RC 20525 (90933) 8/5/96, revised 7/25/05
Mathematics 33 pages

Research Report

Fortran routines for use with the method of *L*-moments

Version 3.04

J. R. M. Hosking

IBM Research Division
T. J. Watson Research Center
Yorktown Heights, NY 10598

Research Division
Almaden • Austin • Beijing • Haifa • T.J. Watson • Tokyo • Zurich

Fortran routines for use with the method of *L*-moments Version 3.04

J. R. M. Hosking

IBM Research Division
P. O. Box 218
Yorktown Heights, NY 10598

Abstract. Hosking (*J. R. Statist. Soc. B*, 1990) and Hosking and Wallis (*Regional frequency analysis: an approach based on L-moments*, Cambridge University Press, 1997) have described *L*-moments and *L*-moment ratios, quantities useful in the summarization and estimation of probability distributions. This report contains details of Fortran routines that should facilitate the use of *L*-moment-based methods.

The routines described in this Research Report are available on the electronic software repository StatLib, at the Internet location `http://lib.stat.cmu.edu/general/lmoments`.

Contents

Index of routines

Introduction

Probability weighted moments of a random variable X with cumulative distribution function F were defined by Greenwood et al. (1979) to be the quantities

$$M_{p,r,s} = E\left[X^p\{F(X)\}^r\{1-F(X)\}^s\right] .$$

Particularly useful special cases are the probability weighted moments

$$\alpha_r = M_{1,0,r} = E[X\{1-F(X)\}^r]$$

and

$$\beta_r = M_{1,r,0} = E[X\{F(X)\}^r] .$$

Hosking (1990) defined "L-moments" to be the quantities

$$\lambda_r = E[XP^*_{r-1}\{F(X)\}]$$

where $P^*_r(\,.\,)$ is the rth shifted Legendre polynomial. L-moments and probability weighted moments are related by

$$\lambda_{r+1} = \sum_{k=0}^{r} p^*_{r,k}\beta_k \tag{1}$$

where

$$p^*_{r,k} = (-1)^{r-k}\binom{r}{k}\binom{r+k}{k} .$$

L-moment ratios are the quantities

$$\tau_r = \lambda_r/\lambda_2 .$$

The foregoing quantities are defined for a probability distribution, but in practice must often be estimated from a finite sample. Let $x_{1:n} \le x_{2:n} \le \ldots \le x_{n:n}$ be the ordered sample. Let

$$a_r = n^{-1}\sum_{j=1}^{n} \frac{(n-j)(n-j-1)\ldots(n-j-r+1)}{(n-1)(n-2)\ldots(n-r)}\, x_{j:n} ,$$

$$b_r = n^{-1}\sum_{j=1}^{n} \frac{(j-1)(j-2)\ldots(j-r)}{(n-1)(n-2)\ldots(n-r)}\, x_{j:n} , \tag{2}$$

$$\ell_{r+1} = \sum_{k=0}^{r} p^*_{r,k} b_k . \tag{3}$$

Then a_r, b_r and ℓ_r are unbiased estimators of α_r, β_r and λ_r respectively. The estimator $t_r = \ell_r/\ell_2$ of τ_r is consistent but not unbiased. Landwehr et al. (1979a, 1979b) also considered "plotting-position" estimators of probability weighted moments. Let $p_{j:n}$ be a plotting

position, i.e. a distribution-free estimator of $F(x_{j:n})$. Then α_r, β_r, λ_r and τ_r are estimated respectively by

$$\tilde{\alpha}_r = n^{-1} \sum_{j=1}^{n} (1 - p_{j:n})^r \, x_{j:n} \,,$$

$$\tilde{\beta}_r = n^{-1} \sum_{j=1}^{n} p_{j:n}^r \, x_{j:n} \,, \tag{4}$$

$$\tilde{\lambda}_r = n^{-1} \sum_{j=1}^{n} P_{r-1}^*(p_{j:n}) \, x_{j:n} \,, \tag{5}$$

$$\tilde{\tau}_r = \tilde{\lambda}_r / \tilde{\lambda}_2 \,.$$

Plotting-position estimators are consistent, but can have large bias and cannot be generally recommended (Hosking and Wallis, 1995; Hosking and Wallis, 1997, sec. 2.8).

Hosking (1990) and Hosking and Wallis (1997, chap. 2) give expositions of the theory of *L*-moments and *L*-moment ratios. Hosking and Wallis (1997, Appendix) give, for many distributions in common use, expressions for the *L*-moments of the distributions and algorithms for estimating the parameters of the distributions by equating sample and population *L*-moments (the "method of *L*-moments"). This report contains details of Fortran routines that should facilitate the use of *L*-moment-based methods. The following routines are included.

- For each of the eleven distributions listed in Table 1 on page 5, routines to evaluate the cumulative distribution function and quantile function of the distribution, to calculate the *L*-moments given the parameters and to calculate the parameters given the low-order *L*-moments.
- Given a sample of data, routines to calculate estimates of the probability weighted moments and *L*-moments of the distribution from which the sample was drawn.
- Routines useful in connection with index-flood procedures for regional frequency analysis (Hosking and Wallis, 1997). Three routines are concerned with cluster analysis, and do not explicitly involve *L*-moments. One routine calculates a weighted average of *L*-moment ratios estimated from data measured at different sites, an essential part of *L*-moment-based index-flood procedures such as the regional *L*-moment algorithm described in Hosking and Wallis (1997, sec. 6.2). One other routine calculates diagnostic statistics useful in deciding whether a set of samples may be regarded as having similar frequency distributions: these statistics are described more fully in Hosking and Wallis (1997, chaps. 3–5).
- Programs to apply the regional *L*-moment algorithm and to calculate the diagnostic statistics for a regional data set.
- Auxiliary routines called by some of the earlier routines: mathematical functions, a pseudo-random number generator and a routine to sort a data array into ascending order.

The routines are written in Fortran-77. The routines aim for accuracy of about 8 significant digits. For this reason double-precision arithmetic is used throughout. Single precision may

be adequate on machines that use 60 bits to represent a single-precision floating-point number.

Caveat: Statements about the accuracy of individual routines are not unconditional guarantees, but are valid over wide ranges of values of the input parameters including those most frequently encountered in practice. Decreased accuracy is particularly likely when the shape parameter of the distribution is close to zero (less than 10^{-4} in absolute value), when the shape parameter is close to the end of the range of its permissible values, or is close to a value at which the mean of the distribution does not exist, or when the shape parameter is large and positive.

Routines for specific distributions

Routines are provided for the eleven distributions listed in Table 1. The following routines are provided for each distribution.

`FUNCTION CDFxxx`	The cumulative distribution function of the distribution.
`FUNCTION QUAxxx`	The quantile function (inverse cumulative distribution function) of the distribution.
`SUBROUTINE LMRxxx`	Calculates the L-moment ratios of the distribution given its parameters.
`SUBROUTINE PELxxx`	Calculates the parameters of the distribution given its L-moments. When the L-moments are the sample L-moments of a set of data, then the resulting parameters are of course the "method of L-moments" estimates of the parameters.

Here `xxx` is the 3-letter code used to identify the distribution, as given in Table 1. For example the cumulative distribution function of the gamma distribution is `FUNCTION CDFGAM`.

Most of these distributions are standard and are as defined in Hosking and Wallis (1997, Appendix).

The generalized Normal distribution combines lognormal distributions (which have positive skewness), reflected lognormal distributions (which have negative skewness) and the Normal distribution (which has zero skewness). The three-parameter lognormal distribution, with cumulative distribution function

$$F(x) = \Phi[\{\log(x-\zeta) - \mu\}/\sigma]\,,$$

is a generalized Normal distribution with parameters $k = -\sigma$, $\alpha = \sigma e^{\mu}$, $\xi = \zeta + e^{\mu}$.

The Pearson type III distribution combines gamma distributions (which have positive skewness), reflected gamma distributions (which have negative skewness) and the Normal distribution (which has zero skewness). We parametrize the Pearson type III distribution by its first three (conventional) moments: the mean μ, the standard deviation σ and the skewness γ. The relationship between these parameters and those of the gamma distribution is as follows. Let X be a random variable with a Pearson type III distribution with parameters μ, σ and γ. If $\gamma > 0$, then $X - \mu + 2\sigma/\gamma$ has a gamma distribution with parameters $\alpha = 4/\gamma^2$, $\beta = \sigma\gamma/2$. If $\gamma = 0$, then X has a Normal distribution with mean μ and standard deviation σ. If $\gamma < 0$, then $-X + \mu - 2\sigma/\gamma$ has a gamma distribution with parameters $\alpha = 4/\gamma^2$, $\beta = |\sigma\gamma/2|$.

