Package: evgmrf (via r-universe)

October 29, 2026

Type Package

Title Gaussian Markov Random Field Models for Extreme Values

Version 0.1.0

Description Fitting extreme value distributions with parameters that vary according to Gaussian Markov random fields (GMRFs). For details of distributions see Coles, S.G. (2001) <doi:10.1007/978-1-4471-3675-0>, GMRFs see Rue, H., & Held, L. (2005) <doi:10.1201/9780203492024>, and the fitting approach see Wood, S.N., Pya, N. & Safken, B. (2016) <doi:10.1080/01621459.2016.1180986>.

License GPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends R (>= 3.5.0)

Imports Rcpp (>= 1.0.10), Matrix (>= 1.6-0), methods, lattice, gridExtra, sparseinv

LinkingTo Rcpp, RcppArmadillo, RcppEigen, Matrix

Config/roxygen2/version 8.1.0

Repository https://byoungman.r-universe.dev

Date/Publication 2026-10-16 13:56:38 UTC

RemoteUrl https://github.com/byoungman/evgmrf

RemoteRef HEAD

RemoteSha 93c7522b8fd4c55715ced495f72be171cb3689df

Contents

COcnty 2
coef.evgmrf 3
COexc 4
COorder 5
COtop50 6

evgmrf 7
evgmrf.control 11
family.evgmrf 14
fitted.evgmrf 16
index2W 17
logLik.evgmrf 17
model.evgmrf 18
plot.evgmrf 20
predict.evgmrf 22
print.evgmrf 25
simulate.evgmrf 26
summary.evgmrf 27
tighten 28
WAprcp 29

Index **31**

`COcnty` *Colorado county annual maxima of daily precipitation*

Description

Annual maxima of daily precipitation for the 64 counties of Colorado over 56 years, together with the county boundaries and the adjacency structure of the counties, for use with `evgmrf`.

Usage

```
data(COcnty)
```

Format

A list with three components:

prcp Numeric matrix with 56 rows and 64 columns. Row `i` is year `1969 + i` (1950 to 2005) and the columns, which are named, are the counties. Units are those of the NA-CORDEX `prec` variable (kg m-2 s-1 in the raw files; multiply by 86400 for mm/day).

polygons Named list of length 64, one element per county, in the same order as the columns of `prcp`. Each element is a list with components `x` (longitude) and `y` (latitude) giving the coordinates of the county boundary in degrees.

adjacency Sparse 64 x 64 symmetric matrix (a **Matrix** `"CsparseMatrix"`) with rows and columns in the same order as `polygons` and the columns of `prcp`. Element (i, j) is 1 if counties i and j are adjacent and 0 otherwise, and the diagonal is zero. It can be passed as `W` to `evgmrf`.

Details

The precipitation data are simulated, and taken from the North American CORDEX (NA-CORDEX) archive. They come from the historical experiment of the WRF regional climate model driven by the GFDL-ESM2M global climate model, on the NAM-22i grid (approximately 0.25 degree resolution). The raw (not bias-corrected) daily precipitation output was used, assuming a 365-day calendar. For each year (1950-2005) the maximum of the daily values was found at every grid cell whose centre lies in Colorado. Each such grid cell was then assigned to a county using `maps::map.where`, and the value for a county is the largest annual maximum over the grid cells assigned to it. County boundaries are from the **maps** package. Two counties are adjacent if their boundaries share at least one point (queen contiguity, as found by `spdep::poly2nb`).

Source

NA-CORDEX archive, file `prec.hist.GFDL-ESM2M.WRF.day.NAM-22i.raw.nc`. https://na-cordex.org. County boundaries from `maps::map("county", "colorado")`.

References

Mearns, L. O., et al. (2017). The NA-CORDEX dataset, version 1.0. NCAR Climate Data Gateway. doi:10.5065/D6SJ1JCH

Examples

```
library(evgmrf)
data(COcnty)

# Map of the mean annual maximum by county
COcnty_prcp_mean <- colMeans(COcnty$prcp)
coords <- do.call(rbind, lapply(COcnty$polygons,
                                function(x) rbind(cbind(x$x, x$y), NA)))
plot(coords, type = "n")
cols <- rev(grey(ppoints(COcnty_prcp_mean)))[rank(COcnty_prcp_mean)]
polygon(coords, col = cols)


# GEV model with a GMRF over the counties, defined by the adjacency matrix
m_poly <- evgmrf(COcnty$prcp, family = "gev", W = COcnty$adjacency)
plot(m_poly, polygons = COcnty$polygons)
```

`coef.evgmrf` *Extract Model Coefficients*

Description

Extract Model Coefficients

Usage

```
## S3 method for class 'evgmrf'
coef(object, ...)
```

Arguments

object	a fitted evgmrf object
...	not used

Value

Fitted values extracted from the object 'object'.

Examples

```
data(COorder)
COmxprcp <- COorder$prcp[, 1, , ]
m_gev <- evgmrf(COmxprcp, family = 'gev')
coef(m_gev)
```

COexc *Colorado daily precipitation above 20mm*

Description

A list of grid longitude (x) and latitude (y) values and an array (prcp) of daily precipitation amounts above 20mm (threshold) for 56 years (dimension 1), for a 28 by 16 grid.

Usage

```
data(COexc)
```

Format

A list of length three

Examples

```
library(evgmrf)
data(COexc)

COexc$z <- colMeans(COexc$prcp, na.rm = TRUE)
image(COexc)
```

`COorder` *Colorado daily precipitation annual order statistics*

Description

The five largest daily precipitation values in each year, for each cell of a 28 by 16 longitude/latitude grid covering Colorado, for 56 years (1950-2005), together with the elevation at each grid cell. The order statistics are suitable for fitting r-largest order statistics models with `evgmrf`.

Usage

```
data(COorder)
```

Format

A list with four components:

x Numeric vector of length 28. Longitudes (degrees east) of the grid columns.

y Numeric vector of length 16. Latitudes (degrees north) of the grid rows.

prcp Numeric array with dimensions `c(56, 5, 28, 16)`, indexed by year, rank, longitude and latitude. `prcp[t, k, i, j]` is the kth largest daily precipitation in year `1969 + t` at longitude `x[i]` and latitude `y[j]`, so that `prcp[, 1, , ]` holds the annual maxima. Units are those of the NA-CORDEX `prec` variable (kg m-2 s-1 in the raw files; multiply by 86400 for mm/day).

elevation Numeric matrix with dimensions `c(28, 16)`. `elevation[i, j]` is the elevation in metres above sea level at longitude `x[i]` and latitude `y[j]`.

