extRemes 2.0: An Extreme Value Analysis Package in R

Eric Gilleland
National Center for
Atmospheric Research

Richard W. Katz
National Center for
Atmospheric Research

Abstract

This article describes the extreme value analysis (EVA) R package **extRemes** version 2.0, which is completely redesigned from previous versions. The functions primarily provide utilities for implementing univariate EVA, with a focus on weather and climate applications, including the incorporation of covariates, as well as some functionality for assessing bivariate tail dependence.

Keywords: extreme value analysis, R.

1. Introduction

In this paper, we describe a new version ($\geq$ 2.0) of the R (R Core Team 2016) package **extRemes** (Gilleland and Katz 2011; Gilleland 2016), which is a major redesign from previous versions. Weather and climate applications are the intended focus of the package, although the functions could be used in other applications. The new version of the package is to be used from the command line and implements various methods from (predominantly univariate) extreme value theory, whereas previous versions provided graphical user interfaces predominantly to the R package **ismev** (Heffernan and Stephenson 2012). Package **ismev** is a companion package to Coles (2001), which was originally written for the S language, ported into R by Alec G. Stephenson, and currently is maintained by Eric Gilleland.

A relatively recent review of extreme value analysis (EVA) software can be found in Gilleland, Ribatet, and Stephenson (2013) where thirteen R packages were addressed. The new version of the package attempts to fill some of the missing gaps in these packages from univariate EVA, although some additional multivariate capacity may be added in the future. For convenience sake, we adopt the notation of Coles (2001) as much as possible.

The first version of **extRemes** was solely a graphical user interface (GUI) to **ismev** designed to shorten the learning curve of EVA, particularly the handling of non-stationarity, for the

weather and climate science communities. Over the years, **extRemes** garnered a large user community, which also resulted in considerable amounts of feedback and questions pertaining to EVA. Because of the importance of Coles (2001) on EVA, for which **ismev** was written, it was decided to keep this package as close as possible to the original so that it can continue to be used along with this text. Therefore, **extRemes** itself has been re-designed, without any GUI interfaces, to better follow an object-oriented structure, making many functions much more user-friendly, as well as a including formula approach to the incorporation of covariates into parameter estimates. In order to maintain the popular GUI-based functions, a new companion package, **in2extRemes** (Gilleland 2015), provides point-and-click GUI windows for some of the main functions in **extRemes**.[1]

Other main EVA packages in R include: **evd** (Stephenson 2002), **evdbayes** (Stephenson and Ribatet 2014), **evir** (Pfaff and McNeil 2012), **fExtremes** (Diethelm Wuertz and many others 2013), **lmom** (Hosking 2014), **SpatialExtremes** (Ribatet 2009; Ribatet, Singleton, and R Core Team 2013) and **texmex** (Southworth and Heffernan 2011). Briefly, summarizing from Gilleland *et al.* (2013), among these packages, **extRemes**, **ismev**, **SpatialExtremes**, and **texmex** are the only ones with substantial ability to account for non-stationary model fitting in the EVA context. The package **texmex** allows for maximum likelihood estimation (MLE), penalized MLE and Bayesian estimation, and primarily contains functionality for fitting the Heffernan and Tawn (2004) conditional extremes models for multivariate analysis, but also includes some univariate capabilities. **SpatialExtremes** includes MLE, maximum composite likelihood and Bayesian estimation. The **ismev** package includes only MLE, whereas version 2 of **extRemes** contains MLE, generalized MLE (Martins and Stedinger 2000, 2001), L-moments (stationary case only), and some functionality for Bayesian estimation. The packages **evd** and **evdbayes** have some functionality for non-stationary estimation, but the main emphasis is on bivariate EVA for the former and (univariate) Bayesian estimation for the latter. Some of the strengths of **extRemes** include: systematic, thorough treatment of univariate extremes (understood to include covariates), an extensive tutorial with use of real/realistic weather and climate data sets, as well as a large number of regular users who provide considerable feedback for continuing improvement, as well as continued financial support through the Weather and Climate Impacts Assessment Science Program (http://www.assessment.ucar.edu/) that allows for ongoing package maintenance and user assistance. Package **extRemes** is available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/package=extRemes.

2. Extreme value analysis background

Suppose we want to win the jackpot in the power ball game of the Colorado Lottery in one single draw, which has a reported odds of winning of 1 in 176 million (http://www.coloradolottery.com/GAMES/POWERBALL/). Recognizing the small chance of winning on one single draw ($P\{$winning the jackpot on one draw$\} \approx 5.7 \times 10^{-9}$), suppose we decide to play the game once every day for a long period of time with the desire of winning the jackpot at least one time.[2]

---

[1]A separate user guide for **in2extRemes** is in preparation, and should be released soon as an NCAR technical note along with a web-based version.

[2]To keep the example simple, we focus solely on winning the jackpot without considering the possibility of winning smaller, if still substantial, prizes.

Simon Denis Poisson (1781–1840) introduced the probability distribution, subsequently named after him, as the limit of the binomial distribution when the probability of success tends to zero at a rate that is fast enough that the expected number of events is constant (Stigler 1986). Let N denote the number of times the grand prize is won over n days. Then N has an approximate Poisson distribution with intensity parameter $n\lambda$. So,

$$\mathsf{P}\{N = 0\} = e^{-n\lambda} \text{ and } \mathsf{P}\{N > 0\} = 1 - e^{-n\lambda}. \tag{1}$$

We can, therefore, use the Poisson distribution to approximate such probabilities. For example, the probability of winning on at least one single draw after ten years of playing every day of the year (the Poisson intensity, here, is $n\lambda \approx (5.7 \times 10^{-9}) \cdot (10 \text{ years}) \cdot (365.25 \text{ days per year})$) yields a considerably low probability of success ($\approx 2.08 \times 10^{-5}$), and after one-hundred years, the probability is still very small at about 0.0002. After one-thousand years, the probability remains very low at about 0.002. The waiting time before winning the jackpot has an approximate exponential distribution with mean $\approx 1/((5.7 \times 10^{-9}) \cdot (365.25))$, which is more than 480 thousand years.

While we do not require events to be as rare as in the example above, statistical EVA refers to events with a low probability of occurrence. The Poisson distribution informs about the frequency of occurrence of a rare event, but what about the magnitude? If we have independent and identically distributed random variables $X_1, \ldots, X_n$ that follow a distribution function (df), F (sometimes called the "parent" df), and we want to approximate the df for the maximum, $M_n = \max\{X_1, \ldots, X_n\}$, then we have that $\mathsf{P}\{M_n \le z\} = F^n(z) \longrightarrow 0$ as $n \longrightarrow \infty$; unless F has a finite upper limit x_U and $z \ge x_U$. Furthermore, in practice, F must be estimated, and small errors in the estimation of F often lead to large errors in the estimation of F^n.

Analogous to the Central Limit Theorem that provides theoretical support for the normal df when approximating the distribution of averages, it is possible to stabilize the distribution of many random variables so that asymptotically their extreme values follow specific df's.

There are two primary approaches to analyzing extremes of a dataset. The first approach reduces the data considerably by taking maxima of long blocks of data, e.g., annual maxima. The second approach is to analyze excesses over a high threshold. Both approaches can be characterized in terms of a Poisson process, which allows for simultaneously fitting of parameters concerning both the frequency and intensity of extreme events.

The generalized extreme value (GEV) df has theoretical justification for fitting to block maxima of data and the generalized Pareto (GP) df has similar justification for fitting to excesses over a high threshold. The GEV df is given by

$$G(z) = \exp\left[-\left\{1 + \xi\left(\frac{z-\mu}{\sigma}\right)\right\}_+^{-1/\xi}\right], \tag{2}$$

where $y_+ = \max\{y, 0\}$, $\sigma > 0$ and $-\infty < \mu, \xi < \infty$ (cf. Coles 2001). Equation 2 envelops three types of df's depending on the sign of the shape parameter, ξ. The heavy-tailed Fréchet df results from $\xi > 0$, and the upper bounded Weibull df (sometimes referred to as the reverse or reflected Weibull df) when $\xi < 0$. The Gumbel type is obtained by taking the limit as $\xi \longrightarrow 0$ giving

$$G(z) = \exp\left[-\exp\left\{-\left(\frac{z-\mu}{\sigma}\right)\right\}\right], \quad -\infty < z < \infty. \tag{3}$$

The quantiles of the GEV df are of particular interest because of their interpretation as return levels; the value expected to be exceeded on average once every $1/p$ periods, where $1 - p$ is the specific probability associated with the quantile. We seek z_p such that $G(z_p) = 1 - p$, where G is as in Equation 2. Letting $y_p = -1/\ln(1 - p)$, then the associated return level z_p is

$$z_p = \begin{cases} \mu + \frac{\sigma}{\xi} \left[y_p^\xi - 1 \right], & \text{for } \xi \neq 0, \\ \mu + \sigma \ln y_p, & \text{for } \xi = 0. \end{cases} \tag{4}$$

If z_p is plotted against $\ln y_p$, then the type of distribution determines the curvature of the resulting line: for $\xi = 0$ (Gumbel) the plot is a straight line, for $\xi > 0$ (Fréchet) the plot is concave with no finite upper bound, and for $\xi < 0$ (Weibull) the curve is convex with an asymptotic upper limit as $p \longrightarrow 0$ at its upper bound, $\mu - \sigma/\xi$.

The GP df is given by

$$H(x) = 1 - \left[1 + \xi \left(\frac{x - u}{\sigma_u} \right) \right]_+^{-1/\xi}, \tag{5}$$

where u is a high threshold, $x > u$, scale parameter $\sigma_u > 0$ (depending on the threshold u), and shape parameter $-\infty < \xi < \infty$. Again, ξ determines three types of df's (with the same interpretation as for the GEV df): heavy tail when $\xi > 0$ (Pareto), upper bound when $\xi < 0$ (Beta) and exponential in the limit as $\xi \longrightarrow 0$, which results in[3]

$$H(x) = 1 - e^{-(x-u)/\sigma}. \tag{6}$$

The GP df is an approximation of the upper tail of a parent df. If G is the limiting df of standardized maxima, a GEV df with location, scale and shape parameters μ, σ and ξ, respectively, then the GP df has the same shape parameter ξ and scale parameter $\sigma_u = \sigma + \xi(u - \mu)$. The quantiles of the GP df are easily obtained by setting Equation 5 equal to $1 - p$ and inverting. However, the quantiles of the GP df cannot be as readily interpreted as return levels because the data no longer derive from specific blocks of equal length. Instead, an estimate of the probability of exceeding the threshold, e.g., called ζ_u, is required. Then, the value x_m that is exceeded on average once every m observations (i.e., an estimated return level) is

$$x_m = \begin{cases} u + \frac{\sigma_u}{\xi} \left[(m\zeta_u)^\xi - 1 \right], & \xi \neq 0, \\ u + \sigma_u \ln(m\zeta_u), & \xi = 0. \end{cases} \tag{7}$$

2.1. Non-stationarity

If it is believed that the extremes of the data are not stationary, then it is possible to incorporate this information into the parameters. For example,

$$\mu(t) = \mu_0 + \mu_1 t + \mu_2 t^2,$$

$$\sigma(t) = \sigma_0 + \sigma_1 t,$$

$$\xi(t) = \begin{cases} \xi_0, & t \leq t_0, \\ \xi_1, & t > t_0. \end{cases}$$

---

[3]In this case, the scale parameter does not depend on u.

In the above models, each parameter varies with time; the location parameter follows a quadratic formula of time, the scale parameter a linear function, and the shape varies between two time periods. Because the scale parameter must be positive everywhere, it is often also modeled using a log link function as given by

$$\ln \sigma(t) = \phi_0 + \phi_1 t.$$

2.2. Point process approach

Another way to model extreme values is through a point process (PP) characterization, which allows for modeling both the frequency of exceeding a high threshold and the values of the excesses above it. This approach was first developed probabilistically by Leadbetter, Lindgren, and Rootzén (1983) and Resnick (1987), and as a statistical technique by Smith (1989) and Davison and Smith (1990). Briefly, if a process is stationary and satisfies an asymptotic lack of "clustering" condition for values that exceed a high threshold, then its limiting form is non-homogeneous Poisson with intensity measure, Λ, on a set of the form $A = (t_1, t_2) \times (x, \infty)$ given by

$$\Lambda(A) = (t_2 - t_1) \cdot \left[1 + \xi \frac{x - \mu}{\sigma} \right]_+^{-1/\xi},$$

where the parameters μ, σ and ξ are interpreted exactly as in Equation 2. In fact, if the time scale is measured in years, then Equation 2 gives the probability that the set $(t_1, t_1+1) \times (x, \infty)$ is empty, which is equivalent to the probability that the annual maximum is less than or equal to x. In other words, the intensity measure needs to be multiplied by n_y (the number of years of data) to expand the time interval from $(0, 1)$ to a yearly time scale (as in Equation 7.8 in Coles 2001).

2.3. Parameter estimation

There are several common methods for estimating the parameters of the GEV and GP df's, as well as of the PP. Version 2.0 of package **extRemes** provides four possibilities: (i) maximum likelihood estimation (MLE), (ii) generalized MLE (GMLE, Martins and Stedinger 2000, 2001), (iii) Bayesian estimation and (iv) L-moments estimation (only without parameter covariates).

For methods (i) to (iii), a likelihood function is needed. Because no analytic solution to the optimization problem exists, the MLE solution is found via numerical routines. Defining $y_+ = \max\{y, 0\}$ and letting $z_1, \ldots, z_m$ represent the block maxima of some sample consisting of m blocks of length n, the log-likelihood for the GEV df is

$$\ell(\mu, \sigma, \xi; z_1, \ldots, z_m) =$$
$$- m \ln \sigma - (1 + 1/\xi) \sum_{i=1}^{m} \ln \left[1 + \xi \left(\frac{z_i - \mu}{\sigma} \right) \right]_+ - \sum_{i=1}^{m} \left[1 + \xi \left(\frac{z_i - \mu}{\sigma} \right) \right]_+^{-1/\xi}. \quad (8)$$

Because the shape parameter of the Gumbel df (i.e., $\xi = 0$) represents a single point in a continuous parameter space (see Equation 2), there is zero probability of obtaining an estimated shape parameter that is exactly zero. It is possible to fit the Gumbel df separately and test the hypothesis of $\xi = 0$ versus $\xi \neq 0$ (or, equivalently, determine if zero falls within

the confidence limits for the shape parameter), but it is arguably preferable to always allow the shape parameter to be nonzero even if such a test supports the Gumbel hypothesis.[4] This can be performed using the likelihood-ratio test (Coles 2001) or by checking whether or not zero falls within the confidence intervals for ξ. The latter approach does not require fitting the Gumbel df to the data. The log-likelihood for the Gumbel type is

$$\ell(\mu, \sigma; z_1, \ldots, z_m) = -m \ln \sigma - \sum_{i=1}^{m} \left(\frac{z_i - \mu}{\sigma}\right) - \sum_{i=1}^{m} \exp\left\{-\left(\frac{z_i - \mu}{\sigma}\right)\right\}. \tag{9}$$

Setting $y_i, \ldots, y_k$ to be only those values of $x_1, \ldots, x_n$ that exceed the threshold, the GP log-likelihood is given by

$$\ell(\sigma_u, \xi; y_1, \ldots, y_k) = -k \ln \sigma_u - (1 + 1/\xi) \sum_{i=1}^{k} \ln \left\{\left[1 + \xi\left(\frac{y_i - u}{\sigma_u}\right)\right]_+\right\}, \tag{10}$$

where $y_+ = \max\{y, 0\}$ so that any value y_i with parameter combination such that $1 + \xi(y_i - u)/\sigma_u \leq 0$ would yield a log-likelihood of $-\infty$. Similar to the case for the Gumbel, the shape parameter of the exponential is also a single point in a continuous parameter space, with log-likelihood

$$\ell(\sigma_u; y_1, \ldots, y_k) = -k \ln \sigma - \frac{1}{\sigma} \sum_{i=1}^{k} (y_i - u). \tag{11}$$

Finally, the PP log-likelihood is

$$\ell(\mu, \sigma, \xi; x_1, \ldots, x_n) =$$
$$- k \ln \sigma - (1/\xi + 1) \sum_{i=1}^{n} \left[1 + \frac{\xi}{\sigma}(x_i - u)\right]_{x_i > u} - n_y \left(1 + \frac{\xi}{\sigma}(u - \mu)\right)^{-1/\xi}, \tag{12}$$

where n_y denotes the number of years of data (so that GEV parameters are for annual maxima), and $[\cdot]_{x_i > u}$ denotes the indicator function. Even though the likelihood (12) involves excesses, the parameterization is in terms of the GEV df, not the GP df.

