

# Package: loopevd (via r-universe)

May 17, 2026

**Title** Loop Functions for Extreme Value Distributions

**Version** 1.0.2

**Description** Performs extreme value analysis at multiple locations using functions from the 'evd' package. Supports both point-based and gridded input data using the 'terra' package, enabling flexible looping across spatial datasets for batch processing of generalised extreme value, Gumbel fits.

**License** GPL (>= 3)

**Imports** terra, evd, ncdf4, stats, utils, ismev, parallel

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Suggests** knitr, rmarkdown, zoo, ozmaps, raster, sp, methods

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Julian O'Grady [aut, cre]

**Maintainer** Julian O'Grady <julian.ogrady@csiro.au>

**Config/pak/sysreqs** libgdal-dev gdal-bin libgeos-dev libnetcdf-dev libproj-dev libsqlite3-dev

**Repository** <https://julianog.r-universe.dev>

**Date/Publication** 2025-07-21 04:50:02 UTC

**RemoteUrl** <https://github.com/cran/loopevd>

**RemoteRef** HEAD

**RemoteSha** f665f441b1d1c6194694bf84b558c0230f05630c

## Contents

add_nc_atts . . . . .	2
annual_max . . . . .	3
centredAndScaled . . . . .	5
df_cievd . . . . .	5
df_fevd . . . . .	6

df_qevd . . . . .	8
df_to_netcdf . . . . .	8
evd_params . . . . .	10
list_fevd . . . . .	11
netcdf_to_df . . . . .	12
plot_empirical . . . . .	13
qevd_vector . . . . .	13
raster_fevd . . . . .	14
raster_qevd . . . . .	15
raster_se_sig . . . . .	16
se_sig . . . . .	17

<b>Index</b>	<b>19</b>
--------------	-----------

---

add_nc_atts	<i>Add Global Attribute Metadata to Netcdf File</i>
-------------	---

---

## Description

Improve FAIR metadata record with consideration of CF conventions <https://cfconventions.org/Data/cf-conventions/cf-conventions-1.10/cf-conventions.html>

## Usage

```
add_nc_atts(
  outfile,
  r,
  creator_name = "",
  creator_email = "",
  references = "",
  title = "",
  summary = "",
  keywords = "",
  history = "",
  licence = "",
  Disclaimer = ""
)
```

## Arguments

outfile	character the file to be edited
r	SpatRaster the dataset in raster format e.g. r = terra::rast(outfile)
creator_name	character, optional. The name of the person (or other creator type specified by the creator_type attribute) principally responsible for creating this data.
creator_email	character, optional. The email address of the person (or other creator type specified by the creator_type attribute) principally responsible for creating this data.

references	character, optional. Published or web-based references that describe the data or methods used to produce it. Recommend URIs (such as a URL or DOI) for papers or other references. This attribute is defined in the CF conventions.
title	character, optional. A short phrase or sentence describing the dataset. In many discovery systems, the title will be displayed in the results list from a search, and therefore should be human readable and reasonable to display in a list of such names. This attribute is also recommended by the NetCDF Users Guide and the CF conventions.
summary	character, optional. A paragraph describing the dataset, analogous to an abstract for a paper.
keywords	character, optional. A comma-separated list of key words and/or phrases. Keywords may be common words or phrases, terms from a controlled vocabulary (GCMD is often used), or URIs for terms from a controlled vocabulary (see also "keywords_vocabulary" attribute).
history	character, optional. Provides an audit trail for modifications to the original data. This attribute is also in the NetCDF Users Guide: 'This is a character array with a line for each invocation of a program that has modified the dataset. Well-behaved generic netCDF applications should append a line containing: date, time of day, user name, program name and command arguments.' To include a more complete description you can append a reference to an ISO Lineage entity; see NOAA EDM ISO Lineage guidance.
licence	character, optional.
Disclaimer	character, optional.

### Value

see [https://wiki.esipfed.org/Attribute\\_Convention\\_for\\_Data\\_Discovery\\_1-3](https://wiki.esipfed.org/Attribute_Convention_for_Data_Discovery_1-3), [http://cfconventions.org/cf-conventions.html](http://cfconventions.org/cf-conventions/cf-conventions.html)

### Examples

```
## Not run:
tf = tempfile("test.nc")
tf
file.copy(system.file("extdata/50km_AnnMax_agcd_v1_tmax_mean_r005_daily_1980-2019.nc",
package = "loopevd"),tf)
r = terra::rast(tf)
add_nc_atts(tf,r,creator_name="add_nc_atts examples")
```

---

annual\_max

*Compute Annual Maximum and Mean of On-the-Hour Records*

---

### Description

annual\_max() takes a data frame of daily (or sub-daily) observations and returns a summary of the annual maximum and mean values, the date/time of each annual maximum, and the fraction of "on-the-hour" samples (data completeness) for each calendar year.