Details

The precipitation data are simulated, and taken from the North American CORDEX (NA-CORDEX) archive. They come from the historical experiment of the WRF regional climate model driven by the GFDL-ESM2M global climate model, on the NAM-22i grid (approximately 0.25 degree resolution, regular longitude/latitude). The raw (not bias-corrected) daily precipitation output was used, assuming a 365-day calendar. The grid is the rectangle of grid cells spanning the cells whose centres lie in Colorado, found using `maps::map.where`; no cells are set to `NA`.

Elevation was obtained by bilinear interpolation of a 1 km global elevation data set onto the centres of the grid cells, so it is a point value at each cell centre rather than a cell average.

Source

NA-CORDEX archive, file `prec.hist.GFDL-ESM2M.WRF.day.NAM-22i.raw.nc`. https://na-cordex.org

References

Mearns, L. O., et al. (2017). The NA-CORDEX dataset, version 1.0. NCAR Climate Data Gateway. doi:10.5065/D6SJ1JCH

Examples

```
library(evgmrf)
data(COorder)

# Mean annual maximum of daily precipitation
COorder$z <- colMeans(COorder$prcp[, 1, , ])
image(COorder)


# Elevation on the same grid, and its relationship with the mean annual
# maximum
image(COorder$x, COorder$y, COorder$elevation)
plot(as.vector(COorder$elevation), as.vector(COorder$z),
     xlab = "Elevation (m)", ylab = "Mean annual maximum")

# r-largest model using the top five values in each year
m_r5 <- evgmrf(COorder$prcp, family = "rlarge")
plot(m_r5)
```

`COtop50` *Colorado daily precipitation top 50*

Description

The 50 largest daily precipitation values over the 56-year simulation period (1950-2005), stored in decreasing order, for each cell of a 28 by 16 longitude/latitude grid covering Colorado. They are suitable for fitting peaks-over-threshold models based on the largest values with `evgmrf`. The grid is the same as that of `COorder`.

Usage

```
data(COtop50)
```

Format

A list with three components:

x Numeric vector of length 28. Longitudes (degrees east) of the grid columns.

y Numeric vector of length 16. Latitudes (degrees north) of the grid rows.

prcp Numeric array with dimensions `c(50, 28, 16)`, indexed by rank, longitude and latitude. `prcp[k, i, j]` is the kth largest daily precipitation over the whole period at longitude `x[i]` and latitude `y[j]`. Units are those of the NA-CORDEX `prec` variable (kg m-2 s-1 in the raw files; multiply by 86400 for mm/day).

Details

The precipitation data are simulated, and taken from the North American CORDEX (NA-CORDEX) archive. They come from the historical experiment of the WRF regional climate model driven by the GFDL-ESM2M global climate model, on the NAM-22i grid (approximately 0.25 degree resolution, regular longitude/latitude). The raw (not bias-corrected) daily precipitation output was used. The grid is the rectangle of grid cells spanning the cells whose centres lie in Colorado, found using `maps::map.where`; no cells are set to `NA`.

Source

NA-CORDEX archive, file `prec.hist.GFDL-ESM2M.WRF.day.NAM-22i.raw.nc`. https://na-cordex.org

References

Mearns, L. O., et al. (2017). The NA-CORDEX dataset, version 1.0. NCAR Climate Data Gateway. doi:10.5065/D6SJ1JCH

Examples

```
library(evgmrf)
data(COtop50)

# Mean of the top 50 daily values
COtop50$z <- colMeans(COtop50$prcp)
image(COtop50)


# Fit the model to the full array of top-50 values
m_co <- evgmrf(COtop50$prcp, family = "poisgpd",
               args = list(r = 50, nper = 56))
plot(m_co)
```

`evgmrf` *Fitting extreme value distributions with parameters that vary according to Gaussian Markov random fields*

Description

Fits extreme value distributions where one or more parameters vary across space or structures using a Gaussian Markov random field (GMRF) framework.

Usage

```
evgmrf(
  z,
  family = "gev",
  model = "icar",
  formula,
  covariates,
  weights = 1,
  inits.method = "different",
  W,
  order = 1,
  lambda0,
  args = list(),
  control = list(),
  trace = 0,
  outer = "bfgs",
  gamma = 1,
  nx,
  ny,
  index,
  auto.weights = FALSE,
  bymfns = NULL,
  hyper = list(kappa = -1),
  hyper_start,
  hyper.jitter = missing(hyper_start),
  supernodal = FALSE
)
```

Arguments

`z`	An array or matrix containing the response data. If `W` is supplied, a matrix is expected. If an r-largest family is chosen, an array must be provided.
`family`	A character string specifying the extreme value error distribution. See `family.evgmrf` for details on supported families and required data formats. Defaults to `"gev"`.
`model`	A character string specifying the GMRF structural framework. See `model.evgmrf`. Defaults to `"icar"`.
`formula`	A `formula` or `list` of `formulas` for any fixed effects. Defaults to `~ -1`, which means intercept- and fixed-effect-free forms.
`covariates`	A `list` of covariates with names matching those in `formula` and dimensions compatible with `z`.
`weights`	An array or matrix or scalar of weights for each value in `z`. Defaults to `1`.
`inits.method`	A character string specifying how initial parameter values should be chosen: `"different"` (Default) uses different values for each points based on point-wise optima; `"same"` uses the same values.
`W`	An optional square adjacency matrix defining the spatial neighbor relationships.

order	A scalar or vector specifying autoregressive order. See model.evgmrf. Defaults to 1.
lambda0	A scalar or vector of initial smoothing parameter values. Defaults to 1. Larger values give smoother starting points.
args	A list of additional arguments required by some families. See family.evgmrf.
control	A list of control parameters. See evgmrf.control.
trace	An integer specifying the verbosity level of the optimization output. Larger values provide more comprehensive iterations. Defaults to 0 (silent).
outer	A character string specifying the smoothing parameter optimizer. One of "newton", "nelder-mead", or "bfgs" (Default).
gamma	A scalar multiplier applied directly to the log-likelihood function representing localized constant weights; defaults to 1.
nx, ny	Positive integers giving the numbers of grid points in the first (nx) and second (ny) dimensions of the spatial grid. See Details.
index	A two-column matrix of integers identifying row and column positions; see Details.
auto.weights	Logical; if TRUE, weights are automatically calculated using an effective sample size estimate. EXPERIMENTAL. Defaults to FALSE.
bymfns	A list of functions if model is "bym3" or "bym4". See model.evgmrf.
hyper, hyper_start	Fixed values and starting values, respectively, for the hyperparameters of the GMRF. Each is either a named list or a list of such lists, one for each parameter of the extreme value distribution (in the order given in family.evgmrf), with NULL for any parameter with nothing to supply. A hyperparameter that is omitted from hyper, or set to -1 (the default for kappa), is estimated; any other value fixes it. hyper_start sets the starting values of those that are estimated, on their natural scales. See Details.
hyper.jitter	Logical; if TRUE, start hyperparameter values are jittered slightly to identify improvement. Defaults to TRUE unless hyper_start is used.
supernodal	Logical; activates CHOLMOD supernodal sparse matrix factorization settings. Defaults to FALSE.