Table 1: Distributions for which L-moment subroutines are provided.

Distribution	Code	Number of parameters	Parameters	$F(x),\ x(F)$
Exponential	`EXP`	2	$\xi\ \alpha$	$F = 1 - \exp\{-(x-\xi)/\alpha\}$ $x = \xi - \alpha \log(1-F)$
Gamma	`GAM`	2	$\alpha\ \beta$	$F = G(x/\beta, \alpha)$ $x(F)$ not explicitly defined
Generalized extreme-value	`GEV`	3	$\xi\ \alpha\ k$	$F = \exp[-\{1-k(x-\xi)/\alpha\}^{1/k}]$ $x = \xi + \alpha\{1-(-\log F)^k\}/k$
Generalized logistic	`GLO`	3	$\xi\ \alpha\ k$	$F = 1/[1+\{1-k(x-\xi)/\alpha\}^{1/k}]$ $x = \xi + \alpha[1-\{(1-F)/F\}^k]/k$
Generalized Normal	`GNO`	3	$\xi\ \alpha\ k$	$F = \Phi[-k^{-1}\log\{1-k(x-\xi)/\alpha\}]$ $x(F)$ not explicitly defined
Generalized Pareto	`GPA`	3	$\xi\ \alpha\ k$	$F = 1-\{1-k(x-\xi)/\alpha\}^{1/k}$ $x = \xi + \alpha\{1-(1-F)^k\}/k$
Gumbel	`GUM`	2	$\xi\ \alpha$	$F = \exp[-\exp\{-(x-\xi)/\alpha\}]$ $x = \xi + \alpha\log(-\log F)$
Kappa	`KAP`	4	$\xi\ \alpha\ k\ h$	$F = [1-h\{1-k(x-\xi)/\alpha\}^{1/k}]^{1/h}$ $x = \xi + \alpha[1-\{(1-F)^h/h\}^k]/k$
Normal	`NOR`	2	$\mu\ \sigma$	$F = \Phi\{(x-\mu)/\sigma\}$ $x(F)$ not explicitly defined
Pearson type III	`PE3`	3	$\mu\ \sigma\ \gamma$	$F = G\big((x-\mu+2\sigma/\gamma)/\lvert\frac{1}{2}\sigma\gamma\rvert, 4/\gamma^2\big), \quad \gamma > 0$ $F = 1-G\big(-(x-\mu+2\sigma/\gamma)/\lvert\frac{1}{2}\sigma\gamma\rvert, 4/\gamma^2\big),\ \gamma < 0$ $x(F)$ not explicitly defined
Wakeby	`WAK`	5	$\xi\ \alpha\ \beta\ \gamma\ \delta$	$F(x)$ not explicitly defined $x = \xi + \frac{\alpha}{\beta}\{1-(1-F)^\beta\} - \frac{\gamma}{\delta}\{1-(1-F)^{-\delta}\}$

$G(x,\alpha) = \{\Gamma(\alpha)\}^{-1}\int_0^x t^{\alpha-1}e^{-t}\,dt$ is the incomplete gamma integral.

$\Phi(x) = (2\pi)^{-1/2}\int_{-\infty}^x \exp(-t^2/2)\,dt$ is the standard Normal cumulative distribution function.

Cumulative distribution functions

```
DOUBLE PRECISION FUNCTION CDFEXP(X,PARA)
DOUBLE PRECISION FUNCTION CDFGAM(X,PARA)
DOUBLE PRECISION FUNCTION CDFGEV(X,PARA)
DOUBLE PRECISION FUNCTION CDFGLO(X,PARA)
DOUBLE PRECISION FUNCTION CDFGNO(X,PARA)
DOUBLE PRECISION FUNCTION CDFGPA(X,PARA)
DOUBLE PRECISION FUNCTION CDFGUM(X,PARA)
DOUBLE PRECISION FUNCTION CDFKAP(X,PARA)
DOUBLE PRECISION FUNCTION CDFNOR(X,PARA)
DOUBLE PRECISION FUNCTION CDFPE3(X,PARA)
DOUBLE PRECISION FUNCTION CDFWAK(X,PARA)
```

Formal parameters

`X` [input; double precision] The argument of the function.

`PARA` [input; double-precision array of length equal to the number of parameters of the distribution] The parameters of the distribution.

Numerical method

`CDFGAM` and `CDFPE3` are calculated from the incomplete gamma integral, auxiliary routine `GAMIND`. `CDFGNO` and `CDFNOR` are calculated from the error function, supplied as the auxiliary routine `DERF`. Routine `CDFWAK`, given an argument x_0 of the Wakeby cumulative distribution function, solves the equation $x(F) = x_0$ for F using Halley's method, the second-order analogue of Newton-Raphson iteration (Churchhouse, 1981, pp. 151-152; Huh, 1986). The other cumulative distribution functions have explicit analytical expressions, given in Table 1: the routines straightforwardly evaluate these expressions.

Accuracy

In principle (but see the caveat on page 3) all routines are accurate to at least 8 decimal places.

Quantile functions

```
DOUBLE PRECISION FUNCTION QUAEXP(F,PARA)
DOUBLE PRECISION FUNCTION QUAGAM(F,PARA)
DOUBLE PRECISION FUNCTION QUAGEV(F,PARA)
DOUBLE PRECISION FUNCTION QUAGLO(F,PARA)
DOUBLE PRECISION FUNCTION QUAGNO(F,PARA)
DOUBLE PRECISION FUNCTION QUAGPA(F,PARA)
DOUBLE PRECISION FUNCTION QUAGUM(F,PARA)
DOUBLE PRECISION FUNCTION QUAKAP(F,PARA)
DOUBLE PRECISION FUNCTION QUANOR(F,PARA)
DOUBLE PRECISION FUNCTION QUAPE3(F,PARA)
DOUBLE PRECISION FUNCTION QUAWAK(F,PARA)
```

Formal parameters

`F` [input; double precision] The argument of the function.

`PARA` [input; double-precision array of length equal to the number of parameters of the distribution] The parameters of the distribution.

Numerical method

Routine `QUAGAM`, given an argument F_0 of the Gamma quantile function, solves the equation $G(x, \alpha) = F_0$ for x using Newton-Raphson iteration. Routines `QUAGNO` and `QUANOR` use the quantile function of the standard Normal distribution, supplied as the auxiliary routine `QUASTN`, and transform it to get the required quantile function. The other quantile functions have explicit analytical expressions, given in Table 1: the routines straightforwardly evaluate these expressions.

Accuracy

Let F be the argument of the quantile function. In principle (but see the caveat on page 3) each routine returns a quantile $x(F_0)$ corresponding to an argument F_0 that satisfies $|F - F_0| \leq 10^{-8}$. In other words, the error of the approximation is equivalent to a perturbation of no more than 10^{-8} in the argument of the function.

L-moment ratios

```
SUBROUTINE LMREXP(PARA,XMOM,NMOM)
SUBROUTINE LMRGAM(PARA,XMOM,NMOM)
SUBROUTINE LMRGEV(PARA,XMOM,NMOM)
SUBROUTINE LMRGLO(PARA,XMOM,NMOM)
SUBROUTINE LMRGNO(PARA,XMOM,NMOM)
SUBROUTINE LMRGPA(PARA,XMOM,NMOM)
SUBROUTINE LMRGUM(PARA,XMOM,NMOM)
SUBROUTINE LMRKAP(PARA,XMOM,NMOM)
SUBROUTINE LMRNOR(PARA,XMOM,NMOM)
SUBROUTINE LMRPE3(PARA,XMOM,NMOM)
SUBROUTINE LMRWAK(PARA,XMOM,NMOM)
```

Formal parameters

`PARA` [input; double-precision array of length equal to the number of parameters of the distribution] The parameters of the distribution.

`XMOM` [output; double-precision array of length `NMOM`] The *L*-moment ratios of the distribution, in the order λ_1, λ_2, τ_3, τ_4, ...

`NMOM` [input; integer] The number of *L*-moments to be found. In routines `LMRGAM` and `LMRPE3`, `NMOM` may be at most 4; in the other routines it may be at most 20.

Numerical method

`LMREXP` *L*-moments and *L*-moment ratios are calculated using the expressions in Hosking and Wallis (1997, Appendix A.2).