Version ≥ 2.0 of package **extRemes** also allows to determine confidence intervals for parameter estimates and associated return levels. For the L-moments approach, parametric bootstrap intervals are calculated. For Bayesian estimation, the upper and lower quantiles of the posterior probability distribution for the parameters and return levels (possibly after a burn-in period) of the MCMC sample are taken, and for MLE/GMLE several options are available: normal approximation (perhaps via the delta method), profile-likelihood, and parametric bootstrap.

2.4. Dependence

Temporal dependence and declustering.   The above theoretical justifications for using the extreme value (EV) df's is predicated on an i.i.d. assumption for the extremes. When taking block maxima over large blocks, this is usually a reasonable assumption. However, in the case of threshold excesses, it is often violated. Perhaps the simplest approach to handling

---

[4]This recommendation is based on practical considerations as given in Coles, Pericchi, and Sisson (2003) and on penultimate approximations in extreme value theory (Reiss and Thomas 2007).

dependence above the threshold is to decluster the excesses by attempting to identify clusters of excesses and replacing them with a single value, such as their maximum. **extRemes** supports two methods for performing such a procedure (`runs` and `intervals`). To determine if dependence is present, one approach that is available with **extRemes** is to estimate the extremal index, where values less than unity imply some dependence, with stronger dependence as the value decreases (with one interpretation of the extremal index being the limit of the reciprocal of the mean cluster length as the threshold increases). Reiss and Thomas (2007) suggest an auto-tail dependence estimate, along with a plot against increasing lags, analogous to the autocorrelation function plots from time series analysis.

Asymptotic tail dependence. In the case of two variables, it is sometimes of interest to investigate the tail dependence between the variables. **extRemes** includes a function for employing a tail dependence test with the null hypothesis that the two variables are tail dependent. We refer the reader to Reiss and Thomas (2007) for specific details of the statistics and the test, but to summarize their interpretation for the examples herein, we provide some brief details.

Specifically, for two random variables X and Y with df's F and G and a threshold u, define $U = F(X)$ and $V = G(Y)$, which are, by the probability integral transform, uniformly distributed random variables, and consider

$$\chi(u) = \mathsf{P}\left\{Y > G^{-1}(u)|X > F^{-1}(u)\right\} = \mathsf{P}\left\{V > u|U > u\right\},$$

$$\bar{\chi}(u) = 2\frac{\log\left(\mathsf{P}\left\{U > u\right\}\right)}{\log\left(\mathsf{P}\left\{U > u,\, V > u\right\}\right)} - 1.$$

Properties of the statistics include (let $\lim_{u \longrightarrow 1} \chi(u) = \chi$ and $\lim_{u \longrightarrow 1} \bar{\chi}(u) = \bar{\chi}$): (i) $0 \leq \chi(u) \leq 1$ and $-1 \leq \bar{\chi} \leq 1$, (ii) if X and Y are stochastically independent, then $\chi(u) = 1 - u$ and $\bar{\chi}(u) = 0$, (iii) if they are completely dependent, $\bar{\chi} = 1$, and (iv) if $\chi = 0$ and $\bar{\chi} < 1$, then there is tail dependence with $\bar{\chi}$ determining the degree of dependence.

For auto-tail dependence (i.e., a clustering at high levels for a single time series), the definitions for $\chi(u)$ and $\bar{\chi}(u)$ are modified (cf. Sibuya 1960; Coles, Heffernan, and Tawn 1999; Coles 2001; Reiss and Thomas 2007) for a series of identically distributed random variables $X_1, \ldots, X_n$ with common df F and $i \leq n - h$ as

$$\rho(u, h) = \mathsf{P}\{X_{1+h} > F^{-1}(u)|X_1 > F^{-1}(u)\}$$

and

$$\bar{\rho}(u, h) = \frac{2\log\mathsf{P}\{X_1 > F^{-1}(u)\}}{\log\mathsf{P}\{X_1 > F^{-1}(u),\, X_{1+h} > F^{-1}(u)\}}.$$

For more on EVA, see Coles (2001); Katz, Parlange, and Naveau (2002); Beirlant, Goegebeur, Teugels, and Segers (2004); De Haan and Ferreira (2006); Reiss and Thomas (2007).

3. Illustration: extRemes

Version 2.0 of **extRemes** depends on **Lmoments** (Karvanen 2006, 2011), **car** (Fox and Weisberg 2011), and **distillery** (Gilleland 2013). The primary function of **extRemes** version 2.0 is
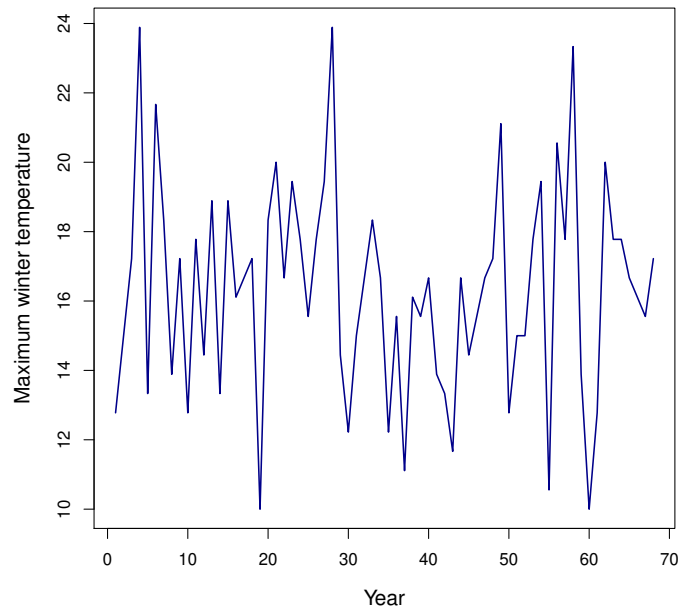
Figure 1: Annual maximum winter temperature at Port Jervis, New York (Thompson and Wallace 1998; Wettstein and Mearns 2002).

called `fevd`. A run through the example in its help file will shed much light on the functionality of the package. Several method functions operate on objects fitted by `fevd` including: `plot` (allows for making probability-probability, quantile-quantile and return level plots, histograms, likelihood (and gradient likelihood) traces (MLE/GMLE), as well as MCMC trace plots (Bayesian)), `print`, `ci` (for calculating confidence intervals for parameters and return levels), `distill` (strips out only the parameter estimates, their standard errors and possibly also the (negative) log-likelihood, AIC and BIC as a named vector), `return.level` (estimates return levels from the fitted object possibly with confidence intervals), `datagrabber` (strips the data component from the resulting fitted list object in order to access the original data), and `predict`.

In addition to fitting the extreme value df's (EVD's) to data, easy to use functions exist for finding their probabilities (`pevd`), densities (`devd`), likelihood for a specific data sample (`levd`), and drawing random samples from EVD's (`revd`). There are also similar functions that work off of an object fitted by `fevd`: `pextRemes` and `rextRemes`.

3.1. Block maxima approach

To demonstrate fitting the GEV df to block maxima data, we will use maximum winter temperature data from 1927 to 1995 at Port Jervis, New York (Thompson and Wallace 1998; Wettstein and Mearns 2002). A time series of the data is shown in Figure 1. Note that `PORTw` is a `data.frame` object with a column named `TMX1` that gives the annual winter maximum temperature.

```
R> data("PORTw", package = "extRemes")
R> plot(PORTw$TMX1, type = "l", col = "darkblue", lwd = 1.5, cex.lab = 1.25,
+     xlab = "Year", ylab = "Maximum winter temperature")
```

	Location	Scale	Shape
Estimates	15.14	2.97	−0.22
Std. err.	0.40	0.28	0.07
95% normal approx.	(14.36, 15.92)	(2.43, 3.51)	(−0.36, −0.07)
confidence intervals			
for parameter estimates			
Covariance of parameter estimates			
location	0.16	0.01	−0.01
scale		0.08	−0.01
shape			0.01
Negative log-likelihood	172.74		
AIC	351.49		
BIC	358.14		
Estimated return levels	(95% confidence interval)		
(degrees centigrade)			
2-year	16.19 (15.38, 16.99)		
20-year	21.65 (20.40, 22.89)		
100-year	23.79 (21.79, 25.78)		

Table 1: Estimated GEV df parameters and return levels fitted to the annual winter maximum temperature (degrees centigrade) at Port Jervis, New York data in Figure 1.

```
R> fit1 <- fevd(TMX1, PORTw, units = "deg C")
R> distill(fit1)
```

Figure 2 shows standard diagnostic plots, and Figure 3 shows the likelihood (top row) and likelihood gradients (bottom row) in a neighborhood of the maximum value attained for each parameter (holding the other parameters fixed at their maximum likelihood estimates). In each case, the assumptions for fitting the GEV to the data appear reasonable. The figures are obtained using the following code.

```
R> plot(fit1)
R> plot(fit1, "trace")
R> return.level(fit1)
R> return.level(fit1, do.ci = TRUE)
R> ci(fit1, return.period = c(2, 20, 100))
```

The final three lines of code give estimated return levels, the last two lines give them along with their (normal approximation) confidence intervals (bottom of Table 1). Note that the two-year return level corresponds to the median of the GEV df for the fit.

Given the large negative estimate for the shape parameter (upper bounded case), it is unlikely that the Gumbel hypothesis would be accepted, but there are two ways we can test the hypothesis using **extRemes** version $\geq$ 2.0. The easiest is to use the following line of code, which gives confidence intervals for each parameter estimate (default used below gives 95% confidence intervals), which reveals that we reject the null hypothesis of a light-tailed df in favor of the upper bounded type.
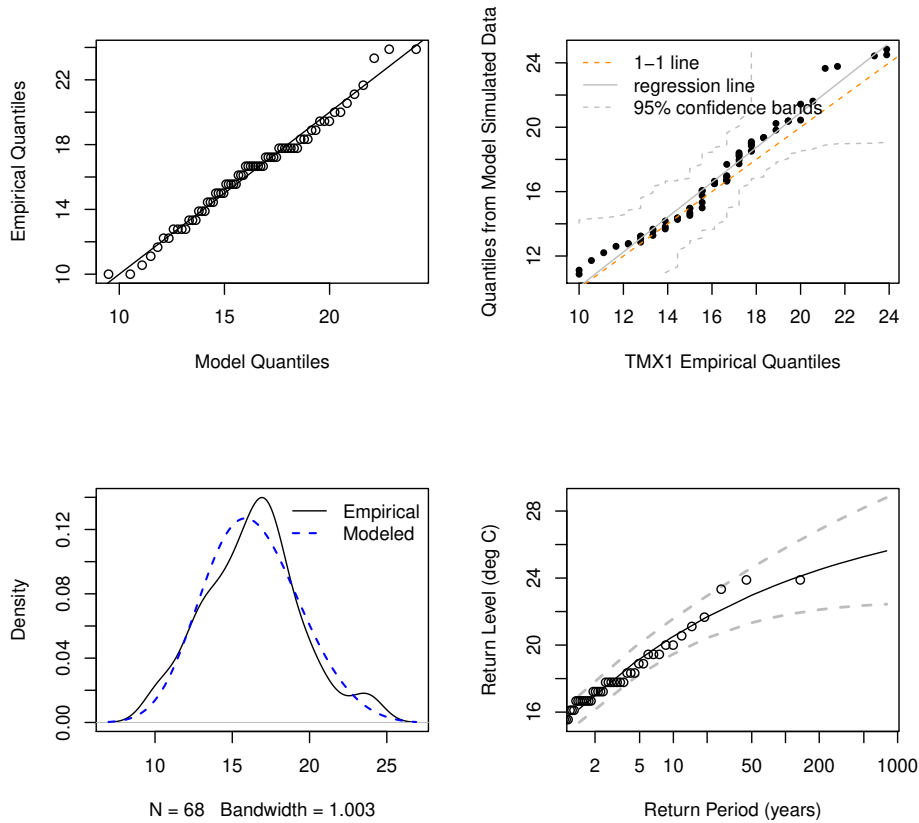
```
R> ci(fit1, type = "parameter")
```

Figure 2: Diagnostics from the GEV df fitted to the Port Jervis maximum winter temperature data shown in Figure 1. Quantile-quantile plot (top left), quantiles from a sample drawn from the fitted GEV df against the empirical data quantiles with 95% confidence bands (top right), density plots of empirical data and fitted GEV df (bottom left), and return level plot with 95% pointwise normal approximation confidence intervals (bottom right). Note that the original function call is attached as the main label.

