "the QDF of the $j$th order statistic for a sample of size $n$ given by nonexceedance probability $F_{j:n}$." The function $\mathrm{B}^{(-1)}(F,a,b)$ is the QDF of the Beta distribution—the $^{(-1)}$ notation represents the inverse of the CDF, which is of course a QDF. It follows that the QDF of the order statistic extrema for an $F$ are

$$x_{1:n}(F) = x[\mathrm{B}^{(-1)}(F,\ 1,\ n)] \tag{1.11}$$

$$x_{n:n}(F) = x[\mathrm{B}^{(-1)}(F,\ n,\ 1)] \tag{1.12}$$

for the minimum $X_{1:n}$ and maximum $X_{n:n}$, respectively.

USING R

In the context of eqs. (1.6) and eq. (1.7), the expectations of extrema for the Exponential distribution are stochastically computed in example 1–2 using the `min()` and `max()` functions. The random variates from the Exponential are computed by the `rexp()` function. The example begins by setting the sample size $n = 4$, the size of a simulation run in `nsim`, and finally, the scale parameter (note that R uses a rate expression for the dispersion parameter) of the Exponential distribution is set to 1000. (A location parameter of 0 is implied.) The example reports (1000, 1500, 500) for the respective mean, maximum, and minimum values. (It is well known that the mean of this Exponential distribution is 1000.)

1–2

```
n <- 4; nsim <- 200000
s <- 1/1000 # inverse of scale parameter = 1000

# Expectation of Expectation of Exponential Distribution
mean(replicate(nsim, mean(rexp(n, rate=s))))
```

```
[1] 1000.262

# Expectation of Maximum from Exponential Distribution
mean(replicate(nsim, max(rexp(n, rate=s))))
[1] 1504.404

# Expectation of Minimum from Exponential Distribution
mean(replicate(nsim, min(rexp(n, rate=s))))
[1] 499.6178
```

The demonstration continues in example 1–3 with the stochastic computation of the expected values of the maximum and minimum through eqs. (1.6) and (1.7). One consideration that is so interesting about using eqs. (1.6) and (1.7) is that sorting a vector of extrema distributed values as for the maximum and minimum computation is not needed. (The quantiles of the Exponential are computed by the `qexp()` function; whereas, Uniform variates are computed by the `runif()` function.) The output of examples 1–2 – 1–3 are consistent with each other.

1–3

```
# Expectation of Maximum from Exponential Distribution
mean(qexp(runif(nsim)^(1/n), rate=s))
[1] 1497.001

# Expectation of Minimum from Exponential Distribution
mean(qexp(1 - runif(nsim)^(1/n), rate=s))
[1] 501.1628
```

◀

It was implied from the two previous examples that eqs. (1.6) and (1.7) provide a more efficient means of computing the distribution of extrema because sorting is computationally expensive. Let us use the `system.time()` function in example 1–4 to measure just how long computing the expectation of a minimum value of a sample of size $n = 4$. We see that use of eq. (1.7) is more than 35 times faster.

1–4
```
system.time(mean(replicate(nsim, min(qexp(runif(n), rate=s)))))
   user  system elapsed
  3.337   0.047   3.502

system.time(mean(qexp(1 - runif(nsim)^(1/n), rate=s)))
   user  system elapsed
  0.059   0.006   0.064
```

◀

The distributions of individual order statistics in eq. (1.10) are easily demonstrated using R. The following example 1–5 defines a function `qua.ostat()` for computation of the quantiles of a given order statistics. The arguments `f` and `para` to the function are the $F_{j:n}$ and *lmomco* parameter list. The parameter list is a data structure specific to the *lmomco* package. The other two arguments are self explanatory. The `qbeta()` is the built-in R function used to compute quantiles of the Beta distribution. Finally, the `par2qua()` function of *lmomco* dispatches the `para` parameter list to the appropriate distribution with $F =$ `betainv.F`.

