Journal of Statistical Software

July 2022, Volume 103, Issue 3. doi: 10.18637/jss.v103.i03

evgam: An R Package for Generalized Additive Extreme Value Models

Benjamin D. Youngman
University of Exeter

Abstract

This article introduces the R package **evgam**. The package provides functions for fitting extreme value distributions. These include the generalized extreme value and generalized Pareto distributions. The former can also be fitted through a point process representation. Package **evgam** supports quantile regression via the asymmetric Laplace distribution, which can be useful for estimating high thresholds, sometimes used to discriminate between extreme and non-extreme values. The main addition of package **evgam** is to let extreme value distribution parameters have generalized additive model forms, the smoothness of which can be objectively estimated using Laplace's method. Illustrative examples fitting various distributions with various specifications are given. These include daily precipitation accumulations for part of Colorado, US, used to illustrate spatial models, and daily maximum temperatures for Fort Collins, Colorado, US, used to illustrate temporal models.

Keywords: generalized extreme value distribution, generalized Pareto distribution, point process, generalized additive model, Laplace's method, R.

1. Introduction

Practical extreme value analyses have typically considered modeling block maxima with the generalized extreme value (GEV) distribution or exceedances of a high threshold with the generalized Pareto distribution (GPD); see Davison and Smith (1990) for a seminal work on the latter approach, and Coles (2001) for a detailed overview of both approaches. Here, the GEV and GPD distributions will be considered *the* extreme value distributions (EVD). Smith (1989) develops a model using Pickands' (1971) point process representation of extremes, which, in some sense, marries the two EVDs.

Various packages have been contributed to the Comprehensive R Archive Network (CRAN) to fit EVDs in R (R Core Team 2022). One of the earliest, package **ismev** (Heffernan and Stephenson 2018), allows users to recreate many of the analyses presented in Coles (2001).

Later R packages, such as **evd** (Stephenson 2002), **evir** (Pfaff and McNeil 2018), **extRemes** (Gilleland and Katz 2016) and **mev** (Belzile, Wadsworth, Northrop, Grimshaw, and Huser 2020), have offered various functions for fitting univariate and multivariate EVDs. For a review see Gilleland, Ribatet, and Stephenson (2013), and for an up-to-date list of packages contributed to CRAN see the dedicated task view by Dutang (2022).

This work focuses on regression-based models for extremes, a flexible class of nonstationary model for extremes achieved by letting EVD parameters vary with covariates. Nonstationarity was considered in early models for extremes, in particular Smith (1986) and Smith's (1989) study of trends in ground-level ozone. Packages **ismev** and **evd** offer some scope for linear forms. Such forms, however, can be restrictive if an involved choice of covariate parametrization is required before sufficient flexibility is achieved (if it can be achieved).

More general regression-based EVD parameter forms can offer more robust analyses. Hall and Tajvidi (2000), Davison and Ramesh (2000) and Ramesh and Davison (2002), for example, consider local-likelihood fitting of trends. Pauli and Coles (2001) use a penalized likelihood approach where smoother EVD parameter estimates incur less penalty. Pauli and Coles's (2001) approach builds on results for exponential family models covered in Green and Silverman (1994), but relies on fixed smoothing parameters to control the amount of penalty. Chavez-Demoulin and Davison (2005) consider generalized additive model (GAM) forms for GPD parameters, which allow a given parameter to be represented with one or more 'smooths', i.e., smooth functions, each of which may have a different smoothness. Yee and Stephenson (2007) consider the vector GAM (VGAM) setting of Yee and Wild (1996) for representing EVD parameters with GAM form. More recently, Randell, Turnbull, Ewans, and Jonathan (2016) use spline forms and roughness-penalized priors to represent variation in EVD parameters when modeling significant wave heights, using Markov chain Monte Carlo for inference. Youngman (2019) models exceedances of a threshold with a GPD with parameters of GAM form and a high threshold estimated by GAM form quantile regression, as proposed in Yee and Stephenson (2007) and Northrop and Jonathan (2011).

GAM forms typically consider additive smooths represented with splines. Various packages contributed to R fit EVDs with parameters of GAM or spline form. In particular, **VGAM** (Yee 2010, 2015) allows the GEV and GPD distributions to be fitted with parameters of GAM form. Various EVDs are also available within package **gamlss** (Rigby and Stasinopoulos 2005; Stasinopoulos and Rigby 2007). Alternatively, **ismev**'s `gamGPDfit()` implements Chavez-Demoulin and Davison (2005), i.e., fits a GPD with parameters of GAM form through backfitting. Marginal spline forms are also allowed for GEV parameters in package **SpatialExtremes** (Ribatet 2020), although the package's focus is multivariate analyses, in particular with max-stable processes. Fitting of the GEV with parameters of GAM forms is also possible with package **mgcv** with option `family = "gevlss"`. EVDs can also be fitted using the integrated nested Laplace approximation (INLA) software (Rue, Martino, and Chopin 2009), which specifies smooths as latent Gaussian random fields (GRF) that depend on hyperparameters. Options for GAM-based quantile regression, which can be useful for threshold estimation, include the packages **VGAM** and **qgam** (Fasiolo, Wood, Zaffran, Nedellec, and Goude 2021). Package **quantreg** (Koenker 2021) allows quantile regression using B-splines.

Estimating GAM forms for EVDs under fixed smoothing penalties is fairly straightforward. For example, parameter estimates can maximize a penalized log-likelihood; recall Pauli and Coles (2001). Smoothing parameter (or hyperparameter) selection, however, is perhaps the most challenging part of fitting a distribution with parameters of GAM form. In package

VGAM this is eased by users specifying the degrees of freedom for smooths, from which smoothing parameter estimates are derived. Degrees of freedom are deemed more intuitive to specify than smoothing parameters themselves. Alternatively, **gamlss** includes function `find.hyper()`, which minimizes a generalized Akaike information criterion to find optimal degrees of freedom. Wood (2011) proposes an objective approach to smoothing parameter estimation for exponential family distributions by treating penalized parameters as multivariate Gaussian random effects, which are integrated out by Laplace's method. This gives a marginal likelihood for smoothing parameters; see Section 2.5. Wood, Pya, and Säfken (2016) extend this approach beyond the exponential family. This method is implemented in function `gam()` from package **mgcv** with option `method` set to `"ML"` or `"REML"`. Laplace's method is used by Rue *et al.* (2009) in INLA to integrate out latent GRFs so that hyperparameters can be optimally estimated. Optimal estimation can be beneficial when degrees of freedom cannot easily be user-specified; a GAM form comprising many smooths is one example.

The aim of package **evgam** (Youngman 2022) is to bring together three things: 1) the flexibility of the different smooths available in package **mgcv** for fitting EVDs with parameters of GAM form; 2) objective inference for *all* parameters; and 3) functions for drawing common inferences from extreme value analyses, such as return level estimates with uncertainty quantified. For 1), in particular, **mgcv** offers GAMs incorporating thin plate regression splines, which are particularly attractive for modeling multidimensional processes, such as spatial processes, or interactions between splines formed by tensor products, as implemented in **mgcv** through `te()`. For 3), **evgam** provides functionality for estimating return levels from nonstationary EVD parameters and straightforward quantification of their uncertainty.

Initially **evgam** performed the analysis of Youngman (2019), i.e., using the asymmetric Laplace distribution (ALD) to estimate a quantile of GAM form, and then estimating the distribution of its excesses with a GPD with parameters of GAM form. This article presents extensions that allow estimation of GEV distribution parameters of GAM form. These can be estimated from block maxima or from threshold exceedances through the point process model of Smith (1989). The point process model allows simultaneous estimation of all parameters required for return level estimation, while potentially being less wasteful of data than the block maxima approach. Furthermore, the point process model is implemented through the intuitive r-largest order statistics model representation; see, e.g., Coles (2001, Section 7.9). Finally, **evgam** allows estimation of EVDs based on censored data, which can be useful for data known to be recorded with little precision, and is also available in **gamlss**.

The next section gives details of EVDs available in **evgam**, deriving return levels from them, and a summary of how they are fitted. Section 3 introduces **evgam**'s main functions. Section 4 presents various examples of use of **evgam**, including spatial and temporal models. A brief summary is given in Section 5.

2. Extreme value modeling

2.1. Extreme value distributions

This section outlines the three EVD models supported by **evgam**, and quantile regression via the ALD; see Yu and Moyeed (2001). Fuller details of the EVD models can be found in Coles (2001, Chapters 3, 4 and 7).

Generalized extreme value distribution

The GEV distribution is appropriate for block maxima of sufficiently large blocks. Here years will be considered as blocks, to help intuition; henceforth we will refer to *annual* maxima. A random variable Y that is GEV distributed has cumulative distribution function (CDF)

$$F_{\text{GEV}}(y; \mu, \psi, \xi) = \exp\left\{ - \left[1 + \xi \left(\frac{y - \mu}{\psi} \right) \right]^{-1/\xi} \right\},$$

which is defined for $\{y : 1 + \xi(y - \mu)/\psi > 0\}$ with $(\mu, \psi, \xi) \in \mathbb{R} \times \mathbb{R}^+ \times \mathbb{R}/\{0\}$. The limit $\xi \to 0$ is used for the $\xi = 0$ case, which corresponds to the Gumbel CDF, $\exp(-\exp\{-[(y - \mu)/\psi]\})$. For all models this limit is invoked in **evgam** if $|\xi| < 10^{-6}$.