`LMRGAM` *L*-moments and *L*-moment ratios are calculated using expressions in Hosking and Wallis (1997, Appendix A.9): analytic expressions for λ_1 and λ_2, rational-function approximations for τ_3 and τ_4.

`LMRGEV` Probability weighted moments β_k are calculated using eq. (9) of Hosking et al. (1985). Each λ_k is then calculated from β_k and the λ_j, $j = 1, ..., k-1$, using the formula

$$\lambda_{k+1} = \binom{2k}{k}\beta_k - \sum_{j=1}^{k} \frac{(2j-1)(2k)!}{(k+j)!\,(k+1-j)!}\,\lambda_j\,,$$

easily proved by induction using (1).

`LMRGLO` The first two *L*-moments are calculated using the expressions in Hosking and Wallis (1997, Appendix A.7). Higher-order *L*-moment ratios are expressed as polynomials in the shape parameter k.

`LMRGNO` The first two *L*-moments are calculated using the expressions in Hosking and Wallis (1997, Appendix A.8). For $k = 0$, higher-order *L*-moment ratios are those

of the Normal distribution. For $k \neq 0$, higher-order L-moment ratios are found by numerical evaluation of the integral

$$\int_{-\infty}^{\infty} \exp\{-(x + k/\sqrt{2})^2\} P_{r-1}(\operatorname{erf} x)\, dx\,,$$

where $P_{r-1}(\,.\,)$ is the $(r-1)$th Legendre polynomial and erf is the error function; this integral is, apart from a linear transformation, the rth L-moment of the distribution.

`LMRGPA` The routine evaluates the explicit formulas given by Hosking and Wallis (1997, Appendix A.5). Higher-order L-moments (λ_r, $r \geq 5$) are computed from the general expression

$$\lambda_r = \frac{\alpha\Gamma(1+k)\Gamma(r-1-k)}{\Gamma(1-k)\Gamma(r+1+k)}\,,$$

which can be derived from the expression for α_r given by Hosking and Wallis (1987, p. 341).

`LMRGUM` The first two L-moments are calculated using the expressions in Hosking and Wallis (1997, Appendix A.3). Higher-order L-moment ratios are the same for all Gumbel distributions. They are initialized in a data statement and copied into array `XMOM` as required.

`LMRKAP` Probability weighted moments β_k are calculated using eq. (12) of Hosking (1994). The λ_k are calculated from the β_k as in routine `LMRGEV`.

`LMRNOR` The first two L-moments are calculated using the expressions in Hosking and Wallis (1997, Appendix A.4). Higher-order L-moment ratios are the same for all Normal distributions. They are initialized in a data statement and copied into array `XMOM` as required.

`LMRPE3` L-moments are calculated from those of the gamma distribution, of which the Pearson type III is a reparametrization. The numerical method is the same as that of routine `LMRGAM`.

`LMRWAK` The routine evaluates the explicit formulas given by Hosking and Wallis (1997, Appendix A.11). Higher-order L-moments (λ_r, $r \geq 5$) are computed from the general expression

$$\lambda_r = \frac{\alpha\Gamma(1+\beta)\Gamma(r-1-\beta)}{\Gamma(1-\beta)\Gamma(r+1+\beta)} + \frac{\gamma\Gamma(1-\delta)\Gamma(r-1+\delta)}{\Gamma(1+\delta)\Gamma(r+1-\delta)}\,,$$

which can be derived from the expression for α_r given by Kotz et al. (1988, p. 514).

Accuracy

In principle (but see the caveat on page 3) all routines are accurate to at least 8 significant digits for λ_1 and λ_2. For higher-order L-moment ratios, routines `LMREXP`, `LMRGLO`, `LMRGPA`, `LMRGUM`, `LMRNOR` and `LMRWAK` are accurate to at least 8 decimal places. Routines `LMRGAM`, `LMRGNO` and `LMRPE3` are accurate to 6 decimal places. Routines `LMRGEV` and `LMRKAP` become less and less accurate as the order of the L-moment ratio increases; their accuracy for τ_r is at least 8 decimal places if $r \leq 12$, at least 4 decimal places if $r \leq 18$.

Parameters

```
SUBROUTINE PELEXP(XMOM,PARA)
SUBROUTINE PELGAM(XMOM,PARA)
SUBROUTINE PELGEV(XMOM,PARA)
SUBROUTINE PELGLO(XMOM,PARA)
SUBROUTINE PELGNO(XMOM,PARA)
SUBROUTINE PELGPA(XMOM,PARA)
SUBROUTINE PELGUM(XMOM,PARA)
SUBROUTINE PELKAP(XMOM,PARA,IFAIL)
SUBROUTINE PELNOR(XMOM,PARA)
SUBROUTINE PELPE3(XMOM,PARA)
SUBROUTINE PELWAK(XMOM,PARA,IFAIL)
```

Formal parameters

`XMOM` [input; double-precision array of length equal to the number of parameters of the distribution] The L-moment ratios λ_1, λ_2, τ_3, τ_4, ...

`PARA` [input; double-precision array of length equal to the number of parameters of the distribution] The parameters of the distribution.

`IFAIL` [output; integer] Used only by routines `PELKAP` and `PELWAK`. For explanation see "Numerical method" below.

Numerical method

Except where indicated below, the routines evaluate expressions for the parameters in terms of the L-moments given in the appendix of Hosking and Wallis (1997).

`PELGAM` Rational approximation is used to express α, the shape parameter of the gamma distribution, as a function of the L-CV $\tau = \lambda_2/\lambda_1$.

`PELGEV` For $-0.8 \le \tau_3 < 1$, the shape parameter k is estimated from rational-function approximations given by Donaldson (1996). This approximation has absolute accuracy better than 3×10^{-7}. If $\tau_3 < -0.8$, Donaldson's approximation is further refined by Newton-Raphson iteration until similar accuracy is achieved. Parameters α and ξ are then calculated as in Hosking and Wallis (1997, Appendix A.6).

`PELKAP` The method is as described by Hosking (1994). Newton-Raphson iteration is used to solve the equations that express τ_3 and τ_4 as functions of k and h, and α and ξ are calculated as functions of λ_1, λ_2, k and h. To ensure a 1-1 relationship between parameters and L-moments, the parameter space is restricted as in Hosking (1994, eq. (13)):

$$k > -1; \qquad \text{if } h < 0 \text{ then } hk > -1; \qquad h > -1; \qquad k + 0.725h > -1\,.$$

On exit, parameter IFAIL is set as follows.

0 Successful exit.

1 L-moments invalid: $\lambda_2 \leq 0$, or τ_3 and τ_4 do not lie within the set of feasible values given in Hosking and Wallis (1997, eq. (2.45)).

2 Routine was called with $\tau_4 \geq (1 + 5\tau_3^2)/6$. This implies that the L-moments are inconsistent with a kappa distribution with parameters restricted as above.

3 Newton-Raphson iteration did not converge.

4 Newton-Raphson iteration reached a point from which no further progress could be made.

5 Newton-Raphson iteration encountered numerical difficulties: overflow would have been likely to occur.

6 Iteration for h and k converged, but overflow would have occurred while calculating ξ and α.

PELWAK The basic method is that of Landwehr et al. (1979b), but the distribution is differently parametrized and the calculations are expressed in terms of L-moments rather than probability weighted moments. The computations are given by Hosking and Wallis (1997, Appendix A.11). First a solution is sought in which all five parameters are estimated, as functions of the first five L-moments. If no solution is found, ξ is set equal to zero and a solution is sought in which the other four parameters are estimated as functions of the first four L-moments. If this too is unsuccessful, then a generalized Pareto distribution is fitted instead, using the first three L-moments and the same method as routine PELGPA.

On exit, parameter IFAIL is set as follows.

0 Successful exit; all five parameters estimated.

1 Solution could only be found by setting $\xi = 0$.

2 Solution could only be found by fitting a generalized Pareto distribution.

3 L-moments invalid: $\lambda_2 \leq 0$, or $|\tau_r| \geq 1$ for some $r \geq 3$.

Accuracy

For shape parameters, routines PELGLO and PELGPA are accurate to at least 8 decimal places; PELGEV is accurate to within 3×10^{-7}; PELGNO has relative accuracy better than 2.5×10^{-6}; PELKAP is accurate to within 5×10^{-6}; PELGAM and PELPE3 have relative accuracy better than 5×10^{-5}. For any of these routines, let ϵ be the relative error of estimation of the shape parameter; for routines PELGUM and PELNOR, let $\epsilon = 10^{-8}$; then the estimate of the scale parameter α (or β, or σ) has relative error approximately ϵ, and the estimate of the location parameter ξ has absolute error approximately $\epsilon\alpha$.