Details

In general the last two dimensions of z will define the grid size in terms of rows and columns; otherwise it is compacted. If z is compacted index or nx and ny can be used to supply grid details, and z will be a matrix or a 3-dimensional array if family == "rlarge".

For a compacted z, each location (a list element, or a column or slice of z) is placed on an nx by ny grid as follows.

- If index is supplied, its kth row gives the row and column position of the kth location. Grid cells not listed in index are treated as missing ("holes"). Here nx and ny default to max(index[, 1]) and max(index[, 2]); an error is raised if a supplied value is smaller than these.

- If index is not supplied, the grid is assumed to be full and locations are ordered with the first dimension varying fastest, i.e. as in expand.grid(seq_len(nx), seq_len(ny)). Then nx * ny must equal the number of locations. If only one of nx and ny is given, the other is set to the number of locations divided by it, and an error is raised if this is not a whole number.
- If W is supplied and neither nx nor ny is, the locations are treated as a single line, with nx equal to the number of locations and ny = 1.

Hyperparameters can be fixed, or given starting values, for some or all of the parameters of the extreme value distribution using hyper and hyper_start. The names available depend on model (see model.evgmrf):

- kappa: positive precision multiplier, used by all models and estimated on the log scale.
- rho: value in $(0, 1)$ used by "car" and "bym2", estimated on the probit scale.
- epsilon: positive precision of the unstructured component of "bym", estimated on the log scale.
- nu: vector of values in $(0, 1)$ weighting the second- and third-order components when order includes 2 and/or 3 ("icar" only), with one element per order above 1 and estimated on the probit scale.

For "bym3" the hyperparameters are the arguments of the relevant function in bymfns, excluding the first. Each must be given a fixed value in hyper or a starting value in hyper_start.

A list of length one for hyper, such as the default list(kappa = -1), is used for every parameter; otherwise hyper must be a list of lists. A single named list of any length for hyper_start is used for every parameter. If the first parameter has no starting values to supply, use list() rather than NULL for it in hyper_start, otherwise the whole argument is treated as a single list. If hyper_start is missing, starting values are chosen automatically.

Value

An object of class evgmrf containing localized parameter arrays and structural optimization summaries.

References

Youngman, B. D. (2022). evgam: An R Package for Generalized Additive Extreme Value Models. Journal of Statistical Software. doi:10.18637/jss.v103.i03

See Also

predict.evgmrf, COorder

Examples

```
data(COorder)
# select top order statistic per year, and 20 x 15 subgrid
COmxprcp <- COorder$prcp[1:20, 1, 1:16, 1:14]
m_gev <- evgmrf(COmxprcp, family = "gev")
```

```
# Fix the precision parameter kappa at 5 for every parameter
m_fixed <- evgmrf(COmxprcp, family = "gev", hyper = list(kappa = 5))

# Estimate kappa and rho for a BYM2 model, starting rho at 0.5
m_bym2 <- evgmrf(COmxprcp, family = "gev", model = "bym2",
                 hyper_start = list(rho = 0.5))
```

`evgmrf.control` *Control parameters for evgmrf model optimization*

Description

Define convergence thresholds, finite difference structures, and regularisation adjustments for the internal numerical solvers used by `evgmrf`.

Usage

```
evgmrf.control(
  eps = 0.005,
  it0 = 20,
  step_size = 0.2,
  reml_eps = 0.005,
  reml_direction = "ad-hoc",
  reml_steptol = 1e-04,
  reml_stepmax = 2.5,
  reml_itlim = 100,
  reml_gradtol = 0.01,
  reml_dgradtol = 1e-04,
  reml_fntol = 1e-06,
  reml_alpha0 = 1,
  reml_rho0 = 0.1,
  line_search_mult = 1,
  inner_optim = "chol",
  phi.tol = 1e-05,
  grad_mult = 0,
  par_mult = 0.1,
  update = FALSE,
  openmp = FALSE,
  threads = 0,
  perturb.tol = 0.01,
  perturb.mult = 5,
  refine.inits = FALSE
)
```

Arguments

eps	A numeric value specifying the convergence tolerance for the inner optimization loop. Defaults to 5e-3.
it0	An integer setting the maximum number of initial iterations allowed for the primary optimization routine. Defaults to 20.
step_size	A numeric value defining the step size scaling factor applied during optimization line searches. Defaults to 0.2.
reml_eps	A numeric value specifying the convergence tolerance for the Restricted Maximum Likelihood (REML) outer loop. Defaults to 5e-3.
reml_direction	A character string or vector detailing the numerical differentiation scheme used to evaluate REML parameter gradients. Supported options are "forward", "backward", "central", or "ad-hoc". If "ad-hoc" is selected, a backward difference is chosen for parameters greater than 1, and a forward difference is applied otherwise. Defaults to "ad-hoc".
reml_steptol	A numeric value defining the minimum step length tolerance for REML updates. Defaults to 1e-4.
reml_stepmax	A numeric value specifying the maximum permissible step length taken during a single REML update iteration. Defaults to 2.5.
reml_itlim	An integer setting the hard iteration limit for the REML optimization process. Defaults to 100.
reml_gradtol	A scalar setting the gradient tolerance for the REML optimization process. Defaults to 1e-2.
reml_dgradtol	A scalar setting the gradient tolerance difference over four iterations for the REML optimization process. Defaults to 1e-4.
reml_fntol	A scalar setting the function tolerance for the REML optimization process. Defaults to 1e-6.
reml_alpha0	A scalar setting the initial line search multiplier for the REML optimization process. Defaults to 1.
reml_rho0	A scalar setting the multiplicative rate at which the line search multiplier is reduced for the REML optimization process. Defaults to 0.5.
line_search_mult	A multiplier for the step when line searches take place. Defaults to 1.
inner_optim	A character string selecting the linear algebraic solver for inner loop parameters; defaults to "chol" (which calls a wrapper to Eigen::SimplicialLLT). If explicitly set to "Cholesky", super is automatically forced to TRUE and Matrix::Cholesky is used with super = TRUE. Only recommended with Matrix(>=1.8-0).
phi.tol	A numeric convergence tolerance threshold for the model's spatial phi parameters. Defaults to 1e-5.
grad_mult	A numeric penalty coefficient applied during outer REML optimization. If any unpenalised inner optimization gradient coordinate exceeds an absolute threshold of 1, this value is added directly as a discrete penalty to the objective function score. Defaults to 0.