Generalized Pareto distribution

The GPD is used to model excesses of a high threshold u. For a random variable Y, it is a model for the conditional distribution $(Y - u) \mid (Y > u)$ with CDF

$$F^{(u)}_{\text{GPD}}(y; \psi_u, \xi) = 1 - \left[1 + \xi \left(\frac{y}{\psi_u} \right) \right]^{-1/\xi},$$

which is defined for $\{y : y > 0 \text{ and } 1 + \xi y/\psi_u > 0\}$ with $(\psi_u, \xi) \in \mathbb{R}^+ \times \mathbb{R}/\{0\}$. The exponential CDF, $1 - \exp(-y/\psi_u)$, is used for the $\xi = 0$ case.

Poisson-GPD point process model

The Poisson-GPD point process model is considered as an extension of the GPD model with high threshold u that allows estimation of GEV parameters. For random variables $\{Y_i\}_{i=1,\ldots,n}$ and $y > u$ the Poisson-GPD model has intensity measure

$$\Lambda(A) = n_y(t_2 - t_1) \left[1 + \xi \left(\frac{y - \mu}{\psi} \right) \right]^{-1/\xi},$$

where $A = [t_1, t_2] \times (y, \infty)$, n_y is the time period under study and $t_i = (i - 0.5)/n$.

Asymmetric Laplace distribution (for threshold estimation)

The ALD is not an EVD in the usual sense. It is useful in threshold-based extreme value analyses for allowing quantile estimation (Yu and Moyeed 2001). The GPD and Poisson-GPD models rely on a 'high' threshold. Coles (2001, Chapter 4) discusses assessing its choice. High can be sometimes be intuitively defined through a high quantile, e.g., 0.9, 0.95 or 0.99. Quantile regression can be used to estimate such thresholds, especially covariate-dependent thresholds. The ALD has density function

$$f_{\text{ALD},\tau}(y; u, \sigma, \tau) = \frac{\tau(1 - \tau)}{\sigma} \exp\left\{ -\rho_\tau \left(\frac{y - u}{\sigma} \right) \right\},$$

where $\rho_\tau(y) = y(\tau - I\{y < 0\})$ is the check function, for indicator function $I\{\}$; see Koenker (2005) for an overview of quantile regression. The modified check function of Oh, Lee, and Nychka (2011) is used in **evgam** to ease inference.

2.2. Return levels

Return levels are often sought from extreme value analyses. Let F_{ann} denote the CDF of the annual maximum. Then the return level, z_p, corresponding to return period $1/p$ years, satisfies $F_{\text{ann}}(z_p) = 1 - p$.

GEV and Poisson-GPD models

For the GEV distribution

$$z_p = \mu - \frac{\psi}{\xi}\left\{1 - [-\log(1-p)]^{-\xi}\right\}, \tag{1}$$

when $\xi \neq 0$ and $\mu - \psi \log(-\log(1-p))$ otherwise. Equation 1 also applies to the Poisson-GPD model if formulated in terms of annual maxima.

GPD model

For a GPD representing independent excesses of threshold u, where n_y observations occur each year and such that $\mathsf{P}(Y > u) = \zeta$,

$$z_p = u + \frac{\psi_u}{\xi}[(n_y \zeta / p)^{\xi} - 1], \tag{2}$$

when $\xi \neq 0$ and $u + \psi_u \log(n_y \zeta / p)$ otherwise.

For the GEV it is typically reasonable to assume that annual maxima are independent. For the GPD, however, excesses of a threshold may occur in clusters, which requires that Equation 2 be adjusted accordingly. This is achieved through the extremal index, $0 < \theta \leq 1$, so that $z_p = u + \frac{\psi_u}{\xi}[(n_y \zeta \theta / p)^{\xi} - 1]$ when $\xi \neq 0$ and $u + \psi_u \log(n_y \zeta \theta / p)$ otherwise. Currently **evgam** only allows relatively simple, constant estimates of θ based on the moment-based estimator of Ferro and Segers (2003). An example is given in Section 4.2.

2.3. Nonstationarity

Outline

Now consider $Y(\boldsymbol{x})$, a random variable indexed by covariate $\boldsymbol{x}$. The purpose of **evgam** is to allow straightforward fitting of EVDs with parameters that vary flexibly with $\boldsymbol{x}$. The following notation will be used. For the GEV, suppose that annual maxima $Y(\boldsymbol{x}) \sim \mathit{GEV}(\mu(\boldsymbol{x}), \psi(\boldsymbol{x}), \xi(\boldsymbol{x}))$; for the GPD, that $Y(\boldsymbol{x}) - u(\boldsymbol{x}) \mid Y(\boldsymbol{x}) > u(\boldsymbol{x}) \sim \mathit{GPD}(\psi_u(\boldsymbol{x}), \xi(\boldsymbol{x}))$; for the Poisson-GPD model, that $Y(\boldsymbol{x}) - u(\boldsymbol{x}) \mid Y(\boldsymbol{x}) > u(\boldsymbol{x}) \sim \mathit{GEV}(\mu(\boldsymbol{x}), \psi(\boldsymbol{x}), \xi(\boldsymbol{x}))$; and for the ALD that $Y(\boldsymbol{x}) \sim \mathit{ALD}(u(\boldsymbol{x}), \sigma(\boldsymbol{x}))$.

Return levels

If covariate $\boldsymbol{x}$ relates to time, return levels typically need different treatment. Two examples are given here for illustration: one for the GEV case, and one for the GPD case. These should be sufficient to inform other situations.

Suppose that covariate $\boldsymbol{x}$ defines month, i.e., $\boldsymbol{x}_i = \text{month}(i)$, for $i = 1, \ldots, n$, and that $Y(\boldsymbol{x}_i) \sim GEV(\mu(\boldsymbol{x}_i), \psi(\boldsymbol{x}_i), \xi(\boldsymbol{x}_i))$ are monthly maxima, which may have a different distribution each month. The CDF of the annual maximum then takes the composite form

$$F_{\text{ann}}(z) = \prod_{\boldsymbol{x}_j=1}^{n_y} \left\{F_{\text{GEV}}(z; \mu(\boldsymbol{x}_j), \psi(\boldsymbol{x}_j), \xi(\boldsymbol{x}_j))\right\}^{n_y w(\boldsymbol{x}_j)}, \tag{3}$$

where $n_y = 12$ and $w(\boldsymbol{x}_j)$ are weights: $w(1) = w(3) = w(5) = w(7) = w(8) = w(10) = w(12) = 31/365$, $w(2) = 28/365$ and $w(4) = w(6) = w(9) = w(11) = 30/365$. (This, for simplicity, considers only 365-day years.) The $1/p$-year return level, z_p, satisfies $F_{\text{ann}}(z_p) = 1 - p$. Unless z_p has closed form, which is rare, it must be found numerically. This approach to return level estimation is implemented in Section 4.2.

The case of covariate $\boldsymbol{x}$ being time-dependent is handled similarly with the GPD. Now suppose $\boldsymbol{x}_i = \text{day}(i)$. The composite form for F_{ann} is then given by

$$F_{\text{ann}}(z) = \prod_{\boldsymbol{x}_j=1}^{n_y} \left\{F_{\text{GPD}}(z; \zeta(\boldsymbol{x}_j), \psi_u(\boldsymbol{x}_j), \xi(\boldsymbol{x}_j))\right\}^{n_y w(\boldsymbol{x}_j)},$$

where $n_y = 365$ and F_{GPD} denotes the unconditional distribution of a random variable Y:

$$F_{\text{GPD}}(y; \zeta, \psi_u, \xi) = 1 - \zeta \left[1 - F^{(u)}_{\text{GPD}}(y - u; \psi_u, \xi)\right], \quad \text{for } y > u, \tag{4}$$

and $\zeta = \mathsf{P}(Y > u)$. Here we would take $w(\boldsymbol{x}_j) = 1/n_y$, for all $\boldsymbol{x}_j$. Again z_p satisfying $F_{\text{ann}}(z_p) = 1 - p$ is typically only found numerically. This approach is demonstrated in Section 4.2 and, additionally, continuous time-dependent $\boldsymbol{x}$ is considered. Then infinitely many values exist for $\boldsymbol{x}$. F_{ann} formed over a product would therefore be an approximation based on the 365-point set $\{1, \ldots, 365\}$. More or fewer points may benefit this approximation's accuracy and computational cost. A 50-point set is used in Section 4.2.

In the above, composite forms for F_{ann} are easily modified for non-monthly maxima or non-daily threshold exceedances. For example, the former might instead use 'seasonal' maxima, where season may be problem-specific, or the latter might use hourly threshold exceedances. The return period need not be defined in terms of years, either.