Routine PELWAK is usually accurate to within 10^{-6}, but can be much less accurate for certain combinations of input parameters. In such cases, however, the quantiles of the fitted distribution are usually still reliable to within an error equivalent to a perturbation of less than 10^{-6} in the argument of the quantile function.

Routines for computing sample *L*-moments

Three routines are provided for computing sample *L*-moments and probability weighted moments of a data sample.

`SUBROUTINE SAMLMR`	Calculates the sample *L*-moment ratios of a data set, indirectly via the probability weighted moments.
`SUBROUTINE SAMLMU`	Calculates the "unbiased" sample *L*-moment ratios of a data set by a more direct method.
`SUBROUTINE SAMPWM`	Calculates the sample probability weighted moments of a data set.

```
SUBROUTINE SAMLMR(X,N,XMOM,NMOM,A,B)
```

Purpose

Calculates the sample L-moment ratios of a data set.

Formal parameters

`X` [input; double-precision array of length `N`] The data, in ascending order.

`N` [input; integer] Sample size.

`XMOM` [output; double-precision array of length `NMOM`] The sample L-moment ratios, in the order ℓ_1, ℓ_2, t_3, t_4, ...

`NMOM` [input; integer] The number of L-moments to be found. At most min(`N`, 20).

`A,B` [input; double precision] Parameters of plotting position. If `A` and `B` are both set to zero, "unbiased" estimates of L-moments are found. Otherwise plotting-position estimates are found, based on the plotting position $p_{j:n} = (j + \mathtt{A})/(n + \mathtt{B})$ for the jth smallest of n observations. For example, `A=-0.35D0` and `B=0.0D0` yields the estimates recommended by Hosking et al. (1985) for use with the generalized extreme-value distribution.

Numerical method

The routine explicitly computes probability weighted moments using (2) or (4), and from them the sample L-moments using (3) or (5).

`SUBROUTINE SAMLMU(X,N,XMOM,NMOM)`

Purpose

Calculates the "unbiased" sample L-moment ratios of a data set.

Formal parameters

`X` [input; double-precision array of length `N`] The data, in ascending order.

`N` [input; integer] Sample size.

`XMOM` [output; double-precision array of length `NMOM`] The sample L-moment ratios, in the order ℓ_1, ℓ_2, t_3, t_4, ...

`NMOM` [input; integer] The number of L-moments to be found. At most min(`N`, 100).

Numerical method

From (2) and (3), ℓ_r is a linear combination of the ordered sample values $x_{1:n}, \ldots, x_{n:n}$, and we can write

$$\ell_r = n^{-1} \sum_{j=1}^{n} w_{j:n}^{(r)} \, x_{j:n} \, .$$

The weights $w_{j:n}^{(r)}$ are related to the "discrete Legendre polynomials" defined by Neuman and Schonbach (1974). In Neuman and Schonbach's notation, the weight $w_{j:n}^{(r)}$ is the discrete Legendre polynomial $(-1)^r P_{r-1}(j-1, n-1)$. The weights can be computed as $w_{j:n}^{(1)} = 1$, $w_{j:n}^{(2)} = 2(j-1)/(n-1) - 1$, and, for $r \geq 3$,

$$(r-1)(n-r+1)w_{j:n}^{(r)} = (2r-3)(2j-n-1)w_{j:n}^{(r-1)} - (r-2)(n+r-2)w_{j:n}^{(r-2)};$$

this recurrence relation is equivalent to equation (9) of Neuman and Schonbach (1974). Routine `SAMLMU` evaluates sample L-moments using this recurrence, and achieves additional efficiency by making use of the symmetry in the weights, $w_{j:n}^{(r)} = (-1)^{r-1} w_{n+1-j:n}^{(r)}$.

This routine is generally equivalent to `SAMLMR`, but is particularly suitable when high-order L-moment ratios are to be calculated. Routine `SAMLMU` is accurate to 8 decimal places for about the first 50 L-moment ratios, whereas `SAMLMR` achieves similar accuracy for only the first 12 L-moment ratios. The routines require approximately the same amount of computing time, but `SAMLMU` can be a little slower than `SAMLMR` when `NMOM` is large or odd.

```
SUBROUTINE SAMPWM(X,N,XMOM,NMOM,A,B,KIND)
```

Purpose

Calculates the sample probability weighted moments of a data set.

Formal parameters

`X` [input; double-precision array of length `N`] The data, in ascending order.

`N` [input; integer] Sample size.

`XMOM` [output; double-precision array of length `NMOM`] The sample probability weighted moments, in the order (if `KIND=1`) a_0, a_1, ..., or (if `KIND=2`) b_0, b_1, ...

`NMOM` [input; integer] The number of probability weighted moments to be found. At most min(`N`, 20).

`A,B` [input; double precision] Parameters of plotting position. If `A` and `B` are both set to zero, unbiased estimates of probability weighted moments are found. Otherwise plotting-position estimates are found, based on the plotting position $p_{j:n} = (j + \mathtt{A})/(n + \mathtt{B})$ for the jth smallest of n observations. For example, `A=-0.35D0` and `B=0.0D0` yields the estimates recommended by Hosking et al. (1985) for use with the generalized extreme-value distribution.

`KIND` [input; integer] Indicates the kind of probability weighted moments to be found. If `KIND=1`, the routine estimates the probability weighted moments $\alpha_r = E[X\{1 - F(X)\}^r]$. If `KIND=2`, the routine estimates the probability weighted moments $\beta_r = E[X\{F(X)\}^r]$.

Routines for regional frequency analysis

These routines implement the methods described by Hosking and Wallis (1997) for regional frequency analysis, which combines the L-moments of data samples from different measuring sites to achieve improved estimates of the frequency distribution at each site.

Three routines provide facilities for cluster analysis, and do not explicitly involve L-moments. Two routines implement the estimation and testing methods from Hosking and Wallis (1997, chaps. 3–6). Four driver programs are also provided: these illustrate the use of the regional frequency analysis routines by applying them to data sets used by Hosking and Wallis (1997).

`SUBROUTINE CLUAGG` Performs cluster analysis by one of several agglomerative hierarchical methods: single-link, complete-link, and Ward's procedure.

`SUBROUTINE CLUINF` Obtains information about clusters arising from agglomerative hierarchical clustering.

`SUBROUTINE CLUKM` Performs cluster analysis by the K-means algorithm.

`SUBROUTINE REGLMR` Calculates regional weighted averages of L-moment ratios.

`SUBROUTINE REGTST` Calculates statistics useful in regional frequency analysis. The statistics are based on the at-site and regional average sample L-moments.

- Discordancy measures for individual sites in a region. These are intended to assess whether the sample L-moments at a site are markedly different from the regional average.
- Heterogeneity measures for a region. These are intended to assess whether the between-site variation in the sample L-moments is consistent with what would be expected of a homogeneous region.
- Goodness-of-fit measures, based on the regional average L-moment ratios. These are intended to assess whether any of five candidate distributions gives an adequate fit to the data.

The statistics are described in more detail by Hosking and Wallis (1997, chaps. 3–5).

`PROGRAM XCLUST` Program to illustrate the use of the cluster analysis routines.

`PROGRAM XFIT` Program to illustrate the use of routine `REGLMR`. The program performs frequency analysis of a regional data set using an index-flood procedure and the method of L-moments.

`PROGRAM XTEST` Program to illustrate the use of routine `REGTST`.

`PROGRAM XSIM` Program to illustrate the use of simulation to find the accuracy of estimates obtained from regional frequency analysis.

```
SUBROUTINE CLUAGG(METHOD,X,NX,N,NATT,MERGE,DISP,IWORK,WORK,NW)
```

Purpose

Performs cluster analysis by one of several agglomerative hierarchical methods: single-link, complete-link, and Ward's procedure.

Formal parameters

`METHOD` [input; integer] Defines which clustering method is used. Should be set to `1` for single-link clustering, `2` for complete-link clustering, `3` for Ward's procedure.

`X` [input; double-precision array of dimension `(NX,NATT)`] `X(I,J)` should contain the `J`th attribute for the `I`th data point.

`NX` [input; integer] The first dimension of array `X`, as declared in the calling program.

`N` [input; integer] Number of data points.

`NATT` [input; integer] Number of attributes for each data point.