par_mult	A numeric scaling factor controlling an L2 ridge regularisation penalty that shrinks outer REML parameters toward their target baseline vectors. Defaults to 0.1.
update	Logical; indicates whether spatial weight structures should be iteratively updated during convergence passes. Defaults to FALSE.
openmp	Logical; toggles whether matrix solving routines exploit multi-core CPU parallelism via OpenMP extensions. Defaults to FALSE.
threads	An integer specifying the target system CPU core thread allocation limit used if openmp = TRUE. A value of 0 defaults to system automatic selection.
perturb.tol	A numeric baseline ridge penalty added directly to the diagonal of near-singular Hessian matrices to guarantee positive-definiteness during standard Cholesky factorization loops. Defaults to 1e-2.
perturb.mult	A scaling multiplier used to exponentially step up the diagonal perturb.tol ridge penalty during iterative Cholesky factorization failures. Defaults to 5.
refine.inits	Logical; if TRUE, initial values for parameters are refined using a diagonal quasi-Newton method. Defaults to FALSE.

Details

Defining explicit arguments simplifies code verification and surfaces automatic variable completions during interactive package sessions.

Value

A structured, named list encompassing all validated logical indicators, character frameworks, and numeric convergence boundaries.

See Also

evgmrf

Examples

```
# Generate default tuning structures
config <- evgmrf.control()

# Configure custom parallel threads and adjust the parameter shrinkage weight
custom_config <- evgmrf.control(openmp = TRUE, threads = 4, par_mult = 0.5)
```

`family.evgmrf` *Supported Families for evgmrf Models*

Description

Documentation for the extreme value distributions (families) and data structure expectations supported by the `evgmrf` fitting function.

Details

Argument `family` of `evgmrf` is a character string giving the distribution to be fitted, and defaults to `"gev"`. The following families are available:

- `"gev"`: Generalized extreme value distribution for block maxima. Years are typically the blocks, so the data are annual maxima. Its parameters are the location, scale and shape.
- `"rlarge"`: r-largest order statistics model, an extension of the GEV in which the r largest values in each block are modelled, so that $r = 1$ is the GEV. It has the same parameters as the GEV, and the same quantiles. The value of r is taken from the data; see below.
- `"gpd"`: Generalized Pareto distribution for excesses of a high threshold. The threshold is not supplied to `evgmrf`: the data must already be excesses, i.e. with the threshold subtracted. Its parameters are the scale and shape.
- `"poisgpd"`: Poisson-GPD point process model, an extension of the GPD that also uses the threshold, and is parametrized by the GEV's location, scale and shape. The data are the values themselves, not excesses. Requires `args$nper` and either `args$u` or `args$r`.
- `"ald"`: Asymmetric Laplace distribution, used for quantile regression, and hence for estimating thresholds for the GPD and Poisson-GPD. Its parameters are the location, which is the quantile being estimated, and the scale. Requires `args$tau`.

Where the shape parameter of a GEV, r-largest or GPD model is estimated to be within 10^{-6} of zero, it is set to 10^{-6} to avoid numerical instability. For the GEV, r-largest and Poisson-GPD families, the scale is modelled on the log scale and the shape on a transformed scale, which `plot` and `predict` label `"logscale"` and `"transshape"`. Setting `type = "response"` gives `"scale"` and `"shape"`.

Data `z` can be supplied as an array with dimensions indexing time (or block), then grid row, then grid column, e.g. year, longitude and latitude. For `"rlarge"` it is a four-dimensional array with dimensions indexing block, order statistic (largest first), grid row and grid column, with r the length of the second dimension. Alternatively, in a compacted form, `z` can be a matrix with one column for each location (a three-dimensional array of block, order statistic and location for `"rlarge"`), or a list with one element for each location. If a matrix or list is supplied, `nx` and `ny`, or `index`, describe the grid, and `W` can be supplied instead. Lists allow numbers of values to vary by location, as they do for excesses. Missing values are ignored, and locations where all values are missing are part of the GMRF but do not contribute to the likelihood.

Family-specific arguments are supplied in `args`:

- `args$tau`: For `"ald"`, the quantile to be estimated. For example, `args = list(tau = .5)` gives a median estimate.

- `args$nper`: For `"poisgpd"`, the length of the period covered by the data, in units of time such as years. Estimates are then on that scale, e.g. annual, so are comparable with those from annual maxima.
- `args$u`: For `"poisgpd"`, the threshold above which values are used, either a single value or one for each location.
- `args$r`: For `"poisgpd"`, used instead of `args$u` to give a threshold for each location, which is its rth largest value. Data that have already been restricted to the largest r values at each location are then equivalent to `args$u` being their minimum at each location.

An estimated quantile from the `"ald"` family can be used as a threshold, by supplying `predict(m_ald)$location`, where `m_ald` is the fitted model, as `args$u` to `"poisgpd"`.