2.4. Inference

For the GEV model, consider annual maxima $\{Y(\boldsymbol{x}_i)\}_{i=1,\ldots,n}$. We might obtain these by dividing a sequence of random variables by year and retaining each year's maximum. Let f_* denote a model's density function. The GEV likelihood is then

$$L(\boldsymbol{\mu}, \boldsymbol{\psi}, \boldsymbol{\xi}) = \prod_{i=1}^{n} f_{\text{GEV}}(y(\boldsymbol{x}_i)); \mu(\boldsymbol{x}_i), \psi(\boldsymbol{x}_i), \xi(\boldsymbol{x}_i)),$$

with $\boldsymbol{\mu} = (\mu(\boldsymbol{x}_1), \ldots, \mu(\boldsymbol{x}_n))$, $\boldsymbol{\psi} = (\psi(\boldsymbol{x}_1), \ldots, \psi(\boldsymbol{x}_n))$ and $\boldsymbol{\xi} = (\xi(\boldsymbol{x}_1), \ldots, \xi(\boldsymbol{x}_n))$. For the GPD, now let $\{Y(\boldsymbol{x}_i)\}_{i=1,\ldots,n}$ be n threshold excesses. They would be obtained by retaining the threshold exceedances from a sequence of random variables and then calculating their excesses of the threshold. The GPD model likelihood is

$$L(\boldsymbol{\psi}_u, \boldsymbol{\xi}) = \prod_{i=1}^{n} f_{\text{GPD}}(y(\boldsymbol{x}_i); \psi_u(\boldsymbol{x}_i), \xi(\boldsymbol{x}_i)),$$

with $\boldsymbol{\psi}_u = (\psi_u(\boldsymbol{x}_1), \ldots, \psi_u(\boldsymbol{x}_n))$ and $\boldsymbol{\xi} = (\xi(\boldsymbol{x}_1), \ldots, \xi(\boldsymbol{x}_n))$.

The Poisson-GPD model's likelihood is slightly more challenging as it requires integration over all possible $\boldsymbol{x}$, i.e., $\mathcal{X}$. Consequently **evgam** currently only considers models where integration is over time-dependent $\boldsymbol{x}$, over which GEV parameters must be constant. Hence, consider $\{Y_t(\boldsymbol{x}_i)\}$ for $i = 1, \ldots, n$ and times $t = 1, \ldots, T$. The Poisson-GPD model's likelihood is

$$\begin{aligned} L(\boldsymbol{\mu}, \boldsymbol{\psi}, \boldsymbol{\xi}) = \prod_{i=1}^{n} \Bigg[\exp\left\{ -n_y \left[1 + \xi(\boldsymbol{x}_i) \left(\frac{y^{(r)}(\boldsymbol{x}_i) - \mu(\boldsymbol{x}_i)}{\psi(\boldsymbol{x}_i)} \right) \right]^{-\frac{1}{\xi(\boldsymbol{x}_i)}} \right\} \times \\ \prod_{t=1}^{r} \psi^{-1} \left[1 + \xi(\boldsymbol{x}_i) \left(\frac{y^{(t)}(\boldsymbol{x}_i) - \mu(\boldsymbol{x}_i)}{\psi(\boldsymbol{x}_i)} \right) \right]^{-\frac{1}{\xi(\boldsymbol{x}_i)} - 1} \Bigg] \end{aligned} \qquad (5)$$

for time period n_y, n-vectors $\boldsymbol{\mu}$, $\boldsymbol{\psi}$ and $\boldsymbol{\xi}$ and where $y^{(t)}(\boldsymbol{x})$, for $t = 1, \ldots, T$, denote the order statistics of sample $y_1(\boldsymbol{x}), \ldots, y_T(\boldsymbol{x})$ with $r < T$ chosen by the user. An example where $\mu(\boldsymbol{x})$, $\psi(\boldsymbol{x})$ and $\xi(\boldsymbol{x})$ vary with spatial locations is given in Section 4.1.

The ALD is fitted to data relating to original random variables $\{Y(\boldsymbol{x}_i)\}$ for $i = 1, \ldots, n$. Its likelihood is therefore

$$L(\boldsymbol{u}, \boldsymbol{\sigma}) = \prod_{i=1}^{n} f_{\text{ALD}}(y(\boldsymbol{x}_i); u(\boldsymbol{x}_i), \sigma(\boldsymbol{x}_i)),$$

with $\boldsymbol{u} = (u(\boldsymbol{x}_1), \ldots, u(\boldsymbol{x}_n))$ and $\boldsymbol{\sigma} = (\sigma(\boldsymbol{x}_1), \ldots, \sigma(\boldsymbol{x}_n))$.

Interval-censored data can also be fitted with **evgam**. Suppose $[y_-(\boldsymbol{x}_i), y_+(\boldsymbol{x}_i)]$ denotes the censoring interval of $y(\boldsymbol{x}_i)$, a realization from $F(\,;\cdot)$. Then the likelihood takes the form

$$L(\cdot) = \prod_{i=1}^{n} \left[F(y_+(\boldsymbol{x}_i); \cdot) - F(y_-(\boldsymbol{x}_i); \cdot) \right].$$

2.5. Generalized additive modeling

Package **evgam** is primarily designed to allow nonstationarity in EVD parameters by assuming GAM forms in covariate $\boldsymbol{x}$.

Basis representations

GAM forms for EVD parameters rely on basis representations. Consider covariate $\boldsymbol{x}$ and GEV parameters $\mu(\boldsymbol{x})$, $\psi(\boldsymbol{x})$ and $\xi(\boldsymbol{x})$. **evgam** relates parameters via fixed link functions to η_*, which has a basis representation. For GEV, $\mu(\boldsymbol{x}) = \eta_\mu(\boldsymbol{x})$, $\log \psi(\boldsymbol{x}) = \eta_\psi(\boldsymbol{x})$ and $\xi(\boldsymbol{x}) = \eta_\xi(\boldsymbol{x})$, where

$$\eta_*(\boldsymbol{x}) = \beta_0 + \sum_{k=1}^{K} \sum_{d=1}^{D_k} \beta_{kd} b_{kd}(\boldsymbol{x})$$

with β_{kd} and b_{kd} basis coefficients and functions, respectively. The upshot of the basis representation is that we can write $\eta_*(\boldsymbol{x}) = \mathbf{x}^\top\boldsymbol{\beta}$ where $\mathbf{x}^\top$ is a row of a design matrix $\mathbf{X}$, which has elements determined by the choice of the b_{kd} basis functions and $1 + \sum_{k=1}^K D_k$ columns, each of which corresponds to an element of $\boldsymbol{\beta}^\top = (\beta_0, \beta_{11}, \ldots, \beta_{KD_K})$. The log link is used through **evgam** for any parameters with support $\mathbb{R}^+$.

Penalized likelihood

Various likelihoods were introduced in Section 2.4. For data $\boldsymbol{y} = \{y_1, \ldots, y_n\}$ with corresponding covariates $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, estimating an EVD corresponds to estimating basis coefficients $\boldsymbol{\beta}$. Each likelihood from Section 2.4 can then be written $L(\boldsymbol{\beta})$ with log-likelihood $\ell(\boldsymbol{\beta})$.

To estimate EVD parameters a penalized log-likelihood of the form

$$\ell(\boldsymbol{\beta}_{\boldsymbol{\lambda}}, \boldsymbol{\lambda}) = \ell(\boldsymbol{\beta}_{\boldsymbol{\lambda}}) - \frac{1}{2}\boldsymbol{\beta}^\top\mathbf{S}_{\boldsymbol{\lambda}}\boldsymbol{\beta},$$

is considered for smoothing parameters $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_K)$, where $\mathbf{S}_{\boldsymbol{\lambda}}$ is a penalty matrix with elements determined by the chosen b_{kd} basis functions. $\mathbf{S}_{\boldsymbol{\lambda}}$ may be written $\mathbf{S}_{\boldsymbol{\lambda}} = \sum_{k=1}^K \lambda_k\mathbf{S}_k$, where rows and columns of matrix $\mathbf{S}_k$ corresponding to $b_{k'd}$, $k' \neq k$, comprise zeros. Often the non-zero terms in the $\mathbf{S}_k$ matrices are non-overlapping. One contrary example are penalties constructed by tensor products (De Boor 1978); see Wood (2011) for fuller details.

Restricted maximum likelihood

Following Wood (2011), $\boldsymbol{\beta}$ can be integrated out using Laplace's method, which results in a restricted log-likelihood of the form

$$\ell(\boldsymbol{\lambda}) = \ell(\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}, \boldsymbol{\lambda}) + \frac{1}{2}\log|\mathbf{S}_{\boldsymbol{\lambda}}|_+ - \frac{1}{2}\log|\mathbf{H}(\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}})| + \text{constant}, \tag{6}$$

where $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}$ maximizes $\ell(\boldsymbol{\beta}_{\boldsymbol{\lambda}}, \boldsymbol{\lambda})$ for given $\boldsymbol{\lambda}$, $\mathbf{H}(\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}) = -\nabla\nabla^\top\ell(\boldsymbol{\beta}, \boldsymbol{\lambda})|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}}$ and $|\mathbf{S}_{\boldsymbol{\lambda}}|_+$ denotes the product of positive eigenvalues of matrix $\mathbf{S}_{\boldsymbol{\lambda}}$. Optimal smoothing parameters, $\hat{\boldsymbol{\lambda}}$, can be found by numerically maximizing $\ell(\boldsymbol{\lambda})$, which is typically best performed through Newton or quasi-Newton methods, as implemented by **evgam**. Fitting a model therefore involves inner iterations, for given $\boldsymbol{\lambda}$, which give $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}$, and outer iterations, which give $\hat{\boldsymbol{\lambda}}$.