1–5
```
"qua.ostat" <-
function(f,j,n,para) {
  betainv.F <- qbeta(f,j,n-j+1) # compute nonexceedance prob.
  return(par2qua(betainv.F,para))
}
# Now demonstrate usage of the qua.ostat() function
PARgpa <- vec2par(c(100,500,0.5), type="gpa") # make parameters
n <- 20; j <- 15; F <- 0.5 # sample size, rank, and nonexceedance
ostat <- qua.ostat(F,j,n,PARgpa); print(ostat)
[1] 571.9805
```

After defining the `qua.ostat()` function using the `function()` "function," the example continues by specifying an *lmomco* parameter list for the Generalized Pareto distribution into variable `PARgpa` using `vec2par()` through the `type="gpa"` argument. A sample size of $n = 20$ is set, and the median of the distribution of the 15th-order statistic for

such a sample is computed. The code reports $x_{15:20}(0.5) = 572$ or the "50th percentile of the 15th value of a sample of size 20." The `qua.ostat()` function actually is incorporated into the *lmomco* package. The function is shown here because it is a good example of syntax brevity by which eq. (1.10) can be implemented using the vectorized nature of the R language. ◀

## 1.2 Sampling Bias and Sampling Variability

The concepts of **sampling bias** and **sampling variability** (Stedinger and others, 1993, p. 18.10) involve the **accuracy** and **precision** of statistical estimation. Because distributional analysis inherently involves finite samples, the concepts of sampling bias and variability are important. R-oriented treatments of these and related concepts are provided by Rizzo (2008, p. 37–38) and Ugarte and others (2008, pp. 245–255). For a given circumstance perhaps statistics such as moments, percentiles, or distribution parameters are to be estimated. Whichever is the case, consider the estimated statistic $\hat{\Theta}$ as a random variable with a true value that is simply denoted as $\Theta$. Values for $\hat{\Theta}$ are dependent on the sampled data values. The bias in the estimation of $\hat{\Theta}$ is defined as the difference between the expectation of the estimate minus the true value or

$$Bias[\hat{\Theta}] = E[\hat{\Theta}] - \Theta \tag{1.13}$$

The sample-to-sample variability (or sampling variability) of a statistic is expressed by **root mean square error**, which is defined as

$$RMSE[\hat{\Theta}] = \sqrt{E[(\hat{\Theta} - \Theta)^2]} \tag{1.14}$$

and upon expansion the error is split into two parts

$$RMSE[\hat{\Theta}] = \sqrt{Bias[\hat{\Theta}]^2 + E[(\hat{\Theta} - E[\hat{\Theta}])^2]} \tag{1.15}$$

or

$$RMSE[\hat{\Theta}] = \sqrt{Bias[\hat{\Theta}]^2 + Var(\hat{\Theta})} \tag{1.16}$$

The square of the *RMSE* is known as the **mean square error** (*MSE*). Rizzo (2008, p. 155) reports for *MSE*, but shown here as *RMSE*, that

$$RMSE[\hat{\Theta}] = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(\hat{\Theta}^{(j)} - \Theta)^2} \tag{1.17}$$

where $\Theta^{(j)}$ is the estimator for the $j$th sample of size $n$.

*Bias*, $Var(\hat{\Theta})$, and *RMSE* are useful measures of statistical performance. They are performance measures because the sampling bias and sampling variability describe the accuracy and precision, respectively, of the given estimator.

If $Bias[\hat{\Theta}] = 0$, then the estimator is said to be **unbiased**. For an unbiased estimator, the sampling variability will be equal to the variance $Var(\hat{\Theta})$ of the statistic. These two measures of statistical performance can exhibit considerable dependency on sample size $n$.

Amongst an ensemble of estimators, the estimator with the smallest $RMSE[\hat{\Theta}]$ or $MSE[\hat{\Theta}]$ is said to be the most **efficient**. If an estimator is resistant to large changes because of the presence of outliers or otherwise influential data values, then the estimator is said to be **robust**. The **relative efficiency** of two estimators is

$$RE(\hat{\Theta}_1, \hat{\Theta}_2) = \frac{MSE(\hat{\Theta}_2)}{MSE(\hat{\Theta}_1)} \tag{1.18}$$

and when two estimators are unbiased, then the relative efficiency can be defined as

$$RE(\hat{\Theta}_1, \hat{\Theta}_2) = \frac{Var(\hat{\Theta}_2)}{Var(\hat{\Theta}_1)} \tag{1.19}$$

Relative efficiency is important in assessing or otherwise comparing the performance of two estimators.