References

Coles, S. G. (2001). *An Introduction to Statistical Modeling of Extreme Values*. Springer-Verlag. doi:10.1007/9781447136750

Oh, H.-S., Lee, T. C. M. and Nychka, D. W. (2011). Fast nonparametric quantile regression with arbitrary smoothing methods. *Journal of Computational and Graphical Statistics*, 20(2), 510-526. doi:10.1198/jcgs.2010.10063

Yu, K. and Moyeed, R. A. (2001). Bayesian quantile regression. *Statistics & Probability Letters*, 54(4), 437-447. doi:10.1016/s01677152(01)001249

See Also

`evgmrf`, `model.evgmrf`

Examples

```
data(COorder)

# GEV distribution for annual maxima
COmxprcp <- COorder$prcp[, 1, , ]
m_gev <- evgmrf(COmxprcp, family = "gev")

# r-largest order statistics model, using the top five values in each year
m_rlarge <- evgmrf(COorder$prcp, family = "rlarge")

# GPD for excesses of 20mm. The excesses are supplied as a list, without
# missing values, with nx and ny giving the grid
data(COexc)
excess <- COexc$prcp - 20
nx <- length(COexc$x)
ny <- length(COexc$y)
ind <- expand.grid(seq_len(nx), seq_len(ny))
excess_list <- lapply(1:nrow(ind),
                      function(i) na.omit(excess[, ind[i, 1], ind[i, 2]]))
m_gpd <- evgmrf(excess_list, family = "gpd", nx = nx, ny = ny)

# Poisson-GPD with a threshold of 20mm, for 56 years of data
```

```
m_pp1 <- evgmrf(COexc$prcp, family = "poisgpd",
                args = list(u = 20, nper = 56))

# Poisson-GPD to the top 50 values at each grid cell, which are used to
# give cell-specific thresholds
data(COtop50)
m_pp2 <- evgmrf(COtop50$prcp, family = "poisgpd",
                args = list(r = 50, nper = 56))

# Estimate the median of the top 50 values, roughly the 99.88th percentile
# of daily precipitation, and use it as the threshold for a Poisson-GPD
m_ald <- evgmrf(COtop50$prcp, family = "ald", args = list(tau = .5))
m_pp3 <- evgmrf(COtop50$prcp, family = "poisgpd",
                args = list(nper = 56, u = predict(m_ald)$location))
```

`fitted.evgmrf` *Extract Model Fitted Values*

Description

Extract Model Fitted Values

Usage

```
## S3 method for class 'evgmrf'
fitted(object, ...)
```

Arguments

`object`	a fitted `evgmrf` object
`...`	not used

Value

Fitted values extracted from the object 'object'.

Examples

```
data(COorder)
COmxprcp <- COorder$prcp[, 1, , ]
m_gev <- evgmrf(COmxprcp, family = 'gev')
fitted(m_gev)
```

index2W *Build a sparse adjacency matrix from grid indices*

Description

Constructs a sparse binary adjacency matrix `W` where `W[i, j] == 1` if points `i` and `j` are rook-adjacent (differ by exactly 1 in one coordinate and 0 in the other), and 0 otherwise.

Usage

```
index2W(ind)
```

Arguments

`ind`	A numeric matrix with two columns giving the row/column (or x/y) coordinates of each point. Each row is one point.

Value

A sparse matrix (`Matrix::dgCMatrix`) of dimension `nrow(ind)` by `nrow(ind)`, symmetric, with 1s marking adjacent pairs and 0s elsewhere.

Examples

```
ind <- as.matrix(expand.grid(1:5, 1:5))
W <- index2W(ind)
```

logLik.evgmrf *Log-likelihood from a fitted* `evgmrf` *object*

Description

Log-likelihood from a fitted `evgmrf` object

Usage

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

Arguments

`object`	a fitted `evgmrf` object
`...`	not used

Value

A scalar

Examples

```
data(COorder)
COmxprcp <- COorder$prcp[, 1, , ]
m_gev <- evgmrf(COmxprcp, family = 'gev')
logLik(m_gev)
```

`model.evgmrf` *Supported GMRF Models for evgmrf Models*

Description

Documentation for the GMRF structures supported by the `evgmrf` fitting function.

Details

Argument `model` of `evgmrf` is a character vector giving the GMRF model for each parameter of the extreme value distribution. It is repeated if of length one, and the default is `"icar"` for every parameter. Use `NA` for a parameter that is not to have a GMRF, i.e. that is constant. In the following, Ω is the precision matrix of a parameter's GMRF and Q is the structure matrix determined by the grid and `order`, or by the neighborhood matrix `W`. The following models are available:

- `"icar"`: Intrinsic conditional autoregressive model, with $\Omega = \kappa Q$ and $\kappa > 0$. Its hyper-parameter is $\log \kappa$.
- `"car"`: Proper conditional autoregressive model, with $\Omega = \kappa[\rho Q + (1 - \rho)I]$ and $0 \leq \rho \leq 1$, where I is the identity matrix. The case $\rho \to 1$ corresponds to the ICAR model and $\rho \to 0$ to independent Gaussian noise. Its hyper-parameters are $(\log \kappa, \Phi^{-1}(\rho))$, where Φ is the standard Gaussian distribution function.
- `"bym2"`: Besag-York-Mollie 2 model, as reparametrized by Riebler et al. (2016). The random effect is the sum of a spatial component with precision $\kappa Q/\rho$ and an independent Gaussian component with precision $\kappa I/(1 - \rho)$, where $\rho \in (0, 1)$ is the proportion of variation that is spatial. Its hyper-parameters are $(\log \kappa, \Phi^{-1}(\rho))$. Setting `decompose = TRUE` in `plot` and `predict` separates the spatial and independent parts.
- `"bym3"`: As `"bym2"`, but with an independent component that has a (not necessarily Gaussian) distribution that is specified by the user, and which may have parameters that need estimating. See below.

By default, no fixed intercept is included for `"icar"` models (`formula = ~ -1`), whereas one is included for the other models (`formula = ~ 1`).

For `"bym3"`, `bymfns` must be supplied to `evgmrf`. It is a `list` with one element for each distribution parameter, with `NULL` for those without a `"bym3"` model. Otherwise the element is a function that

returns the negative log-density of the independent component. Its first argument is the value at which the density is evaluated, and any further arguments are parameters of the distribution. Each such parameter must be given either a starting value, which is then estimated, in `hyper_start`, or a fixed value in `hyper`. For example, `hyper_start = list(list(par1 = -2), NULL, NULL)`. A function with a single argument specifies the distribution completely, so that nothing is estimated. See examples below.