3. Features of evgam

3.1. Function `evgam()`

Basic use

The package **evgam** mainly relies on function `evgam()`. Its main arguments are

```
evgam(formula, data, family)
```

Typically `formula` is a list comprising formulae: one formula compatible with `mgcv::s()` for each EVD parameter. Hence, see the help for `mgcv::s()` for details of its use. If a single formula is supplied, it is repeated for each EVD parameter so that the same form is assumed for each parameter. Use of `data` is the same as for, e.g., `lm()`. Interval-censored data can also be handled with `formula`. Supplying `cens(left, right)` as the response specifies that `data$left` and `data$right` are variables giving lower and upper ends of censoring intervals, respectively. Any response data for which `data$left` and `data$right` are equal are treated as uncensored. (Note that left- and right-censored data can be handled with sufficiently high and low lower and upper interval ranges, respectively.) An example fitting the GPD to censored data is given in Section 4.1.

The default `family` is `"gev"`, which corresponds to the GEV distribution. GPD, Poisson-GPD and ALD models are specified with `"gpd"`, `"pp"` and `"ald"`, respectively. **evgam** also supports fitting of exponential, `"exponential"`, Weibull, `"weibull"`, and Gaussian, `"gauss"`, distributions.

For the ALD, the quantile to be estimated must be given: supplying `ald.args = list(tau = 0.9)`, for example, gives an estimate of the 0.9 quantile. For the point process model, the time period under study and the number of order statistics to use are required: supplying `pp.args = list(ny = 30, r = 50)` specifies a 30-period time period, e.g., 30 years, if parameters representative of annual maxima are sought, and 50 order statistics. Note that `r = -1` uses all order statistics. Fitting ALD and Poisson-GPD models is demonstrated in Section 4.2 and Section 4.1, respectively. Section 4.1 also demonstrates how `pp.args$id` may be used to specify partitions of `data` over which integration is not required.

Additional options

The default values used by **evgam** are designed to be robust. In some circumstances, however, changes to some arguments' default values may improve performance. First consider `trace`, which accepts 0 (default), 1, 2 or -1; increasing numbers report more on optimization iterations, and -1 reports nothing. `trace` can be useful for ensuring that inner and/or outer iterations have converged. There are two arguments that may improve speed for large datasets. First, `maxdata` specifies the maximum number of rows in `data` that will be used in model fitting: if `nrow(data) > maxdata`, then `maxdata` rows of `data` are sampled without replacement. Second, `maxspline` specifies the maximum number of rows in `data` that are supplied to `mgcv::s()` to create bases; all rows of `data` are then used for fitting unless `maxdata > maxspline` is also invoked. Initial values for $\rho_k = \log \lambda_k$, $k = 1, \ldots, K$, are supplied with `rho0`; **evgam**'s default is $\lambda_k = 1$ for all k. Providing a scalar specifies the same initial value for each λ_k, whereas a vector of length K allows different initial values. Argument `inits` allows initial values for $\boldsymbol{\beta}_\lambda$ to be specified in various ways, such as subsets of $\boldsymbol{\beta}_\lambda$. Argument `outer` specifies how the restricted log-likelihood of Equation 6 is optimized: the default, `"BFGS"`, uses the BFGS quasi-Newton method; `"Newton"` uses Newton's method; and `"FD"` uses BFGS with finite-difference approximations to the gradient of Equation 6 w.r.t. each ρ_k. See `evgam()`'s help file for details of its other options.

3.2. Function `qev()`

Also included in **evgam** is `qev()` for quantiles of EVDs. It solves $F_{\text{ann}}(z_p) = p$, numerically where necessary, for z_p. Its arguments are

```
qev(p, loc, scale, shape, m = 1, alpha = 1, theta = 1, family, tau = 0)
```

In the above `p` is p in $F_{\mathrm{ann}}(z_p) = p$, and `loc`, `scale` and `shape` are an EVD's location, scale and shape parameters, respectively. In terms of Section 2.3, `m` corresponds to n_y, `alpha` to $w(\,)$, `theta` to θ, `family` is that supplied to `evgam()`, and `tau` corresponds to $1 - \zeta$.

4. Illustrations

Illustrations for **evgam** are given below. All require **evgam** to be loaded, which is done with

```
R> library("evgam")
```

4.1. Spatial modeling: Colorado precipitation

To illustrate the key functionality of **evgam** the dataset `COprcp` will be used, which contains daily precipitation amounts, `prcp`, in mm on day `date` at locations identified by `meta_row` for part of Colorado, US. This was the domain studied in Cooley, Nychka, and Naveau (2007). Each location's metadata corresponds to a row in `COprcp_meta`.

The COprcp data

The data can be loaded and conjoined with the metadata using

```
R> data("COprcp", package = "evgam")
R> COprcp <- cbind(COprcp, COprcp_meta[COprcp$meta_row, ])
```

The dataset `COprcp` also includes `COelev`, gridded elevation data for the study region. A plot of gridded elevations (Figure 1) can be obtained with

```
R> brks <- pretty(COelev$z, 50)
R> cols <- hcl.colors(length(brks) - 1, "YlOrRd", rev = TRUE)
R> image(COelev, breaks = brks, col = cols, asp = 1)
```

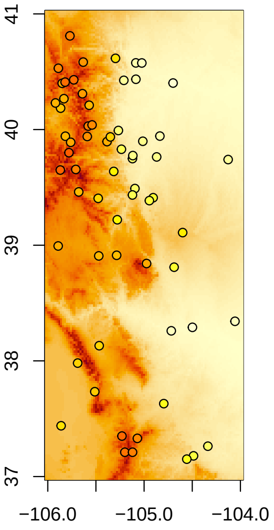

Figure 1: Gridded and station-based elevation data for study region.

The function colplot() is included in **evgam** to plot points that are colored according to a variable. Station elevations can be superimposed on the gridded elevations of Figure 1 with

```
R> colplot(COprcp_meta$lon, COprcp_meta$lat, COprcp_meta$elev,
+    breaks = brks, palette = cols, add = TRUE)
```

Before fitting any models, a 'data.frame' for plotting, COprcp_plot, is created using

```
R> COprcp_plot <- expand.grid(lon = COelev$x, lat = COelev$y)
R> COprcp_plot$elev <- as.vector(COelev$z)
```

Subsequent models will use elevation as a covariate, so it has been included in COprcp_plot. Coordinate and covariate names match those in COprcp_meta.

GEV model

First we model annual maxima using the GEV distribution, introduced in Section 2.1. This model will be implemented by creating a 'data.frame' comprising annual maxima at each station. Since date is of class 'Date', this can be done with

```
R> COprcp$year <- format(COprcp$date, "%Y")
R> COprcp_gev <- aggregate(prcp ~ year + meta_row, COprcp, max)
```

which aggregates over meta_row, i.e., over the station IDs, and then the metadata can be added to COprcp_gev with

```
R> COprcp_gev <- cbind(COprcp_gev, COprcp_meta[COprcp_gev$meta_row, ])
```

The next step is to provide formulae for smooths to pass to mgcv::s(). A spatial model will be fitted that allows spatial variation in the GEV's location and scale parameters. Spatial variation is achieved with thin plate regression splines, which are mgcv::s()'s default. The basis dimension, k, has been specified to differ with GEV parameter. The GEV's shape parameter is assumed constant. The value of k caps a smooth's degrees of freedom, and hence, in some sense, its ultimate wiggliness. In practice, k should be chosen larger than a smooth's expected degrees of freedom so that the smoothing parameters control the effective degrees of freedom. The GEV's location parameter also includes a smooth in elev, station elevation. This is specified as a cubic regression spline, bs = "cr", with k left at its default. The smooths for all GEV parameters are then specified with

```
R> fmla_gev <- list(prcp ~ s(lon, lat, k = 30) + s(elev, bs = "cr"),
+    ~ s(lon, lat, k = 20), ~ 1)
```

To fit the model we issue

```
R> m_gev <- evgam(fmla_gev, COprcp_gev, family = "gev")
```

(but could have omitted family = "gev" above since it is evgam()'s default).