**Usage**

```
annual_max(DF, record_attribute = "sea_level")
```

**Arguments**

**DF** A data.frame containing at least:

- date: a Date or POSIXlt column of observation timestamps
- a numeric column of values

**record\_attribute** A character string giving the name of the column in DF containing the values. Defaults to "sea\_level".

**Details**

For each year, only observations exactly on the hour (minute == 0 & second == 0) are counted toward completeness. If no valid data exist for a year, that year is dropped from the output.

**Value**

a data.frame containing a date column and a numeric column (specified in record\_attribute) for years where at least one nonNA value is present, containing:

- annMax - the annual maximum
- annMean - the annual mean (calendar year)
- datestr - the date/time of the annual maximum, formatted "YYYYmmddHH"
- date - the POSIXlt timestamp of the annual maximum
- pc\_complete - the fraction (0 to 1) of hourly-timestamped samples available in that year

**Examples**

```
# generate example daily data
dates <- seq.Date(as.Date("1990-01-01"), as.Date("1995-12-31"), by = "day")
DF <- data.frame(
  date      = dates,
  sea_level = rnorm(length(dates), mean = 0, sd = 1)
)
# compute annual summary
am <- annual_max(DF)
head(am)
```

---

centredAndScaled	<i>Centre and Scale Numeric Data</i>
------------------	--------------------------------------

---

**Description**

centredAndScaled centres (subtracts the mean) and scales (divides by the standard deviation) each column of a numeric vector or matrix.

**Usage**

```
centredAndScaled(nsloc = NULL)
```

**Arguments**

nsloc            data.frame. If NULL or of length zero, the function returns nsloc unchanged.

**Details**

If nsloc has only one column, the function computes the mean and standard deviation of the entire vector. If nsloc has multiple columns, each column is centred and scaled independently.

**Value**

A numeric vector or matrix of the same dimensions as the input, with each column centred to mean zero and scaled to unit variance. If nsloc is NULL, returns NULL.

**Examples**

```
# Centre and scale a simple vector
centredAndScaled(data.frame(1:10))

# Centre and scale each column of a matrix
mat <- as.data.frame(matrix(stats::rnorm(30), nrow = 10, ncol = 3))
centredAndScaled(mat)
```

---

df_cievd	<i>Return a numeric vector of confidence intervals for EVD quantiles</i>
----------	--

---

**Description**

Return a numeric vector of confidence intervals for EVD quantiles

**Usage**

```
df_cievd(x, p, ci = 0.95, cores = 8)
```

**Arguments**

x	A data.frame of EVD parameters and associated covariance matrix. Must include 'loc', 'scale', 'shape', and 'cov*' column names.
p	A single probability value for the quantile (e.g. 0.99).
ci	Confidence level for the interval (default is 0.95).
cores	Number of parallel cores to use. If >1, parallel processing is used via <code>parallel::parLapply()</code> .

**Details**

This function calculates confidence intervals for extreme value quantiles using the delta method. The required input is a row-wise data frame with EVD parameters and their variance-covariance elements. Internally uses `ismev::gev.rl.gradient()` and `ismev::q.form()`.

**Value**

A numeric vector giving the confidence interval widths for each row in x.

**See Also**

[ismev::gev.rl.gradient\(\)](#), [ismev::q.form\(\)](#)

**Examples**

```
## Not run:
df <- data.frame(loc = 1, scale = 0.5, shape = 0.1,
                 cov_1 = 0.01, cov_2 = 0.001, cov_3 = 0.002,
                 cov_4 = 0.001, cov_5 = 0.01, cov_6 = 0.001,
                 cov_7 = 0.002, cov_8 = 0.001, cov_9 = 0.02)
df_cievd(df, p = 0.99, ci = 0.95, cores = 2)

## End(Not run)
```

---

df\_fevd

---

*Fit EVD Parameters to Rows of a Data Frame of Annual Maxima*


---

**Description**

Fits an extreme value distribution (EVD) to each row of a data.frame, where each row represents a time series of annual maxima. Optionally includes non-stationary covariates in the location parameter. Parallel processing is used to improve efficiency.

