Package ‘eFCM’

September 9, 2025

Type Package

Title Exponential Factor Copula Model

Version 1.0

Maintainer Mengran Li <m.li.3@research.gla.ac.uk>

Description Implements the exponential Factor Copula Model (eFCM) of Castro-Camilo, D. and Huser, R. (2020) for spatial extremes, with tools for dependence estimation, tail inference, and visualization. The package supports likelihood-based inference, Gaussian process modeling via Matérn covariance functions, and bootstrap uncertainty quantification. See Castro-Camilo and Huser (2020) <doi:10.1080/01621459.2019.1647842>.

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

Imports Rcpp, nsRFA, ismev, fields, mnormt, numDeriv, pbmcapply, boot, progress

LinkingTo Rcpp, RcppArmadillo

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation yes

LazyData false

Author Mengran Li [aut, cre],
Daniela Castro-Camilo [aut]

Repository CRAN

Date/Publication 2025-09-09 13:00:06 UTC

Contents

`eFCM-package` *eFCM: Exponential Factor Copula Model*

Description

Implements the exponential Factor Copula Model (eFCM) of Castro-Camilo and Huser (2020) for spatial extremes, with tools for dependence estimation, tail inference,and visualization. The package supports likelihood-based inference, Gaussian process modeling via Matérn covariance functions, and bootstrap uncertainty quantification.

References

Castro-Camilo, D. & Huser, R. (2020). Local likelihood estimation of complex tail dependence structures, with application to U.S. precipitation extremes. *Journal of the American Statistical Association*, **115**(531), 1037–1054. doi:10.1080/01621459.2019.1611584

Li, M. & Castro-Camilo, D. (2025). On the importance of tail assumptions in climate extreme event attribution. *arXiv*. doi:10.48550/arXiv.2507.14019

AIC.fcm *Akaike Information Criterion (AIC) for fcm objects*

Description

Compute the AIC value for a fitted `fcm` model using the formula:

$$\text{AIC} = -2 \cdot \log L + k \cdot p$$

where L is the likelihood, p is the number of parameters, and k is a penalty parameter.

Usage

```
## S3 method for class 'fcm'
AIC(object, ..., k = 2)
```

Arguments

`object`	An object of class `fcm`, created by `fcm()`.
`...`	Currently unused.
`k`	Penalty per parameter (default is `k = 2`).

Value

A numeric scalar giving the AIC value for the fitted model.

See Also

`logLik.fcm()`, `BIC.fcm()`, `AICc.fcm()`

AICc *Corrected Akaike Information Criterion (AICc) for fcm objects*

Description

Compute the AICc value for a fitted `fcm` model using the formula:

$$\text{AICc} = \text{AIC} + \frac{2p(p+1)}{n-p-1}$$

where n is the number of observations and p is the number of parameters.

Usage

```
AICc(object, ...)

## S3 method for class 'fcm'
AICc(object, ...)
```

Arguments

`object`	An object of class `fcm`, created by `fcm()`.
`...`	Currently unused.

Value

A numeric scalar giving the AICc value for the fitted model.

See Also

`AIC.fcm()`, `BIC.fcm()`

`BIC.fcm` *Bayesian Information Criterion (BIC) for fcm objects*

Description

Compute the BIC value for a fitted `fcm` model using the formula:

$$\text{BIC} = -2 \cdot \log L + \log(n) \cdot p$$

where n is the number of observations and p is the number of parameters.

Usage

```
## S3 method for class 'fcm'
BIC(object, ...)
```

Arguments

`object`	An object of class `fcm`, created by `fcm()`.
`...`	Currently unused.

Value

A numeric scalar giving the BIC value for the fitted model.

See Also

`AIC.fcm()`, `AICc.fcm()`

cf_data *Counterfactual daily precipitation data*

Description

An example dataset stored as an object of class "fdata", suitable for direct use with fcm

Usage

```
data(cf_data)
```

Format

An object of class "fdata"

Examples

```
data(cf_data)
dim(cf_data)
```

chi *Tail Dependence Coefficient (Chi Statistic)*

Description

Compute the conditional exceedance probability $\chi_h(u)$, either from a fitted eFCM model (chi.fcm) or empirically (Echi). $\chi_h(u)$ measures the probability of simultaneous exceedances at high but finite thresholds.