`MERGE` [output; integer array of dimension `(2,N)`] `MERGE(1,I)` and `MERGE(2,I)` are the labels of the clusters merged at the `I`th stage. `MERGE(1,N)` and `MERGE(2,N)` are not used.

`DISP` [output; double-precision array of length `N`] `DISP(I)` is a measure of the within-cluster dispersion after the `I`th merge. Dispersion is defined differently for each method: see below. `DISP(N)` is not used.

`IWORK` [local; integer array of length `N`] Work array.

`WORK` [local; double-precision array of length `NW`] Work array.

`NW` [input; integer] Length of array `WORK`. Must be at least `N*(N-1)/2`.

Method

In agglomerative hierarchical clustering, there are initially `N` clusters, each containing one data point, labeled 1 through `N` in the same order as the data points. At each stage of clustering, two clusters are merged. Their labels are saved in the `MERGE` array. The smaller of the two labels is used as the label of the merged cluster. After the `M`th stage of clustering there are `N` − `M` clusters. To find which data points belong to which clusters, use routine `CLUINF`.

Different methods correspond to different criteria for which clusters should be merged at each stage. The following methods are implemented.

- Single-link clustering: the distance between two clusters A and B is defined to be the minimum of the Euclidean distances between pairs of points with one point in A and one in B. At each stage, the two clusters separated by the smallest distance are merged. The square of this distance is saved in the corresponding element of array `DISP`.

- Complete-link clustering: the distance between two clusters A and B is defined to be the maximum of the Euclidean distances between pairs of points with one point in A and one in B. At each stage, the two clusters separated by the smallest distance are merged. The square of this distance is saved in the corresponding element of array `DISP`. `DISP(I)` is therefore the largest squared Euclidean distance between two points that are in the same cluster after the `I`th merge.

- Ward's method: at each stage, the clusters that are merged are chosen to minimize the total within-cluster sum of squared deviations of each attribute about the cluster mean. This sum of squares is saved in the corresponding element of array `DISP`.

This routine makes no claim to efficiency in either computing time or storage requirements. It is computationally practical for small and moderate-sized problems. For large problems, more efficient clustering programs may be preferred, e.g., for Ward's procedure, the `hcon2` program of Murtagh (1985), available from StatLib at `http://lib.stat.cmu.edu/multi/hcon2`.

```
SUBROUTINE CLUINF(NCLUST,N,MERGE,IASSGN,LIST,NUM)
```

Purpose

Obtains information about clusters arising from agglomerative hierarchical clustering.

Agglomerative hierarchical clustering procedures typically produce a list of the clusters merged at each stage of the clustering. This routine uses this list to construct arrays that explicitly show which cluster a given data point belongs to, and which data points belong to a given cluster.

Formal parameters

`NCLUST` [input; integer] Number of clusters.

`N` [input; integer] Number of data points.

`MERGE` [input; integer array of dimension `(2,N)`] `MERGE(1,I)` and `MERGE(2,I)` identify the clusters merged at the `I`th step. This is the array `MERGE` returned by routine `CLUAGG`, and should be left unchanged after exit from that routine.

`IASSGN` [output; integer array of length `N`] Its `I`th element is the number of the cluster to which the `I`th data point belongs.

`LIST` [output; integer array of length `N`] Contains the data points in cluster 1, followed by the data points in cluster 2, etc. Data points in each cluster are listed in increasing order. The last data point in each cluster is indicated by a negative number. See the example below.

`NUM` [output; integer array of length `NCLUST`] Contains the number of data points in each cluster.

Cluster numbers used in arrays `IASSGN`, `LIST` and `NUM` range from 1 to `NCLUST`. They are arbitrary, but are uniquely defined: cluster 1 contains data point 1, cluster `M` ($\texttt{M} \geq 2$) contains data point `J`, where `J=MERGE(2,N-M)`.

Example of the `LIST` array. Suppose that there are 8 data points and 3 clusters, and that the elements of the `LIST` array are 1, –4, 3, 6, –8, 2, 5, –7. Then the clusters are as follows: cluster 1 contains points 1 and 4; cluster 2 contains points 3, 6 and 8; cluster 3 contains points 2, 5 and 7.

```
SUBROUTINE CLUKM(X,NX,N,NATT,NCLUST,IASSGN,LIST,NUM,SS,MAXIT,IWORK,RW,NW)
```

Purpose

Performs cluster analysis by the K-means algorithm. The aim of the K-means algorithm is to partition a set of points into K clusters so that the within-cluster sum of squared distances from the points to their cluster centers is minimized. Except for very small data sets, it is not practical to require that the sum of squares is minimized over all possible partitions. Instead, a local optimum is sought such that no movement of a point from one cluster to another will reduce the within-cluster sum of squares.

Formal parameters

`X` [input; double-precision array of dimension `(NX,NATT)`] `X(I,J)` should contain the `J`th attribute for the `I`th data point.

`NX` [input; integer] The first dimension of array `X`, as declared in the calling program.

`N` [input; integer] Number of data points.

`NATT` [input; integer] Number of attributes for each data point.

`NCLUST` [input; integer] Number of clusters.

`IASSGN` [input and output; integer array of length `N`] On entry, should contain the initial assignment of sites to clusters. On exit, contains the final assignment. The `I`th element of the array contains the label of the cluster to which the `I`th data point belongs. Labels must be between 1 and `NCLUST`, and each of the values 1 through `NCLUST` must occur at least once.

`LIST` [output; integer array of length `N`] Contains the data points in cluster 1, followed by the data points in cluster 2, etc. Data points in each cluster are listed in increasing order. The last data point in each cluster is indicated by a negative number.

`NUM` [output; integer array of length `NCLUST`] Contains the number of data points in each cluster.

`SS` [output; double precision] Within-group sum of squares of the final clusters.

`MAXIT` [input; integer] Maximum number of iterations for the K-means algorithm.

`IWORK` [local; integer array of length `NCLUST*3`] Work array.

`RW` [local; real array of length `NW`] Work array. N.B. this array is of type real, not double precision!

`NW` [input; integer] Length of array `RW`. Must be at least `(N+NCLUST)*(NATT+1)+2*NCLUST`.

Method

This routine is a front end to Applied Statistics Algorithm AS136 (Hartigan and Wong, 1979). Algorithm AS136 is not included in this package, but is available from StatLib at `http://stat.lib.cmu.edu/apstat/136`.

`SUBROUTINE REGLMR(NSITE,NMOM,NXMOM,XMOM,WEIGHT,RMOM)`

Purpose

Calculates regional weighted averages of L-moment ratios, as in Hosking and Wallis (1997, sec. 6.2).

Formal parameters

`NSITE` [input; integer] The number of sites in the region.

`NMOM` [input; integer] The number of L-moments to be found.

`NXMOM` [input; integer] The first dimension of array `XMOM`, as declared in the calling program.

`XMOM` [input; double-precision array of dimension `(NXMOM,NSITE)`] `XMOM(K,I)` should contain the `K`th L-moment ratio for site `I`, as returned by a previous call to routine `SAMLMR`.

`WEIGHT` [input; double-precision array of length `NSITE`] `WEIGHT(I)` should contain the relative weight of site `I` in the regional weighted average. It is recommended that, as in Hosking and Wallis (1997, sec. 6.2), the weights be proportional to the record lengths at each site.

`RMOM` [output; double-precision array of length `NMOM`] The regional weighted average L-moment ratios ℓ_1^{R}, ℓ_2^{R}, t_3^{R}, t_4^{R}, ... , as defined below.

Numerical method

Let the number of sites be N and let the L-moment ratios at site i be $\ell_1^{(i)}$, $\ell_2^{(i)}$, $t_3^{(i)}$, $t_4^{(i)}$, ... Let the weight allotted to site i be w_i. When the data are scaled by dividing by the mean, the L-moment ratios become 1, $t^{(i)}$, $t_3^{(i)}$, $t_4^{(i)}$, ..., where $t^{(i)} = \ell_2^{(i)}/\ell_1^{(i)}$ is the at-site L-CV. The regional weighted average mean, L-CV and L-moment ratios are the weighted averages of these at-site values, and are defined by

$$\ell_1^{\mathrm{R}} = 1, \qquad t^{\mathrm{R}} = \sum_{i=1}^{N} w_i t^{(i)} \Big/ \sum_{i=1}^{N} w_i \,, \qquad t_r^{\mathrm{R}} = \sum_{i=1}^{N} w_i t_r^{(i)} \Big/ \sum_{i=1}^{N} w_i \,, \quad r = 3, 4, ...$$

(cf. Hosking and Wallis, 1997, sec. 6.2, who set w_i to be the record length at site i).

```
SUBROUTINE REGTST(NSITES,NAMES,LEN,XMOM,A,B,SEED,NSIM,NPROB,PROB,
KPRINT,KOUT,RMOM,D,VOBS,VBAR,VSD,H,Z,PARA)
```

Purpose

Calculates the discordancy, heterogeneity and goodness-of-fit measures described by Hosking and Wallis (1997, chaps. 3–5).