**Usage**

```
df_fevd(
  varbIN,
  cores = 1,
  ntries = 1,
  evd_mod_str = "fgumbel",
  nsloc = NULL,
  seed = NULL
)
```

**Arguments**

varbIN	A numeric data.frame where each row is a vector of annual maxima values (e.g., one row per location or site).
cores	An integer specifying the number of parallel cores to use for computation.
ntries	The integer number of tries to fit the evd
evd_mod_str	A character string specifying the EVD fitting function to use. Must be one of "fgumbel", "fgumbelx", or "fgev".
nsloc	Optional data.frame of covariates for non-stationary location modelling. Must have the same number of rows as varbIN.
seed	Optional integer for reproducibility. If supplied, a random seed will be set before model fitting.

**Details**

Each row of varbIN is passed to `evd_params()`, optionally with a corresponding row of covariates from nsloc. Covariates are internally standardised if provided. The function constructs a blank parameter structure (`empty_evd_params`) to ensure consistent output structure, even if fitting fails.

Parallel computation is performed using [parLapply](#).

**Value**

A data.frame with one row per input time series, containing the fitted EVD parameters returned by `evd_params()`.

**See Also**

[list\\_fevd](#) to apply EVD fitting to a list of annual maxima, [evd\\_params](#) for direct fitting to a single time series. #

**Examples**

```
set.seed(42)
df <- as.data.frame(matrix(evd::rgev(1000), nrow = 100))
results <- df_fevd(df, cores = 1, evd_mod_str = "fgumbel")
```

---

df\_qevd *Return a vector of EVD Quantiles*

---

### Description

Return a vector of EVD Quantiles

### Usage

```
df_qevd(x, p, evd_mod_str, interval = NULL, lower.tail = TRUE, cores = 1)
```

### Arguments

x	A data.frame of EVD parameters
p	Probability value (e.g. 0.99)
evd_mod_str	One of "fgumbel", "fgev", or "fgumbelx"
interval	Optional; interval passed to uniroot when needed
lower.tail	Logical; if TRUE (default), computes $P(x \leq y)$
cores	Number of cores to use for parallel processing (default is 1)

### Value

A numeric vector of quantiles

---

df\_to\_netcdf *Convert Data Frame to NetCDF*

---

### Description

This function converts a data frame to a NetCDF file for a list of points (rows are different stations). It attempts to identify CF-compliant coordinate variables, such as latitude and longitude, using default or specified column names.

### Usage

```
df_to_netcdf(
  df,
  output_file,
  lat_var = "lat",
  lon_var = "lon",
  global_atts = list(),
  units = list(),
  longnames = list(),
  cf_standard_names = list()
)
```

**Arguments**

df	A data frame containing the data to be converted.
output_file	The path to the output NetCDF file.
lat_var	The name of the latitude variable in the data frame. Default is "lat".
lon_var	The name of the longitude variable in the data frame. Default is "lon".
global_atts	A list of global attributes to add to the NetCDF file. Default is an empty list.
units	A list of units for each variable in the data frame. Default is an empty list.
longnames	A list of long names for each variable in the data frame. Default is an empty list.
cf_standard_names	A list of CF standard names for each variable in the data frame. Default is an empty list.

**Value**

None. The function writes the data to a NetCDF file.

**Examples**

```
# Example data frame
example_df <- data.frame(
  lat = c(-35.0, -34.5, -34.0),
  lon = c(150.0, 150.5, 151.0),
  variable1 = c(0.1, 0.2, 0.3),
  variable2 = c(0.4, 0.5, 0.6)
)

# Define units, longnames, and CF standard names
units_list <- list(variable1 = "m", variable2 = "cm")
longnames_list <- list(variable1 = "Variable 1 Longname",
  variable2 = "Variable 2 Longname")
cf_standard_names_list <- list(variable1 = "sea_surface_height",
  variable2 = "sea_water_temperature")

tf <- tempfile("test.nc")
# Convert example data frame to NetCDF
df_to_netcdf(example_df, tf, global_atts = list(
  title = "Example NetCDF",
  summary = "This is a test NetCDF file created from an example data frame.",
  source = "Example data",
  references = "N/A"
), units = units_list, longnames = longnames_list,
  cf_standard_names = cf_standard_names_list)
```

---

 evd\_params

*Function to return the EVD (Gumbel or GEV) parameters as a vector.*


---

**Description**

Function to return the EVD (Gumbel or GEV) parameters as a vector.