Having fitted the model, it is sensible to check whether smooths are necessary, and if so whether they are well specified. This can be done through summary() with

```
R> summary(m_gev)

** Parametric terms **

location
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    28.56       0.26  111.89   <2e-16

logscale
            Estimate Std. Error t value Pr(>|t|)
(Intercept)     2.24       0.02  118.07   <2e-16

shape
            Estimate Std. Error t value Pr(>|t|)
(Intercept)     0.08       0.02    5.08 1.92e-07

** Smooth terms **

location
              edf max.df Chi.sq Pr(>|t|)
s(lon,lat) 19.27     29 178.23   <2e-16
s(elev)     5.19      9  19.39  0.00139

logscale
              edf max.df Chi.sq Pr(>|t|)
s(lon,lat) 13.94     19 211.15   <2e-16
```

The necessity of smooths can be checked through p values. In this example they are all $\ll 0.01$, indicating the smooths are beneficial. All one- or two-dimensional smooths can be viewed with `plot()`, i.e.,

```
R> plot(m_gev)
```

which is shown in Figure 2. Often predictions are sought from a fitted model. These are achieved with `predict()`. Predictions for the GEV's three parameters for `COprcp_plot` can be obtained with

```
R> gev_pred <- predict(m_gev, COprcp_plot, type = "response")
R> head(gev_pred)

  location    scale      shape
1 12.79505 5.344232 0.07941214
2 13.09313 5.422900 0.07941214
3 13.38081 5.503366 0.07941214
4 13.67835 5.585679 0.07941214
5 13.97230 5.669885 0.07941214
6 14.27654 5.756028 0.07941214
```

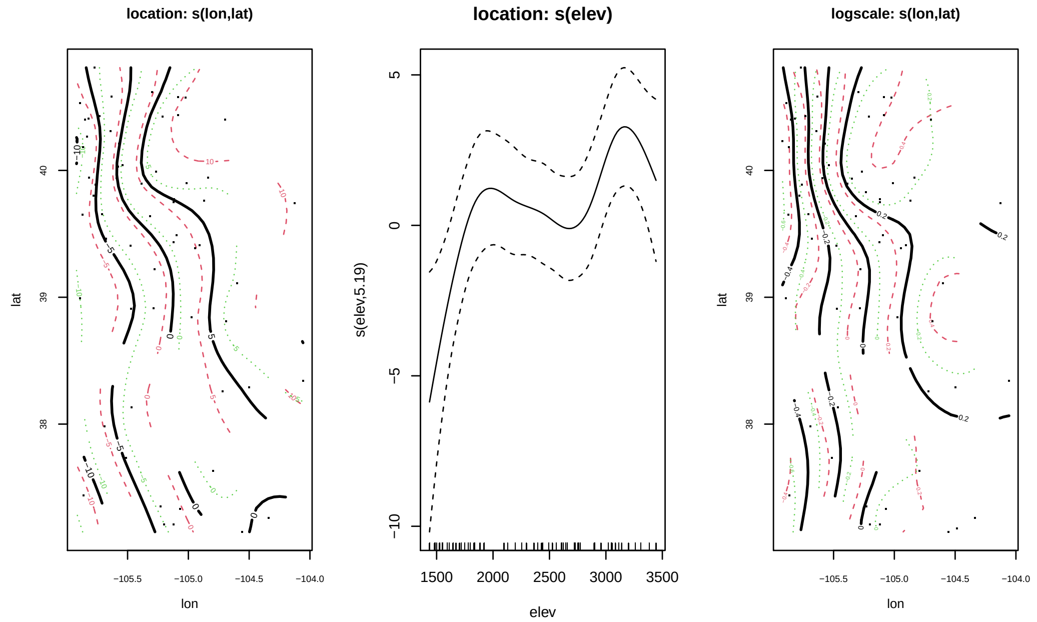

Figure 2: Output of `plot(m_gev)` for Colorado precipitation annual maxima.

where `head()` suppresses estimates for all but the first six rows of `predict()`'s output. Note that `type = "response"` is used to predict parameters on their original scale, similarly to the `predict()` method for '`glm`' objects. Hence `gev_pred` is a three-column '`data.frame`' with columns for the GEV location, scale and shape parameters, respectively. Predictions can be viewed using `image()` and a few lines of code (omitting the constant shape parameter), e.g.,

```
R> for (i in 1:2) {
+    plot.list <- COelev
+    plot.list$z[] <- gev_pred[, i]
+    image(plot.list, asp = 1)
+    title(paste("GEV", names(gev_pred)[i]))
+  }
```

which is shown in Figure 3. Lastly, the 100-year return level for the locations in `COprcp_plot` can be estimated. This is an estimate of the 0.99 quantile of the distribution of the annual maximum for *each* location and achieved with

```
R> gev_rl100 <- predict(m_gev, COprcp_plot, prob = 0.99)
R> head(gev_rl100)
```

```
   q:0.99
1 42.47033
2 43.20524
3 43.93974
4 44.69434
5 45.45587
6 46.23844
```

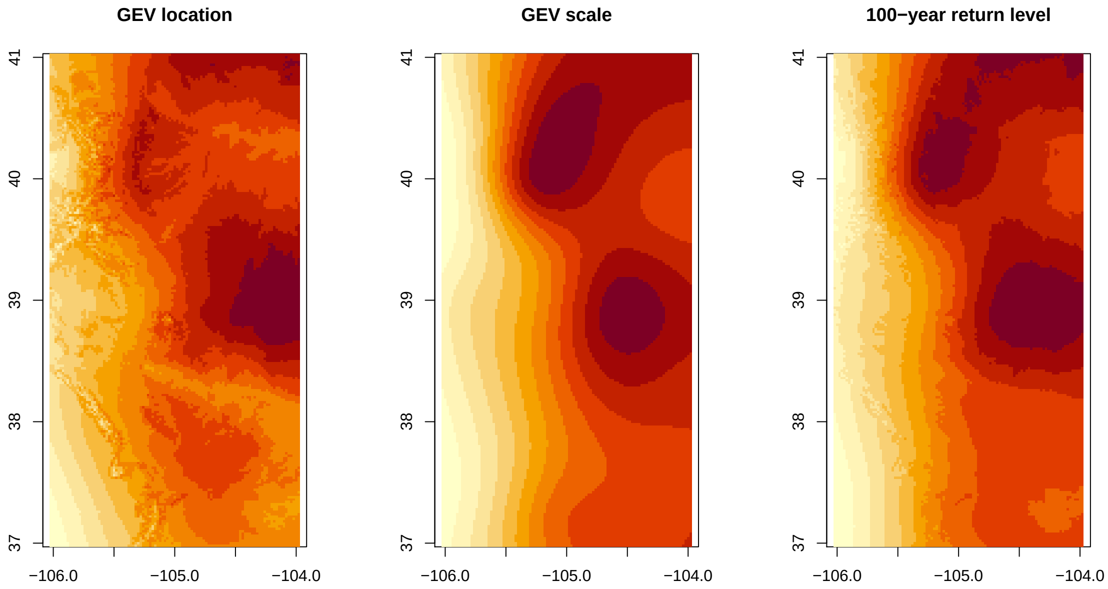

Figure 3: Plots of GEV parameter estimates for Colorado precipitation annual maxima and of the 100-year return level estimate.

and plotted using

```
R> rl100 <- COelev
R> rl100$z[] <- gev_rl100[, 1]
R> image(rl100, asp = 1)
R> title("100-year return level")
```

which is also shown in Figure 3. Uncertainty estimates, in particular for return levels, are covered in Section 4.3.

GPD model

The GPD is used to model excesses of a high threshold. Here, following Cooley *et al.* (2007), the threshold is set at 11.4mm using

```
R> threshold <- 11.4
```

To fit the GPD only threshold exceedances are considered. Setting excesses corresponding to non-exceedances as `NA` ensures that only exceedances are modeled, which is done using

```
R> COprcp$excess <- COprcp$prcp - threshold
R> is.na(COprcp$excess[COprcp$excess <= 0]) <- TRUE
```

A similar formula, in terms of smooths, is used for the GPD model as was used for the GEV model, although this model comprises only two parameters and a non-constant shape parameter is allowed. A smooth with `elev` is included for the GPD's scale parameter, which is partly motivated by use of a constant threshold. A varying threshold model is given in Section 4.2. The GPD model is fitted with

```
R> fmla_gpd <- list(excess ~ s(lon, lat, k = 20) + s(elev, bs = "cr"),
+    ~ s(lon, lat, k = 15))
R> m_gpd <- evgam(fmla_gpd, COprcp, family = "gpd")
```

Summaries, plots and predictions can be produced for `m_gpd` as demonstrated above for `m_gev`, and so are not demonstrated again. Using `predict(..., prob = ...)` if `family = "gpd"` uses Equation 2. The example of Section 4.2 demonstrates return level estimation in the presence of dependence.

Poisson-GPD model

For the point process model, following Section 2.4, our data will be considered realizations of $\{Y_t(\boldsymbol{x})\}$ for location $\boldsymbol{x}$ in region $\mathcal{X}$ and time $t = 1, \dots, T$. Hence covariate $\boldsymbol{x}$ is not time-dependent, and log-likelihood (5) is used. If different locations have different T, $n_y(\boldsymbol{x})$ should be used in log-likelihood (5). **evgam** facilitates that by allowing vector `pp.args$ny`. Note that `names(pp.args$ny)` must match the unique `pp.args$id` values to ensure that the correct $n_y(\boldsymbol{x})$ and $\{Y_t(\boldsymbol{x})\}$ coincide.

Different stations in `COprcp` are identified by variable `id`. We want to assume a constant point process rate for a given `id`. We do this by setting `pp.args$id` to `"id"`. (Double use of 'id' is a coincidence.) For this model `fmla_gev` is re-used and then `evgam()` called with

```
R> pp_args <- list(id = "id", ny = 30, r = 45)
R> m_pp <- evgam(fmla_gev, COprcp, family = "pp", pp.args = pp_args)
```

In the above the 45 largest observations at each station are used, and 30 periods of observation at each station are specified. `COprcp` comprises 30 years' data (aside from a few missing values) at each station; hence the Poisson-GPD model's GEV parameter estimates will represent the distribution of the annual maximum.

Summaries, plots and predictions can be produced for `m_pp` similarly to `m_gev`, and so are again omitted for brevity. Note that the r-largest order statistics at a given station may exhibit dependence similarly to threshold excesses and so the same considerations for `predict(..., prob = ...)` as for the GPD apply.