Alternatively, we can perform the likelihood-ratio test for which we will need to fit the Gumbel df to the same set of data.

```
R> fit0 <- fevd(TMX1, PORTw, type = "Gumbel", units = "deg C")
R> plot(fit0)
R> plot(fit0, "trace")
```

Next, use the `lr.test` function to run the test. This function can be used either on a fitted 'fevd' object or two likelihood values can be passed.

```
R> lr.test(fit0, fit1)
```

Again, the results of the above command provide support for rejecting the null hypothesis of the Gumbel df (p value ≈ 0.01374). Of course, for the shape parameter and longer return
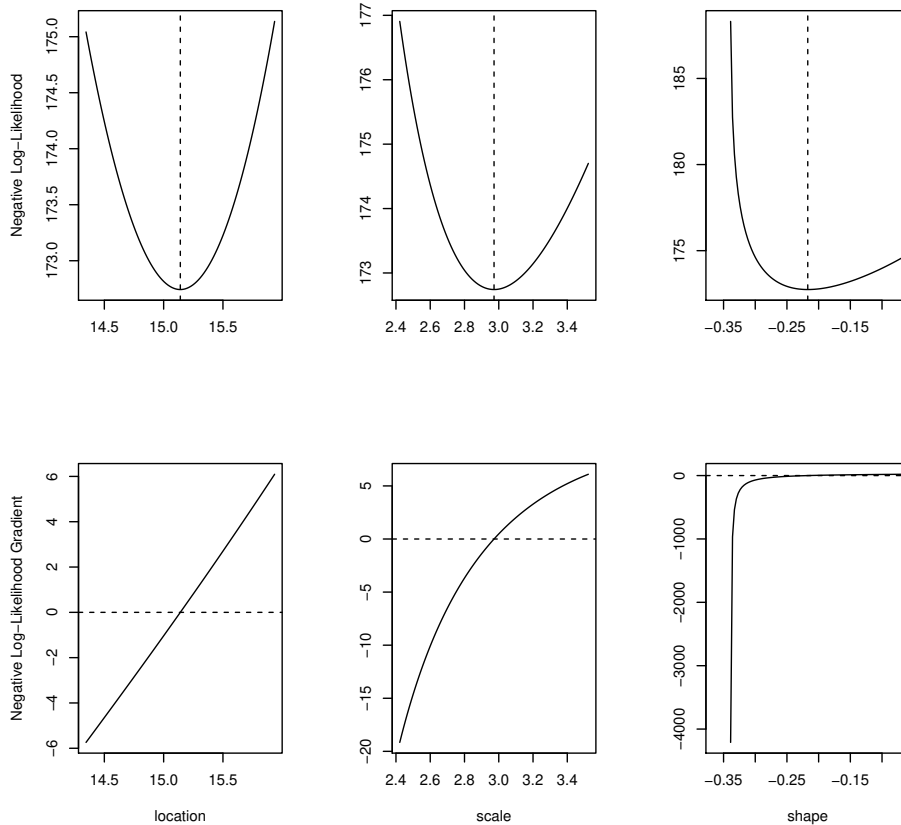
Figure 3: Plots of the likelihood and its gradient for each parameter (holding the other parameters fixed at their maximum-likelihood estimates.

levels, it is often better to obtain confidence intervals from the profile-likelihood method in order to allow for skewed intervals that may better match the sampling df of these parameters (cf. Figure 4). This can be performed in **extRemes** version ≥ 2.0 with the following commands.

```
R> ci(fit1, type = "parameter", which.par = 3, xrange = c(-0.4, 0.01),
+    nint = 100, method = "proflik", verbose = TRUE)
R> ci(fit1, method = "proflik", xrange = c(22, 28), verbose = TRUE)
```

From the first lines above, we get an improved 95% confidence interval for the shape parameter of about $(-0.34, -0.05)$; very similar to the normal approximation limits (Table 1). And from the last lines, we get that approximate 95% confidence interval for the 100-year return level based on the profile-likelihood is about $(22.43, 27.17)$, which is also similar in this case to the normal approximation limits, though shifted a couple of degrees toward the higher end.

When using the profile-likelihood method, it is necessary to specify a range (although the code will try to find reasonable default values) over which to seek out the confidence intervals. Note also that the above commands produce a plot of the profile-likelihood. It is generally important to inspect the plot to be sure that the resulting confidence intervals are

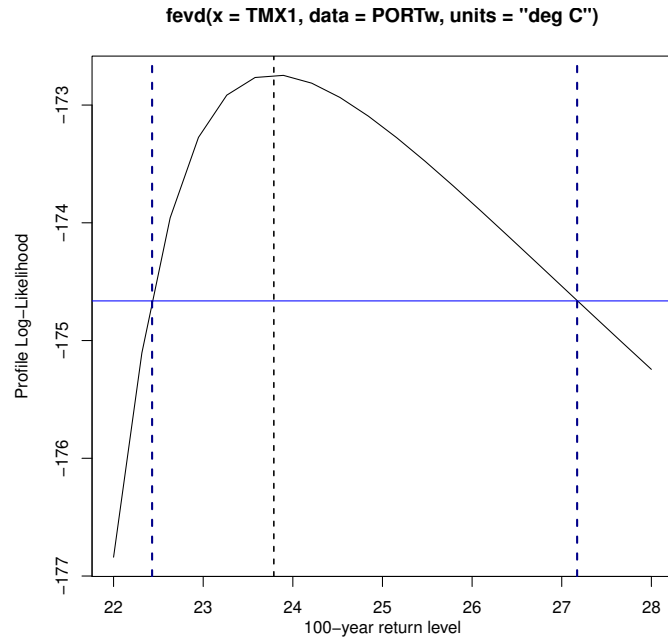**fevd(x = TMX1, data = PORTw, units = "deg C")**

Figure 4: profile-likelihood (solid black line) plot for the 100-year return level estimate for the GEV df fit to maximum winter temperature data (degrees centigrade) at Port Jervis, New York. Blue vertical (long dashed) lines emphasize where the profile likelihood crosses the blue horizontal line, and are the confidence interval estimates (see Coles 2001, for more explanation of this graph). The vertical (short dashed) line emphasizes the location of the MLE for the 100-year return level.

not merely the end-points used. Figure 4 shows the profile-likelihood for the 100-year return level. Because the profile-likelihood clearly crosses both blue vertical dashed lines (where they cross the blue horizontal line), the resulting intervals are believable. In the case of the shape parameter, the plot is not as legible, but both vertical blue lines appear to be in the range.[5]

Suppose now that we wish to know the probability of exceeding certain values based on our fitted GEV df. Recall that, because the fitted model is of the Weibull type, the upper bound for the df is approximately $15.1406132 + 2.9724952/0.2171486 \approx 28.83$ degrees centigrade, so the estimated probability of exceeding this or any value above it will be exactly zero. The code below finds the probability of exceeding 20, 25, 28.8 and 28.9 degrees centigrade based on the fitted GEV df stored in the fitted object `fit1`.

```
R> pextRemes(fit1, q = c(20, 25, 28.8, 28.9), lower.tail = FALSE)
```

It is also possible to simulate data from the fitted GEV df using the `rextRemes` function. For example, to simulate a sample of size 100, we can use the following code.

```
R> z <- rextRemes(fit1, 100)
R> fitz <- fevd(z)
R> plot(fitz)
```

---

[5]The procedure for searching for the up crossings differs from the procedure in **ismev**.

Bayesian estimation can also be employed with `fevd`. Future releases of the package may include greater utility for Bayesian analysis (e.g., Bayes factors, posterior predictive distributions, etc.), but a sample of the posterior distribution by way of MCMC simulation can already currently be obtained.

Independent normal df's are the default for the parameter prior df's.[6] The default can be changed via the `priorFun` argument, which is a character string naming any function that takes `theta` (numeric vector of location, scale and shape parameters) as its first argument, and returns a single numeric giving the prior density value. See `fevdPriorDefault` (the default function used by `fevd`) for an example of the type of function. Other arguments may also be included, and these can be passed into `fevd` via the `priorParams` argument, which is a named list. In the case of the default, the arguments `m` (single numeric or vector of length equal to the number of parameters in the model giving the prior mean(s)), `v` (single numeric or vector of length equal to the number of parameters giving the prior standard deviations) and `log` (logical stating whether to return the log of the prior (default) or not).

The default for the proposal df is to use a random walk approach (cf. Coles 2001, Section 9.1.3) whereby a realization from a normal df (with default mean of zero and standard deviation of 0.1) is added to the previous value. The proposal df can be changed using the `proposalFun` and `proposalParams` arguments. The first is a character naming the function, which must take the arguments `p` (a numeric vector of the previous/current parameters), `ind` (a number indicating which parameter from `p` to propose) as well as any additional arguments, and must return a single proposed parameter. In the MCMC algorithm used by `fevd`, the parameters are updated one at a time in random order at each step of the chain. Analogous to the `priorParams` argument, `proposalParams` is a named list of any other optional arguments to the function named by `proposalFun` besides `p` and `ind`. The default proposal function can optionally take the arguments `mean` and `sd`, which should be single numerics or numeric vectors with length equal to the number of parameters in the GEV model. These can be used to change the normal df parameters, and it is often useful to play with the values of `sd`.

The following code gives an example of fitting the GEV df to the Port Jervis, New York annual maximum winter temperature data using Bayesian estimation with the default prior and proposal functions and values. Note that the plot obtained with argument `"trace"` for the Bayesian fit gives different plots than for the MLE fit. The default parameter estimates from the MCMC samples is to take the mean, which can be changed by the user through the `FUN` argument.

```
R> fB <- fevd(TMX1, PORTw, method = "Bayesian")
R> postmode(fB)
R> plot(fB)
R> plot(fB, "trace")
R> ci(fB)
R> ci(fB, type = "parameter")
```

It is also possible to perform parameter estimation using the GMLE approach proposed by Martins and Stedinger (2000, 2001), with default priors set to those used therein. Chang-

---

[6]The default is to apply the log-transform to the scale parameter so that the normal prior df is for the log-scale parameter rather than the scale parameter, which is more appropriate when using a normal prior df. To apply the normal prior df to the untransformed scale parameter, set the `use.phi` argument to `FALSE`; note that `FALSE` is the default for all other estimation methods.
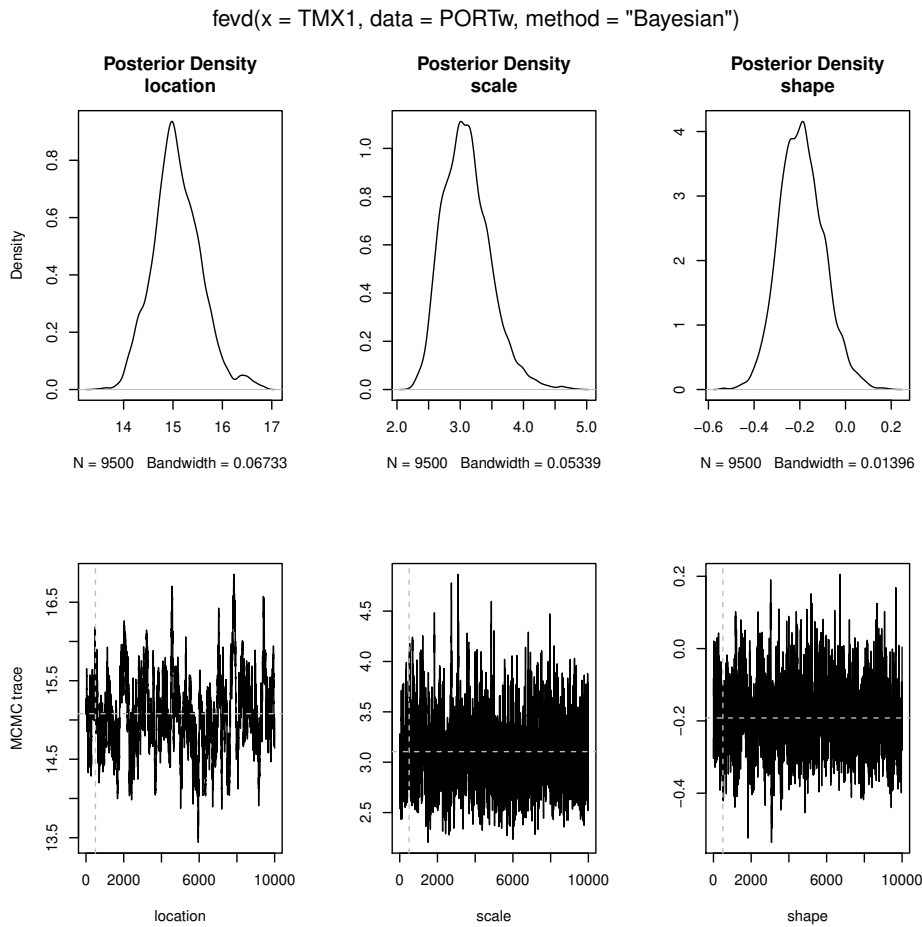
Figure 5: MCMC posterior parameter densities and trace plots from fitting the GEV df to maximum winter temperature data at Port Jervis, New York (cf. Figure 1) using Bayesian estimation. Vertical dashed lines represent the burn-in period, and horizontal dashed lines represent the mean of the MCMC sample (after removing the initial "burn-in" values), which are also the estimates reported via the `print` method function.

ing the prior information is performed in the same manner as for the Bayesian estimation procedure above. The approach does not require a proposal df. See the examples in the help file for `fevd` for an example of using this method.

Finally, we fit the GEV df to data using L-moments estimation. Note that L-moments estimation is the only estimation method of function `fevd` that does not handle incorporation of parameter covariates.

```
R> fitLM <- fevd(TMX1, PORTw, method = "Lmoments")
R> plot(fitLM)
R> ci(fitLM)
R> ci(fitLM, type = "parameter")
```

Next, we proceed with investigation of whether or not the inclusion of covariate information is statistically significant or not. The dominant mode of large-scale variability in mid-latitude

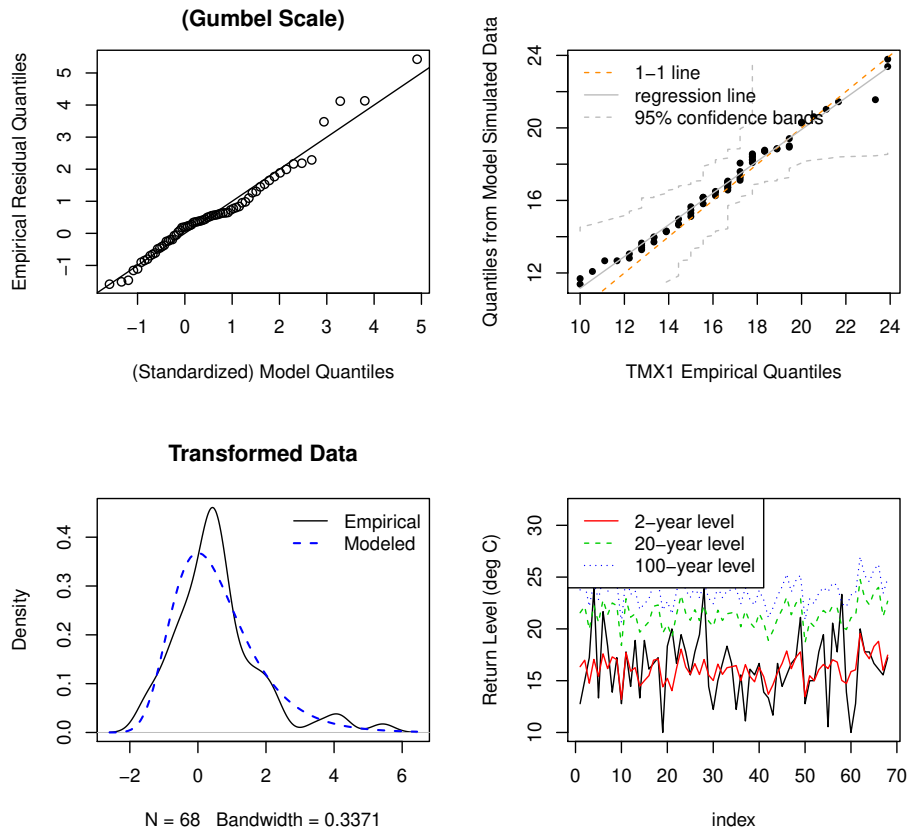fevd(x = TMX1, data = PORTw, location.fun = ~AOindex, units = "deg C")

Figure 6: Diagnostic plots from fitting the GEV df to Port Jervis maximum winter temperature (degrees centigrade) data with AO index as a covariate.