USING R

Sampling bias and sampling variability are used herein as metrics to evaluate and compare the properties of product moments, L-moments, and other statistics. For the sake of brevity, the R functions `mean()`, `sd()`, and occasionally `summary()` generally will be used to compute statistics of the difference $\hat{\Theta} - \Theta$. However, let us take an opportunity to delve into statistics of $\hat{\Theta} - \Theta$ in more detail.

In example 1–6, the function `afunc()` is defined as a high-level interface to the distribution of choice. For the example, the random variates for the standard Normal distribution are accessed through the `rnorm()` function. This style of programming is shown in order to make extension to non-standard R distibutions easier. Such a programming practice is known as abstraction. Next, the function `sam.biasvar()` is defined to compute eqs. (1.13) and (1.16) as well as $Var(\hat{\Theta})$.

1–6

```
MN <- 0; SD <- 1 # parameters of standard normal
# Define a separate function to implement a distribution
"afunc" <- function(n,mean,sd) {
   return(rnorm(n, mean=mean, sd=sd))
}
nsim <- 100000; n <- 10 # no. simulations and sample size to sim.

# Define function to compute sampling statistics
"sam.biasvar" <- function(h,s, verbose=TRUE, digits=5) {
   b <- mean(h) - s     # solve for the bias

   mse  <- mean((h - s)^2) # mean square error
   rmse <- sqrt(mse)       # root MSE

   vh   <- sqrt(mean((h - mean(h))^2)) # sqrt(variance
   # of the statistic), which lacks a n-1 division

   nv   <- sqrt(rmse^2 - b^2) # alternative estimation
```

```
  if(verbose) {
    cat(c("Bias␣(B)␣␣␣␣␣␣␣␣␣=␣",round(b,digits=digits),   "\n",
          "MSE(h,s)␣␣␣␣␣␣␣␣␣=␣",round(mse,digits=digits), "\n",
          "RMSE(h,s)␣␣␣␣␣␣␣␣=␣",round(rmse,digits=digits),"\n",
          "sqrt(Var(h))␣␣␣␣␣=␣",round(vh,digits=digits),  "\n",
          "sqrt(RMSE^2-B^2)␣=␣",round(nv,digits=digits),  "\n"),
        sep="")
  }
  return(list(bias=b, mse=mse, rmse=rmse, sd=vh))
}
```

The `sam.biasvar()` is demonstrated in example 1–7 for a sample of size $n = 10$ for a large simulation size `nsim=100000`. First, the `Rmean` list is generated to hold the sampling statistics of the `mean()` function, and second, the `Rmedn` list is generated to hold the sampling statistics of the `median` function. The reported biases are near zero because the mean and median are both unbiased estimators.

1–7

```
# Sampling statistics of the mean()
Rmean <- sam.biasvar(replicate(nsim,mean(afunc(n,MN,SD))),MN)
Bias (B)          = -0.00158
MSE(h,s)          = 0.10058
RMSE(h,s)         = 0.31714
sqrt(Var(h))      = 0.31713
sqrt(RMSE^2-B^2) = 0.31713
# Report the theoretical to show equivalence
cat(c("Theoretical␣=␣",
      round(SD/sqrt(n), digits=3), "\n"), sep="")
Theoretical = 0.316

# Sampling statistics of the median()
Rmedn <- sam.biasvar(replicate(nsim,median(afunc(n,MN,SD))),MN)
Bias (B)          = 0.00132
MSE(h,s)          = 0.13717
```

```
RMSE(h,s)         = 0.37036
sqrt(Var(h))      = 0.37036
sqrt(RMSE^2-B^2) = 0.37036

RE <- (Rmean$sd/Rmedn$sd)^2 # RE^{mean}_{median} in LaTeX
cat(c("Relative␣efficiency␣=␣",
      round(RE,digits=3), "\n"), sep="")
Relative efficiency = 0.733
```

A natural followup question is asked of the mean and the median. Which has the smaller variance? The end of example 1–7 reports that $RE(\text{mean}, \text{median}) \approx 0.73$, which is less than unity so the conclusion is that the arithmetic mean has a smaller sampling variance than the median. ◀

A previous demonstration of *MSE* computation is made for a trimmed mean and the median using `sam.biasvar()` in example 1–8. ◀