Censored response data and tensor products: GPD model revisited

Cooley *et al.* (2007) allude to precipitation being recorded with relatively little precision. Sometimes such data may want to be treated as censored. For example, continuous data recorded to the nearest integer, x, say, could be treated as interval-censored on $[x-0.5, x+0.5)$. Alternatively, measurement x might be given with stated tolerance δ, i.e., $x \pm \delta$, so that the response should be treated as interval-censored on $[x - \delta, x + \delta]$. Cooley *et al.* (2007) state that some precipitation values were recorded to the nearest tenth of an inch, i.e., $\sim$ 2.5mm. One option for setting up the censoring interval is

```
R> delta <- 2.5
R> COprcp$lo <- pmax(COprcp$excess - delta, 1e-6)
R> COprcp$hi <- COprcp$excess + delta
```

Tensor products, e.g., De Boor (1978) and Wood (2006), can be used to specify interactions between smooths. For example, instead of a thin plate regression spline, a two-dimensional

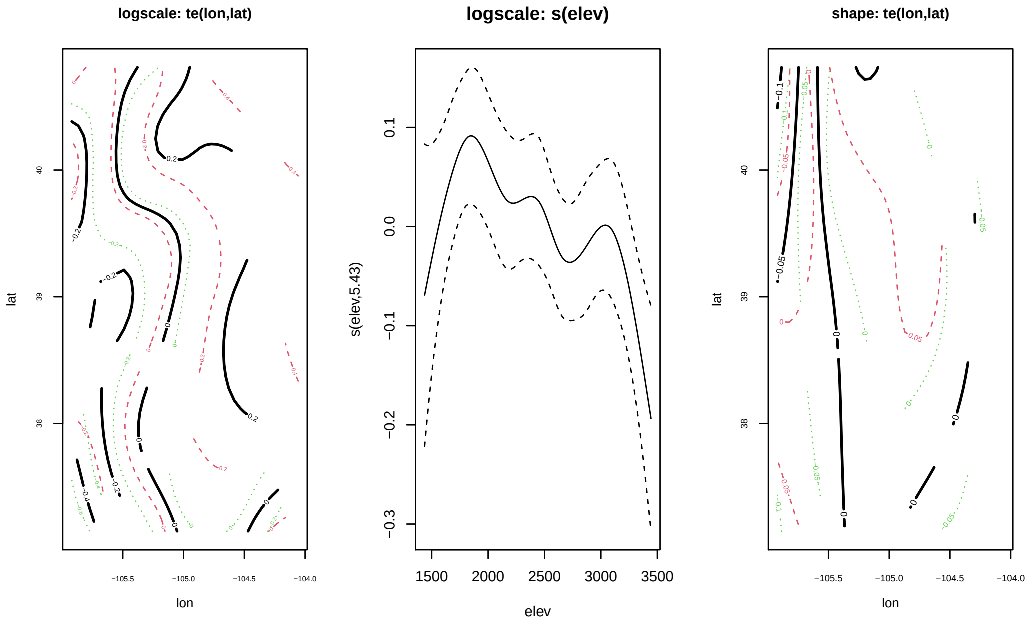

Figure 4: Output of `plot(m_gpg_cens)` for Colorado precipitation threshold exceedances treated as censored with spatial smooths formed from tensor products.

smooth can be formed through the tensor product of two one-dimensional smooths. The earlier GPD formula is modified for interval-censored response data and spatial smooths formed from two cubic regression splines with

```
R> fmla_gpd_cens <- list(cens(lo, hi) ~ te(lon, lat, k = c(6, 8)) +
+    s(elev, bs = "cr"), ~ te(lon, lat, k = c(6, 8)))
```

which specifies rank 6 and rank 8 cubic regression splines for longitude and latitude, respectively (a choice based on the tall rectangular shape of the domain). The GPD is then fit as above, but with a new `formula`, and plotted with

```
R> m_gpd_cens <- evgam(fmla_gpd_cens, COprcp, family = "gpd")
R> plot(m_gpd_cens)
```

which is shown in Figure 4.

4.2. Temporal modeling: Fort Collins temperatures

This example considers `FCtmax`, a data frame comprising daily maximum temperatures, `tmax`, in degrees Celsius at Fort Collins, Colorado, US. The data cover 1st January 1970 to 31st December 2019. There are 95 missing values during this period. Two different approaches to assuming that the distribution of extreme temperatures changes throughout the year are considered. The aim is to estimate the 100-year return level.

The data are loaded using

```
R> data("FCtmax", package = "evgam")
```

GEV model for monthly maxima

The first model uses monthly maxima and the first step is to identify the monthly maxima. Dates are identified by `date`, of class '`Date`', so years and months are obtained with

```
R> FCtmax$year <- format(FCtmax$date, "%Y")
R> FCtmax$month <- format(FCtmax$date, "%m")
```

and then `aggregate()` can be used to find the monthly maxima with

```
R> FCtmax_mnmax <- aggregate(tmax ~ year + month, FCtmax, max)
```

Separating `FCtmax_mnmax` by month with `split()`, i.e.,

```
R> FCtmax_mn <- split(FCtmax_mnmax, FCtmax_mnmax$month)
```

is one option to proceed, which gives a '`list`' of '`data.frame`'s, each comprising monthly maxima over years for a given month.

GEV parameter estimates for each month's maxima are obtained with

```
R> fmla_simple <- list(tmax ~ 1, ~ 1, ~ 1)
R> gev_fits <- lapply(FCtmax_mn, evgam, formula = fmla_simple,
+    family = "gev")
R> gev_pars <- sapply(gev_fits, coef)
```

where `fmla_simple` specifies that for a given month all GEV parameters are constant.

The function `qev()` is then used to estimate the 100-year return level using Equation 3 from Section 2.3. This requires the weights $w(\boldsymbol{x}_i)$ for $i = 1, \ldots, 12$. These are simply

```
R> weights <- (1/365.25) * c(31, 28.25, 30)[c(1, 2, 1, 3, 1, 3, 1, 1, 3, 1,
+    3, 1)]
```

and are supplied to `qev()`, documented in Section 3.2, using

```
R> rl_100_gev <- qev(0.99, gev_pars[1, ], exp(gev_pars[2, ]), gev_pars[3, ],
+    m = 12, alpha = weights, family = "gev")
```

This gives a 100-year return level estimate of 39.37°C.

GPD model for daily threshold exceedances

What is an extreme temperature at one time of the year is different from that occurring at another time of the year. As a result, extreme values are now defined as exceedances of a time-varying threshold. The threshold itself is estimated as the 99th percentile by quantile regression. Hence $\zeta = 0.01$, given Equation 4 from Section 2.3, so we set

```
R> zeta <- 0.01
```

A threshold estimate that varies over the course of a year and that is the same and continuous from year to year is sought. This is achieved through a *cyclic* cubic regression spline, specified with `bs = "cc"` in `mgcv::s()`. The variable `cyc` is therefore created using

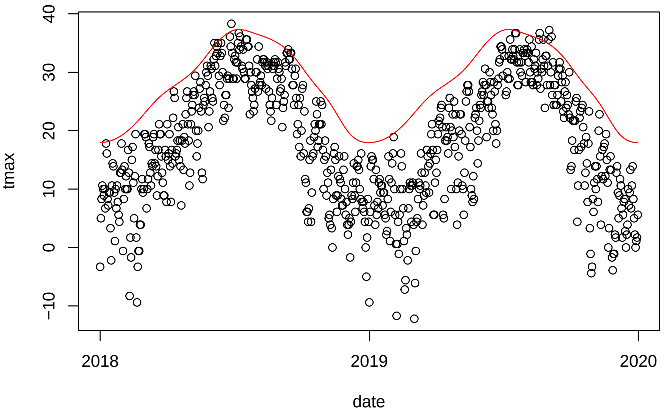

Figure 5: Daily maximum temperatures at Fort Collins for 2018 and 2019 with a cyclic estimate of the 99th percentile superimposed.

```
R> FCtmax$cyc <- as.integer(FCtmax$date) %% 365.25
```

The formula for the model is specified, and then the model fitted, using

```
R> FC_fmla_ald <- list(tmax ~ s(cyc, bs = "cc", k = 15),
+    ~ s(cyc, bs = "cc"))
R> FC_ald <- evgam(FC_fmla_ald, FCtmax, family = "ald",
+    ald.args = list(tau = 1 - zeta))
```

Variables for the estimated threshold, `threshold`, and resulting excesses, `excess`, are added to `FCtmax` using

```
R> FCtmax$threshold <- predict(FC_ald)$location
R> FCtmax$excess <- FCtmax$tmax - FCtmax$threshold
R> is.na(FCtmax$excess[FCtmax$excess <= 0]) <- TRUE
```

It is quite useful to superimpose the threshold estimate on a scatter plot of the data, which is shown in Figure 5 for 2018 and 2019's data, and obtained using

```
R> use <- FCtmax$year %in% c("2018", "2019")
R> plot(FCtmax[use, c("date", "tmax")])
R> lines(FCtmax[use, c("date", "threshold")], col = "red")
```

Having established that the estimated threshold is satisfactory, its excesses are modeled as GPD realizations with

```
R> FC_fmla_gpd <- list(excess ~ s(cyc, bs = "cc", k = 15), ~ 1)
R> FC_gpd <- evgam(FC_fmla_gpd, FCtmax, family = "gpd")
```

which assumes a cyclic form for the scale parameter and a constant shape parameter. Note that setting non-exceedances to `NA` earlier ensured they were ignored by `evgam()`.