Formal parameters

`NSITES` [input; integer] Number of sites in region.

`NAMES` [input; `CHARACTER*12` array of length `NSITES`] Site names.

`LEN` [input; double-precision array of length `NSITES`] Record lengths at each site.

`XMOM` [input; double-precision array of dimension `(5,NSITES)`] Contains the first 5 sample L-moment ratios for each site, in the order mean, L-CV, t_3, t_4, t_5. `XMOM(K,I)` should contain the `K`th L-moment ratio for site `I`. Elements `XMOM(1,.)` are not used by the routine. Note that `XMOM(2,.)` should contain L-CV, not the usual ℓ_2!

`A,B` [input; double precision] Parameters of plotting position. Should be the same as the values used to calculate the L-moments in the `XMOM` array.

`SEED` [input; double precision] Seed for random number generator. Should be a whole number in the range `2D0` to `2147483647D0`.

`NSIM` [input; integer] Number of simulated worlds used in calculation of heterogeneity and goodness-of-fit measures. If `NSIM` is set to zero, the routine will calculate only the discordancy measures.

`NPROB` [input; integer] Number of quantiles to be calculated for each distribution that gives a good fit to the data.

`PROB` [input; double-precision array of length `NPROB`] Probabilities for which quantiles are to be calculated.

`KPRINT` [input; integer] Output flag. Should be set to zero to suppress printed output, to `1` to produce printed output.

`KOUT` [input; integer] Channel to which output is directed.

`RMOM` [output; double-precision array of length 5] Regional weighted average L-moment ratios.

`D` [output; double-precision array of length `NSITES`] Discordancy measure for each site.

`VOBS` [output; double-precision array of length 3] The observed values of three heterogeneity statistics. `V(1)` is the weighted standard deviation of the at-site L-CVs. `V(2)` is the weighted average distance from a site to the regional average on a graph of L-CV vs. L-skewness. `V(3)` is the weighted average distance from a site to the regional average on a graph of L-skewness vs. L-kurtosis.

`VBAR` [output; double-precision array of length 3] The means of the `NSIM` simulated values of the V statistics.

`VSD` [output; double-precision array of length 3] The standard deviations of the `NSIM` simulated values of the V statistics.

`H` [output; double-precision array of length 3] Heterogeneity measures for the region: `H(J)=(VOBS(J)-VBAR(J))/VSD(J)`.

`Z` [output; double-precision array of length 5] Goodness-of-fit measures for five distributions: `(1)` generalized logistic, `(2)` generalized extreme-value, `(3)` generalized Normal (lognormal), `(4)` Pearson type III (3-parameter gamma), and `(5)` generalized Pareto.

`PARA` [output; double-precision array of dimension `(5,6)`] Parameters of regional distributions fitted by the above five distributions, plus `(6)` Wakeby. `PARA(I,J)` contains the `I`th parameter of the `J`th distribution.

Numerical method

The routine straightforwardly carries out the calculations described in the "Formal definition" subsections of Hosking and Wallis (1997, chaps. 3–5). Computed quantities are as defined in the following equations in Hosking and Wallis (1997): discordancy measures are the D_i in eq. (3.3); heterogeneity statistic `V(1)` is V in eq. (4.4); heterogeneity statistics `V(2)` and `V(3)` are V_2 and V_3 in eqs. (4.6) and (4.7); heterogeneity measure `H(1)` is H in eq. (4.5); heterogeneity measures `H(2)` and `H(3)` are computed analogously to H in eq. (4.5) but using V_2 and V_3 in place of V; goodness-of-fit measures are Z^{DIST} in eq. (5.6).

PROGRAM XCLUST

Purpose

This program illustrates the use of the cluster analysis routines. An example data set and the results obtained by applying PROGRAM XCLUST to it are given in the files APPALACH.DAT and APPALACH.OUT respectively. This cluster analysis is described in Hosking and Wallis (1997, sec. 9.2).

The data are site characteristics for 104 streamflow gaging stations in central Appalachia. The attributes for each site are drainage basin area, gage elevation, gage latitude and gage longitude. Nonlinear transformations are applied to two of the attributes: a logarithmic transformation to drainage basin area and a square root transformation to gage elevation. These transformations give a more symmetric distribution of the values of the site characteristics at the 104 sites, reducing the likelihood that a few sites will have site characteristics so far from the other sites that they will always be assigned to a cluster by themselves, and, in Hosking and Wallis's judgement, give a better correspondence between differences in site characteristics and the degree of hydrologic dissimilarity between different basins. All four attributes are then standardized by dividing by the standard deviation of their values at the 104 sites. Finally, the drainage basin area variable is multiplied by 3 to give it an importance in the clustering procedure equal to that of the other variables together. The transformation and weighting of the variables involves subjective decisions whose ultimate justification is the physical plausibility of the regions that are ultimately obtained from the clustering procedure.

Clusters are formed by Ward's method, using routine CLUAGG. Hosking and Wallis (1997) judged that seven clusters is an appropriate number; information about these clusters is printed using routine CLUINF. The clusters obtained by Ward's method are adjusted using the K-means procedure (routine CLUKM). This yields clusters that are a little more compact in the space of transformed site characteristics. The cluster numbers in the output file are not those used by Hosking and Wallis (1997), who renumbered the clusters according to the average drainage area of the sites in each cluster.

PROGRAM XFIT

Purpose

This program is an example of how routine REGLMR and others can be used to perform frequency analysis on a regional data set using an index-flood procedure that fits a regional distribution using the method of L-moments. The program fits a Wakeby distribution to the regional L-moments using the regional L-moment algorithm described by Hosking and Wallis (1997, sec. 6.2). It should be easy to change the program to fit other distributions.

An example data set and the results obtained by applying PROGRAM XFIT to it are given in files MAXWIND.DAT and MAXWIND.OUT respectively. The data are annual maximum wind speeds at 12 sites in the U.S., and are taken from Simiu et al. (1979). These 12 sites, all in the south-east of the U.S. and close to the coast, were judged to constitute a homogeneous region suitable for the application of an index-flood procedure (Wallis, 1993).

PROGRAM XTEST

Purpose

This program is an example of the use of routine REGTST. The program reads a file of at-site sample L-moments for one or more regions and calls routine REGTST to calculate test statistics for the regions.

An example data set and the results obtained by applying PROGRAM XTEST to it are given in files CASCADES.DAT and CASCADES.OUT respectively. The data set is given in Hosking and Wallis (1997, Table 3.4) and used in the "Example" sections of Hosking and Wallis (1997, chaps. 3–6). The data are annual precipitations at 19 sites in the northwest U.S., the "North Cascades" region of Plantico et al. (1990). Data were obtained from the Historical Climatology Network (Karl et al., 1990). The results in CASCADES.OUT indicate that the region is acceptably close to homogeneous and can be well described by a lognormal or Pearson type III distribution.

PROGRAM XSIM

Purpose

This program illustrates the use of Monte Carlo simulation to derive the properties of estimated quantiles in regional frequency analysis. The program implements the algorithm in Table 6.1 of Hosking and Wallis (1997, p. 95), with inter-site correlation matrix taking the form of equal correlation between each pair of sites (Hosking and Wallis, 1997, eq. (6.11)).

The region is set up as in Hosking and Wallis (1997, sec. 6.5). The region contains 19 sites with record lengths as for the North Cascades data. The sites have lognormal frequency distributions with L-CV varying linearly from 0.0978 at site 1 to 0.1228 at site 19, and each site has L-skewness $\tau_3 = 0.0279$. The intersite correlation is 0.64, the average intersite correlation for the North Cascades sites. $10,000$ realizations of this region are made and the regional L-moment algorithm described by Hosking and Wallis (1997, sec. 6.2) is used to fit a lognormal distribution to the data generated at each realization. The regional average relative RMSE of the estimated growth curve is calculated from the simulations, and quantiles of the distribution of the regional average of the ratio of the estimated to the true at-site growth curve are computed from a histogram accumulated during the simulations. From these quantiles, 90% error bounds for the growth curve are computed as in Hosking and Wallis (1997, eq. (6.19)). These results are contained in the output from PROGRAM XSIM, which is file XSIM.OUT.