Northern Hemisphere temperature variability is the North Atlantic Oscillation-Arctic Oscillation (NAO-AO). Therefore, we begin by investigating whether or not this parameter should be included as a covariate in our GEV model for these data. It is generally a good idea to begin with the location parameter. The following commands will perform this fit and display various results.[7]

```
R> fit2 <- fevd(TMX1, PORTw, location.fun = ~ AOindex, units = "deg C")
R> plot(fit2)
R> plot(fit2, "trace")
```

Note that the diagnostic plots resulting from `plot(fit2)` are similar to those for `fit1`, but now the first quantile-quantile and density plots are on the Gumbel transformed scale (the data are transformed to a stationary sample) and the return level plot now displays the data time series with "effective" return levels, which are the return levels that would be estimated if the model were fixed at the parameter values for the specific covariate(s) used (cf. Gilleland and Katz 2011, Figure 2).

---

[7]Warnings are not critical here, as sometimes the gradient log-likelihood is not defined. In such cases, finite differences are used instead.

	2-year level	20-year level	100-year level
AO index $= -1$	15.05	20.26	22.47
AO index $= 1$	17.36	22.56	24.77

Table 2: Estimated "effective" return levels for the GEV df fitted to maximum winter temperature (degrees centigrade) at Port Jervis, New York with AO index as a (linear) covariate in the location parameter.

The fitted model location parameter estimate is $\hat{\mu}(\text{AO index}) = 15.25 + 1.15\text{AO index}$. That is, the estimated parameters (standard errors) are $\hat{\mu}_0 \approx 15.25$ (0.36), $\hat{\mu}_1 \approx 1.15$ (0.32), as well as $\hat{\sigma} \approx 2.68$ (0.24) and $\hat{\xi} \approx -0.18$ (0.07).

We can use the likelihood-ratio test again to test whether or not inclusion of the AO index into the location parameter is statistically significant.

```
R> lr.test(fit1, fit2)
```

The test yields a very low p value (≈ 0.0005653) indicating that the null hypothesis of the stationary model should be rejected. To obtain specific "effective" return levels, we must set up an object that informs the `return.level` function about which covariate(s) choice(s) we want to make. For example, the following code obtains "effective" return level estimates for values when the AO index is -1 and 1.

```
R> v <- make.qcov(fit2, vals = list(mu1 = c(-1, 1)))
R> return.level(fit2, return.period = c(2, 20, 100), qcov = v)
```

It is useful to study the format of v, which is a matrix with columns named for the parameters and an additional threshold column (with all `NA`s for the GEV df). Note that the location parameter is modeled with two parameters in this case, so its names are `mu0` and `mu1`. The rows represent specific parameter values for each covariate. Columns of parameters that are fixed have all ones. Results from running the above code are reproduced in Table 2.

It is also possible to incorporate covariates into other parameters. To investigate whether or not AO index should be included as a covariate in the scale parameter, the following code can be used.

```
R> fit3 <- fevd(TMX1, PORTw, location.fun= ~ AOindex,
+    scale.fun = ~ AOindex, units = "deg C")
R> lr.test(fit2, fit3)
```

The p value from the above likelihood-ratio test is on the order of 0.5 indicating that incorporation of the AO index to the scale parameter, in addition to the location parameter, is not statistically significant. The likelihood-ratio test requires that the models be nested in this way. We could also utilize the AIC or BIC, which could be applied regardless of whether or not the models are nested. For `fit2` (AO index in the location parameter) the AIC is 341.5984 and BIC is 350.4764. If we fit the model with AO index in only the scale parameter, we get an AIC of 353.4567 and BIC of 362.3347, and note that varying the scale parameter without varying the location parameter is generally not recommended. In each case, the AIC and BIC both favor the model with AO index in the location parameter rather than the scale parameter (lower values of AIC/BIC are better).

If we want to fit a model with a log link function in the scale parameter, then we can modify the previous code above by setting the `use.phi` argument to `TRUE`.

```
R> fit4 <- fevd(TMX1, PORTw, location.fun =  ~ AOindex,
+    scale.fun =  ~ AOindex, use.phi = TRUE, units = "deg C")
```

The above code fits a model with AO index as a covariate in the location parameter as before and now incorporates the index in the scale parameter in a non-linear manner. Specifically, $\sigma(\text{AO index}) = \exp(\phi_0 + \phi_1 \text{AO index})$. However, a check of the likelihood-ratio test against the model with AO index in the location parameter only fails to reject the null hypothesis (p value > 0.5).

It can sometimes be advantageous to use $\phi = \log \sigma$ in the numerical optimization for models without covariates in the scale parameter. This can easily be performed by setting the `use.phi` argument to `TRUE` for those models, and results will be displayed for $\sigma = \exp \phi$.

```
R> fevd(TMX1, PORTw, use.phi = TRUE)
```

Comparison of the above result with that of `fit1` shows that the estimates (and their standard errors) are nearly, but not quite, identical.

4. Threshold excess approach

To demonstrate the threshold excess approach, we will make use of the hurricane damage data set available with **extRemes** (Pielke and Landsea 1998; Katz 2002). Figure 7 shows plots of these data where the top right and bottom left panels are similar to Figures 3 and 4 in Katz (2002).

Figure 7 can be reproduced in R with the following commands, where the `qqnorm` function that allows for the confidence bands is contributed by Peter Guttorp.

```
R> data("damage", package = "extRemes")
R> par(mfrow = c(2, 2))
R> plot(damage$Year, damage$Dam, xlab = "Year",
+    ylab = "U.S. Hurricane Damage (billion USD)", cex = 1.25,
+    cex.lab = 1.25, col = "darkblue", bg = "lightblue", pch = 21)
R> plot(damage[, "Year"], log(damage[, "Dam"]), xlab = "Year",
+    ylab = "ln(Damage)", ylim = c(-10, 5), cex.lab = 1.25,
+    col = "darkblue", bg = "lightblue", pch = 21)
R> qqnorm(log(damage[, "Dam"]), ylim = c(-10, 5), cex.lab = 1.25)
```

Before fitting the GP df to the data, it is first necessary to choose a threshold. It is important to choose a sufficiently high threshold in order that the theoretical justification applies thereby reducing bias. However, the higher the threshold, the fewer available data remain. Thus, it is important to choose the threshold low enough in order to reduce the variance of the estimates. Different methods have been suggested for choosing the threshold. The **extRemes** package has two main functions for diagnosing an appropriate choice: `threshrange.plot` and `mrlplot`. The former repeatedly fits the GP df to the data for a sequence of threshold choices along
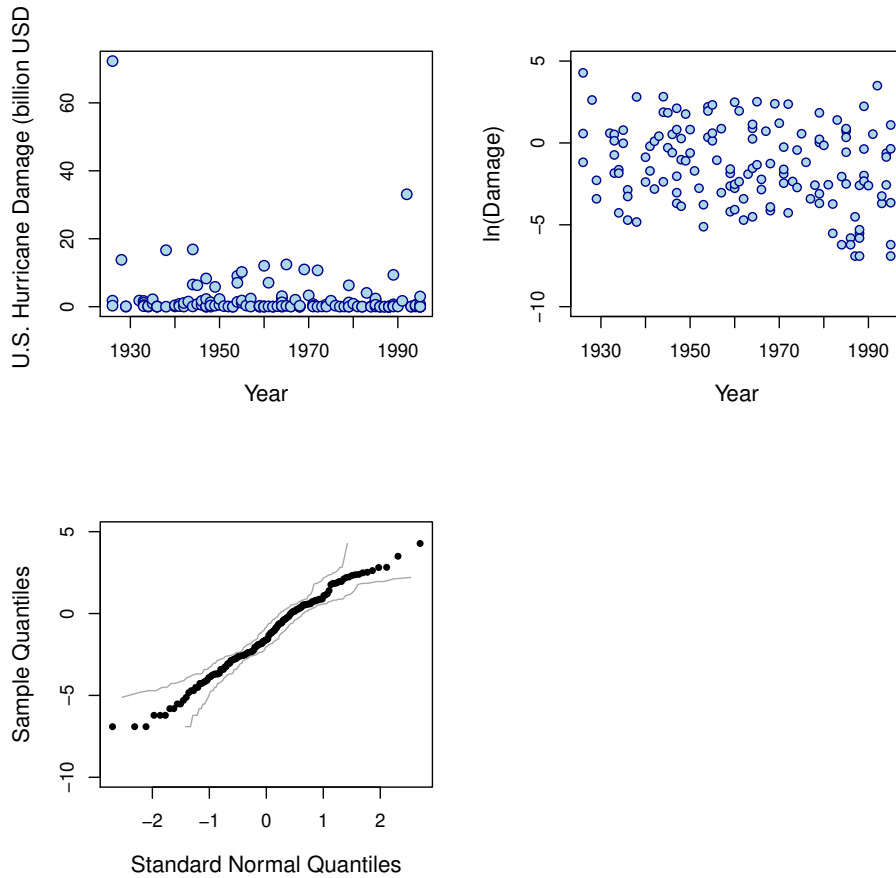
Figure 7: Estimated economic damage caused by US hurricanes from 1926 to 1995. From top left to bottom left: scatter plot of damage against year, scatter plot of log-transformed damage against year and normal quantile plot of log damage with 95% confidence bands. The last two are similar to Figures 3 and 4 in Katz (2002), but where all 144 values are shown (Katz 2002 removed values below 0.01 billion USD).

with some variability information[8] while the latter plots the mean excess values for a range of threshold choices with uncertainty (cf. Coles 2001, Section 4.3.1). The code below produces the plots in Figure 8. We also note that only seven values exceed 12 billion USD and only five exceed 13 billion USD. Therefore, in order to ensure we have enough data and to more easily interpret the mean residual life plot, we will restrict it to the range of zero to twelve.

```
R> threshrange.plot(damage$Dam, r = c(1, 15), nint = 20)
R> mrlplot(damage$Dam, xlim = c(0, 12))
```

The top row of panels shows the fitted (reparameterized) scale and shape parameters over a range of twenty equally spaced thresholds from 1 billion USD to 15 billion USD; the scale parameter, $\sigma(u)$, is adjusted to σ^*, so that it is not a function of the threshold, by $\sigma^* = \sigma(u) - \xi u$. From these plots, a subjective selection of 6 billion USD appears to yield estimates

---

[8]**extRemes** version ≥ 2.0 plots confidence intervals, where previously a multiple of the standard error was used via **ismev**.
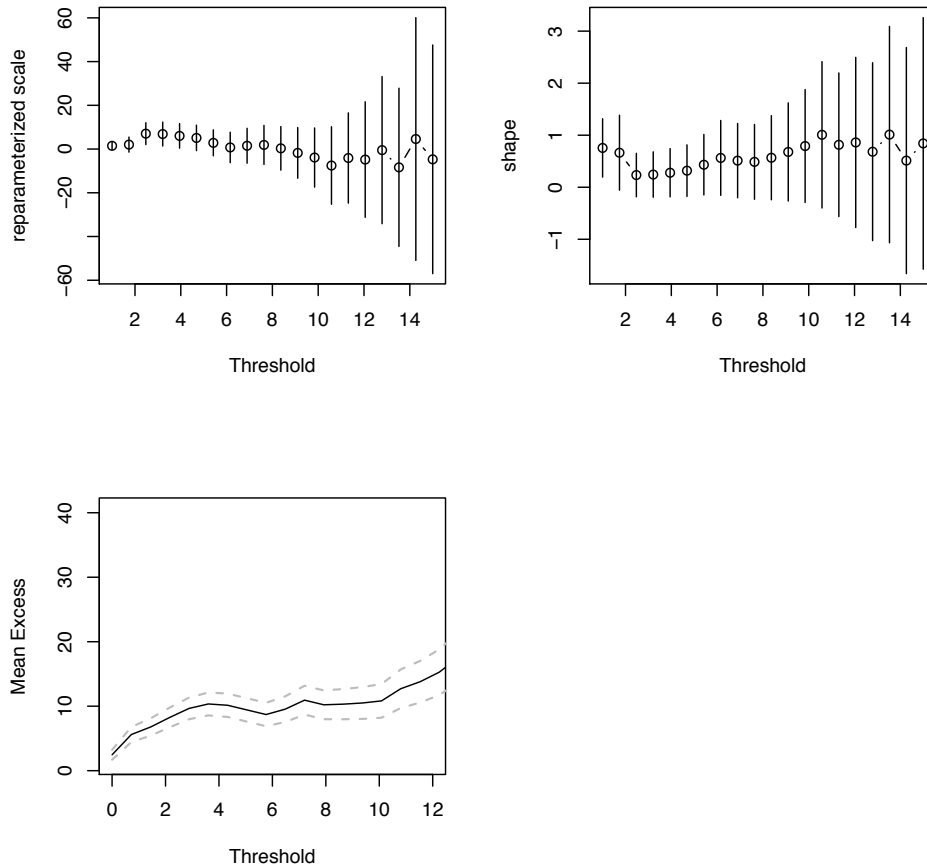
Figure 8: Threshold selection diagnostic plots for the GP df fit to the US hurricane damage data in Figure 7. Reparameterized scale parameter with 95% confidence limits over a range of 20 equally spaced thresholds from 1 billion USD to 15 billion USD (top left), shape parameter with 95% confidence intervals for the same range of thresholds (top right), and the mean residual life (mean excess) plot (bottom left) with dashed gray lines indicating the 95% confidence intervals for the mean excesses.

that will not change much, within uncertainty bounds, as the threshold increases further, while remaining low enough as to utilize as much of the data as possible.

The last panel of Figure 8 (bottom left) is often more difficult to interpret, but the idea is to select a threshold whereby the graph is linear, again within uncertainty bounds, as the threshold increases.[9] Again, 6 billion USD appears to be a reasonable choice for the threshold as a reasonably straight line could be placed within the uncertainty bounds from this point up.