**Usage**

```
evd_params(
  x,
  evd_mod_str,
  nsloc = NULL,
  empty_evd_params,
  ntries = 3,
  silent = FALSE,
  returncs = TRUE
)
```

**Arguments**

x	numeric vector of data to be fitted.
evd_mod_str	either a string "fgumbel", "fgumbelx" or "fgev" from the extreme value distribution (evd) in the evd package
nsloc	A data frame with the same number of rows as the length of x, for linear modelling of the location parameter. The data frame is treated as a covariate matrix (excluding the intercept). A numeric vector can be given as an alternative to a single column data frame.
empty_evd_params	A preallocated vector or array used to store the return value when fitting fails
ntries	number of tries to fit the evd
silent	logical: should the report of error messages be suppressed?
returncs	logical: should the centered and scaled values be returned

**Value**

a vector of estimate, var.cov, AIC, centered and scaled values

**Examples**

```
# Ten records of 20 random data generated from the fgumbel EVD
am = lapply(1:10, function(x) evd::rgumbel(20))
tab = as.data.frame(t(sapply(am,function(x) evd_params(x,"fgumbel"))))
```

list\_fevd

*Fit EVD Parameters to a List of Annual Maxima***Description**

Fits an extreme value distribution to each element of a list of annual maxima series, optionally using non-stationary covariates, and returns a table of fitted parameters.

**Usage**

```
list_fevd(
  lst,
  evd_mod_str,
  nsloc = NULL,
  outfile = NULL,
  pc_complete = 0.8,
  minyear = 1800,
  maxyear = 2100,
  mslAtt = "annMean"
)
```

**Arguments**

lst	A list of data.frames, each as returned by <code>annual_max()</code> , containing at least: <ul style="list-style-type: none"> <li>• annMax: annual maximum values</li> <li>• annMean: annual mean values</li> <li>• datestr: timestamp strings for the maxima</li> <li>• date: POSIX timestamps for the maxima</li> <li>• pc_complete: completeness fraction per year</li> <li>• zero: the zero level</li> <li>• nsloc: optional covariate matrix used in non-stationary models.</li> </ul> If non-stationary fitting is required, each element may also include an nsloc matrix of covariates.
evd_mod_str	A character string specifying which fitting function from <b>evd</b> to use: "fgumbel", "fgumbelx" or "fgev".
nsloc	Optional matrix of covariates for non-stationary location modelling. Must have the same number of rows as years retained after filtering.
outfile	Optional character path to a NetCDF file in which to write the results (not currently implemented).
pc_complete	Numeric scalar (0-1). Minimum completeness fraction for a year to be included. Defaults to 0.8.
minyear	Numeric. Minimum calendar year to include. Defaults to 1800.
maxyear	Numeric. Maximum calendar year to include. Defaults to 2100.
mslAtt	character. Name of the attribute to be removed from annMax in each data.frame (e.g. "annMean" or "zero"). Defaults to "annMean".

**Value**

A data.frame with one row per list element, containing the parameters returned by `evd_params()` for each annual-max series.

**Examples**

```
dates = seq.Date(as.Date("1990-01-01"), as.Date("2019-12-31"), "day")
lst = lapply(1:10, function(x) loopevd::annual_max(data.frame(date = dates,
  sea_level = stats::rnorm(length(dates), mean=x/10, sd = x),
  zero = rep(0, length(dates)))))
loopevd::list_fevd(lst, "fgumbel", pc_complete=0)
```

---

netcdf\_to\_df

*Read Station-based evd NetCDF File into a Data Frame*


---

**Description**

This function reads a NetCDF file with EVD parameters (e.g. loc and scale) with a station dimension and converts it back into a data frame. The function assumes that the NetCDF file has a 'station' dimension with associated variables such as latitude, longitude, and other station-specific data.

**Usage**

```
netcdf_to_df(netcdf_file, exclude_cov = FALSE)
```

**Arguments**

<code>netcdf_file</code>	The path to the NetCDF file.
<code>exclude_cov</code>	Logical, if TRUE, variables starting with 'cov' will be excluded. Default is FALSE.

**Value**

A data frame containing the data from the NetCDF file.