PROGRAM XSIM can also be used to find the average value taken by heterogeneity measures over different simulations of a given region. This is useful when assessing the accuracy of estimated quantiles in regional frequency analysis, as discussed by Hosking and Wallis (1997, sec. 6.4). File XSIMH.OUT contains the output from running PROGRAM XSIM with parameter values DSEED=619145091D0, NREP=100, NSIM=500, and KPRINT=1, and omitting the section between dashed lines near the end of the program. The first of the AVERAGE HETEROGENEITY MEASURES in this output is the source of the statement "the average H value of simulated regions is 1.08" in Hosking and Wallis (1997, p. 98).

Auxiliary routines

`FUNCTION DERF` Error function, double precision. The numerical method is the same as that of Algorithm 5666 of Hart et al. (1968).

`FUNCTION DIGAMD` Euler's psi function, $\psi(x) = d\{\log\Gamma(x)\}/dx$, double precision. The numerical method is the same as that of Algorithm AS103 (Bernardo, 1976).

`FUNCTION DLGAMA` Natural logarithm of the gamma function, double precision. The numerical method is the same as that of Algorithm ACM291 (Pike and Hill, 1966).

`SUBROUTINE DURAND(SEED,N,X)` Fills the double-precision array `X`, of length `N`, with pseudo-random numbers uniformly distributed on the interval $(0, 1)$. The double-precision variable `SEED` is the seed for the random-number generator. It should be initialized to an integer value between 2 and 2147483647 $(2^{31} - 1)$, though small values, less than 200000 say, should be avoided. It should be left unchanged between successive calls to the routine.

The routine uses the algorithm of Lewis et al. (1969).

This routine is supplied for completeness and portability, but on most computers it is more efficient to replace routine `DURAND` by a routine programmed in assembly language. For example, on an IBM RS6000 44P model 170 workstation, the Engineering and Scientific Subroutine Library for AIX, Version 4.2, contains a routine, also called `DURAND`, which has an identical specification to the one described here and is about 7 times as fast.

`FUNCTION GAMIND(X,A,G)` Incomplete gamma integral, double precision. The function is defined to be

$$G(x, \alpha) = \{\Gamma(\alpha)\}^{-1} \int_0^x t^{\alpha-1} e^{-t}\, dt\,.$$

The three arguments of the routine are x, α and $\log\{\Gamma(\alpha)\}$. The numerical method is the same as that of Algorithm AS239 (Shea, 1988), except that a different approximation to $G(x, \alpha)$—Hill's approximation (Johnson et al., 1994, eq. (18.35))—is used for large α.

`FUNCTION QUASTN` Quantile function of standard Normal distribution, double precision. The numerical method is the same as that of Algorithm AS241 (Wichura, 1988).

`SUBROUTINE SORT(X,N)` Sorts the double-precision array `X`, of length `N`, into ascending order.

Data files

`APPALACH.DAT` Example data set: data for 104 streamflow gaging stations in central Appalachia. The columns of data are as follows.

`siteid`	The site's Hydrologic Unit Code, a unique identifier.
`lat`	Gage latitude, in degrees.
`long`	Gage longitude, in degrees west of the Greenwich meridian.
`area`	Drainage basin area, in square miles.
`elev`	Gage elevation, in feet.
`n`	Sample size.
`l1`	Sample mean, ℓ_1.
`t`	Sample L-CV, t.
`t3`	Sample L-skewness, t_3.
`t4`	Sample L-kurtosis, t_4.
`t5`	Sample 5th L-moment ratio, t_5.

Only the four columns `lat`, `long`, `area` and `elev` are used by `PROGRAM XCLUST`. The data in these columns, and the streamflow data used to compute the sample L-moments, were obtained from "Hydrodata" CD-ROMs (Hydrosphere, 1993), which reproduce data from the U.S. Geological Survey's WATSTORE data files.

`APPALACH.OUT` Output when `PROGRAM XCLUST` is applied to the data in `APPALACH.DAT`.

`MAXWIND.DAT` Example data set: annual maximum wind speeds at 12 sites in the southeast U.S., from Simiu et al. (1979). Numbers in parentheses are those used by Simiu et al. (1979, Table 3.4.1) to identify the sites.

`MAXWIND.OUT` Output when `PROGRAM XFIT` is applied to the data in `MAXWIND.DAT`.

`CASCADES.DAT` Example data set: L-moments of annual precipitation at 19 sites in the northwest U.S. The columns in the table are the site identifier used by the Historical Climatology Network, the record length, and the at-site sample L-moments ℓ_1, L-CV, t_3, t_4, t_5.

`CASCADES.OUT` Output when `PROGRAM XTEST` is applied to the data in `CASCADES.DAT`.

`XSIM.OUT` Output from `PROGRAM XSIM`.

`XSIMH.OUT` Output from `PROGRAM XSIM` with parameters `DSEED=619145091D0`, `NREP=100`, `NSIM=500`, and other modifications as described on page 25.

Links between routines

The following functions and subroutines call other routines.

CDFGAM calls DERF, DLGAMA, GAMIND.
CDFGNO calls DERF.
CDFNOR calls DERF.
CDFPE3 calls DERF, DLGAMA, GAMIND.
CDFWAK calls QUAWAK.
LMRGAM calls DLGAMA.
LMRGEV calls DLGAMA.
LMRGNO calls DERF.
LMRKAP calls DLGAMA, DIGAMD.
LMRPE3 calls DLGAMA.
PELGAM calls DLGAMA.
PELGEV calls DLGAMA.
PELGNO calls DERF.
PELKAP calls DLGAMA, DIGAMD.
PELPE3 calls DLGAMA.
QUAGAM calls DERF, DLGAMA, GAMIND, QUASTN.
QUAGNO calls QUASTN.
QUANOR calls QUASTN.
QUAPE3 calls DERF, DLGAMA, GAMIND, QUAGAM, QUASTN.
CLUKM calls routines KMNS, OPTRA and QTRAN from Hartigan and Wong (1979).
REGTST calls most of the PELxxx, QUAxxx and auxiliary routines.
GAMIND calls DERF.

Machine dependence

Some constants initialized in data statements in some of the routines may need to be changed to give good results on different machines. The list below explains how the values of these constants should be chosen. The values assigned to these constants in the listings of the routines are appropriate for an IBM 390 computer using the FORTVS2 compiler, for which double-precision constants have precision of approximately 15 decimal digits and maximum magnitude approximately 10^{75}.

Routine Constant

`CDFxxx` `SMALL` [used by routines `CDFGEV`, `CDFGLO`, `CDFGNO`, `CDFGPA`, `CDFKAP`] Should be as small as possible, so long as underflow or overflow conditions do not occur.

`CDFWAK` `UFL` A number x such that the operation e^x just does not cause underflow.

`LMRKAP` `OFL` A number x such that the operation e^x just does not cause overflow.

`PELKAP` `OFLEXP` A number x such that the operation e^x just does not cause overflow.

`PELKAP` `OFLGAM` A number x such that the operation $\Gamma(x)$ just does not cause overflow.

`QUAWAK` `UFL` A number x such that the operation e^x just does not cause underflow.

`DERF` `X1` A number x just large enough that $\operatorname{erf} x = 1$ to machine accuracy.

`DLGAMA` `BIG` A number x such that

$$\log \Gamma(x) = (x - \tfrac{1}{2}) \log x - x + \tfrac{1}{2} \log(2\pi)$$

to machine precision. The value $10^{d/2}$, where d is the number of decimal digits in a double-precision constant, should be adequate.

`DLGAMA` `TOOBIG` A number x such that the operation x^2 just does not cause overflow.

`GAMIND` `UFL` A number x such that the operation e^x just does not cause underflow.

In routines that use iterative methods (`CDFWAK`, `LMRGNO`, `PELGEV`, `PELKAP`, `QUAGAM`, `GAMIND`), constants `EPS` and `MAXIT` control the test for convergence of the iterations. If an iterative procedure fails frequently, it may be helpful to change the values of these constants.

Change history

Initial release (July 1988)

Routines: CDFGEV, CDFGLO, CDFGNO, CDFGPA, CDFGUM, CDFKAP, CDFWAK, LMRGEV, LMRGLO, LMRGNO, LMRGPA, LMRGUM, LMRKAP, LMRWAK, PELGEV, PELGLO, PELGNO, PELGPA, PELGUM, PELKAP, PELWAK, QUAGEV, QUAGLO, QUAGNO, QUAGPA, QUAGUM, QUAKAP, QUAWAK, SAMLMR, SAMPWM, SAMREG, SAMXMP, ALGAMD, DIGAMD, ERFD, QUASTN, SORT.