Argument `order` can be supplied to `evgmrf` to specify the order of the autoregressive form of each parameter. Orders above one are only available for `"icar"` models on a grid; if `W` is supplied it defines Q and `order` is not used. On a grid, the form of order q, for $q = 1, 2, 3$, penalizes qth-order differences in each grid direction. The default of 1 is a first-order form. If `order` is a vector of integers, it is coerced to a `list` with `lapply(order, seq_len)`, after being repeated for each parameter if of length one. If a `list`, each element gives the order or orders for the corresponding distribution parameter. For example, for the GEV distribution `order = list(1, 1:2, 1:3)` gives a first-order form for the location parameter, an additive combination of first- and second-order forms for the scale parameter, and an additive combination of first-, second- and third-order forms for the shape parameter. In contrast, `order = list(1, 1, 2)` uses only a second-order form for the shape parameter. In an additive combination the weight of the lowest order is one, to ensure identifiability with κ. Relative variances can be fixed with `alpha`. If `order = 1:2` and `alpha = .5`, and lambda is the smoothing parameter, then the first-order term has smoothing parameter lambda, the second-order term has alpha lambda, the third-order term has alpha^2 lambda, and so on.

References

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B*, 36(2), 192-236. doi:10.1111/j.25176161.1974.tb00999.x

Besag, J., York, J. and Mollie, A. (1991). Bayesian image restoration, with two applications in spatial statistics. *Annals of the Institute of Statistical Mathematics*, 43(1), 1-20. doi:10.1007/BF00116466

Riebler, A., Sorbye, S. H., Simpson, D. and Rue, H. (2016). An intuitive Bayesian spatial model for disease mapping that accounts for scaling. *Statistical Methods in Medical Research*, 25(4), 1145-1165. doi:10.1177/0962280216660421

See Also

`evgmrf`

Examples

```
data(COorder)
COmxprcp <- COorder$prcp[, 1, , ]

# ICAR models for all three GEV parameters (the default)
m_gev <- evgmrf(COmxprcp, family = "gev")

# Proper CAR model for the GEV location parameter
m_gev_car <- evgmrf(COmxprcp, family = "gev", model = c("car", "icar", "icar"))
plot(m_gev_car, which = 1)
```

```
# As above, but with constant shape parameter, i.e. no GMRF
m_gev_car2 <- evgmrf(COmxprcp, family = "gev", model = c("car", "icar", NA))

# BYM2 model for the GEV location parameter, with its estimate split into
# spatial and independent parts
m_gev_bym <- evgmrf(COmxprcp, family = "gev", model = c("bym2", "icar", "icar"))
plot(m_gev_bym, which = "location", decompose = TRUE)

# Custom (here Gaussian) independent component for the location parameter,
# with a standard deviation to be estimated, starting from exp(-2)
custom_fn1 <- function(x, par1) - dnorm(x, 0, exp(par1), log = TRUE)
m_gev_bym3 <- evgmrf(COmxprcp, family = "gev", model = c("bym3", "icar", "icar"),
                     bymfns = list(custom_fn1, NULL, NULL),
                     hyper_start = list(list(par1 = -2), NULL, NULL))

# As above, but with a fixed variance, so no parameters are estimated
custom_fn2 <- function(x) - dnorm(x, log = TRUE)
m_gev_bym4 <- evgmrf(COmxprcp, family = "gev", model = c("bym3", "icar", "icar"),
                     bymfns = list(custom_fn2, NULL, NULL))

# Equivalent to the above, fixing par1 = 0 with hyper
m_gev_bym5 <- evgmrf(COmxprcp, family = "gev", model = c("bym3", "icar", "icar"),
                     bymfns = list(custom_fn1, NULL, NULL),
                     hyper = list(list(par1 = 0), NULL, NULL))

# Higher-order forms: first order for location, first and second combined
# for scale, first, second and third combined for shape
m_gev_ord1 <- evgmrf(COmxprcp, family = "gev", order = list(1, 1:2, 1:3))

# Only a second-order form for the shape parameter
m_gev_ord2 <- evgmrf(COmxprcp, family = "gev", order = list(1, 1, 2))
```

`plot.evgmrf` *Plot Method for Fitted* `evgmrf` *Objects*

Description

Produces spatial or line plots of estimated parameters, linear predictors, or predicted quantiles from a fitted extreme value Markov random field (`evgmrf`) model.

Usage

```
## S3 method for class 'evgmrf'
plot(
  x,
  which,
  polygons,
  lims,
```

```
  nlev = 15L,
  edge.drop,
  mfrow,
  mfcol,
  plot.args = list(asp = 1),
  show = TRUE,
  ...
)
```

Arguments

x	An object of class evgmrf, typically representing a fitted spatial extreme value model.
which	A numeric vector of indices or character vector of names specifying which parameter estimates, linear predictors, or quantiles to plot. Defaults to plotting all components returned by predict.evgmrf.
polygons	An optional polygon dataset (such as spatial polygons or boundary geometries) used to map predicted values spatially. If omitted, predictions are displayed as grid maps via levelplot or 1D lines via xyplot.
lims	An optional list specifying the plot limits (e.g., color bar ranges) for each plot component. Defaults to the range of values in each plotted component.
nlev	An integer or vector of integers giving the number of color levels/intervals used in image or polygon plots. Defaults to 15L.
edge.drop	An optional integer or logical specification passed internally to drop outer edge pixels/nodes from spatial grid outputs prior to plotting.
mfrow	A two-element integer vector c(rows, cols) specifying the grid layout dimensions for arranging multiple panel plots by row.
mfcol	A two-element integer vector c(rows, cols) specifying the grid layout dimensions for arranging multiple panel plots by column.
plot.args	A list of additional arguments passed to the underlying plotting functions (e.g., levelplot or xyplot). Defaults to list(asp = 1).
show	Logical; if TRUE plots are shown in graphics device and if FALSE a list is returned. Defaults to TRUE.
...	Arguments passed on to predict.evgmrf (e.g., type, prob, or decompose).

Details

This function computes model predictions by calling predict.evgmrf with any extra arguments supplied via ..., then renders visual displays of the output:

- If polygons is supplied, values are rendered onto spatial polygon regions.
- If polygons is missing and a prediction component is 2D, a heat map grid is plotted using levelplot.
- If a prediction component is 1D (e.g., single row or column), a line plot is rendered using xyplot.

Multiple plot panels are automatically arranged in a grid using grid.arrange.

Value

Invisibly returns the arranged grid of plot objects (a grob layout from gridExtra).