One additional consideration concerns the rate of exceeding the threshold. Although it is not important for fitting the GP df to the data, it will come into play when calculating return levels from the GP df. In order to obtain an estimate of the rate of exceeding 6 billion USD,

---

[9]The reason for checking for linearity is that if the GP df is valid for excesses of $X_1, \ldots, X_n$ over a threshold u_0, then for $u > u_0$, the GP is also valid for excesses of $X_1, \ldots, X_n$ over u subject to an appropriate change in the scale parameter. Then, the mean for the GP df is given by $\mathsf{E}[X - u | X > u] = \sigma(u)/(1-\xi) = (\sigma(u_0) + \xi u)/(1-\xi)$, which is a linear function of the threshold.
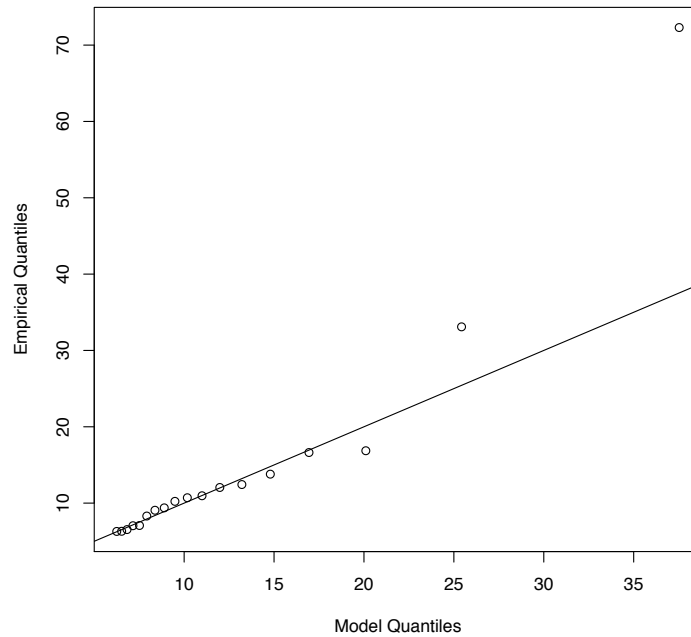
Figure 9: Quantile-quantile plots for the GP df fit to the hurricane economic damage data.

we will make use of the `time.units` argument to `fevd`. Because the interest is in the annual return level, we need to take into account the number of data points per year. From Figure 7, it is clear that some years have more than one hurricane, and in fact, some years do not have a hurricane. A reasonable solution is to use the average number of hurricanes per year.

```
R> range(damage$Year)
R> 1995 - 1926 + 1
R> dim(damage)
R> 144 / 70
R> fitD <- fevd(Dam, damage, threshold = 6, type = "GP",
+    time.units = "2.06/year")
R> plot(fitD)
```

Inspection of the diagnostic plots from the above code shows a large apparent "outlier" toward the upper end (the devastating Greater Miami hurricane in 1926). But such behavior is fairly common for the most extreme observation (in part because of considerable uncertainty), and does not provide strong evidence against the validity of the assumptions for using the GP df. It is possible to calculate probabilities from the model fit using the `pextRemes` function. Suppose, for example, that we are interested in knowing the conditional probability of exceeding 20, 40, 60 and 100 billion US dollars in economic damage, given a hurricane whose damage exceeds 6 billion US dollars. The following line of code shows us that the estimates, according to the fitted model, are about 0.16, 0.05, 0.02, and 0.01, respectively.

```
R> pextRemes(fitD, c(20, 40, 60, 100), lower.tail = FALSE)
```

For a second example, we will look at precipitation data from a rain gauge in Fort Collins, Colorado, USA, which come from the Colorado Climate Center, Colorado State University
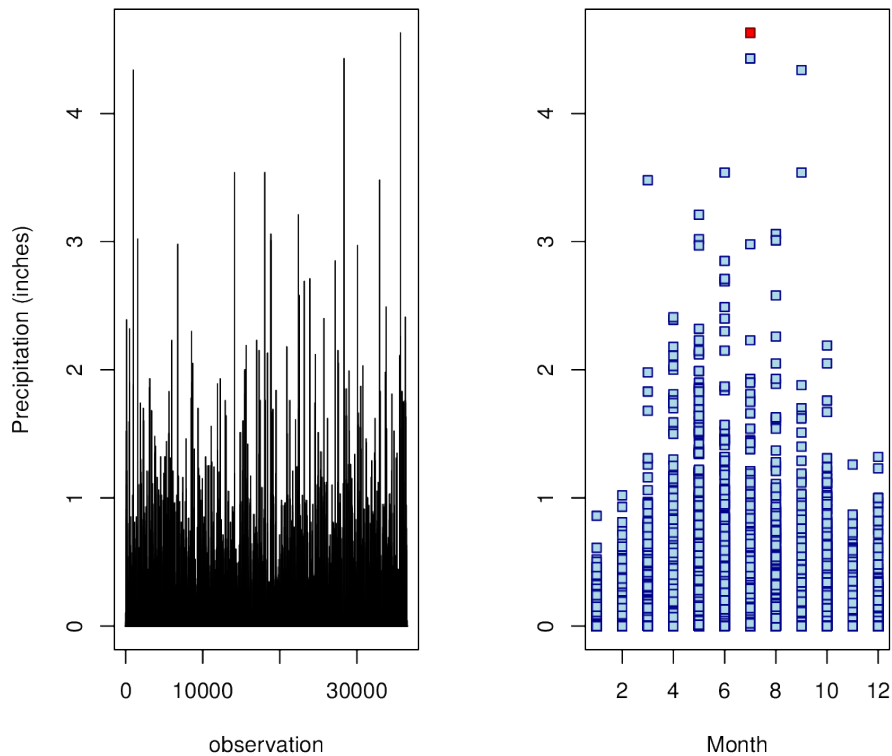
Figure 10: Precipitation (inches) data in Fort Collins, Colorado, USA, against the time of the observations in days (`obs` column of data frame, left) and against month (right). The red circle indicates the value at this gauge on the day of the July 28, 1997, flooding event that occurred in the area (though not near this particular rain gauge). In the data set, it is associated with the date of July 29, 1997, because it is an accumulation over the 24 hours prior.

and are analyzed in Katz *et al.* (2002). The data are available with **extRemes** by using `data("Fort", package = "extRemes")` and plotted in Figure 10. From the plots, there is no apparent long-term trend over time, but some seasonality in extremes exists. A flooding event occurred in this area on July 28, 1997, though not in the precise vicinity of this particular rain gauge.

As in the hurricane damage example above, it is first necessary to determine a threshold in order to fit the GP df to these data. Figure 11 shows the results of the code below.

```
R> threshrange.plot(Fort$Prec, c(0.01, 2), nint = 30)
```

As always, it is a somewhat subjective decision as to what is an appropriate threshold choice. However, the two parameters do not seem to vary substantially, and within uncertainty bounds, beyond about 0.395 inches. The following code fits first a stationary model and then a model that incorporates the seasonality evidenced by Figure 10 (right). It is not necessary to specify the `time.units` argument for these data because the data are recorded daily, yielding approximately 365.25 days per year, which is the default assumption for `fevd`.

```
R> fitFC <- fevd(Prec, Fort, threshold = 0.395, type = "GP")
R> plot(fitFC)
```

threshrange.plot(x = Fort$Prec, r = c(0.01, 2), nint = 30)
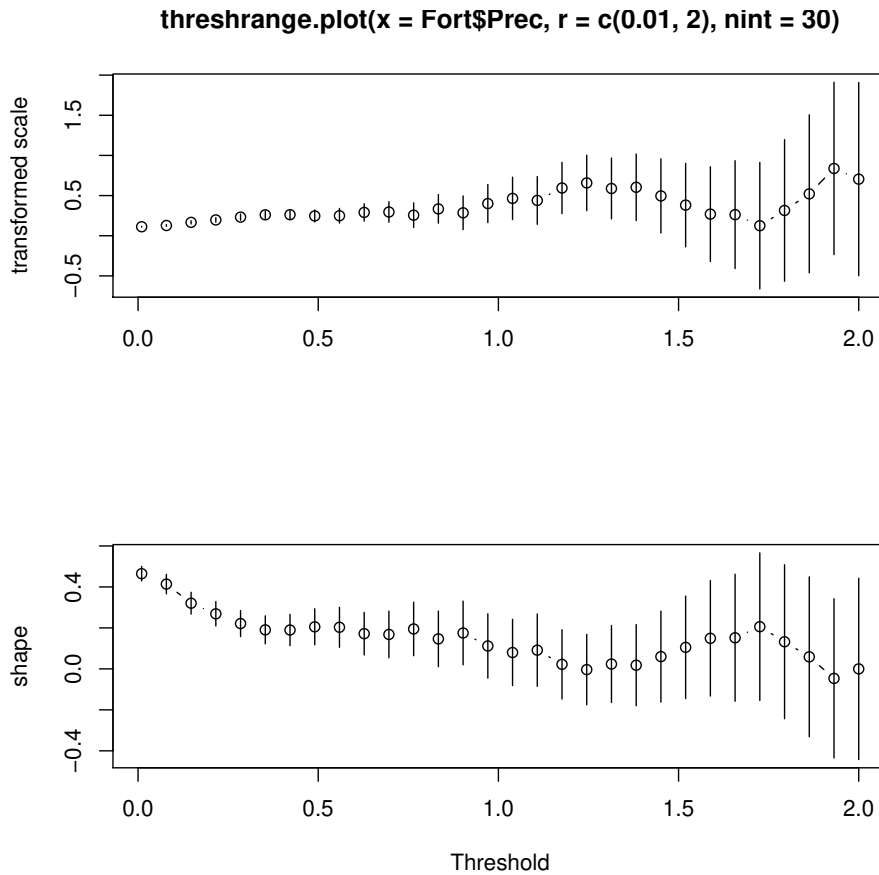

Figure 11: Results from fitting the GP df to the Fort Collins, Colorado, USA precipitation data from Figure 10 over a range of thresholds.

```
R> fitFC2 <- fevd(Prec, Fort, threshold = 0.395, scale.fun =
+     ~ cos(2 * pi * tobs / 365.25) + sin(2 * pi * tobs / 365.25),
+     type = "GP", use.phi = TRUE, units = "inches")
R> plot(fitFC2)
R> lr.test(fitFC, fitFC2)
```

The object `fitFC2` models the scale parameter as

$$\sigma(t) = \exp\left(\phi_0 + \phi_1 \cos(2\pi \cdot t/365.25) + \phi_2 \sin(2\pi \cdot t/365.25)\right), \, t = 1, 2, \dots, 365. \qquad (13)$$

The result of the likelihood-ratio test yields a very small p value ($\approx 5.22 \times 10^{-6}$) indicating that the addition of the seasonality terms are statistically significant. We can obtain 95% profile-likelihood confidence intervals for the parameter ϕ_1 (the second parameter in the fitted object) and ϕ_2 (the third parameter) in `fitFC2` using the `ci` method function. Specifying `verbose` to be `TRUE` gives some progress information on the screen, but also creates the profile-likelihood plot. The `xrange` argument was determined by trial and error. The resulting profile-likelihood plots are displayed in Figure 12.

```
R> ci(fitFC2, type = "parameter", which.par = 2, method = "proflik",
+     xrange = c(-0.5, 0.01), verbose = TRUE)
```
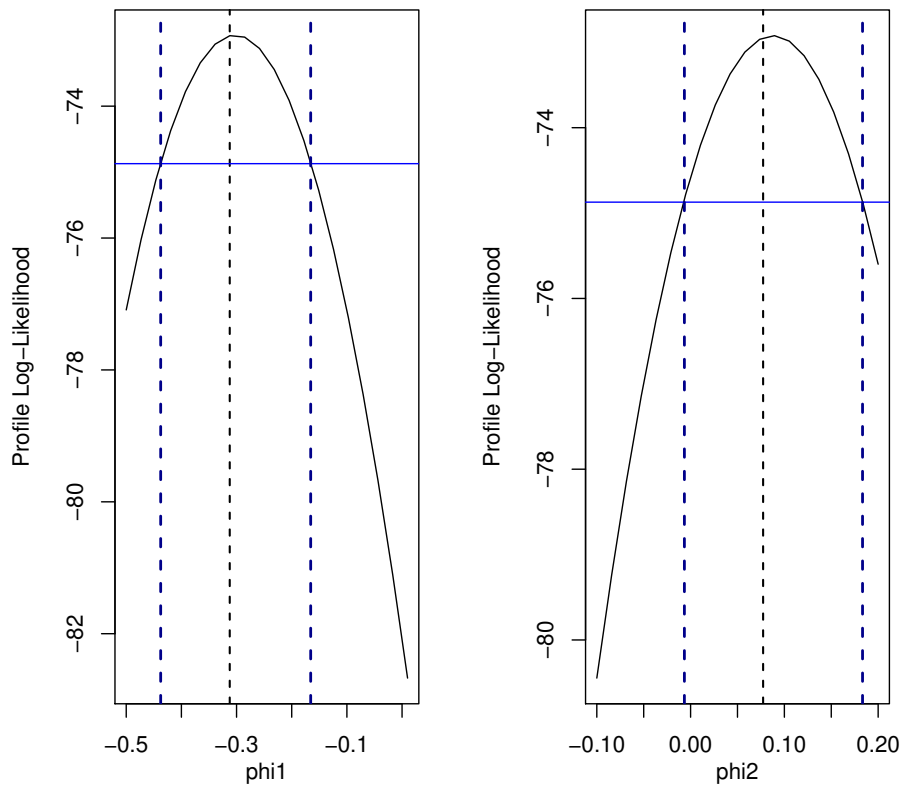
Figure 12: Profile-likelihood plot for parameters ϕ_1 (left) and ϕ_2 (right) in the GEV df fit to Fort Collins precipitation data with a fixed threshold of 0.395 and seasonal variation in the scale parameter according to Equation 13.

```
R> ci(fitFC2, type = "parameter", which.par = 3, method = "proflik",
+    xrange = c(-0.1, 0.2), verbose = TRUE)
R> ci(fitFC2, type = "parameter")
```

Although zero falls in the 95% confidence interval for ϕ_2 $(-0.01, 0.18)$, it is really just a trigonometric "trick" to re-express a single sine wave with unknown phase and amplitude so that a linear model can be obtained instead of a non-linear one. We will keep the second term in our subsequent analysis.

Now that we have a reasonable model selected, we may be interested in the probability of exceeding 4.63 inches (the value that occurred at this rain gauge on July 29, 1997). For illustrative purposes we will also find the probability of exceeding 1.54 inches (the value that occurred at this rain gauge the day before the flooding event). Because the model contains covariate information in the parameters, care must be taken when calling `pextRemes`. It is necessary to tell `pextRemes` the desired values of the covariates for which to calculate the probability.

```
R> v <- matrix(1, 730, 5)
R> v[, 2] <- cos(2 * pi * rep(1:365 / 365.25, 2))
R> v[, 3] <- sin(2 * pi * rep(1:365 / 365.25, 2))
R> v[, 5] <- 0.395
```

```
R> v <- make.qcov(fitFC2, vals = v, nr = 730)
R> FCprobs <- pextRemes(fitFC2, q = c(rep(1.54, 365), rep(4.63, 365)),
+    qcov = v, lower.tail = FALSE)
```

Alternatively, it is possible to calculate the scale parameter value for our covariate(s) of interest, and use `pevd` to find the probabilities. However, they must be calculated for one value of the scale parameter at a time. Below it is shown how to do this for the 29th of July (the day of the year for which the highest amount (4.63 in) occurred, which corresponds to the 210th day of the year assuming 28 days for February.

```
R> phi <- -1.24022241 - 0.31235985 * cos(2 * pi * 1:365 / 365.25) +
+    0.07733595 * sin(2 * pi * 1:365 / 365.25)
R> sigma <- exp(phi)
R> pevd(c(1.54, 4.63), scale = sigma[210], shape = 0.177584305,
+    threshold = 0.395, type = "GP", lower.tail = FALSE)
```

For the GP df, it is not clear how to interpret these probabilities because the rate of exceeding the threshold may also vary seasonally, but cannot be modeled under the current framework. One possible solution is to employ a PP model instead, as will be discussed in the next section.