An informative example in the context of the trimmed mean is provided. We compute in example 1–8 the mean square errors (*MSE*) of the `sen.mean()`, `trim.mean()` (Rizzo, 2008, p. 156), and `median()` estimators and compare the three errors to those reported by (Rizzo, 2008, pp. 156–157). The example begins by defining a `trim.mean()` function and using the same sample size $n = 20$ as used by Rizzo. For this particular example, the `set.seed()` function is used to set a seed for the random number generator in current use by R. By setting the seed, users for this example should precisely reproduce the output shown.[1]

1–8

```
"trim.mean" <- function(x) { # mimicking Rizzo (2008)
  x <- sort(x); n <- length(x); return(sum(x[2:(n-1)])/(n-2))
}
n <- 20; nsim <- 75000
set.seed(1000) # set the seed for the random number generator
```

[1] Note that the general practice in this presentation is to be independent of specific seeds so users should expect numerically different, but stochastically similar results for other examples herein.

```
S1 <- replicate(nsim, sen.mean(rnorm(n))$sen)
sam.biasvar(S1,0, verbose=FALSE)$mse
[1] 0.04990509

# Sampling statistics of the trim.mean()
# Rizzo (2008) p.156 reports mse=0.0518
S2 <- replicate(nsim, trim.mean(rnorm(n)))
sam.biasvar(S2,0, verbose=FALSE)$mse
[1] 0.05124172

# Rizzo (2008) p.157 reports mse=0.0748
S3 <- replicate(nsim, median(rnorm(n)))
sam.biasvar(S3,0, verbose=FALSE)$mse
[1] 0.07363024
```

The example continues using the `sam.biasvar()` function that is created in example 1–6 and also used in example 1–7 to perform `nsim` simulations of the `sen.mean()`, `trim.mean()`, and `median()` estimates of the standard Normal distribution. The results here show numerical equivalency between the values reported by Rizzo. Further, the results show that the equivalent algorithms for `sen.mean()` and `trim.mean()` have smaller mean square errors than the familiar median. This is a natural consequence of the median using far less numerical information contained in the sample than the trimmed mean uses. ◀

## 1.3 L-estimators—Special Statistics Related to L-moments

Jurečková and Picek (2006, pp. 63–70) summarize statistical estimators known as **L-estimators**. L-estimators $T_n$ for sample of size $n$ are based on the order statistics and are expressed in a general form as

$$T_n = \sum_{i=1}^{n} c_{i:n} h(X_{i:n}) + \sum_{i=1}^{n} d_j h^{\star}(X_{[np_j+1]:n}) \tag{1.20}$$

where $X_{i:n}$ are the order statistics, $c_{1:n}, \cdots, c_{n:n}$ and $a_1, \cdots, a_n$ are given coefficients or weight factors, $0 < p_1 < \cdots < p_k < 1$, and $h(a)$ and $h^{\star}(a)$ are given functions for argument $a$. The coefficients $c_{i:n}$ for $1 \leq i \leq n$ are generated by a bounded weight function $J(a)$ with a domain $[0,1]$ with a range of the real-number line $\mathbb{R}$ by either

$$c_{i:n} = \int_{(i-1)/n}^{i/n} J(s)\, \mathrm{d}s \tag{1.21}$$

or approximately

$$c_{i:n} = \frac{J(i/[n+1])}{n} \tag{1.22}$$

The quantity to the left of the + in eq. (1.20) uses all of the order statistics whereas the quantity to the right of the + is a linear combination of a finite number of order statistics (quantiles). L-estimators generally have the form of either quantity, but not both. Estimators defined by the left quantity are known and type I and those of the right are known as type II.

The simplest examples suggested by Jurečková and Picek (2006, p. 64) of an L-estimator of distribution location are the sample median and the **midrange**, in which the later is defined as

$$T_n = \frac{X_{1:n} + X_{n:n}}{2} \tag{1.23}$$

A simple L-estimator of distribution scale is the **sample range** or

$$R_n = X_{n:n} - X_{1:n} = \text{largest} - \text{smallest} \tag{1.24}$$

Two particularly interesting L-estimators that have immediate connection to the L-moments are Sen weighted mean and Gini mean difference statistics. These two statistics are described in the following two sections.