References

Youngman, B. D. (2022). evgam: An R Package for Generalized Additive Extreme Value Models. Journal of Statistical Software. doi:10.18637/jss.v103.i03

See Also

evgmrf, predict.evgmrf, levelplot

Examples

```
data(COorder)
COmxprcp <- COorder$prcp[1:20, 1, 1:16, 1:14]
m_gev <- evgmrf(COmxprcp)
plot(m_gev)
plot(m_gev, type = 'response')
plot(m_gev, prob = .99)
```

---

predict.evgmrf *Predictions and standard errors from a fitted* evgmrf *object*

Description

Obtains predicted values, extreme value quantiles, and associated standard errors from a fitted Gaussian Markov random field spatial extremes model.

Usage

```
## S3 method for class 'evgmrf'
predict(
  object,
  type = "link",
  se.fit = FALSE,
  prob = NULL,
  index = NULL,
  nx = NULL,
  ny = NULL,
  set2NA = FALSE,
  simplify2array = FALSE,
  xid = NULL,
  yid = NULL,
  loop = TRUE,
  progress = FALSE,
  chunksize = 100,
```

```
  se.method = "direct",
  nsim = 1000,
  decompose = FALSE,
  random2zero = c(FALSE, FALSE),
  drop.parametric = TRUE,
  openmp = object$control$openmp,
  threads = object$control$threads,
  supernodal = FALSE,
  ...
)
```

Arguments

object	A fitted evgmrf object.
type	A character string specifying the prediction scale. Supported options are "link" (the linear predictor scale) or "response" (the inverse-link parameter scale). If prob is supplied, this is automatically overridden to evaluate on the quantile scale. Defaults to "link".
se.fit	Logical; if TRUE, calculates associated standard errors for the predictions alongside point estimates. Defaults to FALSE.
prob	A scalar or vector of probabilities mapping to the target extreme value quantiles to be estimated. Defaults to NULL.
index	A matrix containing spatial row and column coordinate index maps used to reconstruct missing grid layouts. Defaults to object$index.
nx, ny	Positive integers giving the numbers of grid points in the first (nx) and second (ny) dimensions of the spatial grid. Defaults to object$nx and object$nx, respectively.
set2NA	Set parameters to NA where there are no data. Defaults to FALSE.
simplify2array	Logical; if TRUE, coerces and binds the underlying prediction surfaces into a single multidimensional array matrix. Defaults to FALSE.
xid, yid	Integer index vectors identifying localized grid subsets to extract. Defaults to the entire structural dimensions tracked in object$xid and object$yid.
loop	Logical; if TRUE, forces internal sparse solvers to iterate via memory-conserving blocks rather than loading full dense matrices. Defaults to TRUE.
progress	Logical; if TRUE, renders active text progress bars to the R console tracking standard error computation loops. Defaults to the value of loop.
chunksize	An integer determining the slice sizing processed concurrently per standard error iterative loop. Defaults to 100.
se.method	A character string selecting the error propagation method. Supported options are "direct" (analytical delta method solving) or "simulation" (stochastic Monte Carlo sampling). Defaults to "direct".
nsim	An integer tracking random sampling pathways drawn when evaluating stochastic errors under se.method = "simulation". Defaults to 1000.

decompose	Logical; if TRUE, random additive terms inside Besag-York-Mollié (BYM) formulations are broken down into individual spatial and random elements. Defaults to FALSE.
random2zero	Logical vector; first component corresponds to fitted values and second to standard errors. When TRUE, random additive terms or corresponding standard errors inside Besag-York-Mollié (BYM) formulations are set to zero. Defaults to c(FALSE, FALSE).
drop.parametric	Logical; if TRUE and decompose = TRUE, deletes fixed parametric background terms from the final evaluated list. Defaults to FALSE.
openmp	Logical; switches whether analytical solvers exploit shared memory multi-core CPU parallelism extensions. Defaults to object$control$openmp.
threads	An integer controlling maximum system CPU threads allocated if openmp = TRUE. Defaults to object$control$threads.
supernodal	Logical; activates CHOLMOD supernodal sparse matrix factorization settings. Defaults to FALSE.
...	Unused auxiliary flags passed along for generic matching alignment with predict.

Details

Analytical standard errors are solved using Cholesky factorizations of the preconditioned Hessian matrix evaluated at the maximum likelihood convergence point. If Besag-York-Mollié (BYM2) configurations are detected, structural projections adjust the scaling matrix properties automatically. When calculating quantile-scale standard errors under se.method = "direct", the function calls analytical gradients attached as a functional derivative attribute to object$quantile0.

See evgmrf for more details on nx, ny and index.

Value

If se.fit = FALSE, the function returns a list or array of point predictions. If se.fit = TRUE, it returns a nested list containing:

- fitted: The point predictions or quantiles mapped to the grid layout.
- se: The associated standard errors matched element-for-element to the shapes inside fitted.

References

Youngman, B. D. (2022). evgam: An R Package for Generalized Additive Extreme Value Models. Journal of Statistical Software. doi:10.18637/jss.v103.i03

See Also

evgmrf, simulate.evgmrf, family.evgmrf

Examples

```
data(COorder)
COmxprcp <- COorder$prcp[, 1, , ]
m_gev <- evgmrf(COmxprcp)

# Evaluate location and scale predictions
link_preds <- predict(m_gev, type = "link")

# Calculate point estimates and standard errors for the 95th percentile
q_preds <- predict(m_gev, prob = 0.95, se.fit = TRUE, se.method = "direct")
```

---

print.evgmrf *Print a fitted* evgmrf *object*

Description

Print a fitted evgmrf object

Usage

```
## S3 method for class 'evgmrf'
print(x, ...)
```

Arguments

x	a fitted evgmrf object
...	not used

Value

The call of the evgmrf object

Examples

```
data(COorder)
COmxprcp <- COorder$prcp[, 1, , ]
m_gev <- evgmrf(COmxprcp, family = 'gev')
print(m_gev)
```

`simulate.evgmrf` *Simulations from a fitted* `evgmrf` *object*

Description

Simulates parameters or quantiles from the posterior or predictive distribution of a fitted `evgmrf` model.