It is also possible to vary the threshold in addition to (or instead of) the scale parameter. For example, the following code fits a GP df with a varying threshold.

```
R> u <- numeric(dim(Fort)[1])
R> u[ is.element(Fort$month, c(1, 2, 11, 12))] <- 0.395
R> u[ is.element(Fort$month, c(3, 4, 9, 10))] <- 0.25
R> u[ is.element(Fort$month, 5:8) ] <- 0.895
R> fitFC3 <- fevd(Prec, Fort, threshold = u, scale.fun =
+    ~ cos(2 * pi * tobs / 365.25) + sin(2 * pi * tobs / 365.25),
+    type = "GP", verbose = TRUE)
R> plot(fitFC3)
```

5. Point process approach

Fitting the GP df to excesses over a high threshold and also estimating the frequency of exceeding the threshold by fitting a Poisson distribution is sometimes referred to as the orthogonal approach to estimating the PP, or Poisson-GP model. An alternative way to estimate the PP is to estimate both the frequency and intensity df's simultaneously in a single model framework by optimizing the likelihood given in Equation 12. Such a modeling approach is sometimes referred to as a two-dimensional PP.

```
R> fit <- fevd(Prec, Fort, threshold = 0.395, type = "PP",
+    units = "inches", verbose = TRUE)
R> plot(fit)
R> plot(fit, "trace")
R> ci(fit, type = "parameter")
```
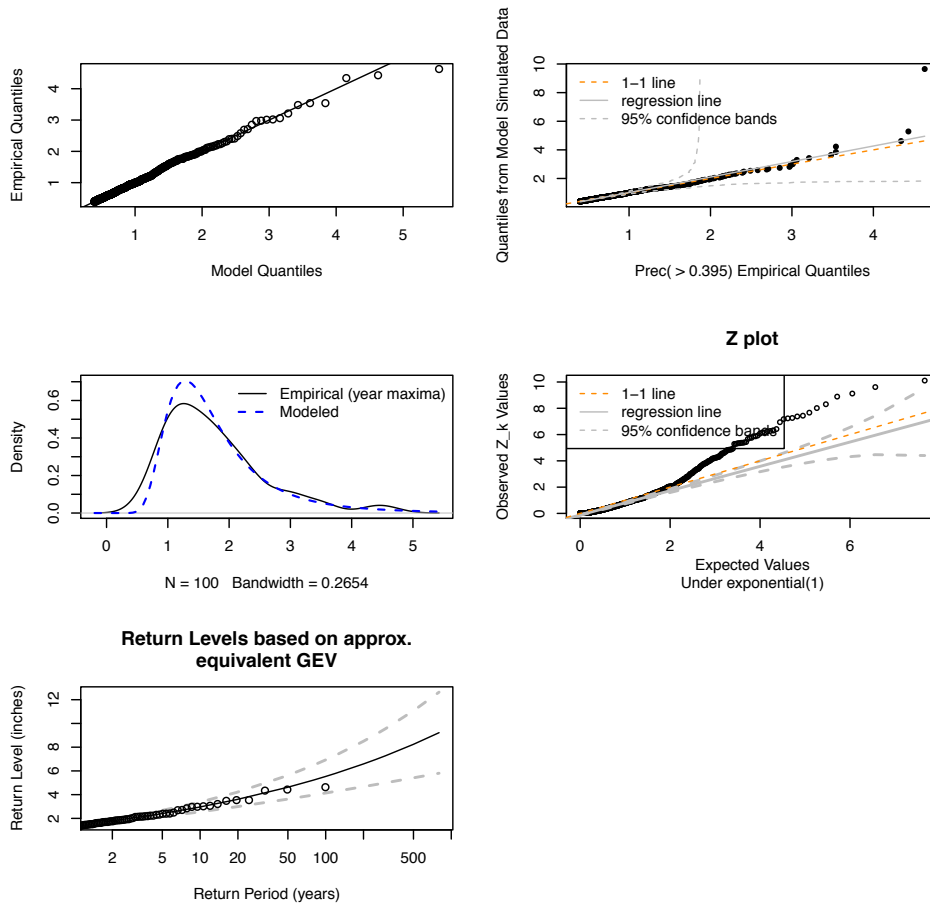
Figure 13: Diagnostic plots from fitting a PP model to the Fort Collins, Colorado precipitation data set. See text for an explanation of the plots.

Figure 13 shows the results for the first `plot` command above. The top two quantile-quantile plots are those of the corresponding GP df so that only data exceeding the threshold are shown. The density plot (first column second row), on the other hand, shows the density for the equivalent GEV df (i.e., the fitted two-dimensional PP parameterized in terms of the GEV df) so that the empirical density shown as the solid black line in the figure is for block maxima data where the function attempts to find the block maxima[10] from user input in the original fitted object; the fit appears to be reasonable, especially considering that these are not the data to which the model is being fitted, and the GEV df is not being directly fitted to them. For non-stationary models, the density plot is not provided.

The next diagnostic plot (labeled Z plot) is a quantile-quantile plot first proposed by Smith and Shively (1995) (the upper left quantile-quantile plot was also proposed there). the Z plot is created in the following manner. First, for observations beginning at time T_0, the times

---

[10]The function uses the `span` and `npy` values of the fitted object in order to find the block maxima. That is, a vector of integers from 1 to `span`, each repeated `npy` times, is created, and the maximum of the data is found from this stratification. If `npy` and/or `span` is not an integer, then it is rounded to the nearest integer value. If the stratification vector is longer than the original data, it is simply truncated to be the same length. If it is shorter than the original data, then the final stratification is extended to the length of the data.

when the data exceed the high threshold, T_k, $k = 1, 2, \ldots$ are obtained, and the Poisson parameter (which may vary with time t) are integrated from exceedance time $k - 1$ to k to give the random variables

$$Z_k = \int_{T_{k-1}}^{T_k} \lambda(t)dt = \int_{T_{k-1}}^{T_k} \{1 + \xi(t)(u - \mu(t))/\sigma(t)\}^{-1/\xi(t)} \, dt, \, k \geq 1. \tag{14}$$

By construction, the random variables Z_k should be independent exponentially distributed with mean 1, and the quantile-quantile plot of Z_k against the expected quantile values from a mean-one exponential df allows for a diagnostic test of this assumption. The Z plot in Figure 13 is an example of such a quantile-quantile plot for the fit to the Fort Collins precipitation data. In this case, the plot suggests that the assumptions for the PP model are not met; in particular, that the mean waiting time between events does not follow a mean one exponential df. For the present example, it turns out that while the threshold of 0.395 may be high enough to adequately model the threshold excesses, it may be too low to model the frequency of occurrence. A possible solution for this case (not shown) is to increase the threshold; e.g., a threshold of 2 inches has been found to yield a fairly straight Z plot. However, 2 inches is too high if considering other aspects, such as modeling seasonality.

Similar to the density plot, the return level plot (lower right panel) estimates the block maxima from the fitted object and uses the equivalent GEV representation to estimate them for annual maxima.

With the PP model, we now have a threshold to choose, as well as location, scale and shape parameters to estimate, any of which can incorporate covariate information just as with the block maxima and threshold excess approaches.

Sometimes interest is in minima or deficits under a low threshold rather than maxima and excesses over a high threshold. Although slightly more complicated, the same theory applies because of the relation $\min\{X_1, \ldots, X_n\} = -\max\{-X_1, \ldots, -X_n\}$ and similarly for deficits under a low threshold. For example, the Phoenix minimum temperature data (Figure 14) included with **extRemes**, `data("Tphap", package = "extRemes")`, is of interest because of the urban heat island effect, which is well known to be greater for minimum temperature than for maximum temperature. The data were provided by the US National Weather Service Forecast office at the Phoenix Sky Harbor Airport, and represent maximum (`MaxT`) and minimum (`MinT`) daily temperature in degrees Fahrenheit for the summer months of July through August 1948 to 1990 (Balling, Skindlov, and Phillips 1990; Tarleton and Katz 1995).

Although it may be prudent to vary the threshold, given the clear increasing trend over time, we begin by fitting a fixed stationary PP model to the (negative) minimum temperature data. As before, to help select a threshold, we can use the following code. Because of the discrete nature of the data, we use thresholds chosen to be between integer values in order to minimize the effect on the GP fit.

```
R> threshrange.plot(-Tphap$MinT, r = c(-75.5, -68.5), nint = 20, type = "PP")
```

Inspection of the plots in Figure 15 suggests that a threshold of -73 for the fixed stationary model would be appropriate. Next, note that there are 62 days (observations) per year included in the dataset.

```
R> fit1 <- fevd(-MinT ~ 1, Tphap, threshold = -73, type = "PP",
+    units = "deg F", time.units = "62/year", verbose = TRUE)
```
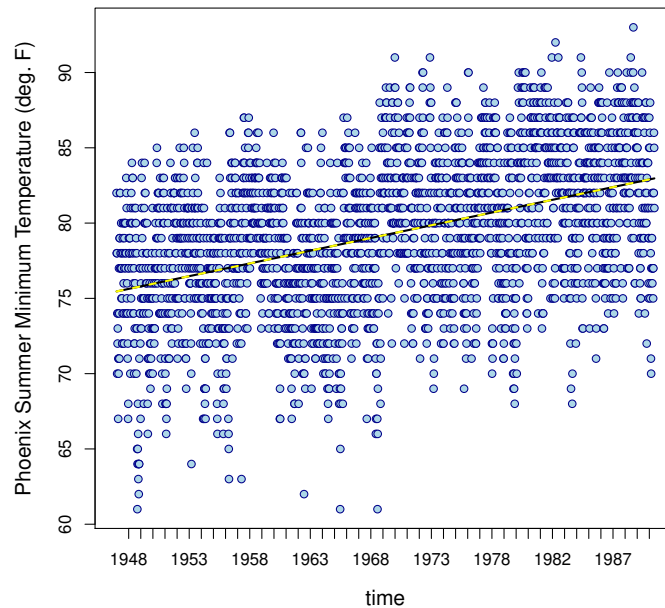
Figure 14: Summer (July through August) minimum temperature (deg. Fahrenheit) at Phoenix Sky Harbor Airport from 1948 to 1990. Yellow dashed line is a regression line fit to the data.

```
R> plot(fit1)
R> plot(fit1, "trace")
```

The diagnostic plots display results for the negative transformation of minimum temperature. As expected, the plots suggest that our assumptions for using the PP model are not met. From Figure 14, it appears practical to incorporate a linear trend into the threshold. The code below uses the threshold model $u(\text{year}) = -68 - 7(\text{year} - 48)/42$ (orange line in Figure 16).

```
R> fit2 <- fevd(-MinT ~ 1, Tphap, threshold = c(-68,-7),
+    threshold.fun = ~ I((Year - 48)/42), location.fun = ~ I((Year - 48)/42),
+    type = "PP", time.units = "62/year", verbose = TRUE)
R> plot(fit2)
```

Examination of Figure 16, suggests that the data may be dependent even over the threshold, which may impact uncertainty information (not shown) pertaining to parameter and return level estimates. We take this issue up in Section 6. A further problem is the Z plot (not shown), which shows that the assumptions for the frequency part of the model are not met. In this case, increasing the threshold does not help. However, it will be shown that accounting for the dependence above the threshold solves the problem for this example.

Suppose we want to compare the 100-year "effective" return level estimates between, say 1952 (i.e., the fifth year) and 1988 (year 41) along with confidence intervals. We can do this in the following manner.

```
R> v1952 <- make.qcov(fit2, vals = list(mu1 = (5 - 48) / 42))
R> v1988 <- make.qcov(fit2, vals = list(mu1 = (41 - 48) / 42))
```
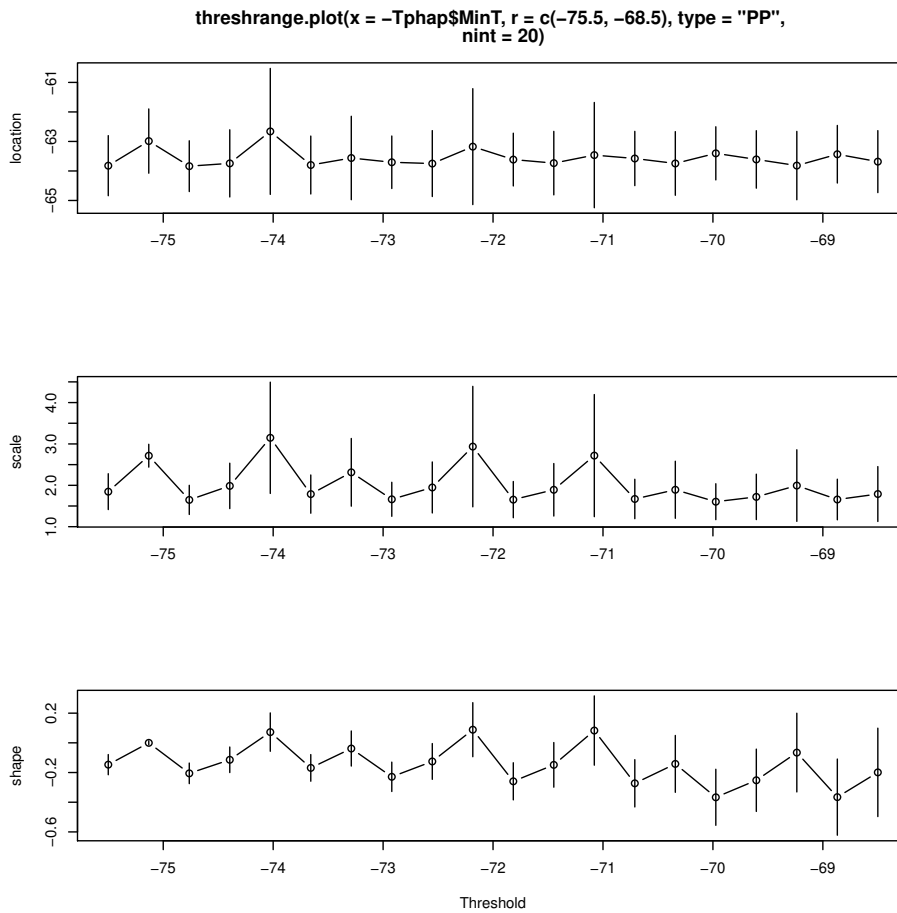
Figure 15: PP model fit to the (negative) minimum temperature data from Figure 14 over a range of thresholds.

```
R> return.level(fit2, return.period = 100, do.ci = TRUE, qcov = v1988,
+    qcov.base = v1952)
```

Recalling that the model is fit to the negative of minimum temperature, the above code tells us that the estimated difference in "effective" return levels (95% confidence interval based on the normal approximation) between 1988 and 1952 is about 7.23 (6.07, 8.39) degrees centigrade. With zero not contained in the interval, this result suggests that the 100-year return level of annual minimum temperature has risen, with statistical significance at the 5% level, on the order of seven degrees centigrade between 1952 and 1988. Of course, it is possible using the code to obtain "effective" return level values for any arbitrary year into the future, but caution should be exercised in practice concerning whether or not it is believable for the upward linear trend in minimum temperatures to continue to be valid. Generally, it is ill-advised to extrapolate such a statistical trend beyond the range of the data.