It is not reasonable to assume that these excesses of the threshold are independent. Hence to estimate the 100-year return level using F_{ann} for the GPD's nonstationary case, introduced in

Section 2.3, allowance needs to be made for clustering: i.e., an estimate of the extremal index, θ, is needed. The function `extremal()` is used to give an estimate based on the moment-based estimator of Ferro and Segers (2003). This is implemented with

```
R> theta <- extremal(!is.na(FCtmax$excess), FCtmax$date)
```

where `FCtmax$date` is supplied to allow the missing values in `FCtmax$tmax` to be identified. This gives an extremal index estimate of 0.498, corresponding to an average cluster size, defined in terms of grouped threshold exceedances, of 2.01 days.

To estimate the 100-year return level, finite values of the continuous variable `cyc` need to be chosen. We could simply choose `1:365`, but there may be occasions when the numerical estimate is computationally expensive. If the cyclic form is fairly smooth, fewer points can then be used. This is demonstrated here with the use of 50 points instead. A '`data.frame`' of 50 `cyc` values is created using

```
R> rl_df <- data.frame(cyc = seq(0, 365.25, l = 51)[-1])
R> rl_df$threshold <- predict(FC_ald, rl_df, type = "response")$location
R> rl_df[, c("psi", "xi")] <- predict(FC_gpd, rl_df, type = "response")
```

and then `qev()` used to estimate the 100-year return level with

```
R> rl_100_gpd <- qev(0.99, rl_df$threshold, rl_df$psi, rl_df$xi, m = 365.25,
+    theta = theta, family = "gpd", tau = 1 - zeta)
```

which gives a 100-year return level estimate of 39.1°C.

Return level estimates corresponding to monthly maxima can also be obtained with this model. For example, using `rl_df_jan <- data.frame(cyc = 1:31)` instead of `rl_df` above would use 31 `cyc` points, i.e., each day in January, to estimate the 100-January return level.

4.3. Uncertainty estimation

The above Colorado precipitation and Fort Collins temperature examples are used in this section to demonstrate the various options for uncertainty estimation available with **evgam**.

Standard errors for EVD parameters

First consider uncertainty estimates for parameters of an EVD. The GEV model of Section 4.1 will be used for demonstration. The key function here is `predict()` using argument `se.fit = TRUE`. Standard error estimates for GEV parameters estimated for each row of `COprcp_plot` using `m_gev` can be obtained with

```
R> gev_pred <- predict(m_gev, COprcp_plot, type = "response", se.fit = TRUE)
R> head(gev_pred$se.fit)
```

```
  location    scale      shape
1 1.994659 1.040021 0.01565573
2 1.959301 1.039746 0.01565573
3 1.926856 1.039144 0.01565573
```

```
4 1.890763 1.038210 0.01565573
5 1.855742 1.036937 0.01565573
6 1.815885 1.035324 0.01565573
```

which shows just the standard error estimates, stored as `se.fit`.

Standard errors for return levels

Uncertainty estimates for return levels can also be produced. These rely on the Delta method and are achieved with

```
R> gev_rl100_pred <- predict(m_gev, COprcp_plot, prob = c(0.95, 0.99),
+    se.fit = TRUE)
R> head(gev_rl100_pred$se.fit)

    q:0.95   q:0.99
1 4.517385 6.742178
2 4.495358 6.726219
3 4.473734 6.709576
4 4.449786 6.690467
5 4.425445 6.670132
6 4.398455 6.647173
```

which shows standard error estimates for the 0.95 and 0.99 quantiles of the GEV distribution.

Simulation of EVD parameters and return levels

Sampling distributions of EVD parameters or return levels can be skewed. Standard errors will not capture this. `simulate()` can generate samples of parameters or return levels. `nsim =` 5 samples for each GEV parameter from the model of Section 4.1 for each row of `COprcp_plot` are generated using

```
R> gev_sim <- simulate(m_gev, nsim = 5, newdata = COprcp_plot,
+    type = "response")
R> lapply(gev_sim, head, n = 5)

$location
      [,1]      [,2]     [,3]     [,4]     [,5]
1 12.51588  9.381315 14.20692 17.63984 13.55366
2 12.81884  9.700676 14.44408 17.92113 13.70323
3 13.11570 10.005825 14.67694 18.17498 13.87273
4 13.41513 10.340238 14.91654 18.42055 14.02173
5 13.71189 10.674159 15.15552 18.64599 14.17825
6 14.00757 11.049978 15.40300 18.84150 14.31041

$scale
      [,1]     [,2]     [,3]     [,4]     [,5]
1 6.478315 4.473480 5.475476 5.043488 6.412343
```

```
2 6.564390 4.554520 5.570748 5.133926 6.490269
3 6.652005 4.637652 5.668431 5.226687 6.569083
4 6.741185 4.722934 5.768596 5.321840 6.648783
5 6.831950 4.810422 5.871313 5.419447 6.729363
6 6.924318 4.900168 5.976646 5.519570 6.810815

$shape
        [,1]       [,2]      [,3]       [,4]       [,5]
1 0.07837903 0.06982153 0.1002618 0.08797334 0.06268998
2 0.07837903 0.06982153 0.1002618 0.08797334 0.06268998
3 0.07837903 0.06982153 0.1002618 0.08797334 0.06268998
4 0.07837903 0.06982153 0.1002618 0.08797334 0.06268998
5 0.07837903 0.06982153 0.1002618 0.08797334 0.06268998
6 0.07837903 0.06982153 0.1002618 0.08797334 0.06268998
```

Supplying argument `prob` gives simulations that represent EVD quantiles. The above can be modified to give `nsim = 5` samples from the 100-year return level's sampling distribution for each row of `COprcp_plot` with

```
R> gev_rl_sim <- simulate(m_gev, nsim = 5, newdata = COprcp_plot,
+    prob = 0.99)
R> head(gev_rl_sim)

      [,1]     [,2]     [,3]     [,4]     [,5]
1 32.63712 44.06802 39.26888 34.90557 32.47945
2 33.23410 44.80969 39.87423 35.51968 33.02982
3 33.85811 45.57303 40.50001 36.13713 33.60609
4 34.48200 46.34947 41.14598 36.78463 34.16951
5 35.12506 47.14500 41.81274 37.44435 34.74744
6 35.76651 47.96397 42.50965 38.14608 35.30630
```

Suppose that a 95% confidence interval for the 100-year return level for the third station, Boulder, in `COprcp_meta` is sought. This can be approximately achieved by estimating quantiles of the sampling distribution of the 100-year return level estimate. A 10,000-member sample can be drawn from this distribution with

```
R> gev_rl_boulder_sim <- simulate(m_gev, nsim = 1e4,
+    newdata = COprcp_meta[3, ], prob = 0.99)
```

and then its 2.5th and 97.5th empirical percentiles used to form an approximate 95% confidence interval using

```
R> quantile(gev_rl_boulder_sim, c(0.025, 0.975))

     2.5%     97.5%
 97.53343 116.49043
```

This could also have been achieved with `predict()` using `se.fit = TRUE` if a symmetric sampling distribution was a fair assumption.

Simulations of numerically-estimated return levels

Approximate confidence intervals can also be obtained for numerically-estimated return levels. This is demonstrated for the example of Section 4.2, which uses Equation 4. First, parameters are simulated from the ALD and GPD models for each row in `rl_df`, introduced in Section 4.2, using

```
R> FC_sim_ald <- simulate(FC_ald, newdata = rl_df, nsim = 1e3,
+    type = "response")
R> FC_sim_gpd <- simulate(FC_gpd, newdata = rl_df, nsim = 1e3,
+    type = "response")
```

which gives 1000 samples. Then the 100-year return level is calculated for each sample using

```
R> rl_sim <- qev(0.99, FC_sim_ald[[1]], FC_sim_gpd[[1]], FC_sim_gpd[[2]],
+    m = 365.25, theta = theta, family = "gpd", tau = 1 - zeta)
```

Again, the 2.5th and 97.5th percentiles estimated from the return level sample can be used to form an approximate 95% confidence interval using

```
R> quantile(rl_sim, c(0.025, 0.975))

    2.5%    97.5%
38.64717 39.77747
```

Note that 39.37°C, the estimate obtained earlier from fitting separate GEV distributions to monthly maxima, falls well within this interval. Note also that uncertainty in the extremal index estimate, `theta`, calculated in Section 4.2, is not propagated here.

5. Summary and discussion

The R package **evgam** has been developed to allow the fitting of various EVDs with parameters of GAM form. Such forms are an intuitive and robust way of allowing parameters to vary with covariates. Examples in which parameters vary over space, through two-dimensional thin plate plates or the tensor product of two one-dimensional splines, and with time, specifically over the course of a year such that continuity is imposed from year to year, have been given. Examples also demonstrate fitting GEVs and GPDs, the Poisson-GPD model for extremes, and use of the ALD for threshold estimation through quantile regression. Various options for prediction and uncertainty estimation relevant to extreme value analyses have also been presented. Further functionality is planned for package **evgam**.

