

# Package ‘SCI’

January 20, 2025

**Type** Package

**Title** Standardized Climate Indices Such as SPI, SRI or SPEI

**Version** 1.0-2

**Date** 2016-05-02

**Author** Lukas Gudmundsson & James H. Stagge

**Maintainer** Lukas Gudmundsson <lukas.gudmundsson@env.ethz.ch>

**Depends** fitdistrplus, lmomco

**Suggests** evd

**Description** Functions for generating Standardized Climate Indices (SCI).

SCI is a transformation of (smoothed) climate (or environmental) time series that removes seasonality and forces the data to take values of the standard normal distribution. SCI was originally developed for precipitation. In this case it is known as the Standardized Precipitation Index (SPI).

**License** GPL (>= 2)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2016-05-03 11:03:16

## Contents

SCI-package . . . . .	2
dist.start . . . . .	3
fitSCI . . . . .	4
genlog . . . . .	9
pe3 . . . . .	10

<b>Index</b>	<b>12</b>
--------------	-----------

**Description**

Functions for generating Standardized Climate Indices (SCI). SCI is a transformation of (smoothed) climate (or environmental) time series that removes seasonality and forces the data to take values of the standard normal distribution. SCI was originally developed for precipitation. In this case it is known as the Standardized Precipitation Index (SPI).

**Details**

Package: SCI  
Type: Package  
Version: 1.0-2  
Date: 2016-05-02  
License: GPL (>= 2)

**Author(s)**

Lukas Gudmundsson & James Stagge

Maintainer: Lukas Gudmundsson <lukas.gudmundsson@env.ethz.ch>

**References**

Stagge, J.H. ; Tallaksen, L.M.; Gudmundsson, L.; van Loon, A.; Stahl, K.: Candidate Distributions for Climatological Drought Indices (SPI and SPEI), 2015, International Journal of Climatology, 35, 4027-4040, doi:10.1002/joc.4267.

Stagge, J.H. ; Tallaksen, L.M.; Gudmundsson, L.; van Loon, A.; Stahl, K.: Response to comment on "Candidate Distributions for Climatological Drought Indices (SPI and SPEI)", 2016, International Journal of Climatology, 36, 2132-2138, doi:10.1002/joc.4564.

**Examples**

```
## create artificial data, resembling precipitation
set.seed(101)
n.years <- 60
date <- rep(1:n.years,each=12) + 1950 + rep((0:11)/12,times=n.years)
PRECIP <- (0.25*sin( 2 * pi * date) + 0.3)*rgamma(n.years*12, shape = 3, scale = 1)
PRECIP[PRECIP<0.1] <- 0

## apply SCI transformation
spi.para <- fitSCI(PRECIP,first.mon=1,time.scale=6,distr="gamma",p0=TRUE)
spi <- transformSCI(PRECIP,first.mon=1,obj=spi.para)
plot(date,spi,t="1")
```

---

dist.start

*Rough estimates for parameters of selected distributions*


---

## Description

Produces rough parameter estimates for specific distributions (`distr`) that are useful as starting values for maximum likelihood estimation.

## Usage

```
dist.start(x, distr, ...)
lmom.start(x, distr = c("gamma", "genlog", "gev", "gumbel",
                        "lnorm", "norm", "pe3", "weibull"), ...)
mom.start(x, distr = c("gamma", "gumbel", "logis", "lnorm", "norm",
                       "weibull"), ...)
```

## Arguments

<code>x</code>	numeric vector
<code>distr</code>	A character string "name" naming a distribution for which the corresponding density function ( <code>dname</code> ), the corresponding distribution function ( <code>pname</code> ) and the quantile function ( <code>qname</code> ) must be defined (see for example <a href="#">GammaDist</a> )
<code>...</code>	arguments passed to other functions, currently not used.

## Details

`lmom.start` uses L-moments for parameter estimation. In most cases it relies on functionality of the `lmomco` package. Currently available distributions are: "[gamma](#)", "[genlog](#)", "[gev](#)", "[gumbel](#)", "[logis](#)", "[lnorm](#)", "[norm](#)", "[pe3](#)", "[weibull](#)".

`mom.start` uses moments (e.g. mean, standard deviation) for parameter estimation. Some estimates are precise, others only approximations that provide reasonable starting values. Currently available distributions are: "[gamma](#)", "[gumbel](#)", "[logis](#)", "[lnorm](#)", "[norm](#)", "[weibull](#)".