Next, we continue with the Fort Collins precipitation example. Recall that we had fit a GP df to these data, and found that inclusion of a seasonally varying covariate in the scale parameter was significant. The following fits a PP model to the data first without any covariates and then with a seasonally varying location parameter, as well as for seasonally
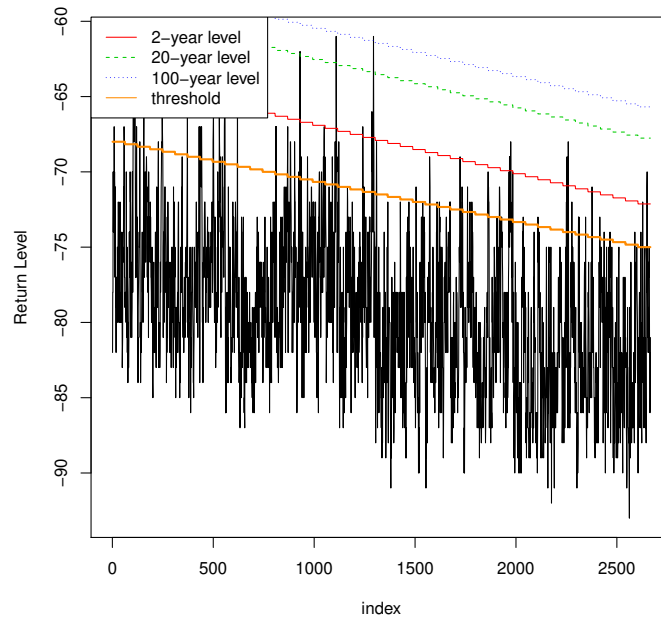
Figure 16: "Effective" return levels for negative summer minimum temperature (deg. Fahrenheit) at Sky Harbor Airport.

varying scale parameter; along with a likelihood-ratio test for the inclusion of the seasonally varying covariate.

```
R> fitFCpp <- fevd(Prec, Fort, threshold = 0.395, type = "PP")
R> fitFCpp2 <- fevd(Prec, Fort, threshold = 0.395, location.fun =
+    ~ cos(2 * pi * tobs / 365.25) + sin(2 * pi * tobs / 365.25),
+    type = "PP", units = "inches")
R> fitFCpp3 <- fevd(Prec, Fort, threshold = 0.395, location.fun =
+    ~ cos(2 * pi * tobs / 365.25) + sin(2 * pi * tobs / 365.25),
+    scale.fun = ~ cos(2 * pi * tobs / 365.25) + sin(2 * pi * tobs / 365.25),
+    type = "PP", use.phi = TRUE, units = "inches")
R> lr.test(fitFCpp, fitFCpp2)
R> lr.test(fitFCpp2, fitFCpp3)
```

The p value for the likelihood-ratio statistic comparing the stationary fit to the one that incorporates seasonality into the location parameter is less than 2.2×10^{-16}, which implies that the inclusion of seasonality in this parameter is significant, as expected. The additional inclusion of seasonality into the scale parameter is also highly significant suggesting that the model with seasonality in both terms should be preferred. Note that we do not fit a model with seasonally varying scale parameter without also varying the location parameter because the concept of the scale parameter depends on the location parameter.

It is possible to calculate probabilities of exceeding a specified amount for any particular value(s) of the covariates. Below, we calculate the probability of exceeding 1.54 inches for each day of a year, along with the effective 100-year return levels (with 95% confidence intervals based on a normal approximation).
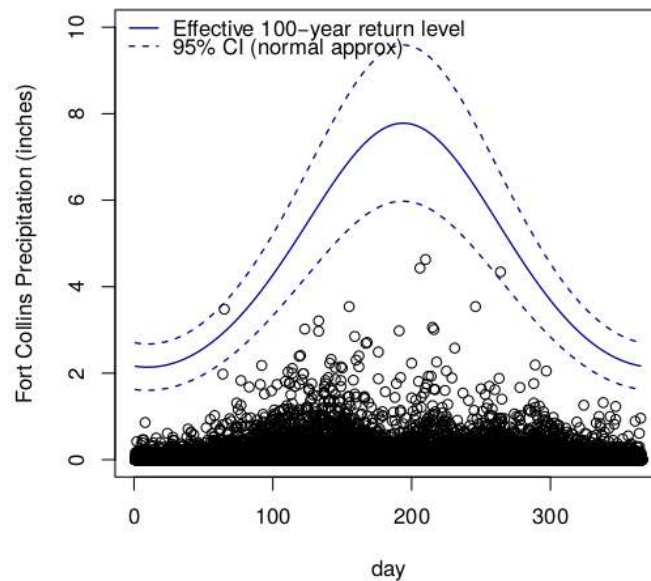
Figure 17: Fort Collins daily precipitation (inches) with effective 100-year return levels from PP model fit with seasonally varying location and scale parameters.

```
R> v <- make.qcov(fitFCpp3, vals = list(
+    mu1 = cos(2 * pi * 1:365 / 365.25), mu2 = sin(2 * pi * 1:365 / 365.25),
+    phi1 = cos(2 * pi * 1:365 / 365.25), phi2 = sin(2 * pi * 1:365/365.25)))
R> v[1:10, ]
R> p1.54 <- pextRemes(fitFCpp3, rep(1.54, 365), lower.tail = FALSE, qcov = v)
R> ciEffRL100 <- ci(fitFCpp3, return.period = 100, qcov = v)
```

To see how the PP model varies during the year, we can plot the above results using the code below.

```
R> plot(Fort$tobs, Fort$Prec, ylim = c(0, 10), xlab = "day",
+    ylab = "Fort Collins Precipitation (inches)")
R> lines(ciEffRL100[, 1], lty = 2, col = "darkblue", lwd = 1.25)
R> lines(ciEffRL100[, 2], col = "darkblue", lwd = 1.25)
R> lines(ciEffRL100[, 3], lty = 2, col = "darkblue", lwd = 1.25)
R> legend("topleft", legend = c("Effective 100-year return level",
+    "95% CI (normal approx)"), col = "darkblue", lty = c(1, 2),
+    lwd = 1.25, bty = "n")
```

Figure 17 shows the results from the code above, where it can be seen that the peak is in July.

6. Dependent sequences

One way to check whether or not there exists temporal dependence in threshold excess data is to look at the auto-tail dependence function plot (Reiss and Thomas 2007, Equations 2.65
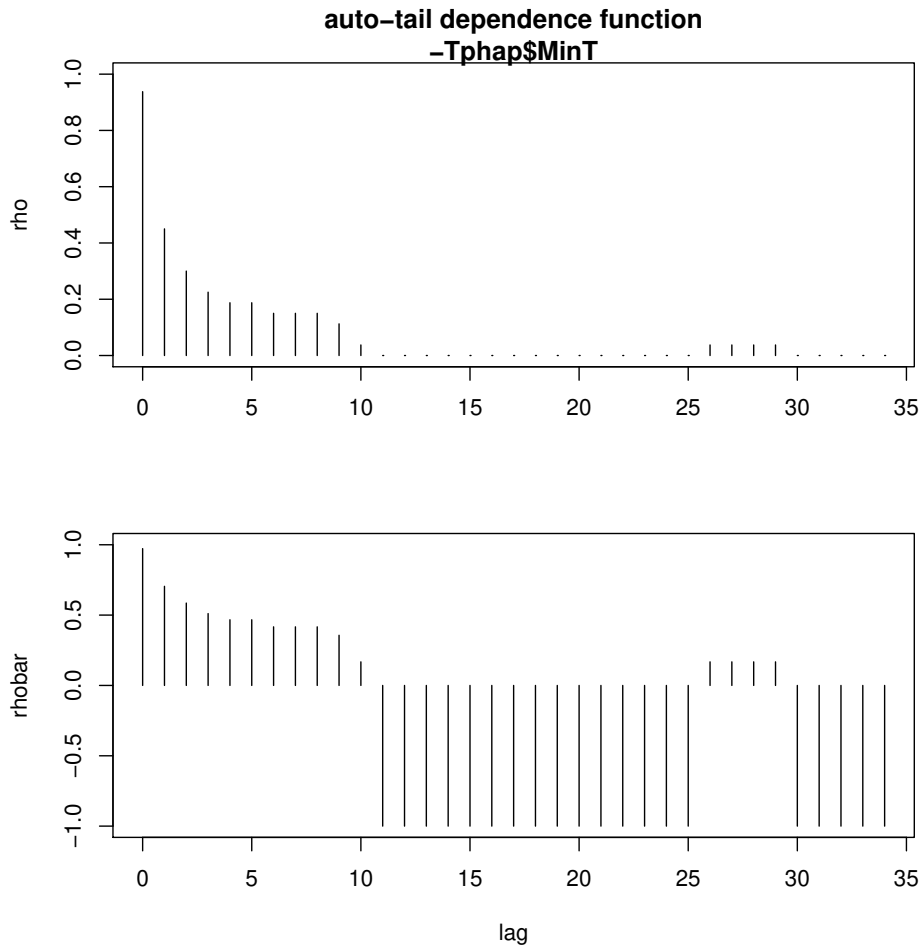
Figure 18: Auto-tail dependence function for Phoenix Sky Harbor (negative) minimum temperature (Figure 14) using the upper 0.99 quantile. On the top is the function based on $\hat{\rho}$ (Reiss and Thomas 2007, Equation 2.65), and on the bottom it is based on $\hat{\bar{\rho}}$ (Reiss and Thomas 2007, Equation 13.28).

and 13.28). Here, we will use the Phoenix Sky Harbor data as an example. Such a graphic can be made using **extRemes** by way of the `atdf` function. It is necessary to specify a threshold in the form of a quantile (between zero and one) that should be high enough to capture only the extremes, but low enough to include sufficient data.

```
R> atdf(-Tphap$MinT, 0.99)
```

Inspection of Figure 18 suggests that auto tail dependence exists in this series as the first few lags have estimated coefficients well above $1 - 0.9 = 0.1$ and zero, respectively. One issue that presents itself in the figure is the presence of numerous values equal to -1, which may be caused by seasonality, which is being ignored here. It is also possible to estimate the strength of tail dependence using the function **extremalindex**. For this, we will use the same non-constant threshold as applied in the last example of the previous section. Two methods are available for estimating the extremal index: intervals (default; Ferro and Segers 2003) and runs (Coles 2001). The former method also estimates the optimal run length for runs
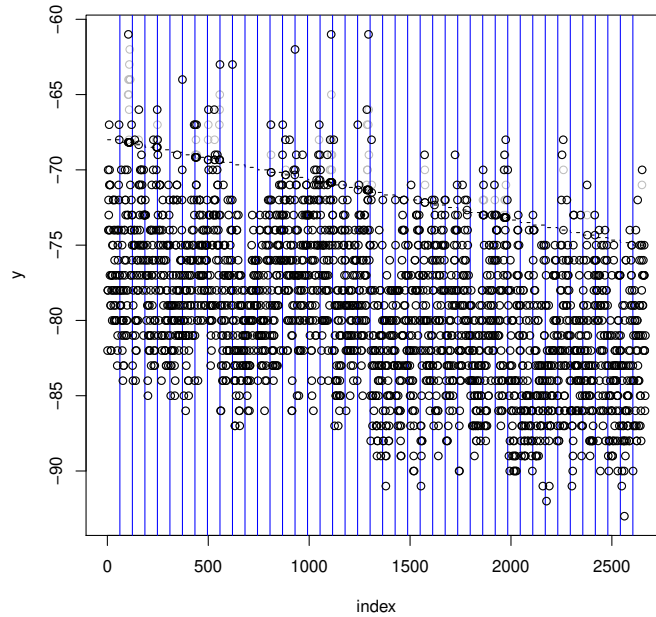
Figure 19: De-clustered (negative) Phoenix minimum temperature data (Figure 14) by runs declustering with a run length of 1. Vertical blue lines represent groups in which declustering was performed and each new group forces the start of a new cluster. Gray symbols above the threshold indicate that the value is replaced to that of the threshold (i.e., will not be used in estimating the EV df parameters).

declustering (an approach for handling tail dependence when estimating threshold excess and PP models).

```
R> extremalindex(-Tphap$MinT, -68 - 7 * (Tphap$Year - 48) / 42)
```

The above code yields an estimated extremal index of about 0.40, which is suggestive of strong tail dependence (in agreement with our findings in Figure 18). It also shows that there are sixty clusters with an estimated run length of about 5. Now that we have evidence of tail dependence, it is prudent to account for such dependence in our analysis. One way to do so is to try to remove the dependence by declustering the excesses. This can be done using the `decluster` function. Recall that the data is for summer months only, so it does not make sense to decluster data that cross over years. Therefore, we use the `groups` argument to specify that we want the decluster procedure to apply within each year only.

```
R> y <- -Tphap$MinT
R> u <- -68 - 7 * (Tphap$Year - 48) / 42
R> grp <- Tphap$Year - 47
R> look <- decluster(y, threshold = u, groups = grp)
R> look
R> plot(look)
```

The above code used runs declustering (default) with a run length of 1. It is also possible to decluster according to the intervals method proposed by Ferro and Segers (2003), which is a method for estimating the run length, and then applying runs declustering.

auto−tail dependence function
look2

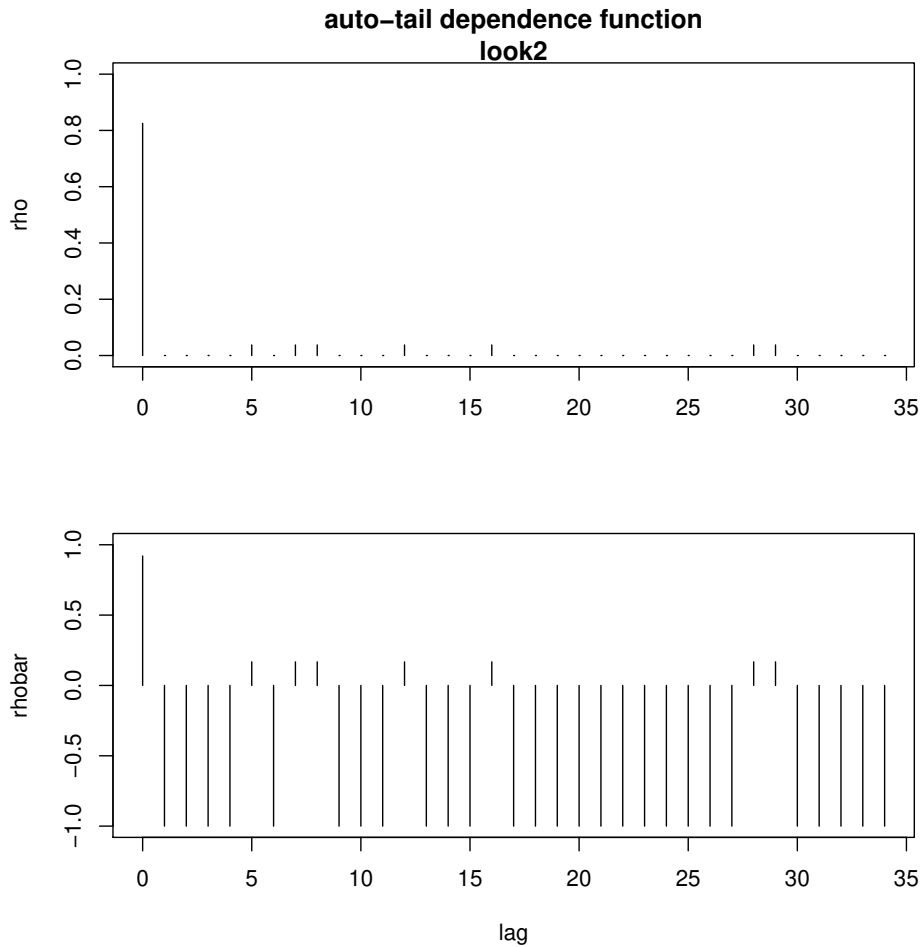Figure 20: Auto-tail dependence function for the declustered series from Figure 19.

```
R> look2 <- decluster(y, threshold = u, method = "intervals", groups = grp)
R> look2
R> plot(look2)
```

Note that now a run length of 5 is used, which we could have provided in our first decluster because we had already estimated the run length to be 5 from the intervals method when we called extremalindex. Note that the new extremal index estimate is around 0.78, indicating that the declustered series is not as strongly dependent in the tails. The new series contained in the object look2 can be applied directly to other functions as it is a kind of vector object. For example, we can check the auto-tail dependence plot.

```
R> atdf(look2, 0.99)
```

The declustered data show considerably less, if any, tail dependence; although the seasonality issue remains. It is possible to use the object look2 directly in a new call to fevd, which is shown below. Note that even though fevd generally prefers a data frame object to be passed, it is possible to run it without such an object, which is how it is called below. In order to allow for a varying threshold without a data frame object, we must first create a vector of the values rather than simply supplying a formula as done previously.

```
R> u <- (Tphap$Year - 48) / 42
R> u <- -68 - 7 * u
R> fitDC <- fevd(y, data = data.frame(y = c(look), time = 1:2666),
+    threshold = u, location.fun = ~ time, type = "PP",
+    time.units = "62/year")
R> plot(fitDC)
```

Note the use of `c(look2)` in the above code. The `c` function removes all of the attributes associated with the declustered object so that it can be entered into a data frame object. Inspection of the plot produced above (not shown) shows that the Z plot is now considerably straighter, although it appears that some issues may remain.