Usage

```
## S3 method for class 'fcm'
chi(object, h, u = 0.95, ...)

Echi(object, which = c(1, 2), u = 0.95)
```

Arguments

object	an object of class "fcm", created by fcm().
h	a positive numeric value representing the spatial distance (in kilometers).
u	a numeric value between 0 and 1 specifying the quantile threshold. Default is 0.95.
...	currently ignored.
which	A length-two integer vector giving the indices of the columns in object$data to be used for the empirical chi calculation.

Details

For two locations s_1 and s_2 separated by distance h, with respective vector components $W(s_1)$ and $W(s_2)$, the conditional exceedance probability is defined as

$$\chi_h(u) \;=\; \lim_{u\to 1} \Pr\bigl(F_{s_1}(W(s_1)) > u \;\mid\; F_{s_2}(W(s_2)) > u\bigr).$$

For the eFCM, the conditional exceedance probability $\chi_{\mathrm{eFCM}}(u)$ can be computed as

$$\chi_{\mathrm{eFCM}}(u) = \frac{1 - 2u + \Phi_2\bigl(z(u), z(u); \rho\bigr) - 2\exp\Bigl(\frac{\lambda^2}{2} - \lambda\, z(u)\, \Phi_2\bigl(q; 0, \Omega\bigr)\Bigr)}{1-u}.$$

Here, $z(u) = F_1^{-1}(u;\lambda)$ is the marginal quantile function, $\Phi_2(\cdot,\cdot;\rho)$ denotes the bivariate standard normal CDF with correlation ρ, $q = \lambda(1-\rho)$, and Ω is the correlation matrix.

Value

A named numeric value, the chi statistic for given `h` and `u`.

Methods

- `chi.fcm()`: Model-based estimate from an object of class `"fcm"`.
- `Echi()`: Empirical estimate.

References

Castro-Camilo, D. and Huser, R. (2020). Local likelihood estimation of complex tail dependence structures, applied to US precipitation extremes. *Journal of the American Statistical Association*, 115(531), 1037–1054.

Examples

```
fit <- fcm(...)
chi(fit, h = 150, u = 0.95)
```

`chiplot.fcm` *Chi Plot for Fitted eFCM Model*

Description

Plots the eFCM conditional exceedance probability $\chi_h(u)$.

Usage

```
## S3 method for class 'fcm'
chiplot(
  object,
  h = NULL,
  method = c("default", "hessian", "boot"),
  ci = 0.95,
  emp = TRUE,
  which = c(1, 2),
  ...
)

chiplot(object, ...)
```

Arguments

`object`	An object of class `"fcm"` returned by `fcm()`.
`h`	A positive numeric distance in kilometers. If `NULL` and `emp = TRUE`, inferred from `coord[which, ]`.
`method`	Character. Method for computing confidence intervals. One of `"default"`, `"hessian"`, or `"boot"`.
`ci`	Confidence level for interval estimation.
`emp`	Logical. Whether to add empirical chi estimates.
`which`	Integer vector of length 2. Locations to compute empirical chi.
`...`	Further arguments passed to base plotting functions (e.g., `main`, `xlab`, `ylab`, etc.).

Value

A (invisible) list containing chi estimates and confidence bounds:

chi.u Estimated chi values.

chi.lower Lower confidence bounds (if applicable).

chi.upper Upper confidence bounds (if applicable).

chi.emp Empirical chi curve (if `emp = TRUE`).

h Distance used.

References

Castro-Camilo, D. and Huser, R. (2020). Local likelihood estimation of complex tail dependence structures, applied to US precipitation extremes. *Journal of the American Statistical Association*, 115(531), 1037–1054.

See Also

`chi()`, `Echi()`

coef.fcm *Extract Model Coefficients*

Description

Extract estimated model parameters from objects returned by `fcm()`. Optionally computes confidence intervals via either the observed Hessian (Delta method, on the log scale) or bootstrap sampling.