Computational details

The results in this paper were obtained using R 4.1.3 with the **evgam** 1.0.0 package. In particular, simulation results may differ with linear algebra package used; the reported results are based on using **OpenBLAS** 0.3.8. R itself and **evgam** are available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/.

Acknowledgments

I thank an Editor, Reviewer and Yousra El Bachir for comments that have brought improvements to this article and **evgam**, and Simon Brown, Steven Chan and Rob Shooter for highlighting bugs and/or functionality improvements that have improved **evgam**.

References

Belzile L, Wadsworth JL, Northrop PJ, Grimshaw SD, Huser R (2020). ***mev**: Multivariate Extreme Value Distributions.* R package version 1.13.1, URL https://CRAN.R-project.org/package=mev.

Chavez-Demoulin V, Davison AC (2005). "Generalized Additive Modelling of Sample Extremes." *Journal of the Royal Statistical Society C*, **54**(1), 207–222. doi:10.1111/j.1467-9876.2005.00479.x.

Coles SG (2001). *An Introduction to Statistical Modeling of Extreme Values.* Springer-Verlag. doi:10.1007/978-1-4471-3675-0.

Cooley D, Nychka D, Naveau P (2007). "Bayesian Spatial Modeling of Extreme Precipitation Return Levels." *Journal of the American Statistical Association*, **102**(479), 824–840. doi:10.1198/016214506000000780.

Davison AC, Ramesh NI (2000). "Local Likelihood Smoothing of Sample Extremes." *Journal of the Royal Statistical Society B*, **62**(1), 191–208. doi:10.1111/1467-9868.00228.

Davison AC, Smith RL (1990). "Models for Exceedances Over High Thresholds." *Journal of the Royal Statistical Society B*, **52**(3), 393–425. doi:10.1111/j.2517-6161.1990.tb01796.x.

De Boor C (1978). *A Practical Guide to Splines.* Springer-Verlag.

Dutang C (2022). *CRAN Task View: Extreme Value Analysis.* Version 2022-03-23, URL https://CRAN.R-project.org/view=ExtremeValue.

Fasiolo M, Wood SN, Zaffran M, Nedellec R, Goude Y (2021). "Fast Calibrated Additive Quantile Regression." *Journal of the American Statistical Association*, **116**(535), 1402–1412. doi:10.1080/01621459.2020.1725521.

Ferro CAT, Segers J (2003). "Inference for Clusters of Extreme Values." *Journal of the Royal Statistical Society B*, **65**(2), 545–556. doi:10.1111/1467-9868.00401.

Gilleland E, Katz RW (2016). "**extRemes** 2.0: An Extreme Value Analysis Package in R." *Journal of Statistical Software*, **72**(8), 1–39. doi:10.18637/jss.v072.i08.

Gilleland E, Ribatet M, Stephenson AG (2013). "A Software Review for Extreme Value Analysis." *Extremes*, **16**(1), 103–119. doi:10.1007/s10687-012-0155-0.

Green PJ, Silverman BW (1994). *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*. Chapman &Hall/CRC. doi:10.1201/b15710.

Hall P, Tajvidi N (2000). "Nonparametric Analysis of Temporal Trend When Fitting Parametric Models to Extreme-Value Data." *Statistical Science*, **15**(2), 153–167. doi:10.1214/ss/1009212755.

Heffernan JE, Stephenson AG (2018). ***ismev**: An Introduction to Statistical Modeling of Extreme Values*. R package version 1.42, URL https://CRAN.R-project.org/package=ismev.

Koenker R (2005). *Quantile Regression*. Cambridge University Press. doi:10.1017/cbo9780511754098.

Koenker R (2021). ***quantreg**: Quantile Regression*. R package version 5.88, URL https://CRAN.R-project.org/package=quantreg.

Northrop PJ, Jonathan P (2011). "Threshold Modelling of Spatially Dependent Non-Stationary Extremes with Application to Hurricane-Induced Wave Heights." *Environmetrics*, **22**(7), 799–809. doi:10.1002/env.1106.

Oh HS, Lee TCM, Nychka DW (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.

Pauli F, Coles SG (2001). "Penalized Likelihood Inference in Extreme Value Analyses." *Journal of Applied Statistics*, **28**(5), 547–560. doi:10.1080/02664760120047889.

Pfaff B, McNeil A (2018). ***evir**: Extreme Values in R*. R package version 1.7-4, URL https://CRAN.R-project.org/package=evir.

Pickands J (1971). "The Two-Dimensional Poisson Process and Extremal Processes." *Journal of Applied Probability*, **8**(4), 745–756. doi:10.2307/3212238.

Ramesh NI, Davison AC (2002). "Local Models for Exploratory Analysis of Hydrological Extremes." *Journal of Hydrology*, **256**(1–2), 106–119. doi:10.1016/s0022-1694(01)00522-4.

Randell D, Turnbull K, Ewans K, Jonathan P (2016). "Bayesian Inference for Nonstationary Marginal Extremes." *Environmetrics*, **27**(7), 439–450. doi:10.1002/env.2403.

R Core Team (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Ribatet M (2020). ***SpatialExtremes**: Modelling Spatial Extremes*. R package version 2.0-9, URL https://CRAN.R-project.org/package=SpatialExtremes.

Rigby RA, Stasinopoulos DM (2005). "Generalized Additive Models for Location, Scale and Shape." *Journal of the Royal Statistical Society C*, **54**(3), 507–554. doi:10.1111/j.1467-9876.2005.00510.x.

Rue H, Martino S, Chopin N (2009). "Approximate Bayesian Inference for Latent Gaussian Models by Using Integrated Nested Laplace Approximations." *Journal of the Royal Statistical Society B*, **71**(2), 319–392. doi:10.1111/j.1467-9868.2008.00700.x.

Smith RL (1986). "Extreme Value Theory Based on the r Largest Annual Events." *Journal of Hydrology*, **86**(1–2), 27–43. doi:10.1016/0022-1694(86)90004-1.

Smith RL (1989). "Extreme Value Analysis of Environmental Time Series: An Application to Trend Detection in Ground-Level Ozone." *Statistical Science*, **4**(4), 367–377. doi:10.1214/ss/1177012400.

Stasinopoulos DM, Rigby RA (2007). "Generalized Additive Models for Location Scale and Shape (GAMLSS) in R." *Journal of Statistical Software*, **23**(7), 1–46. doi:10.18637/jss.v023.i07.

Stephenson AG (2002). "**evd**: Extreme Value Distributions." *R News*, **2**(2), 31–32. URL https://CRAN.R-project.org/doc/Rnews/Rnews_2002-2.pdf.

Wood SN (2006). "Low-Rank Scale-Invariant Tensor Product Smooths for Generalized Additive Mixed Models." *Biometrics*, **62**(4), 1025–1036. doi:10.1111/j.1541-0420.2006.00574.x.

Wood SN (2011). "Fast Stable Restricted Maximum Likelihood and Marginal Likelihood Estimation of Semiparametric Generalized Linear Models." *Journal of the Royal Statistical Society B*, **73**(1), 3–36. doi:10.1111/j.1467-9868.2010.00749.x.

Wood SN, Pya N, Säfken B (2016). "Smoothing Parameter and Model Selection for General Smooth Models." *Journal of the American Statistical Association*, **111**(516), 1548–1563. doi:10.1080/01621459.2016.1180986.

Yee TW (2010). "The **VGAM** Package for Categorical Data Analysis." *Journal of Statistical Software*, **32**(10), 1–34. doi:10.18637/jss.v032.i10.

Yee TW (2015). *Vector Generalized Linear and Additive Models: With an Implementation in R*. Springer-Verlag, New York. doi:10.1007/978-1-4939-2818-7.

Yee TW, Stephenson AG (2007). "Vector Generalized Linear and Additive Extreme Value Models." *Extremes*, **10**(1–2), 1–19. doi:10.1007/s10687-007-0032-4.

Yee TW, Wild CJ (1996). "Vector Generalized Additive Models." *Journal of the Royal Statistical Society B*, **58**(3), 481–493. doi:10.1111/j.2517-6161.1996.tb02095.x.

Youngman B (2022). ***evgam**: Generalised Additive Extreme Value Models*. R package version 1.0.0, URL https://CRAN.R-project.org/package=evgam.

Youngman BD (2019). "Generalized Additive Models for Exceedances of High Thresholds with an Application to Return Level Estimation for U.S. Wind Gusts." *Journal of the American Statistical Association*, **114**(528), 1865–1879. doi:10.1080/01621459.2018.1529596.

Yu K, Moyeed RA (2001). "Bayesian Quantile Regression." *Statistics & Probability Letters*, **54**(4), 437–447. doi:10.1016/s0167-7152(01)00124-9.

Affiliation:

Benjamin D. Youngman
Department of Mathematics
University of Exeter
Laver Building, North Park Road
Exeter, EX4 4QE, United Kingdom
E-mail: b.youngman@exeter.ac.uk
URL: http://ex.ac.uk/youngman

Journal of Statistical Software https://www.jstatsoft.org/
published by the Foundation for Open Access Statistics https://www.foastat.org/
July 2022, Volume 103, Issue 3 *Submitted:* 2020-04-18
doi:10.18637/jss.v103.i03 *Accepted:* 2021-12-04