`dist.start` calls first `lmom.start` to estimate parameters. In case of failure `mom.start` is called, hopefully producing reasonable parameter estimates.

## Value

named list, names correspond to distribution parameters. In case of failure, the same list with NA values is returned.

## Author(s)

Lukas Gudmundsson & James Stagge

**Examples**

```

lmom.start(rgamma(100,shape=0.5,rate=1),"gamma")

mom.start(rgamma(100,shape=0.5,rate=1),"gamma")

dist.start(rgamma(100,shape=0.5,rate=1),"gamma")

```

---

```

fitSCI                                Standardized Climate Index (SCI)

```

---

**Description**

fitSCI identifies parameters for the Standardized Climate Index (SCI) transformation. transformSCI applies the transformation

**Usage**

```

fitSCI(x, ...)
## Default S3 method:
fitSCI(x, first.mon, time.scale, distr, p0,
       p0.center.mass=FALSE, scaling=c("no","max","sd"),mledist.par = list(),
       start.fun = dist.start, start.fun.fix = FALSE, warn = TRUE, ...)

transformSCI(x, ...)
## Default S3 method:
transformSCI(x, first.mon, obj, sci.limit = Inf, warn=TRUE, ...)

```

**Arguments**

x	numeric vector, representing a monthly univariate time series.
first.mon	value in 1:12 indicating the month of the first element of x
time.scale	The time scale (integer) of the SCI calculation. The time scale is the window length of an backward looking running mean.
distr	A character string "name" naming a distribution for which the corresponding density function (dname), the corresponding distribution function (pname) and the quantile function (qname) must be defined (see for example <a href="#">GammaDist</a> )
p0	if TRUE, model Probability of zero (precipitation) months is modeled with a mixed distribution as $D(x) = p_0 + (1 - p_0)G(x)$ , where $G(x) > 0$ is the reference distribution (e.g. Gamma) $p_0$ is the probability of a zero (precipitation) month.
p0.center.mass	If TRUE, the Probability of zero (precipitation) is estimated using a "center of mass" estimate based on the Weibull plotting position function (see details). Only applies if p0=TRUE.

<code>scaling</code>	Indicates whether to do some scaling of <code>x</code> prior to parameter identification. "no" (the default) indicates no scaling. "max" indicates scaling by the maximum of <code>x</code> , such that <code>x &lt;- x/max(x, na.rm=TRUE)</code> . "sd" stands for scaling by the standard deviation. Scaling can stabilize parameter estimation.
<code>mledist.par</code>	named list that can be used to pass parameters to <code>mledist</code> in package <b>fitdistr-plus</b> .
<code>start.fun</code>	Function with arguments <code>x</code> and <code>distr</code> estimating initial parameters of the function <code>distr</code> for each month. The function should return a named list corresponding to the parameters of <code>distr</code> . (See also <code>dist.start</code> )
<code>start.fun.fix</code>	logical argument, indicating if parameter estimates of <code>start.fun</code> should be used if maximum likelihood estimation breaks down. This stabilizes the implementation but can introduce biases in the resulting SCI.
<code>obj</code>	an object of class <code>fitSCI</code> , output from <code>fitSCI</code> .
<code>sci.limit</code>	Truncate absolute values of SCI that are large than <code>sci.limit</code> . See details.
<code>warn</code>	Issue warnings if problems in parameter estimation occur.
<code>...</code>	further arguments passed to methods

## Details

`fitSCI` estimates the parameters for transforming a meteorological and environmental time series to a Standardized Climate Index (SCI). `transformSCI` applies the standardisation. Typical SCI are the Standardized Precipitation Index (SPI), the Standardized Runoff Index (SRI) or the Standardized Precipitation Evapotranspiration Index (SPEI).

To reduce biases in the presence of many zero (precipitation) events, the probability of these events ( $p_0$ ) can be estimated using a "center of mass" estimate based on the Weibull plotting position function (`p0.center.mass=TRUE`). Following Stagge et al. (2014) the probability of zero events is then estimated as  $p_0 = \frac{n_p}{n+1}$ , where  $n_p$  refers to the number of zero events and  $n$  is the sample size. The resulting mixed distribution used for SCI transformation is then

$$D(x) = \begin{cases} p_0 + (1 - p_0)G(x) & \text{if } x > 0 \\ \frac{n_p + 1}{2(n + 1)} & \text{if } x = 0 \end{cases}$$

where  $G(x) > 0$  is a model (e.g. gamma) distribution.