The Z plot (and qq-plot) can be re-created with the internal function `eeplot`. The following code demonstrates this procedure, and also plots the Z_k (and W_k; i.e., the transformed values to the standard exponential scale from the qq-plot) values against the times of exceeding the threshold. This latter plot can diagnose if there are any trends in time. For brevity, we do not show these plots here, but no apparent trends through time are evident.

```
R> par(mfrow = c(2, 2))
R> tmp <- eeplot(fitDC)
R> plot(tmp)
```

7. Dependence in extremes between two variables

In the previous section, tail dependence was investigated for a single variable over time. Similar analyses are available in **extRemes** when determining whether or not two variables are tail dependent. The primary functions for this type of analysis are `taildep` and `taildep.test`. As an example, we will simulate two highly correlated normally distributed series, and then check to see if they are strongly dependent in their tails or not.

```
R> z <- matrix(rnorm(2000), ncol = 2)
R> rho <- cbind(c(1, 0.9), c(0.9, 1))
R> rho <- chol(rho)
R> t(rho) %*% rho
R> z <- t(rho %*% t(z))
R> cor(z[, 1], z[, 2])
R> plot(z[, 1], z[, 2])
R> taildep(z[, 1], z[, 2], 0.99)
R> taildep.test(z[, 1], z[, 2])
```

8. Advanced

8.1. Optimization options

Sometimes it is helpful to modify the default arguments to the numerical optimization routines. This can be done in `fevd` by way of the `optim.args` argument. Here, we switch to

the `"BFGS"` method and use finite differences instead of the exact gradients in our fit to the negative of the minimum temperature data at Phoenix Sky Harbor Airport.[11]

```
R> fit <- fevd(-MinT ~ 1, Tphap, threshold = -73, type = "PP",
+    units = "deg F", time.units = "62/year", use.phi = TRUE,
+    optim.args = list(method = "BFGS", gr = NULL), verbose = TRUE)
```

8.2. Super-heavy tails

The asymptotic results that provide justification for use of the extreme value distributions do not guarantee that a non-degenerate limiting df exists for any given random variable. One example is the case of a random variable that follows a df with a super-heavy tail. The following code demonstrates one way of drawing a random sample from a super-heavy tail df, and results from attempting to fit these draws to a GP df.

```
R> z <- revd(1000, loc = 2, scale = 1.5, shape = 0.2)
R> z <- exp(z)
R> hist(z)
R> hist(z[z < 1000])
R> threshrange.plot(z)
R> fitZ <- fevd(z, threshold = exp(8), type = "GP")
R> plot(fitZ, "qq")
R> threshrange.plot(log(z))
R> fitlZ <- fevd(log(z), threshold = 8, type = "GP")
R> plot(fitlZ, "qq")
```

Quantile-quantile plots for one realization of data simulated from the procedure above are displayed in Figure 21. In each case, the plots are for quantiles of a random draw from the GP df fit to the simulated `z` data against those for the actual simulated data `z`. Clearly, the fit to the data shown on the right (the same as the left, but after having taken the log transformation), is better. The data set simulated above is an example of data that follow a log-Pareto df (cf. Reiss and Thomas 2007; Cormann and Reiss 2009). In particular, `z` is simulated by drawing a random sample from a Pareto-type ($\xi > 0$) df, and then exponentiating the result.

---

[11]Some optimization routines do not utilize gradient information, in which case it is not necessary to specify `gr = NULL`. If `gr` is an option for a routine, then the default is to use the exact gradients; use `gr = NULL` to force finite differences, if desired, in such cases.
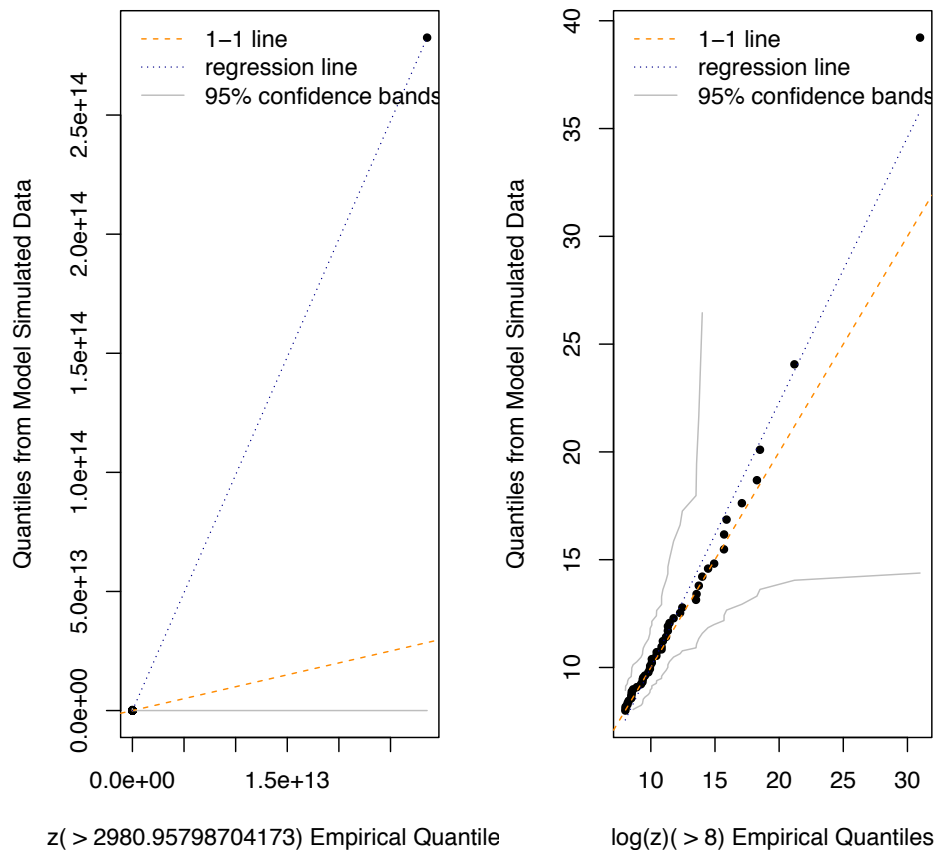
Figure 21: Quantile-quantile plots of GP model-simulated data vs. empirical quantiles of simulated super-heavy tailed data, `z` (left) and `log(z)` (right).

References

Balling R, Skindlov J, Phillips DH (1990). "The Impact of Increasing Summer Mean Temperature on Extreme Maximum and Minimum Temperatures in Phoenix, Arizona." *Journal of Climate*, **3**, 1491–1494. doi:10.1175/1520-0442(1990)003<1491:tioism>2.0.co;2.

Beirlant J, Goegebeur Y, Teugels J, Segers J (2004). *Statistics of Extremes: Theory and Applications*. Wiley Series in Probability and Statistics. John Wiley & Sons. doi:10.1002/0470012382.

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

Coles S, Heffernan J, Tawn J (1999). "Dependence Measures for Extreme Value Analysis." *Extremes*, **2**, 339–365. doi:10.1023/a:1009963131610.

Coles S, Pericchi L, Sisson S (2003). "A Fully Probabilistic Approach to Extreme Value Modelling." *Journal of Hydrology*, **273**, 35–50. doi:10.1016/s0022-1694(02)00353-0.

Cormann U, Reiss RD (2009). "Generalizing the Pareto to the Log-Pareto Model and Statistical Inference." *Extremes*, **12**, 93–105. doi:10.1007/s10687-008-0070-6.

Davison A, Smith R (1990). "Models for Exceedances over High Thresholds." *Journal of the Royal Statistical Society B*, **52**, 237–254.

De Haan L, Ferreira A (2006). *Extreme Value Theory: An Introduction*. Springer-Verlag. `doi:10.1007/0-387-34471-3`.

Diethelm Wuertz and many others (2013). **fExtremes**: *Rmetrics – Extreme Financial Market Data*. R package version 3010.81. URL `https://CRAN.R-project.org/package=fExtremes`.

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

Fox J, Weisberg S (2011). *An R Companion to Applied Regression*. 2nd edition. Sage Publications, Thousand Oaks. URL `http://socserv.socsci.mcmaster.ca/jfox/Books/Companion`.

Gilleland E (2013). **distillery**: *Method Functions for Confidence Intervals and to Distill Information from an Object*. R package version 1.0-1, URL `https://CRAN.R-project.org/package=distillery`.

Gilleland E (2015). **in2extRemes**: *Into the **extRemes** Package*. R package version 1.0-2, URL `https://CRAN.R-project.org/package=in2extRemes`.

Gilleland E (2016). **extRemes**: *Extreme Value Analysis*. R package version 2.0-8, URL `https://CRAN.R-project.org/package=extRemes`.

Gilleland E, Katz R (2011). "New Software to Analyze How Extremes Change Over Time." *Eos, Transactions American Geophysical Union*, **92**(2), 13–14. `doi:10.1029/2011eo020001`.

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`.

Heffernan J, Stephenson A (2012). **ismev**: *An Introduction to Statistical Modeling of Extreme Values*. R package version 1.39, Original S functions written by Janet E. Heffernan with R port and R documentation provided by Alec G. Stephenson, URL `https://CRAN.R-project.org/package=ismev`.

Heffernan J, Tawn J (2004). "A Conditional Approach for Multivariate Extreme Values." *Journal of the Royal Statistical Society B*, **66**(3), 497–546. `doi:10.1111/j.1467-9868.2004.02050.x`.

Hosking J (2014). **lmom**: *L-Moments*. R package version 2.4. URL `https://CRAN.R-project.org/package=lmom`.

Karvanen J (2006). "Estimation of Quantile Mixtures via L-Moments and Trimmed L-Moments." *Computational Statistics & Data Analysis*, **51**(2), 947–959. `doi:10.1016/j.csda.2005.09.014`.

Karvanen J (2011). **Lmoments**: *L-Moments and Quantile Mixtures. R*. R package version 1.1-3, URL `https://CRAN.R-project.org/package=Lmoments`.

Katz R (2002). "Stochastic Modeling of Hurricane Damage." *Journal of Applied Meteorology and Climatology*, **41**, 754–762. `doi:10.1175/1520-0450(2002)041<0754:smohd>2.0.co;2`.

Katz R, Parlange M, Naveau P (2002). "Statistics of Extremes in Hydrology." *Advances in Water Resources*, **25**, 1287–1304. `doi:10.1016/s0309-1708(02)00056-8`.

Leadbetter M, Lindgren G, Rootzén H (1983). *Extremes and Related Properties of Random Sequences and Processes.* Springer-Verlag.

Martins ES, Stedinger JR (2000). "Generalized Maximum-Likelihood Generalized Extreme-Value Quantile Estimators for Hydrologic Data." *Water Resources Research*, **36**(3), 737–744. `doi:10.1029/1999wr900330`.

Martins ES, Stedinger JR (2001). "Generalized Maximum-Likelihood Pareto-Poisson Estimators for Partial Duration Series." *Water Resources Research*, **37**(10), 2551–2557. `doi:10.1029/2001wr000367`.

Pfaff B, McNeil A (2012). **evir**: *Extreme Values in* R. R package version 1.7-3. URL `https://CRAN.R-project.org/package=evir`.

Pielke F, Landsea C (1998). "Normalized Hurricane Damages in the United States." *Weather and Forecasting*, **13**(3), 621–631. `doi:10.1175/1520-0434(1998)013<0621:nhditu>2.0.co;2`.

R Core Team (2016). R: *A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. URL `https://www.R-project.org/`.

Reiss RD, Thomas M (2007). *Statistical Analysis of Extreme Values: With Applications to Insurance, Finance, Hydrology and Other Fields.* 3rd edition. Birkhäuser.

Resnick S (1987). *Extreme Values, Point Processes and Regular Variation.* Springer-Verlag. `doi:10.1007/978-0-387-75953-1`.

Ribatet M (2009). *A User's Guide to the* **SpatialExtremes** *Package.* École Polytechnique Fédéral de Lausanne, Switzerland.

Ribatet M, Singleton R, R Core Team (2013). **SpatialExtremes**: *Modelling Spatial Extremes.* R package version 2.0-0. URL `https://CRAN.R-project.org/package=SpatialExtremes`.

Sibuya M (1960). "Bivariate Extreme Statistics, I." *The Annals of the Institute of Statistical Mathematics*, **11**(2), 195–210. `doi:10.1007/bf01682329`.

Smith R (1989). "Extreme Value Analysis of Environmental Time Series: An Application to Trend Detection in Ground-Level Ozone." *Statistical Science*, **4**, 367–393. `doi:10.1214/ss/1177012400`.

Smith R, Shively T (1995). "Point Process Approach to Modeling Trends in Tropospheric Ozone." *Atmospheric Environment*, **29**, 3489–3499. `doi:10.1016/1352-2310(95)00030-3`.

Southworth H, Heffernan J (2011). **texmex**: *Threshold Exceedences and Multivariate Extremes.* R package version 1.2. URL `https://CRAN.R-project.org/package=texmex`.

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

Stephenson A, Ribatet M (2014). ***evdbayes**: Bayesian Analysis in Extreme Value Theory*. R package version 1.1-1. URL https://CRAN.R-project.org/package=evdbayes.

Stigler S (1986). *The History of Statistics: The Measurement of Uncertainty Before 1900*. Belkap Press of Harvard University Press, Cambridge.

Tarleton L, Katz R (1995). "Statistical Explanation for Trends in Extreme Summer Temperatures at Phoenix, Arizona." *Journal of Climate*, **8**(6), 1704–1708. doi:10.1175/1520-0442(1995)008<1704:seftie>2.0.co;2.

Thompson D, Wallace J (1998). "The Arctic Oscillation Signature in the Wintertime Geopotential Height and Temperature Fields." *Geophysical Research Letters*, **25**, 1297–1300. doi:10.1029/98gl00950.

Wettstein J, Mearns L (2002). "The Influence of the North Atlantic-Arctic Oscillation on Mean, Variance and Extremes of Temperature in the Northeastern United States and Canada." *Journal of Climate*, **15**, 3586–3600. doi:10.1175/1520-0442(2002)015<3586:tiotna>2.0.co;2.

Affiliation:

Eric Gilleland
Joint Numerical Testbed
Research Applications Laboratory
National Center for Atmospheric Research
PO Box 3000
Boulder, Colorado, 80307, United States of America
E-mail: EricG@ucar.edu
URL: http://www.ral.ucar.edu/staff/ericg/