Version 2 (August 1991)

Routines added: CDFGAM, CDFNOR, CDFPE3, LMRGAM, LMRNOR, LMRPE3, PELGAM, PELNOR, PELPE3, QUAGAM, QUANOR, QUAPE3, REGLMR, REGTST, XFIT, XTEST, DERF, DLGAMA, GAMIND.

Routines modified: all except CDFGUM, LMRGPA, LMRWAK.

Routines replaced: ALGAMD (by DLGAMA), ERFD (by DERF), SAMREG (by REGLMR), SAMXMP (by XTEST).

Version 3 (August 1996)

Routines added: CDFEXP, LMREXP, PELEXP, QUAEXP, SAMLMU, CLUAGG, CLUINF, CLUKM, XCLUST, XSIM.

Routines modified:

CDFGUM Removed initialization of an unused variable.

CDFWAK Iterative procedure modified to reduce the (already small) chance that it will fail to converge. Minor bug fixes.

LMRKAP Removed initialization of an unused variable.

PELGEV More accurate numerical approximation.

PELGNO Simpler and more accurate numerical approximation. Routine QUASTN no longer required. Returns $\alpha = -1$ if τ_3 is too large ($|\tau_3| \geq 0.95$).

PELKAP Modified to reduce the chance of numeric overflow. In consequence, two new values of the IFAIL parameter are possible.

REGTST Changed critical values for discordancy measure. Order in which output is printed was changed: table of parameters now printed before table of quantiles. Placement of IMPLICIT statement corrected: this can cause minor changes in the simulation results returned by the routine. Minor bugs and inconsistencies removed. Minor changes to comments.

XFIT Minor bug fix. Minor changes to comments.

XTEST Minor bug fixes. Minor changes to comments.

DERF Corrected error when argument of function was exactly zero. Removed initialization of an unused variable.

DLGAMA Minor bug fix.

DURAND Speed increased.

Version 3.01 (December 1996)

Routines modified:

XSIM	Replaced call to nonexistent function CDFSTN by call to DERF.

Version 3.02 (March 1997)

Routines modified:

CLUAGG	Implemented single-link and complete-link clustering.
CLUINF	Added validity check for parameters. Minor changes to comments.
XCLUST	Minor change to FORMAT statement 6080.
XSIM	Changed random number seed.

The file XSIM.OUT was also changed, in consequence of the changes to XCLUST and XSIM.

Version 3.03 (June 2000)

Routines modified:

REGTST	CHARACTER variable declarations changed to conform to Fortran-77 standard.
XTEST	CHARACTER variable declarations changed to conform to Fortran-77 standard.
XSIM	RETURN statements replaced by STOP.
QUASTN	Changed WRITE(6,7000) statement to be consistent with FORMAT statement 7000.

The file APPALACH.OUT was also changed, as a consequence of the change to PROGRAM XCLUST made at Version 3.02.

Version 3.04 (July 2005)

Routines modified:

PELWAK	Minor bug fix in test for validity of L-moments.
SAMLMU	Set XMOM(1) to sample mean, not zero, when all data values are equal.
REGTST	Print flagged values of discordancy measure even when there is only one.
XCLUST	Removed declarations of unused variables.
XSIM	Removed declarations of unused variables.

References

Bernardo, J. M. (1976). Algorithm AS 103: psi (digamma) function. *Applied Statistics*, **25**, 315–317.

Churchhouse, R. F. (ed.) (1981). *Numerical methods*, Handbook of applicable mathematics, vol. 3. New York: Wiley.

Donaldson, R. W. (1996). Calculating inverse CV, skew and PWM functions for Pearson-3, log-normal, extreme-value, and log-logistic distributions. *Communications in Statistics—Simulation and Computation*, **25**, 741–747.

Greenwood, J. A., Landwehr, J. M., Matalas, N. C., and Wallis, J. R. (1979). Probability weighted moments: definition and relation to parameters of several distributions expressable in inverse form. *Water Resources Research*, **15**, 1049–1054.

Hart, J. F., et al. (1968). *Computer approximations*. New York: Wiley.

Hartigan, J. A., and Wong, M. A. (1979). Algorithm AS 136: A K-means clustering algorithm. *Applied Statistics*, **28**, 100–108.

Hosking, J. R. M. (1990). L-moments: analysis and estimation of distributions using linear combinations of order statistics. *Journal of the Royal Statistical Society, Series B*, **52**, 105–124.

Hosking, J. R. M. (1994). The four-parameter kappa distribution. *IBM Journal of Research and Development*, **38**, 251–258.

Hosking, J. R. M., and Wallis, J. R. (1987). Parameter and quantile estimation for the generalized Pareto distribution. *Technometrics*, **29**, 339–349.

Hosking, J. R. M., and Wallis, J. R. (1995). A comparison of unbiased and plotting-position estimators of L-moments. *Water Resources Research*, **31**, 2019–2025.

Hosking, J. R. M., and Wallis, J. R. (1997). *Regional frequency analysis: an approach based on L-moments*. Cambridge, England: Cambridge University Press.

Hosking, J. R. M., Wallis, J. R., and Wood, E. F. (1985). Estimation of the generalized extreme-value distribution by the method of probability-weighted moments. *Technometrics*, **27**, 251–261.

Huh, M. Y. (1986). Computation of percentage points. *Communications in Statistics—Simulation and Computation*, **15**, 1191–1198.

Hydrosphere (1993). *Hydrodata CD-ROMs, vol. 4.0: USGS peak values*. Hydrosphere Data Products, Boulder, Colo.

Johnson, N. L., Kotz, S., and Balakrishnan, N. (1994). *Continuous univariate distributions, vol. 1*, 2nd edn. New York: Wiley.

Karl, T. R., Williams, C. N., Quinlan, F. T., and Boden, T. A. (1990). United States historical climatology network (HCN) serial temperature and precipitation data. *ORNL/CDIAC-30 NDP-019/R1*, Carbon Dioxide Information Analysis Center, Oak Ridge National Laboratory, Oak Ridge, Tenn.

Kotz, S., Johnson, N. L., and Read, C. B. (eds.) (1988). *Encyclopedia of statistical sciences, vol. 9*. New York: Wiley.

Landwehr, J. M., Matalas, N. C., and Wallis, J. R. (1979a). Probability weighted moments compared with some traditional techniques in estimating Gumbel parameters and quantiles. *Water Resources Research*, **15**, 1055–1064.

Landwehr, J. M., Matalas, N. C., and Wallis, J. R. (1979b). Estimation of parameters and quantiles of Wakeby distributions. *Water Resources Research*, **15**, 1361–1379. Correction: *Water Resources Research*, **15** (1979), 1672.

Lewis, P. A. W., Goodman, A. E., and Miller, J. M. (1969). A pseudorandom number generator for the System/360. *IBM Systems Journal*, **8**, 136–146.

Murtagh, F. (1985). *Multidimensional clustering algorithms*. Berlin: Springer.

Neuman, C. P., and Schonbach, D. I. (1974). Discrete (Legendre) orthogonal polynomials: a survey. *International Journal for Numerical Methods in Engineering*, **8**, 743–770.

Pike, M. C., and Hill, I. D. (1966). Algorithm 291: logarithm of the gamma function. *Communications of the Association for Computing Machinery*, **9**, 684.

Plantico, M. S., Karl, T. R., Kukla, G., and Gavin, J. (1990). Is recent climate change across the United States related to rising levels of anthropogenic greenhouse gases? *Journal of Geophysical Research*, **95**, 16617–16637.

Shea, B. L. (1988). Algorithm AS 239: chi-squared and incomplete gamma integral. *Applied Statistics*, **37**, 466–473.

Simiu, E., Chanery, M. J., and Filliben, J. J. (1979). Extreme wind speeds at 129 stations in the contiguous United States. *Building Science Series 118*, National Bureau of Standards, Washington, D.C.

Wallis, J. R. (1993). Regional frequency studies using *L*-moments. In *Concise encyclopedia of environmental systems*, ed. P. C. Young, 468–476. Oxford, U.K.: Pergamon.

Wichura, M. (1988). Algorithm AS 241: the percentage points of the normal distribution. *Applied Statistics*, **37**, 477–484.