Uncertainty in distribution parameters can cause unrealistically large (small) SCI values if values in `x` exceed the values used for parameter estimation (see `fitSCI`). Therefore `transformSCI` allows for a truncation of the SCI series such that `abs(sci) <= sci.limit`. The truncation can be disabled by setting `sci.limit=Inf`.

## Value

`fitSCI` returns an object of class "fitSCI" with the following components:

<code>dist.param</code>	A column matrix containing the parameters of distribution <code>distr</code> for each month. Row names correspond to the distribution parameters. If <code>p0=TUE</code> an additional row named <code>P0</code> is introduced, indicating the probability of zero (precipitation) events.
-------------------------	--

`dist.para.flag` an vector indicating possible issues occurring throughout parameter estimation. Possible values are: 0. no problems occurred; 1. starting values could not be estimated; 2. `mledist` crashed with unknown error; 3. `mledist` did not converge; 4. all values in this month are NA; 5. all values in this month are constant, distribution not defined.

`time.scale` The time scale (integer) of the SCI calculation.

`distr` A character string "name" naming a distribution used

`p0` logical indicating whether probability of zero (precipitation) events is estimated separately.

`p0.center.mass` logical indicating whether probability of zero (precipitation) events is estimated using the "centre of mass" estimator (see Stagge et al. (2014) for details).

`scaling` numeric value that has been used to scale  $x$  (see argument `scaling`). A value of 1 results from `scaling="no"`, other values are the maximum value or the standard deviation of  $x$ , depending on the choice of the parameter scaling.

`call` the function call

`transformSCI` returns a numeric vector containing the SCI, having values of the standard normal distribution.

**Note**

This function is intended to be used together with `transformSCI`.

**Author(s)**

Lukas Gudmundsson & James Stagge

**References**

Stagge, J.H. ; Tallaksen, L.M.; Gudmundsson, L.; van Loon, A.; Stahl, K.: Candidate Distributions for Climatological Drought Indices (SPI and SPEI), 2015, International Journal of Climatology, 35, 4027-4040, doi:10.1002/joc.4267.

Stagge, J.H. ; Tallaksen, L.M.; Gudmundsson, L.; van Loon, A.; Stahl, K.: Response to comment on "Candidate Distributions for Climatological Drought Indices (SPI and SPEI)", 2016, International Journal of Climatology, 36, 2132-2138, doi:10.1002/joc.4564.

McKee, T.; Doesken, N. & Kleist, J.: The relationship of drought frequency and duration to time scales Preprints, 8th Conference on Applied Climatology, 1993, 179-184.

Shukla, S. & Wood, A. W.: Use of a standardized runoff index for characterizing hydrologic drought Geophysical Research Letters, 2008, 35, L02405.

Vicente-Serrano, S. M.; Begueria, S. & Lopez-Moreno, J. I.: A Multiscalar Drought Index Sensitive to Global Warming: The Standardized Precipitation Evapotranspiration Index J. Climate, Journal of Climate, American Meteorological Society, 2009, 23, 1696-1718.

**See Also**

[dist.start](#)