**Examples**

```
tf = tempfile("test.nc")
# Ten records of 20 random data generated from the fgumbel EVD
am = lapply(1:10, function(x) evd::rgumbel(20))
tab = as.data.frame(t(sapply(am, function(x) evd_params(x, "fgumbel"))))
tab$lon = rnorm(10, sd=10) #station latitude
tab$lat = rnorm(10, sd=20) #station longitude
loopevd::df_to_netcdf(df = tab, output_file = tf)
tab2 = loopevd::netcdf_to_df(tf)
```

---

plot\_empirical                    *Plot the Empirical Return Level Data*

---

**Description**

Plot the Empirical Return Level Data

**Usage**

```
plot_empirical(x, xns = NULL, unitz = "-", ...)
```

**Arguments**

x	A numeric vector, which may contain missing values.
xns	A numeric vector, corrected for the non-stationary change in location, which may contain missing values.
unitz	y-label
...	parameters sent to base::plot

**Value**

r

**Examples**

```
ns = seq(-1,1,,50)
x = evd::rgev(50,loc=3)+ns
xns = x-ns
plot_empirical(x,xns)
```

---

qevd\_vector                    *Return a Vector of EVD Quantiles*

---

**Description**

Return a Vector of EVD Quantiles

**Usage**

```
qevd_vector(x, p, evd_mod_str, interval = NULL, lower.tail = TRUE, nams = NULL)
```

**Arguments**

<code>x</code>	vector of EVD parameters
<code>p</code>	vector of probabilities.
<code>evd_mod_str</code>	either a string "fgumbel", "fgev" or "fgumbelx" from the extreme value distribution (evd) in the evd package
<code>interval</code>	A length two vector containing the end-points of the interval to be searched for the quantiles, passed to the uniroot function.
<code>lower.tail</code>	Logical; if TRUE (default), $P(x \leq y)$ , otherwise $P(X > x)$ .
<code>nams</code>	names of the values of <code>x</code> (optional)

**Value**

gives the quantile function corresponding to `p`

**See Also**

[evd::qgev\(\)](#), [evd::qgumbelx\(\)](#)

**Examples**

```
qevd_vector(c(1,0.5),1-0.05,"fgumbel",nams = c("loc","scale"))
df = data.frame(loc = 1,scale = 0.5)
qevd_vector(df,1-0.05,"fgumbel")
```

---

raster_fevd	<i>Turn Raster of Annual Maximums into Extreme Value Distributions parameters for Netcdf Output</i>
-------------	---

---

**Description**

Turn Raster of Annual Maximums into Extreme Value Distributions parameters for Netcdf Output

**Usage**

```
raster_fevd(
  r,
  evd_mod_str,
  nsloc = NULL,
  outfile = NULL,
  cores = 1,
  ntries = 1,
  silent = FALSE,
  seed = NULL
)
```

**Arguments**

r	SpatRaster
evd_mod_str	either a string "fgumbel", "fgev" or "fgumbelx" from the extreme value distribution (evd) in the evd package
nsloc	A data frame with the same number of rows as the length of x, for linear modelling of the location parameter. The data frame is treated as a covariate matrix (excluding the intercept). A numeric vector can be given as an alternative to a single column data frame.
outfile	filename to write to netcdf
cores	positive integer. If cores > 1, a 'parallel' package cluster with that many cores is created and used. You can also supply a cluster object. Ignored for functions that are implemented by terra in C++ (see under fun)
ntries	The integer number of tries to fit the evd
silent	logical: should the report of error messages be suppressed?
seed	set the seed for fitting.

**Value**

the parameters of the evd in a SpatRasterDataset

**See Also**

[evd::fgev\(\)](#), [evd::fgumbelx\(\)](#)

**Examples**

```
require(terra)
r = rast(system.file("extdata/50km_AnnMax_agcd_v1_tmax_mean_r005_daily_1980-2019.nc",
  package = "loopevd"))
r2 = aggregate(r,4) #lower the resolution for a fast example
gumbel_r = raster_fevd(r2,"fgumbel",seed = 1)
plot(gumbel_r$loc,main = "location")
```

---

raster\_qevd

*Return a raster of EVD Quantiles*


---

**Description**

Return a raster of EVD Quantiles

**Usage**

```
raster_qevd(x, p, evd_mod_str, interval = NULL, lower.tail = TRUE)
```

**Arguments**

<code>x</code>	SpatRasterDataset of EVD parameters, e.g. loc, scale, shape
<code>p</code>	probability value.
<code>evd_mod_str</code>	either a string "fgumbel", "fgev" or "fgumbelx" from the extreme value distribution (evd) in the evd package
<code>interval</code>	A length two vector containing the end-points of the interval to be searched for the quantiles, passed to the uniroot function.
<code>lower.tail</code>	Logical; if TRUE (default), probabilities are $P(x \leq y)$ , otherwise $P(X > x)$ .