### *Sen Weighted Mean*

A special location statistic, which is based on the order statistics, is the **Sen weighted mean** (Sen, 1964) or the quantity $\mathscr{S}_{n,k}$. The $\mathscr{S}_{n,k}$ is a robust estimator (Jurečková and Picek, 2006, p. 69) of the mean of a distribution and is defined as

$$\mathscr{S}_{n,k} = \binom{n}{2k+1}^{-1} \sum_{i=1}^{n} \binom{i-1}{k} \binom{n-i}{k} X_{i:n} \tag{1.25}$$

where $X_{i:n}$ are the order statistics and $k$ is a weighting or trimming parameter. A sample version $\hat{\mathscr{S}}_{n,k}$ results when $X_{i:n}$ are replaced by their sample counterparts $x_{i:n}$. Readers should note that $\mathscr{S}_{n,0} = \mu = \overline{X}_n$ or the arithmetic `mean()`, and $\mathscr{S}_{n,k}$ is the sample `median()` if either $n$ is even and $k = (n/2) - 1$ or $n$ is odd and $k = (n-1)/2$.

USING R

The *lmomco* package provides support for $\hat{\mathscr{S}}_{n,k}$ through the `sen.mean()` function, which is demonstrated in example 1–9. In the example, a fake data set is set into `fake.dat`, and a "Sen" object `sen` is created. A list `sen` is returned by the `sen.mean()` function.

1–9

```
fake.dat <- c(123, 34, 4, 654, 37, 78)
# PART 1
sen <- sen.mean(fake.dat)
print(sen); mean(fake.dat) # These should be the same values
$sen
[1] 155
$source
[1] "sen.mean"

[1] 155
```

USING R

```
# PART 2
sen.mean(fake.dat, k=(length(fake.dat)/2) - 1); median(fake.dat)
# Again, these are the same values.
$sen
[1] 57.5
$source
[1] "sen.mean"

[1] 57.5
```

The first part of the example shows that by default $\hat{\mathscr{S}}_{n,0} = \mu$ (155 for the example), but the second part shows that $k$ can be chosen to yield the median (57.5 for the example). ◀

Finally, $\mathscr{S}_{n,k}$ is equivalent to the first symmetrically trimmed TL-moment (not yet introduced, $\lambda_1^{(k)}$). Let us demonstrate the numerical equivalency $\mathscr{S}_{n,k} = \lambda_1^{(k)}$ in example 1–10 by computing a two sample (two data point) trimming from each tail (side) of a Normal distribution having a $\mu = 100$ and $\sigma = 1$ or in moment-order listing: NOR$(100, 1)$. The magnitude of the difference between $\mathscr{S}_{n,k}$ and the first TL-moment for symmetrical trimming $k$ is zero.

1–10

```
fake.dat <- rnorm(20, mean=100) # generate a random sample
lmr <- TLmoms(fake.dat, trim=2) # compute trimmed L-moments
sen <- sen.mean(fake.dat, k=2)  # compute Sen mean
print(abs(lmr$lambdas[1] - sen$sen)) # should be zero
[1] 0
```

◀

### *Gini Mean Difference*

Another special statistic that also is based on order statistics, which is closely related to the second L-moment $\lambda_2$, is the **Gini mean difference** (Gini, 1912). The Gini mean difference $\mathscr{G}$ is a robust estimator (Jurečková and Picek, 2006, p. 64) of distribution scale or spread and is defined as respective population $\mathscr{G}$ and sample $\hat{\mathscr{G}}$ statistics as

$$\mathscr{G} = \mathrm{E}[X_{2:2} - X_{1:2}] \tag{1.26}$$

$$\hat{\mathscr{G}} = \frac{2}{n(n-1)} \sum_{i=1}^{n} (2i - n - 1)\, x_{i:n} \tag{1.27}$$

where $X_{i:n}$ are the order statistics, $x_{i:n}$ are the sample order statistics, and $n \geq 2$. The statistic $\mathscr{G}$ is a measure of the expected difference between two randomly drawn values from a distribution. Hence, the statistic is a measure of distribution scale or spread.