## Examples

```

##
## generate artificial data
##
set.seed(101)
n.years <- 60
date <- rep(1:n.years,each=12) + 1950 + rep((0:11)/12,times=n.years)
## Precipitation
PRECIP <- (0.25*sin( 2 * pi * date) + 0.3)*rgamma(n.years*12, shape = 3, scale = 1)
PRECIP[PRECIP<0.1] <- 0
## Potential Evapotranspiration
PET <- 0.5*sin( 2 * pi * date)+1.2+rnorm(n.years*12,0,0.2)
## display test data
matplot(date,cbind(PRECIP,PET),t=c("h","l"),col=c("blue","red"),lty=1)
legend("topright",legend=c("PRECIPitation","temperature"),fill=c("blue","red"))

##
## example SPI
##
spi.para <- fitSCI(PRECIP,first.mon=1,distr="gamma",time.scale=6,p0=TRUE)
spi.para
spi <- transformSCI(PRECIP,first.mon=1,obj=spi.para)
plot(date,spi,t="l")

##
## effect of time.scale on SPI
##
spi.1.para <- fitSCI(PRECIP,first.mon=1,time.scale=1,distr="gamma",p0=TRUE)
spi.12.para <- fitSCI(PRECIP,first.mon=1,time.scale=12,distr="gamma",p0=TRUE)
spi.1 <- transformSCI(PRECIP,first.mon=1,obj=spi.1.para)
spi.12 <- transformSCI(PRECIP,first.mon=1,obj=spi.12.para)
matplot(date,cbind(spi.1,spi.12),t="l",lty=1,col=c("red","blue"),lwd=c(1,2))
legend("topright",legend=c("time.scale=1","time.scale=12"),fill=c("red","blue"))

##
## example SPEI
##
if(require(evd)){
  spei.para <- fitSCI(PRECIP-PET,first.mon=1,time.scale=6,distr="gev",p0=FALSE)
  spei <- transformSCI(PRECIP-PET,first.mon=1,obj=spei.para)
  plot(date,spei,t="l")
}

##
## effect of changing different distribution for SPEI computation
##
spei.genlog.para <- fitSCI(PRECIP-PET,first.mon=1,time.scale=6,distr="genlog",p0=FALSE)
spei.genlog <- transformSCI(PRECIP-PET,first.mon=1,obj=spei.genlog.para)
if(require(evd)){lines(date,spei.genlog, col="red")} else {plot(date,spei.genlog,t="l")}
## in this case: only limited effect.
## generally: optimal choice of distribution: user responsibility.

```

```

##
## use a 30 year reference period for SPI parameter estimation
##
sel.date <- date>=1970 & date<2000
spi.ref.para <- fitSCI(PRECIP[sel.date],first.mon=1,distr="gamma",time.scale=6,p0=TRUE)
## apply the the parameters of the reference period to all data
## also outside the reference period
spi.ref <- transformSCI(PRECIP,first.mon=1,obj=spi.ref.para)
plot(date,spi.ref,t="l",col="blue",ylim=c(-5,5),lwd=2)
lines(date[sel.date],spi.ref[sel.date],col="red",lwd=3)
legend("bottom",legend=c("reference period","extrapolation period"),fill=c("red","blue"),
      horiz=TRUE)

##
## use "start.fun.fix" in instances where maximum likelyhood estimation fails
##
## force failure of maximum likelyhood estimation by adding "strange" value
## a warning should be issued
xx <- PRECIP-PET; xx[300] <- 1000
spei.para <- fitSCI(xx,first.mon=2,time.scale=1,p0=FALSE,distr="gev")
spei.para$dist.para
## use start.fun, usually ment for estimating inital values for
## parameter optimisation if maximum likelihood estimation fails
spei.para <- fitSCI(xx,first.mon=2,time.scale=1,p0=FALSE,distr="gev",
                  start.fun.fix=TRUE)
spei.para$dist.para

##
## usage of sci.limit to truncate unrealistic SCI values
##
PRECIP.mod <- PRECIP
PRECIP.mod[300] <- 100 ## introduce spuriously large value
spi.mod.para <- fitSCI(PRECIP.mod,first.mon=1,time.scale=3,p0=TRUE,distr="gamma")
plot(transformSCI(PRECIP.mod,first.mon=1,obj=spi.mod.para,sci.limit=Inf),
      t="l",col="blue",lwd=2)
lines(transformSCI(PRECIP.mod,first.mon=1,obj=spi.mod.para,sci.limit=4),col="red")

##
## how to modify settings of function "mledist" used for parameter identification
##
## identify parameters with standard settings
spi.para <- fitSCI(PRECIP,first.mon=1,distr="gamma",time.scale=6,p0=TRUE)
## add lower and upper limits for parameter identification
lower.lim <- apply(spi.para$dist.para,1,min) - 0.5*apply(spi.para$dist.para,1,sd)
upper.lim <- apply(spi.para$dist.para,1,max) + 0.5*apply(spi.para$dist.para,1,sd)
spi.para.limit <- fitSCI(PRECIP,first.mon=1,distr="gamma",time.scale=6,p0=TRUE,
                       mledist.par=list(lower=lower.lim, upper=upper.lim))

##
## how to write an own start.fun
## (required if distributions not mentioned in "dist.start" are used)
##
## function with same arguments as "dist.start"