Usage

```
## S3 method for class 'fcm'
coef(object, ..., method = c("default", "hessian", "boot"), ci = 0.95)
```

Arguments

`object`	An object of class `"fcm"`, typically the result of `fcm()`.
`...`	Further arguments passed to or from other methods.
`method`	Character string specifying the method used to compute confidence intervals. One of `"default"` (point estimate only), `"hessian"` (Delta method on log-scale), or `"boot"` (percentile bootstrap).
`ci`	Confidence level for the interval estimation (e.g., 0.95). If `NULL`, no confidence interval is returned.

Details

If `method = "hessian"`, confidence intervals are constructed on the log scale using the Delta method, then exponentiated to return to the original parameter scale. If `method = "boot"`, confidence intervals are computed as empirical quantiles of the bootstrap replicates.

Value

If `method = "default"`, returns a named vector of parameter estimates. If `method = "hessian"` or `method = "boot"`, returns a `data.frame` with columns:

- `par`: the estimated parameter
- `lower`: lower bound of the CI
- `upper`: upper bound of the CI

See Also

`fcm()`

Examples

```
data(fit)
coef(fit)
coef(fit, method = "hessian", ci = 0.95)
coef(fit, method = "boot", ci = 0.95)
```

---

counterfactual *Weekly Maxima of Counterfactual Precipitation in Europe*

Description

Weekly maxima of precipitation under natural-only forcing over a European domain. This processed dataset is used in the vignettes and examples to illustrate model fitting and attribution mapping with eFCM.

Usage

```
data(counterfactual)
```

Format

A numeric matrix or data frame with dimensions (weeks × stations). Rows index consecutive winter-season weeks; columns correspond to stations. Units are millimetres.

Details

This dataset is derived from daily-resolution counterfactual simulations (not included in the package due to size constraints) by aggregating to weekly maxima. It is intended for examples, tests, and vignettes where a lightweight dataset is preferred.

Examples

```
data(counterfactual)
dim(counterfactual)
plot(apply(counterfactual, 1, mean), type = "l",
     xlab = "Week", ylab = "Mean precipitation (mm)")
```

fcm *Fit the exponential Factor Copula Model (eFCM)*

Description

Fits the eFCM at a specified grid point using local neighborhood data.

Usage

```
fcm(
  s,
  object,
  theta0 = c(2, 100),
  thres = 0.9,
  nu = 0.5,
  hessian = TRUE,
  control = list(),
  censorL = TRUE,
  boot = FALSE,
  R = 1000,
  progress = TRUE,
  lower = c(1, 1),
  upper = Inf,
  sample_prop = 0.9,
  sample_ids = NULL,
  parallel = FALSE,
  ncpus = 4,
  mc.set.seed = TRUE,
  ...
)
```

Arguments

`s`	A single integer specifying the grid point index.
`object`	An object of class `"fdata"`, typically created by `fdata()`.
`theta0`	A numeric vector of initial values for the copula parameters (λ, δ).
`thres`	A numeric scalar indicating the quantile-based threshold (default is 0.9).
`nu`	Numeric Matérn smoothness parameter.
`hessian`	Logical; whether to return the Hessian matrix. Default is TRUE.
`control`	A list of control options for `nlminb()`.
`censorL`	Logical; if TRUE (default), uses the censored likelihood.
`boot`	Logical; whether to perform bootstrap estimation (default `FALSE`).
`R`	Integer; number of bootstrap replicates if `boot = TRUE`.
`progress`	Logical; if `TRUE` (default), show a progress bar during bootstrap using **pbapply**.

lower, upper	Numeric vectors of parameter bounds for optimization.
sample_prop	Numeric in (0,1). Proportion of rows to sample in each replicate (default 0.9). Ignored if sample_ids is provided.
sample_ids	Optional list of integer vectors. Each element specifies the row indices to use for a bootstrap replicate; when supplied, overrides sample_prop.
parallel	Logical; if TRUE, run neighbourhood selection in parallel using **pbmcapply**. On Windows, pbmclapply will fall back to serial execution (progress still shown).
ncpus	Integer; number of worker processes when parallel = TRUE on Unix-alikes.
mc.set.seed	Logical; seed the RNG streams in workers (default TRUE). Effective on Unix-alikes; on Windows (serial fallback) it has no effect.
...	Additional arguments passed to bootstrap_fcm().