USING R

The *lmomco* package provides support for $\hat{\mathscr{G}}$ through the `gini.mean.diff()` function, which is demonstrated in example 1–11. In the example, a fake data set is set into `fake.dat`, a "Gini" object is created, and assigned to variable `gini`. A list `gini` is returned. The $\hat{\mathscr{G}}$ statistic is listed in `gini$gini` and the second sample L-moment ($\hat{\lambda}_2$) is listed in `gini$L2`. Thus, $\hat{\mathscr{G}} = 237$.

1–11

```
fake.dat <- c(123,34,4,654,37,78) # fake data
gini <- gini.mean.diff(fake.dat)  # from lmomco
str(gini) # ouput the list structure
List of 3
 $ gini  : num 237
 $ L2    : num 119
 $ source: chr "gini.mean.diff"
```

◀

By definition $\mathscr{G} = 2\lambda_2$ where $\lambda_2$ is the second L-moment. Example 1–12 computes the sample L-moments using the `lmoms()` function of *lmomco* and demonstrates the numerical equivalency of $\mathscr{G} = 2\lambda_2$ by the `print()` function outputting zero.

1–12

```
lmr <- lmoms(fake.dat) # compute L-moments from lmomco
print(abs(gini$gini/2 - lmr$lambdas[2])) # should be zero
[1] 0
```

◀

After reporting within discussion of order-based inference that "linear functions of the ordered sample values can form not only useful estimators but even optimal ones," Barrett (2004, p. 27) goes on to report that the quantity

$$V = \frac{1.7725}{n(n-1)} \sum_{i=1}^{n} (2i - n - 1) X_{[i:n]} \tag{1.28}$$

is "more easily calculated than the unbiased sample variance $[\hat{\sigma}^2]$, and for normal $X$ it is about 98 [percent] efficient relative to $[\hat{\sigma}^2]$ for all sample sizes." Barrett apparently has made a mistake on the units—the units of $V$ are not squared like those of variance. Therefore, a conclusion is made that $V^2 \approx \hat{\sigma}^2$ is what is meant. Emphasis is needed that these two statistics are both variance estimators.

There are many specific connections of eq. (1.28) to this presentation that are particularly interesting to document because Barrett (2004) makes no reference to L-moments, no reference to the Gini mean difference, and a single reference to L-estimators (Barrett, 2004, p. 122). The connections are:

- Eq. (1.28) is very similar to eq. (1.27): $1.7725 \times \hat{\mathscr{G}}/2 = V$;
- The Gini mean difference is related to the second L-moment $\lambda_2$ by $\mathscr{G} = 2\lambda_2$. Thus, $\lambda_2$ is related to $V$;
- The sample standard deviation is $\hat{\sigma} = \sqrt{\hat{\sigma}^2}$;

- In terms of L-moments, the standard deviation of the Normal distribution is $\sigma = \sqrt{\pi}\lambda_2$ by eq. (**??**); and
- The value $\sqrt{\pi} = 1.772454\ldots$, which has an obvious connection to eq. (1.28).

Barrett (2004) indicates that the "efficiency" of $V$ is "about 98 percent" for all sample sizes. Assuming that **relative efficiency** $RE$ by eq. (1.19) is meant, let us use R to test this claim. In example 1–13, the variance of $V$ and the familiar definition $\hat{\sigma}^2$ by the `var()` function are computed for a large sample size of $n = 2000$ for a very large number of simulations.

1–13
```
n <- 2000; nsim <- 200000
"Barrett" <- function(n) {
    gini <- gini.mean.diff(rnorm(n))$gini
    return((sqrt(pi)*gini/2)^2)
}
GiniVar    <- var(replicate(nsim, Barrett(n)    ))
ClassicVar <- var(replicate(nsim, var(rnorm(n)) ))
RE <- ClassicVar/GiniVar # relative efficiency
print(RE)
[1] 0.9738433
# Barrett (2004, p. 27) reports 98 percent.
```

The example estimates $RE \approx 0.97$, which is acceptably close to the value reported by Barrett. Therefore, the computed value is consistent with Barrett's value. Barrett also states that this $RE$ holds for all sample sizes. This conclusion is tested in example 1–14 for a sample size of $n = 10$.