```



```

my.start <- function(x,distr="gamma"){
### code based on "mmedist" in package "fitdistrplus"
  ppar <- try({
    n <- length(x)
    m <- mean(x)
    v <- (n - 1)/n * var(x)
    shape <- m^2/v
    rate <- m/v
    list(shape = shape, rate = rate)},TRUE)
  if (class(ppar) == "try-error") ## function has to be able to return NA parameters
    ppar <- list(shape = NA, rate = NA)
  return(ppar)
}
my.start(PRECIP)
spi.para <- fitSCI(PRECIP,first.mon=1,time.scale=6,p0=TRUE,
  distr="gamma",start.fun=my.start)

```

---

genlog

*Generalized Logistic Distribution*


---

## Description

Density, distribution and quantile function of the generalized logistic distribution

## Usage

```

pgenlog(q, shape, scale, location)
dgenlog(x, shape, scale, location)
qgenlog(p, shape, scale, location)

```

## Arguments

x, q	vector of quantiles.
p	vector of probabilities.
shape	shape parameter
scale	scale parameter
location	location parameter

## Details

The functions of the genlog family are a reimplementaion of the Generalized Logistic Distribution in the **lmomco** package, making the code compatible with the standard nomenclature for distribution in R. The original functions in **lmomco** are [pdfglo](#) (density function), [quaglo](#) (quantile function) and [cdfglo](#) (distribution function).

## Value

dgenlog gives the density (pdf), pgenlog gives the distribution function (cdf), and qgenlog gives the quantile function (inverse cdf).

**Author(s)**

James Stagge & Lukas Gudmundsson

**References**

Asquith, W.H., 2013: *lmomco* – L-moments, trimmed L-moments, L-comoments, censored L-moments, and many distributions. R package version 1.7.8 , Tech University, Lubbock, Texas.

**Examples**

```
dgenlog(1, shape=1, scale=2, location=3)
```

---

pe3

*Pearson Type III distribution*

---

**Description**

Density, distribution and quantile function of the Pearson Type III distribution

**Usage**

```
dpe3(x, shape, scale, location)
ppe3(q, shape, scale, location)
qpe3(p, shape, scale, location)
```

**Arguments**

x, q	vector of quantiles.
p	vector of probabilities.
shape	shape parameter
scale	scale parameter
location	location parameter

**Details**

The functions of the `pe3` family are a reimplement of the Pearson Type III Distribution in the **lmomco** package, making the code compatible with the standard nomenclature for distributions in R. The original functions in **lmomco** are [pdfpe3](#) (density function), [quape3](#) (quantile function) and [cdfpe3](#) (distribution function).

**Value**

`dpe3` gives the density (pdf), `ppe3` gives the distribution function (cdf), and `qpe3` gives the quantile function (inverse cdf).

**Author(s)**

James Stagge & Lukas Gudmundsson

**References**

Asquith, W.H., 2013: Imomco – L-moments, trimmed L-moments, L-comoments, censored L-moments, and many distributions. R package version 1.7.8 , Tech University, Lubbock, Texas.

**Examples**

```
dpe3(1, shape=1, scale=2, location=3)
```

# Index

## \* **distribution**

genlog, 9

pe3, 10

## \* **package**

SCI-package, 2

## \* **ts**

fitSCI, 4

cdfglo, 9

cdfpe3, 10

dgenlog (genlog), 9

dist.start, 3, 5, 6

dpe3 (pe3), 10

fitdistrplus, 5

fitSCI, 4, 5

gamma, 3

GammaDist, 3, 4

genlog, 3, 9

gev, 3

gumbel, 3

lmom.start (dist.start), 3

lnorm, 3

logis, 3

mledist, 5, 6

mom.start (dist.start), 3

norm, 3

pdfglo, 9

pdfpe3, 10

pe3, 3, 10

pgenlog (genlog), 9

ppe3 (pe3), 10

qgenlog (genlog), 9

qpe3 (pe3), 10

quaglo, 9

quape3, 10

SCI (SCI-package), 2

SCI-package, 2

transformSCI, 6

transformSCI (fitSCI), 4

weibull, 3