Details

The exponential Factor Copula Model (eFCM) assumes that the process $W(s) = Z(s) + V$, where $Z(s)$ is a zero-mean Gaussian process with correlation $\rho(h) = \exp(-h/\delta)$ and $V \sim \mathrm{Exp}(\lambda)$ is a latent common factor independent of $Z(s)$ and s. This leads to nontrivial tail dependence between spatial locations.

Value

An object of class "fcm", which is a list including:

pars	Estimated parameters.
hessian	Hessian matrix (if requested).
nllh	Negative log-likelihood.
data.u	Pseudo-uniform transformed data.
gridID	Location of the selected grid point.
arg	Model arguments (e.g., thres, nu).
neigh	Neighbourhood indices used for estimation.
coord	Coordinates of the locations.
data	Observed data matrix at selected locations.
boot	(optional) Matrix of bootstrap samples of parameter estimates.

References

Castro-Camilo, D. and Huser, R. (2020). Local likelihood estimation of complex tail dependence structures, applied to US precipitation extremes. *JASA*, 115(531), 1037–1054.

See Also

fdata, coef, summary

Examples

```
# Load precipitation data for counterfactual scenarios
data("counterfactual")
data("LonLat")
coord = LonLat  # station coordinates (longitude-latitude)

cf_data <- fdata(counterfactual, coord, cellsize = c(1, 1))
fit = fcm(s = 1, cf_data, boot = T, R = 1000)
```

`fdata` *Transform datasets for factor copula modeling*

Description

Prepares and organizes datasets for use with the exponential Factor Copula Model (eFCM). The function converts raw station-level observations and their spatial coordinates into an "fdata" object, which contains the data, grid structure, and neighborhood information required for model fitting with `fcm()`.

Usage

```
fdata(
  data,
  coord,
  grid = NULL,
  neigh = NULL,
  theta0 = NULL,
  cellsize = c(0.5, 0.5),
  parallel = TRUE,
  ncpus = 4,
  mc.set.seed = TRUE,
  ...
)
```

Arguments

`data`	A matrix or data.frame. Each column corresponds to a station, with rows containing observations (on the original scale).
`coord`	A two-column matrix or data frame of station coordinates (longitude and latitude), one row per station.
`grid`	Optional two-column matrix or data frame of grid locations (longitude, latitude) at which the model will be fitted. If `NULL` (default), a regular grid is generated based on `cellsize`.
`neigh`	Optional list of neighborhood station indices for each grid point. If `NULL`, neighborhoods are constructed using `neighborhood_HT()`.

theta0	Optional matrix or data.frame with two columns: initial lambda and delta. Must match number of stations.
cellsize	Numeric vector of length 1 or 2, specifying longitude and latitude resolution.
parallel	Logical; if TRUE, run neighbourhood selection in parallel using **pbmcapply**. On Windows, pbmclapply will fall back to serial execution (progress still shown).
ncpus	Integer; number of worker processes when parallel = TRUE on Unix-alikes.
mc.set.seed	Logical; seed the RNG streams in workers (default TRUE). Effective on Unix-alikes; on Windows (serial fallback) it has no effect.
...	Additional arguments passed to neighborhood_HT().

Value

An object of class "fdata", which is a list with elements:

data	Original input data
coord	Coordinates of stations
grid	Grid points with assigned IDs
neigh	List of neighbor station indices per grid point
theta0	Initial values matrix
N	Number of stations

See Also

fcm(), neighborhood_HT()

Examples

```
# Load precipitation data for counterfactual scenarios
data("counterfactual")
data("LonLat")
coord = LonLat  # station coordinates (longitude-latitude)
cf_data <- fdata(counterfactual, coord, cellsize = c(1, 1))
```

---

fit *Example fitted eFCM object*

Description

An example output of the fcm function, obtained by fitting the exponential Factor Copula Model to a subset of precipitation data.

Usage

```
data(fit)
```

Format

An object of class `"fcm"`

Examples

```
data(fit)
summary(fit)
```

`logLik.fcm` *Log-likelihood of a fitted factor copula model*

Description

Extract the log-likelihood value from a fitted `fcm` object.

Usage

```
## S3 method for class 'fcm'
logLik(object, ...)
```

Arguments

`object`	An object of class `fcm`, typically returned by `fcm()`.
`...`	Additional arguments (currently unused).

Value

A numeric value giving the log-likelihood of the fitted model.