Usage

```
## S3 method for class 'evgmrf'
simulate(
  object,
  nsim = 1,
  seed = NULL,
  type = "link",
  prob = NULL,
  simplify2array = TRUE,
  decompose = FALSE,
  supernodal = FALSE,
  ...
)
```

Arguments

`object`	A fitted `evgmrf` object.
`nsim`	An integer specifying the number of simulation paths to generate. Defaults to `1`.
`seed`	An integer giving the seed for simulations, passed directly to `set.seed`. If `NULL`, the current random number generator state is preserved.
`type`	A character string giving the type of simulation sought. Supported options are `"link"` (the linear predictor scale) or `"response"` (the inverse-link parameter scale). If `prob` is supplied, this parameter is overridden to compute values on the quantile scale. Defaults to `"link"`.
`prob`	A scalar or vector of probabilities mapping to quantiles to be estimated. If supplied, the simulations will evaluate predictions on the quantile scale. Defaults to `NULL`.
`simplify2array`	Logical; if `TRUE` (the default), simulated paths are coerced and dropped into spatial arrays dimensioned to match the grid layout (`nx` by `ny` by `nsim`).
`decompose`	Logical; if `TRUE`, structural additive terms within Besag-York-Mollié (BYM) models are returned broken down into their individual spatial and non-spatial components. Defaults to `FALSE`.
`supernodal`	Logical; activates CHOLMOD supernodal sparse matrix factorization settings. Defaults to `FALSE`.
`...`	Unused arguments. Passed along for generic compatibility with `simulate`.

Details

The simulation routine draws multivariate normal samples using a precision-based Cholesky solver applied to the preconditioned Hessian matrix evaluated at the maximum likelihood estimates. These coefficient variations are then mapped through design matrices and inverse-link functions according to the target extreme value family definitions.

Value

A `list` or `array` of simulated paths structured according to the choice of `simplify2array` and `decompose`.

References

Youngman, B. D. (2022). evgam: An R Package for Generalized Additive Extreme Value Models. Journal of Statistical Software. doi:10.18637/jss.v103.i03

See Also

`evgmrf`, `predict.evgmrf`

Examples

```
data(COorder)
COmxprcp <- COorder$prcp[, 1, , ]
m_gev <- evgmrf(COmxprcp)
sim_paths <- simulate(m_gev, nsim = 100, type = "response")
```

`summary.evgmrf` *Summary method for a fitted* `evgmrf` *object*

Description

Summary method for a fitted `evgmrf` object

Usage

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

## S3 method for class 'summary.evgmrf'
print(x, ...)
```

Arguments

`object`	a fitted `evgmrf` object
`...`	not used
`x`	a `summary.evgmrf` object

Details

The main purpose of summary.evgmrf is to make clear the model that has been fitted. It will give details of fixed effect specifications, which parameters vary according to GMRFs, and the specifications of those GMRFs, where relevant.

Value

A summary.evgmrf object

Examples

```
data(COorder)
COmxprcp <- COorder$prcp[, 1, , ]
m_gev <- evgmrf(COmxprcp, family = 'gev')
summary(m_gev)
```

---

tighten *Reducing array dimensions to drop NA-filled dimensions*

Description

Function tighten takes a three-dimensional array with NAs in, and reduces dimension, if possible.

Usage

```
tighten(x)
```

Arguments

x	an array, or matrix if W supplied, or array if W supplied and r-largest

Details

To do.

Value

An array

Examples

```
# To follow
```

WAprcp *Top 50 daily precipitation values over Washington State*

Description

Simulated daily precipitation extremes over Washington State, taken from the North American CORDEX (NA-CORDEX) archive. For each grid cell, the 50 largest daily precipitation values over the 56-year simulation period (1950-2005) are stored in decreasing order. Grid cells that fall outside Washington State are set to NA.

Usage

```
data(WAprcp)
```

Format

A list with three components:

x Numeric vector of length 31. Longitudes (degrees east) of the grid columns, ranging from -124.625 to -117.125.

y Numeric vector of length 14. Latitudes (degrees north) of the grid rows, ranging from 45.625 to 48.875.

prcp Numeric array with dimensions `c(50, 31, 14)`, indexed by rank, longitude and latitude. `prcp[k, i, j]` is the kth largest daily precipitation at longitude `x[i]` and latitude `y[j]`. Cells outside Washington State are NA. Units are those of the NA-CORDEX `prec` variable (kg m-2 s-1 in the raw files; multiply by 86400 for mm/day).

Details

The data come from the historical experiment of the WRF regional climate model driven by the GFDL-ESM2M global climate model, on the NAM-22i grid (approximately 0.25 degree resolution, regular longitude/latitude). The raw (not bias-corrected) precipitation output was used.

Source

NA-CORDEX archive, file `prec.hist.GFDL-ESM2M.WRF.day.NAM-22i.raw.nc`. https://na-cordex.org

References

Mearns, L. O., et al. (2017). The NA-CORDEX dataset, version 1.0. NCAR Climate Data Gateway. doi:10.5065/D6SJ1JCH

Examples

```
library(evgmrf)
data(WAprcp)


# Fit the model to the full array of top-50 values
m_wa1 <- evgmrf(WAprcp$prcp, family = "poisgpd",
                args = list(r = 50, nper = 56))
plot(m_wa1, set2NA = TRUE)

# Fit the model using only the grid cells with data, with an explicit
# neighbourhood matrix W
some_prcp <- apply(is.finite(WAprcp$prcp), 2:3, any)
some_prcp_index <- which(some_prcp, arr.ind = TRUE)
W <- index2W(some_prcp_index)
WAprcp_list <- apply(some_prcp_index, 1,
                     function(idx) WAprcp$prcp[, idx[1], idx[2]],
                     simplify = FALSE)
m_wa2 <- evgmrf(WAprcp_list, family = "poisgpd", W = W,
                args = list(r = 50, nper = 56))
```

Index

∗ **datasets**
  COcnty, 2
  COexc, 4
  COorder, 5
  COtop50, 6
  WAprcp, 29
∗ **models**
  family.evgmrf, 14
  model.evgmrf, 18

COcnty, 2
coef.evgmrf, 3
COexc, 4
COorder, 5, *6*, *10*
COtop50, 6

evgmrf, *2*, *5, 6*, 7, *11*, *13–15*, *18, 19*, *22*, *24*, *27*
evgmrf.control, *9*, 11

family.evgmrf, *8, 9*, 14, *24*
fitted.evgmrf, 16

grid.arrange, *21*

index2W, 17

levelplot, *21, 22*
logLik.evgmrf, 17

model.evgmrf, *8–10*, *15*, 18

plot.evgmrf, 20
predict, *24*
predict.evgmrf, *10*, *21*, 22, *22*, *27*
print.evgmrf, 25
print.summary.evgmrf *(*summary.evgmrf*)*, 27

set.seed, *26*
simulate, *26*
simulate.evgmrf, *24*, 26

summary.evgmrf, 27

tighten, 28

WAprcp, 29

xyplot, *21*