1–14
```
n <- 10
GiniVar    <- var(replicate(nsim, Barrett(n)    ))
ClassicVar <- var(replicate(nsim, var(rnorm(n)) ))
RE <- ClassicVar/GiniVar # relative efficiency
print(RE)
[1] 0.8752343
```

Example 1–14 estimates $RE_{n=10} \approx 0.88$, which is clearly at odds with Barrett's statement—$RE$ is in fact a function of sample size. Another experiment shows that $RE_{n=20} \approx 0.93$. ◀

## 1.4 Summary

In this chapter, a special class of statistics based on order samples or the order statistics is formally introduced, and 14 code examples are provided. The primary results are an expression for the expectation of an order statistic, the Sen weighted mean, and Gini mean difference. Foreshadowing, the L-moments and TL-moments, the connections between these and the Sen weighted mean and Gini mean difference are shown using R.

# References


Asquith, W.H., 2008, *lmomco*—L-moments, Trimmed L-moments, L-comoments, and Many Distributions: R package version 0.95, dated August 23, 2008, initial package release January 31, 2006, `www.cran.r-project.org/package=lmomco`.

Baclawski, Kenneth, 2008, Introduction to probability with R: Boca Raton, Fla., Chapman and Hall/CRC, ISBN 978–1–4200–6521–3, 363 p.

Barrett, Vic, 2004, Environmental statistics—Methods and applications: Chichester, West Sussex, England, John Wiley, ISBN 0–471–48971–9, 293 p.

David, H.A., 1981, Order statistics, 2nd ed.: New York, John Wiley, ISBN 0–471–02723–5, 360 p.

Gilchrist, W.G., 2000, Statistical modelling with quantile functions: Boca Raton, Fla., Chapman and Hall/CRC, ISBN 1–58488–174–7, 320 p.

Gini, C., 1912, Variabilità e mutabilità, contributo allo studio delle distribuzioni e delle relazione statistiche: Studi Economico-Giuridici della Reale Università di Cagliari, v. 3, pp. 3–159.

Hosking, J.R.M., 1990, L-moments—Analysis and estimation of distributions using linear combinations or order statistics: Journal of Royal Statistical Society, series B, v. 52, no. 1, pp. 105–124.

Hosking, J.R.M., 2006, On the characterization of distributions by their L-moments: Journal of Statistical Planning and Inference v. 136, no. 1, pp. 193–198.

Hosking, J.R.M., 2008, *lmom*—L-moments: R package version 1.0, dated July 3, 2008, initial package release July 3, 2008, `www.cran.r-project.org/package=lmom`.

Hosking, J.R.M., and Wallis, J.R., 1997, Regional frequency analysis—An approach based on L-moments: Cambridge, Cambridge University Press, ISBN 0–521–43045–3, 224 p.

Jurečková, J., and Picek, J., 2006, Robust statistical methods with R: Boca Raton, Fla., Chapman and Hall/CRC, ISBN 1–58488–454–1, 197 p.

Karvanen, J., 2007, *Lmoments*—L-moments and Quantile Mixtures: R package version 1.1–2, dated February 5, 2007, initial package release October 12, 2005, `www.cran.r-project.org/package=Lmoments`.

Ribatet, Mathieu, 2007, *POT*—Generalized Pareto and peaks over threshold: R package version 1.0–8, dated July 3, 2008, initial package release September 6, 2005, `www.cran.r-project.org/package=POT`.

Ribatet, Mathieu, 2007, *RFA*—Regional Frequency Analysis: R package version 0.0–7, dated July 3, 2007, initial package release September 14, 2005, `www.cran.r-project.org/package=RFA`.

Rizzo, M.L., 2008, Statistical computing with R: Boca Raton, Fla., Chapman and Hall/CRC, ISBN 978–1–58488–545–0, 399 p.

Sen, P.K., 1964, On some properties of the rank-weighted means: Journal Indian Society of Agricultural Statistics: v. 16, pp. 51–61.

Stedinger, J.R., Vogel, R.M., and Foufoula-Georgiou, E., 1993, Frequency analysis of extreme events, in Handbook of Hydrology, chapter 18, editor-in-chief D.A. Maidment: New York, McGraw-Hill, ISBN 0–07–039732–5.

Ugarte, M.D., Militino, A.F., and Arnhold, A.T., 2008, Probability and statistics with R: Boca Raton, Fla., Chapman and Hall/CRC, ISBN 978–1–58488–891–8, 700 p.