See Also

`AIC.fcm()`, `BIC.fcm()`, `AICc.fcm()`

`LonLat` *Spatial coordinates of European stations*

Description

A dataset containing the longitude and latitude of monitoring stations used in the eFCM examples and vignettes.

Usage

```
data(LonLat)
```

Format

A data frame with n rows and 2 variables:

lon Longitude (decimal degrees, WGS84).

lat Latitude (decimal degrees, WGS84).

Examples

```
data(LonLat)
head(LonLat)
```

`neighborhood_HT` *Homogeneous neighborhood selection*

Description

Identifies homogeneous neighbors around a given grid point using a combination of the Hosking-Wallis (1993) and Anderson-Darling (1987) tests for marginal homogeneity.

Usage

```
neighborhood_HT(
  data,
  coord,
  s0,
  miles = FALSE,
  min.neigh = 5,
  max.neigh = 20,
  pr = 0.9,
  alpha = 0.05,
  dmax = 150,
  which.test = c(1, 2)
)
```

Arguments

`data`	A matrix or data.frame. Each column corresponds to a station, with rows containing observations (on the original scale).
`coord`	A two-column matrix or data frame of station coordinates (longitude and latitude), one row per station.
`s0`	Numeric vector of length 2: the longitude and latitude of the target grid point.
`miles`	Logical; whether to compute distance in miles (default: FALSE).
`min.neigh`	Minimum number of neighbors to accept (default: 5).
`max.neigh`	Maximum number of neighbors to test (default: 20).
`pr`	Probability threshold for quantile filtering (e.g. 0.9).

alpha — Significance level for homogeneity tests.

dmax — Maximum distance (in km) to consider.

which.test — Integer vector specifying which test(s) to run:

- 1 = HW test (Hosking–Wallis)
- 2 = AD test (Anderson–Darling)
- c(1, 2) = both tests

Value

A vector of station indices considered homogeneous with the grid point.

References

Castro-Camilo, D. and Huser, R. (2020). *JASA* 115, 1037–1054. Hosking, J. and Wallis, J. (1993). *Water Resour. Res.* 29, 271–281. Scholz, F.W. and Stephens, M.A. (1987). *JASA* 82, 918–924.

See Also

fdata()

Examples

```
neighborhood_HT(counterfactual, coord = LonLat, s0 = c(30, 39), which.test = c(1, 2))
```

---

pfcm — *The Distribution of Univariate Factor Copula Model*

Description

Density, distribution function, quantile function and random generation for the distribution of univariate factor copula model with rate parameter equal to lambda.

Usage

```
pfcm(w, lambda)

dfcm(w, lambda)

qfcm(u, lambda, tol = 1e-08, niter = 1000L)

rfcm(n, lambda)
```

Arguments

`w`	A numeric value representing the spatial process.
`lambda`	A numeric value representing rate parameter λ.
`u`	a numeric vector of probabilities, with values in the interval from 0 to 1, at which the quantile function is to be computed.
`tol`	a scalar indicating the desired level of numerical accuracy for the algorithm; default is 1e-9.
`niter`	a scalar indicating the maximum number of iterations.
`n`	an integer value specifying the number of samples to generate

Details

The univariate eFCM distribution is

$$F(w; \lambda) = \Phi(w) - exp(\lambda^2/2 - \lambda w)\Phi(w - \lambda),$$

where λ is the rate parameter.

Value

`dfcm` gives a numeric value representing the density of the factor copula model evaluated at `w`, `pfcm` gives a numeric value representing the CDF evaluated at `w`, `qfcm` gives the quantile function (QF) of the factor copula model. and `rfcm` generate a numeric vector of random samples drawn.

Examples

```
pfcm(w = 1, lambda = 0.5)
dfcm(w = 1, lambda = 0.5)
qfcm(u = 0.5, lambda = 0.5)
rfcm(n = 1000, lambda = 0.5)
```

`pmfcm` *CDF of the exponential Factor Copula Model (vector input)*

Description

Computes the eFCM-based $P(W \leq w)$ for a single d-dimensional vector w.