**Value**

gives the quantile function corresponding to p

**See Also**

[evd::qgev\(\)](#), [evd::qgumbelx\(\)](#)

**Examples**

```
require(terra)
r = rast(system.file("extdata/50km_AnnMax_agcd_v1_tmax_mean_r005_daily_1980-2019.nc",
                    package = "loopevd"))
r2 = aggregate(r,4) #lower the resolution for a fast example
gumbel_r = raster_fevd(r2,"fgumbel")
AEP_10pc = raster_qevd(gumbel_r,1-0.1,"fgumbel") # 10% Annual Exceedance Probability.
```

---

raster_se_sig	<i>Calculate One-Sided Confidence Level (%)</i>
---------------	---

---

**Description**

Computes the one-sided confidence level, defined as  $(1 - p\text{-value}) \times 100$ , for testing whether each mean ( $\mu$ ) differs from zero under a normal approximation.

**Usage**

```
raster_se_sig(muvari)
```

**Arguments**

<code>muvari</code>	SpatRaster, of mean (location) values, variances corresponding to each $\mu$ to test against zero.
---------------------	--

**Details**

For each element:

1. Calculate the standard error:  $se = \sqrt{vari}$ .
2. Compute the absolute z-score:  $z = \text{abs}(\mu / se)$ .
3. The one-sided p-value is  $1 - \text{phi}(z)$ , where phi is the CDF of the standard normal.
4. The confidence level is  $(1 - \text{p-value}) \times 100 = \text{phi}(z) \times 100$ .

**Value**

A SpatRaster of confidence levels (0-100%), each rounded to one decimal place.

**Examples**

```
require(terra)
r = rast(system.file("extdata/50km_AnnMax_agcd_v1_tmax_mean_r005_daily_1980-2019.nc",
                    package = "loopevd"))
r2 = aggregate(r,4) #lower the resolution for a fast example
gev_r = raster_fevd(r2,"fgev")
raster_se_sig(c(gev_r$shape,gev_r$cov_9))
```

---

se\_sig

---

*Calculate One-Sided Confidence Level (%)*


---

**Description**

Computes the one-sided confidence level, defined as  $(1 - \text{p-value}) \times 100$ , for testing whether each mean ( $\mu$ ) differs from zero under a normal approximation.

**Usage**

```
se_sig(muvari)
```

**Arguments**

muvari            Numeric array, of mean (location) values, variances corresponding to each mu to test against zero.

**Details**

For each element:

1. Calculate the standard error:  $se = \sqrt{vari}$ .
2. Compute the absolute z-score:  $z = \text{abs}(\mu / se)$ .
3. The one-sided p-value is  $1 - \text{phi}(z)$ , where phi is the CDF of the standard normal.
4. The confidence level is  $(1 - \text{p-value}) \times 100 = \text{phi}(z) \times 100$ .

**Value**

A numeric vector of confidence levels (0-100%), each rounded to one decimal place.

**Examples**

```
# Single value
se_sig(muvari = cbind(2,1))

# Vector of values
se_sig(muvari = cbind(c(-1, 0, 1),c(1, 2, 3)))
```

# Index

`add_nc_atts`, 2  
`annual_max`, 3  
  
`centredAndScaled`, 5  
  
`df_cievd`, 5  
`df_fevd`, 6  
`df_qevd`, 8  
`df_to_netcdf`, 8  
  
`evd::fgev()`, 15  
`evd::fgumbelx()`, 15  
`evd::qgev()`, 14, 16  
`evd::qgumbelx()`, 14, 16  
`evd_params`, 7, 10  
  
`ismev::gev.rl.gradient()`, 6  
`ismev::q.form()`, 6  
  
`list_fevd`, 7, 11  
  
`netcdf_to_df`, 12  
  
`parLapply`, 7  
`plot_empirical`, 13  
  
`qevd_vector`, 13  
  
`raster_fevd`, 14  
`raster_qevd`, 15  
`raster_se_sig`, 16  
  
`se_sig`, 17