Usage

```
pmfcm(
  w,
  lambda,
  delta,
  dist = NULL,
```

```
  coord = NULL,
  smooth = 0.5,
  abseps = 1e-05,
  releps = 1e-05,
  maxpts = 25000,
  miles = FALSE
)
```

Arguments

`w`	Numeric vector of length d.
`lambda, delta`	Positive scalars: common-factor rate λ and range δ.
`dist`	Optional $d \times d$ distance matrix. If `NULL`, provide `coord`.
`coord`	Optional two-column matrix/data.frame of coordinates (lon, lat) to build `dist`.
`smooth`	Matérn smoothness ν (default `0.5`).
`abseps, releps`	Absolute/relative tolerances for the MVN CDF.
`maxpts`	Maximum number of function evaluations for the MVN CDF.
`miles`	Logical; passed to `fields::rdist.earth()` if `coord` is used.

Value

A single numeric CDF value in $[0, 1]$.

Examples

```
data(LonLat)
d <- 2
w <- rep(0.3, d)
pmfcm(w, lambda = 2, delta = 100, coord = LonLat[1:2, ])
```

`qqplot` *Q–Q Plot for Fitted Factor Copula Model*

Description

Produce a Q–Q plot comparing empirical exceedances to the fitted eFCM tail, with an optional GPD tail overlay for diagnostic comparison.

Usage

```
qqplot(object, ...)

## S3 method for class 'fcm'
qqplot(
  object,
  which = 1,
```

```
  gpd = TRUE,
  thres = 0.9,
  main = "Q-Q plot",
  xlab = "Theoretical quantiles (exceedances)",
  ylab = "Empirical exceedances",
  ...
)
```

Arguments

object	An object of class "fcm" returned by fcm().
...	Additional graphical arguments forwarded to plot().
which	Integer scalar. Station (column) index to plot.
gpd	Logical; if TRUE, add a GPD tail fit to the Q–Q plot.
thres	Numeric in (0,1); the probability threshold used to pick the empirical quantile u = quantile(x, thres). Defaults to 0.9.
main, xlab, ylab	Character. Graphical labels passed to plot().

Details

The function first selects a threshold u as the empirical thres-quantile of the selected station series x. It then forms exceedances $Y = X - u \mid X > u$, fits the eFCM (implicitly via qfcm() and a scalar λ estimate), and plots empirical exceedances against theoretical eFCM quantiles in the tail. If gpd=TRUE, a GPD is fitted to the exceedances (threshold 0) and its theoretical tail quantiles are added for visual comparison.

Value

A numeric vector of fitted eFCM theoretical tail quantiles, invisibly returned.

---

rmfcm — *Random generation from the exponential Factor Copula Model (eFCM)*

Description

Draws n samples from the eFCM.

Usage

```
rmfcm(
  lambda,
  delta,
  dist = NULL,
  coord = NULL,
  nu = 0.5,
```

```
    n = 5e+05,
    miles = FALSE,
    seed = NULL
  )
```

Arguments

lambda, delta	Positive scalars: rate λ and range δ.
dist	Optional $d \times d$ distance matrix. If NULL, provide coord.
coord	Optional two-column matrix/data.frame of station coordinates (lon, lat). Used to build dist via fields::rdist.earth().
nu	Matérn smoothness parameter (default 0.5).
n	Number of simulated rows (default 5e5).
miles	Logical passed to fields::rdist.earth (default FALSE).
seed	Optional integer seed for reproducibility.

Value

A numeric matrix of size n x d (rows = samples, cols = stations).

Examples

```
data(LonLat)
sim <- rmfcm(lambda = 2, delta = 100, coord = LonLat[1:2, ], n = 10000)
dim(sim)  # 10000 x 2
```

---

summary.fcm *Summarizing Factor Copula Model Fits*

Description

Summary method for objects of class "fcm", returned by fcm().

Usage

```
## S3 method for class 'fcm'
summary(object, ...)
```

Arguments

object	An object of class "fcm", typically output from fcm().
...	Additional arguments (ignored).

Value

Invisibly, a list with summary components:

- `grid`: the selected grid point
- `neighbors`: indices and coordinates of neighbors
- `coef`: parameter estimates with 95\
- `objective`: negative log-likelihood
- `information criteria`: `c(AIC, BIC, AICc)`
- `args`: fitting arguments

Examples

```
data(fit)
summary(fit)
```

Index